Fakultät für Physik und Astronomie
Ruprecht–Karls–Universität Heidelberg

Diplomarbeit

im Studiengang Physik

vorgelegt von

Lutz Christian Marquard

aus Bruchsal

1994

Anwendung von Filteroperatoren und Neuronalen Netzwerken zur Photon- und Ringerkennung für Ringbild-Cherenkov-Detektoren

Die Diplomarbeit wurde ausgeführt am

Physikalischen Institut

unter der Betreuung von

Herrn Prof. Dr. H.J. Specht.

Zusammenfassung

In dieser Arbeit wird der Einsatz von Techniken der Digitalen Bildverarbeitung und künstlicher Neuronaler Netzwerke bei der Mustererkennung im CERES/NA45-Experiment untersucht.
Zielsetzung des Experimentes ist das Studium von Kernmaterie bei hohen Energiedichten, wie sie in Kollisionen von ultrarelativistischen Schwerionen erreicht werden können. Das Experiment besteht im wesentlichen aus zwei ringabbildenden Čerenkov-Detektoren zur Teilchenidentifikation und Winkelmessung. Die in den Čerenkov-Zählern erzeugten Photonen werden durch UV-Detektoren nachgewiesen, die auf einem schachbrettartigen Padreadout Ladungssignale (Hits) induzieren. Von einem geladenen Teilchen emittierte Čerenkov-Photonen werden auf einen Čerenkov-Ring abgebildet. Neben den Hits, die von Čerenkov-Photonen herrühren, treten Untergrundsignale auf, die elektronischer oder physikalischer Natur sein können.
Zunächst wurde aus der Technik von Filteroperationen im Fourierraum ein neues, schnelles Konzept zur Suche nach Ringkandidaten entwickelt. Bei Verwendung einer Doppelringmaske geeigneter Breite stellt dieses Konzept in Bezug auf Ringfindeeffizienz, Untergrundrejektion und Laufzeit eine Alternative zur bisher durchgeführten Hough-Transformation dar. Das neue Konzept ist darüberhinaus sehr ausbaufähig. Es erlaubt die Berücksichtigung der Struktur der lokalen Bildumgebung und das Entwerfen von Filtern für einen spezifischen Zweck.
Im Hauptteil der Arbeit wurden erstmalig Neuronale Netzwerke zur Erkennung von einzelnen Hits, überlappenden Hits und Untergrund sowie zur Rekonstrukion der Hits entwickelt. Die Netzwerke sind in der Lage, über 90% der einzelnen und überlappenden Hits sowie gleichzeitig über 80% des Untergrundes richtig zu klassifizieren. Bei Verwendung dieser Netzwerke kann die begrenzte Doppelhitauflösung der bisherigen Analyse um 15% für den ersten und um 35% für den zweiten RICH-Detektor verbessert werden. Mit Hilfe der Netze wurden bisher unverstandene charakteristische Verteilungen der Hits nach den Beiträgen von Hits, überlappenden Hits und Untergrund aufgeschlüsselt. Die Anwendung der Netzwerke bei der Rekonstruktion der Ringe ergab im Vergleich zur bisherigen Analyse gleiche Ringfindeeffizienz, Untergrundrejektion und vergleichbare Laufzeit, erlaubt aber eine Vereinfachung und Kürzung (Straffung) der Analyse. Desweiteren ist eine genauere Rekonstruktion der Hits sowie eine genauere Bestimmung der Ringmittelpunkte möglich. Es stellte sich jedoch heraus, daß Backpropagation-Netzwerke zur Rekonstruktion der Hitposition ungeeignet sind. Somit konnte gezeigt werden, daß bereits ein einfacher Netzwerktyp wie das Multilayer-Feedforward-Netzwerk erfolgreich bei der Analyse eingesetzt werden kann.

Inhaltsverzeichnis

Einleitung

Die Hochenergiephysik bietet eine Vielzahl herausfordernder Probleme der Mustererkennung. Die Standardmethoden zur Lösung solcher Probleme sind typischerweise auf lineare Abhängigkeiten beschränkt, wie z.B. Fischer-Diskriminanten, Hauptkomponenten-Analysen oder ARMA-Modelle. Künstliche Neuronale Netzwerke stellen leistungsfähige nicht-lineare Erweiterungen der konventionellen Methoden dar. Insbesondere Multilayer-Feedforward-Netzwerke haben sich aufgrund ihrer Einfachheit und enormen Leistungsfähigkeit in den letzten Jahren in einem weiten Feld von Anwendungen immer weiter verbreitet. Die Hochenergiephysik ist hierbei keine Ausnahme. Feedforward-Netzwerke sind nicht nur auf dem Gebiet der Mustererkennung, sondern generell auf dem Gebiet der Implementierung von Abbildungen sehr leistungsfähig. Durch die rasante Entwicklung der Computertechnologie in den letzten Jahren hat die Informationsverarbeitung, bis dahin Spezialdisziplin in Gebieten der Astronomie, Fernerkundung, Nachrichtentechnik und Informatik, einen enormen Auftrieb erfahren. Dieser Fortschritt erlaubt nun ihren breiten Einsatz als Hilfsmittel zur Auswertung in allen Bereichen der Naturwissenschaften. Damit ist die Grundlage für neue interdisziplinäre Forschungsrichtungen gelegt, die das betreffende Fachgebiet und die Informationsverarbeitung miteinander verknüpfen. Bei der Mustererkennung im CERES-Experiment sind begrenzte zweidimensionale digitalisierte Bilder, also Bildmatrizen endlicher Größe zu verarbeiten.
Das CERES/NA45-Experiment gehört zur zweiten Generation von Schwerionen-Experimenten am CERN-SPS und wurde zur Messung von e^+e^--Paaren in Proton-Kern- und Kern-Kern-Kollisionen entwickelt. Das Experiment besteht im wesentlichen aus zwei ringabbildenden Čerenkov-Detektoren zur Teilchenidentifikation und Ortsmessung. Die in den Čerenkov-Zählern erzeugten Čerenkov-Photonen werden mit Hilfe von UV-Detektoren dadurch nachgewiesen, daß auf einem zweidimensionalen, schachbrettartigen Padreadout Ladungssignale (Hits) induziert werden. Unter einem festen Winkel entlang der Teilchentrajektorie emittierte Čerenkov-Photonen werden auf einen Čerenkov-Ring abgebildet. Die Bildmatrizen haben eine Größe von 288×288 bzw. 286×286 Pixeln. Neben den Hits, die von Čerenkov-Photonen herrühren, treten Untergrundsignale auf, die elektronischer - i.w. elektronisches Rauschen - oder physikalischer Natur - z.B. ionisierende Teilchen, die

die UV-Zähler durchqueren - sein können. Eine ausführliche Beschreibung des CERES-Experimentes findet sich in Kapitel 1.

Im ersten Teil der vorliegenden Arbeit wird der Einsatz von Methoden der digitalen Bildverarbeitung zur Erkennung von Čerenkov-Ringen untersucht. Im einzelnen wird in Kapitel 2 aus der Strategie der Berechnung von Filteroperationen im Fourierraum ein neues, schnelles Konzept zur Identifizierung von Ringkandidaten entwickelt. Die Anwendung dieses Konzeptes auf Daten und der Vergleich mit dem bisherigen Verfahren erfolgt dann in Kapitel 3.

Im zweiten, wesentlich größeren Teil der Arbeit wird erstmalig der Einsatz von Neuronalen Netzwerken bei der Mustererkennung auf Hitniveau untersucht. Genauer werden Neuronale Netzwerke zur Klassifizierung von einzelnen Hits, überlappenden Hits und Untergrund sowie zur Rekonstruktion der Hits entwickelt und eingesetzt. Die Motivation hierfür, die den z.T. schwerwiegenden Problemen des bisherigen Analyseverfahrens entspringt, ist in Kapitel 1 dargelegt. In Kapitel 4 wird die mathematisch-theoretische Grundlage der Neuronalen Netzwerke vorgestellt; insbesondere werden die Fragen nach der Funktionsweise, der Implementierbarkeit und Lernbarkeit von Funktionen sowie dem Verhalten bei der Generalisierung erörtert. Die Entwicklung der Netzwerke zur Lösung der gestellten Aufgaben - Netzwerksimulation, Erzeugung der Lern- und Testdaten, Netzwerkarchitektur, optimaler Satz an Input-Variablen, Trainingsablauf - wird in Kapitel 5 beschrieben; ebenso werden erste Anwendungen dieser Netzwerke (Rejektion der verschiedenen Untergrundklassen, Doppelhiterkennung, Trennung von Daten und Monte-Carlo-simulierten Daten) behandelt. Die Anwendung der entwickelten Netzwerke auf Daten wird dann in Kapitel 6 untersucht. Neben dem Testen der Netzwerke auf Daten wird unter ihrer Verwendung ein neues Hitanalyseverfahren entwickelt und bei verschiedenen quantitativen Analysen der Hits - Polarwinkelverteilung der Hits, Verteilung der Hitamplituden, Anzahl der Hits in Abhängigkeit von der Multiplizität, Anzahl der Hits pro Ring - eingesetzt. Dann wird unter Verwendung der Netze ein neues Analyseverfahren zur Rekonstruktion von Ringen entwickelt und mit dem bisherigen Verfahren verglichen. Schließlich wird der Einsatz Neuronaler Netzwerke zur Rekonstruktion der Hitpositionen untersucht. In Kapitel 7 schließt die Arbeit mit einer kritischen Diskussion der Ergebnisse sowie einem Ausblick auf mögliche Verbesserungen und Weiterentwicklungen der Mustererkennung im CERES-Experiment.

Kapitel 1

Das CERES-Experiment

Das CERES/NA45-Experiment[1] gehört zur zweiten Generation von Schwerionen-Experimenten am Super Proton Synchrotron (SPS) des CERN.
Ziel des Experimentes ist die systematische Untersuchung der Produktion von Elektron-Positron-Paaren in Nukleon-Kern- und Kern-Kern-Kollisionen im Bereich invarianter Massen von 50 MeV/c^2 bis über 2 GeV/c^2 bei zentralen Rapiditäten im Bereich $2 < \eta < 2.6$ [1]. Neben der Messung von Kontinuumspaaren und Vektormesonen [2] gehören zum weiteren Themenkreis die Messung von Photonen über die Konversionsmethode [3], die Untersuchung von Pionen mit hohen Transversalimpulsen $p_\perp > 0.8$ GeV/c [4], sowie die Messung von e^+e^--Paaren aus $\gamma\gamma$-Reaktionen in distanten Kern-Kern-Kollisionen [5].

Die endgültige Installation dieses Elektronpaar-Spektrometers erfolgte im Jahre 1990 in der Nordhalle des CERN-Geländes am H8-Strahlrohr. Im Herbst desselben Jahres wurden die ersten Testdaten aufgezeichnet. Die in dieser Arbeit verwendeten Datensätze wurden während der Schwefel-Strahlzeit im Frühjahr 1992 aufgenommen.

In diesem Kapitel wird nur auf die für diese Arbeit relevanten experimentellen Grundlagen eingegangen. Eine umfassende Beschreibung des Detektors und seiner einzelnen Komponenten zum Zeitpunkt der Strahlzeit 1992 wurde bereits gegeben [2]. Die Probleme des bisherigen Analyseverfahrens des CERES-Experimentes werden ausführlich dargestellt, weil sich daraus die Aufgabenstellungen (Motivation) für die vorliegende Arbeit ergeben.

[1] CERES ist die Abkürzung von Čerenkov Ring Electron Spectrometer.

1.1 Der experimentelle Aufbau

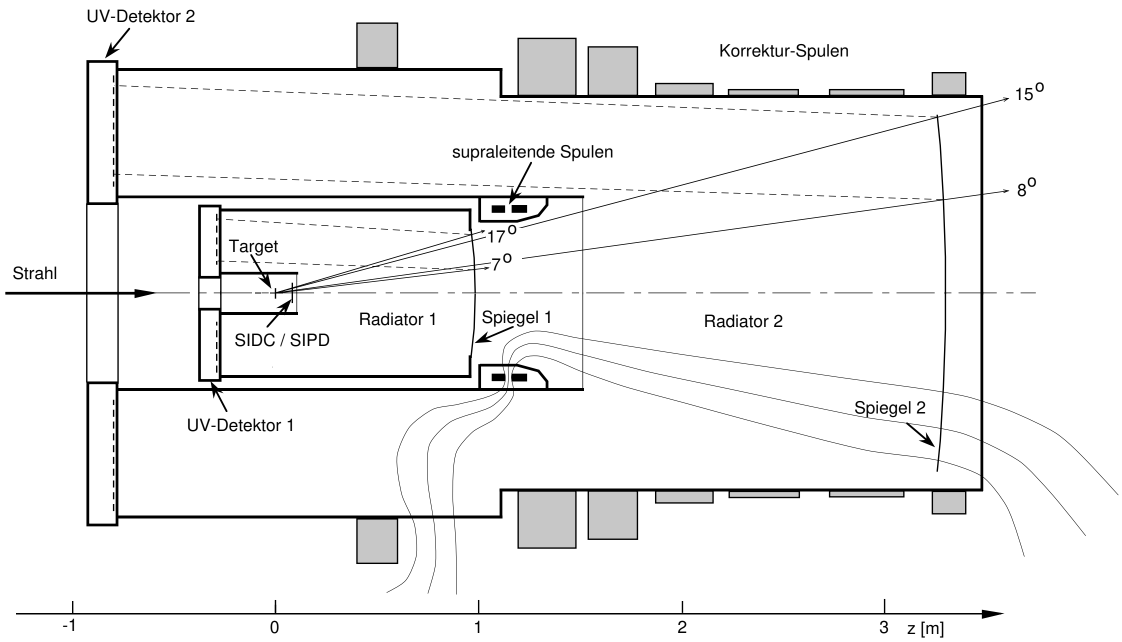

Abbildung 1.1: *Querschnitt des CERES-Experimentes parallel zur Strahlachse.* Der Verlauf der Magnetfeldlinien ist in der unteren Hälfte schematisch eingezeichnet.

In Abb. 1.1 ist ein Querschnitt des Spektrometers parallel zur Strahlachse zu sehen. Das Experiment besteht im wesentlichen aus zwei azimutalsymmetrischen ringabbildenden Čerenkov-Detektoren (**RICH**[2] 1/2) zur Teilchenidentifikation und Ortsbestimmung, die vor (RICH 1) und hinter (RICH 2) einer supraleitenden Doppelspule angebracht sind. Es ist das erste Experiment, das eine vollständige Ereignisrekonstruktion mit Hilfe von RICH-Detektoren durchführt.
Das Prinzip des RICH-Detektors wurde 1977 von *Seguinot* und *Ypsilantis* [6] vorgeschlagen. Es besteht darin, einen sphärischen Spiegel so anzuordnen, daß sich in dessen Krümmungsmittelpunkt das Target und in dessen Fokalfläche ein ortsempfindlicher Detektor befindet. Der Raum zwischen Detektor und Spiegel ist mit einem Radiator vom Brechungsindex n gefüllt, in dem geladene Teilchen Čerenkov-Photonen unter dem Winkel

$$\Theta_c = \arccos\left(\frac{1}{n\beta}\right)$$

emittieren, falls die Bedingung $\beta > 1/n$ erfüllt ist. Im Falle von hochrelativistischen Elektronen ist es sinnvoll, statt der Geschwindigkeit β den Lorentzfaktor γ

[2]RICH ist die Abkürzung von Ring Imaging Cherenkov.

einzuführen. Es ergibt sich für den Schwellenwert

$$\gamma_{th} = \left(1 - \frac{1}{n^2}\right)^{-\frac{1}{2}} .$$

Die unter einem festen Winkel entlang der Teilchentrajektorie emittierten Photonen werden in der Fokalfläche auf einen Ring abgebildet, den man als **Čerenkov-Ring** bezeichnet. Bei konstantem Brechungsindex ist der Ringradius ein Maß für die Geschwindigkeit des Teilchens und kann somit zur Teilchenidentifikation eingesetzt werden. Für den Öffnungswinkel im Limes $\gamma \to \infty$ ergibt sich näherungsweise die einfache Beziehung

$$\Theta_\infty \approx \frac{1}{\gamma_{th}} ,$$

die nur für $n \to 1$ exakt ist. Der Radius des abgebildeten Ringes ist für $\gamma \geq \gamma_{th}$ durch

$$R = R_\infty \left(1 - \left(\frac{\gamma_{th}}{\gamma}\right)^2\right)^{-\frac{1}{2}}$$

gegeben; für die Anzahl der Photonen auf dem Ring gilt

$$N = N_\infty \left(1 - \left(\frac{\gamma_{th}}{\gamma}\right)^2\right) .$$

Die im CERES-Experiment verwendeten RICH-Detektoren weichen vom klassischen RICH-Prinzip insofern ab, als durch Verschiebung des Targets aus dem Krümmungsmittelpunkt des Spiegels eine fast ebene Bildfläche strahlaufwärts vom Target entsteht.

Die in den RICH-Detektoren erzeugten Čerenkov-Photonen werden durch **UV-Detektoren** nachgewiesen, die folgende Funktionen erfüllen: die Konversion der eintretenden Photonen in Photoelektronen, die Verstärkung der Photoelektronen durch Lawinenbildung im Zählergas und die zweidimensionale Auslese der Ladungssignale durch eine hochgranulare, in Metall**pads** segmentierte Rückwand. Ein in dieser Weise induziertes Ladungssignal, das von einem Čerenkov-Photon herrührt, bezeichnet man als **Hit**.
Die Zählerrückwand besteht aus 53800 (48400) schachbrettartig angeordneten Pads mit einem Rastermaß von 2.74 (7.62) mm[3]. Der Radius eines asymptotischen Čerenkov-Ringes beträgt 14.5 (16.4) Pads oder 4 (12.5) cm. Das kleinste quadratische Raster aus Pads, das den Akzeptanzbereich des RICH-Detektors abdeckt, ist von der Größe 288×288 (286×286). Die Amplitude eines Pads wird in Einheiten

[3] Die Werte in Klammern beziehen sich auf RICH 2.

von 3000 (2200) Elektronen mit Werten zwischen 0 und 255 gemessen. Die Größe eines Pads entspricht einem Winkel von 2.18 (1.82) mrad. Die Ausleseelektronik (**Padreadout**) ist direkt auf die Rückwand des Detektors aufgesteckt.

Die Form der Hits wird durch die influenzierte Ladungsverteilung und die Diffusion der Elektronen in den Zählern bestimmt. Die resultierende Verteilung läßt sich nach *Endo* [7] durch eine unendliche Reihe approximieren, die in graphischer Darstellung einen glockenförmigen Verlauf hat. Die eindimensionale Projektion dieser unendlichen Reihe ist proportional $\cosh^{-1} ax$. Aufgrund der Größe der Pads wird aber kein kontinuierliches Signal beobachtet, sondern die auf den einzelnen Pads induzierte Ladung, die gleich dem Integral der kontinuierlichen Verteilung über die Fläche des Pads ist. Wie in Anhang B gezeigt wird, stellt eine zweidimensionale Gaußverteilung eine sehr gute Approximation der tatsächlichen Ladungsverteilung dar. Die gemessene Breite beträgt FWHM = 1.90 (1.48) Pads oder $\sigma = 0.81$ (0.63) Pads. Die **Amplitude eines Hits** ist definiert als die Summe der Amplituden aller zum Hit gehörenden Pads. Die Hitamplituden haben aufgrund des Verstärkungsprozesses in den UV-Detektoren ein exponentiell abfallendes Spektrum.

1.2 Die Mustererkennung

Abb. 1.2 zeigt eine Darstellung des Padreadouts für ein typisches S-Au-Ereignis in RICH 1 und RICH 2. Es lassen sich verschiedene Arten von Signalen oder **Mustern** unterscheiden:

1. **Physikalische Signale**:

   hiermit sind die Signale von Čerenkov-Photonen, also einzelne und überlappende Hits gemeint. Ringe sind somit auch physikalische Signale, da sie aus Hits bestehen.

2. **Untergrundsignale**:

   Elektronischer Untergrund: hierbei handelt es sich im wesentlichen um gaußverteiltes Rauschen, also um einzelne Pads, deren Amplituden gaußverteilt schwanken. Elektronischen Ursprungs sind auch Nullpunktsverschiebungen ganzer Module. Ferner treten noch vertikale Streifen auf, bei denen immer nur jedes zweite Pad aufleuchtet.
   Physikalischer Untergrund: das sind zum einen **Cluster**, die von ionisierenden Teilchen herrühren, die die UV-Zähler durchqueren oder in ihnen gestoppt werden und lokal hohe Ladungsdichten erzeugen. Cluster können sowohl auf wenige Pads begrenzt sein und Hits großer Amplitude ähneln, als auch größere Gebiete bis hin zu mehreren hundert Pads abdecken. Zum anderen handelt es sich um Nachbilder vorhergehender Ereignisse, deren Rest-

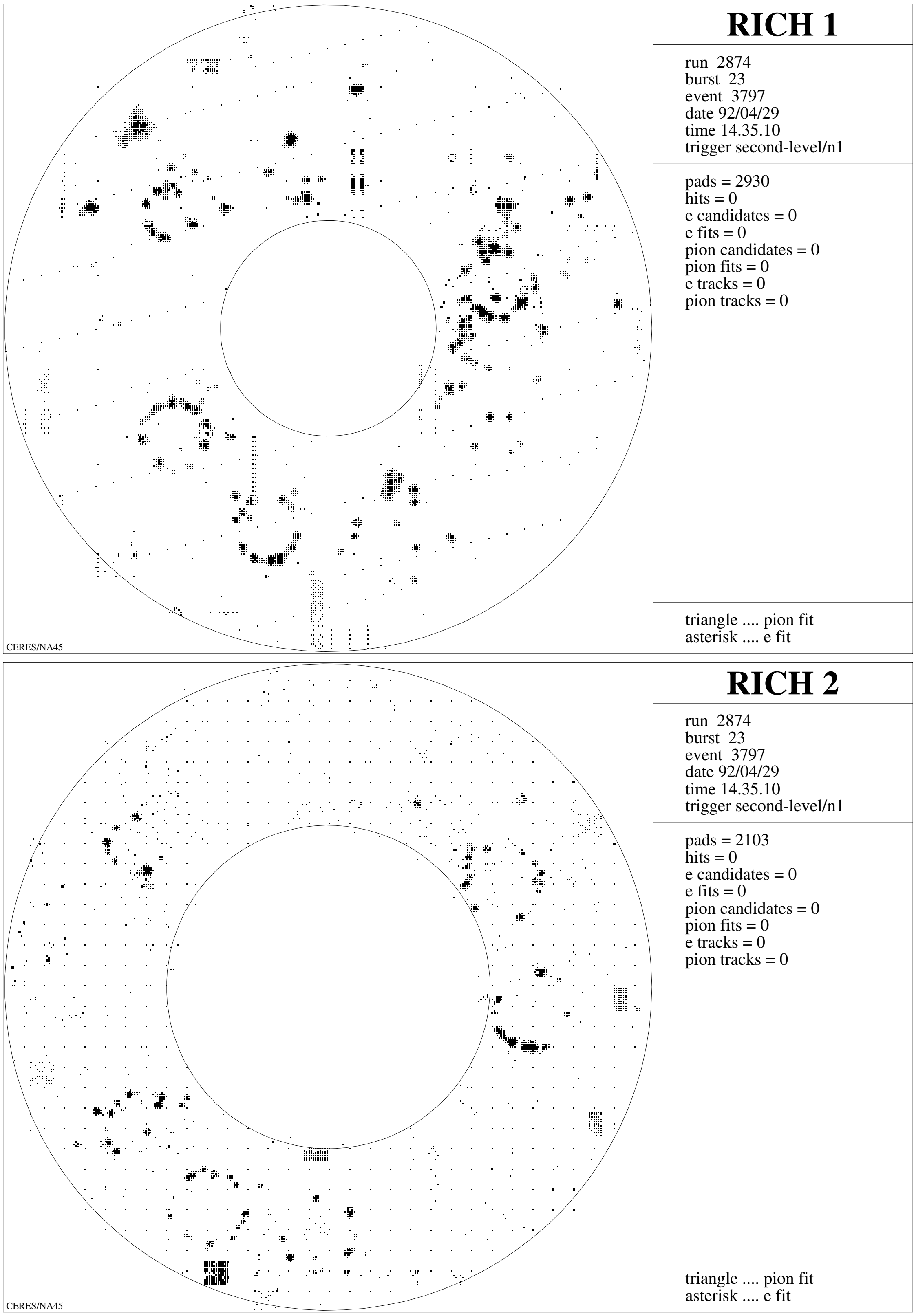

Abbildung 1.2: *Ein typisches S-Au-Ereignis in RICH 1 und RICH 2.* Deutlich sind die im Text beschriebenen verschiedenen Physik- und Untergrundsignale zu erkennen.

ladungen auf der hochohmigen Widerstandsschicht des Detektors noch nicht abgeflossen sind. Diese Signale weisen sehr kleine Amplituden auf und sind i.a. breiter als „aktuelle" Signale.

Das bisherige Analyseverfahren des CERES-Experimentes zur **Rekonstruktion der Elektronringe**, d.h. zur Bestimmung der Mittelpunkte von Čerenkov-Ringen, die von einzelnen Elektronen herrühren, gliedert sich wie folgt:

1. Anwendung von Säuberungsalgorithmen zur Rejektion des elektronischen Rauschens bzw. der Cluster.
2. Suche der Ringkandidaten.
3. Rekonstruktion der Hits.
4. Anwendung eines Ringfites auf die für jeden Ringkandidaten rekonstruierten Hits.
5. Anwendung von Qualitätsschnitten auf die Ringe.

Eine detailierte Beschreibung der bei den einzelnen Analyseschritten angewendeten Algorithmen wurde bereits mehrfach gegeben [2], [3], [4], [5].

Dieses Analyseverfahren, insbesondere die Rekonstruktion der Hits, weist z.T. schwerwiegende Probleme auf: generell gilt, daß entweder die Effizienz, einen Hit zu finden, oder die Rejektion des Untergrundes zu klein ist. Zur Trennung der Hits vom Untergrund sind drei verschiedene Algorithmen notwendig. Die einzelnen Analysealgorithmen greifen z.T. auf komplizierte konventionelle Techniken wie den *Chaincode* [8] oder den *ILRM-Fit* [9] zurück. Der Säuberungsalgorithmus zur Rejektion der Cluster ist nicht in der Lage, Clustern überlagerte Hits zu erhalten, wenn die Amplitude dieser Hits einen bestimmten Schwellenwert unterschreitet.
Besonders gravierend sind die folgenden Probleme des bisherigen Hitrekonstruktionsalgorithmus:

- Die im Algorithmus durchgeführte Schwerpunktsbildung erlaubt es nicht, die Mittelpunkte von zwei oder mehr sich überlappenden Hits exakt zu bestimmen, da die Amplitude eines Pads, das zu mehr als einem Hit gehört, nicht mehr zwischen diesen Hits aufgespalten werden kann; die erhaltenen Schwerpunkte geben also nicht die wahre Position wieder, sondern sind zueinander hin verschoben. Desweiteren ist die durchgeführte Schwerpunktsbildung aufgrund der endlichen Größe der Pads und der endlichen Anzahl benutzter Pads mit einem systematischen Fehler behaftet, der am größten ist, wenn daß Hitzentrum zwischen zwei Pads liegt.

- Aufgrund der Breite der Hits werden in der im Algorithmus verwendeten 3×3-Matrix im Mittel nur 85% (94%) der gesamten Ladung eines Hits erfaßt (siehe Anhang B). Hieraus ergibt sich folgende wichtige Konsequenz:
  Hits großer Amplitude oder nicht rejizierte Cluster werden zum Teil in viele einzelne „Hits" aufgespalten (*Hitsplitting*).

- *Der Algorithmus ist nicht sensitiv auf die Form der Hits*, sondern einzig auf Muster, deren Schwerpunkte im zentralen Pad der verwendeten 3×3-Matrix liegen. Dadurch werden auch nicht-gaußförmige Muster als Hits akzeptiert.

- *Es gibt eine dem Algorithmus immanente Grenze der Doppelhitauflösung.* Der Algorithmus ist nämlich nicht in der Lage, Hits zu trennen, die einen Abstand ≤ 1 Pad haben. Dies führt dazu, daß ca. 25% aller Hits auf Ringen durch Überlappung verlorengehen, bei vollständig oder nahezu vollständig überlappenden Ringen sind es sogar bis zu 40%(*Pileup*).
  Da die Form der Hits bekannt ist, ist die theoretische Grenze der Doppelhitauflösung durch die begrenzte Rekonstruktionsgenauigkeit gegeben, die durch die Digitalisierung hervorgerufen wird. (Ist die Form der Signale nicht bekannt, so ist die theoretische Grenze der Doppelpulsauflösung gleich der Halbwertsbreite der Signale.) Um untere Grenzen für die Doppelhitauflösung zu erhalten, wurden durch eine Monte-Carlo-Simulation zweidimensional gaußverteilte Hits mit den gemessenen Amplituden- und Breitenverteilungen erzeugt und die Positionen der Hits sowohl durch Anpassung einer gaußförmigen Ladungsverteilung als auch durch Schwerpunktsbildung in der 3×3-Matrix bestimmt. Desweiteren wurde die Bestimmung der Hitpositionen durch Schwerpunktsbildung für Hits durchgeführt, die mit gaußförmig verteiltem Rauschen mit einem Signal-zu-Rausch-Verhältnis von etwa 50 (RICH 1) bzw. 100 (RICH 2) überlagert waren. Die für die Rekonstruktionsgenauigkeiten erhaltenen Werte sind in folgender Tabelle angegeben:

Verfahren	Rekonstruktionsgenauigkeit für Hits [Pads]	Rekonstruktionsgenauigkeit für Hits + Rauschen [Pads]
Anpassung einer Gaußverteilung	0.025 ± 0.003	-
Schwerpunktsbildung in 3×3-Matrix	0.072 ± 0.005	0.15 ± 0.02 (RICH 1), 0.07 ± 0.01 (RICH 2)

Es ist nun grundsätzlich interessant, Ansätze und Methoden der **Digitalen Bildverarbeitung** (für einen Überblick siehe *Jähne* [10]) wie **Unitäre Transformationen** - insbesondere die **Diskrete Fouriertransformation** -, **Filteroperationen**, **Segmentierungsverfahren** oder **Elastic Tracking** (siehe *Gyulassy et al.* [11]) bei der Mustererkennung des CERES-Experimentes einzusetzen und zu untersuchen, ob sie bezüglich Funktionalität (Effizienz, Rejektion) und Geschwindigkeit dem bisherigen Analyseverfahren überlegen sind. Die diskrete Fouriertransformation und die Technik der Filteroperationen wurden im ersten Teil der vorliegenden Arbeit bei der Suche nach Ringkandidaten eingesetzt und mit dem bisherigen Verfahren verglichen. Die Methode der Filteroperationen wurde angewendet, weil sie in der Digitalen Bildverarbeitung das grundlegende Verfahren zur Erkennung von Bildstrukturen darstellt und sich mit ihr sehr unterschiedliche Operationen durchführen lassen.

Da die Probleme des bisherigen Analyseverfahrens hauptsächlich bei der Rekonstruktion der Hits zu finden sind, ist es von vorrangigem Interesse, die Erkennung von Hits und Untergrund sowie die Bestimmung der Hitpositionen zu verbessern. Ein verbesserter Algorithmus zur Hitrekonstruktion, der die oben angeführten Probleme nicht mehr aufweisen würde, würde insbesondere die Doppelhitauflösung, die Doppelringauflösung sowie die quantitativen Analysen der Hits erheblich verbessern. Neuronale Netzwerke haben sich auf dem Gebiet der Mustererkennung als sehr leistungsfähige Instrumente erwiesen, die gerade auch in vielen Bereichen der Hochenergiephysik mit großem Erfolg eingesetzt worden sind (für einen Überblick siehe *Denby* [12]). Deshalb liegt dem zweiten, wesentlich größeren Teil dieser Arbeit die Aufgabenstellung zugrunde, software-implementierte Neuronale Netzwerke zu entwickeln, die zur Klassifizierung von einzelnen Hits, überlappenden Hits und Untergrund sowie zur Rekonstruktion der Hits dienen sollen. Ziel ist es, die Effizienz der Erkennung von einzelnen Hits, überlappenden Hits und Untergrund zu steigern sowie die bislang komplexe Aufeinanderfolge einzelner Analysealgorithmen zu vereinfachen oder gar zu ersetzen.

Kapitel 2

Theorie der Filteroperatoren

Die Fourier-Transformation ist ein wichtiges Hilfsmittel der Digitalen Bildverarbeitung. Eine Matrix im Ortsraum ist nur eine von vielen Möglichkeiten, die Information, die ein Bild enthält, darzustellen. Diskrete unitäre Transformationen führen die Bilddarstellung im Ortsraum in eine andere äquivalente Darstellung über. In diesem Kapitel wird die diskrete Fourier-Transformation als Vertreter der unitären Transformationen gewählt, weil durch sie die Filteroperationen, die im Ortsraum Faltungen von Matrizen sind, auf einfache Matrizenmultiplikationen im Fourierraum zurückgeführt werden können. Dies führt zu einer enormen Reduzierung des Rechenaufwandes. Es werden verschiedene Klassen von Filteroperatoren vorgestellt, die für den Einsatz bei der Suche nach Ringkandidaten im CERES-Experiment von Interesse sind. Basierend auf der Strategie der Durchführung von Faltungen im Fourierraum wird ein neues, schnelles Konzept zur Identifizierung von Ringkandidaten entwickelt.

2.1 Faltungen im Fourierraum

Die **diskrete Fourier-Transformation** (**DFT**) (siehe Anhang A) ist das diskrete Analogon der Fourier-Integrale. Aus den in Anhang A zusammengefaßten Eigenschaften folgt, daß die DFT eine Matrix im Ortsraum in eine äquivalente Darstellung im Fourierraum überführt. Die DFT eines diskreten Bildes und das Bild selbst sind also nur zwei verschiedene Darstellungen derselben Information.
Die DFT ist aber nur eine von vielen Möglichkeiten, ein Bild in eine äquivalente Darstellung zu transformieren. Jede **unitäre Transformation** ist in der Lage, dies zu leisten. Weitere für die Bildverarbeitung wichtige unitäre Transformationen sind die **Kosinus-** und die **Sinus-Transformation**, die **Hadamard-Transformation** sowie die **Haar-Transformation**. Eine ausführliche Darstellung der diskreten

unitären Transformationen im Hinblick auf die Bildverarbeitung findet sich z.B. bei *Jaroslavskij* [13].

Für reelle Matrizen, die in dieser Arbeit betrachtet werden, ergibt sich aus der Symmetrieeigenschaft der DFT (Anhang A) eine wichtige Konsequenz:
reelle Matrizen besitzen ein hermitesches Spektrum:

$$\hat{F}^*_{u,v} = \hat{F}_{M-u,N-v} \ ,$$

wobei $F^*_{m,n}$ die zu $F_{m,n}$ konjugiert komplexe Matrix bezeichnet. Das komplexe Spektrum reller Matrizen ist daher durch die Werte in einem Halbraum vollständig bestimmt. Den Rest des Spektrums erhalten wir durch Spiegelung am Symmetriezentrum $(M/2, N/2)$. Für die praktische Berechnung heißt das, daß das reelle Bild durch sein komplexes Spektrum ersetzt werden kann, weil es den gleichen Speicherplatz benötigt.

Durch das Faltungstheorem (Gl. A.9) gewinnt die DFT eine enorme praktische Bedeutung:
Filteroperationen sind nichts anderes als Faltungen eines Bildes mit einer Maske. Eine Filteroperation ist also vollständig beschrieben durch die Koeffizienten ihrer Maske im Ortsraum, mit der das Bild zu falten ist. Die Filteroperation kann auch im Fourierraum durchgeführt werden. Nach dem Faltungstheorem ist dort die DFT des Bildes mit der DFT der Maske zu multiplizieren. Faltung im Ortsraum und Multiplikation im Fourierraum sind nur verschiedene Darstellungen desselben Operators.
Berechnet man die Filterung nicht als Faltung im Ortsraum, sondern als Multiplikation im Fourieraum, so läßt sich eine enorme Reduzierung des Rechenaufwandes erreichen, da jetzt ja nur noch *eine* komplexe Multiplikation pro Bildpunkt durchzuführen ist.
Es sind zwar zusätzlich noch eine DFT und eine inverse DFT durchzuführen, aber es wurden von *Blahut* [14], *Brigham* [15] und anderen **Fast Fourier Transformation (FFT)-Algorithmen** entwickelt, die die für die DFT erforderliche Rechenzeit erheblich reduzieren. Der Rechenaufwand pro **Pixel** (Bildpunkt) bei Verwendung eines **vektorisierten Basis-4-Algorithmus** für die DFT ist in Tab. 2.1 aufgeschlüsselt. Den Berechnungen liegt eine quadratische komplexe Bildmatrix der Größe M ($M \in \mathbb{N}$) zugrunde. Ein Basis-4-Algorithmus beruht auf einer sukzessiven Zerlegung der Ausgangsmatrix in vier Teilmatrizen. (Entsprechend wird bei einem Basis-2-Algorithmus die Ausgangsmatrix sukzessive halbiert.) Diese Zerlegung bedeutet nichts anderes als eine geschickte Berechnung von Zwischenergebnissen, die dann von mehreren weiteren Rechenoperationen genutzt werden. Bei einer sukzessiven Zerlegung einer $M \times M$-Matrix in vier Teilmatrizen gibt es $\log_4 M^2 = \log_2 M$ Stufen. Auf der ersten Stufe fallen nur triviale Multiplikationen an, und es muß nur bei 3/4 der Punkte multipliziert werden. Vektorisiert bedeutet, daß die Struktur des Algorithmus nicht nur für ein Matrixelement, sondern *gleichzeitig* für die ganze Matrix angewendet wird. Für eine detaillierte Beschreibung sei

Operation	komplexe Multiplikationen	komplexe Additionen
Hintransformation	$3/4(\log_2 M - 2)$	$2\log_2 M$
Filterung	1	0
Rücktransformation	$3/4(\log_2 M - 2)$	$2\log_2 M$
Summe komplexe Operationen	$3/2\log_2 M - 2$	$4\log_2 M$
Summe reelle Operationen	$6\log_2 M - 8$	$11\log_2 M - 4$

Tabelle 2.1: *Zusammenstellung des Rechenaufwandes pro Pixel für die Durchführung einer Filteroperation im Fourierraum.* Die DFT wurde unter Verwendung eines vektorisierten Basis-4-Algorithmus berechnet. Die angegebenen Formeln beziehen sich auf eine quadratische komplexe Bildmatrix der Größe M.

auf *Blahut* [14] verwiesen.
Berechnet man die Faltung einer quadratischen komplexen Bildmatrix der Größe M mit einer Maske nach der Definitionsgleichung A.8, so sind pro Pixel M^2 komplexe Multiplikationen und $M^2 - 1$ komplexe Additionen erforderlich.
Der im CERES-Experiment vorliegende Fall von quadratischen, reellen Matrizen der maximalen Größe von 288 Pixeln erfordert also bei der Berechnung der Filteroperation nach der Definitionsgleichung 82944 reelle Multiplikationen und 82943 relle Additionen pro Pixel; bei Verwendung eines vektorisierten Basis-4-Algorithmus für die DFT sind für die Filteroperation jedoch nur 40 reelle Multiplikationen und 84 reelle Additionen erforderlich. Die Reduzierung des Rechenaufwandes ist also enorm.
Als weiterer Vorteil kommt hinzu, daß die Multiplikation kommutativ, assoziativ und zusammen mit der Addition distributiv ist. Das bedeutet, daß mehrere Operatoren, die hintereinander auf das Bild angewendet werden sollen, additiv und multiplikativ zu einem Operator zusammengefaßt werden können. Dieser Gesamtoperator braucht dann nur einmal für alle Bilder, auf die er angewendet werden soll, berechnet zu werden. Assoziativität impliziert aber auch, daß wir eine komplizierte Faltungsmaske in einfachere zerlegen und diese hintereinander auf das Bild anwenden können. Bei einer geschickten Zerlegung kann der Rechenaufwand weiter vermindert werden. Für eine ausführliche Darstellung solcher Methoden sei auf *Jähne* [10] verwiesen.

