

Klassifikation kosmischer Strahlung anhand der Rohdaten des IceCube Arrays mittels neuronaler Netze

Bachelorarbeit aus der Physik

vorgelegt von

Johannes Bartl

Tag der Abgabe: 21. Oktober 2019

Friedrich-Alexander-Universität Erlangen-Nürnberg



1. Gutachterin: Prof. Dr. Gisela Anton
2. Gutachter: Dr. Thorsten Glüsenkamp

Inhaltsverzeichnis

1	Einleitung	3
2	Theoretische Grundlagen	5
2.1	IceCube	5
2.2	Kosmische Strahlung	6
2.2.1	Heitler-Matthews Modell	8
2.2.2	Superpositions-Modell	9
2.2.3	Geladene Teilchen im Eis	10
2.2.4	Simulationen: Corsika und MMC	10
2.3	Neuronale Netze	12
2.3.1	Layer	14
2.3.2	Softmax-Funktion	20
2.3.3	Categorical Cross Entropy	20
2.3.4	Metriken zur Evaluation	21
3	Motivation für die Arbeit	24
4	Datensatz	25
5	Implementierung des Netzwerks	31
5.1	Input	31
5.2	Skalierung	33
5.3	CNNs	33
5.4	Anpassung der Output-Form	34
5.5	LSTM	34
5.6	Softmax	34
6	Ergebnisse	35
6.1	Trainingsprozess	35
6.2	Evaluation der ungewichteten Verteilung	37
6.3	Evaluation der gewichteten Verteilung	39
6.4	Physikalische Eigenschaften der Klassifikation	39
7	Zusammenfassung und Ausblick	45

1 Einleitung

Im Jahr 1912 stieg der österreichische Physiker Viktor Hess insgesamt sieben mal mit einem Ballon auf [10]. Er wollte nachweisen, dass eine bisher unerforschte Strahlung aus dem All durch die Atmosphäre dringt. Durch systematische Messungen konnte er einen Anstieg der ionisierenden Strahlung beobachten, je höher er flog. Außerdem konnte er durch Nachtfahrten die Sonne als Quelle ausschließen. Diese Strahlung aus dem All ist heute bekannt als kosmische Strahlung. Für diese Leistung erhielt er im Jahr 1936 den Nobelpreis. Hess' Entdeckung stellt für viele Physiker den Beginn der Astroteilchenphysik dar. Das noch sehr junge Feld der Physik ist in den letzten zwei Jahrzehnten stark gewachsen und stellt eine Überschneidung der Bereiche Astronomie, Teilchen- und Kernphysik dar.

Ein weiterer Schritt im Bereich der Astroteilchenphysik war die Fertigstellung des *Ice-Cube Neutrino Observatory*, am 18. Dezember 2010 [9]. IceCube ist ein Projekt zur Erforschung astrophysikalischer Neutrinos, ausgesendet von energiereichen Quellen im Universum. Neben den Neutrinos sind Signale der kosmischen Strahlung im Detektor nachweisbar. Wenn die kosmischen Strahlen mit den Atomen und Molekülen der Atmosphäre interagieren, erzeugen sie dabei eine Vielzahl neuer Teilchen. Unter anderem entstehen Myonen, welche häufig als Pakete im Detektor sichtbar sind. Das bedeutet eine Vielzahl von Myonen trifft in kurzer Zeit aus der gleichen Raumregion ein. Dies erklärt sich dadurch, dass Teilchen bei Interaktionen mit hoher Geschwindigkeit nur geringfügig ihre Richtung ändern. Jedes Primärteilchen der kosmischen Strahlung trägt charakteristische Eigenschaften, wie die Teilchenart oder die Energie, und erzeugt damit Myonenpakete, die sich in der Anzahl der Teilchen und deren Energieverteilung unterscheiden. Diese Myonenpakete ergeben, abhängig von ihren Eigenschaften, ein unterschiedliches Signal im Detektor.

Maschinelles Lernen mittels neuronaler Netze hat in den letzten Jahren einige bemerkenswerte Erkenntnisse gebracht. Es ist ein Teilgebiet der künstlichen Intelligenz und ermöglicht es dem Computer anhand großer Datenmengen Regeln zu entwickeln und damit selbstständig Lösungen für komplexe Probleme zu finden.

Im Rahmen dieser Arbeit sollen neuronale Netze eingesetzt werden, um die Teilchenart der kosmischen Primärteilchen zu bestimmen. Dafür betrachtet man die erzeugte Signatur der Myonenpakete im Detektor. Hierzu wird zunächst im Unterabschnitt 2.1 in kurzem Umfang der Aufbau des IceCube Detektors und dessen Funktionsweise beschrieben. Danach, im Unterabschnitt 2.2, wird näher auf die kosmische Strahlung eingegangen. Dabei wird vor allem die Zusammensetzung und die Entwicklung innerhalb der Atmosphäre betrachtet, aber auch die Interaktionen der Teilchen im Eis. Im folgenden Teil, in Unterabschnitt 2.3, wird dann auf neuronale Netze eingegangen. Dabei wird der Aufbau und die Funktionsweise aller im Rahmen dieser Arbeit verwendeten Komponenten erklärt. In Abschnitt 3 wird das Ziel der Arbeit näher beschrieben. Darauf folgend in Abschnitt 4 wird der verwendete Datensatz hinsichtlich der Teilchenart, der Energie- und Winkelverteilung und der Verteilung der Energie innerhalb eines Myonenpakets untersucht. Anschließend, in Abschnitt 5 wird das verwendete neuronale Netz diskutiert und es wird näher auf die gewählten Parameter eingegangen. Im

vorletzten Kapitel 6 werden die Ergebnisse präsentiert und diskutiert. Abschließend wird in Abschnitt 7 ein Fazit gezogen und ein Ausblick auf eine mögliche Fortsetzung der Arbeit gegeben.

2 Theoretische Grundlagen

2.1 IceCube

Das *IceCube Neutrino Observatory* (IceCube) ist ein Teilchendetektor, welcher dazu entwickelt wurde um astrophysikalische Neutrinos nachzuweisen. Neutrinos sind subatomare, elektrisch neutrale und fast masselose Teilchen. Sie interagieren nur gravitativ und über die schwache Wechselwirkung, was zu einer sehr geringen Wechselwirkungsrate führt. Aus diesem Grund ist es nötig einen Detektor zu entwickeln, der ein großes Volumen abdeckt. Dieser Detektor befindet sich am Südpol nahe der Amundsen-Scott Station und umfasst ein Kubikkilometer Eis. In Abbildung 1 ist eine grafische Darstellung des Detektors zu sehen. Er besteht aus 86 vertikal angebrachten Stahlkabeln, die als *Strings* bezeichnet werden. Diese sind ins Eis eingelassen und erstrecken sich bis hin zu einer Tiefe von 2450 m. An jedem vertikalen String sind 60 *digitale optische Module* (DOMs) befestigt, die vom Eis vollständig umschlossen werden. Die Module sind in einer Tiefe von 1450 m bis 2450 m mit dem vertikalen Abstand von 17 m angebracht. Die Anordnung aller Strings entspricht von oben betrachtet einer hexagonalen Form, wobei die einzelnen Strings circa 125 m auseinander liegen. Acht dieser 86 Strings, welche sich im Zentrum des Detektors befinden, formen den Subdetektor DeepCore (vgl. Abbildung 1). Mit einem vertikalen Abstand von 7 m und einem horizontalen Abstand von 70 m ist die Anordnung von DeepCore kompakter. Der Subdetektor ermöglicht die Erforschung der Neutrinooszillation. Insgesamt sind im Detektor 5160 DOMs verbaut. Jeder davon enthält ein Photo-Multiplier-Tube (PMT) mit einem Durchmesser von 25.4 cm und ist in der Lage, optische Signale, wie Photonen, in elektrische Ladung und damit in ein elektrisches Signal umzuwandeln. Dabei ist die Stärke des Signals proportional zum Eingangssignal [19].

Wenn Neutrinos mit den Atomen im Eis interagieren, erzeugen sie geladene Sekundärteilchen. Eis ist ein isolierendes, dielektrisches Medium. Bewegt sich nun eines dieser geladenen Teilchen durch Eis mit einer Geschwindigkeit, die größer ist als die Lichtgeschwindigkeit in eben diesem Medium, so geben sie Cherenkov-Strahlung ab (vgl. Abbildung 2). Diese Strahlung wird in einem Kegel um die Teilchenspur emittiert, wobei der Öffnungswinkel θ dieses Kegels direkt mit der Geschwindigkeit des Teilchens $\beta = \frac{v}{c}$ und dem Brechungsindex n des Mediums zusammenhängt.

$$\cos(\theta) = \frac{1}{n\beta} \quad (1)$$

Durch die Messung des Cherenkov-Lichts mit Hilfe der optischen Module lässt sich die Richtung und Energie der erzeugten Sekundärteilchen und des Neutrinos rekonstruieren. Geladene Teilchen, darunter Myonen, können allerdings nicht nur durch Neutrinointeraktionen produziert werden. Auch ein Luftschauer, ausgelöst durch kosmische Strahlung, enthält Myonen, die in IceCube detektiert werden.

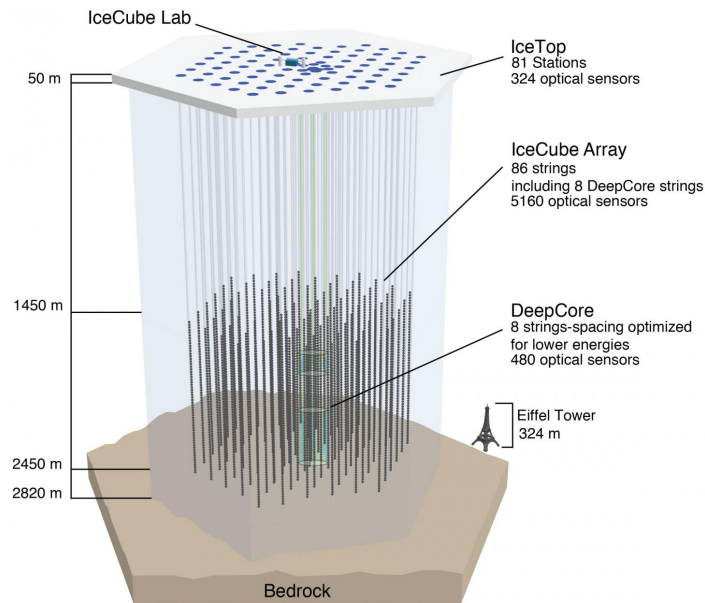


Abbildung 1: Grafische Darstellung des IceCube Detektors. Zu sehen sind die 86 vertikal ins Eis eingelassenen Strings inklusive der daran angebrachten DOMs. Hier bezeichnet als IceCube Array. Außerdem zu sehen ist der Subdetektor DeepCore [5].

2.2 Kosmische Strahlung

Die Kosmische Strahlung besteht aus hochenergetischen Teilchen, die sich annähernd mit Lichtgeschwindigkeit durch das Weltall bewegen. Die Teilchen sind ionisierte Atomkerne, wobei 90% davon Wasserstoff und 9% Helium sind. Aber auch schwerere Kerne, unter anderem Eisen, wurden gemessen. Diese Teilchen der kosmischen Strahlung werden Primärteilchen genannt.

Wenn eines der Primärteilchen in die Erdatmosphäre eintritt, kollidiert es mit einem Atom oder Molekül aus der Luft (hauptsächlich N_2 , O_2 und Ar) in einer typischen Höhe von 15-35 km. Dabei werden neue Teilchen, sogenannte Sekundärteilchen, erzeugt, die weiter kollidieren können. Dies löst eine Kettenreaktion aus, die zur Folge hat, dass sich die Zahl der Teilchen rapide erhöht. Dieses Phänomen wird als Luftschauer bezeichnet [15, Kap.1, S.1].

Jeder Luftschauer, der durch ein Teilchen der kosmischen Strahlung ausgelöst wird, hat eine elektromagnetische, eine myonische und eine hadronische Komponente. Zu der elektromagnetischen Komponente zählen Photonen, Elektronen und Positronen. Die Myonische enthält Myonen und Myon-Neutrinos (vgl. Abbildung 3).

Bei der Interaktion des Primärteilchens mit einem Luftteilchen entstehen geladene und neutrale Pionen und Kaonen, außerdem Baryonen, wie Protonen und Neutronen. Die langlebigen Teilchen davon, dazu zählen die geladenen Pionen und Kaonen sowie die

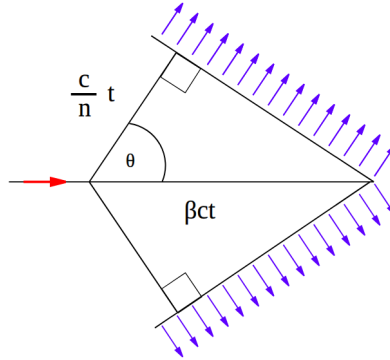


Abbildung 2: Abbildung des Cherenkov-Lichtkegels. $\beta c t$ ist die zurückgelegte Strecke des Teilchens in der Zeit t . Der rote Pfeil gibt die Richtung des Teilchens an. $\frac{c}{n} t$ ist die Strecke, die das Licht zurückgelegt hat, welches zum Zeitpunkt 0, hier ist das der Schnittpunkt der beiden Strecken, emittiert wurde. Dadurch lässt sich der Öffnungswinkel θ des Lichtkegels, dargestellt durch blau Pfeile, berechnen.[25]

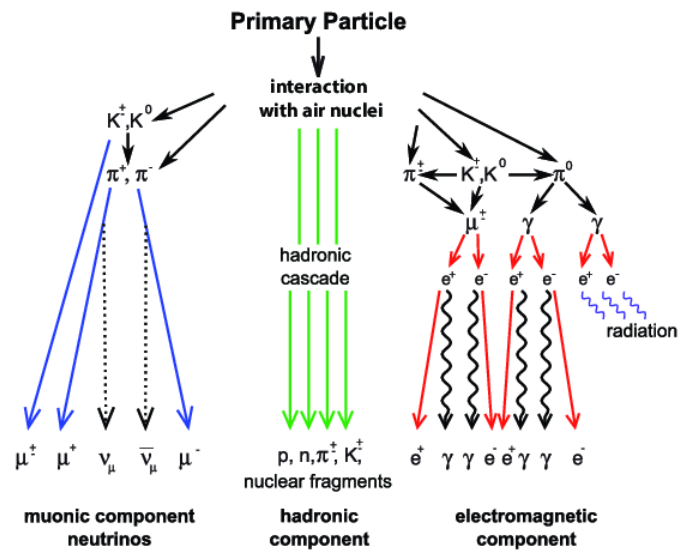


Abbildung 3: Schematischer Überblick über die Sekundärteilchen eines Luftschauers. Das 3-Komponenten-Modell dient der Übersichtlichkeit, wobei hier jede Komponente in einer anderen Farbe dargestellt ist [1].

stabilen Baryonen, also Protonen und Neutronen, bilden die hadronische Komponente und damit den Kern des Schauers. Diese Teilchen können erneut mit einem Luftteilchen interagieren, und damit weitere Teilchenschauer auslösen, oder zerfallen. Dies ist abhängig von ihrer Energie und ihrer Halbwertszeit. Die geladenen Pionen zerfallen hauptsächlich in ein Myon und ein Neutrino ($\pi^\pm \rightarrow \mu^\pm + \nu_\mu \setminus \bar{\nu}_\mu$). Die geladenen Kaonen haben zwei Hauptzerfallskanäle. Der Erste ist der Zerfall in ein Myon und ein Neutrino ($K^\pm \rightarrow \mu^\pm + \nu_\mu \setminus \bar{\nu}_\mu$) und der Zweite in ein geladenes und ein ungeladenes Pion ($K^\pm \rightarrow \pi^\pm \pi^0$).