2.2 Filteroperatoren

In den folgenden Abschnitten werden drei Klassen von Filteroperatoren besprochen, die für die Suche nach Ringkandidaten im CERES-Experiment von Interesse sind. Eine ausführliche Darstellung des Designs von Filtern findet sich bei *Jähne* [10] und *Haberäcker* [16].

2.2.1 Glättungsfilter

Die einfachsten Glättungsfilter sind **Rechteckfilter**. Bei ihnen handelt es sich im Ortsraum um quadratische Matrizen, deren Koeffizienten alle gleich 1 sind. Zusätzlich haben sie einen Vorfaktor zur Skalierung, weil bei jeder Glättungsmaske die Summe aller Koeffizienten 1 sein muß. Rechteckfilter sind nur in der Lage, bestimmte Wellenzahlen auszulöschen. Je größer die Rechteckfilter, desto besser kann man ihre DFT durch eine Funktion der Gestalt $\sin(k)/k,\ k \in \mathbb{R}$ approximieren.

Bessere Glättungsfilter sind die **Gauß-** oder **Binominalfilter**. Die zugrundeliegende Idee ist, die Gauß-Funktion im Diskreten durch die Binominal-Verteilung zu approximieren. Letztere läßt sich durch das Pascalsche Dreieck darstellen. Faßt man die Koeffizienten einer Zeile des Pascalschen Dreiecks als Komponenten eines Vektors auf, so erhält man eine eindimensionale Binominalmaske. Faltet man diese mit sich selbst (einmal als Zeilen-, einmal als Spaltenvektor aufgefaßt), so ergibt sich die zugehörige zweidimensionale Binominalmaske. Entsprechend der Ordnung n des Binoms, aus der eine zweidimensionale Binominalmaske entstanden ist, bezeichnet man die Maske mit $B^n,\ n \in \mathbb{N}$. Binominalfilter unterdrücken Rauschen oberhalb einer bestimmten Wellenzahl effektiv. Dies ist klar, da die DFT eines Binominalfilters je nach seiner Größe eine mehr oder wenige gute Approximation einer Gaußschen Glockenkurve darstellt.

2.2.2 Laplacefilter

Laplacefilter oder **Laplaceoperatoren** stellen diskrete Approximationen der zweiten Ableitung einer Funktion dar. Ein einfaches Beispiel eines Laplaceoperators ist

$$L_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} . \tag{2.1}$$

Es läßt sich zeigen, daß jeder Operator $I - B$ ein Laplaceoperator ist, wobei I der Identitätsoperator und B ein Binominalfilter ist. Mit einer 3×3-Binominalmaske ergibt sich z.B. folgender Laplacefilter:

$$L_2 = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{pmatrix} . \tag{2.2}$$

Der Laplaceoperator entfernt nicht nur konstante Amplitudenbereiche des Bildes, sondern als Ableitungsoperator zweiter Ordnung auch jede konstante Amplitudensteigung.

2.2.3 Sobeloperatoren

Die **generalisierten Sobeloperatoren** stellen einen Satz von vier Gradientenoperatoren in Achsen- und Diagonalenrichtung dar:

$$S_1 = \frac{1}{8}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad S_2 = \frac{1}{8}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix},$$
$$S_3 = \frac{1}{8}\begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{pmatrix}, \quad S_4 = \frac{1}{8}\begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}. \tag{2.3}$$

Die Sobeloperatoren verbessern die Kantendetektion im Vergleich zu Ableitungsoperatoren und glätten gleichzeitig senkrecht zur Kantendetektionsrichtung.

2.3 Das neue Konzept zur Suche nach Ringkandidaten

Aus der in Abschnitt 2.1 erläuterten Strategie der Durchführung von Filteroperationen im Fourierraum mit Hilfe der DFT wurde ein neues, schnelles Konzept zur Suche nach Elektronringkandidaten entwickelt:
hierzu sind die Bilder der UV-Detektoren mit einer Ringmaske zu falten, deren Radius gleich dem eines asymptotischen Čerenkov-Ringes ist. Diese Ringmaske kann auf verschiedene Arten variiert und/oder mit beliebigen Filteroperationen zu einer neuen Filteroperation verkettet werden.
Die zu transformierenden und zu faltenden Matrizen sind von der Größe 288×288 (286×286) Pads. Aus Konsistenzgründen übernehmen wir die Konvention aus Anhang A, die Zeilen- und Spaltenindizes dieser Matrizen von 0 bis 287 (285) laufen zu lassen.

Im folgenden wird die Funktionsfähigkeit dieses Konzeptes am Beispiel der einfachsten Ringmaske mit asymptotischem Radius gezeigt; diese Ringmaske hat eine Breite von 1 Pad und eine Amplitude von 1. Dabei ist zu beachten, daß das zentrale Pixel der Maske auf das Matrixelement (0,0) gesetzt werden muß, während alle Pixel, die negative Indizes haben, gemäß der Periodizität der DFT (Gl. A.4) um die Länge der Matrix verschoben werden müssen. Als DFT dieser Ringmaske

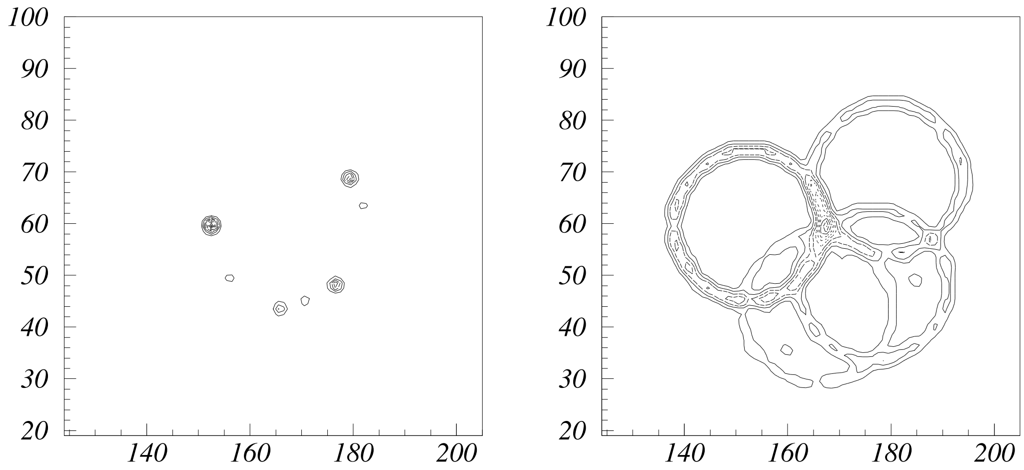

Abbildung 2.1: *Rechts: der asymptotische Ring, der mit der einfachsten Ringmaske mit asymptotischem Radius im Fourierraum gefaltet wird. Links: der Realteil der vom Fourierraum in den Ortsraum zurücktransformierten Faltung. Es ist deutlich ein Maximum an der Stelle des ursprünglichen Ringmittelpunktes zu erkennen.* Beide Darstellungen sind Contour-Plots.

erhält man das diskrete Analogon des Beugungsbildes einer Ringblende.
Wir wählen nun einen beliebigen asymptotischen Ring (Abb. 2.1). Seine DFT läßt sich in guter Näherung durch eine Funktion der Gestalt $\sin(k)/k,\ k \in \mathbb{R}$ approximieren, wobei Real- und Imaginärteil eine ähnliche Struktur aufweisen. Faltet man den Ring mit der oben beschriebenen Ringmaske, so ergibt sich im Fourierraum für den Realteil wieder ein $\sin(k)/k$-artiger Verlauf, wohingegen der Imaginärteil nahezu konstant ist. Der Realteil des in den Ortsraum zurücktransformierten Bildes ist in Abb. 2.1 zu sehen. Es ist deutlich zu erkennen, daß sich ein Maximum an der Stelle ausgebildet hat, an der sich der ursprüngliche Ringmittelpunkt befand. Es ist klar, daß dieses Maximum das globale Maximum ist. Im Imaginärteil findet man an der Stelle des ursprünglichen Ringmittelpunktes zwar auch ein Maximum; das Faltungsbild wiederholt sich jedoch periodisch, wobei die Periode gerade der Gitterkonstanten des reziproken Gitters $(2\pi/M)$ entspricht. Hier ist eine eindeu-

tige Bestimmung des Ringzentrums also nicht möglich.
Somit ist gezeigt, daß mit dem entwickelten Konzept der Durchführung von Faltungsoperationen mit entsprechenden Ringmasken im Fourierraum eine Identifizierung von Ringkandidaten möglich ist.

Kapitel 3

Anwendung der Filteroperatoren auf reale Daten

In diesem Kapitel wird die Anwendung des im vorangehenden Kapitel entwickelten neuen Konzeptes zur Idenfizierung von Ringkandidaten auf realen Daten untersucht und mit dem bisherigen Verfahren verglichen.

3.1 Anwendung des neuen Konzeptes

Das im vorangehenden Abschnitt beschriebene neue Konzept der Durchführung von Filteroperationen im Fourierraum wurde anstelle der Hough-Transformation[1] in den bisherigen Algorithmus zur Suche nach Elektronringkandidaten implementiert und die Parametereinstellungen der neuen Transformation entsprechend angepaßt. Das so entstandene neue Verfahren zur Rekonstruktion der Elektronringe wurde in Bezug auf die Ringfindeeffizienz, das Signal-zu-Untergrund-Verhältnis sowie die Rekonstruktionsgenauigkeit - die Begriffe werden unten definiert - untersucht und mit dem alten Verfahren (also mit der Hough-Transformation) verglichen.
Im einzelnen wurden folgende Klassen von Filteroperatoren untersucht:

1. *Ringmasken*:
   einfache Ringmasken unterschiedlicher Breite und Amplitude.
   Doppelringmasken unterschiedlicher Breite und Amplitude. Hierbei ist unter einer Doppelringmaske eine einfache Ringmaske zu verstehen, an deren inneren und äußeren Rand einfachen Ringmasken der gleichen negativen

[1]Zur Erklärung der Hough-Transformation siehe Abschnitt 6.7.

Amplitude angehängt sind. Man kann eine solche Maske also als Überlagerung einer breiteren und einer schmaleren Ringmaske aufassen (daher der Begriff Doppelringmaske). Die Breite der schmaleren Ringmaske bezeichnet man als **innere Breite**, die Breite der breiteren Ringmaske als **äußere Breite** der Doppelringmaske. Die Begriffe **innere** und **äußere Amplitude** sind analog definiert. Zur Verdeutlichung des Begriffes Doppelringmaske ist in Abb. 3.1 der Querschnitt durch den Ring einer Doppelringmaske der inneren und äußeren Breite 3 dargestellt.

2. *Verkettung von Rechteckfiltern und Ringmasken*;
   Verkettung von Binominalfiltern und Ringmasken;
   Verkettung von Laplace-Operatoren und Ringmasken;
   Verkettung von Sobeloperatoren und Ringmasken.
   Es wurden Rechteck- und Binominalfilter verschiedener Größe, verschiedene Laplace-Operatoren sowie Sobeloperatoren in Verbindung mit Ringmasken aus 1. untersucht.

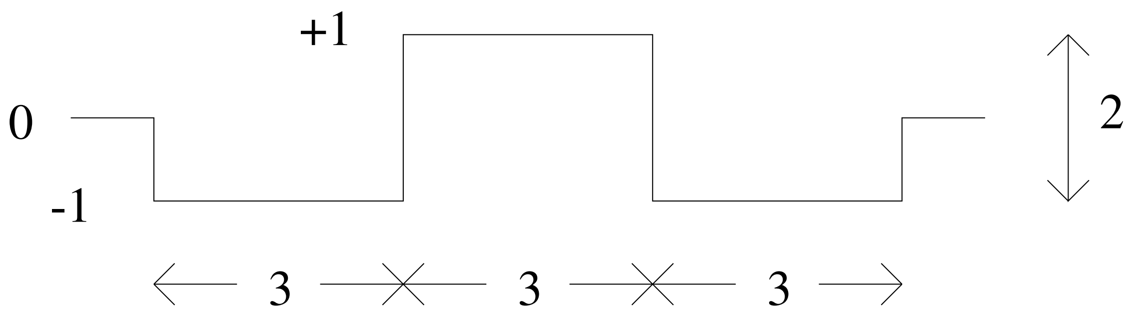

Abbildung 3.1: *Querschnitt durch den Ring einer Doppelringmaske.* Die Doppelringmaske hat die innere und äußere Breite 3 sowie die innere Amplitude 1 und die äußere Amplitude -1.

Die Vorgehensweise war wie folgt:
5000 mit Monte-Carlo-Techniken erzeugte einzelne und überlappende Elektronringe wurden realen Daten überlagert und das neue Verfahren mit dem jeweiligen Filteroperator zur Identifikation der Ringkandidaten angewendet. Die Abstände der von dem jeweiligen Verfahren rekonstruierten Ringmittelpunkte zu den generierten Ringzentren wurden in ein Histogramm gefüllt. Als Verteilung dieser Abstände ergibt sich - wie man dies auch erwartet - eine Gaußverteilung, die einem Polynom ersten Grades überlagert ist. Die Gaußverteilung habe den Mittelwert μ und die Standardabweichung σ. σ stellt ein Maß für die Genauigkeit der Rekonstruktion der Ringmittelpunkte dar und wird daher im folgenden als **Rekonstruktionsgenauigkeit** bezeichnet. Die Fläche B des Trapezes unter der

Gaußverteilung über dem Intervall $[\mu - 3\sigma, \mu + 3\sigma]$ ist ein Maß für die Anzahl der Ringe pro Ereignis, die das jeweilige Verfahren fälschlicherweise als MC-Ringe wiederfindet (**Untergrund**). Die Fläche S der Gaußverteilung über dem Trapez ist ein Maß für die Anzahl der wiedergefundenen MC-Ringe pro Ereignis (**Signal**). Also ist der Quotient aus S und der Gesamtzahl der erzeugten MC-Ringe ein Maß für die Effizienz, einen Ring zu finden, und wird im weiteren Verlauf dieses Kapitels als **Ringfindeffizienz** bezeichnet. Der Quotient aus S und B stellt das bei der Untersuchung bestimmte **Signal-zu-Untergrund-Verhältnis** dar.

Die erhaltenen Ergebnisse sind in Tabelle 3.1 zusammengefaßt. Hierbei ist für jede untersuchte Klasse von Filteroperatoren derjenige Operator aufgeführt, für den sich bezüglich der zu messenden Größen die besten Werte ergaben. Zum Vergleich wurde auch die bisherige Kandidatensuche mittels der Hough-Transformation untersucht. Die gemessenen Werte sind mit ihren statistischen Fehlern angegeben. Unter der einfachsten Doppelringmaske der Breite 3 Pads ist die in Abb. 3.1 dargestellte Doppelringmaske mit einer inneren und äußeren Breite von 3 Pads sowie einer inneren bzw. äußeren Amplitude von 1 bzw. -1 zu verstehen.
Aus Tab. 3.1 geht hervor, daß die Faltung mit der einfachsten Doppelringmaske der Breite 3 Pads sowie deren Verkettung mit der 3×3-Rechteckmaske bezüglich der Ringfindeeffizienz und dem Signal-zu-Untergrund-Verhältnis gleich gut wie das bisherige Verfahren sind; bezüglich der Rekonstruktionsgenauigkeit sind sie um ca. 10% besser.

Bei der Ringkandidatensuche verwendeter Filteroperator	Rekonstruktionsgenauigkeit in Pads	Ringfindeeffizienz in Prozent	Signal-zu-Untergrund-Verhältnis
Einfache Ringmasken: Einfachste Ringmaske	0.336 ± 0.011	62.7 ± 1.6	51.6 ± 3.6
Doppelringmasken: Einfachste Doppelringmaske der Breite 3 Pads	0.333 ± 0.014	76.8 ± 1.4	63.8 ± 11.9
Verkettung von Rechteckfiltern und Ringmasken: 3×3-Rechteckmaske und obige Doppelringmaske	0.320 ± 0.013	75.3 ± 1.4	63.3 ± 11.7
Verkettung von Binominalfiltern und Ringmasken: B^2-Binominalmaske und obige Doppelringmaske	0.328 ± 0.012	74.8 ± 1.4	55.1 ± 4.7
Verkettung von Laplace-Operatoren und Ringmasken: Laplace-Operator $4(B^2 - I)$ und obige Doppelringmaske	0.827 ± 0.038	67.0 ± 1.5	33.1 ± 1.4
Verkettung von Sobel-Operatoren und Ringmasken: Generalisierte Sobel-Operatoren und einfachste Ringmaske	0.935 ± 0.048	60.1 ± 1.6	29.2 ± 2.9
Bisheriges Verfahren zur Kandidatensuche (Hough-Transformation)	0.373 ± 0.010	75.8 ± 1.0	64.3 ± 8.1

Tabelle 3.1: *Ergebnisse der Untersuchung des neuen Konzeptes zur Ringkandidatensuche mittels zurücktransformierter Filteroperationen im Fourierraum.* Für Erläuterungen sei auf den Text verwiesen.

Kapitel 4

Theorie der Neuronalen Netzwerke

Mit „Neuronale Netzwerke" hat ein Begriff in die Lehre der Informationsverarbeitung Einzug gehalten, der ursprünglich aus dem medizinisch-biologischen bzw. neurophysiologischen Bereich stammt. In diesem Kapitel wird die mathematisch-theoretische Grundlage der Neuronalen Netzwerke vorgestellt.

4.1 Von der Biologie zur Mathematik

Künstliche Neuronale Netzwerke stellen eine Abstraktion des biologischen Vorbildes dar. Um das mathematische Modell besser zu verstehen, das künstlichen Neuronalen Netzwerken zugrundeliegt, ist es sinnvoll, die Grundeinheit biologischer Netzwerke näher zu betrachten.

Dieser elementare Baustein ist die **Nervenzelle**, auch **Neuron** genannt. Es ist möglich, die Nervenzelle in weitere Funktionseinheiten zu gliedern, wie aus Abb. 4.1 hervorgeht. Das Neuron bildet das eigentliche informationsverarbeitende Schaltelement, indem es durch Gewichtung und Summierung von Eingangssignalen einen Ausgangsimpuls liefert, falls die Summe einen bestimmten Schwellenwert überschritten hat.
Das Neuron empfängt seine Signale über **Dendriten**, die über **Synapsen** mit den Ausgängen (**Axonen**) der benachbarten Zellen verbunden sind (Die Begriffe sind in Abb. 4.1 zeichnerisch erklärt). Die Gewichtung der Signale geschieht an den Synapsen durch sogenannte **Neurotransmitter**, welche auf chemischem Wege erregend (**exzitatorisch**) oder hemmend (**inhibitorisch**) wirken können. Die im Zellkörper (**Soma**) von den Dendriten aus ankommenden elektrischen Signale bewirken eine Depolarisation der Zellmembran. Ab einer bestimmten Schwelle führt

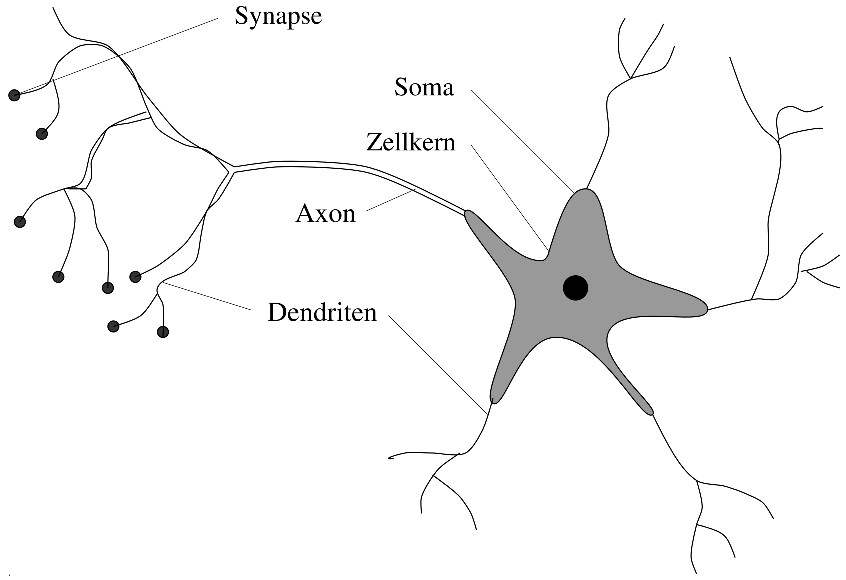

Abbildung 4.1: *Schematische Darstellung einer Nervenzelle.* Das Neuron erhält an den Synapsen Signale, die von Dendriten an den Zellkörper (Soma) weitergeleitet und dort akkumuliert werden. Bei Übersteigen eines bestimmten Schwellenwertes wird das Signal von der Zelle mit Hilfe des Axons an andere Zellen weitergeleitet.

die Depolarisation zu einem elektrischen Impuls auf dem Axon, das diesen zu benachbarten Neuronen weiterleitet. Die für die Übertragung von einer Zelle zur nächsten benötigte Zeit ist kleiner als 1/1000 Sekunde. Die Großhirnrinde des Menschen ist aus etwa 10^{11} bis 10^{12} Neuronen mit durchschnittlich jeweils 10^3 Dendriten aufgebaut. Das gesamte Netzwerk enthält also etwa 10^{15} synaptische Verbindungen. Für weiterführende Werke auf dem Gebiet der biologischen Neuronalen Netzwerke sei auf *Schöneburg, Hansen* und *Gawelczyk* [17] sowie *Rumelhart* und *McClelland* [18] verwiesen.

Diese Erkenntnisse aus der Biologie erlauben es nun, ein mathematisches Modell für ein Neuron zu entwickeln. Was Größe (Anzahl der Neuronen), Aufbau (Art der Vernetzung) und differenzierte Funktionsweise der Neuronen anbelangt, stellen die auf diese Weise entwickelten künstlichen Modelle natürlich nur eine sehr grobe Annäherungen an das in der Natur angewandte Konzept der Informationsverarbeitung dar. Aber eine realistische Beschreibung der biologisch-chemischen Vorgänge im Gehirn ist auch nicht beabsichtigt; vielmehr wird versucht, ein System zu konstruieren, das Eigenschaften hervorbringt, die auch das menschliche Gehirn auszeichnen, wie z.B. Parallelität, Fähigkeit zur Mustererkennung, Robustheit und Lernfähigkeit.

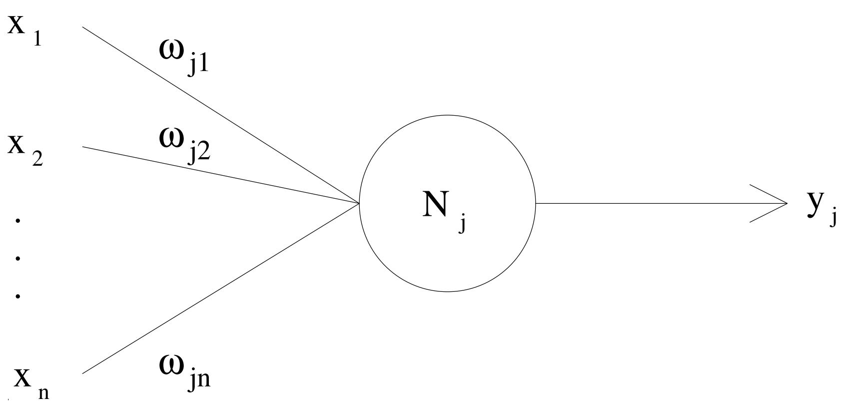

Abbildung 4.2: *Darstellung des mathematischen Modell-Neurons.* Das Modell-Neuron mit der Nummer j summiert die Eingangswerte x_k mit den entsprechenden Gewichten ω_{jk} auf und gibt nach einer Transformation in Form einer im Text näher erläuterten Aktivierungsfunktion den Wert y_j weiter.

In der zeichnerischen Darstellung des **mathematischen Modell-Neurons**, wie sie durch Abb. 4.2 gegeben ist, werden die Dendriten symbolisch als Verbindungslinien zu den **Eingangswerten** x_k dargestellt, denen ein **Gewichtswert** ω_{jk} zugeordnet wird. Dieser Gewichtswert repräsentiert je nach Vorzeichen eine hemmende oder erregende synaptische Verbindung zum Neuron N_j. Die **Aktivität** y_j des Neurons j kann als Funktion der n gewichteten Eingangswerte als

$$y_j = g\left(\sum_{j=0}^{n} \omega_{jk} x_k + \Theta_j\right) \tag{4.1}$$

geschrieben werden. Dabei bezeichnet Θ_j die **Schwelle** des Neurons und $g(x)$ eine beschränkte, i.a. nichtlineare Funktion, die **Aktivierungsfunktion** genannt wird.

Es werden i.a. nur Aktivierungsfunktionen zugelassen, deren Wertebereich im Intervall $[-1, 1]$ bzw. $[0, 1]$ liegt. Eine sehr gebräuchliche Aktivierungsfunktion, die auch in den in dieser Arbeit implementierten Backpropagation-Netzwerken (zur Erklärung diese Begriffes sei auf Abschnitt 4.5 verwiesen) verwendet wird, ist die **Sigmoidfunktion**, die durch

$$f(x) := \frac{1}{1 + \exp(-\frac{x}{T})} \tag{4.2}$$

definiert ist. f ist semilinear, und es gilt $f(x) \in [0, 1]$. Die Wahl des Parameters T, den man in Analogie zur Fermi-Statistik als **Temperatur** bezeichnet, bestimmt die Steilheit der Kurve; für $T \to 0$ ergibt sich die Heavyside-Funktion. In Abb. 4.3 ist die Sigmoidfunktion für verschiedene Werte von T graphisch dargestellt.

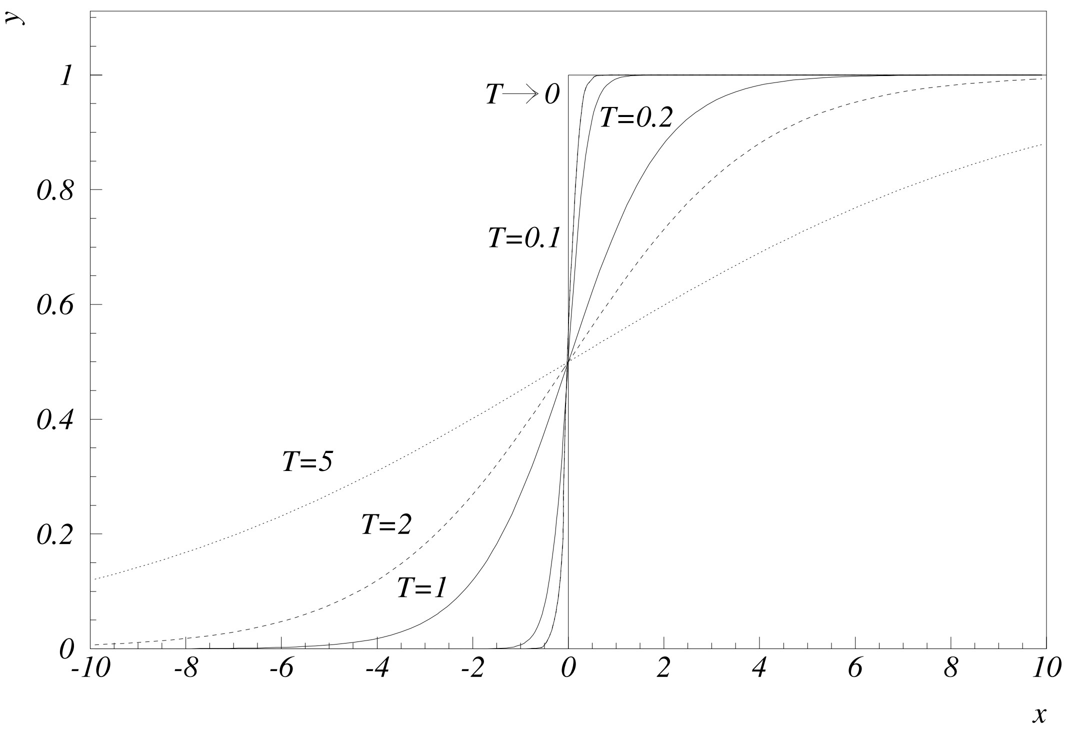

Abbildung 4.3: *Die Sigmoidfunktion.* Je kleiner die Temperatur T ist, desto ausgeprägter ist das Schwellwertverhalten der Sigmoidfunktion.

4.2 Das Multilayer-Feedforward-Netzwerk

Sowohl in der Biologie, als auch bei den künstlichen Neuronalen Netzwerken bestimmt die **Topologie**, d.h. die Anordnung und Verknüpfungsstruktur der Neuronen, entscheidend die Funktion des Gesamtsystems. Statt Topologie ist auch der Begriff **Architektur** gebräuchlich.

Die Topologie des **Multilayer-Feedforward-Netzwerk** - auch **Multilayer-Perceptron (MLP)** genannt - ist wie folgt definiert: die Neuronen des Netzwerkes sind in mehreren Schichten (**Lagen**) in der Weise angeordnet, daß Verknüpfungen nur zwischen Neuronen aus verschiedenen benachbarten Schichten bestehen. Das Signal wird dabei nur von der ersten Schicht (**Eingabeschicht**) in Richtung der letzten Schicht (**Ausgabeschicht**) weitergeleitet. Aus dieser Tatsache - das Signal wird nur in Vorwärtsrichtung durch das Netzwerk geleitet - rührt die Bezeichnung **Feedforward**. Die Schicht(en) zwischen Eingabe- und Ausgabeschicht nennt man **verborgene Schicht(en)**. Die Neuronen bezeichnet man entsprechend ihrer Zugehörigkeit zur Eingabe- oder Ausgabeschicht bzw. zu einer verborgenen Schicht oft auch als **Eingabe-, Ausgabeneuronen** bzw. **verborgene Neuronen**.

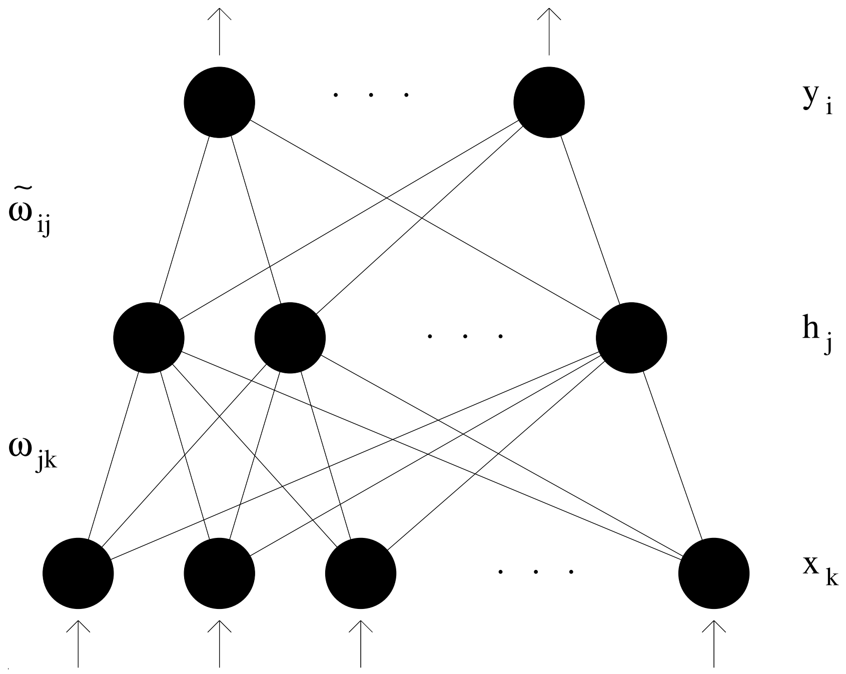

Abbildung 4.4: *Darstellung eines Multilayer-Feedforward-Netzwerkes.* Die Beschreibung eines dreilagigen Netzes mit variabler Anzahl von Neuronen kann durch die Variablen x_k für die Eingangswerte (Eingabeneuronen), h_j für die Aktivität der verborgenen Neuronen und y_i für die Ausgangswerte (Ausgabeneuronen) erfolgen. Die Variablen ω_{jk} und $\tilde{\omega}_{ij}$ beschreiben die Gewichte zwischen Eingabeschicht und verborgener Schicht bzw. verborgener Schicht und Ausgabeschicht.

Die Topologie eines Multilayer-Feedforward-Netzwerkes ist in Abb. 4.4 anhand eines dreilagigen Netzes zeichnerisch erklärt.

Mathematisch gesehen stellt ein Multilayer-Feedforward-Netzwerk mit n Eingabeneuronen und m Ausgabeneuronen eine Abbildung F eines n-dimensionalen Vektors $\vec{x}$ auf einen m-dimensionalen Vektor $\vec{y} := F(\vec{x})$ dar. Die Abbildungsfunktion F hängt bei gegebener Aktivierungsfunktion $g(x)$ nur von den Gewichten ω_{jk} und den Schwellen Θ_j ab. Die Bestimmung der ω_{jk} und der Θ_j ist die Aufgabe eines Optimierungsprozesses, der im Bereich der künstlichen Neuronalen Netzwerke als **Trainieren** bezeichnet wird. Hierauf wird in Abschnitt 4.4 ausführlich eingegangen.

Um eine geschlossene, einfache mathematische Form zur Behandlung der Gewichte zu erhalten, ohne den Schwellen Θ_j eine Sonderrolle zukommen zu lassen, kann man in jeder Schicht ein zusätzliches Neuron N_0 mit einer konstanten Aktivität von 1 einführen, so daß Θ_j durch ω_{j0} beschrieben wird.

Damit ist eine mathematische Beschreibung der Vorgänge in den einzelnen Schichten möglich: wir gehen von einem dreilagigen Feedfordward-Netzwerk aus und übernehmen die Nomenklatur von Abb. 4.4. Die Werte h_j der Neuronen der verborgenen Schicht bzw. die Werte y_i der Neuronen in der Ausgabeschicht berechnen sich zu

$$h_j = g\left(\sum_{k=0}^{n} \omega_{jk} x_k\right) = g\left(\vec{\omega_j} \cdot \vec{x}\right) \quad \text{bzw.} \tag{4.3}$$

$$y_i = g\left(\sum_{j=0}^{l} \tilde{\omega}_{ij} h_j\right) = g\left(\vec{\tilde{\omega}_i} \cdot \vec{h}\right) \ , \tag{4.4}$$

wobei l gleich der Anzahl der Neuronen in der verborgenen Schicht ist. In Matrixschreibweise ergibt sich die Abbildung F somit zu

$$F(\vec{x}) = g\left(\tilde{W} \cdot g(W \cdot \vec{x})\right), \tag{4.5}$$

wenn man die Matrizen W und $\tilde{W}$ durch

$$W := \begin{pmatrix} \omega_{10} & \dots & \omega_{1n} \\ \vdots & \ddots & \vdots \\ \omega_{l0} & \dots & \omega_{ln} \end{pmatrix} \quad \text{bzw.} \quad \tilde{W} := \begin{pmatrix} \tilde{\omega}_{10} & \dots & \tilde{\omega}_{1l} \\ \vdots & \ddots & \vdots \\ \tilde{\omega}_{m0} & \dots & \tilde{\omega}_{ml} \end{pmatrix} \tag{4.6}$$

definiert. Diese Matrizen beschreiben die Kopplungen der Synapsen zwischen Eingabeschicht, verborgener Schicht und Ausgabeschicht.

Aus Gl. 4.5 wird der Einfluß der Aktivierungsfunktion und die Bedeutung ihrer Nicht-Linearität ersichtlich. Ohne die Aktivierungsfunktion wäre die verborgene Neuronenschicht sinnlos, denn die beiden Matrizen könnten durch eine einzige, nämlich ihr Produkt ersetzt werden. Die Nicht-Linearität ermöglicht das Lernen bei Multilayer-Feedforward-Netzwerken (siehe Abschnitt 4.4) und die Separierung von Klassen in mehrdimensionalen Räumen (siehe Abschnitt 5).

4.3 Das Theorem von Kolmogorov

Im Jahre 1957 veröffentlichte der russische Mathematiker *Andrei Kolmogorov* ein verblüffendes Theorem bezüglich der Darstellung beliebiger kontinuierlicher Abbildungen vom n-dimensionalen Würfel $[0,1]^n$ nach $\mathbb{R}$ durch Funktionen von nur einer Variablen [19]. Dieses Theorem schürte das Interesse einer Vielzahl von Mathematikern, was dazu führte, daß in den nächsten zwanzig Jahren mehrere Verbesserungen dieses Theorems entdeckt wurden, zu denen auch das unten angeführte

„Mapping-Netzwerk-Existenztheorem von Kolmogorov" gehört. Der Beweis dieses Theorems stützt sich unmittelbar auf das Ergebnis von *Sprecher* [20], das eine Verbesserung von Kolmogorovs Originalresultat darstellt. Ein Beweis findet sich z.B. bei *Hecht-Nielsen* [21]. Zuvor benötigen wir noch folgende Definition: Ein Neuronales Netzwerk, das zur Approximation einer Abbildung $f : R^n \longrightarrow \mathbb{R}^m$ eingesetzt wird, heißt **Mapping-Netzwerk**.

Theorem 4.3.1 (Das Mapping-Netzwerk-Existenztheorem von Kolmogorov) *Sei f eine beliebige kontinuierliche Abbildung $f : [0,1]^n \longrightarrow R^m$, $f(x) = y$. Dann kann f exakt durch ein dreilagiges Feedforward-Netzwerk implementiert werden, das n Neuronen in der (x-) Eingabeschicht, $(2n+1)$ Neuronen in der verborgenen Schicht und m Neuronen in der (y-) Ausgabeschicht besitzt.*

Das in Theorem 4.3.1 angegebenen Netzwerk, das auch als **Kolmogorov-Netzwerk** bezeichnet wird, ist in Abb. 4.5 dargestellt. **Node** ist hierbei nur ein anderes gebräuchliches Wort für Neuron (aus Darstellungsgründen gewählt).

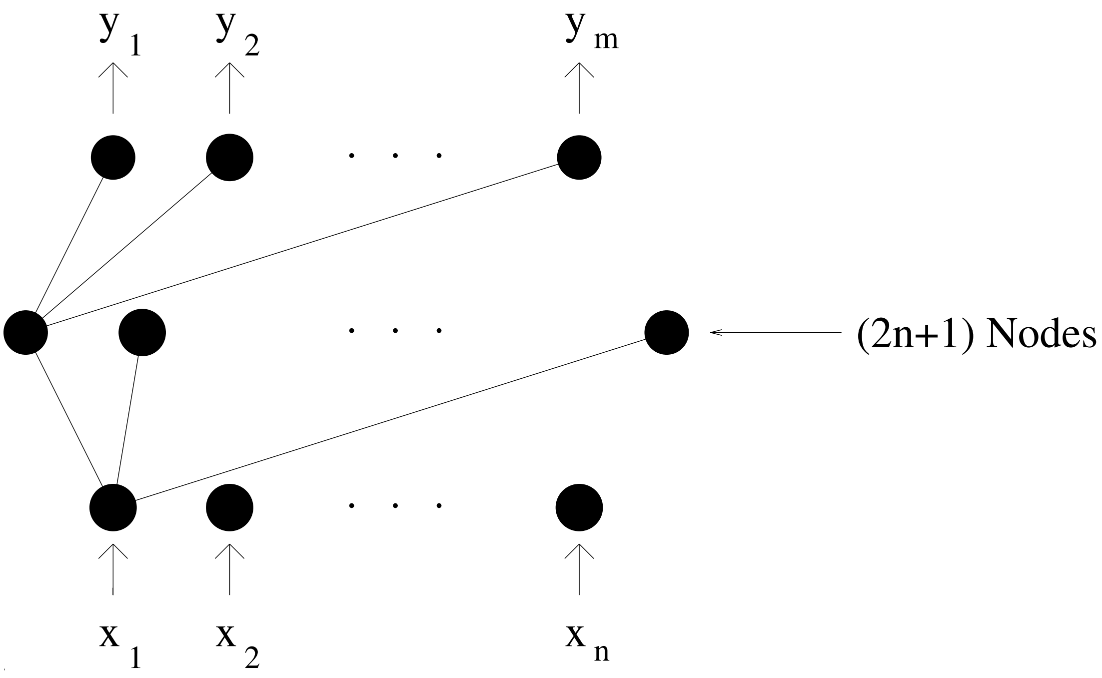

Abbildung 4.5: *Das Kolmogorov-Netzwerk.* Dieses Netzwerk kann jede beliebige kontinuierliche Abbildung $f : [0,1]^n \longrightarrow \mathbb{R}^m$ *exakt* implementieren.