Zu den kurzlebigen Teilchen, die bei der Interaktion des Primärteilchens entstehen, zählen neutrale Pionen und Kaonen. Die neutralen Pionen zerfallen sofort in zwei Photonen ($\pi_0 \rightarrow 2\gamma$). Die neutralen Kaonen haben auch zwei Hauptzerfallskanäle. Der Erste ist der Zerfall in zwei neutrale Pionen ($K^0 \rightarrow \pi^0 \pi^0$) und der Zweite in ein geladenes und ein neutrales Pion ($K^0 \rightarrow \pi^+ \pi^-$).

Die in allen eben beschriebenen Prozessen entstandenen Myonen gelangen zum größten Teil mit geringen Energieverlusten durch die Atmosphäre. Ein geringer Teil davon zerfällt weiter ($\mu^- \rightarrow e^- \bar{\nu}_e \nu_\mu$ oder $\mu^+ \rightarrow e^+ \nu_e \bar{\nu}_\mu$). Jedes entstandene Photon kann durch die Paarbildung weitere Subschauer auslösen [11].

2.2.1 Heitler-Matthews Modell

Das Heitler-Matthews Modell beschreibt eine vereinfachte Struktur der Luftschauer, woraus sich eine gute Annäherung an die Realität ergibt. Es wird vereinfacht angenommen, dass sich alle Teilchen gleich verhalten und jeweils nach einer festen Strecke λ_{int} , die sie zurückgelegt haben, interagieren. In Abbildung 4 ist die Struktur eines Luftschauers nach diesem Modell grafisch dargestellt. Es wird angenommen, dass bei der Interaktion eines Hadrons mit der Energie E n_{tot} neue Teilchen entstehen. Die Energie E wird dabei gleichmäßig auf die neu entstandenen Teilchen verteilt ($\frac{E}{n_{tot}}$). Zwei Drittel der neuen Teilchen sind geladene Pionen (n_{ch}). Ein Drittel sind ungeladene Pionen (n_{neut}). Die neutralen Pionen zerfallen, wie oben beschrieben, in zwei Photonen und bilden damit den elektromagnetischen Teil. Die geladenen Pionen dagegen interagieren erneut, nachdem sie eine gewisse Interaktionslänge λ_{in} (vgl. Abbildung 4) durchlaufen haben und falls ihre Energie dann noch größer als E_{dec} ist. E_{dec} ist eine Zerfallsenergie. Für den Fall, dass die Energie kleiner als E_{dec} ist, zerfallen die geladenen Pionen in je ein Myon. Kaonen werden im Modell vernachlässigt, da sie entweder direkt in ein Myon oder über ein Pion in ein Myon bzw. Photonen zerfallen und somit ein ähnliches Verhalten wie Pionen zeigen. Die Zahl der Myonen N_μ eines Luftschauers wird durch Gleichung 2 beschrieben und hängt bei diesem Modell von der Primärenergie, also der Energie des Primärteilchens, der Luftdichte (enthalten in E_{dec}) und der geladenen und totalen Multiplizität von hadronischen Interaktionen (enthalten in α) ab.

$$N_\mu = \left(\frac{E_0}{E_{dec}} \right)^\alpha \quad \text{mit} \quad \alpha = \frac{\ln n_{ch}}{\ln n_{tot}} \approx 0.82 \dots 0.94 \quad (2)$$

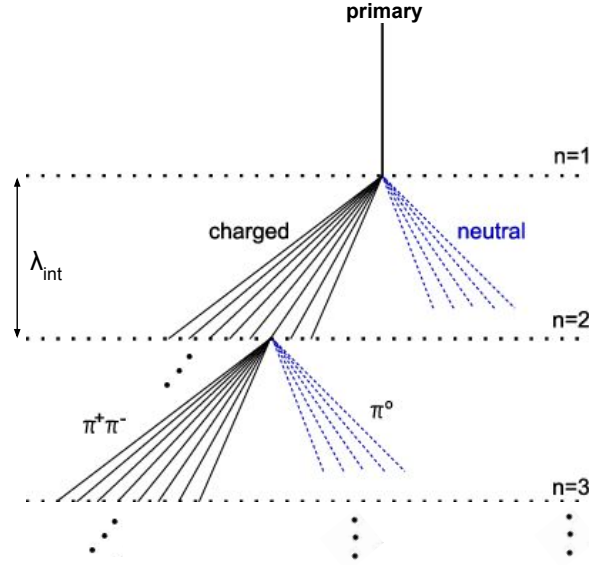


Abbildung 4: Das Diagramm zeigt die Struktur eines Luftschauers, ausgelöst von einem Proton, nach dem Heitler-Matthews Modell. Die geladenen Teilchen, hier hauptsächlich Pionen, sind in schwarz dargestellt und als *charged* bezeichnet, alle neutralen Teilchen in blau [15, Kap.16.2, S.315].

Hier bezeichnet E_0 die Primärenergie. Die Parameter n_{ch}, n_{tot}, E_{dec} müssen mit detaillierten Simulationen bestimmt werden [15, Kap.16.2, S.315].

2.2.2 Superpositions-Modell

Eine Erweiterung des Heitler-Matthews Modells ist das Superpositions-Modell [15, Kap.16.4, S.320]. Da die Bindungsenergie in einem Atomkern pro Nukleon mit ≈ 5 MeV deutlich kleiner ist als die typische Interaktionsenergie eines Primärteilchens mit der Luft, kann man einen Kern mit der Masse A als A unabhängige Primärteilchen betrachten. Dementsprechend trägt jedes der A Primärteilchen die Energie $E_h = E_0/A$. Daraus ergibt sich nach dem Heitler-Matthews-Modell folgende Vorhersage:

$$N_{\mu}^A(E_0) = A \cdot \left(\frac{E_0/A}{E_{dec}} \right)^{\alpha} \text{ mit } \alpha < 1 \quad (3)$$

Aus Gleichung 3 ergibt sich, dass je schwerer das Primärteilchen ist, welches den Luftschauer auslöst, desto mehr Myonen sind für eine bestimmte Energie zu erwarten. Luftschauer, ausgelöst durch einen Eisen-Kern, enthalten circa 40% mehr Myonen als Proton-Schauer mit der gleichen Primärenergie E_0 .

Aus Gleichung 3 ergibt sich eine weitere Konsequenz. Je größer die Masse des Primärteilchens ist, desto geringer ist die Energie pro Myon, ebenfalls betrachtet bei gleicher Primärenergie. Dies folgt, da die gleiche Gesamtenergie auf mehr Teilchen verteilt wird. In der Realität wird die Energie bei jeder Interaktion nicht exakt gleichmäßig auf alle entstehenden Teilchen aufgeteilt. Dennoch ist zu erwarten, dass die Energie pro Myon bei einem Eisen-Schauer niedriger ist als bei einem Proton-Schauer mit gleicher Primärenergie.

2.2.3 Geladene Teilchen im Eis

Bisher wurde die Entwicklung der kosmischen Strahlung und der Luftschaue in der Atmosphäre, also in Luft, betrachtet. In diesem Kapitel geht es um die Wechselwirkungen geladener Teilchen in Materie. Das ist relevant, da der Detektor von Eis umgeben ist. IceCube kann, wie in Unterabschnitt 2.1 erwähnt, nur geladene Teilchen nachweisen, die sich mit sehr hoher Geschwindigkeit bewegen. Betrachtet werden hier Myonen, da alle weiteren geladenen Teilchen des Schauers, vor allem Elektronen, nahe der Eisoberfläche absorbiert werden.

In Abbildung 5 ist der mittlere Energieverlust von positiv geladenen Myonen in Kupfer über die Energie des Teilchens aufgetragen. Anzumerken ist, dass die Werte bei dem Durchgang geladener Myonen durch Eis abweichen. Dennoch ist der qualitative Verlauf des Energieverlustes gleich, weshalb es sinnvoll ist diesen, vor allem für sehr hochenergetische Myonen, wie sie in Luftschauern auftreten, zu betrachten. Bei sehr hohen Energien werden vor allem die Strahlungsverluste dominant gegenüber den Ionisationsverlusten. Die Strahlungsverluste steigen linear mit der Energie der Teilchen an. Zu den Verlusten zählt Paarproduktion, Bremsstrahlung und photonukleare Beiträge [7], wobei die Summe der drei Effekte vor allem im hohen Energiebereich, starken stochastischen Schwankungen [4, Kap.33.6, S.216] unterliegt.

2.2.4 Simulationen: Corsika und MMC

Im Folgenden werden zwei Programme eingeführt, welche die kosmische Strahlung von der Entstehung bis hin zur Detektion in IceCube simulieren. Die Simulation von Luftschauern wird von CORSIKA übernommen.

COSmic Ray SIMulations for KAscade (CORSIKA) [2] ist ein Programm für detaillierte Monte-Carlo Simulationen von Luftschauern, ausgelöst durch hochenergetische kosmische Strahlen. Dabei können Protonen, Photonen und weitere Teilchen als Primärteilchen simuliert werden. Die Teilchen werden durch die Atmosphäre verfolgt, bis sie interagieren oder zerfallen.

Um diese Luftschaue möglichst realitätsnah zu simulieren, werden viele bekannte Einflüsse auf die Teilchen berücksichtigt. Dies betrifft sowohl den Transport der Teilchen durch die Atmosphäre, als auch die Interaktionen mit Luft. Die berücksichtigten Effekte zu beschreiben, würde den Rahmen der Arbeit überschreiten. Alle Sekundärteilchen werden entlang ihrer Laufbahn verfolgt und die zugehörigen Parameter gespeichert. CORSIKA simuliert bis zur Eisoberfläche. Bei der verwendeten Simulation werden nicht alle Elemente von Wasserstoff bis Eisen genutzt. Stattdessen werden fünf Pri-

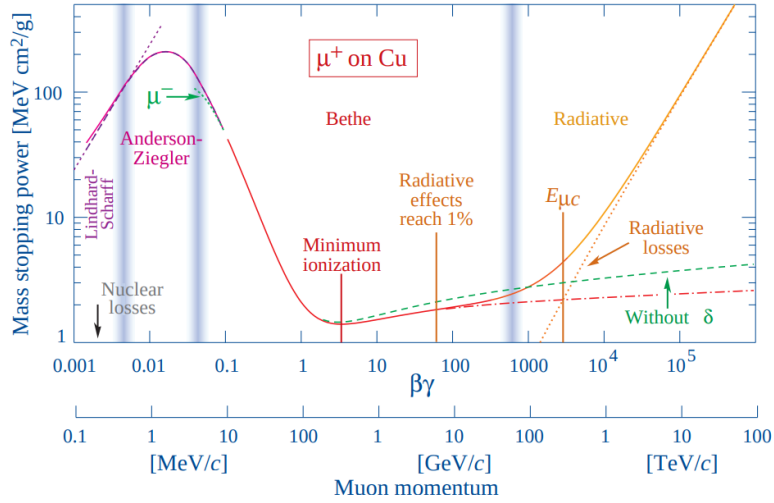


Abbildung 5: Energieverlust für positive Myonen in Kupfer aufgetragen über die Energie des Teilchens. Im unteren Energiebereich wird der Energieverlust durch Stoßprozesse am besten durch das Modell Anderson-Ziegler beschrieben. Danach treten Ionisationsverluste auf, welche gut mit Bethe-Bloch beschrieben werden. Bei höheren Energien werden die Strahlungsverluste, hier als *Radiative Losses* bezeichnet, dominant. Die gelbe Kurve beschreibt die Summe aller in diesem Bereich auftretenden Einzeleffekte [4, Kap.33.2, S.213].

märteilchen simuliert, die repräsentativ für alle vorkommenden Elemente sind. Diese sind Wasserstoff und Helium für die leichten Elemente, Stickstoff und Aluminium für die Schwereren und Eisen für die Schwersten. Die Verteilung der simulierten Primärteilchen entspricht nicht der Realität. Deshalb wird die Verteilung entsprechend dem GaisserH3a [14] Modell gewichtet. Dadurch erhält man eine Rate, die angibt wie häufig der Schauer eines Primärteilchens im Detektor nachgewiesen wird.

Das Programm *Muon Monte Carlo* (MMC) [7] simuliert die Energieverluste der Myonen, die sich von der Eisoberfläche durch den Detektor ausbreiten. Die Simulation, der von den Myonen ausgesendeten Cherenkov-Strahlung, bis hin zur Detektion durch die PMTs wird von anderen Modulen übernommen, auf die hier nicht näher eingegangen wird. Auch die berücksichtigten Effekte von MMC werden hier nicht betrachtet. Gespeichert werden nur die Simulationen, die für IceCube relevant sind, also nur falls Teilchen des Luftschauers im Detektor nachgewiesen werden. Dies ist nicht bei allen simulierten Primärteilchen der Fall, da die Schauer den Detektor verfehlen oder frühzeitig vollständig im Eis absorbiert werden könnten.

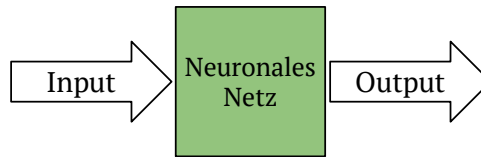


Abbildung 6: Grafische Darstellung der Grundidee eines Neuronales Netzes. Das Netz erhält einen Input, verarbeitet diese Information und gibt einen Output.

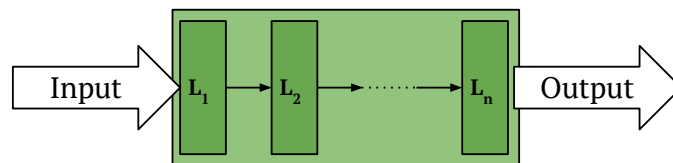


Abbildung 7: Skizze der inneren Struktur eines neuronalen Netzes. Dieses besteht aus aneinandergereihten Layern L_1, L_2, \dots, L_n .