Theorem 4.3.1 stellt die mathematische Existenzberechtigung der Multilayer-Feedforward-Netzwerke dar, da es die Durchführbarkeit der Implementierung einer Abbildung durch ein Multilayer-Feedforward-Netzwerk garantiert. Bis zum Jahre

1986 war kein vergleichbares Existenztheorem bekannt. In den letzten Jahren sind allerdings noch stärker Theoreme gefunden worden, wie z.B. das „Backpropagation-Theorem", das in Abschnitt 4.5 behandelt wird.

Wir können nun analog zu Gl. 4.3 und Gl. 4.4 die Formeln angeben, nach denen sich die Werte der verborgenen Neuronen und der Ausgabeneuronen des Kolmogorov-Netzwerkes berechnen. Aber dies ist auch schon alles, was wir wissen. Es ist kein Beispiel von W, $\tilde{W}$ und g bekannt, daß zu einer exakten Implementierung einer Abbildung führt. Der Beweis des Theorems 4.3.1 ist nicht konstruktiv; somit handelt es sich um ein reines Existenztheorem, daß nicht angibt, wie man W, $\tilde{W}$ und g findet. Aus diesem Grund ist man bei der Implementierung einer Abbildung durch ein Neuronales Netzwerk auf die empirische Vorgehensweise angewiesen.

4.4 Trainieren und Testen eines Neuronalen Netzwerkes

Beim Einsatz von Neuronalen Netzwerken unterscheidet man zwischen drei verschiedenen Abschnitten, der **Lern-** oder **Trainingsphase**, der **Testphase** und der **Anwendungsphase**.

Die Aufgabe während des Trainierens eines Neuronalen Netzwerkes besteht darin, einen Satz von Gewichtsparametern und damit eine Abbildungsfunktion zu finden, die jedem Inputvektor den gewünschten Outputvektor zuweist. Zu diesem Zweck werden dem Neuronalen Netz wiederholt Inputvektoren (**Lernmuster**) präsentiert, zu denen der Outputvektor bekannt ist. Einem bestimmten Lernalgorithmus folgend werden dann die Gewichtsparameter des Netzes verändert. Diesen Vorgang nennt man **Lernen** oder **Trainieren**. Die Gesamtheit aller zum Trainieren verwendeten Input- und jeweils zugeordneten Outputvektoren bezeichnet man als **Lern-** oder **Trainingssample**. Wenn ein Neuronales Netz mit allen Input- und Outputvektoren des Lernsample einmal trainiert worden ist, ist es üblich zu sagen, daß eine **Epoche** vergangen ist.

Das Trainieren eines Netzes erfolgt so lange, bis der sich jeweils ergebende **aktuelle** Outputvektor den gewünschten, **korrekten** Outputvektor optimal approximiert. Zur Bestimmung der Genauigkeit der Approximation benötigen wir zusätzliche, zufällig ausgewählte Inputvektoren und ihre zugehörigen Outputvektoren. Die Menge all dieser Vektoren heißt **Testsample**. Dieses muß *unabhängig* vom Lernsample sein; würde man nämlich dieselben Datensätze zum Trainieren und Testen verwenden, wäre alles, was man bestimmt, wie gut das Netzwerk das Trainingssample gelernt hat. In Wirklichkeit wollen wir aber bestimmen, wie gut das Netzwerk gelernt hat, die Abbildungsfunktion für *beliebige* Inputvektoren zu approximieren. Damit ist auch klar, daß der Einsatz eines Neuronalen Netzwerkes sinnlos ist, wenn die zur Verfügung stehende Datenmenge zu klein ist. Als weitere Bedingung ist an das Testsample zu stellen, daß er *umfassend genug* sein muß, d.h., daß es einen repräsentativen Querschnitt aller möglichen Fälle darstellen muß. Das Gleiche muß natürlich auch für das Lernsample gelten.

Sei $\vec{x}_k$ der k-te Inputvektor und $\vec{y}_k =: f(\vec{x}_k)$ der zugehörige korrekte Outputvektor des Testsamples; sei $\vec{w}$ der Gewichtsvektor und $Y(\vec{x}_k, \vec{w})$ der aktuelle Outputvektor des Netzwerkes. Sei ferner

$$F_k(\vec{x}_k, \vec{w}) := |f(\vec{x}_k - Y(\vec{x}_k, \vec{w})|^2 \tag{4.7}$$

der quadratische Fehler der vom Netz durchgeführten Approximation für den k-ten Inputvektor. Der **mittlere quadatische Fehler** $F(\vec{w})$ des Netzwerkes ist dann definiert durch

$$F(\vec{w}) := \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} F_k(\vec{x}_k, \vec{w}) \ . \tag{4.8}$$

$F(\vec{w})$ ist wohldefiniert für die meisten Netzwerktypen (wie z.B. das Backpropagation-Netzwerk (siehe Abschnitt 4.5), das Hopfield-, das Hamming-, das Counterpropagation- oder das GMDH-Netzwerk (siehe *Lippmann* [22], *Peterson* und *Rögnvaldsson* [23] sowie *Rumelhart* und *McClelland* [18], [24]). Ferner ist $F(\vec{w})$ für die angeführten Netzwerktypen nichtnegativ und differenzierbar. Die Bestimmung des mittleren quadratischen Fehlers $F(\vec{w})$ bezeichnet man als **Testen** des Neuronalen Netzwerkes. Da $F(\vec{w})$ eine Funktion des Gewichtsvektors $\vec{w}$ ist, den das Netzwerk berechnet hat, können wir $F(\vec{w})$ als eine Oberfläche auffasen, die sich „über" dem Gewichtsvektorraum des Netzwerkes befindet, wobei F die Höhe der Oberfläche an der Stelle $\vec{w}$ angibt. Diese Oberfläche wird als die **Fehleroberfläche** des Neuronalen Netzes bezeichnet. Das Ziel ist nun klarerweise, die Gewichte zu finden, die F minimieren. I.a. ist das Minimum von F ungleich Null, da das Netz i.a. nicht in der Lage ist, die gewünschte Abbildung exakt zu implementieren.

Ziel des Trainierens und Testens ist das folgende: man will erreichen, daß ein Inputvektor, mit dem das Netz bisher noch nicht gefüttert worden ist, *interpoliert* wird, d.h., wenn dieser Vektor zwischen oder nahe an Inputvektoren aus dem Lernsample liegt, soll der Output des Netzes in einer vernünftigen Beziehung zu den Outputvektoren stehen, die man für die Inputvektoren aus dem Lernsample erhalten würde. Diese Eigenschaft, noch nie zuvor „gesehene" Inputvektoren im Rahmen der Approximationsgenauigkeit richtig zuzuordnen, bezeichnet man als **Generalisierung**. Diese ist stark vom verwendeten Lernalgorithmus, von der Repräsentativität der benutzten Lern- und Testsamples und der Komplexitivität des jeweiligen Netzes abhängig.

Unter bestimmten Bedingungen läßt sich bei Feedforward-Netzwerken das Phänomen des **Übertrainierens** beobachten. Hier liegt die dem Generalisieren entgegengesetze Eigenschaft vor: das Netz beginnt, das Lernsample „auswendig" zu lernen

(**Überfitten**) und die Generalisierung wird immer schlechter. Das Vorhandensein von Übertrainieren läßt sich dadurch erkennen, daß der mittlere quadratische Fehler bzgl. des Lernsamples mit zunehmender Trainingsdauer immer kleiner wird, während der mittlere quadratische Fehler bzgl. des Testsamples zunächst zwar auch fällt, dann aber ab einem gewissen Punkt wieder ansteigt. Hier kommt es darauf an, das Trainieren zu diesem Zeitpunkt zu beenden.
Ist ein Netzwerk trainiert und getestet, so kann es im **Ausführungsmodus** betrieben werden, was allerdings nur sinnvoll ist, wenn es generalisieren kann.

Ein Neuronales Netz ist nicht immer in der Lage, ein Lernmuster richtig zu klassifizieren, denn es gibt eine theoretische Grenze, das **Bayessche Limit** [25]. Ein Lernmuster x aus dem Überlappungsbereich zweier Klassen gehört nämlich zu beiden Klassen und kann nur mit Hilfe einer statistischen Aussage einer Klasse zugewiesen werden. Die bezüglich einer Fehlklassifizierung optimale Entscheidung für solche Lernmuster wird durch den **Bayesschen Klassifizierer** gegeben. Im Fall von n möglichen Klassen K_i lautet dieser

$$P(K_i|x) = \frac{P(K_i)P(x|K_i)}{\sum_{i=1}^{n} P(K_i)P(x|K_i)} \quad . \tag{4.9}$$

Hierbei bedeutet

- $P(K_i|x)$ die **A posteriori**-Wahrscheinlichkeit der Klassen nach Beobachtung von x, d.h. die Wahrscheinlichkeit, daß der gemessene Vektor x der Klasse K_i angehört hat;
- $P(K_i)$ die **A priori**-Wahrscheinlichkeitsverteilung der Klassen;
- $P(x|K_i)$ die Wahrscheinlichkeitsverteilung der Vektoren x in jeder Klasse K_i.

Der Bayessche Klassifizierer weist also jedes Lernmuster derjenigen Klasse K_i zu, die die höchste A posteriori-Wahrscheinlichkeit besitzt. Das Problem besteht nun darin, die verschiedenen Wahrscheinlichkeitsverteilungen zu berechnen, denn dazu müssen häufig Annahmen über die Verteilung der Lernmuster oder der Klassen gemacht werden.
Lippman und *Richard* [26] haben gezeigt, daß die Abbildung F_{g,W^*} die beste Approximation von $P(K_i|x)$ bezüglich der Quadratnorm ist, d.h. $\|F_{g,W^*}(x) - P(K_i|x)\|$ ist minimal. W^* stellt die Gewichte dar, die ein Neuronales Netz durch Minimierung der Fehlerquadrate bezüglich des korrekten Output-Vektors berechnet, d.h. $\|F_{g,W^*}(x) - t(x)\|, t(x) \in [0,1]$, ist minimal für alle x. Dies bedeutet, daß ein Multilayer-Feedforward-Netzwerk eine Approximation des optimalen Bayesschen Klassifizierers bestimmt, ohne *a priori* Annahmen über Verteilungen zu verwenden.

Ein weiterer wichtiger Punkt ist die Frage nach der Lernbarkeit, d.h. ob Methoden existieren, die es dem Netz ermöglichen, die richtigen Gewichte für die erforderte Approximation aus der empirischen Beobachtung von Lernmustern zu gewinnen. Es konnte von *White* [27] bewiesen werden, daß es Lernalgorithmen gibt, die die Wahrscheinlichkeit dafür, daß der Approximationsfehler des Netzwerkes eine beliebige Größe überschreitet, gegen Null gehen lassen, wenn die Erfahrung des Netzwerkes, d.h. die Größe des Lernsamples, anwächst.
Das Hauptproblem ist in diesem Fall die Kontrolle der Netzwerkkomplexität als Funktion der Erfahrung des Netzes. Beliebig genaue Approximationen an beliebige Funktionen erfordern beliebig komplexe Netzwerke. In der Realität sind aber nur Neuronale Netzwerke endlicher Größe möglich; außerdem lernen zu komplexe Netze zuviel, d.h. sie lernen die Lernmuster auswendig (Überfitten) und liefern dadurch schlechte Ergebnisse bei der Generalisierung. Es ist aber möglich, bei bekannter Größe des Lernsample die Anzahl der verborgenen Neuronen so zu bestimmen, daß die Gefahr von Übertrainieren vermieden wird. *Baum* und *Haussler* [28] haben gezeigt, daß der Fehler, den ein Multilayer-Feedforward-Netzwerk mit einer verborgenen Schicht bei der Generalisierung macht, von der Ordnung $\mathcal{O}(N_\omega/N_p)$ ist, wobei N_ω gleich der Anzahl der Gewichte und N_p gleich der Anzahl der Lernmuster ist. Man kann also folgende *Faustregel* aufstellen: *die Bedingung* $N_\omega \leq 0.1 \cdot N_p$ *sollte erfüllt sein.*

Zum Schluß beantworten wir noch die Frage, welche Funktionen durch Multilayer-Feedforward-Netzwerke mit einer *glatten* Aktivierungsfunktion, wie z.B. die Sigmoidfunktion, implementiert werden können. Es wurde von *Hornik, Stinchcombe* und *White* [29] gezeigt, daß Multilayer-Feedforward-Netzwerke mit nur einer verborgenen Schicht und einer glatten Aktivierungsfunktion in der Lage sind, beliebig genaue Approximationen an meßbare Funktionen und ihre Ableitungen zu erreichen.

4.5 Das Backpropagation-Netzwerk

Ein Multilayer-Feedforward-Netzwerk, das mittels des Backpropagation-Lernalgorithmus trainiert wird, heißt **Backpropagation-Netzwerk**. Der Backpropagation-Lernalgorithmus wird in diesem Abschnitt erklärt.

Das Backpropagation-Netzwerk ist eine der wichtigsten historischen Entwicklungen auf den Gebiet der künstlichen Neuronalen Netzwerke. Es ist ein vor allem auf dem Gebiet der Mustererkennung sehr leistungsfähiges Netzwerk, das erfolgreich in vielen Bereichen der Hochenergiephysik eingesetzt worden ist, so z.B. bei der Trennung von Photonen und Leptonen mittels Kalorimeterinformationen vom ALEPH-Detektor [30] oder bei der Identifizierung des b-Quarks mittels der Kinematik der hadronischen Zerfallsprodukte [31]. Für einen Überblick über Anwendungen von

Neuronalen Netzwerken in der Hochenergiephysik sei auf *Denby* [12] verwiesen.

Im Jahre 1986 stellten *Rumelhart* und *McClelland* den **Backpropagation-Lernalgorithmus** vor, der in Multilayer-Feedforward-Netzwerken zum erstenmal auch Gewichte in verborgenen Schichten verändern konnte [24]. Dies wurde dadurch erreicht, daß die Abweichungen der Outputvektoren von den gewünschten Vektoren - d.h. die Fehler - an die darunterliegenden Schichten weitergegeben wurden (**Backpropagation**). Essentiell war hierbei die Verwendung einer sigmoiden Aktivierungsfunktion.
Rumelhart und *McClelland* bewiesen auch, daß ihr Algorithmus ein Verfahren darstellt, mit dem das nichtlineare Optimierungsproblem zur Auffindung der Gewichte gelöst werden kann. Der Algorithmus entspricht einem Gradientenverfahren zur Lösung nichtlinearer Gleichungssysteme, bei dem die freien Parameter des Systems proportional zum Gradienten der Fehleroberfläche verändert werden.

Zu Beginn des Lernvorganges werden die Gewichte ω_{ij} und ω_{jk} des Netzes mit zufälligen Werten aus einem Intervall [-a,a] initialisiert. Dies ist nötig, da der Backpropagation-Lernalgorithmus nicht in der Lage ist, Symmetrien in den Gewichten zu brechen. Danach werden Paare $(\vec{x}^{(p)}, \vec{t}^{(p)})$ aus dem Lernsample an die Eingabe- bzw. Ausgabeneuronen angelegt und die Gewichte so verändert, daß die **globale Fehlerfunktion**

$$E := \frac{1}{2}\sum_{p=1}^{M}\sum_{i=1}^{m}(y_i^{(p)} - t_i^{(p)})^2 \tag{4.10}$$

minimiert wird. Sie stellt ein Maß für die mittlere quadratische Abweichung vom gewünschten Ergebnis dar. Dabei bezeichnet y_i den Wert des i-ten Ausgabeneurons, t_i den gewünschten Wert und p die Nummer des Musters aus dem Lernsample, der M Paare enthält. Diese von *Rumelhart* und *McClelland* vorgeschlagene Fehlerfunktion ist zwar in der Physik sehr gebräuchlich, aber nach *van Ooyen* und *Nienhuis* [32] im Fall von Neuronalen Netzwerken nicht optimal, wenn man die Geschwindigkeit der Konvergenz beim Lernprozeß betrachtet.
Im folgenden wird das **Online-Updating** dargestellt, bei dem nach jedem Lernmuster eine Veränderung der Gewichte erfolgt, d.h. in Formel 4.10 wird nicht über p summiert. Ein Iterationsschritt des Algorithmus wird als **Lern-** oder **Trainingszyklus** bezeichnet. Er ist dadurch gekennzeichnet, daß zuerst ein Inputvektor an die Eingabeneuronen angelegt und die Aktivitäten h_j und y_i in der verborgenen Schicht bzw. der Ausgabeschicht gemäß den Gleichungen 4.3 und 4.4 berechnet werden. Dann erfolgt die Veränderung der Gewichte in entgegengesetzter Richtung. Zuerst werden die Kopplungen $\tilde{\omega}_{ij}$ zwischen verborgener Schicht und Ausgabeschicht proportional zum Fehlergradienten entsprechend

$$\Delta\tilde{\omega}_{ij} = -\eta\frac{\partial E}{\partial\tilde{\omega}_{ij}} + \alpha\Delta\tilde{\omega}_{ij}^{alt} = -\eta\delta_i h_j + \alpha\Delta\tilde{\omega}_{ij}^{alt} \tag{4.11}$$

verändert, wobei

$$\delta_i := (y_i - t_i) \cdot g'(\tilde{\vec{\omega}}_i \cdot \vec{h}) \ . \tag{4.12}$$

Daraufhin fogen die Gewichte ω_{jk} zwischen verborgener Schicht und Ausgabeschicht, die analog nach

$$\Delta\omega_{jk} = -\eta \frac{\partial E}{\partial \omega_{jk}} + \alpha \Delta\omega_{jk}^{alt} = -\eta \delta_j x_k + \alpha \omega_{jk}^{alt} \tag{4.13}$$

mit

$$\delta_j := \sum_{i=1}^{m} \omega_{ij} \delta_i g'(\vec{\omega}_j \cdot \vec{x}) \tag{4.14}$$

behandelt werden. g' steht hierbei für die erste Ableitung der Aktivierungsfunktion. Die Proportionalitätskonstante η wird als **Lernparameter** oder **Learning Rate** bezeichnet und muß empirisch dem Problem angepaßt werden. Sie definiert gewissermaßen die Schrittweite, die beim Suchprozeß im Parameterraum der Gewichte verwendet wird.
Die Größen $\Delta\omega_{ij}^{alt}$ und $\Delta\omega_{jk}^{alt}$ bezeichnen die im vorangegangenen Iterationsschritt berechneten Änderungen und stellen zusammen mit dem **Momemtumfaktor** oder **Momentum Term** α bereits eine Variation des ursprünglichen Backpropagation-Lernalgorithmus dar. Sie dienen dazu, die Lernrate in Parameterregionen mit „flachem" Gradienten zu vergrößern, oder in Bereichen mit „steilem" Gradienten mögliche Oszillationen zu dämpfen, und die Bewegung des Suchprozesses in die richtige Richtung zu lenken. Der Wert für α muß ebenfalls problemspezifisch ermittelt werden; sinnvolle Werte liegen aber nach *Schöneburg et al.* [17] nur im Bereich $\alpha \in [0, 1]$.
Die rekursive Berechnung des Fehlers δ_j in der verborgenen Schicht aus den δ_i's in der Ausgabeschicht macht das **Prinzip der Fehlerrückverfolgung (Error-Backpropagation)** deutlich. Sie zeigt auch die Flexibilität bei der Erweiterung des Netzes zu mehr als drei Schichten.
Bei der Auswahl der Muster aus dem Lernsample ist darauf zu achten, daß die Lernmuster dem Netz in zufälliger Reihenfolge präsentiert werden, und, falls ein Klassifizierungsproblem vorliegt, die Lernmuster aus verschiedenen Klassen mit der gleichen Wahrscheinlichkeit gelernt werden. Wird dies nicht beachtet, so kann das Ergebnis des Optimierungsprozesses ein Nebenminimum liefern.

Welche Abbildungen können nun durch Backpropagation-Netzwerke implementiert werden ? Das folgende Theorem (Theorem 4.5.1) zeigt, daß das Backpropagation-Netzwerk in der Lage ist, im wesentlichen jede beliebige Funktion von praktischer Bedeutung mit beliebiger Genauigkeit zu implementieren. Ein Beweis dieses Theorems findet sich z.B. bei *Hecht-Nielsen* [33]. Zuvor benötigen wir noch folgende Definition: Eine Funktion $f : [0, 1]^n \longrightarrow \mathbb{R}^m$ heißt **quadrat-integrierbar**, wenn jede Koordinatenfunktion von f auf dem n-dimensionalen Einheitswürfel quadrat-integrierbar ist, d.h. zu $L_2([0, 1]^n)$ gehört.

Theorem 4.5.1 (Das Backpropagation-Approximationstheorem)
Sei $f : [0,1]^n \longrightarrow I\!R^m$ *eine quadrat-integrierbare Funktion und* $\epsilon \in I\!R^+$ *beliebig. Dann existiert ein dreilagiges Backpropagation-Netzwerk, das f bis auf einen mittleren quadratischen Fehler von* ϵ *genau approximiert.*

Obwohl Theorem 4.5.1 feststellt, daß drei Schichten in jedem Fall ausreichen, sind bei der Lösung realer Probleme oft vier, fünf oder sogar mehr Schichten erforderlich. Dies liegt daran, daß eine Approximation mit drei Schichten für viele Probleme eine unpraktisch große Anzahl an verborgenen Neuronen erfordern würde. Deshalb sollte Theorem 4.5.1 die Auswahl einer geeigneten Netzwerkarchitektur für ein reales Problem nicht beeinflussen - es schafft nur die Gewißheit, daß eine geeignete Architektur existieren *muß*. Dieses Theorem ist wie Theorem 4.3.1 ein reines Existenztheorem.

4.6 Geometrische Interpretation des Netzwerk-Outputs

Die mathematische Beschreibung der Prozesse in einem Neuronalen Netzwerk eröffnet auch eine geometrische Interpretation des Klassifizierungsverfahrens. Wir betrachten hierzu die Arbeitsweise eines einzelnen Neurons. Gleichung 4.1 beschreibt zusammen mit der Definitionsgleichung der Sigmoidfunktion (Gl. 4.2) das Verhalten eines Neurons. Setzt man das Argument der Aktivierungsfunktion Null, d.h.

$$\sum_{j=0}^{n} \omega_{jk} \cdot x_k + \Theta_j = 0 \quad \text{oder} \quad \vec{\omega}_j \cdot \vec{x} + \Theta_j = 0 \ , \tag{4.15}$$

so erhält man die Gleichung einer Hyperebene im n-dimensionalen Raum der Input-Variablen x_k. Betrachten wir zunächst den Grenzfall einer Stufenfunktion, so führt das Neuron einen Schnitt durch, bei dem es für Lernmuster (hier als Punkte im Hyperraum zu interpretieren), die auf der einen Seite der Ebene liegen, eine Aktivität von Eins, und für Lernmuster auf der anderen Seite eine Aktivität von Null am Output-Neuron produziert. Die Ebene wird durch einen Satz von Gewichtswerten festgelegt, die die Komponenten eines Vektors darstellen, der zum Normalenvektor der Ebene parallel ist. Wenn die Ebenengleichung normiert wird, beschreibt die Schwelle Θ_j den Abstand der Ebene vom Koordinatenursprung (siehe Abb. 4.6).

Ändert man die Stufenfunktion in eine sigmoide Aktivierungsfunktion, so bildet sich in der Umgebung der Trennebene eine Übergangszone. Für weit von dieser Ebene entfernte Punkte ist der Output noch immer nahe bei Eins bzw. nahe bei Null. Nähert man sich jedoch in senkrechter Richtung der Ebene, dann folgt der Wert des Outputneurons dem Verlauf der Sigmoidfunktion.

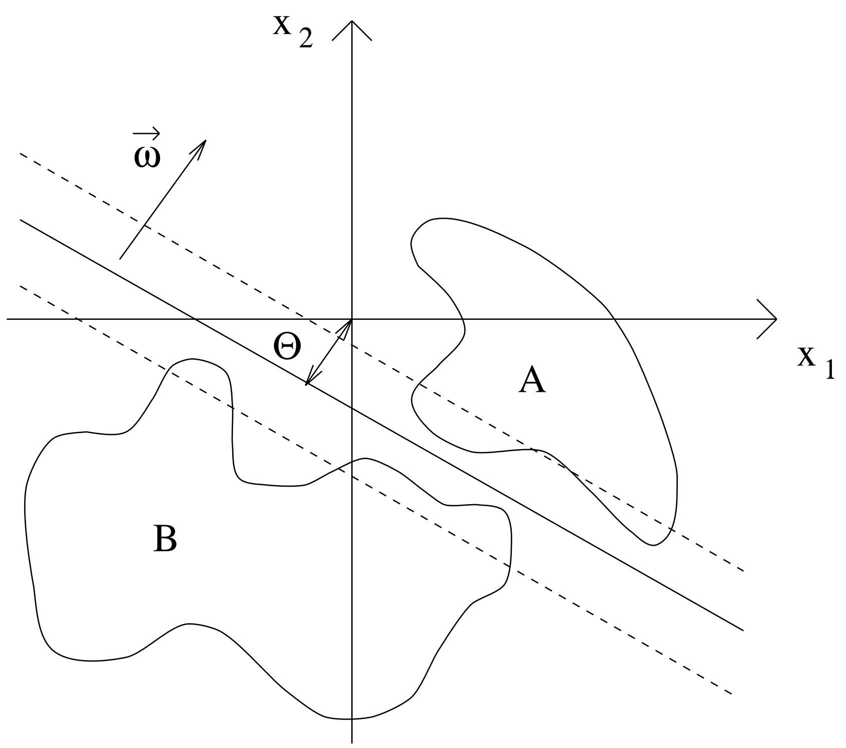

Abbildung 4.6: *Trennung zweier Klassen A und B.* Ein Neuron definiert mit seinem Gewichtsvektor $\vec{\omega}$ und der Schwelle Θ eine Hyperfläche im Raum der Inputvektoren. Die gestrichelten Linien geben den Bereich der Übergangszone im Fall einer sigmoiden Aktivierungsfunktion an.

Durch Normierung der Ebenengleichung kann man eine weitere Information aus dem Gewichtsvektor $\vec{\omega}$ ablesen:

$$\begin{aligned} y &= g(\vec{\omega}\cdot\vec{x}+\Theta) \\ &= g\left(|\vec{\omega}|\cdot\left(\frac{\vec{\omega}}{|\vec{\omega}|}\cdot\vec{x}+\frac{\Theta}{|\vec{\omega}|}\right)\right) \\ &= \frac{1}{1+\exp\left(-\frac{|\vec{\omega}|}{T}\cdot\left(\frac{\vec{\omega}}{|\vec{\omega}|}\cdot\vec{x}+\frac{\Theta}{|\vec{\omega}|}\right)\right)}\ . \end{aligned} \tag{4.16}$$

Aus den Eigenschaften der Sigmoidfunktion, speziell der Abhängigkeit der Steilheit der Kurve vom Parameter T (siehe Abbschnitt 4.1), folgt nun, daß die inverse Länge des Gewichtsvektors als Multiplikator für den Parameter T aufgefaßt werden kann: ist der Gewichtsvektor lang, so beschreibt er eine schmale Übergangszone; ist er kurz, beschreibt er eine breite Zone.

Mit dieser Interpretation eines einzelnen Neurons ist es möglich, den Output eines

Multilayer-Feedforward-Netzwerkes mit einem Ausgabeneuron anschaulich zu verstehen. Die Neuronen in der bzw. den verborgenen Schichten arbeiten als *weiche* Schwellen in einem hochdimensionalen Raum. Das Ausgabeneuron kombiniert die Entscheidungen der tieferligenden Neuronen, ähnlich dem Verfahren einer Diskrimimanzanalyse. Geometrisch gesprochen, formt es Volumina aus den verschiedenen Ebenen und trennt damit die Klassen, die man zu separieren wünscht. Obwohl es in den einzelnen Neuronen sanfte Übergänge gibt, existiert bezüglich der endgültigen Entscheidung keinerlei *Unschärfe*, da am Ende ein definitiver Schnitt beim Wert des Ausgabeneuron angebracht wird. Dieser Schnitt legt die endgültige Güte der Trennung fest. Hierauf sowie generell auf Fragen der praktischen Anwendung Neuronaler Netzwerke wird in Kapitel 5 ausführlich eingegangen.

Kapitel 5

Simulation Neuronaler Netze mit MC-Daten

In diesem Kapitel wird die Entwicklung der Neuronalen Netzwerke zur Lösung der gestellten Aufgaben beschrieben; im einzelnen werden die Realisierung von software-implementierten Multilayer-Feedforward-Netzwerken mit Backpropagation-Lernalgorithmus, die Erzeugung und Optimierung der Lern- und Testsamples, die Bestimmung des optimalen Satzes an Input-Variablen sowie die Optimierung der Netzwerkarchitektur und des Trainingsablaufes behandelt.
Die entwickelten Netzwerke werden dann angewendet, um die Trennbarkeit der Hits von den verschiedenen Untergrundarten sowie die Trennbarkeit von Doppelhits zu bestimmen. Ferner wird die Trennbarkeit der erzeugten Lern- und Testdaten von realen Daten mit Hilfe Neuronaler Netze untersucht.

5.1 Der Lund-Simulator JETNET 2.0

Die in dieser Arbeit entwickelten Neuronalen Netzwerke wurden mit Hilfe des Softwarepaketes JETNET Version 2.0 implementiert, das an der University of Lund, Schweden, von *Lönnblad*, *Peterson* und *Rögnvaldsson* [34] in FORTRAN 77 geschrieben worden ist. JETNET 2.0 ist eine Sammlung von Programmodulen, die die Simulation von Multilayer-Feedforward-Netzwerken sowohl mit Backpropagation- als auch mit Self-Organizing-Map-Lernalgorithmus (zur Erklärung dieses Begriffes sei auf *Hecht-Nielsen* [35] verwiesen) erlaubt. JETNET 2.0 wurde in dieser Arbeit zur Implementierung von Backpropagation-Netzwerken eingesetzt.

Die Parameter von JETNET 2.0, die sich für die Netzwerksimulationen als relevant herausgestellt haben, und - soweit angebbar - die gewählten Werte sind in Tab. 5.1 zusammengestellt. Alle nicht in dieser Tabelle angeführten Parameter wurden auf den in der Programmdokumentation [34] angegebenen Default-Werten belassen.

5.2 Die Erzeugung der Lern- und Testsamples

Eine der Hauptaufgabenstellungen dieser Arbeit ist die Klassifizierung von einzelnen Hits, überlappenden Hits und Untergrund. Bei der Klassifizierung von überlappenden Hits kann man sich auf Doppelhits beschränken, da nach Anhang D die Wahrscheinlichkeit für das Auftreten von Dreifachhits auf einem Cherenkov-Ring unter einem Prozent liegt, und die Wahrscheinlichkeit für das Auftreten von n-fach überlappenden Hits mit wachsendem n abnimmt. In diesem Abschnitt wird die Erzeugung der zur Lösung dieser Klassifizierungsprobleme erforderlichen Lern- und Testsamples behandelt.

Zum Trainieren und Testen der zu entwickelnden Neuronalen Netzwerke wurden keine realen Daten, sondern Monte-Carlo-simulierte Lernmuster verwendet, da nach Anhang B Hits durch geeignete Monte-Carlo-Hits sehr gut approximiert werden. Elektronisches Rauschen und einzelne defekte Pads sind mit Monte-Carlo-Techniken ebenfalls sehr gut simulierbar. Die Monte-Carlo-Daten können somit zum besseren Verständnis der zu entwickelnden Netzwerke benutzt werden, da die einzelnen Effekte (Rauschen, Defekte) beliebig eingeschaltet bzw. verstärkt werden können.
Wie jeder Algorithmus, der auf Monte-Carlo-Daten basiert, so werden auch die hier entwickelten Netzwerke eine Abhängigkeit vom verwendeten Modell aufweisen. Dieser Punkt, der das vielleicht größte Problem bei der Anwendung Neuronaler Netzwerke auf physikalische Prozesse darstellt, ist bis jetzt noch weitgehend unverstanden. *Babbage et al.* [30] haben vorgeschlagen, den Unterschied zwischen den Effizienzen, die man erhält, wenn man Neuronale Netze sowohl auf Monte-Carlo-Daten als auch auf reale Daten anwendet, als den systematischen Fehler zu betrachten, den das Netzwerk bei der Klassifizierung macht. Hierzu ist zu sagen, daß eine vernünftige Abschätzung der systematischen Unsicherheit von Neuronalen Netzen voraussetzt, mindestens *eine* bestimmte Gewichtsmatrix zur *exakten* Implementierung einer Funktion durch ein Neuronales Netz zu kennen, was bis jetzt jedoch noch nicht der Fall ist.
Um die Modellabhängigkeit zu reduzieren, scheint es naheliegend, im Lernsample für eine der beiden zu trennenden Klassen reale Daten zu verwenden. Sinnvollerweise wird man dann für diejenige Klasse reale Daten verwenden, die man besser versteht. Doch dies ist gefährlich, da das Netzwerk - wenn das Lernsample aus Monte-Carlo- und realen Daten gemischt ist - möglicherweise einen Weg findet, Monte-Carlo-Daten und reale Daten zu trennen, der nichts mit der gewünschten Trennung zwischen Signal und Untergrund zu tun hat. Es ist also sicherer, Monte-Carlo-Daten sowohl für die Signal- als auch für die Untergrundklasse zu verwenden.

JETNET-Parameter	Einstellung
Anzahl der Schichten [c]	3 oder 4
Anzahl der Eingabeneuronen (Input-Variablen) [a]	25
Anzahl der Ausgabeneuronen [c]	1 für Klassifizierungsprobleme
Anzahl der verborgenen Neuronen [c]	variabel
Aktivierungsfunktion g [b]	$g(x) = 1/(1 + \exp -\frac{x}{T})$
Temperatur T [c]	zu Beginn des Trainings $T = 1$, nach Erreichen eines lokalen Minimums $T = 3$
Fehlerfunktion	Globale Fehlerfunktion [d] (mittlerer quadratischer Fehler)
Lernrate η [c]	zu Beginn des Trainings und nach Erreichen eines lokalen Minimums $\eta = 0.005$, ansonsten wird η während des Trainings sukzessive verkleinert ($\eta \rightarrow 0$)
Momentumfaktor α [c]	$\alpha = 0$
Initialisierungsintervall $\Delta\omega$ der Gewichte	$\Delta\omega = -0.1 ... + 0.1$
Updating Procedure	Online-Updating [d]
Größe des Lernsamples [e]	ca. $8 \cdot 10^4$ Lernmuster
Größe des Testsamples [e]	ca. $2.5 \cdot 10^4$ Lernmuster
Anzahl der Training Cycles [c]	ca. $25 - 30 \cdot 10^6$
Anzahl der Training Cycles, nach denen getestet wird [c]	ca. 10^5

Tabelle 5.1: *Zusammenstellung der wichtigsten Parameter von JETNET 2.0 für die in dieser Arbeit durchgeführten Simulationen.*

[a] Zur Wahl der Input-Variablen siehe Abschnitt 5.5.
[b] Das ist die in Abschnitt 4.1 definierte Sigmoidfunktion.
[c] Die durchgeführten Trainingsabläufe sind in Abschnitt 5.3 ausführlich erläutert.
[d] Zur Definition siehe Abschnitt 4.5.
[e] Die Erzeugung des Lern- und Testsamples wird in Abschnitt 5.2 erläutert.

Nach Anhang B stellt die 5×5-Matrix eines Hits die optimale Matrix zu seiner Beschreibung dar. Deshalb werden als Lernmuster für einzelne Hits 5×5-Matrizen von Hits verwendet; die Lernmuster für Doppelhits und Untergrund werden so erzeugt, daß sie nichts anderes als 5×5-Matrizen sind, deren Elementen in physikalisch sinnvoller Weise Werte zugewiesen werden.
Ein weiterer wesentlicher Grund für die Wahl der 5×5-Matrizen als Lernmuster ist die Kompatibilität mit dem Hitrekonstruktionsalgorithmus des bisherigen Analyseverfahrens:
dabei wird ausgehend von allen Pads, die gefeuert haben, der Schwerpunkt der Pads innerhalb der 3×3-Matrix um dieses Pad bestimmt. Liegt dieser nicht im zentralen Pad der Matrix, wird diese so lange um ein Pad in x- oder y-Richtung verschoben, bis die Bedingung erfüllt ist; die maximale Anzahl der Iterationsschritte liegt jedoch bei drei. Der so erhaltene Schwerpunkt wird als Schwerpunkt eines Hitkandidaten aufgefaßt. Eine Untersuchung ergab, daß bei 99% der mit diesem Algorithmus durchgeführten Hitrekonstruktionen die Anzahl der Iterationsschritte kleiner als drei war. Da die 5×5-Matrix eines Hits die Eigenschaft besitzt, daß der Schwerpunkt des Hits im zentralen Pad liegt, können also nach dem oben beschriebenen Konzept (Verwendung der 5×5-Matrix eines Hits als Lernmuster) entwickelte Neuronalen Netze auf die vom alten Hitrekonstruktionsalgorihmus gefundenen Hits angewendet werden. Daher werden die 5×5-Matrizen für Doppelhits und Untergrundelemente so erzeugt, daß ihr Schwerpunkt ebenfalls im zentralen Pad liegt, um Anwendbarkeit der zu entwickelnden Netze zusätzlich zum alten Hitrekonstruktionsalgorihmus bezüglich aller drei Klassen zu erhalten.

5.2.1 Die Erzeugung der Monte-Carlo-Einzelhits

Die 5×5-Matrix eines Hits, dessen Ladungsverteilung zweidimensional Gaußverteilt mit einem Sigma von 0.81 Pads in RICH 1 bzw. 0.63 Pads in RICH 2 ist, nennen wir einen **idealen MC-Einzelhit**. Um möglichst **realistische MC-Einzelhits** zu erhalten, müssen wir noch folgende Modifikationen an den idealen MC-Einzelhits berücksichtigen: Hits können mit Gaußverteiltem Rauschen überlagert sein. Einzelne Pads eines Hits können defekt sein. Aufgrund der Nullpunktsverschiebungen ganzer Module kann auch der Nullpunkt von Hits verschoben sein. Es gibt eine untere und eine obere Schranke für die Padamplitude, und damit auch für die Hitamplitude. Zur Erzeugung möglichst realistischer MC-Einzelhits wurden folgende Merkmalsparameter berücksichtigt:

1. *Schwerpunkt*: die Schwerpunktskoordinaten der Hits werden im zentralen Pad der 5×5-Matrix uniform verteilt erzeugt.

2. *Hitparameter*: die Hits werden mit einem Sigma von 0.81 in RICH 1 bzw. 0.63 in RICH 2 und mit einer mittleren Amplitude von 141 in RICH 1 bzw. 245 in RICH 2 erzeugt. Das entspricht den gemessenen Werten.

3. *Verteilung der Hitamplituden*: es werden jeweils gleichgroße Unterklassen mit exponentieller und uniformer Verteilung erzeugt. Würde man nur eine Unterklasse mit exponentieller Verteilung erzeugen, wären die kleinen Hitamplituden überrepräsentiert.

4. *Schwellen*: als minimale Padamplitude wird 3, als maximale 235 gewählt, um den realistischen Sättigungswert des Padreadouts zu berücksichtigen. Damit ergibt sich für die maximale Hitamplitude ein Wert von 4122 in RICH 1 bzw. 3309 in RICH 2. Dabei wird festgelegt, daß ein Hit maximale Amplitude hat, wenn das zentrale Pad und seine 4-er Nachbarschaft maximale Amplitude besitzen.