2.3 Neuronale Netze

Künstliche neuronale Netze sind ein Teilgebiet der künstlichen Intelligenz. Im Aufbau und der Funktionsweise sind sie vergleichbar mit einem natürlichen neuronalen Netz. Diese natürlichen neuronalen Netze bilden die Struktur und Informationsarchitektur des menschlichen Gehirns. Die Nervenzellen, auch als Neuronen bezeichnet, sind über Synapsen verbunden und geben elektrische Signale weiter. Eine Vielzahl dieser simplen Strukturen bilden ein neuronales Netz, welches komplexe Informationen verarbeiten kann. Die Terminologie entspringt der Biologie und wurde entsprechend auch auf künstliche neuronale Netze übertragen. Der Aufbau eines künstlichen neuronalen Netzes lässt sich stark vereinfacht darstellen, verdeutlicht aber die Grundidee sehr gut (vgl. Abbildung 6). Ein neuronales Netz bekommt eine Eingangsinformation, auch Input genannt, verarbeitet die Information und gibt diese dann in Form des Outputs wieder aus. Der Input kann beispielsweise, wie in dieser Arbeit, ein Detektorsignal sein und der zugehörige Output eine Wahrscheinlichkeitsverteilung, die angibt, welches Primärteilchen das Signal im Detektor verursacht hat. Der innere Aufbau eines Netzwerks ist in Abbildung 7 dargestellt. Das Netzwerk besteht aus Schichten, sogenannten Layern. Ein Layer besteht seinerseits aus Neuronen. In Abbildung 8 ist dies grafisch dargestellt. Im Neuron werden die Eingabewerte x_j mit w_j gewichtet und ein Offset b addiert. Darauf wird dann eine nichtlineare Funktion a , häufig als Aktivie-

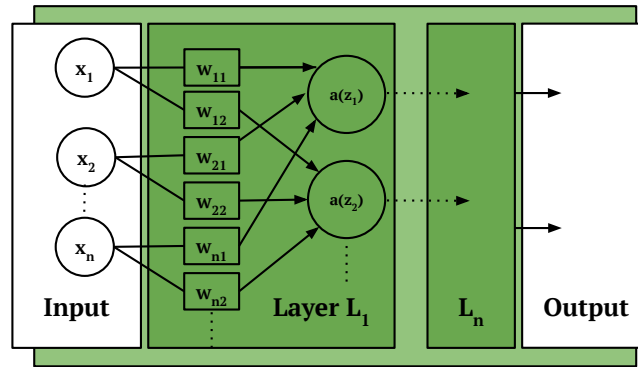


Abbildung 8: Einzelne Neuronen innerhalb der Layer, dargestellt als Kreise. x_j ist der Input. Die Verbindungen zwischen den Neuronen tragen Gewichte w_{ij} . Hier ist a die Aktivierungsfunktion.

rungsfunktion bezeichnet, angewendet. Der Ausgabewert eines Neurons $f(x)$ ist folglich gegeben durch:

$$f(x) = a(z) = a \left(\sum_j w_j x_j + b \right) \quad (4)$$

Die interne Funktionsweise eines Layers unterscheidet sich je nach Typ und verarbeitet die Information demnach entsprechend (vgl. Abschnitt 2.3.1).

Neuronale Netze können durch einen Trainingsprozess gewisse Regeln oder Strukturen erlernen. Dabei hat man Input-Werte und kennt deren Zielwerte. Dann vergleicht man den Output des Netzwerks mit den Zielwerten und misst die Abweichung. Die Parameter im Netzwerk (w, b) werden dann entsprechend angepasst um die Abweichung zu minimieren. Der Algorithmus für die Anpassung nennt sich *Backpropagation* [16, Kap.6.5], wobei hier nicht näher darauf eingegangen wird. Der Trainingsprozess wird in Abbildung 9 verdeutlicht. Das Netz, das aus miteinander verketteten Layern besteht, ordnet den Eingabedaten, Vorhersagen Y' zu. Die Verlustfunktion vergleicht diese Vorhersagen mit den tatsächlichen Zielwerten Y und berechnet einen Verlust, der beschreibt, wie gut die Vorhersagen Y' des neuronalen Netzes mit den erwarteten Zielwerten Y übereinstimmen. Der Optimierer passt dann entsprechend die Gewichte w in den Layern an. Ein mehrfaches Durchlaufen dieser Schleife kann den Verlust minimieren und die Vorhersagen verbessern. Das Training wird üblicherweise auf einem sehr großen Datensatz durchgeführt. Ein kompletter Durchlauf aller Input-Daten wird als Epoche bezeichnet. Dabei können die Input-Daten auch in gleich große Gruppen, sogenannte *Batches*, eingeteilt werden. Die Gewichte werden nach jedem durchlaufenen Batch angepasst, wodurch das Training beschleunigt wird [8]. Die Anpassung der Gewichte erfolgt schrittweise, wobei die sogenannte Lernrate angibt wie groß diese Schritte sind [16, Kap.4.3].

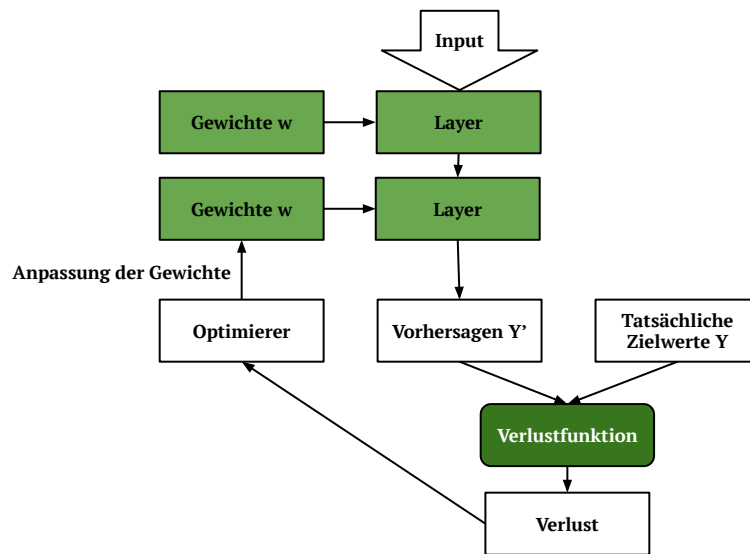


Abbildung 9: Trainingsprozess eines künstlichen neuronalen Netzes. Jedem Input wird eine Vorhersage Y' zugeordnet, die mit dem tatsächlichen Zielwert Y mittels Verlustfunktion verglichen wird. Daraus ergibt sich der Verlust, der die Abweichung beschreibt. Der Optimierer passt dann entsprechend die Gewichte w an, um den Verlust zu minimieren [8, Kap.3.1, S.86].

Bei neuronalen Netzen werden häufig zwei Datensätze verwendet. Das erste ist das Trainingsset. Dieses wird, wie der Name sagt, zum Trainieren des neuronalen Netzes verwendet. Der zweite Datensatz ist das Validierungsset, wobei die Daten repräsentativ für das Trainingsset sein sollten. Auf diesem Datensatz werden vom Netzwerk ebenfalls Vorhersagen erstellt, wobei diese Daten nicht beim Training verwendet werden. Daran lässt sich erkennen, wie gut es auf neuen, bisher unbekannten Daten arbeitet.

2.3.1 Layer

Convolutional Neural Network

Der Begriff *Convolutional Neuronal Network* (CNN) [6] impliziert, dass eine mathematische Operation, die Faltung, auf den Input dieses Layers angewandt wird [16, Kap.9.1]. Prinzipiell lässt sich diese Faltungsoperation auf Inputs mit beliebiger Dimension anwenden.