5. *Defekte*: es werden 10 Unterklassen mit jeweils 0, 1,..., 9 Defekten generiert, wobei die Unterklasse mit 0 Defekten genauso viele Lernmuster enthält wie die Unterklassen mit 1, 2 und 3 bzw. 4, 5 und 6 bzw. 7, 8 und 9 Defekten zusammen. Die Positionen der Defekte in der 5×5-Matrix werden uniform verteilt erzeugt.

6. *Gaußverteiltes Rauschen*: bezüglich des Mittelwertes des Rauschens werden 7 Unterklassen mit den jeweiligen Mittelwerten 0, 1, 2, 3, 5, 7 und 10 erzeugt. Mittelwert 0 bedeutet, daß keine Überlagerung mit Gaußverteiltem Rauschen vorliegt. Die Unterklasse 0 ist genauso groß wie die Unterklassen 1, 2 und 3 bzw. 5, 7 und 10 zusammen. Für die Unterklassen 1, 2 und 3 werden bezüglich des Sigmas nur Unterklassen mit Sigma $= 1$ und 2 , für die Unterklassen 5, 7 und 10 nur solche mit Sigma $= 1$, 3 und 5 erzeugt.

7. *Nullpunktsverschiebungen*: als Wahrscheinlichkeit für das Auftreten von Nullpunktsverschiebungen einzelner Pads wird 10% gewählt. Die Nullpunktsverschiebungen werden im Intervall [min. Padamplitude, max. Padamplitude] uniform verteilt generiert.

8. *Entartete Hits*: Hits, für die nur das zentrale Pad größer Null ist, werden verworfen. Hits mit Defekten, die nicht mindestens zwei Pads mit Amplituden größer Null besitzen, werden verworfen. Es gibt keine Möglichkeit, solche Hits vom Untergrund zu trennen.

9. *Kompatibilität mit dem bisherigen Hitrekonstruktionsalgorithmus*: alle erzeugten Matrizen, die dem oben beschriebenen Schwerpunktsalgorithmus nicht genügen, werden verworfen.

Insgesamt werden also $10 \cdot 2 \cdot (1 + 3 \cdot 2 + 3 \cdot 3) = 320$ Unterklassen in der Klasse der MC-Einzelhits erzeugt.

5.2.2 Die Erzeugung der Monte-Carlo-Doppelhits

Zur Erzeugung der **MC-Doppelhits** wird die Strategie verfolgt, sie durch Superposition zweier nach Abschnitt 5.2.1 erzeugter MC-Einzelhits zu gewinnen.
Hierbei werden zwei MC-Einzelhits überlagert, wobei die Koordinaten des Abstandes zwischen den Schwerpunkten der beiden Einzelhits im offenen Intervall]0 Pads,5 Pads[uniform verteilt sind. Es werden nur solche Matrizen akzeptiert, die dem oben beschriebenen Schwerpunktsalgorithmus genügen. Außerdem wird das Vorhandensein einer unteren und oberen Schwelle für die Padamplitude (und damit auch für die Hitamplitude) berücksichtigt. Die Parametereinstellungen sind mit denen im vorangehenden Abschnitt identisch.

5.2.3 Die Erzeugung der Monte-Carlo-Untergrundelemente

Zur Beschreibung des Untergrundes bezüglich der Klassifizierungsprobleme hat es sich als sinnvoll erwiesen, vier Grundklassen von MC-Untergrundelementen zu definieren:

1. **Ebenen.** Das sind 5×5-Matrizen, deren Elementen ein konstanter Wert zugewiesen wird.

2. **Hitformen.** Das sind 5×5-Matrizen von zweidimensional Gaußverteilten Ladungsverteilungen, die ein kleineres oder größeres Sigma als 0.81 Pads in RICH 1 bzw. 0.63 Pads in RICH 2 besitzen.

3. **Einzelpads.** Das sind 5×5-Matrizen, für die das zentrale Element von Null verschieden ist, und für die alle Elemente außerhalb des zentralen Elements verschwinden.

4. **Rauschelemente.** Das sind 5×5-Matrizen, deren Elemente Gaußverteilte Werte mit einem bestimmten Mittelwert und Sigma haben.

Aus jeder dieser Grundklassen ist in Abb. 5.1 ein Untergrundelement als Beispiel dargestellt. Um möglichst realistische MC-Untergrundelemente zu erhalten, müssen in Analogie zur Erzeugung der MC-Einzelhits folgende Modifikationen berücksichtigt werden: Überlagerung mit Gaußverteiltem Rauschen, Vorhandensein von einzelnen defekten Pads, einer unteren und einer oberen Schranke für die Padamplitude sowie von Nullpunktsverschiebungen einzelner oder mehrerer Pads.

Zur Erzeugung möglichst realistischer MC-Untergrundelemente werden folgende Merkmalsparameter berücksichtigt:

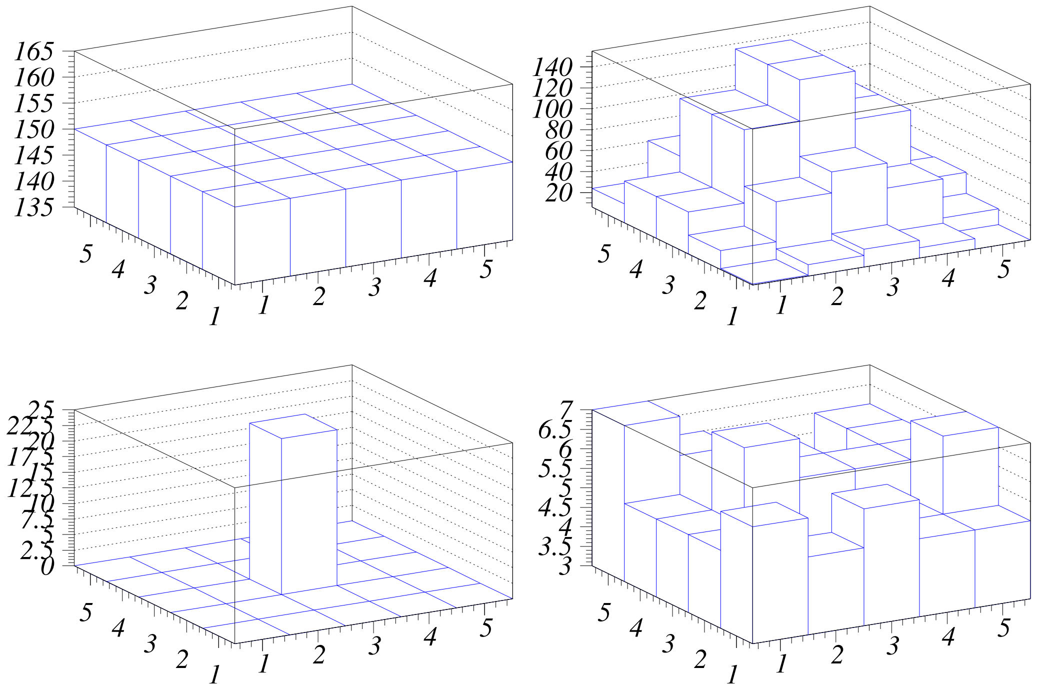

Abbildung 5.1: *Beispiele für MC-Untergrundelemente aus jeder der vier Grundklassen.* Links oben ein Beispiel aus der Klasse der Ebenen, rechts oben aus der der Hitformen (hier ist $\sigma = 1.22$ Pads), links unten aus der der Einzelpads und rechts unten aus der der Rauschelemente (hier ist $\mu = 5$ und $\sigma = 1$). Alle Darstellungen sind Lego-Plots.

1. *Schwerpunkt*: die Schwerpunktskoordinaten der Hits werden im zentralen Pad der 5×5-Matrix uniform verteilt erzeugt.

2. *Schwellen*: als minimale Padamplitude wird 3, als maximale 235 gewählt (zur Begründung siehe vorangehenden Abschnitt).

3. *Parameter der Ebenen*: die Amplitude der Ebenen wird im Intervall [minimale Padamplitude, maximale Padamplitude] uniform verteilt erzeugt.

4. *Parameter der Hitformen*: bezüglich des Faktors, um den das Sigma einer Hitshape größer oder kleiner als 0.81 Pads in RICH 1 bzw. 0.63 Pads in RICH 2 ist, werden jeweils gleichgroße Unterklassen mit den Faktoren 0.25, 0.5, 1.5, 2 und 3 generiert. Bezüglich der Verteilung der Amplituden der Hitformen werden jeweils gleichgroße Unterklassen mit exponentieller und uniformer

Verteilung erzeugt (zur Begründung siehe vorangehenden Abschnitt). Die Hitformen werden mit einer mittleren Amplitude von 141 in RICH 1 bzw. 245 in RICH 2 erzeugt. Das entspricht den gemessenen Werten. Als maximale Amplitude der Hitformen wird $(25 \cdot 235 =)$ 5875 gewählt.

5. *Parameter der Einzelpads*: die Amplitude der Einzelpads wird im Intervall [minimale Padamplitude, maximale Padamplitude] uniform verteilt erzeugt.

6. *Parameter der Rauschelemente*: bezüglich des Mittelwertes des Rauschens werden 7 Unterklassen mit dem jeweiligem Mittelwert 0, 1, 2, 3, 5, 7 und 10 erzeugt. Mittelwert 0 bedeutet, daß keine Überlagerung mit Gaußverteiltem Rauschen vorliegt. Die Unterklasse 0 ist genauso groß wie die Unterklassen 1, 2 und 3 bzw. 5, 7 und 10 zusammen. Für die Unterklassen 1, 2 und 3 werden bezüglich des Sigmas Unterklassen mit Sigma = 1 und 2 , für die Unterklassen 5, 7 und 10 nur solche mit Sigma = 1, 3 und 5 erzeugt.

7. *Defekte*: es werden 10 Unterklassen mit jeweils 0, 1,..., 9 Defekten generiert, wobei die Unterklasse mit 0 Defekten genauso viele Lernmuster enthält wie die Unterklassen mit 1 bis 3 bzw. 4 bis 6 bzw. 7 bis 9 Defekten zusammen. Die Positionen der Defekte in der 5×5-Matrix werden uniform verteilt erzeugt.

8. *Überlagertes Gaußverteiltes Rauschen*: es gilt das Gleiche wie in Punkt 6. Gesagte.

9. *Nullpunktsverschiebungen*: als Wahrscheinlichkeit für das Auftreten von Nullpunktsverschiebungen einzelner Pads wird 10% gewählt. Die Nullpunktsverschiebungen werden im Intervall [min. Padamplitude, max. Padamplitude] uniform verteilt generiert.

10. *Kompatibilität mit dem bisherigen Hitrekonstruktionsalgorithmus*: alle erzeugten Matrizen, die dem oben beschriebenen Schwerpunktsalgorithmus nicht genügen, werden verworfen.

Man erhält also insgesamt $2 \cdot 10 \cdot (1 + 3 \cdot 2 + 3 \cdot 3) + 5 \cdot 2 \cdot 10 \cdot (1 + 3 \cdot 2 + 3 \cdot 3) + 10 \cdot (1 + 3 \cdot 2 + 3 \cdot 3) = 2080$ Unterklassen in der Klasse der MC-Untergrundelemente.

5.3 Das Trainieren und Testen der Netzwerke

5.3.1 Das Trainieren

Wie bereits in Abschnitt 4.2 ausgeführt, verkörpern die Neuronen in der Eingabeschicht die zur Verfügung stehenden Meßwerte. Die Neuronen in der Ausgabeschicht stellen eine beliebige Repräsentierung von Klassen dar. Bei unseren

Klassifizierungsproblemen handelt es sich immer um die Trennung zweier Klassen: Trennung von Einzelhits und Untergrund, Trennung von Einzel- und Doppelhits sowie Trennung von Einzel- und Doppelhits einerseits und Untergrund andererseits. Die Klassifizierung von Einzelhits, Doppelhits und Untergrund erhält man dann durch Hintereinanderschaltung der Trennung von Einzel-/Doppelhits und Untergrund im ersten Schritt und Trennung der erhaltenen Einzel-/Doppelhits im zweiten Schritt. Somit könnte man zwei Ausgabeneuronen verwenden: eine, die für die Zugehörigkeit zur ersten Klasse den Wert 1 liefert, und eine andere, die für die zweite Klasse den Wert 0 annimmt. Da in unserem Fall jedoch beide Klassen jeweils komplementär sind, genügt ein Neuron zur Unterscheidung.

Wie in Abschnitt 5.4 gezeigt wird, ist die 25-30-1-Architektur, wenn man nur dreilagige Netze betrachtet, bzw. die 25-10-10-1-Architektur, wenn man nur vierlagige Netze betrachtet, optimal für unsere Klassifizierungsprobleme. Das 25-30-1-Netz besitzt $(25+1)\cdot(30+1)+(30+1) = 837$ Gewichte - der Eingabeschicht und jeder verborgenen Schicht wird ein zusätzliches Neuron mit konstanter Aktivität zugeordnet, das die Schwelle repräsentiert; siehe Abschnitt 4.2. Das 25-10-10-1-Netz besitzt entsprechend $(25+1)\cdot(10+1)\cdot(10+1)+(10+1) = 3157$ Gewichte. Damit die in Abschnitt 4.4 angegebene Faustregel erfüllt ist, müssen also mindestens $10 \cdot 3157 = 31570$ Lernmuster von jeder zu trennenden Klasse erzeugt werden. Aus diesem Grund wurden zur Klassifizierung von Einzelhits, Doppelhits und Untergrund Lernsamples der Größe $8 \cdot 10^4$ Lernmuster gemäß Abschnitt 5.2 erzeugt.

Ein weiterer wichtiger Punkt, der beim Trainieren eines Neuronalen Netzes beachtet werden muß, ist folgender: die Lernmuster zum Trainieren dürfen nicht zufallsverteilt eingegeben werden, sondern es muß immer abwechselnd ein Lernmuster aus jeder Klasse eingegeben werden. Das in unserem Fall paarweise geordnete Lernsample verhindert, daß einem Netzwerk nacheinander sehr viele Input-Vektoren nur einer Klasse präsentiert werden, und dieses damit alle Lernmuster als dieser Klasse zugehörig ansieht. Mit Netzen, die auf diese Weise trainiert wurden, konnte keine Trennung erreicht werden.

Abgesehen von der Netzwerkarchitektur und den Input-Variablen, die in eigenen Abschnitten noch behandelt werden, haben sich bei den durchgeführten Netzwerksimulationen bezüglich des Konvergenzverhaltens der Netze folgende Parameter als relevant erwiesen:

1. Die Lernrate η. Anschaulich läßt sich η als die Schrittweite interpretieren, mit der sich das Netz beim Suchprozeß im Gewichtsvektorraum lokal in Richtung des Minimums bewegt.

2. Der Momentumfaktor α. In anschaulicher Interpretation dient α dazu, die Lernrate in Parameterregionen mit „flachem" Gradienten zu vergrößern, oder in Bereichen mit „steilem" Gradienten mögliche Oszillationen zu dämpfen,

und die Bewegung des Suchprozesses in die richtige Richtung zu lenken.

3. Die Temperatur T. Die anschauliche Bedeutung von T liegt darin, daß es die Steilheit der Fehleroberfläche bestimmt.

Folgender Ablauf des Netzwerktrainings hat sich im Laufe dieser Arbeit als optimal in Bezug auf die Konvergenzgeschwingigkeit herauskristallisiert:

- $\alpha = 0$ während des gesamten Trainings.
- $250 - 375$ Epochen mit $T = 1$ und $\eta = 0.005$.
  Mit diesem relativ großen Wert für η wird erreicht, daß sich das Netz mit großen „Schritten" auf der Fehleroberfläche bewegt und so kleinere lokale Minima überspringt. Die Anzahl der Epochen ist zumeist ausreichend, daß das Netz in die Nähe des absoluten Minimums gelangt.
- $175 - 200$ Epochen mit $T = 1$ und $\eta = 0.001$.
  Das kleinere η ermöglicht dem Netz, gegen das absolute Minimum zu konvergieren.
- Hat das Netzwerk eine stabile Lage erreicht, kann man sich nie sicher sein, daß nicht ein lokales Minimum vorliegt. Um dem Netz zu ermöglichen, sich aus einem vermeintlichen lokalen Minimum zu befreien, wird zum Abschluß noch einmal mit relativ großem T bzw. η trainiert: $150 - 175$ Epochen mit $T = 3$ und $\eta = 0.005$.
  (Dies entspricht in einem mechanischen Analogon dem kräftigen Anstoßen einer Kugel, die in einer Mulde liegt. Entspricht diese Position nur einem relativen Minimum, so besteht bei einem kräftigen Stoß die Möglichkeit, daß sich die Kugel über die begrenzenden Wälle hinwegsetzt und in eine tiefere Position rollt. Andernfalls wird sie wieder die Ausgangslage einnehmen.)

5.3.2 Das Testen

Die Testsamples, die von den jeweiligen Lernsamples *unabhängig* sein müssen, bestanden aus jeweils $2.5 \cdot 10^4$ Lernmuster, da bei dieser Größe noch gewährleistet ist, daß die Testsamples *umfassend genug* sind (zur Erklärung der Begriffe siehe Abschnitt 4.4).

Das Testen eines Netzwerkes erfolgte folgendermaßen: jeweils nach Beendigung einer bestimmten Anzahl von Trainingszyklen wurde das Netz mit seinen bis dahin erreichten Gewichtsparametereinstellungen auf das Testsample angewendet und bestimmt, ob das Netzwerk die Lernmuster des Testsamples richtig oder falsch

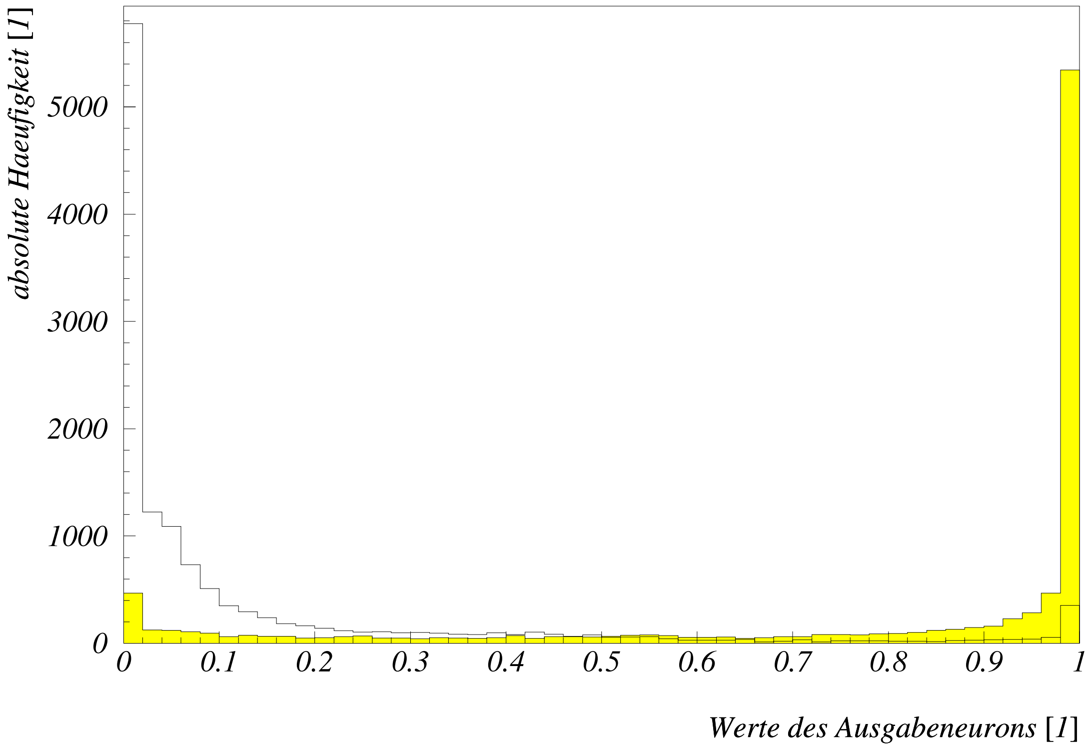

Abbildung 5.2: *Typische Verteilung des Ausgabeneurons eines trainierten Netzes.* Die Häufigkeitsverteilung der ersten Klasse ist dunkel, die der zweiten Klasse hell aufgetragen. Das Netz wurde zur Klassifizierung von Einzelhits und Untergrund trainiert.

klassifizierte. Die Ergebnisse wurden durch Effizienz- und Rejektionswerte angegeben, die im folgenden definiert werden. Der Anteil an Lernmuster aus der ersten Klasse des Testsamples, die durch ein Neuronales Netzwerk richtig klassifiziert worden sind, heißt **Effizienz** des Netzes. Entsprechend wird der Anteil an Lernmuster aus der zweiten Klasse des Testsamples, die durch ein Neuronales Netzwerk richtig klassifiziert worden sind, als **Rejektion** des Netzes bezeichnet. (Zur Erklärung der Begriffe erste und zweite Klasse siehe Abschnitt 5.3.1.) Das arithmetische Mittel aus Effizienz und Rejektion eines Netzes ist ein Maß für seine Leistungsfähigkeit, genauer für die Fähigkeit des Netzes zu generalisieren, weshalb diesem arithmetische Mittel ein eigener Name gegeben und als **Leistungsfähigkeit** bezeichnet wird.

Abb. 5.2 zeigt eine typische Verteilung des Ausgabeneurons eines für unsere Klassifizierungsprobleme trainierten Netzes, die man erhält, wenn man das Netz auf das

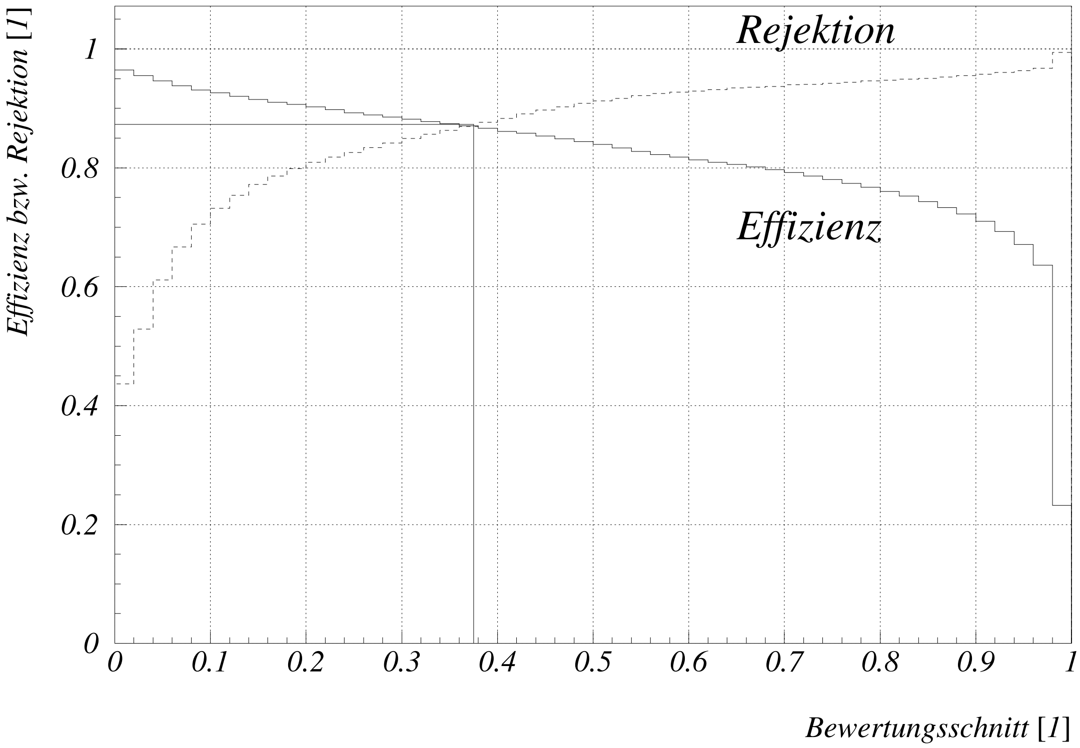

Abbildung 5.3: *Effizienz bzw. Rejektion des Netzes aus Abb. 5.2 über der Größe des Bewertungsschnittes.* Im Schaubild ist eingezeichnet, wie man den Gleichgewichtspunkt des Netzes findet.

Testsample anwendet. Im Idealfall befindet sich das Netzwerk jetzt im absoluten Minimum der Fehleroberfläche, was wir aber nicht mit Sicherheit wissen. Man erkennt, daß das Netz nicht in der Lage ist, eine perfekte Abbildung durchzuführen, d.h. nicht jedes Element aus der ersten Klasse liefert einen Wert von Eins und nicht jedes Element aus der zweiten Klasse den Wert Null. Es ist jedoch ersichtlich, daß die Leistungsfähigkeit des Netzes sehr gut ist ($\approx 90\%$); bei den Werten Null und Eins haben sich ausgeprägte Maxima gebildet, die zur Mitte hin stark abfallen. Somit können die beiden Klassen durch einen Schnitt in der Mitte getrennt werden. Das Netz ist also in der Lage zu generalisieren, was die Beendigung des Trainierens rechtfertigt. Da nach Abschnitt 4.4 das Ziel des Trainierens und Testens war, die Fähigkeit der Generalisierung zu erreichen, *wurden alle in dieser Arbeit durchgeführten Trainingsabläufe nach Erreichen der maximalen Leistungsfähigkeit des Netzes beendet.*
Es hat sich gezeigt, daß sich für alle behandelten Klassifizierungsprobleme eine sehr gute Leistungsfähigkeit der Netze ($\gtrsim 90\%$) und damit die Fähigkeit der Generali-

sierung erreichen läßt.

Integriert man die beiden Häufigkeitsverteilungen aus Abb. 5.2, so erhält man die Effizienz bzw. Rejektion des Netzes über der Größe des Bewertungsschnittes. Diese Kurven sind in Abb. 5.3 dargestellt. Der Schnittpunkt der Kurven der Effizienz und der Rejektion eines Netzes über der Größe des Bewertungsschnittes heißt **Gleichgewichtspunkt** des Netzes. (Die Werte für Effizienz, Rejektion und Leistungsfähigkeit sind im Gleichgewichtspunkt gleich.) In unserem Beispiel (Abb. 5.3) befindet sich der Gleichgewichtspunkt bei 0.38 mit einer Effizienz bzw. Rejektion bzw. Leistungsfähigkeit von 0.88. *Bei der Anwendung der entwickelten Netzwerke wurde immer vom Gleichgewichtspunkt des betreffenden Netzes ausgegangen.*

5.4 Die Netzwerkarchitektur

Die Bestimmung der hinsichtlich Laufzeit und/oder Leistungsfähigkeit (Generalisierung) optimalen Anzahl von Neuronen in der/den verborgenen Schicht(en) gehört zu den ungelösten Problemen im Bereich der Neuronalen Netzwerke. Die Theoreme 4.3.1 und 4.5.1 machen keine Aussage über die Auswahl einer geeigneten Netzwerkarchitektur für ein reales Problem; sie schaffen nur die Gewißheit, daß eine geeignete Architektur existieren muß. Es gibt bis heute keine allgemeingültige Regel für dieses Problem; vielmehr kursieren mehr oder weniger fundierte Faustregeln.

Es ist klar, daß für ein dreilagiges Netz die Anzahl der verborgenen Neuronen mindestens $n+1$ betragen muß, wobei n die Anzahl der Eingabeneuronen beziffert, um ein geschlossenes Volumen im n-dimensionalen Raum beschreiben zu können. Desweiteren gilt: je größer die Anzahl der verborgenen Neuronen ist, desto mehr Wege zum absoluten Minimum gibt es, da mit jedem zusätzlichen Neuron die Dimension des Gewichtsparameterraumes anwächst, und somit auch die Anzahl der Permutationen der Gewichtsparameter ansteigt.
Nach Abschnitt 4.4 sollte zur Vermeidung von Übertrainieren die Faustregel Anzahl der Gewichte $\leq 0.1 \cdot$ Anzahl der Lernmuster erfüllt sein. Diese obere Schranke für die Anzahl der Neuronen spielt in unserem Fall jedoch keine Rolle, da wir die Lernmuster mit Monte-Carlo-Techniken erzeugen. Viele Neuronen benötigen allerdings viel Rechenzeit.
Das Ergebnis sollte daher mit möglichst wenig Neuronen erreicht werden. Es ist also sinnvoll, mit mehreren Neuronen zu lernen, und nach erfolgter Konvergenz die Anzahl so lange zu reduzieren, wie noch eine ausreichende Leistungsfähigkeit erreicht wird.

Dieses Optimierungsverfahren bringt im allgemeinen[1] keine Steigerung der Leistungsfähigkeit des Netzwerkes, ermöglicht jedoch die Ermittlung der minimalen Anzahl von Neuronen in der/den verborgenen Schicht(en), ohne daß die Leistungsfähigkeit des Netzes deutlich abnimmt.

Das oben beschriebene Optimierungsverfahren wurde sowohl bezüglich drei- als auch vierlagiger Netze angewendet. In Tabelle 5.2 sind einige der untersuchten Architekturen zusammengestellt; die angegebenen Leistungsfähigkeits-Werte beziehen sich auf die Klassifizierung von MC-Einzelhits und MC-Untergrundelementen. Für die anderen Klassifizierungsprobleme ergaben sich ähnliche Werte, so daß diese Tabelle Aussagen bezüglich aller Klassifizierungsprobleme zuläßt. Man erkennt, daß die 25-30-1-Architektur für eine verborgene Schicht bzw. die 25-10-10-1-

Architektur	Leistungsfähigkeit
Dreilagige Netze:	
25-100-1	88.3%
25-50-1	88.1%
25-40-1	87.3%
25-30-1	87.6%
25-26-1	84.7%
Vierlagige Netze:	
25-30-10-1	87.9%
25-20-20-1	87.4%
25-10-10-1	87.1%
25-5-5-1	85.2%

Tabelle 5.2: *Leistungsfähigkeit für einige der im Rahmen des im Text beschriebenen Optimierungsverfahrens untersuchten Netzwerkarchitekturen.* Die angegebenen Leistungsfähigkeits-Werte beziehen sich auf die Klassifizierung von Einzelhits und Untergrund; für die anderen Klassifizierungsprobleme ergaben sich ähnliche Werte.

[1] Falls sich das Netzwerk in einem lokalen Minimum befindet, kann es durch die Anwendung dieses Verfahrens aus diesem in das absolute Minimum gelangen, und sich so die Leistungsfähigkeit noch erhöhen.

Architektur für zwei verborgene Schichten optimal bezüglich des oben beschriebenen Verfahrens ist. Zwischen diesen beiden Netzen besteht kein signifikanter Unterschied bezüglich der Leistungsfähigkeit; mit dem dreilagigen Netz läßt sich jedoch überflüssige Rechenzeit sparen. Eine Vergrößerung dieser Netze zeigt empirisch keine Verbesserung mehr; die Neuronen in der/den verborgenen Schicht(en) beginnen lediglich damit, die gleichen Gewichte anzunehmen und identische Schritte durchzuführen.
Die 25-30-1-Architektur stellt somit eine für unserer Klassifizierungsprobleme optimale Netzwerkarchitektur dar.

5.5 Die Input-Variablen

Die Tatsache, daß die 5×5-Matrix eines Hits die zu seiner Beschreibung optimale Matrix darstellt (siehe Anhang B), suggeriert, die 25 Elemente (Inhalte) dieser Matrix als Input-Variable für Neuronale Netzwerke zu verwenden, die zur Klassifizierung von einzelnen Hits und/oder Doppelhits eingesetzt werden sollen. Anstatt dieser **absoluten** Variablen wäre es noch denkbar, **relative** Variablen zu verwenden, d.h. man nimmt als Variablen die Verhältnisse der Inhalte der Matrixelemente, die ungleich dem zentralen Element sind, zum Inhalt des zentralen Elements - insgesamt 24 - sowie als weitere Variable die Summe aller Padinhalte. Für Doppelhits wäre als 25. oder 26. Variable noch der Abstand der Pads mit den beiden größten Amplituden denkbar.

Um zu bestimmen, ob die absoluten oder relativen Variablen den zur Lösung unserer Klassifizierungsprobleme optimalen Satz an Input-Variablen darstellen, wurde folgende Untersuchung durchgeführt:
ein Netzwerk wurde in der in den Abschnitten 5.2, 5.3 und 5.4 erklärten Vorgehensweise trainiert, wobei das Netz zum einen mit den 25 Inhalten (Amplituden) der 5×5-Matrizen, d.h. mit absoluten Variablen trainiert wurde; zum anderen wurde das Netz mit relativen Variablen trainiert, d.h. als Input-Variablen wurden die 24 Verhältnisse der Inhalte der Matrixelemente, die ungleich dem zentralen Element sind, zum zentralen Element sowie die Summe aller Padinhalte verwendet.
Für relative Variablen ergab sich das in Abb. 5.4 gezeigte Ergebnis. Man erkennt, daß die Verteilungen eine Peakstruktur aufweisen, während die Verteilungen des mit absoluten Variablen (Abb. 5.2) trainierten Netzes glatt sind. Zudem liegt für ersteres Netz das Maximum der Häufigkeitsverteilung der ersten Klasse nicht bei Eins, wie es sein sollte. Offenbar rührt die Peakstruktur daher, daß das Netzwerk Lernmuster mit Defekten im zentralen Pad nicht einzuordnen weiß, da die Peakstruktur verschwindet, wenn man mit Lernmuster trainiert, die keine Defekte besitzen. Dies wird durch die Tatsache bekräftigt, daß in den beiden Häufigkeitsverteilungen zwei größere Peaks an derselben Stelle auftreten. Die Verhältnisse

der Padamplituden gehen dann gegen Unendlich; aber gleichgültig, durch welche Werte diese Verhältnisse ersetzt werden, es ergibt sich immer eine Peakstruktur in den Häufigkeitsverteilungen der zu trennenden Klassen.

Wie in Abschnitt 5.3.2 gesehen, läßt sich mit den 25 absoluten Variablen eine

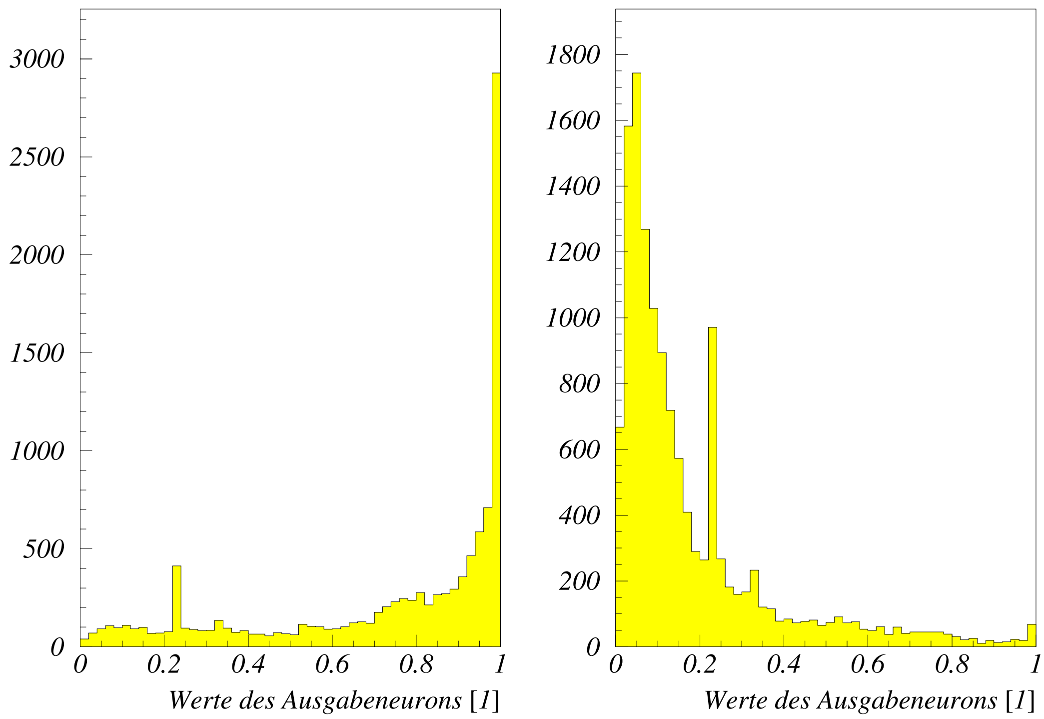

Abbildung 5.4: *Häufigkeitsverteilung der ersten (links) und der zweiten Klasse (rechts) des mit relativen Variablen trainierten Netzes.* Dem Netz lag dasselbe Lern- bzw. Testsample zugrunde wie dem aus Abb. 5.2.

sehr gute Leistungsfähigkeit der Netze erreichen. Anstatt der 25 absoluten oder relativen Variablen wäre es natürlich auch denkbar, weniger aber komplexere Variablen zu benutzen. Selbst wenn sich mit solchen Variablen eine vergleichbar gute Leistungsfähigkeit wie bei Verwendung der 25 absoluten Variablen erreichen ließe - falls dies überhaupt möglich ist -, so ist die Benutzung der 25 absoluten Variablen aus folgendem Grund dennoch vorzuziehen: sie stellen bei der Anwendung der entwickelten Netzwerke auf realen Daten die am leichtesten zu gewinnenden Input-Variablen dar; von jedem zu klassifizierenden Muster sind nur die Inhalte seiner 5×5-Matrix auszulesen und ansonsten keine weiteren Berechnungen mehr durchzuführen.

Somit ist klar, daß die absoluten Variablen den zur Lösung unserer Klassifizierungsprobleme optimalen Satz an Input-Variablen darstellen.

Die **Normierung der Input-Variablen** ist ein weiterer wichtiger Punkt bei der Anwendung der Netzwerke. Abb. 5.5 zeigt die Häufigkeitsverteilung des Ausgabeneurons, die man für ein Netzwerk erhält, das mit den nicht normierten absoluten Padinhalten als Input-Variablen trainiert worden ist. Es ist eine Peakstruktur in den Verteilungen zu erkennen, und die Maxima liegen nicht bei Null bzw. Eins. Es erfolgt keine wirkliche Trennung, sondern lediglich eine Minimierung des globalen Fehlers durch Ausnutzung der unterschiedlichen Häufigkeiten bestimmter Unterklassen der Daten. Dies wird durch die Maxima deutlich, die in beiden Häufigkeitsverteilungen an derselben Stelle, aber mit unterschiedlicher Höhe auftreten.

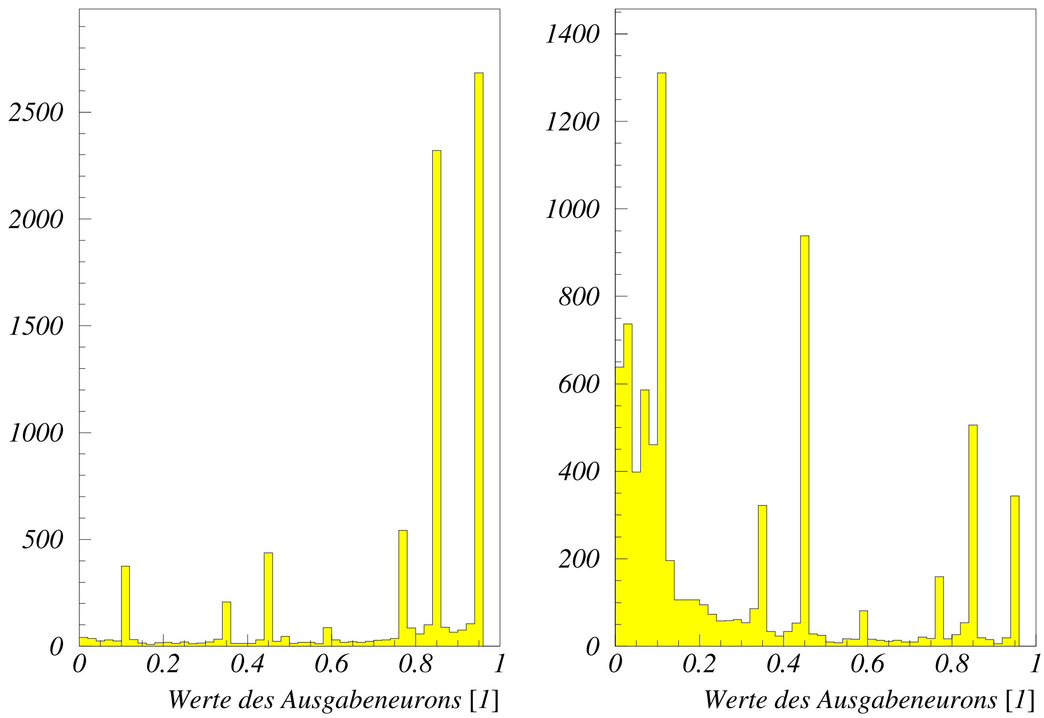

Abbildung 5.5: *Häufigkeitsverteilung der ersten (links) und der zweiten Klasse (rechts) des mit nicht normierten absoluten Variablen trainierten Netzes.* Dem Netz lag dasselbe Lern- bzw. Testsample zugrunde wie dem aus Abb. 5.2.