In dieser Arbeit werden 1D-CNNs verwendet. Dabei ist der Input zweidimensional, darstellbar durch eine $m \times n$ Matrix (vgl. Abbildung 10). Diese Matrix wird von Filtern, häufig als Kernel bezeichnet, analysiert. Dabei legt man eine feste Anzahl n_{Filter} der Filter fest. In Abbildung 10 ist einer der Kernel dargestellt. Die Kernel haben eine

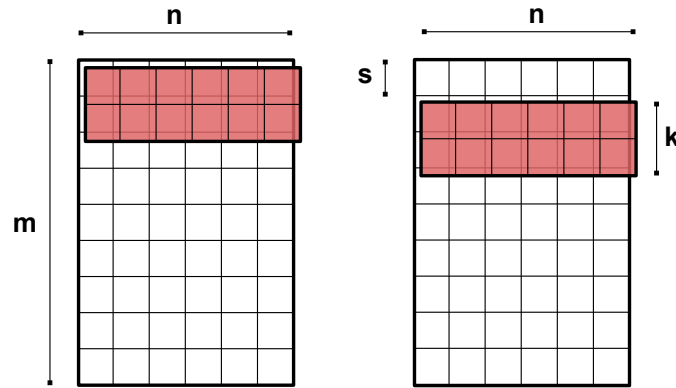


Abbildung 10: Darstellung der Input-Matrix in weiß, und des Kernels in rot, mit den verwendeten Bezeichnungen. Die beiden Matrizen stellen zwei Zeitschritte dar, in denen der Kernel über die Input-Matrix rastert

festen Pixelgröße k , die dann mit einer konstanten Schrittweite s über die Pixel-Matrix des Inputs rastern. Die Ausbreitung des Kernels erfolgt in eine festgelegte Richtung (1D). Da diese beliebig wählbar ist, wird im Folgenden oBdA. die Ausbreitung entlang der m -Achse gewählt. Die Größe des Kernels ist dementsprechend definiert als $k \times n$, wobei k ebenfalls beliebig wählbar ist. Jeder Kernel lässt sich als Matrix K darstellen. Dabei enthält jeder Eintrag ein festes Gewicht. Für jeden Ausbreitungsschritt des Kernels wird nun die Faltungsoperation angewandt und zwar auf zwei Matrizen mit gleicher Dimension. Dieser Prozess ist in Abbildung 11 verdeutlicht. Die erste Matrix ist eine Teilmatrix des Inputs und zwar genau der Teil, welcher vom Kernel „verdeckt“ wird. Die zweite Matrix ist der Kernel selbst. Dabei werden die Einträge der beiden Matrizen elementweise miteinander multipliziert und am Ende aufsummiert. Jede Summe bildet einen Eintrag in der Output-Matrix O des Layers. Auch hier kann eine Aktivierungsfunktion a verwendet werden. Dabei wird auf jede Summe eine nichtlineare Funktion angewandt. Die Matrix O ist in Abbildung 12 dargestellt. Für jeden der n_{Filter} Kernel erhält die Output-Matrix eine weitere Spalte, hier verdeutlicht durch die unterschiedlichen Farben. Die Größe der Output-Matrix ist nicht nur von der Zahl der verwendeten Kernel abhängig, sondern auch davon, wie sich der Kernel verhalten, wenn sie an den Rand der Input-Matrix stoßen. Das Verhalten des Kernels am Rand wird als Padding bezeichnet. Beim hier verwendeten Padding bleibt die Größe der Dimension m der Input- und der Output-Matrix identisch, unter der Annahme, dass eine Schrittgröße s von 1 gewählt wurde. Um dies zu erreichen werden gegebenenfalls Nullen an die Ränder der Matrix gehängt. Bei einer Veränderung der Schrittgröße s ändert sich demnach trotzdem die Größe der Output-Matrix indirekt proportional dazu. Die Output-Matrix ist ebenfalls zweidimensional und hat aufgrund des Paddings die Größe $m/s \times n_{\text{Filter}}$.

Ein CNN kann lokale Muster in der Datenstruktur erkennen und extrahieren. Der große

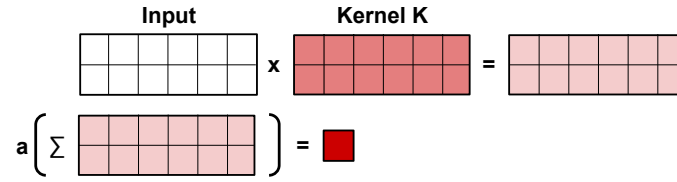


Abbildung 11: Darstellung der Faltung für einen Ausbreitungsschritt. Als Input wird hier der „verdeckte“ Teil der Input-Matrix während dieses Ausbreitungsschritts bezeichnet. Der verwendete Kernel K ist in rot dargestellt. Durch die elementweise Multiplikation erhält man eine neue Matrix mit der gleichen Dimension, hier in hellrot. Zunächst werden alle Einträge der neuen Matrix aufsummiert und dann wird darauf die Aktivierungsfunktion a angewandt.

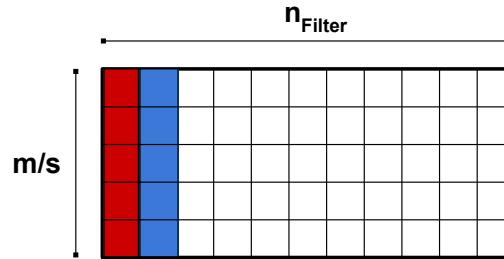


Abbildung 12: Darstellung der Output-Matrix O. Die Dimension davon ist $m/s \times n_{\text{Filter}}$. Jede Farbe symbolisiert dabei den Output eines der n_{Filter} Kernel.

Vorteil dabei ist die Translationsinvarianz der erlernten Muster. Das bedeutet, dass ein lokal vom Netzwerk erlerntes Muster, eine Information über die Struktur der Daten, global erkannt werden kann. Dadurch haben sich die CNNs vor allem im Bereich der Bildklassifikation durchgesetzt.

Long Short-Term Memory

Die vom Detektor gemessenen Luftschauer innerhalb des Detektors, folglich der Input des Netzwerks, ist eine zeitliche Sequenz, wie in Unterabschnitt 5.1 beschrieben. Demnach sind die Reihenfolge und die vorhergehenden Zeitschritte relevant für die Analyse des aktuellen Zeitschritts. Ein Methode um aus dieser Art von Daten effizient Informationen zu gewinnen, ist die Daten schrittweise zu verarbeiten. Dabei kann das Netzwerk auf den bereits erhaltenen Informationen aufbauen und kontinuierlich neue Information ergänzen oder Alte ersetzen. Da dies nicht mit einem CNN-Layer möglich ist, werden im Folgenden rekurrente neuronale Netze (RNNs) eingeführt. Ein

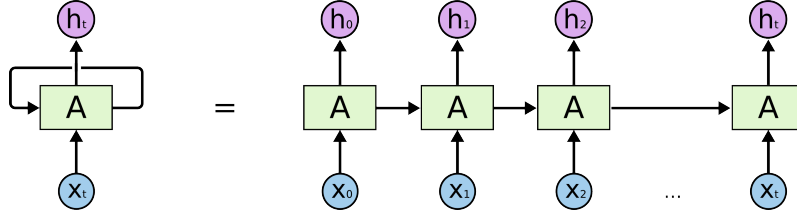


Abbildung 13: Ein rekurrentes neuronales Netz besteht aus einer Kette sich wiederholender Module. Hier bezeichnet x_t den zeitabhängigen Input, wobei jedes x_i mit $i \in [0, t]$ einen Zeitschritt darstellt. A bezeichnet das Modul, welches eine innere Struktur mit trainierbaren Parameter enthält. h_t ist der Output des LSTM-Layers und h_i mit $i \in [0, t]$ der Output jedes Moduls, welcher an das nachfolgende Modul weitergegeben wird [23].

spezieller Typ dieser Netze ist das *Long Short-Term Memory* (LSTM) [17]. Dieses bietet den Vorteil, Informationen über eine lange Zeit zu speichern und jederzeit abrufen zu können, weshalb sie heutzutage sehr weitreichend eingesetzt werden.

Der Aufbau eines RNN-Layers ist eine Kette sich wiederholender Module, wie in Abbildung 13 dargestellt. In einer Kette teilen sich alle Module die gleichen Gewichte. Demnach ist die Zahl der trainierbaren Parameter auch unabhängig von der Anzahl der Zeitschritte. Der Index t beschreibt den aktuellen Zeitschritt des Moduls. Der Input x_t ist hier die zeitliche Sequenz. In diesem konkreten Fall stellt jeder Sequenzpunkt einen Zeitschritt dar. Der Output jedes Moduls wird an das nächste Modul weitergegeben. Im Folgenden wird nun schrittweise der interne Aufbau dieser Module eines LSTM-Layers beschrieben.

Der Grundbaustein ist der sogenannte cell state c_t , vergleichbar einem Speicher innerhalb des Moduls, welcher die bereits aus den Daten gewonnenen Informationen enthält. Jedes Modul hat nun die Möglichkeit Informationen hinzuzufügen oder zu löschen. Dies wird durch sogenannte *Gates* ermöglicht. Diese sind vergleichbar mit einem Tor, wobei entschieden wird, welche Informationen durchkommen und welche nicht. Jedes Gate ist ein eigenes neuronales Netz mit trainierbaren Parametern. Die Zahl der Neuronen in diesem Netzwerk ist wählbar und stellt gleichzeitig auch die Dimension des Outputs da. Die gesamte innere Struktur eines Moduls ist in Abbildung 14 zu sehen.

Das erste Gate ist das *forget-Gate* (vgl. Abbildung 15), welches das Löschen einer Information ermöglicht. Dabei wird ein Wert f erzeugt, der später mit dem cell state c_t multipliziert wird. Der Wert f ist abhängig vom Output des vorangehenden Moduls h_{t-1} und vom aktuellen Input x_t und berechnet sich entsprechend Gleichung 5. Hier ist σ die sogenannte *Sigmoid*-Funktion, eine nichtlineare Aktivierungsfunktion, und definiert als $\sigma(x) = \frac{e^x}{1+e^x}$. Der Wertebereich der Funktion ist $[0,1]$. Die Multiplikation mit 0 bedeutet das vollständige Löschen der Information, die

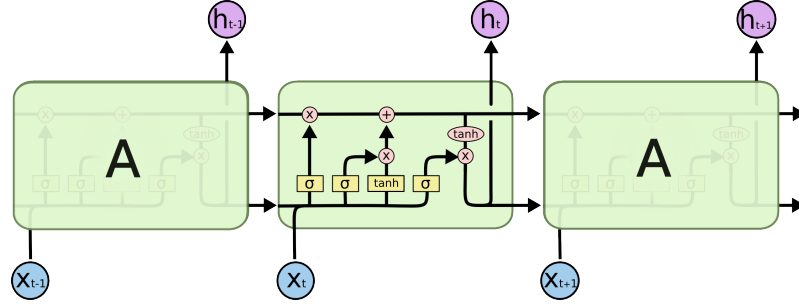
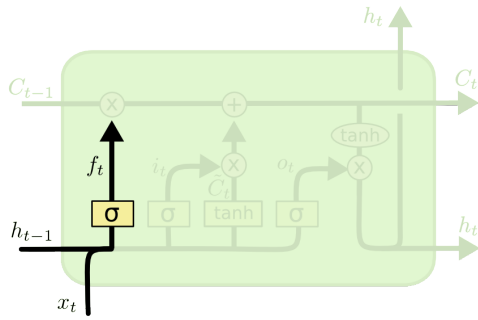


Abbildung 14: Innere Struktur eines LSTM Moduls [23].

Multiplikation mit 1 die vollständige Erhaltung. W_f und b_f sind trainierbare Gewichte.

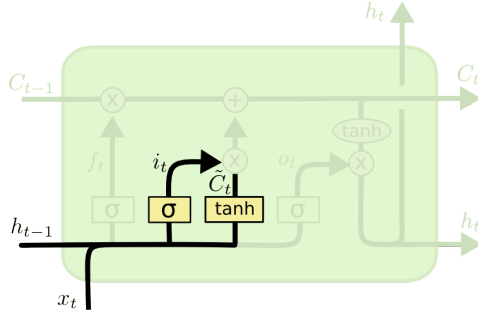


$$f = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$

Abbildung 15: Darstellung des forget-Gates innerhalb des LSTM-Moduls. x_t ist ein Zeitschritt des Inputs und h_{t-1} der Output des vorherigen Moduls, welcher hier einen zusätzlichen Input darstellt. σ ist die Aktivierungsfunktion und f_t der Output des forget-Gates [23].

Das darauf folgende *input-Gate* bestimmt, welche neue Information im cell state c_t gespeichert wird. Dieser Prozess besteht aus zwei Teilen. Zuerst wird entschieden, welche Werte aktualisiert werden (vgl. Gleichung 6). Danach wird ein Vektor mit neuen möglichen Werten erzeugt, welche zum aktuellen cell state c_t addiert werden

können (vgl. Gleichung 7). Dieser Prozess ist in Abbildung 16 dargestellt.

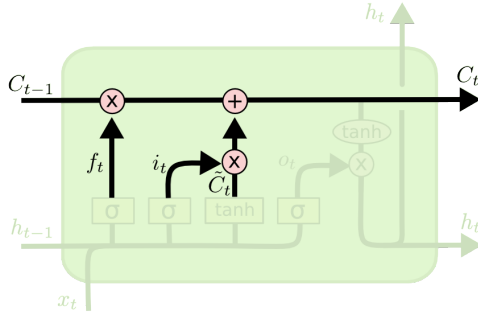


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (7)$$

Abbildung 16: Darstellung des input-Gates innerhalb des LSTM-Moduls. Der Input ist auch hier x_t und h_{t-1} . Für die Berechnung von i_t wird ebenfalls die Sigmoid-Funktion verwendet. Zu Erzeugung des Vektor \tilde{c}_t mit neuen Werte der Tangens hyperbolicus [23].

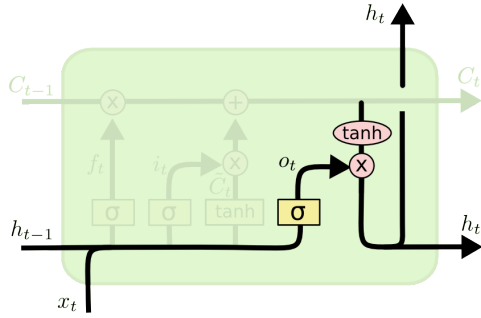
Kombiniert man beide Gates, ergibt sich Gleichung 8 und die entsprechende grafische Darstellung in Abbildung 17. Der cell state c_t wird hier aktualisiert durch die Multiplikation mit f_t . Danach wird ein neuer Vektor berechnet, durch $i_t \cdot \tilde{c}_t$, und addiert. Der neue cell state c_t wird an das folgende Modul weitergegeben.



$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (8)$$

Abbildung 17: Kombination des forget-Gates und des input-Gates. Dabei wird der cell state c aktualisiert durch die Multiplikation mit f und die Addition des erzeugten Vektors mit neuen Werte $i_t \cdot \tilde{c}_t$. Der neue cell state c_t wird an das nächste Modul weitergegeben [23].

Zuletzt entscheidet das *output-Gate* welcher Wert h_t an das nächste Modul weitergegeben wird. Dieser ist abhängig vom neu berechneten cell state c_t , von x_t und vom alten Wert h_{t-1} . Die Berechnung wird in Gleichung 9 und Gleichung 10 durchgeführt. Außerdem ist das output-Gate in Abbildung 18 dargestellt.



$$o_t = \sigma(W_0[h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (10)$$

Abbildung 18: Grafische Darstellung des Output-Gates. Hier wird der Output h_t des Moduls berechnet. Dieser ergibt sich aus h_{t-1}, x_t und dem neuen cell state c_t [23].