Bei dem in Abb. 5.2 betrachteten Netz wurden die Input-Variablen auf das Intervall $[0,1]$ linear normiert, bevor sie den Netzen eingegeben wurden. In unserem Fall, da

es sich um Padamplituden handelt, ist die Normierung besonders einfach, weil der Wertebereich der Padinhalte bekannt ist, und alle vorkommenden Amplituden im Lernsample repräsentiert sind. In einem solchen Fall ist eine lineare Normierung nach $[0,1]$ ausreichend, wie man folgendermaßen sieht:
die Notwendigkeit einer Normierung wird klar, wenn wir die in Abschnitt 4.1 durch Gl. 4.1 beschriebene Funktionsweise eines Neurons betrachten. Ist einer der an einem Neuron anliegenden Eingangswerte sehr viel größer als die restlichen, so wird die Aktivität des Neurons durch diesen Wert numerisch dominiert. Zwar ist das Netz unter gewissen Voraussetzungen in der Lage, die richtige „Gewichtung" dieses Eingangswertes zu lernen, aber dies erfordert sehr viel mehr Lernzyklen, als wenn alle Eingangswerte von der gleichen Größenordnung sind. Damit erhöht sich aber auch die Wahrscheinlichkeit, daß das Netz in einem lokalen Minimum steckenbleibt. Damit nicht bestimmte Eingangswerte unterdrückt werden, muß die Normierung außerdem die Eigenschaft haben, die Eingangsvariablen nach Möglichkeit gleichmäßig in ihrem Wertebereich zu verteilen.
In unserem Fall erfüllt die lineare Normierung nach $[0,1]$ beide Bedingungen und wird daher als Normierung der Input-Variablen benutzt.

5.6 Die Rejektion der einzelnen Untergrundklassen

Die Untergrundklasse der Hitformen läßt sich in Unterklassen mit konstantem Sigma unterteilen. Zur Bestimmung der Rejektion der Hitformen in Abhängigkeit von ihrem Sigma wurden von jeder solchen zu untersuchenden Untergrundklasse $4 \cdot 10^4$ Lernmuster nach Abschnitt 5.2.3 erzeugt. Aus diesen wurde zusammen mit $4 \cdot 10^4$ MC-Einzelhits ein Lernsample gebildet. Mit jedem so erhaltenen Lernsample wurde in der in den Abschnitten 5.3, 5.4 und 5.5 erklärten Vorgehensweise ein Netzwerk trainiert und der für die Rejektion erhaltene Wert festgehalten. Die Ergebnisse sind in Abbildung 5.6 über dem Verhältnis Sigma der entsprechenden Hitshapeklasse zu Sigma der Einzelhits aufgetragen. Abb. 5.6 zeigt auch das analoge Schaubild für **ideale** MC-Einzelhits und Hitformen, d.h. MC-Einzelhits und Hitformen, bei denen Rauschen, Defekte und Nullpunktsverschiebungen nicht berücksichtigt sind.

Schlüsselt man die realistischen Hitformen in Unterklassen auf, bei denen *nur* Rauschen bzw. *nur* Defekte berücksichtigt werden, so stellt man fest, daß die Rejektion im wesentlichen durch das Vorhandensein von Defekten verlorengeht: ist nur Rauschen berücksichtigt, so liegt die Rejektion im Mittel ca. 2% unter den für ideale Hitformen gemessenen Werten, wohingegen man durchschnittlich etwa 10% an Rejektion verliert, wenn nur Defekte berücksichtigt werden.
Wie aus Abb. 5.6 ersichtlich wird, sind ideale Hitformen bis zu einem Sigma von

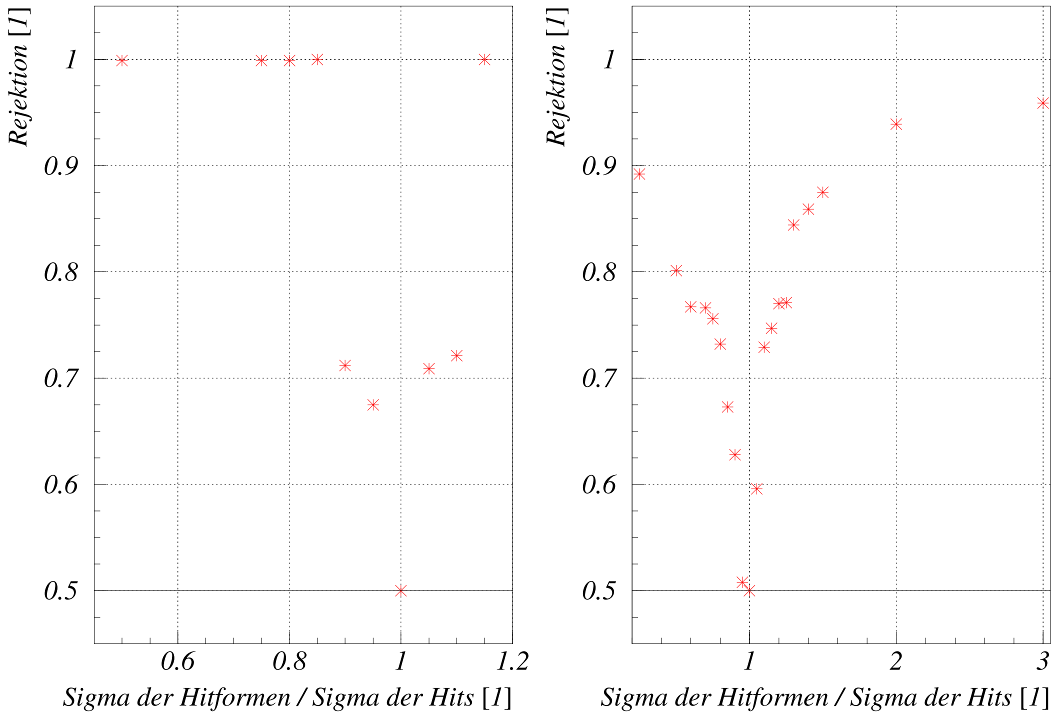

Abbildung 5.6: *Rejektion idealer (links) und realistischer (rechts) Hitformen in Abhängigkeit von ihrem Sigma.* Aufgetragen ist die Rejektion des Netzes mit der besten Leistungsfähigkeit über dem Verhältnis vom Sigma der entsprechenden Hitformen zum Sigma der Einzelhits.

85% des Sigmas der Einzelhits bzw. ab einem Sigma von 115% sehr gut von idealen Einzelhits zu trennen, d.h. die Rejektion ist dort über 95%. Realistische Hitformen sind bis zu einem Sigma von 50% des Sigmas der Einzelhits bzw. ab einem Sigma von 130% von MC-Einzelhits zu trennen, d.h. die Rejektion beträgt dort über 80%.

Zur Bestimmung der Rejektion der restlichen Untergrundklassen wurde analog zur Untersuchung der Hitformen vorgegangen. Die erhaltenen Ergebnisse sind in Tabelle 5.3 zusammengestellt. Dabei steht der Begriff **ideal** wieder dafür, daß Rauschen, Defekte und Nullpunktsverschiebungen nicht berücksichtigt worden sind; der Begriff **realistisch** steht für die Berücksichtigung dieser Modifikationen.
Aus Tab. 5.3 geht hervor, daß Ebenen, Einzelpads und Rauschelemente auch dann, wenn sie so realistisch wie möglich erzeugt werden, sehr gut, d.h. mit einer Rejektion von über 95%, von Hits getrennt werden können.

Untergrundklasse	Rejektion im	
	idealen Fall	realistischen Fall
Hitformen:		
Faktor = 0.85	100%	74%
Faktor = 0.9	68%	61%
Faktor = 1.3	100%	81%
Ebenen	100%	98%
Einzelpads	98%	95%
Rauschelemente	98%	96%

Tabelle 5.3: *Rejektion der restlichen Untergrundklassen.* Die Begriffe „ideal" und „realistisch" sind im Text erklärt.

5.7 Die Trennung von Monte-Carlo-Daten und realen Daten

In diesem Abschnitt wird untersucht, ob, und wenn ja, wie gut die erzeugten Monte-Carlo-Daten durch Neuronale Netze von den entsprechenden realen Daten getrennt werden können.

Zunächst wird die Trennung von realen einzelnen Hits und MC-Einzelhits betrachtet. Hierzu wurden 3000 reale einzelne Hits zum Trainieren und weitere 1000 zum Testen aus S-Au-Ereignissen dadurch gewonnen, daß sie visuell überprüft und anschließend klassifiziert wurden. Aus diesen wurde zusammen mit 3000 bzw. 1000 MC-Einzelhits ein Lern- bzw. Testsample gebildet und in der in den Abschnitten 5.3, 5.4 und 5.5 erklärten Vorgehensweise ein Netzwerk trainiert. Dabei wurden die realen Hits als Elemente der ersten Klasse und die MC-Hits als Elemente der zweiten Klasse im Sinne von Abschnitt 5.3.1 betrachtet; den realen Hits wurde also der Wert 1, den MC-Hits der Wert 0 im Ausgabeneuron zugeordnet. In Abb. 5.7 sind die erhaltenen Häufigkeitsverteilungen des Ausgabeneurons für beide Klassen dargestellt.
Bei der Untersuchung der Trennung von realem Untergrund und MC-Untergrundelementen ergaben sich in analoger Vorgehensweise wie oben die in Abb. 5.8 gezeigten Verteilungen.

Aus Abb. 5.7 wird deutlich, daß das Netz - wenn überhaupt - nur sehr schlecht in

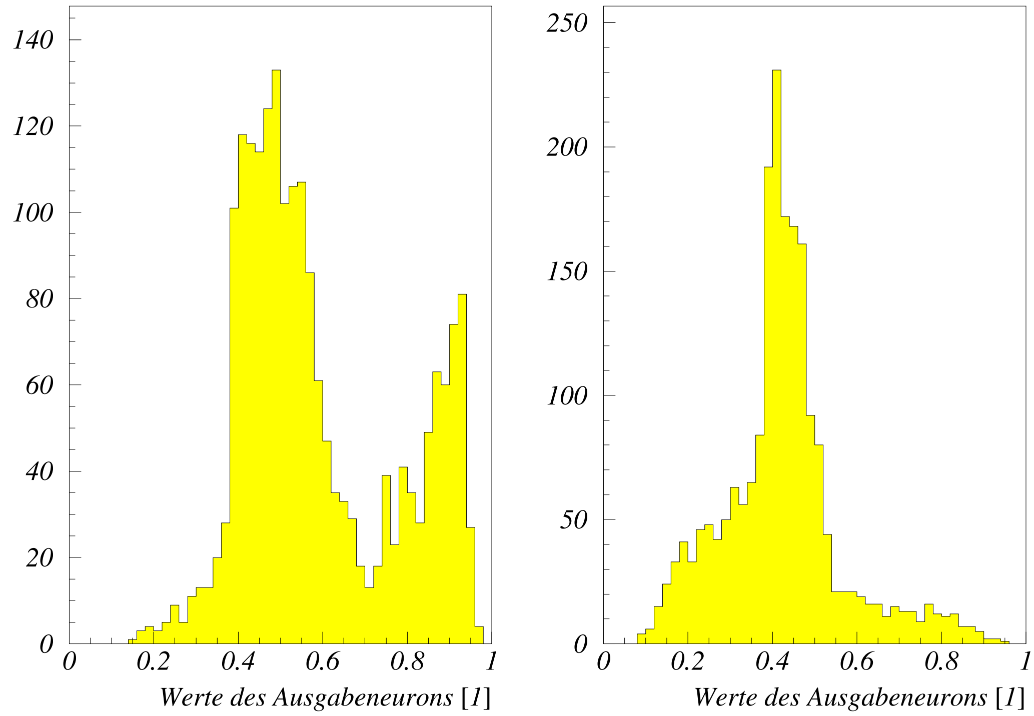

Abbildung 5.7: *Häufigkeitsverteilung des Ausgabeneurons des zur Trennung von realen einzelnen Hits (links) und MC-Einzelhits (rechts) trainierten Netzes.* Der beste Leistungsfähigkeits-Wert des Netzes liegt bei 71.4%.

der Lage ist, die beiden Klassen zu unterscheiden. In Abb. 5.8 ist zu sehen, daß diesmal scheinbar eine Trennung möglich ist. Diese Trennung ist aber nicht auf wirklich unterschiedliche Daten, sondern lediglich auf eine unterschiedliche Häufigkeit von bestimmten Unterklassen der Daten zurückzuführen. Dies wird durch die Maxima deutlich, die in beiden Häufigkeitsverteilungen an derselben Stelle, aber mit unterschiedlicher Höhe auftreten. Das Neuronale Netz optimiert den globalen Fehler, indem es Maxima aus der ersten Klasse, die niedriger als die aus der zweiten Klasse sind, näher bei Null plaziert. Entsprechend kommen Maxima aus der zweiten Klasse, die kleiner als die aus der ersten Klasse sind, näher bei Eins zu liegen. Die Effizienzen zur "Trennung" der untersuchten realen Daten von ihren entsprechenden Monte-Carlo-Daten liegen zwischen $70 - 80\%$. Nun liegt aber ohne Verwendung eines Neuronalen Netzes die Effizienz, Elemente aus einer der beiden Klassen richtig zu klassifizieren, bereits bei 50%; ferner ist ein Neuronales Netz, wie in Abschnitt 5.2 ausgeführt, prinzipiell in der Lage, einen Weg zu finden, um zwischen realen Daten und Monte-Carlo-Daten zu trennen. Somit stellen die

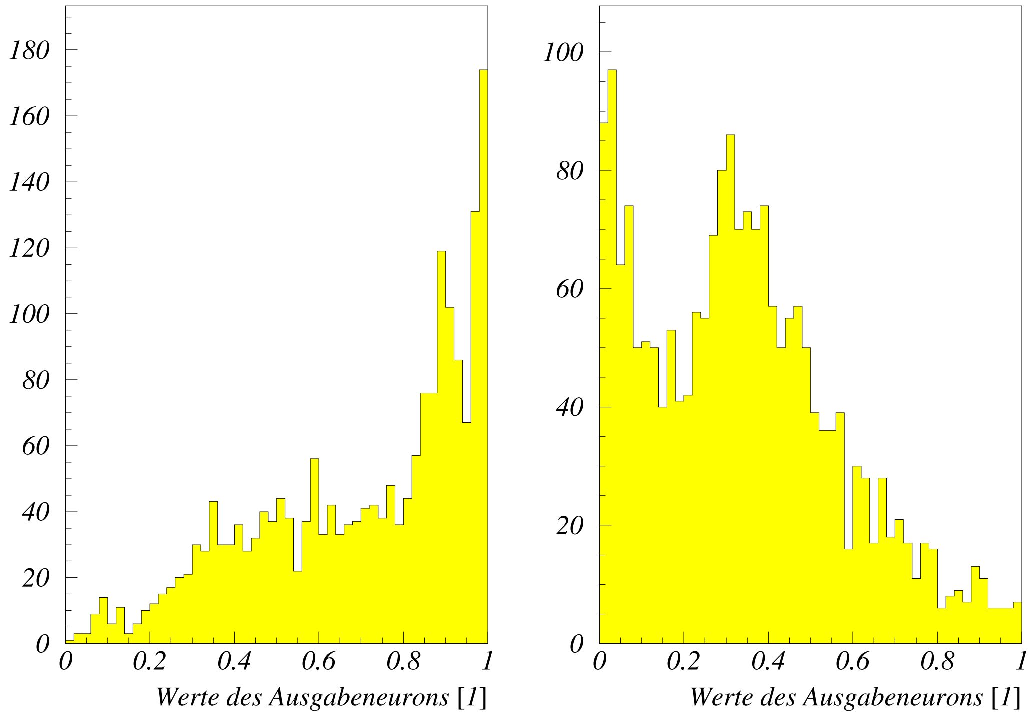

Abbildung 5.8: *Häufigkeitsverteilung des Ausgabeneurons des zur Trennung von realem Untergrund (links) und MC-Untergrundelementen (rechts) trainierten Netzes.* Der beste Leistungsfähigkeits-Wert des Netzes beträgt 78.8%.

erzeugten MC-Einzelhits und MC-Untergrundelemente eine sehr gute Simulation der realen einzelnen Hits bzw. des realen Untergrundes dar.

Netzwerke, die in der in diesem Abschnitt angegebenen Weise trainiert werden, könnten zur Überprüfung von Monte-Carlo-Generatoren im Experiment eingesetzt werden und wichtige Hinweise auf Unvollständigkeiten geben.

5.8 Die Erkennung von Doppelhits

In diesem Abschnitt wird die Erkennung von Doppelhits mittels Neuronaler Netzwerke behandelt und untersucht, wie die Effizienz dieser Erkennung
a) vom Abstand zwischen den Schwerpunkten
b) vom Verhältnis der Amplituden
der Hits abhängt, aus denen der Doppelhit besteht.

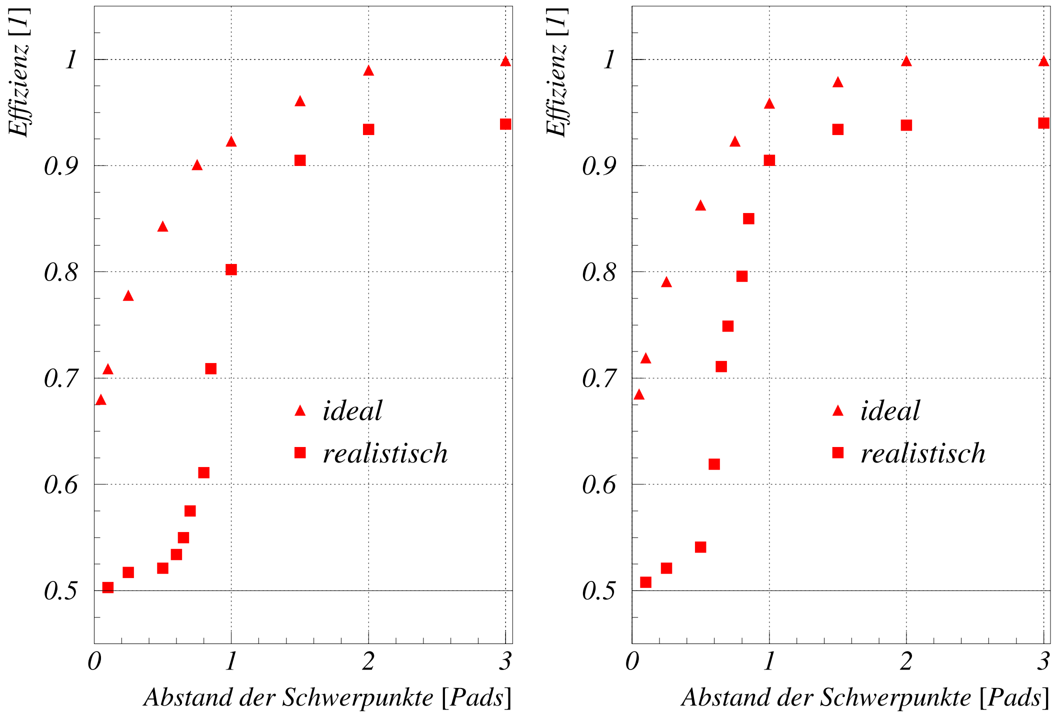

Abbildung 5.9: *Effizienz zur Erkennung von MC-Doppelhits mittels der entwickelten Netze in Abhängigkeit vom Abstand zwischen den Schwerpunkten der Hits, die den Doppelhit bilden; links für RICH 1, rechts für RICH 2.*

Die Klasse der MC-Doppelhits läßt sich in Unterklassen mit konstanten Abstand im Sinne von Punkt a) bzw. konstantem Amplitudenverhältnis im Sinne von Punkt b) gliedern. Zur Bestimmung der oben angeführten Abhängigkeiten wurden von jeder zu untersuchenden Unterklasse $4 \cdot 10^4$ Lernmuster nach Abschnitt 5.2.2 erzeugt und zusammen mit $4 \cdot 10^4$ MC-Einzelhits jeweils ein Lernsample gebildet. Bezüglich Punkt a) wurde noch zwischen RICH 1 und RICH 2 unterschieden. Mit jedem so erhaltenen Lernsample wurde gemäß der bisherigen Vorgehensweise ein Netzwerk trainiert und der für die Effizienz zur Erkennung der Doppelhits erhaltene Wert *bei maximaler Leistungsfähigkeit* des Netzes festgehalten. Dabei wurden die Doppelhits als Elemente der ersten Klasse und die Einzelhits als Elemente der zweiten Klasse im Sinne von Abschnitt 5.3.2 betrachtet.

Die Ergebnisse für Punkt a) sind in Abb. 5.9 dargestellt. Schlüsselt man die realistischen MC-Doppelhits in Doppelhits auf, bei denen *nur* Rauschen bzw. *nur* Defekte berücksichtigt werden, so stellt man fest, daß die Effizienz im wesentlichen durch

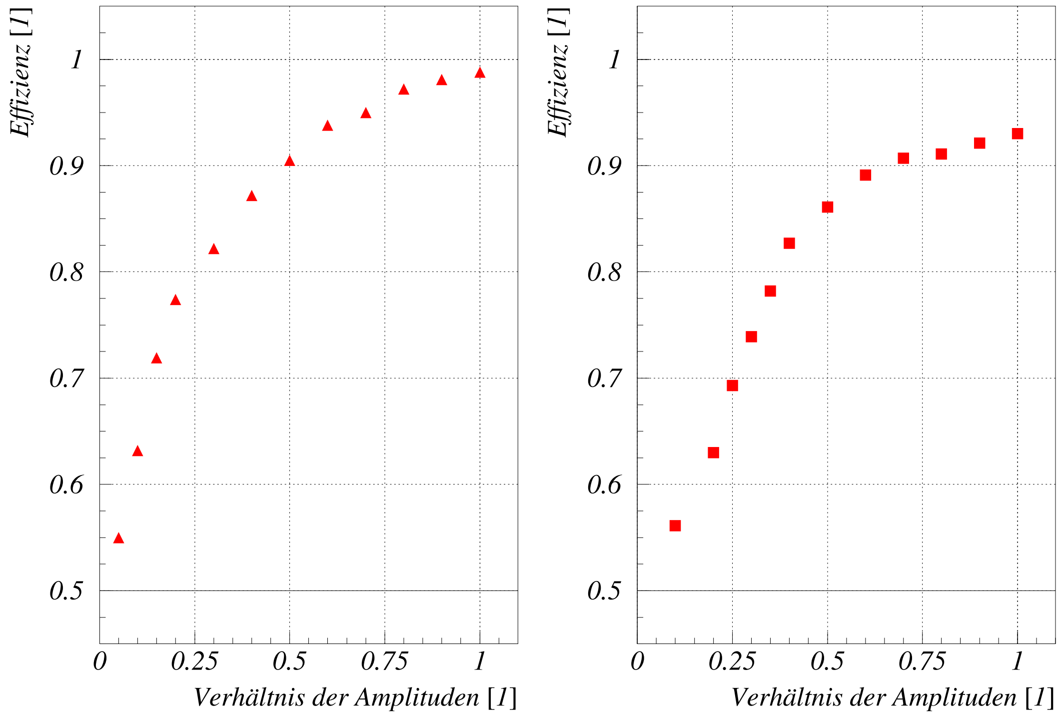

Abbildung 5.10: *Effizienz zur Erkennung von MC-Doppelhits mit Hilfe der entwickelten Neuronalen Netze in Abhängigkeit vom Verhältnis der Amplituden der einzelnen Hits, aus denen der Doppelhit besteht; links für ideale, rechts für realistische MC-Doppelhits.*

das Vorhandensein von Defekten verlorengeht: sind nur Defekte berücksichtigt, so liegt die Effizienz im Mittel ca. 11% unter den für ideale Doppelhits gemessenen Werten, während man durchschnittlich nur etwa 3% an Effizienz verliert, wenn nur Rauschen berücksichtigt wird.
Die Ergebnisse für Punkt b) sind aus Abb. 5.10 ersichtlich. Schlüsselt man wieder wie oben die realistischen MC-Doppelhits nach Rauschen und Defekten auf, so stellt man auch hier fest, daß die Effizienz im wesentlichen durch das Vorhandensein von Defekten verlorengeht (die Defekte kosten im Mittel ca. 10% an Effizienz, das Rauschen hingegen im Durchschnitt nur etwa 3%).

Da die Effizienz zur Trennung der Einzel- und Doppelhits ohne Verwendung eines Neuronalen Netzes bereits bei 50% liegt, ist es sinnvoll, erst ab einer Effizienz von 70% von einer Trennung der zu separierenden Klasse durch das jeweilige Netz zu sprechen. Mit diesem Trennkriterium geht aus den Abbildungen 5.9 und 5.10 hervor: ideale Doppelhits lassen sich durch Neuronale Netzwerke noch trennen, wenn der Abstand zwischen den Schwerpunkten größer als 0.1 Pads und das Verhältnis der Amplituden kleiner als 1/10 ist. Realistische Doppelhits sind durch Neuronale Netze noch zu trennen, wenn der Abstand der Schwerpunkte größer als 0.85 Pads in RICH 1 bzw. 0.65 Pads in RICH 2 und das Amplitudenverhältnis kleiner als 1/4 ist. Damit kann die dem bisherigen Hitrekonstruktionsalgorihmus immanente Grenze der Doppelhitauflösung um 90% für ideale Hits und um 15% bzw. 35% für realistische Hits in RICH 1 bzw. RICH 2 verbessert werden. Der schlechtere Wert in RICH 1 ist durch die größere Hitbreite zu erklären.

Kapitel 6

Anwendung Neuronaler Netze auf reale Daten

Die entwickelten Neuronalen Netzwerke werden in diesem Kapitel auf realen Daten getestet. Unter Verwendung dieser Netze wird dann ein neues Hitanalyseverfahren entwickelt und bei verschiedenen quantitativen Analysen der Hits (Polarwinkelverteilung der Hits, Verteilung der Hitamplituden, Anzahl der Hits als Funktion der Multiplizität, Anzahl der Hits pro Ring) eingesetzt. Als nächstes wird unter Verwendung der Netzwerke ein neues Analyseverfahren zur Rekonstruktion von Elektronringen entwickelt und mit dem bisherigen Verfahren verglichen. Schließlich wird der Einsatz Neuronaler Netzwerke zur Rekonstruktion der Hitpositionen untersucht.

6.1 Das Testen der entwickelten Netze auf realen Daten

Wie in Abschnitt 5.3.1 gezeigt wurde, ist die Entwicklung von Neuronalen Netzwerken zur Klassifizierung von

1. einzelnen Hits und Untergrund,
2. Einzel- und Doppelhits einerseits und Untergrund andererseits
3. Einzel- und Doppelhits,

ausreichend, da sich die Klassifizierung von einzelnen Hits, Doppelhits und Untergrund durch Hintereinanderausführung der 2. und 3. Klassifizierung ergibt.

Die zur Lösung dieser Klassifizierungsprobleme in Kap. 5 entwickelten Netzwerke wurden nun auf realen Daten getestet. Die hierzu benötigten Daten wurden aus Schwefel-Gold-Ereignissen dadurch gewonnen, daß sie in beiden RICH-Detektoren mit dem Auge überprüft und dann klassifiziert wurden; insgesamt 1000 Elemente von jeder Klasse. Da nach Abschnitt 5.4 die 25-30-1-Architektur für dreilagige Netze bzw. die 25-10-10-1-Architektur für vierlagige Netze optimal ist, wurden die Untersuchungen für beide Netzwerktopologien durchgeführt. Es sind wieder diejenigen Effizienz- und Rejektionswerte angegeben, *für die die zugehörige Leistungsfähigkeit des Netzes maximal ist.*

Trennung von einzelnen Hits und Untergrund:

Die erhaltenen Ergebnisse sind in folgender Tabelle zusammengefaßt:

	25-30-1-Topologie		25-10-10-1-Topologie	
	RICH 1	RICH 2	RICH 1	RICH 2
Effizienz	98%	98%	96%	95%
Rejektion	83%	84%	84%	84%

Die Untersuchung wurde noch für größere Netze, d.h. für Netze mit einer größeren Anzahl von verborgenen Neuronen, durchgeführt, was jedoch zu keiner Verbesserung der Klassifizierungsfähigkeit führte. Aus der obigen Tabelle ist ersichtlich, daß die 25-30-1-Architektur nach Abschnitt 5.4 erwartungsgemäß optimal bezüglich der Klassifizierung von einzelnen Hits und Untergrund ist. Aus diesem Grund wurden im weiteren Verlauf alle Anwendungen auf realen Daten mit einer 25-30-1-Topolgie durchgeführt.

Schlüsselt man die Untergrundrejektion nach der Rejektion bezüglich der Cluster bzw. bezüglich des elektronischen Rauschens auf, so ergeben sich folgende Werte:

	RICH 1	RICH 2
Rejektion bzgl. Cluster	81%	82%
Rejektion bzgl. Rauschen	95%	96%

Man erkennt, daß die Rejektion im wesentlichen bei der Klassifizierung der Cluster verlorengeht.

Trennung von Einzel- und Doppelhits:

Hier ergaben sich folgende Ergebnisse:

	RICH 1	RICH 2
Effizienz für Einzelhits	91%	92%
Effizienz für Doppelhits	94%	96%

Trennung von Einzel- und Doppelhits einerseits und Untergrund andererseits:

Die Ergebnisse sind in folgender Tabelle zusammengefaßt:

	RICH 1	RICH 2
Effizienz	97%	97%
Rejektion	83%	84%

Schlüsselt man die Untergrundrejektion wieder nach Cluster und elektronischem Rauschen auf, so stellt man auch hier fest, daß für den Verlust an Rejektion im wesentlichen die Cluster verantwortlich sind (Rejektion bzgl. Cluster ist gleich 81%, die Rejektion bzgl. Rauschen beträgt 94%).

Die obigen Untersuchungen wurden außerdem noch mit Daten für Einzel- und Doppelhits durchgeführt, die aus Ereignissen gewonnen wurden, in denen UV-Photonen von einer D_2-Lampe erzeugt worden waren. Diese Ereignisse enthalten wenig Untergrund und eignen sich somit besonders gut zum Studium der Trennbarkeit dieser Klassen. Die erhaltenen Ergebnisse sind um maximal 1% größer als die entsprechenden aus den obigen Untersuchungen. Dies spricht dafür, daß der Einfluß an Subjektivität bei der Gewinnung der realen Daten aus Schwefel-Gold-Ereignissen sehr klein war.

Als Folgerung ist also festzuhalten: es wurden Neuronale Netzwerke entwickelt, die in der Lage sind, über 90% der Einzel- und Doppelhits sowie gleichzeitig über 80% des Untergrundes richtig zu klassifizieren.

6.2 Die Häufigkeitsverteilungen des Ausgabeneurons

Die in Kap. 5 entwickelten und im letzten Abschnitt auf realen Daten getesteten Netzwerke zur Klassifizierung von einzelnen Hits, Doppelhits und Untergrund wurden nun in den Hitrekonstruktionsalgorithmus des bisherigen Analyseverfahrens implementiert. Um mehrfache Zählungen eines Hits auszuschließen, wurde ein Fallunterscheidungsalgorithmus entwickelt, der die Abstände des zu klassifizierenden Hits zu seinen beiden nächstliegenden Hits, die Verhältnisse der Amplituden dieser Hits sowie das durch die Netzwerke erhaltene Klassifizierungsergebnis für die beiden nächsten Nachbar-Hits berücksichtigt, und ebenfalls in den alten Hitrekonstruktionsalgorithmus implementiert.

Der so erhaltene **neue Hitrekonstruktionsalgorithmus** wurde auf 1000 Schwefel-Gold-Ereignisse angewendet, wobei auf die Säuberungsalgorithmen des bisherigen Analyseverfahrens verzichtet wurde, da die Rejektion des Untergrundes jetzt von den Neuronalen Netzen des Hitrekonstruktionsalgorithmus geleistet wird. Diese neue Vorgehensweise - Anwendung des neuen Algorithmus unter Verzicht auf die alten Säuberungsalgorithmen - wird im folgenden als **neues Hitanalyseverfahren** bezeichnet. Als **bisheriges Hitrekonstruktionsverfahren** bezeichnen wir die Hintereinanderausführung der alten Säuberungsalgorithmen und des alten Hitrekonstruktionsalgorithmus.
Als erstes wurden nur die Netze zur Klassifizierung von einzelnen Hits (1. Klasse) und Untergrund (2. Klasse) implementiert. Wie in Abschnitt 5.3.1 festgelegt, wird dem Ausgabeneuron eines Netzes während des Trainierens der Wert 1 zugeordnet, falls es sich um Lernmuster der ersten Klasse handelt, und der Wert 0, falls Lernmuster der zweiten Klasse vorliegen. Abb. 6.1 zeigt die erhaltene Häufigkeitsverteilung für das Ausgabeneuron in jedem der beiden RICHes. Als nächstes wurden alle zur Klassifizierung von einzelnen Hits, Doppelhits und Untergrund erforderlichen Netzwerke (siehe Abschnitt 6.1) implementiert. Die erhaltene Häufigkeitsverteilung für das Ausgabeneuron des Netzes zur Klassifizierung von Einzel- und Doppelhits einerseits (1. Klasse) sowie Untergrund andererseits (2. Klasse) in jedem der beiden RICH-Detektoren ist in Abb. 6.2 gezeigt; die erhaltene Häufigkeitsverteilung für das Ausgabeneuron des Netzes zur Klassifizierung von Einzel- (1. Klasse) und Doppelhits (2. Klasse) in jedem der beiden RICHes ist ebenfalls in Abb. 6.2 zu sehen.

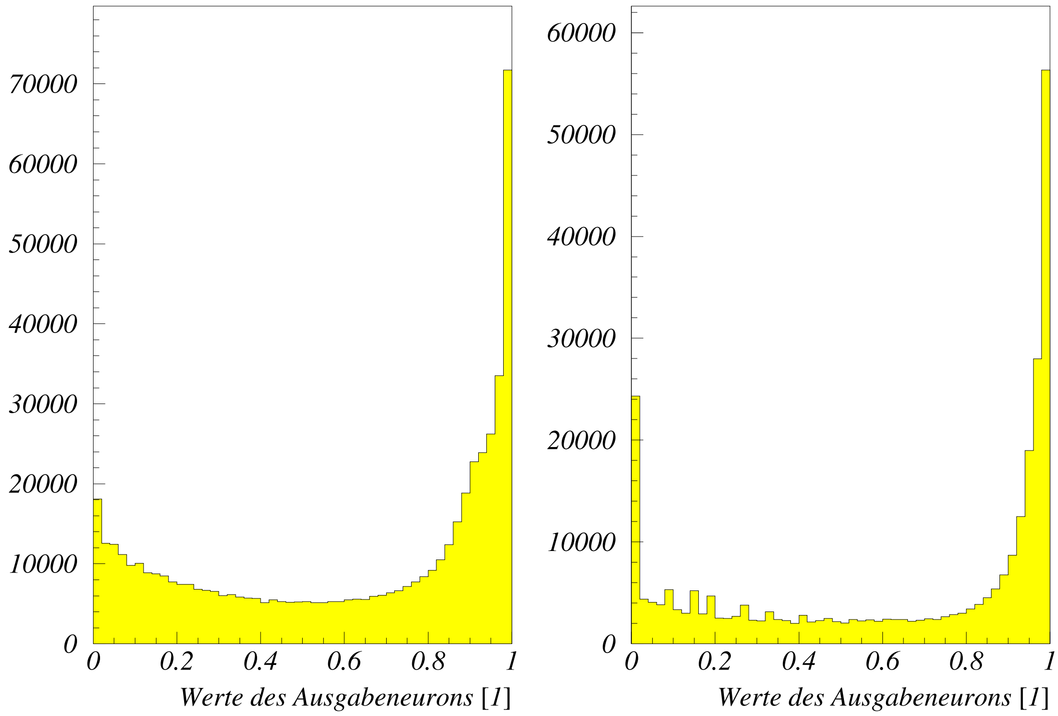

Abbildung 6.1: *Häufigkeitsverteilung des Ausgabeneurons des Netzwerkes zur Klassifizierung von einzelnen Hits (1. Klasse) und Untergrund (2. Klasse); links für RICH 1, rechts für RICH 2.*

Die dargestellten Schaubilder geben die gute Trennung der einzelnen Klassen durch die eingesetzten Neuronalen Netzwerke wieder.

6.3 Die Verteilung des Polarwinkels der Hits

Bestimmt man die Verteilung des Polarwinkels der Hits relativ zur Strahlachse nach dem bisherigen Hitrekonstruktionsverfahren, so stimmt die in RICH 2 gemessene Verteilung mit der aus Monte-Carlo-Simulationen erwarteten überein, in RICH 1 weist die gemessene Verteilung (Abb. 6.3) jedoch gegenüber der MC-Simulation einen deutlichen Überschuß bei kleinen Polarwinkeln auf.

Wird nun das neue Hitanalyseverfahren angewendet, so ergeben sich nach Einzelhits, Doppelhits und Untergrund aufgeschlüsselt die in Abb. 6.3 gezeigten Verteilungen. Hierbei ist unter Untergrund derjenige Teil des Untergrundes zu verstehen,

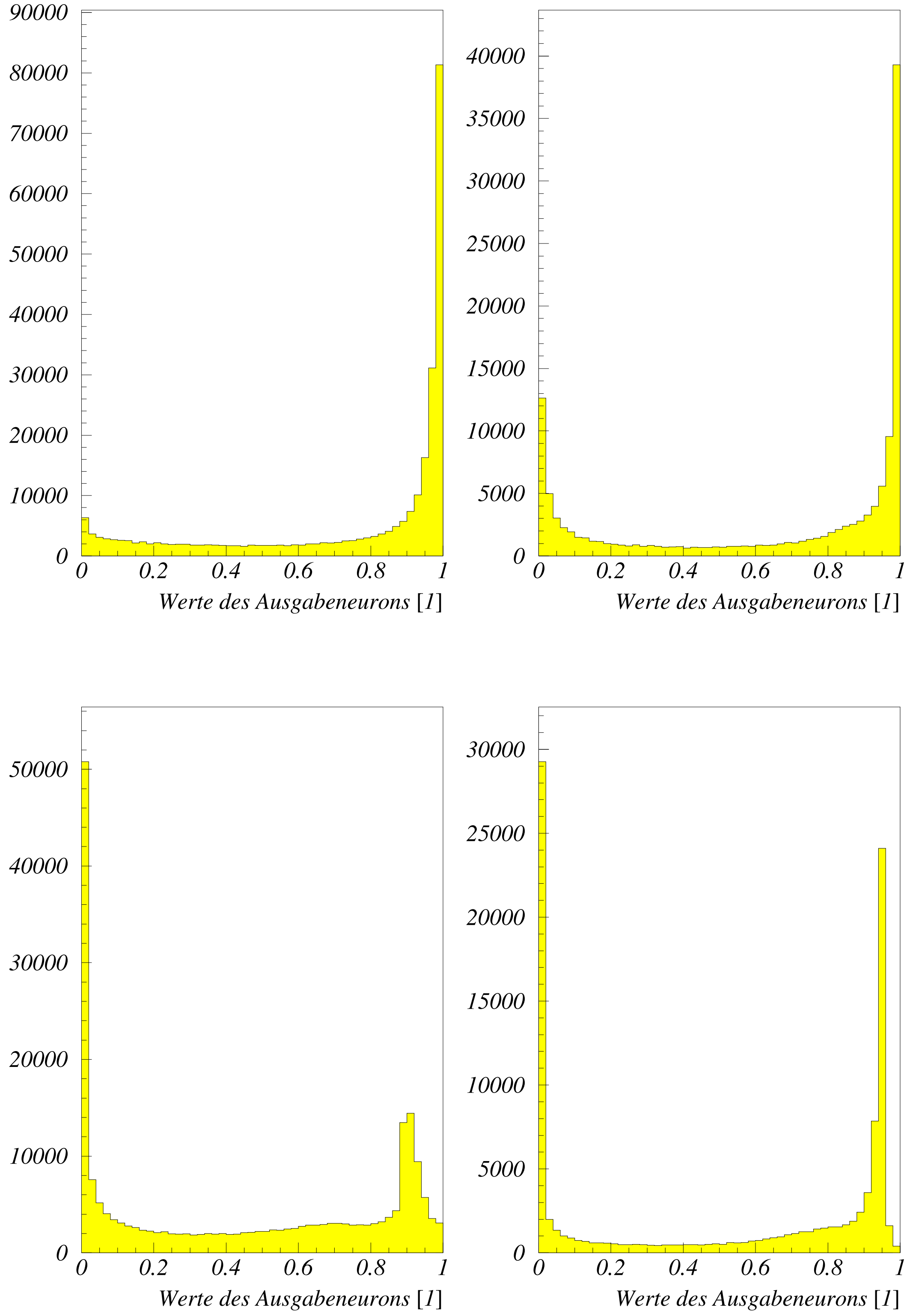

Abbildung 6.2: *Oben: Häufigkeitsverteilung des Ausgabeneurons des Netzwerkes zur Klassifizierung von Einzel- und Doppelhits einerseits (1. Klasse) und Untergrund andererseits (2. Klasse); links für RICH 1, rechts für RICH 2.*
Unten: Häufigkeitsverteilung des Ausgabeneurons des Netzwerkes zur Klassifizierung von Einzel- (1. Klasse) und Doppelhits (2. Klasse); links für RICH 1, rechts für RICH 2.

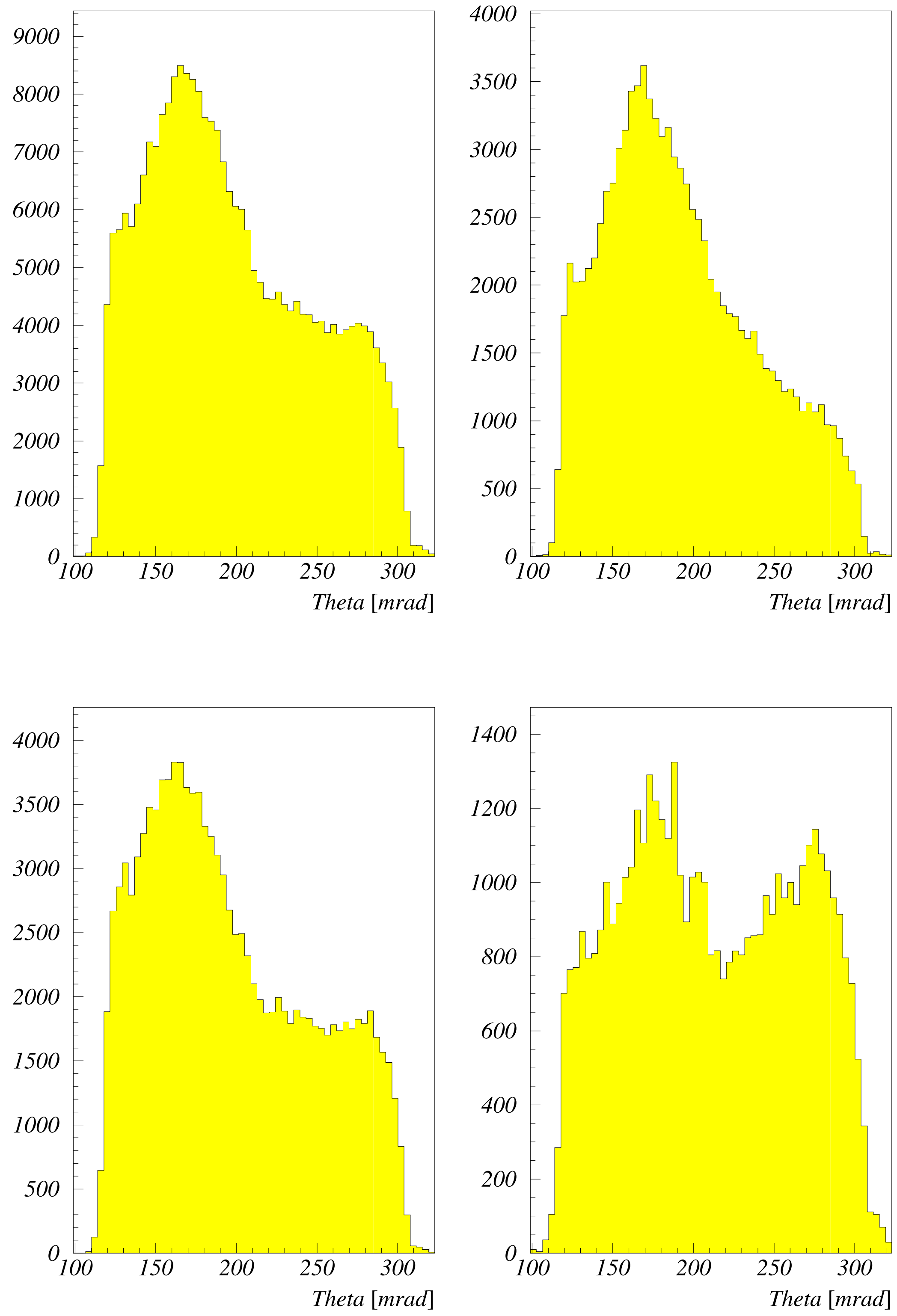

Abbildung 6.3: *Verteilung des Polarwinkels der Hits relativ zur Strahlachse in RICH 1: nach dem bisherigen Hitrekonstruktionsverfahren (oben links) und nach dem neuen Hitanalyseverfahren; oben rechts für Einzelhits, unten links für Doppelhits und unten rechts für Untergrund.*

der vom bisherigen Hitrekonstruktionsverfahren als Hits rekonstruiert, aber von den Neuronalen Netzen als Untergrund klassifiziert wird (**„falsche" Hits**).

Man erkennt, daß der Überschuß bei kleinen Polarwinkeln von Einzel- und Doppelhits herrührt, also kein Effekt der schlechten Untergrundrejektion des bisherigen Verfahrens, sondern physikalischen Ursprungs ist, während die „falschen" Hits sowohl bei kleinen als auch bei großen Polarwinkeln einen etwa gleich großen Überschuß aufweisen. Es muß also einen physikalischen Effekt geben, der von der Monte-Carlo-Simulation nicht beschrieben wird. Die Herkunft dieses Effektes ist bis jetzt noch nicht verstanden.

6.4 Die Verteilung der Hitamplituden

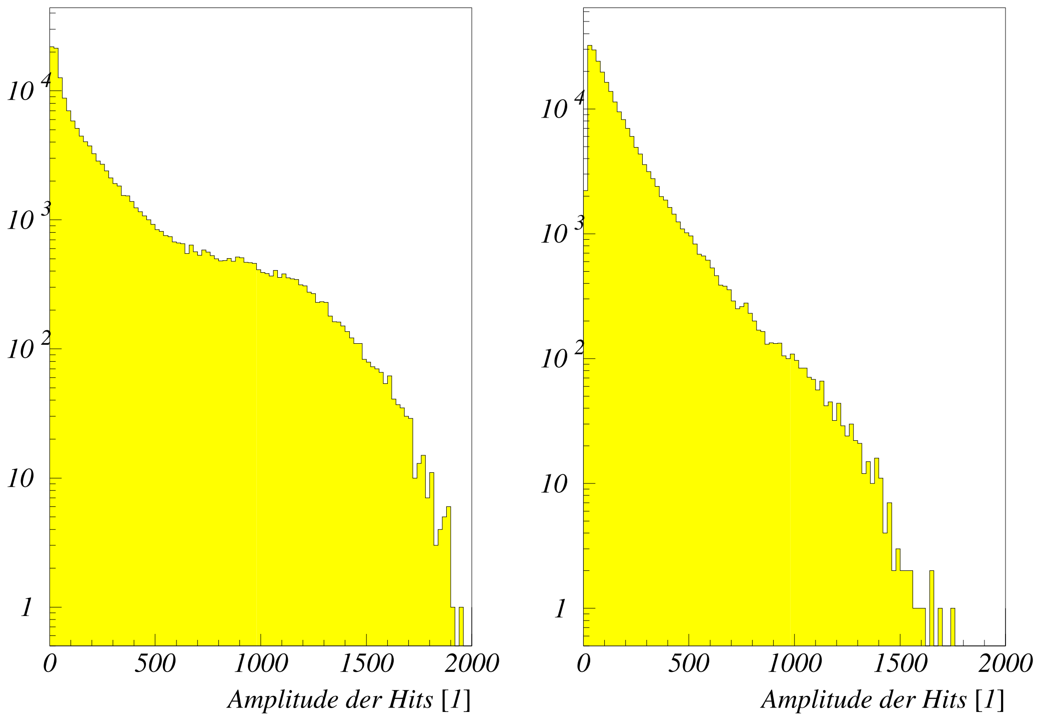

Abbildung 6.4: *Verteilung der Hitamplituden in RICH 2: links nach dem bisherigen, rechts nach dem neuen Hitanalyseverfahren.* Die nach dem neuen Verfahren erhaltene Verteilung stimmt mit dem theoretisch erwarteten exponentiell abfallenden Verlauf sehr viel besser überein als die nach dem bisherigen Verfahren erhaltene.

Für die Hitamplituden erwartet man theoretisch ein exponentiell abfallendes Spektrum, dessen Steigung ein Maß für die Verstärkung im UV-Detektor ist. Die für RICH 1 nach dem bisherigen Verfahren gemessene Verteilung folgt in guter Näherung einem exponentiellen Verlauf; in RICH 2 weist die gemessene Verteilung hingegen einen deutlichen Überschuß bei kleinen Hitamplituden sowie einen leichten Überschuß bei großen Hitamplituden auf, wie aus Abb. 6.4 hervorgeht.

Bei Anwendung des neuen Hitanalyseverfahrens ergeben sich nach Einzelhits, Doppelhits und Untergrund aufgeschlüsselt die in Abb. 6.5 dargestellten Verteilungen. Dabei ist unter Untergrund das Gleiche wie im vorangehenden Abschnitt zu verstehen. Aus Abb. 6.5 geht hervor, daß der Überschuß bei kleinen Hitamplituden durch „falsche" Hits, d.h. durch vom bisherigen Analyseverfahren fälschlicherweise als Hits rekonstruierten Untergrund verursacht wird. Der Überschuß bei großen Hitamplituden wird durch Doppelhits und „falsche" Hits hervorgerufen. Die Verteilung der Einzelhits zeigt in guter Näherung einen exponentiell abfallenden Verlauf.
Abb. 6.4 zeigt die Verteilung der Hitamplituden für Einzel- und Doppelhits. Man erkennt, daß diese Verteilung mit dem theoretisch erwarteten exponentiell abfallenden Verlauf sehr viel besser übereinstimmt als das nach dem bisherigen Verfahren erhaltene Spektrum. Die starke Abweichung des nach dem bisherigen Verfahren erhaltenen Spektrums beruht als im wesentlichen auf der schlechten Untergrundrejektion.

6.5 Die Anzahl der Hits als Funktion der Multiplizität

Bestimmt man die Anzahl der Hits in den beiden RICH-Detektoren in Abhängigkeit von der Multiplizität der Teilchen, so stellt man fest, daß beide Größen proportional zueinander sind, wie man dies auch erwartet. In RICH 2 erhält man einen sehr kleinen Offset; der Offset in RICH 1 ist größer - bei der in Abb. 6.6 gezeigten Messung für S-Au-Daten betrug er 75 Hits. Führt man diese Untersuchung mit Monte-Carlo-simulierten Daten durch, so stellt man fest, daß der Offset und die Geradensteigung, die man aus der Simulation erhält, mit den Ergebnissen aus der Messung mit realen Daten in RICH 2 übereinstimmen. In RICH 1 ergibt sich jedoch ein sehr kleiner Offset, außerdem ist hier die Geradensteigung für reale Daten um einen Faktor von ca. 1.7 größer als für MC-Daten. Diese Sachverhalte gehen aus Abb. 6.6 hervor.

Um eine Aufschlüsselung des Offsets und der Geradensteigung nach Einzelhits, Doppelhits und Untergrund zu erhalten, wurde die obige Untersuchung sowohl mit realen Daten als auch mit MC-Daten unter Einsatz des neuen Hitrekonstrukti-

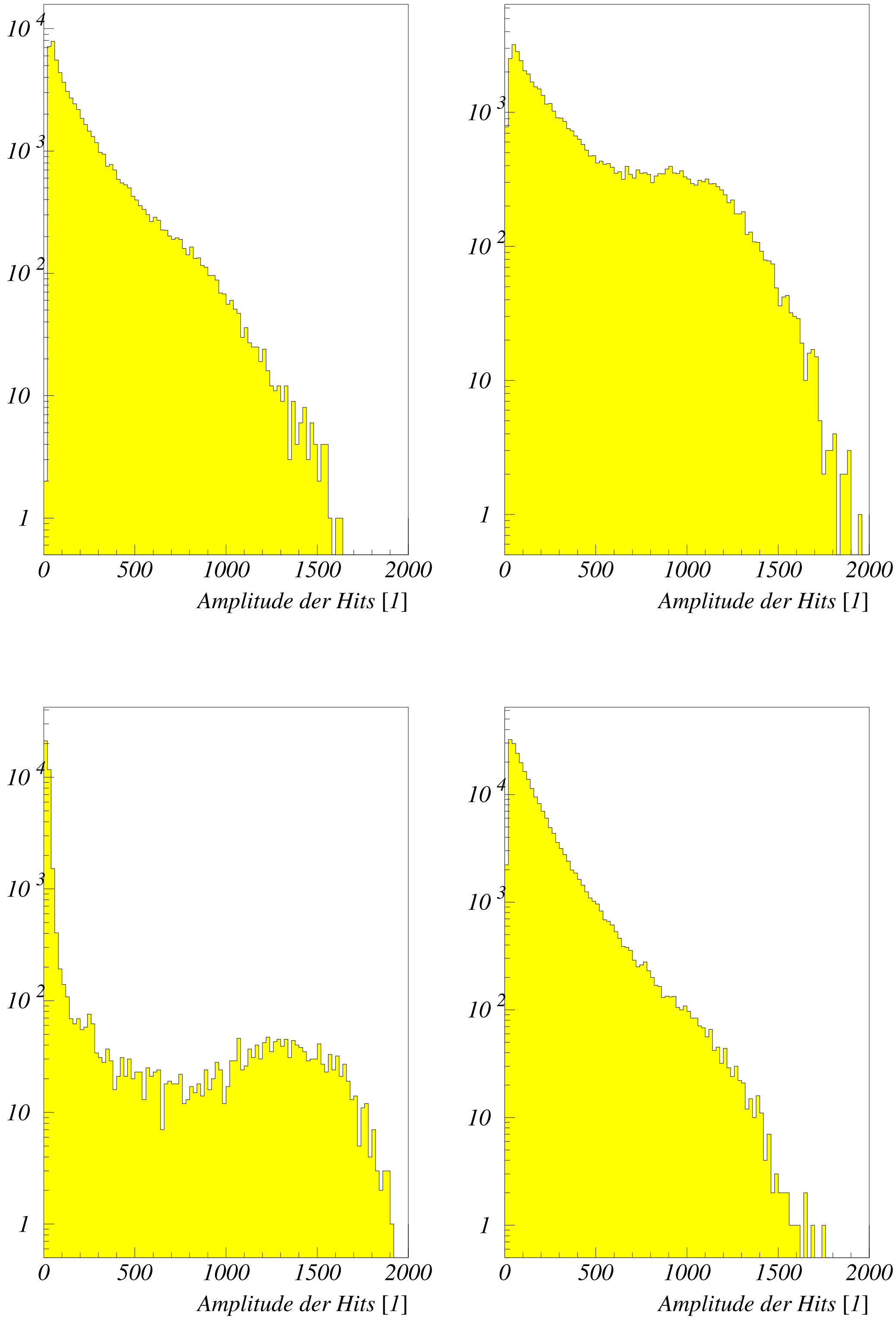

Abbildung 6.5: *Verteilung der Hitamplitude in RICH 2 nach dem neuen Hitanalyseverfahren; aufgeschlüsselt nach Einzelhits (oben links), Doppelhits (oben rechts) und Untergrund (unten links). Unten rechts ist noch die Verteilung für Einzel- und Doppelhits zusammen gezeigt.*

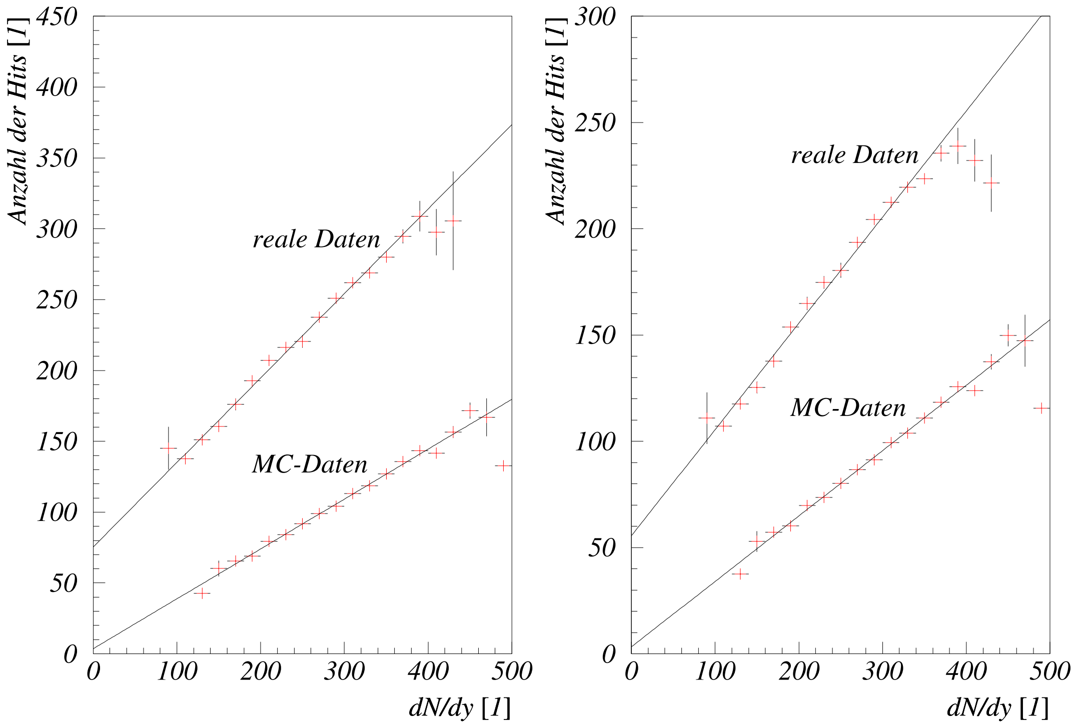

Abbildung 6.6: *Anzahl der Hits als Funktion der Multiplizität in RICH 1: links nach dem bisherigen, rechts nach dem neuen Hitanalyseverfahren.* Der nach dem neuen Verfahren erhaltene Offset für reale Daten ist deutlich kleiner als der nach dem bisherigen Verfahren erhaltene.

onsalgorithmus durchgeführt. Die Ergebnisse sind in Abb. 6.7 gezeigt. Aus der Abbildung geht hervor, daß der Offset zu ungefähr gleichen Teilen von Einzelhits, Doppelhits und „falschen" Hits hervorgerufen wird; bei Einsatz des neuen Verfahrens verkleinert sich also der Offset ungefähr um 1/3. Vergrößert man den Bewertungsschnitt des Netzwerkes zur Trennung der Einzel- und Doppelhits vom Untergrund derart, daß ca. 95% der Hitkandidaten als Untergrund klassifiziert werden - dies ist bei einem Schnitt von ca. 0.8 der Fall; die Effizienz beträgt dort ca. 65% -, so erhält man: ca. 61% des Offsets werden durch „falsche" Hits, ca. 28% durch Einzelhits und ca. 11% durch Doppelhits verursacht. Auch bei einer weiteren Vergrößerung des Schnittes auf 0.9 - die Rejektion beträgt hier ca. 98%, die Effizienz hat ungefähr den Wert 55% - bleibt nach Abzug der „falschen" Hits ein restlicher Offset von ca. 22 Hits, der zu ca. 73% von Einzelhits verursacht wird.
Aus Abb. 6.7 geht weiter hervor, daß sich die Anteile an der Geradensteigung wie folgt aufspalten: ca. 40% Einzelhits, ca. 51% Doppelhits und ca. 9% Untergrund.

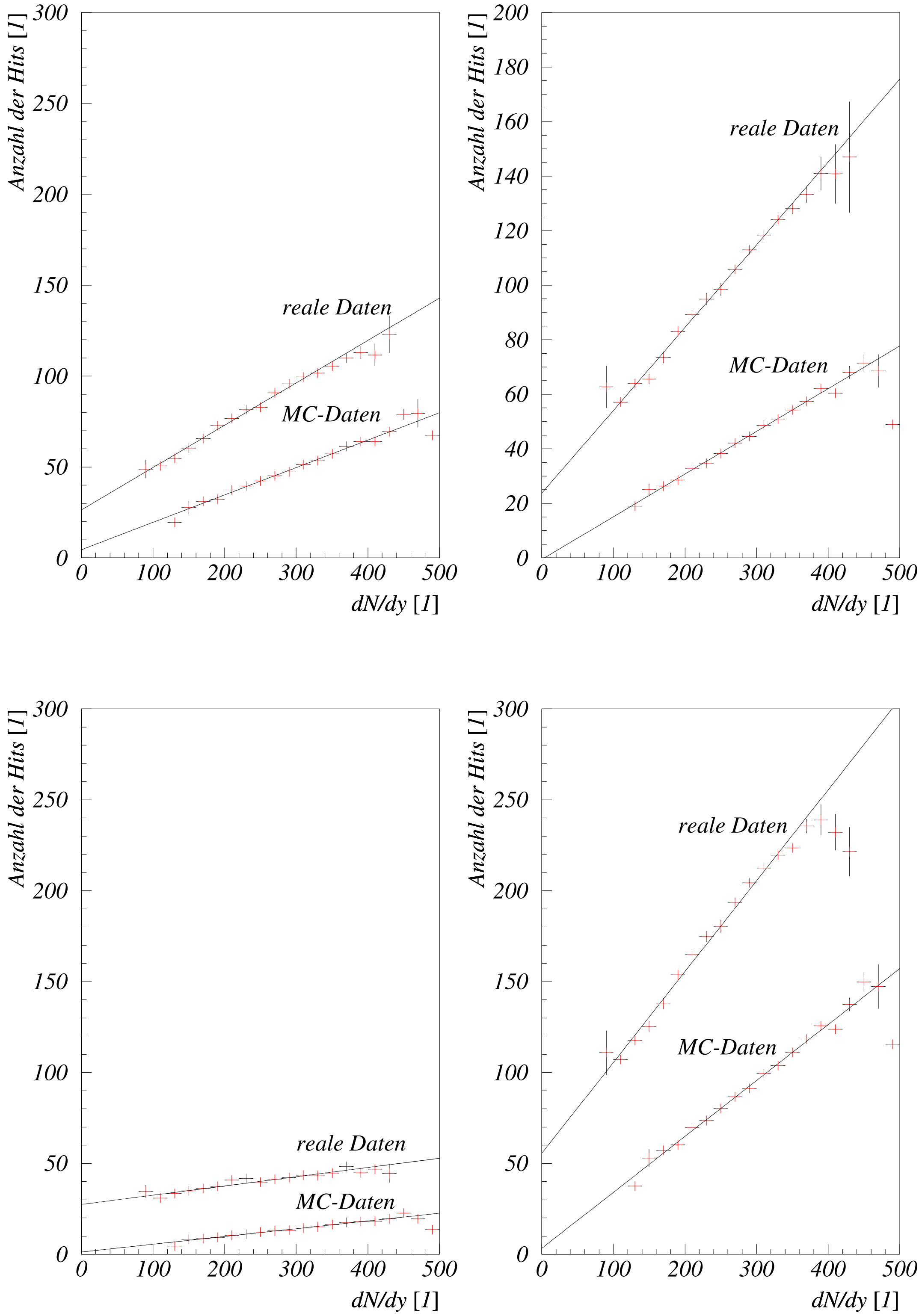

Abbildung 6.7: *Anzahl der Hits als Funktion der Multiplizität in RICH 1 nach dem neuen Hitanalyseverfahren sowohl mit realen Daten als auch mit MC-Daten; oben links für Einzelhits, oben rechts für Doppelhits, unten links für Untergrund, rechts unten für die Summe von Einzel- und Doppelhits.* Die Schaubilder erlauben eine Aufschlüsselung des Offsets und der Geradensteigung nach den einzelnen Klassen.

Das bedeutet, daß die Effizienz, einen Hit zu finden, nach dem neuen Verfahren um einen Faktor von ca. 0.92 kleiner als nach dem bisherigen Verfahren ist.
Vergleicht man die erhaltenen Geradensteigungen für reale Daten mit den entsprechenden für MC-Daten, so ergibt sich: für Einzelhits ist die Geradensteigung für reale Daten um einen Faktor von ungefähr 1.5 größer als für MC-Daten, für Doppelhits ist die Geradensteigung um einen Faktor von ca. 1.9 größer, für Einzel- und Doppelhits zusammen beträgt der Faktor ca. 1.6.

Aus den durchgeführten Untersuchungen folgt: der für reale Daten im Gegesatz zu MC-Daten gemessene Offset wird zu ca. 1/3 durch Untergrund verursacht, der vom bisherigen Hitrekonstruktionsalgorithmus fälschlicherweise als „Hits" erkannt wird; dieser Teil des Offsets ist also nicht physikalischer Natur und wird aufgrund der schlechten Untergrundrejektion des bisherigen Verfahrens gemessen. Aus der erhaltenen Zusammensetzung des anderen Teils des Offsets (zu gleichen Teilen bestehend aus Einzel- und Doppelhits) folgt, daß dieser Teil physikalischen Ursprungs ist. Dies impliziert die Existenz eines physikalischen Effektes in RICH 1, der durch die Monte-Carlo-Simulation nicht beschrieben wird. Durch diesen physikalischen Prozeß werden Hits erzeugt, von denen ca. 45% nahezu ideale oder ideale Hits sind, d.h. der Wert des Ausgabeneurons für diese Hits ist praktisch 1. Von diesen (nahezu) idealen Hits sind ca. 73% Einzel- und 27% Doppelhits. Die um einen Faktor von ca. 1.7 größere Geradensteigung für reale Daten im Vergleich zu MC-Daten wird im wesentlichen (ca. 91%) von Einzel- (ca. 40%) und Doppelhits (ca. 51%) (also „echten" Hits) hervorgerufen. Diese Tatsache bekräftigt die Existenz des postulierten physikalischen Effektes in RICH 1. Der Ursprung dieses Prozesses ist bis jetzt noch nicht verstanden.

6.6 Die Anzahl der Hits pro Ring

Das neue Hitanalyseverfahren wurde nun eingesetzt, um die Anzahl der Hits pro Ring in den beiden RICH-Detektoren zu bestimmen; die erhaltenen Verteilungen wurden mit denen durch das bisherige Analyseverfahren gemessenen verglichen.

Dabei wurden mit Monte-Carlo-Techniken 10000 einzelne und überlappende Elektronringe in RICH 1 bzw. RICH 2 simuliert und S-Au-Daten überlagert. Zur Rekonstruktion der Elektronringe wurde das bisherige Analyseverfahren des CERES-Experimentes[1] angewendet, wobei nur Ringe mit mehr als fünf Hits zugelassen wurden. Die erhaltene Anzahl der Hits für jeden rekonstruierten MC-Ring wurde in ein Histogramm gefüllt. Als nächstes wurden die zu jedem rekonstruierten MC-Ring gehörenden Hits nach dem neuen Hitanalyseverfahren analysiert. Je nach Klassifizierung durch die im neuen Hitrekonstruktionsalgorithmus implementier-

[1] Zur Erklärung des bisherigen Analyseverfahrens siehe Abschnitt 1.2.

ten Netzwerke und durch den Fallunterscheidungsalgorithmus wurde der Hit nicht, einfach bzw. doppelt gezählt und die so erhaltene Anzahl der Hits ebenfalls in ein Histogramm gefüllt. Die ursprünglichen, generierten Verteilungen für die re-

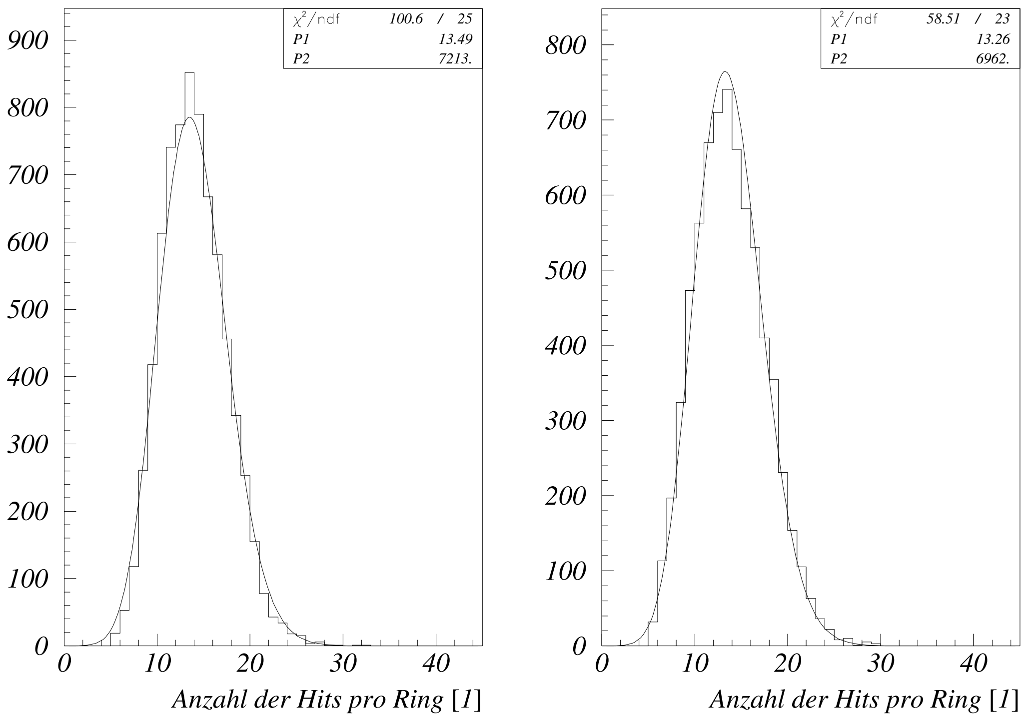

Abbildung 6.8: *Verteilung der Anzahl der Hits pro Ring für die generierten Monte-Carlo-Elektronringe; links für RICH 1, rechts für RICH 2.*

konstruierten MC-Elektronringe sind in Abb. 6.8 gezeigt; die nach dem bisherigen Analyseverfahren erhaltenen Verteilungen und die mittels der Neuronalen Netzwerke in der oben beschriebenen Vorgehensweise erhaltenen Verteilungen sind in Abb. 6.9 dargestellt. Da man theoretisch erwartet, daß die Anzahl der Hits pro Ring Poisson-verteilt ist, ist allen gezeigten Verteilungen eine Poisson-Verteilung angepaßt.

Man erkennt, daß die mit Hilfe der Neuronalen Netze gemessenen Verteilungen von den jeweils angepaßten Poissonverteilungen insofern abweichen, als für mittlere Hitzahlen zu wenig und für große Hitzahlen zu viele Hits gefunden werden. Dies liegt daran, daß auf Ringen mit vielen Hits im Gegensatz zu Ringen mit weniger Hits die Überlappungen komplexerer Natur sind, d.h. es tritt öfter der Fall ein, daß an den Überlappungen mehr als zwei Hits beteiligt sind. In diesem Fall

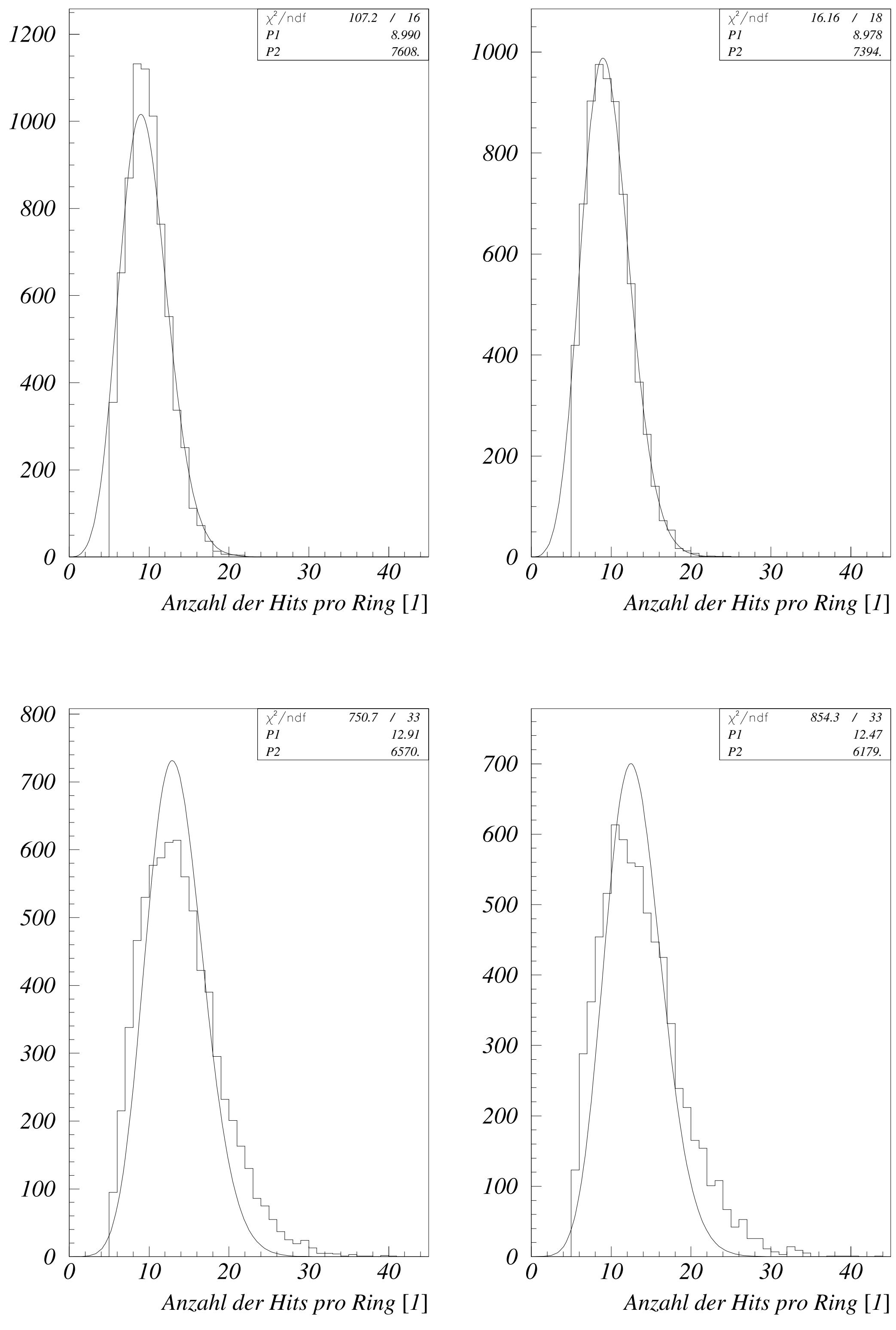

Abbildung 6.9: *Verteilung der Anzahl der Hits pro Ring nach dem bisherigen Verfahren (oben) und nach dem im Text beschriebenen Verfahren unter Verwendung der entwickelten Neuronalen Netzwerke (unten); links jeweils für RICH 1, rechts für RICH 2.*

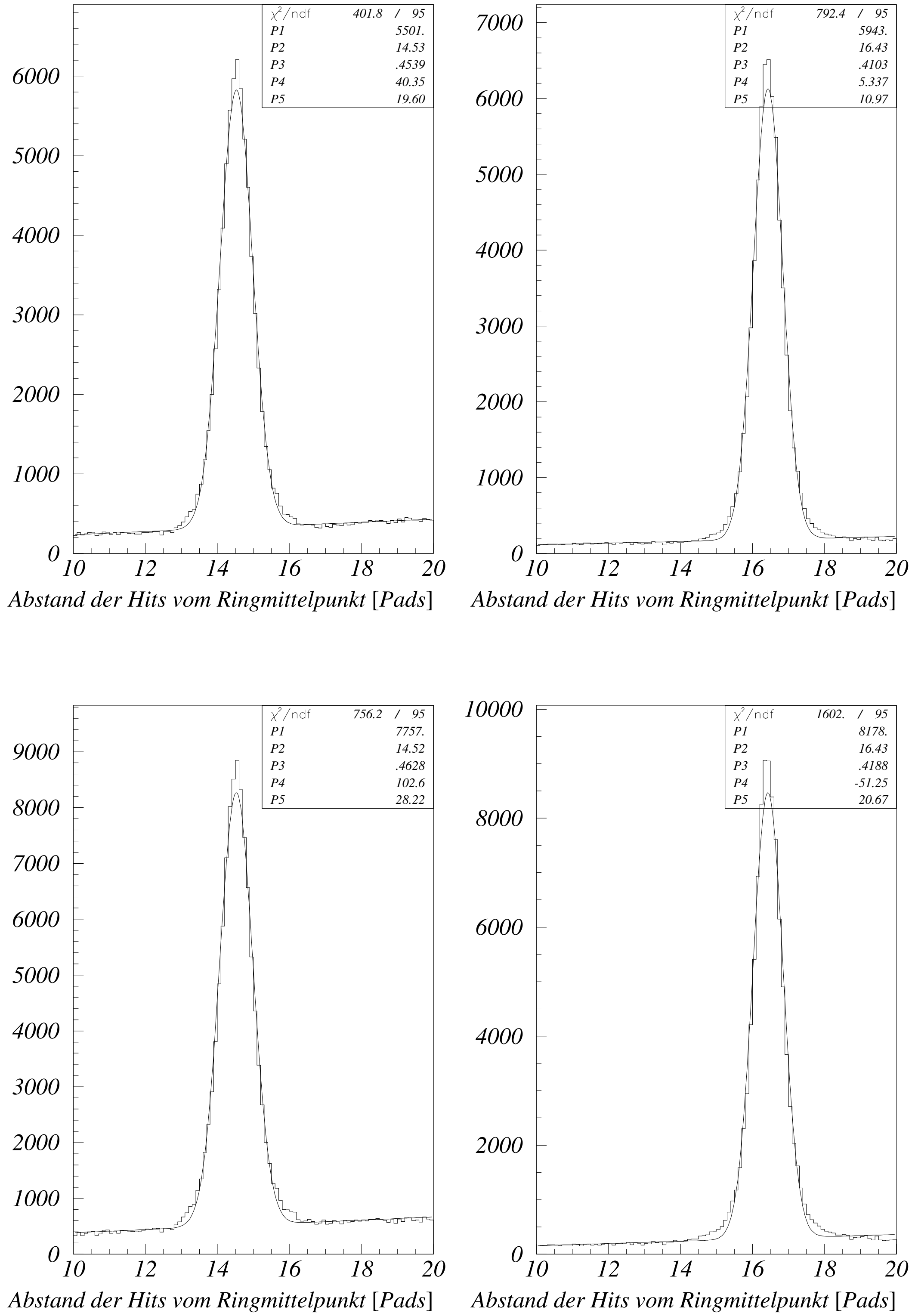

Abbildung 6.10: *Streuung aller Hits um die rekonstruierten Ringmittelpunkte nach dem bisherigen Analyseverfahren (oben) und unter Verwendung der entwickelten Neuronalen Netzwerke (unten); links jeweils für RICH 1, rechts für RICH 2.*

kommt es vor, daß die Netzwerke Hits mehrfach zählen.
Man erkennt weiter, daß die Mittelwerte der mit Hilfe der Netze gemessenen Verteilungen größer sind als die der nach der bisherigen Analyse ermittelten; und zwar um ca. 40% in beiden RICH-Detektoren. Diese Mittelwerte stimmen mit denen der Verteilungen für die generierten MC-Ringe fast überein.

Um die Genauigkeit der Hitrekonstrukion nach beiden Verfahren miteinander zu vergleichen, wurden bei der obigen Untersuchung zusätzlich die Abstände aller Hits eines Ereignisses zu den Mittelpunkten der rekonstruierten Monte-Carlo-Elektronringe in ein Histogramm gefüllt; dies wurde zum einen für alle vom bisherigen Analyseverfahren rekonstruierten Hits und zum anderen für alle mittels der Netzwerke analysierten Hits durchgeführt. Das RMS der so erhaltenen Streuung aller Hits um den Ringradius ist ein Maß für die Genauigkeit der Hitrekonstruktion. Die gemessenen Verteilungen sind in Abb. 6.10 gezeigt. An die Verteilungen ist jeweils die Überlagerung einer Gaußfunktion (mit Standardabweichung σ) und eines Polynoms ersten Grades angepaßt, weil eine solche Verteilung erwartet wird. Der Fläche des Trapezes unter der Gaußverteilung im Intervall [-3σ,+3σ] um den Ringmittelpunkt ist ein Maß für die Anzahl der Untergrundhits pro Ring.

Wie aus den Fitparametern zu erkennen ist, stimmen die Sigmas der jeweiligen Gaußverteilungen überein. Es zeigt sich, daß auch die Flächen der Trapeze unter den Gaußverteilungen jeweils übereinstimmen. Die Rekonstruktionsgenauigkeit der Hitposition und die Anzahl der Untergrundhits pro Ring ist also für beide Verfahren gleich.

Als Folgerungen ergeben sich:

1. Da man bei Verwendung Neuronaler Netzwerke einen größeren Mittelwert für die Anzahl der Hits pro Ring mißt, erwartet man eine bessere Unterdrückung von falschen Ringen bei der Suche nach Ringkandidaten.

2. Da Einzel- und Doppelhits klassifiziert werden können, ist eine genauere Rekonstruktion der Hits möglich. Zur Bestimmung der Position von Einzelhits ist die Schwerpunktsbestimmung ausreichend, wie sie im bisherigen Hitrekonstruktionsverfahren durchgeführt wird; um die Zentren der (Einzel-)Hits zu bestimmen, die einen Doppelhit bilden, ist die Anpassung einer Doppel-Gaußverteilung erforderlich. Da die Rekonstruktionsgenauigkeit $\sigma_{Ringmittelpunkt}$ des Ringmittelpunktes von der Rekonstruktionsgenauigkeit $\sigma_{Hitposition}$ der Hitposition in der folgenden Weise abhängt:

$$\sigma_{Ringmittelpunkt} = \frac{\pi}{2} \frac{\sigma_{Hitposition}}{\sqrt{N-2}} ,$$

wobei N für die Anzahl der Hits pro Ring steht, ist dann auch eine genauere Rekonstruktion des Ringmittelpunktes möglich.

6.7 Anwendung auf die Rekonstruktion der Elektronringe

Beim bisherigen Analyseverfahren zur Rekonstruktion der Elektronringe wird zur Identifizierung der Ringkandidaten eine **Hough-Transformation** mit den Positionen aller Pads durchgeführt, die nach Anwendung der Säuberungsalgorithmen noch verbleiben (d.h. die dann noch eine Amplitude größer Null besitzen). Man spricht deshalb von einer Hough-Transformation **auf Padniveau**. Bei der Hough-Transformation handelt es sich im wesentlichen um eine Transformation aus dem Datenraum in den Parameterraum desjenigen Modells, mit dem die Daten beschrieben werden sollen. Zur Erklärung der Hough-Transformation siehe *Hough* [36], der Einsatz der Hough-Transformation zur Detektion von parametrisierbaren Kurven ist bei *Duda et al.* [37] und *Kimme et al.* [38] beschrieben.

Unter Verwendung des neuen Hitanalyseverfahrens wurde nun ein **neues Analyseverfahren zur Rekonstruktion der Elektronringe** entwickelt. Dieses neue Verfahren gliedert sich wie folgt:

1. Anwendung des neuen Hitanalyseverfahrens.
   Es wird also der neue Hitrekonstruktionsalgorithmus unter Verzicht auf die alten Säuberungsalgorithmen angewendet.

2. Suche der Ringkandidaten **auf Hitniveau**, d.h. die Suche nach Ringkandidaten wird jetzt mit allen (einzelnen und überlappenden) Hits durchgeführt, die vom neuen Hitanalyseverfahren identifiziert worden sind. Hierzu wurden die Parametereinstellungen des alten Algorithmus zur Kandidatensuche der neuen Vorgehensweise entsprechend angepaßt. Es kann wahlweise eine Hough-Transformation mit den Positionen der rekonstruierten Hits oder mit den Positionen aller zu den rekonstruierten Hits gehörenden Pads durchgeführt werden.

3. Anwendung eines Ringfites auf die für jeden Ringkandidaten nach dem neuen Hitanalyseverfahren rekonstruierten Hits.

4. Anwendung von Qualitätsschnitten auf die Ringe. Hier wie auch bei der Anwendung des Ringfites wurden die alten Algorithmen dem neuen Verfahren entsprechend angepaßt.

Das bisherige und das neue Analyseverfahren zur Rekonstruktion der Elektronringe wurden bezüglich der Effizienz, einen Elektronring wiederzufinden, (**Ringfindeeffizienz**) sowie bezüglich der Anzahl der fälschlicherweise gefundenen Ringe (**Fake-Ringe**) pro Ereignis miteinander verglichen.

Hierzu wurden 1000 durch Monte-Carlo-Techniken simulierte einzelne und überlappende Elektronringe S-Au-Daten in RICH 1 überlagert. Beim neuen Analyseverfahren wurde die Kandidatensuche sowohl mit den Positionen der rekonstruierten Hits als auch mit den Positionen aller zu den rekonstruierten Hits gehörenden Pads durchgeführt. Für jedes Verfahren wurde die Ringfindeeffizienz und die Anzahl der Fake-Ringe pro Ereignis sowohl nach der Kandidatensuche (auf **Kandidatenniveau**) als auch nach Durchführung des Ringfites (auf **Fitniveau**) bestimmt. Die erhaltenen Ergebnisse sind in Tab. 6.1 zusammengefaßt. Aus der Tabelle geht hervor, daß das neue Analyseverfahren mit den Positionen der Hits wegen der großen Anzahl der Fake-Ringe pro Ereignis schlechter als das bisherige Analyseverfahren ist; das neue Analyseverfahren mit den Positionen aller zu den Hits gehörenden Pads ist jedoch gleich gut wie die bisherige Analyse.

Als Folgerungen sind festzuhalten:

1. Es wurde ein Analyseverfahren zur Rekonstruktion der Elektronringe entwickelt, daß bezüglich der Ringfindeeffizienz und der Anzahl der Fake-Ringe pro Ereignis gleich gut wie das bisherige Analyseverfahren ist.

2. Da das neue Analyseverfahren ohne die Säuberungsalgorithmen der bisherigen Analyse auskommt, kann auf die alten Säuberungsalgorithmen verzichtet werden. Die entwickelten Netzwerke stellen also eine Alternative zu den alten Säuberungsalgorithmen dar. Somit ist außerdem gezeigt, daß die Aufeinanderfolge von sechs Algorithmen bei der bisherigen Analyse auf die Aufeinanderfolge von vier Algorithmen bei der neuen Analyse vereinfacht und gestrafft werden kann.

6.8 Die Rekonstruktion der Hitposition

Es wurde nun versucht, die Positionen der Hits mit Hilfe Neuronaler Netze zu rekonstruieren. Eine solches Verfahren zur Bestimmung der Hitposition ist insbesondere für Hits von Interesse, die Defekte aufweisen oder von Rauschen überlagert sind, da die bislang durchgeführte Schwerpunktsbildung zum Finden der Hitposition in diesen Fällen zwangsläufig nicht exakt ist. Das gleiche gilt für zwei oder mehr sich überlappende Hits: die Schwerpunktsbildung erlaubt keine korrekte Bestimmung der Positionen solcher Hits. Die Schwerpunktsbildung liefert außerdem einen systematischen Fehler, wie aus Abb. 6.12 hervorgeht.

Zur Beschreibung eines Hits wurde analog zu Abschnitt 5.2 seine 5×5-Matrix gewählt. Es war also ein Netzwerk zu entwickeln, daß die Abbildung der Inhalte der 5×5-Matrix eines Hits auf seinen Mittelpunkt implementiert. Diese Abbildung

Analyseverfahren	Ringfindeeffizienz auf Kandidatenniveau	Fake-Ringe pro Ereignis auf Kandidatenniveau	Ringfindeeffizienz auf Fitniveau	Fake-Ringe pro Ereignis auf Fitniveau
Bisheriges Analyseverfahren	77.9%	20	77.2%	10
Neues Analyseverfahren mit den Positionen der Hits	83.4%	33	75.0%	19
Neues Analyseverfahren mit den Positionen aller zu den Hits gehörenden Pads	80.7%	20	77.1%	9

Tabelle 6.1: *Ringfindeeffizienz und Anzahl der Fake-Ringe pro Ereignis für das bisherige und das neue Analyseverfahren.*

ist aufgrund der durch die Digitalisierung hervorgerufenen begrenzten Rekonstruktionsgenauigkeit zwangsläufig nicht exakt (siehe Abschnitt 1.2). Die Abbildung - im folgenden mit A bezeichnet - ist meßbar, da es eine konvergente Folge von Treppenfunktionen $(T_n)_{n=1,2,\ldots}$ gibt mit $A = \lim_{n\to\infty} T_n$. Also existiert nach Abschnitt 4.4 ein Multilayer-Feedforward-Netzwerk mit *einer* verborgenen Schicht und einer *glatten* Aktivierungsfunktion, mit dem A im Rahmen der begrenzten Rekonstruktionsgenauigkeit beliebig genau approximiert werden kann. Von dem Lernalgorithmus, der es dem Netz ermöglicht, die richtigen Gewichte für die erforderte Approximation aus der empirischen Beobachtung von Lernmustern zu gewinnen, ist nach Abschnitt 4.4 nur bekannt, daß ein solcher existiert. Außerdem ist nichts über die Anzahl der erforderlichen verborgenen Neuronen bekannt.

Zunächst wurde analog zu *Müller* und *Reinhardt* [39] eine Netzwerkarchitektur mit zwei Ausgabeneuronen gewählt, wobei das erste Ausgabeneuron die x-Koordinate und das zweite Ausgabeneuron die y-Koordinate der Position des Hits im zentralen Pad der 5×5-Matrix repräsentierte. Das Zentrum der 5×5-Matrix hat dabei die Koordinaten (0,0). Die während des Trainierens vorgegebenen gewünschten Output-Werte, also die Koordinaten, wurden mit einer Genauigkeit von 1/100 Pad angegeben. Die Anzahl der Eingabeneuronen betrug entsprechend der in Abschnitt 5.5 beschrieben Vorgehensweise 25; die Anzahl der verborgenen Schichten sowie die Anzahl der verborgenen Neuronen pro Schicht wurden nach dem in Abschnitt 5.4 erläuterten Optimierungsverfahren ermittelt. Es wurde ein Lernsample von $8 \cdot 10^4$ idealen MC-Einzelhits (siehe Abschnitt 5.2.1) erzeugt und obiges Netzwerk nach der in Abschnitt 5.3.1 erklärten Vorgehensweise trainiert. Der Test erfolgte auf einem Testsample von ebenfalls $8 \cdot 10^4$ idealen MC-Einzelhits. In Abbildung 6.11 ist die Abhängigkeit der rekonstruierten Positionen der Hits aus dem Testsample von ihren generierten Positionen gezeigt. Da die Abhängigkeit in x- und y-Richtung gleich ist, wird nur auf die Abhängigkeit in x-Richtung eingegangen. Abb. 6.12 zeigt zum Vergleich den analogen Plot zu Abb. 6.11 für die Bestimmung der Hitposition durch die Schwerpunktsbildung bzw. durch die Anpassung einer gaußförmigen Ladungsverteilung (Gauß-Fit-Methode). Zwar verursacht die Gauß-Fit-Methode - wie deutlich zu erkennen ist - im Vergleich zur Schwerpunktsbildung den kleineren Rekonstruktionsfehler; im bisherigen Analyseverfahren wird jedoch die Schwerpunktsbildung zur Bestimmung der Hitposition benutzt, da diese wesentlich weniger Rechenzeit benötigt und robuster bei Auftreten von elektronischem Rauschen und einzelnen defekten Pads ist. Man sieht, daß die Bestimmung der Hitposition durch das entwickelte Neuronale Netz deutlich schlechter als die Schwerpunktsbildung, und damit auch schlechter als die Gauß-Fit-Methode ist.

Um eine Verbesserung der Rekonstruktionsgenauigkeit zu erreichen, wurde als nächstes die oben beschriebene Netzwerkentwicklung wiederholt, nur daß diesmal die während des Trainierens vorgegebenen gewünschten Output-Werte (Koordinaten) nur auf 1/10 Pad genau angegeben wurden. Das Ergebnis ist in Abb. 6.11 zu

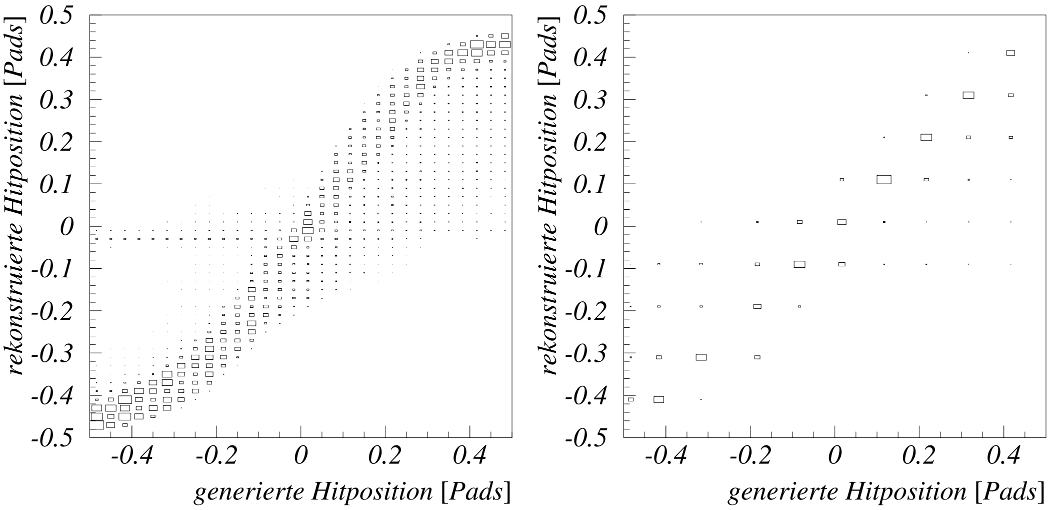

Abbildung 6.11: *Die durch die Neuronale Netzwerke mit zwei Ausgabeneuronen rekonstruierte Hitposition in Abhängigkeit vom generierten Hitzentrum.* Die Netze hatten zwei Ausgabeneuronen zur Repräsentation der x- bzw. y-Koordinate, wobei die während des Trainierens vorgegebenen Output-Werte auf 1/100 Pad (links) bzw. 1/10 Pad (rechts) genau waren.

sehen. Es ist deutlich zu erkennen, daß auch dieses Netzwerk keine Verbesserung der Rekonstruktionsgenauigkeit bringt.

Eine andere Möglichkeit ist, über das zentrale Pad der 5×5-Matrix eines Hits ein 11×11-Raster zu legen (ungerade, damit das Zentrum der Matrix nicht zwischen zwei Elemente der feineren Matrix fällt) und die so entstehenden 11 Abschnitte für die x- bzw. y-Position durch jeweils ein Ausgabeneuron zu repräsentieren. Insgesamt sind also 22 Ausgabeneuronen erforderlich. Dabei wird während des Trainierens den beiden Ausgabeneuronen, die den x- bzw. y-Koordinatenbereich repräsentieren, in dem sich das Hitzentrum befindet, der Wert 1 zugeordnet und allen restlichen Ausgabeneuronen der Wert 0. Die Anzahl der Eingabeneuronen betrug entsprechend der oben beschriebenen Vorgehensweise 25; die Anzahl der verborgenen Schichten sowie die Anzahl der verborgenen Neuronen wurden nach

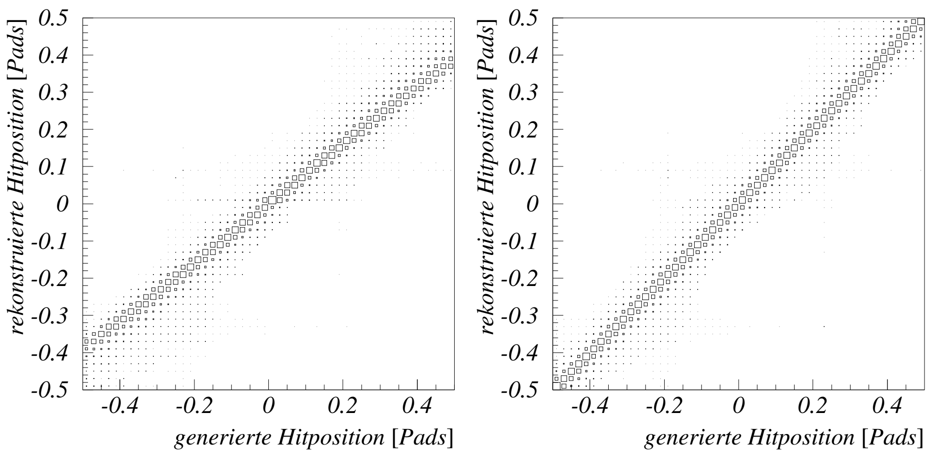

Abbildung 6.12: *Die rekonstruierte Hitposition in Abhängigkeit vom generierten Hitmittelpunkt nach der Schwerpunktsbildung (links) und nach der Gauß-Fit-Methode (rechts).* Beim bisherigen Analyseverfahren wird vor allem aus Rechenzeitgründen die Schwerpunktsbildung benutzt.

dem in Abschnitt 5.4 erläuterten Optimierungsverfahren ermittelt. Als optimale Topologie ergab sich ein dreilagiges Netz mit 178 verborgenen Neuronen. Dieses Netzwerk wurde analog zu den oben entwickelten Netzen trainiert und getestet. Das Ergebnis ist in Abb. 6.13 gezeigt. Auch hier ist zu erkennen, daß das Neuronale Netz der Schwerpunktsbildung bzw. der Gauß-Fit-Methode klar unterlegen ist.

Ein **reales Backpropagation-Netzwerk**, d.h. ein Backpropagation-Netzwerk von im Rahmen der Speicherkapazität realisierbarer Größe, ist somit nicht in der Lage, die Abbildung von den Inhalten der 5×5-Matrix eines Hits auf seine Position zu implementieren. Dies ist aber auch nicht verwunderlich, wenn man sich die Klassen von Funktionen vergegenwärtigt, die bis jetzt erfolgreich durch Backpropagation-Netzwerke implementiert worden sind: dabei handelt es sich um einfache reelle Funktionen der Gestalt $ax + b$, $\sin(ax + b)$ und ax^b, $a, b, x \in \mathbb{R}$ [39]. Die hier zu approximierende Funktion ist um ein vielfaches komplexer: sie ist eine Funktion von $\mathbb{R}^{25}$ nach $\mathbb{R}^2$, die aufgrund der Digitalisierung nicht-trivial ist. Sind die Hits mit elektronischem Rauschen überlagert und/oder besitzen sie einzelne

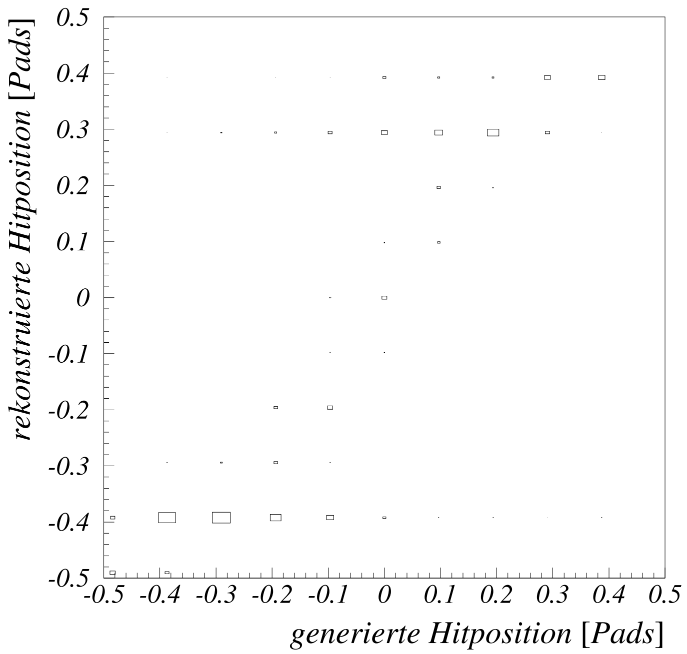

Abbildung 6.13: *Die durch das Neuronale Netze mit 22 Ausgabeneuronen rekonstruierte Hitposition in Abhängigkeit vom generierten Hitzentrum.* Das Netz hatte 22 Ausgabeneuronen zur Repräsentation der Bereiche, in denen die x- bzw. y-Koordinate der Hitposition liegt. Das Netz war dreilagig mit 178 verborgenen Neuronen.

defekte Pads, so wird diese Funktion hochgradig nicht-trivial. Nach dem oben Gesagten ist aber möglicherweise unter Verwendung eines anderen Lernalgorithmus mit einer glatten Aktivierungsfunktion die Implementierung dieser Funktion in einem Netzwerk von realisierbarer Größe erreichbar.
Die mit einem realen Backpropagation-Netzwerk erreichbare Rekonstruktionsgenauigkeit bei der Bestimmung der Hitposition ist weit schlechter als die Rekonstruktionsgenauigkeit von Schwerpunktsbildung und Gauß-Fit-Methode.

Als Folgerung ergibt sich somit, daß Backpropagation-Netzwerke zur Bestimmung der Position eines Hits aus den Amplituden der zu ihm gehörenden Pads ungeeignet sind.

Kapitel 7

Diskussion der Ergebnisse und Ausblick

In diesem letzten Kapitel werden die erhaltenen Ergebnisse zusammengefaßt, kritisch diskutiert und ein Ausblick auf mögliche Weiterentwicklungen und Verbesserungen der Mustererkennung im CERES-Experiment gegeben.

7.1 Filteroperationen im Fourierraum

Da die Methode der Filteroperationen in der Digitalen Bildverarbeitung das grundlegende Verfahren zur Erkennung von Bildstrukturen darstellt, wurde im ersten Teil dieser Arbeit diese Technik zur Suche nach Ringkandidaten eingesetzt. Aus der Strategie der Durchführung von Filteroperationen im Fourierraum wurde ein neues Konzept zur Identifizierung von Ringkandidaten entwickelt. Dadurch sollte eine effizientere und genauere Identifizierung von Ringkandidaten erreicht werden. Zunächst wurde gezeigt, wie sich bei Berechnungen von Faltungen mit Hilfe der DFT eine enorme Reduzierung des Rechenaufwandes erreichen läßt. Dann wurde die Funktionsfähigkeit des neuen Konzeptes nachgewiesen. Die Anwendung dieses Konzeptes auf reale Daten ergab, daß es bei Verwendung einer Doppelringmaske geeigneter Breite in Bezug auf die Rekonstruktionsgenauigkeit besser als das bisherige Verfahren ist; bezüglich der Ringfindeeffizienz und des Signal-zu-Untergrund-Verhältnisses sind beide Verfahren gleich gut. Die Verwendung von Binominal-, Laplace- oder Sobel- (Gradienten-) Operatoren brachte keine Verbesserung der untersuchten Größen; die Verwendung von Rechteckmasken brachte eine Verbesserung bezüglich der Rekonstruktionsgenauigkeit.
Das neue Verfahren wurde nur bezüglich der Funktionalität und nicht bezüglich der Laufzeit optimiert; diese ist für das neue Verfahren im Vergleich zum bisherigen

um einen Faktor von ca. 2.5 größer. Das neue Verfahren ist jedoch nicht grundsätzlich langsamer als das alte, da bei einer geschickten Berechnung der DFT beide Verfahren ungefähr gleich viele Rechenschritte benötigen. Abhilfe kann z.B. ein Basis-2- oder Basis-4-Algorithmus zur Berechnung der DFT schaffen, oder eine durch ein Neuronales Netzwerk implementierte DFT. Mit diesen Techniken läßt sich die für die Durchführung einer Faltung erforderliche Anzahl an reellen Additionen und Multiplikationen pro Pixel auf mindestens 40 bzw. 84 reduzieren. Durch Neuronale Netze implementierte DFT's sind sowohl als Software- wie auch als Hardware-Implementierungen kommerziell erhältlich.
Was Funktionalität und Laufzeit anbetrifft stellt das neue Verfahren somit eine Alternative zur bisherigen Suche nach Ringkandidaten dar. Es ist der Hough-Transformation insofern überlegen, als es sehr ausbaufähig ist: mit Faltungen lassen sich sehr unterschiedliche Operationen durchführen. Damit ist die Technik der adaptiven Filterung möglich, d.h. man berücksichtigt die Struktur der lokalen Bildumgebung und wählt aus einem Satz von Filtermasken eine geeignete aus. Aufgrund der Assoziativität der Faltung bzw. der Multiplikation könnte die Rejektion des elektronischen Rauschens und der Cluster in einem Analyseschritt mit der Identifizierung der Ringkandidaten durchgeführt und somit auf die entsprechenden beiden Säuberungsalgorithmen verzichtet werden. Zur Rejektion des Rauschens bieten sich Glättungs-, Bandpaß- oder Medianfilter an; die Rejektion der Cluster müßte mit adaptiver Filterung erfolgen, da diese sowohl auf wenige Pads begrenzt als auch auf größere Gebiete ausgedehnt sein können.
Das neue Verfahren erlaubt das Design von Filtern für einen spezifischen Zweck: man muß nur das gewünschte Faltungsergebnis vorgeben, seine DFT berechnen und die so erhaltenen Fourierkoeffizienten durch die Fourierkoeffizienten der DFT des Bildes dividieren (dieses Verfahren ist in der Digitalen Bildverarbeitung als inverse Filterung bekannt). Als Resultat erhält man dann die DFT des gewünschten Filteroperators. Wie gezeigt wurde, bildet sich im Realteil der zurücktransformierten Faltung eines Ringes mit einer entsprechenden Ringmaske ein globales Maximum an der Stelle, an der sich der ursprüngliche Ringmittelpunkt befand. In dieser Arbeit wurde nachgewiesen, daß bereits bei der Anwendung von Standardringmasken das neue Verfahren gleich gut wie das bisherige ist. Durch die Technik der inversen Filterung sind nun Verbesserungsmöglichkeiten gegeben: man könnte z.B. einen Filteroperator konstruieren, für den sich ein schmaleres globales Maximum und kleinere Nebenmaxima ergeben. Damit würde sich die Rekonstruktionsgenauigkeit der Ringzentren verbessern und die Anzahl der fälschlicherweise als Ringe rekonstruierten Strukturen (Untergrundringe) verkleinern.

7.2 Neuronale Netzwerke

Das bisherige Analyseverfahren weist bei der Rekonstruktion der Hits einige besonders gravierende Probleme auf. Diese beruhen im wesentlichen auf der im bisherigen Hitrekonstruktionsalgorithmus durchgeführten Schwerpunktsbestimmung in einer 3×3 Pads großen Matrix um jedes Pad. Die Schwerpunktsbildung hat zur Folge, daß der Algorithmus nicht sensitiv auf die Form der Hits ist; sie erlaubt außerdem nicht, die Zentren von zwei oder mehr sich überlappender Hits exakt zu bestimmen. Aufgrund der Beschränkung auf die 3×3-Matrix werden Hits großer Amplitude oder nicht rejizierte Cluster z.T. in viele einzelne „Hits" aufgespalten (Hitsplitting). Desweiteren ist der Algorithmus nicht in der Lage, Hits zu trennen, die einen Abstand ≤ 1 Pad haben, wodurch ca. 25% aller Hits durch Überlappung verlorengehen (Pileup). Mit einem verbesserten Algorithmus zur Hitrekonstruktion, der diese Probleme nicht mehr aufweisen würde, ließen sich insbesondere die Doppelhitauflösung, die Doppelringauflösung sowie die quantitativen Analysen der Hits erheblich verbessern. Deshalb ist es von vorrangigem Interesse, die Erkennung von Hits und Untergrund sowie die Bestimmung der Hitpositionen zu verbessern. Im zweiten, wesentlich größeren Teil dieser Arbeit wurde daher versucht, software-implementierte Neuronale Netzwerke zur Klassifizierung von einzelnen Hits, überlappenden Hits und Untergrund sowie zur Rekonstruktion von Hits zu entwickeln. Das Ziel war, eine Steigerung der Effizienz der Einzelhit-, Mehrfachhit- und Untergrunderkennung zu erreichen sowie die bislang komplexe Aufeinanderfolge einzelner Analysealgorithmen zu vereinfachen oder gar zu ersetzen.
Da zum Trainieren und Testen der zu entwickelnden Netzwerke Monte-Carlo-simulierte Daten verwendet wurden, war eine Untersuchung der Form der Signale einzelner UV-Photonen (Hits) sowie der Verteilung ihrer Ladung auf die einzelnen Pads erforderlich. Es stellte sich heraus, daß ein Hit in RICH 1 bzw. RICH 2 in sehr guter Näherung durch eine zweidimensionale Gaußsche Glockenkurve mit $\sigma = 0.81$ bzw. 0.63 Pads beschrieben wird. Die 5×5-Matrix um einen Hit stellt die optimale Matrix zu seiner Beschreibung dar. Durch eine theoretische Herleitung der Wahrscheinlichkeit für das Auftreten von n-fach überlappenden Hits auf Čerenkov-Ringen wurde gezeigt, daß Überlappungen vom Grad ≥ 3 vernachlässigbar sind. Die berechnete Wahrscheinlichkeit für das Auftreten eines Dreifachhits unter Verwendung der Erwartungswerte für die Anzahl der Hits pro Ring liegt nämlich unter einem Prozent, und die Wahrscheinlichkeit für das Auftreten von n-fach überlappenden Hits nimmt mit wachsendem n ab.
Die Monte-Carlo-Hits wurden mit den gemessenen Amplituden- und Breitenverteilungen erzeugt; zusätzlich wurden noch die Überlagerung mit elektronischem Rauschen, einzelne defekte Pads, Nullpunktsverschiebungen einzelner Pads sowie die obere und untere Schranke für die Padamplitude berücksichtigt. Bei der Simulation des Monte-Carlo-Untergrundes kristallisierte sich die Zurückführung der

Untergrundelemente auf vier Grundklassen heraus, an denen jeweils die oben aufgeführten Modifikationen vorgenommen wurden. Zur Bestimmung der hinsichtlich Leistungsfähigkeit und Laufzeit optimalen Anzahl von verborgenen Neuronen wurde ein Optimierungsverfahren entwickelt und angewendet. Das Ergebnis war, daß für die in dieser Arbeit behandelten Klassifizierungsprobleme eine 25-30-1-Architektur optimal ist. Eine Untersuchung mit relativen und absoluten Input-Variablen ergab, daß die linear auf das Intervall [0,1] normierten absoluten Inhalte der 5×5-Matrix um einen Hit den optimalen Satz an Input-Variablen für die behandelten Klassifizierunsprobleme darstellen.
Die Anwendung der entwickelten Neuronalen Netze auf Monte-Carlo-simulierte Daten ergab, daß Hits bis zu einer Breite von 50% der gemessenen Breite der Hits bzw. ab einer Breite von 130% von den Hits mit gemessener Breite durch Neuronale Netze getrennt werden können. Als Kriterium für die Trennbarkeit wurde dabei eine Effizienz von 80% festgelegt, mit der Elemente aus der zu trennenden Klasse richtig klassifiziert werden. "Hits" mit einer Breite, die größer als die gemessene ist, bilden die erste Grundklasse des Untergrundes. Die restlichen drei Grundklassen (Ebenen, Einzelpads und Rauschelemente) können auch dann, wenn sie so realistisch wie möglich erzeugt werden, mit einer Rejektion von über 95% von den Netzen erkannt werden. Legt man als Kriterium für die Trennung von Einzel- und Doppelhits eine Effizienz von 70% fest, mit der ein Doppelhit erkannt wird, so gilt: ideale Doppelhits lassen sich durch Neuronale Netzwerke noch trennen, wenn der Abstand zwischen den Schwerpunkten größer als 0.1 Pads und das Verhältnis der Amplituden kleiner als 1/10 ist. Realistische Doppelhits sind durch Neuronale Netze noch zu trennen, wenn der Abstand der Schwerpunkte größer als 0.85 Pads in RICH 1 bzw. 0.65 Pads in RICH 2 und das Amplitudenverhältnis kleiner als 1/4 ist. Damit kann die begrenzte Doppelhitauflösung des bisherigen Hitrekonstruktionsverfahrens um 90% für ideale Hits und um 15% bzw. 35% für realistische Hits in RICH 1 bzw. RICH 2 verbessert werden. Der schlechtere Wert in RICH 1 ist durch die größere Hitbreite zu erklären. Der Effizienzverlust, der sich bei der Erkennung von realistischen Einzel-, Doppelhits oder Untergrund im Vergleich zu idealen Daten ergibt, wird im wesentlichen durch das Vorhandensein von einzelnen defekten Pads verursacht. Weder die zum Trainieren erzeugten Monte-Carlo-Hits noch der Monte-Carlo-Untergrund konnte durch Neuronale Netze von den entsprechenden realen Daten wirklich getrennt werden. Dies zeigt, daß die erzeugten Monte-Carlo-Daten eine sehr gute Simulation der realen Daten darstellen.
Auf reale Daten angewendet sind die entwickelten Netzwerke in der Lage, über 90% der einzelnen Hits und Doppelhits sowie gleichzeitig über 80% des Untergrundes richtig zu klassifizieren.
Die Anwendung dieser Netzwerke auf bisher unverstandene quantitative Analysen der Hits ergab, daß der Überschuß der Polarwinkelverteilung der Hits bei kleinen Winkeln für RICH 1-Daten im Vergleich zu Monte-Carlo-simulierten Daten von Einzel- und Doppelhits und nicht von Untergrund herrührt. Die starke Abweichung der mit dem bisherigen Verfahren erhaltenen Verteilung der Hitamplituden

von der erwarteten exponentiellen Form ist hingegen im wesentlichen ein Effekt der schlechten Untergrundrejektion des bisherigen Verfahrens. Die Untersuchung der Anzahl der Hits in Abhängigkeit von der Multiplizität der Teilchen ergab, daß der für RICH 1-Daten im Gegensatz zur Monte-Carlo-Simulation erhaltene Offset zu etwa 2/3 physikalischer Natur ist; 30% des gesamten Offsets wird von idealen oder nahezu idealen Hits hervorgerufen, d.h. der Wert des Ausgabeneurons für diese Hits ist praktisch 1. Von diesen (nahezu) idealen Hits sind ca. 73% Einzel- und 27% Doppelhits. Die um einen Faktor von ca. 1.7 größere Geradensteigung für reale Daten im Vergleich zu Monte-Carlo-Daten wird im wesentlichen von Einzel- (ca. 40%) und Doppelhits (ca. 51%), also „echten" Hits, hervorgerufen. Dies impliziert die Existenz eines physikalischen Effektes in RICH 1, der durch die Monte-Carlo-Simulation nicht beschrieben wird.

Verwendet man die Neuronalen Netze zur Bestimmung der Anzahl der Hits pro Ring, so erhält man im Vergleich zur bisherigen Analyse einen um ca. 40% größeren Mittelwert, der mit dem erwarteten Mittelwert übereinstimmt. Folglich wird eine bessere Unterdrückung von Untergrundringen bei der Suche nach Ringkandidaten erwartet. Da Einzel- und Doppelhits nun klassifiziert werden können, ist eine genauere Rekonstruktion der Hits möglich. Zur Bestimmung der Position von Einzelhits ist die Schwerpunktsbestimmung ausreichend, wie sie im bisherigen Hitrekonstruktionsverfahren durchgeführt wird; um die Zentren der Hits zu bestimmen, die einen Doppelhit bilden, ist die Anpassung einer Doppel-Gaußverteilung erforderlich. Mit den so bestimmten Hitpositionen ist dann auch eine genauere Rekonstruktion des Ringmittelpunktes möglich.

Die Anwendung der entwickelten Netzwerke auf die Rekonstruktion der Elektronringe ergab im Vergleich zur bisherigen Analyse gleiche Ringfindeeffizienz und die gleiche Anzahl von Untergrundringen pro Ereignis. Da das neue Analyseverfahren ohne die Säuberungsalgorithmen der bisherigen Analyse auskommt, kann auf diese verzichtet werden. Die entwickelten Netzwerke stellen also eine Alternative zu den alten Säuberungsalgorithmen dar. Somit kann die Aufeinanderfolge von sechs Algorithmen bei der bisherigen Analyse auf die Aufeinanderfolge von vier Algorithmen vereinfacht und gestrafft werden.

Das neue Analyseverfahren wurde nur bezüglich der Funktionalität und nicht bezüglich der Laufzeit optimiert; die benötigte Rechenzeit ist für das neue Verfahren um einen Faktor von ca. 2.3 größer. Dieser Verlust an Geschwindigkeit kommt i.w. dadurch zustande, daß bei der neuen Analyse die Schwerpunktsbestimmung bei der Hitrekonstruktion auf alle Pads angewendet wird, deren Amplitude größer Null ist. Dies ist jedoch überflüssig. Somit könnte alleine schon durch eine geschickte Beschränkung der Schwerpunktsbildung auf einen Bruchteil dieser Pads eine deutliche Reduzierung des Rechenaufwandes erreicht werden. Eine weitere, enorm effektive Strategie zur Optimierung der Laufzeit besteht darin, die Werte der Sigmoidfunktionen, die bei einem im Ausführungsmodus betriebenen Netzwerk zu berechnen sind, durch eine Look-up-Table zu erheben. Bei hardwaremäßigen Implementierungen Neuronaler Netzwerke werden die Sigmoidfunktionen auf diese

Weise realisiert. Die schnellsten Hardware-Implementierungen werden durch analoge Chips erreicht. Mit diesen ist eine Rechenzeit von der Größenordnung $1\mu s$ pro Netzwerkschicht realisierbar. Das neu entwickelte Analyseverfahren zur Rekonstruktion der Elektronringe ist somit auch bezüglich der Laufzeit der bisherigen Analyse nicht prinzipiell unterlegen.
Im letzten Abschnitt der Arbeit wurde versucht, die Positionen der Hits mit Hilfe Neuronaler Netze zu rekonstruieren. Es konnte gezeigt werden, daß Multilayer-Feedforward-Netzwerke mit Backpropagation-Lernalgorithmus im Rahmen der Speicherkapazität zur Rekonstruktion der Hitzentren ungeeignet sind. Hierfür gibt es zwei mögliche Gründe: entweder reicht die im Rahmen der Speicherkapazität erreichbare Größe des Netzwerkes nicht aus, um die gewünschte Funktion zu implementieren, oder der Backpropagation-Lernalgorithmen ist für das gestellte Problem ungeeignet. Im zweiten Fall kann möglicherweise mit Hilfe eines anderen Lernalgorithmus die gewünschte Funktion in einem Netzwerk von realisierbarer Größe implementiert werden.
In dieser Arbeit wurde erstmalig der Einsatz Neuronaler Netzwerke bei der Mustererkennung auf Hitniveau untersucht. Es konnte gezeigt werden, daß bereits mit einem simplen Netzwerktyp wie dem Multilayer-Feedforward-Netzwerk unter Verwendung des Backpropagation-Lernalgorithmus ein Analyseverfahren entwickelt werden kann, welches in Bezug auf seine Funktionalität gleich gut wie die bisherige Analyse ist. Damit sind aber längst nicht alle Möglichkeiten ausgeschöpft. Vielversprechend ist z.B. die Untersuchung des Einsatzes von selbstorganisierenden sensorischen Karten (Self-Organizing-Mapping-Netzwerke), die in der Lage sind, ohne irgendwelche externe Überwachung Eigenschaften zu finden, nach denen sich die eingegebenen Daten separieren lassen (es wird also während des Trainierens kein gewünschter Output vorgegeben). Mit einem solchen Netzwerktyp könnten möglicherweise Eigenschaften gefunden werden, die eine effizientere Erkennung der zu klassifizierenden Hits und des Untergrundes erlauben sowie zusätzlich eine Trennung der verschiedenen Untergrundarten im gleichen Schritt ermöglichen. Kombiniert man eine selbstorganisierende Karte mit einer weiteren Schicht von Neuronen, für die der Grossberg-Lernalgorithmus angewendet wird, so erhält man einen neuen Netzwerktyp, das Counterpropagation-Netzwerk. Dieses ist in der Lage, bezüglich des statistischen Fehlers optimale Look-up-Tables von Funktionen zu implementieren. Solche Netzwerke sind neben den Klassifizierungsproblemen insbesondere für die Rekonstruktion der Hitpositionen von Interesse, da in diesem Fall eine nicht-triviale Funktion zu implementieren ist. Es existiert noch eine Vielzahl weiterer Netzwerktypen für die unterschiedlichsten Aufgabengebiete, die jedoch den Nachteil besitzen, daß ihre mathematisch-theoretischen Grundlagen bis jetzt noch unzureichend erforscht sind. Da bereits das Multilayer-Feedforward-Netzwerk erfolgreich bei der Analyse angewendet werden kann, ist eine weitergehende Untersuchung des Einsatzes von Neuronalen Netzwerken gerechtfertigt.

Anhang A

Die diskrete Fourier-Transformation

Die **diskrete Fourier-Transformation** (**DFT**) ist eine Abbildung von der Menge aller komplexen $M \times N$-Matrizen ($M, N \in \mathbb{N}_0$) auf sich selbst, wobei jeder Matrix $(F_{m,n})$ eine Matrix $(\hat{F}_{u,v})$ ($0 \le m, u \le M-1,\ 0 \le n, v \le N-1$) nach der Vorschrift

$$\hat{F}_{u,v} := \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F_{m,n} \exp\left(-\frac{2\pi imu}{M}\right) \exp\left(-\frac{2\pi inv}{N}\right) \tag{A.1}$$

zugeordnet wird.
Die **inverse DFT** ist durch

$$F_{m,n} := \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{F}_{u,v} \exp\left(\frac{2\pi imu}{M}\right) \exp\left(\frac{2\pi inv}{N}\right) \tag{A.2}$$

gegeben.

Die DFT besitzt folgende wichtige Eigenschaften (eine ausführliche Darstellung ist bei *Bracewell* [40] zu finden):

1. *Mittelwert*:
   Auf das Matrixelement $\hat{F}_{0,0}$ des Spektrums wird der Mittelwert der Urbildmatrix $(F_{m,n})$ abgebildet:

   $$\hat{F}_{0,0} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F_{m,n}\ . \tag{A.3}$$

2. *Periodizität*:
   Bei der Berechnung der jeweiligen Fourier-Transformierten bleibt die Periodizität im Orts- und Fourierraum erhalten:

   $$\text{Ortsraum:} \qquad F_{m+kM,n+lN} = F_{m,n}\ ;$$

$$\text{Fourierraum:} \quad \hat{F}_{u+kM,v+lN} = \hat{F}_{u,v} \quad \forall\, k,l \in \mathbb{Z} \,. \tag{A.4}$$

3. *Linerarität*:
   Die DFT ist eine lineare Operation:

$$\begin{aligned} \text{Ortsraum:} \quad & H_{m,n} = aF_{m,n} + bG_{m,n} \;; \\ \text{Fourierraum:} \quad & \hat{H}_{u,v} = a\hat{F}_{u,v} + b\hat{G}_{u,v} \quad (a,b \in \mathbb{C}) \,. \end{aligned} \tag{A.5}$$

4. *Symmetrie*:
   Symmetrie für Matrizen bedeutet, daß $F_{m,n} = \pm F_{-m,-n}$, wobei $+$ für **gerade** und $-$ für **ungerade** Matrizen gilt. Unter Ausnutzung der Periodizität (Gln. A.4) lautet die Symmetriebedingung:

$$F_{m,n} = \pm F_{M-m,N-n} \,. \tag{A.6}$$

   Der Punkt $(M/2, N/2)$ ist das Symmetriezentrum. Jede Matrix kann als Summe einer geraden und einer ungeraden Matrix geschrieben werden.

   Es gilt: die Symmetrieeigenschaften gerade und ungerade bleiben unter der DFT erhalten.

5. *Verschiebungssatz*:
   Eine Verschiebung um einen konstanten Betrag im Ortsraum ergibt eine Phasenverschiebung der spektralen Amplitude im Fourierraum.

$$\begin{aligned} \text{Ortsraum:} \quad & H_{m,n} = F_{m-k,n-l} \quad (k,l \in \mathbb{N}_0) \;; \\ \text{Fourierraum:} \quad & \hat{H}_{u,v} = \hat{F}_{u,v} \exp\left(-\frac{2\pi iku}{M}\right) \exp\left(-\frac{2\pi ilv}{N}\right) \,. \end{aligned} \tag{A.7}$$

6. *Faltungstheorem*:
   Für Matrizen ist die **Faltung** durch

$$H_{m,n} = F_{m,n} * G_{m,n} := \sum_{k}^{M-1} \sum_{l}^{N-1} F_{k,l} G_{m-k,n-l} = \sum_{k}^{M-1} \sum_{l}^{N-1} F_{m-k,n-l} G_{k,l} \tag{A.8}$$

   definiert.
   Das Faltungstheorem lautet:

$$\begin{aligned} \text{Ortsraum:} \quad & H_{m,n} = F_{m,n} * G_{m,n} \;; \\ \text{Fourierraum:} \quad & \hat{H}_{u,v} = MN\hat{F}_{u,v}\hat{G}_{u,v} \,. \end{aligned} \tag{A.9}$$

   Die diskrete Fourier-Transformierte der Faltung zweier Matrizen ist also bis auf einen konstanten Faktor gleich dem Produkt der diskreten Fourier-Transformierten der beiden Matrizen.

7. *Das Parsevalsche Theorem*:
   Es lautet:
   $$\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|F_{m,n}|^2 = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|\hat{F}_{u,v}|^2 \ . \tag{A.10}$$
   Das Betragsquadrat eines komplexen Spektrums wird als als **Energie-** oder **Powerspektrum** bezeichnet. Das Parsevalsche Theorem sagt nichts anderes aus, als daß sich die gleiche Gesamtenergie ergibt, gleichgültig, ob man diese örtlich oder über alle Wellenlängen aufaddiert. Mathematisch gesehen ist das Parsevalsche Theorem nichts anderes als die Erhaltung der Norm.

Anhang B

Untersuchung der Signale einzelner UV-Photonen

Da die Strategie verfolgt wurde, zum Trainieren und Testen der zu entwickelnden Netzwerke Monte-Carlo-simulierten Daten zu verwenden, war es erforderlich, die Form der Hits und die Verteilung ihrer Ladung auf die einzelnen Pads zu kennen. Dies wird in diesem Anhang untersucht.

B.1 Messung der Form der Signale

Aufgrund der Digitalisierung wird kein kontinuierliches Signal beobachtet, sondern die auf den einzelnen Pads induzierte Ladung - das ist das Integral der kontinuierlichen Verteilung über die Fläche des Pads. Wir legen nun eine 5×5 Pads große Matrix so über das diskrete Signal, daß das zentrale Pad des Hits mit dem zentralen Pad der Matrix identisch ist. Die so jeder Ladungsverteilung in kanonischer Weise zugeordnete Matrix wird im folgenden als **5×5-Matrix des Hits** bzw. **der Ladungsverteilung** bezeichnet.

Von *Drees* [41] und mir wurde ein Verfahren zur Messung der Form der Hits entwickelt, das unten vorgestellt wird. Das Verfahren hat den Vorteil, daß keinerlei Annahmen über die Form der Hits eingehen. Das Verfahren läßt sich in folgende Schritte gliedern:

1. Ausgangspunkt ist die 5×5-Matrix eines beliebigen Hits.
2. Wir verfeinern die Rasterung um einen bestimmten Faktor x und erhalten eine $(5 \cdot x) \times (5 \cdot x)$-Matrix, die im folgenden als die **$(5 \cdot x) \times (5 \cdot x)$-Matrix des Hits** bezeichnet wird. x muß ungerade sein, damit der Mittelpunkt des Hits

nicht zwischen zwei Matrixelementen zu liegen kommt. Wir wählen zunächst $x = 11$.

3. In der nun erhaltenen 55×55-Matrix liegt das Hitzentrum in einem bestimmten Matrixelement, das i.a. vom zentralen Matrixelement verschieden ist. In diesem Fall führen wir eine Translation[1] der Matrixelemente durch, so daß das Hitzentrum im zentralen Matrixelement zu liegen kommt.

4. Die auf diese Weise erhaltene 55×55-Matrix wird gespeichert.

5. Die Schritte 1 bis 4 werden so lange wiederholt, bis die Anzahl der erhaltenen 55×55-Matrixen ausreicht, um eine vernünftige Aussage bezüglich des statistischen Fehlers treffen zu können.

6. Die in Schritt 4 gespeicherten Matrizen werden „übereinandergestapelt", d.h. die Inhalte der entsprechenden Matrixelemente der einzelnen Matrizen werden addiert.

7. Die Inhalte der Matrixelemente der so gewonnen Matrix werden durch die Anzahl der verwendeten Hits dividiert, um eine Normierung zu erhalten[2].

8. Da die Zentren der Hits uniform im zentralen Pad der 5×5-Matrix verteilt sind, sollte sich dann durch „Herausmitteln" bis auf die Amplitude wieder der ursprüngliche, an der Stelle (0,0) zentrierte Hit in einem 55×55-Raster ergeben. Für $x \rightarrow \infty$ sollten wir bis auf die Amplitude wieder die ursprüngliche kontinuierliche Ladungsverteilung erhalten.

Führt man dieses Verfahren mit Monte-Carlo-simulierten idealen Hits in RICH 1 durch, so erwartet man als Ergebnis eine in einer 55×55-Matrix gerasterte zweidimensionale Gaußverteilung mit einem Sigma von 0.81 Pads, die bei (0,0) zentriert ist[3]. Das erhaltene Ergebnis stimmt mit dem erwarteten bis auf das Sigma überein: es ergibt sich ein Sigma von 0.883 Pads.
Als nächstes wurde das oben beschriebene Verfahren mit 500 realen Hits in RICH 1 durchgeführt, die durch visuelle Überprüfung und anschließender Klassifizierung aus S-Au-Daten gewonnen wurden. Man erhält eine in einer 55×55-Matrix gerasterte zweidimensionale Gaußverteilung mit einem Sigma von 0.887 Pads, die bei (0,0) zentriert ist.
In gleicher Weise wurde das Verfahren auch auf ideale MC-Hits und reale Hits in RICH 2 angewendet. Die realen Hits in RICH 2 wurden analog zu denen in RICH 1 gewonnen (insgesamt 506 Hits). Das Sigma der übereinandergestapelten MC-Hits ergab sich zu 0.747 Pads; für das Sigma der übereinandergestapelten realen Hits

[1] Man kann diese Translation auch als Umbenennung der Matrixelemente auffassen.
[2] Auf diesen Punkt kann auch verzichtet werden.
[3] Wir werden im folgenden sehen, daß hier noch eine Korrektur vorzunehmen ist.

wurde 0.750 Pads gemessen.
Desweiteren wurde das Verfahren noch mit den Verfeinerungsfaktoren $x = 5$ und $x = 15$ für RICH 1 durchgeführt. Für $x = 5$ bzw. $x = 15$ ergab sich das Sigma der der übereinandergestapelten MC-Hits zu 0.883 Pads bzw. 0.883 Pads; das Sigma der übereinandergestapelten realen Hits wurde zu 0.887 Pads bzw. 0.887 Pads gemessen.

Anhand der Messungen ist zu erkennen: die erhaltene Verteilung hat ein größers RMS als die erwartete; und zwar ein um 10.0% größeres in RICH 1 bzw. 18.6% größeres in RICH 2. Dies ist aber nur ein scheinbarer Widerspruch. Das größere RMS der erhaltenen Verteilung wird nämlich erwartet: das RMS einer digitalisierten Verteilung ist größer als das der ursprünglichen kontinuierlichen, wenn diese nicht konstant ist. Diesen Sachverhalt kann man sich z.B. anhand einer linearen Verteilung leicht klar machen. Je feiner die Rasterung, desto kleiner wird die Zunahme des RMS; geht die Feinheit der Rasterung gegen Unendlich, so geht die Zunahme des RMS gegen Null, denn dann liegt wieder die kontinuierliche Verteilung vor. Dem beschriebenen Verfahren liegt die 5×5-Matrix des jeweiligen Hits, also immer dieselbe Digitalisierung zugrunde. Deshalb erwarten wir, daß die gemessene Zunahme des RMS unabhängig vom Verfeinerungsfaktor ist. Dies wird auch beobachtet.
Die für reale Hits und MC-Hits erhaltenen Verteilungen unterscheiden sich *weder* in ihrer Form *noch* in ihrer Breite (RMS). Gäbe es einen signifikanten Unterschied, so wäre er mit dem entwickelten Verfahren feststellbar.
Als Folgerung ist also festzuhalten, daß ein Hit in RICH 1 bzw. RICH 2 in sehr guter Näherung durch eine zweidimensionale Gaußsche Glockenkurve mit $\sigma = 0.81$ Pads bzw. 0.63 Pads beschrieben wird.

	RICH 1	RICH 2
3×3-Matrix	84.5 %	93.9 %
5×5-Matrix	99.8 %	100.0 %

Tabelle B.1: *Durchschnittlicher Anteil an der Gesamtladung eines Hits innerhalb der 3×3- bzw. 5×5-Matrix.*

B.2 Untersuchung der Verteilung der Ladung eines Hits

Zur Bestimmung des Anteils an der Gesamtladung eines Hits, der sich im Mittel innerhalb seiner 3×3- bzw. 5×5-Matrix befindet, wurden jeweils 10^6 MC-Hits in RICH 1 bzw. RICH 2 mit den in Abschnitt B.1 gemessenen Breiten und uniform im zentralen Pad der 5×5-Matrix verteilten Zentrumskoordinaten erzeugt. Aus den sich jeweils ergebenden Ladungsanteilen wurden das arithmetische Mittel gebildet. Die erhaltenen Werte sind in Tabelle B.1 angegeben.

Als Folgerung ist somit festzuhalten, daß die 5×5-Matrix eines Hits die kleinste Matrix ist, die praktisch noch seine gesamte Ladung erfaßt. Sie stellt somit die optimale Matrix zu seiner Beschreibung dar.

Anhang C

Untersuchungen zur Laufzeit

In diesem Anhang werden die in den Abschnitten 3.1 und 6.7 entwickelten neuen Analyseverfahren zur Rekonstruktion der Ringe in Bezug auf ihre Laufzeit untersucht.

C.1 Filteroperationen im Fourierraum

In Abschnitt 3.1 wurde aus dem Konzept der Durchführung von Filteroperationen im Fourierraum ein neues Verfahren zur Rekonstruktion der Elektronringe entwickelt. Bestimmt man die Zeit, die für die Verarbeitung eines Ereignisses im Durchschnitt benötigt wird, so erhält man (0.412 ± 0.096) Sekunden für das bisherige Verfahren und (1.02 ± 0.25) Sekunden für das neue Verfahren bei Verwendung der einfachsten Doppelringmaske der Breite 3 Pads zusammen mit der 3×3-Rechteckmaske; dies bedeutet also, daß das neue Konzept in Bezug auf die Geschwindigkeit um einen Faktor von ca. 2.5 schlechter als das bisherige Verfahren ist. Da das neue Verfahren bei einer geschickten Berechnung der DFT weniger Rechenschritte als das alte benötigt, liegt der Grund für den Verlust an Laufzeit in der fehlenden Optimierung.

Allerdings war das vorrangige Ziel bei der Entwicklung des neuen Konzeptes die Optimierung der Funktionalität - d.h. der drei zu messenden Größen Ringfindeeffizienz, Signal-zu-Untergrund-Verhältnis und Rekonstruktionsgenauigkeit -, und nicht die Optimierung der Laufzeit, da das obige Rechenzeitproblem aus folgenden Gründen eine untergeordnete Rolle spielt:
es werden ständig neue, effektivere Algorithmen zur schnellen Berechnung der DFT entwickelt (FFT-Algorithmen; siehe *Blahut* [14]). In der vorliegenden Arbeit wurde der Complex Fast Fourier Transform Algorithm von *Singleton* [42] aus der Programmbibliothek des CERN verwendet. Dieser Algorithmus wurde gewählt, weil

er die Berechnung der DFT von Matrizen erlaubt, deren Zeilen- und Spaltenlänge kein Vielfaches von 2 ist. Somit können die im CERES-Experiment auftretenden Bildmatrizen in unveränderter Größe transformiert werden. Basis-2- oder Basis-4-Algorithmen sind jedoch wesentlich schneller: mit ihnen läßt sich die Anzahl der erforderlichen Additionen und Multiplikationen um einen Faktor von ca. 10^3 reduzieren. (siehe Abschnitt 2.1).
Die neueste Strategie zur schnellen Berechnung von Fourier-Transformationen besteht darin, die DFT durch ein Neuronales Netzwerk zu implementieren. Dadurch kann der Rechenaufwand im Vergleich zu konventionellen FFT-Algorithmen erheblich reduziert werden. So ist etwa in dem kommerziellen Neural Network Software Package DynaMind von NeuroDynamX [43] ein DFT Network File enthalten, daß die Berechnung der DFT von quadratischen Matrizen beliebiger Größe (im Rahmen der Speicherkapazität) erlaubt. Diese DFT-Netzwerke erfordern kein Training mehr; die Gewichtsparameter sind bereits berechnet. Es ist interessant zu erwähnen, daß die Netzwerke in DynaMind zur Implementierung der DFT folgende ungewöhnliche Eigenschaften besitzen: sie bestehen nur aus Input- und Output-Layer, besitzen also keine Hidden-Nodes, und die verwendeten Aktivierungsfunktionen sind linear.

C.2 Neuronale Netzwerke

In Abschnitt 6.7 wurde unter Verwendung der entwickelten Neuronalen Netzwerke ein neues Analyseverfahren zur Rekonstruktion der Elektronringe konzipiert. Mißt man die Zeit, die das jeweilige Verfahren zur Analyse eines Ereignisses im Durchschnitt benötigt, so erhält man (1.02 ± 0.28) Sekunden für das bisherige Verfahren und (2.34 ± 0.96) Sekunden für das neue Verfahren mit den Positionen aller zu den Hits gehörenden Pads. Dieser Verlust an Geschwindigkeit um einen Faktor von ca. 2.3 kommt i.w. dadurch zustande, daß bei der neuen Analyse der Schwerpunktsalgorithmus des Hitrekonstruktionsverfahrens auf alle Pads angewendet wird, deren Amplitude größer Null ist, während bei der bisherigen Analyse der Schwerpunktsalgorithmus nur auf solche Pads angewendet wird, die nach Anwendung der Säuberungsalgorithmen noch verbeiben.

Auch hier war das vorangige Ziel bei der Entwicklung des neuen Verfahrens die Maximierung der Ringfindeeffizienz und der Unterdrückung der Untergrundringe und nicht die Minimierung der Geschwindigkeit. Die gängige Techniken zur Optimierung der Laufzeit sind:

1. Ersetzung von Fließkommazahlen durch Integereinheiten.
2. Vermeidung von totem Coding; also z.B. Überprüfung auf überflüssige Statements (insbesondere Schleifen), Vermeidung von geschachtelten Loops.

3. Lokalität des Coding; hier ist z.B. zu erwähnen, daß der Hauptzweig einer if-Abfrage an erster Stelle stehen muß. Möglichst viele Operationen, die auf die gleichen Daten zugreifen, sollten zusammengefaßt werden.

4. Beim Einsatz von Multiprozessoren ist auf parallelisierbares Coding zu achten. Neue Vektorrechner wie der RC 6000 von IBM verfügen über ein sogenanntes Pipelining zur Realisierung paralleler Strukturen.

5. Viele Compiler besitzen einen Optimizer-Mode. Mit diesem kann z.B. bei C-Compilern bis zu 30% an Geschwingigkeit gewonnen werden.

6. Verwendung von Look-up-Tables anstelle der Funktionen.

Diese Techniken sind bis auf den letzten Punkt in den untersuchten Analysealgorithmen überwiegend realisiert (sie werden vom Compiler z.T. automatisch geleistet). Mittlerweile ist auch die verbesserte Version JETNET 3.0 [44] erschienen. Da bei einem im Ausführungsmodus betriebenen Netzwerk i.w. Sigmoidfunktionen zu berechnen sind, bietet es sich insbesondere hier an, die Werte der Sigmoidfunktionen durch eine Look-up-Table zu erheben, was eine erhebliche Reduzierung des Rechenaufwandes bedeutet. Bei hardwaremäßigen Implementierungen Neuronaler Netzwerke werden die Sigmoidfunktionen auf diese Weise realisiert. Zum Aufbau eines Neuronalen Netzes in Hardware ist ein Chip für schnelle Matrix-Vektor-Multiplikation erforderlich, die die Grundoperation eines Netzes bildet. Eine Reihe solcher Bausteine mit einer Rechenleistung von mindestens 400 Millionen Verbindungen pro Sekunde sind bereits auf dem Markt. Die schnellsten Implementierungen werden durch analoge Chips wie den Intel ETANN-Chip [45] erreicht. Die Latenzzeit des ETANN bis zum Erhalt des Ergebnisses ist von der Größenordnung $1\mu s$ pro Schicht, unabhängig von der Anzahl der Eingabeneuronen oder der verborgenen Neuronen. Der ETANN bietet den Vorteil, daß er über das Intel INNTS-Interface [45] durch das Softwarepaket DynaMind angesteuert werden kann.

Anhang D

Wahrscheinlichkeit für das Auftreten von n-fach überlappenden Hits auf Čerenkov-Ringen

Wir betrachten zunächst einen Ring mit einer festen, aber beliebigen Anzahl von N Photonen, $N \in \mathbb{N}$.
Die Wahrscheinlichkeit, daß zwei der N Photonen im Winkelabstand ϕ_1, $0 < \phi_1 < 2\pi$, angeordnet sind, beträgt

$$\frac{N-1}{2\pi}\left(1-\frac{\phi_1}{2\pi}\right)^{N-2} \quad .$$

(Die Wahrscheinlichkeit, daß zwei Photonen einen Winkel zwischen ϕ_1 und $\phi_1 + \Delta\phi_1$ einschließen, ist $\Delta\phi_1/2\pi$; die Wahrscheinlichkeit, daß die $N-2$ verbleibenden Photonen außerhalb des Bogens liegen, den die beiden Photonen einschließen, beträgt $(1-\phi_1/2\pi)^{N-2}$; es gibt $N-1$ Kombinationen zwischen zwei Photonen.) Die Wahrscheinlichkeit, daß drei der N Photonen in den Winkelabständen ϕ_1, ϕ_2, $0 < \phi_1 + \phi_2 < 2\pi$, angeordnet sind, beträgt

$$\frac{(N-1)\cdot(N-2)}{2\pi\cdot(2\pi-\phi_1)}\left(1-\frac{\phi_1+\phi_2}{2\pi}\right)^{N-3} \quad .$$

Verallgemeinert man obige Formeln auf festes, aber beliebiges n, so erhält man für die Wahrscheinlichkeit $p_N(n, \phi_1, \ldots, \phi_{n-1})$, daß n der N Photonen auf einem Ring in den Winkelabständen $\phi_1, \ldots, \phi_{n-1}, 0 < \sum_{i=1}^{n-1} \phi_i < 2\pi$, angeordnet sind, die Formel

$$p_N(n, \phi_1, \ldots, \phi_{N-1}) =$$

$$\frac{(N-1)\cdot\ldots\cdot(N-n+1)}{2\pi\prod\limits_{k=1}^{n-2}\left(2\pi-\sum\limits_{l=1}^{k}\phi_l\right)}\left(1-\frac{\sum\limits_{m=1}^{n-1}\phi_m}{2\pi}\right)^{N-n} \quad \text{für } N\geq n\ . \qquad \text{(D.1)}$$

Diese Formel läßt sich durch vollständige Induktion nach n beweisen.

Die Anzahl N der Photonen auf dem Ring sei jetzt - wie dies bei Čerenkov-Ringen der Fall ist - Poisson-verteilt mit Mittelwert $\langle N\rangle$, also

$$f_{\langle N\rangle}(N) := \frac{e^{-\langle N\rangle}\langle N\rangle^N}{N!}\ , \quad N\in \mathbb{N}_0\ . \qquad \text{(D.2)}$$

Die Wahrscheinlichkeit $p_{\langle N\rangle}(n,\phi_1,\ldots,\phi_{n-1})$, daß n der Photonen auf einem Čerenkov-Ring mit mittlerer Photonenzahl $\langle N\rangle$ in den Winkelabständen $\phi_1,\ldots,\phi_{n-1}$ angeordnet sind, ist dann durch

$$p_{\langle N\rangle}(n,\phi_1,\ldots,\phi_{n-1}) = \frac{\sum\limits_{N=n}^{\infty} f_{\langle N\rangle}(N)\, p_N(n,\phi_1,\ldots,\phi_{n-1})\, N}{\sum\limits_{N=n}^{\infty} f_{\langle N\rangle}(N)\, N} \qquad \text{(D.3)}$$

gegeben. Da Formel D.1 nur für $N\geq n$ definiert ist, wird in Gl. D.3 von $N=n$ an summiert. Einsetzen von D.1 und D.2 in D.3 ergibt

$$p_{\langle N\rangle}(n,\phi_1,\ldots,\phi_{n-1}) =$$

$$\frac{\langle N\rangle^n e^{-\langle N\rangle}}{2\pi\prod\limits_{k=1}^{n-2}\left(2\pi-\sum\limits_{l=1}^{k}\phi_l\right)}\sum_{N=n}^{\infty}\frac{\langle N\rangle^{N-n}}{(N-n)!}\left(1-\frac{\sum\limits_{m=1}^{n-1}\phi_m}{2\pi}\right)^{N-n} \Bigg/ \sum_{N=n}^{\infty}\frac{e^{-\langle N\rangle}\langle N\rangle^N}{N!}N\ .$$

Unter Verwendung der Potenzreihenentwicklung der Exponentialfunktion erhält man schließlich

$$p_{\langle N\rangle}(n,\phi_1,\ldots,\phi_{n-1}) =$$
$$\frac{\langle N\rangle^n}{2\pi\prod\limits_{k=1}^{n-2}\left(2\pi-\sum\limits_{l=1}^{k}\phi_l\right)}\exp\left(-\langle N\rangle\sum_{m=1}^{n-1}\frac{\phi_m}{2\pi}\right) \Bigg/ \sum_{N=n}^{\infty}\frac{e^{-\langle N\rangle}\langle N\rangle^N}{N!}N\ . \qquad \text{(D.4)}$$

Die Wahrscheinlichkeit $w_{\langle N\rangle}(n,\Delta\phi_1,\ldots,\Delta\phi_{n-1})$, daß n der Photonen auf einem Čerenkov-Ring mit mittlerer Photonenzahl $\langle N\rangle$ einen Winkel zwischen ϕ_1 und $\phi_1+\Delta\phi_1,\ldots,\phi_{n-1}$ und $\phi_{n-1}+\Delta\phi_{n-1}$ einschließen, erhalten wir nun durch sukzessive Integration von Gl. D.4 bzgl. der Winkel ϕ_i über die einzelnen Intervalle $[0,\Delta\phi_i]$ $(0\leq i\leq n-1)$:

$$w_{\langle N\rangle}(n,\Delta\phi_1,\ldots,\Delta\phi_{n-1}) =$$
$$\int\limits_0^{\Delta\phi_{n-1}}\ldots\left(\int\limits_0^{\Delta\phi_1} p_{\langle N\rangle}(n,\phi_1,\ldots,\phi_{n-1})\, d\phi_1\right)\ldots d\phi_{n-1}\ . \qquad \text{(D.5)}$$

Wir werten dieses $(n-1)$-dimensionale Integral für verschiedene n aus.
Für $n = 2$ ergibt sich

$$w_{\langle N \rangle}(2, \Delta\phi_1) = \frac{\langle N \rangle \left(1 - e^{-\langle N \rangle \frac{\Delta\phi_1}{2\pi}}\right)}{\sum\limits_{N=2}^{\infty} \frac{e^{-\langle N \rangle}\langle N \rangle^N}{N!} N} \, . \tag{D.6}$$

Um das Integral für $n = 3$ zu berechnen, verwenden wir die Integralexponentialfunktion und erhalten

$$\begin{aligned} & w_{\langle N \rangle}(3, \Delta\phi_1, \Delta\phi_2) = \\ & \quad \langle N \rangle^2 e^{-\langle N \rangle} \left(1 - e^{-\frac{\langle N \rangle}{2\pi}\Delta\phi_2}\right) \left(\frac{\langle N \rangle}{2\pi}\Delta\phi_1 + \ln(2\pi) - \ln(2\pi - \Delta\phi_1)\right) \\ & \quad / \sum_{N=3}^{\infty} \frac{e^{-\langle N \rangle}\langle N \rangle^N}{N!} N \, , \end{aligned} \tag{D.7}$$

wobei nur Terme erster Ordnung berücksichtigt sind, was für $\Delta\phi_1,\ \Delta\phi_2 \ll 1$ gerechtfertigt ist. Für $n = 4$ erhalten wir unter Benutzung geeigneter linearer Transformationen

$$\begin{aligned} & w_{\langle N \rangle}(4, \Delta\phi_1, \Delta\phi_2, \Delta\phi_3) = \\ & \quad \langle N \rangle^3 e^{-\langle N \rangle} \left(e^{-\frac{\langle N \rangle}{2\pi}\Delta\phi_3} - 1\right) \\ & \quad \left(\frac{\langle N \rangle}{2\pi}\Delta\phi_2 \left(\frac{1}{(2\pi - \Delta\phi_1)^2} - \frac{1}{(2\pi)^2}\right) + (\ln(2\pi - \Delta\phi_1))^2 - (\ln 2\pi)^2\right) \\ & \quad / \sum_{N=4}^{\infty} \frac{e^{-\langle N \rangle}\langle N \rangle^N}{N!} N \, . \end{aligned} \tag{D.8}$$

Da Formel D.5 bei Einsetzung geeigneter Winkelabstände $\Delta\phi_i,\ 0 \le i \le n-1$, nichts anderes als die Wahrscheinlichkeit für das Auftreten n-fach überlappender Hits auf Čerenkov-Ringen angibt, können wir nun aus D.6, D.7 und D.8 die Wahrscheinlichkeit für das Auftreten von Doppel-, Dreifach- und Vierfachhits auf Čerenkov-Ringen berechnen.

Bei dem im CERES-Experiment vorliegenden Fall beträgt der Radius R eines asymptotischen Čerenkov-Ringes 14.5 Pads in RICH 1 bzw. 16.4 Pads in RICH 2; für die mittlere Anzahl $\langle N \rangle$ der Čerenkov-Photonen pro Ring berechnet man 11.8 bzw. 12.5. Als Kriterium für das Auftreten von Überlappung legen wir fest: zwei Hits überlappen sich, wenn der Abstand ihrer Mittelpunkte ≤ 3 Pads ist. Es gilt dann $\Delta\phi_i = 3$ Pads$/R,\ 0 \le i \le n-1$.
Die mit diesen Werten berechneten Wahrscheinlichkeiten sind in Tab. D.1 zusammengefaßt.

n	RICH 1	RICH 2
2	32.2%	30.5%
3	1.15%	0.570%
4	0.406%	0.191%

Tabelle D.1: *Wahrscheinlichkeit für das Auftreten von n-fach überlappenden Hits auf asymptotischen Čerenkov-Ringen im CERES-Experiment.* Die Wahrscheinlichkeiten wurden nach D.6, D.7 und D.8 mit den im Text angegebenen Werten berechnet.

Literaturverzeichnis

[1] CERES/NA45, Proposal CERN/SPSC 88-25, SPSC/P237 und Addendum SPSC/P237 Add 1 (1988).

[2] T. Ullrich: Doktorarbeit, Universität Heidelberg (1994).

[3] D. Irmscher: Doktorarbeit, Universität Heidelberg (1993).

[4] A. Schön: Doktorarbeit, Universität Heidelberg (1993).

[5] T. Günzel: Doktorarbeit, Universität Heidelberg (1993).

[6] J. Seguinot und T. Ypsilantis: Nucl. Instr. and Meth. 142 (1977) 377.

[7] I. Endo et al.: Nucl. Instr. and Meth. 51 (1981) 188.

[8] D.E. Knuth: The Art of Computer Programming, Volume 1: Fundamental Algorithms, Addison Wesley P. C., Reading, Mass., 1969.

[9] N.I. Chernov, G.A. Ososkov: Comp. Phys. Comm. 33 (1984) 329.

[10] B. Jähne: Digitale Bildverarbeitung, 2. Auflage, Springer-Verlag, Berlin Heidelberg 1991.

[11] M. Gyulassy, M. Harlander: Elastic Tracking and Neural Network Algorithm for complex Pattern Recognition, Comp. Phys. Comm. 66 (1991) 31.

[12] B. Denby: Tutorial on Neural Network Applications in High Energy Physics, Fermilab-Conf. -92/121-E (1992) 4.

[13] J.P. Jaroslavskij: Einführung in die digitale Bildverarbeitung, Deutscher Verlag der Wissenschaften, Berlin 1985.

[14] R.E. Blahut: Fast Algorithms for Digital Signal Processing, Addison-Wesley, Reading, Mass., 1885.

[15] E.O. Brigham: FFT: Schnelle Fourier-Transformation, Oldenburg-Verlag, München, 1982.

[16] P. Haberäcker: Digitale Bildverarbeitung, 3. Auflage, Hanser-Verlag, München, Wien 1989.

[17] E. Schöneburg, N. Hansen, A. Gawelczyk: Neuronale Netzwerke - Einführung, Überblick und Anwendungsmöglichkeiten, Markt und Technik Verlag, München, 1990.

[18] D.E. Rumelhart, J.L. McClelland: Parallel Distributed Processing, Volume 2: Psychological and biological models, MIT Press, Cambridge, 1986.

[19] A.N. Kolmogorov: On the Representation of Continuous Functions of Many Variables by Superimposition of Continuous Functions of One Variable and Addition [in Russian], Dokl. Akad. Nauk USSR, **114**, 953-956, 1957.

[20] D.A. Sprecher: On the structure of continuous functions of several variables, Trans. Am. Math. Soc., 115, 340-355, March 1965.

[21] R. Hecht-Nielsen: Counterpropagation Networks, Proc. of the Int. Conf. on Neural Networks, II, 19-32, IEEE Press, New York, June 1987.

[22] R.P. Lippmann: An Introduction to Computing with Neural Nets, IEEE ASSP Magazine (1987) 4.

[23] C. Peterson, T. Rögnvaldsson: Introduction to Artificial Neural Networks, Lund University Preprint, LU TP 91-23, 1991.

[24] D.E. Rumelhart, J.L. McClelland: Parallel Distributed Processing, Volume 1: Foundations, MIT Press, Cambridge, 1986.

[25] R.O. Duda, P.E. Hart: Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.

[26] M.D. Richard, R.P. Lippman: Neural Network classifiers estimate bayesian a posteriori probabilities, Neural Computation, 3 (1991) 461.

[27] H. White: Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings, Neural Networks, 3 (1990) 535.

[28] E. Baum, D. Haussler: What Net Size gives Valid Generalization ?, Neural Computation 1 (1989) 151.

[29] K. Hornik, M. Stinchcombe, H. White: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks, 3 (1990) 551.

[30] W.S. Babbage, L.F. Thompson: The use of neural networks in $\gamma - \pi^0$ discrimination, Nuc. Instr. and Meth. A330 (1993) 482.

[31] L. Lönnblad, C. Peterson, T. Rögnvaldsson: Using Neural Networks to identify Jets, Nucl. Phys. B349 (1991) 675.

[32] A. van Ooyen, B. Nienhuis: Improving the Convergence of the Back-Propagation-Algorithm, Neural Networks, Vol. 5 (1992) 465.

[33] R. Hecht-Nielsen: Theory of the backpropagation neural network, Proc. of the Int. Joint Conf. on Neural Networks, I, 593-611, IEEE Press, New York, June 1989.

[34] L. Lönnblad, C. Peterson, T. Rögnvaldsson: Pattern Recognition in High Energy Physics with Artificial Neural Networks - JETNET 2.0, Comp. Phys. Comm. 1 (1992) 70.

[35] R. Hecht-Nielsen: Neurocomputing, Addison Wesley P. C., Reading, Mass., 1990.

[36] P.V. Hough: Method and means for recognizing complex patterns. US Patent 3069654 (1962).

[37] R.O. Duda, P.E. Hart: Use of the Hough-Transformation to detect Lines and Curves in Pictures, Comm. ACM, Vol. 15, No. 1, Jan. 1972, p. 11.

[38] C. Kimme, D. Ballard, J. Sklansky: Finding Circles by an Array of Accumulators, Comm. ACM, Vol. 18, No. 2, Feb. 1975, p. 120.

[39] B. Müller, J. Reinhardt: Neural Networks, An Introduction, Springer-Verlag, Berlin Heidelberg 1990.

[40] R. Bracewell: The Fourier Transform and its Applications. McGraw Hill, New York 1965.

[41] A. Drees: Private Mitteilung, 1994.

[42] R.C. Singleton: An Algorithm for Computing the Mixed Radix FFT, IEEE Trans. Audio Electroacoust., Vol. Au-1, pp. 93-107 (1969).

[43] DynaMind Version 3.0, NeuroLink Version 2.0: NeuroDynamX, Inc., PO Box 323, Boulder, Colorado 80306.

[44] L. Lönnblad, C. Peterson, T. Rögnvaldsson: JETNET 3.0 - A Versatile Artificial Neural Network Package, Lund University (Sweden) Preprint, LU TP 93-29, 1993.

[45] Intel 80170NX Electrically Trainable Analog Neural Network, Intel Corporation, Santa Clara, California.

Danksagung

Mein besonderer Dank gilt

Herrn Prof. Dr. H.J. Specht für die freundliche Aufnahme in die Heidelberger Forschungsgruppe der CERES-Kollaboration,

Herrn Prof. Dr. P. Glässel für die Überlassung dieses hochaktuellen und faszinierenden Themas sowie die freien Gestaltungsmöglichkeiten,

Herrn Prof. Dr. J. Bille, der sich als Zweitgutachter zur Verfügung gestellt hat,

den Herren Doktores A. Drees, A. Pfeiffer und T. Ullrich für viele Hinweise, Anregungen und kritische Diskussionen.

Erklärung:

Ich versichere, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 21. Juli 1994 Lutz Christian Marquard.