2.3.2 Softmax-Funktion

Die Softmax-Funktion ist eine Aktivierungsfunktion. Das bedeutet sie ist nichtlinear und monoton steigend. Wie in Unterabschnitt 2.3 beschrieben, wird sie innerhalb jedes Neurons eines Layers auf die berechneten Werte angewandt. Die Softmax-Funktion wird häufig innerhalb des letzten Layers eines neuronalen Klassifikations-Netzes verwendet. Jeder Datenpunkt in einem Datensatz ist einer Klasse zugeordnet. Hier im verwendeten Datensatz ist jedem detektierten Luftschauer das Element des auslösenden Primärteilchens zugeordnet. Eine Klassifizierung des Netzwerks bedeutet, dass es anhand eines Inputs entscheidet, welcher Klasse dieser Datenpunkt angehört. Dazu wandelt das Netzwerk die Output-Werte des vorherigen Layers in eine Wahrscheinlichkeitsverteilung um. Der neue Output ist ein Vektor. Jeder Eintrag im Vektor repräsentiert eine Klasse und enthält eine Wahrscheinlichkeit. Diese Wahrscheinlichkeit gibt Auskunft darüber, wie sicher das Netzwerk ist, dass dieser Datenpunkt dieser Klasse angehört. Die Softmax-Funktion ist definiert in Gleichung 11. y_i bezeichnet dabei jedes Element im Output-Vektor y des vorherigen Layers. Die Summe im Nenner ist für die Normierung der Wahrscheinlichkeitsverteilung auf 1 verantwortlich.

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad \text{wobei} \quad \sum_i S(y_i) = 1 \quad (11)$$

2.3.3 Categorical Cross Entropy

Das Neuronale Netz erstellt Vorhersagen, die einen Zielwert möglichst gut rekonstruieren sollen. Dafür haben neuronale Netze eine Funktion, welche die Vorhersage mit dem tatsächlichen Soll-Wert vergleicht und daraus die Abweichung, hier als Verlust bezeichnet, berechnet (vgl. Unterabschnitt 2.3). In diesem neuronalen Netz wird die durch Gleichung 12 beschriebene Categorical Cross-Entropy Verlustfunktion verwendet. Dabei ist y_i mit $i \in [1, n]$ der tatsächliche Soll-Wert. p_i ist die Vorhersage des Netzwerks in Form einer Wahrscheinlichkeitsverteilung, welche die Softmax-Funktion

ausgibt. Diese Wahrscheinlichkeitsverteilung wird für eine Klassifizierung in n verschiedene Klassen verwendet.

Die One-Hot Kodierung wird häufig bei der Klassifikation verwendet um den Soll-Wert y_i zu beschreiben. Dabei wird für jedes Trainingsbeispiel ein Vektor erzeugt. Für jede mögliche Klasse erhält dieser Vektor einen Eintrag mit 0. Nur an einer Stelle, je nachdem welcher Klasse der Input des Beispiels angehört, erhält der Vektor eine 1. Für eine Klassifikation mit n möglichen Klassen würde ein Datenpunkt, der der m -ten Klasse angehört, folgendermaßen aussehen: $(0, \dots, 0, 1, 0, \dots, 0)$ mit einer 1 an der Stelle m . Der Verlust L wird für jedes Trainingsbeispiel separat berechnet und kann dann gemittelt werden.

$$L = -w_m \sum_i y_i \ln p_i \quad (12)$$

Nicht jeder Datensatz, welches vom Netzwerk verwendet wird, ist ausgewogen in die einzelnen Klassen verteilt. Das bedeutet, es ist möglich, dass ein Großteil des Datensatzes einer Klasse angehört, wohingegen nur ein kleiner Teil einer zweiten Klasse zugehörig ist. Für diesen Fall können Klassengewichte verwendet werden. Das Gewicht lässt sich folgendermaßen für die m -te Klasse berechnen, wobei die Anzahl aller Trainingsbeispiele in der m -ten Klasse N_m ist:

$$w_m = \frac{1}{N_m} \quad (13)$$

Diese Gewichte werden bei der Berechnung des Verlusts berücksichtigt. Ohne Gewichtung trägt jedes Trainingsbeispiel das Gewicht 1. Mit einer Gewichtung wird dieser Faktor durch w_m ersetzt. Dadurch erhalten die Daten aus der unterrepräsentierten Klasse den gleichen Einfluss auf den Verlust, wodurch ein besseres Training möglich ist.

2.3.4 Metriken zur Evaluation

Die Verlustfunktion ist ein Maß dafür, wie stark die Vorhersagen des Netzwerks von den tatsächlichen Werten abweichen. Der Wert geht dann auch in den Trainingsprozess mit ein, um die Abweichungen zu verringern. Es gibt noch andere Funktionen, die ebenfalls die Performance des Netzes bewerten, sogenannte Metriken. Die Ergebnisse davon haben keinen Einfluss auf das Training. Im Folgenden werden zwei Metriken näher betrachtet, die auch später auf das hier verwendete Netzwerk angewandt werden.

Klassifikationsrate

Die Klassifikationsrate ist sehr intuitiv. Dabei wird die Anzahl aller vom Netzwerk richtig klassifizierten Trainingsbeispiele durch die Gesamtzahl aller Beispiele, auf denen die Performance evaluiert wurde, geteilt. Hier wird der Fall einer binären Klassifikation betrachtet. Wie in Abschnitt 2.3.2 erwähnt, ist der Output des Netzwerks eine normierte Wahrscheinlichkeitsverteilung. Hier im binären Fall die Wahrscheinlichkeit für Klasse A und Klasse B: $[p_A, p_B]$ wobei $p_B = 1 - p_A$. Es wird eine Klassifikationsgrenze definiert mit dem Wertebereich $[0, 1]$. Liegt die Wahrscheinlichkeit, dass ein Trainingsbeispiel

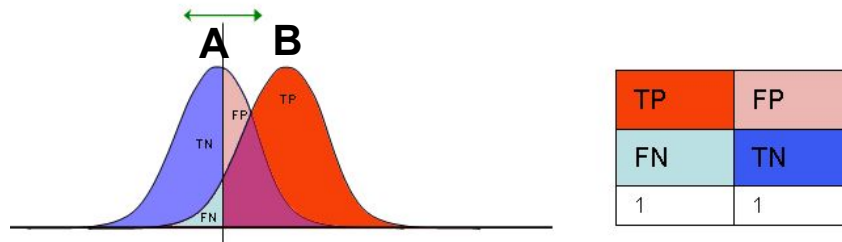


Abbildung 19: Verdeutlichung der Berechnung der ROC-Kurve. Der linke Peak stellt alle Ereignisse der Klasse A dar, der Rechte alle der Klasse B. Die vertikale Linie ist die Klassifikationsgrenze. Die Tabelle rechts veranschaulicht die daraus entstehenden Mengen in unterschiedlichen Farben: True Positive (TP), True Negative (TN), False Negative (FN) und False Positive (FP) [24].

einer bestimmten Klasse angehört, hier p_A über dieser Grenze, so wird es entsprechend als A klassifiziert, anderenfalls als B.

Receiver Operating Characteristic (ROC)

Eine weitere Metrik ist die *Receiver Operating Characteristic Curve* (ROC-Kurve). Dieser Graph zeigt die Performance des Klassifikationsmodells bei allen Klassifikationsgrenzen. Im Folgenden nehmen wir zwei Klassen an, A und B. Das neuronale Netz trifft Vorhersagen auf dem verwendeten Datensatz mit einer festgelegten Klassifikationsgrenze. In Abbildung 19 ist dies grafisch veranschaulicht. Jede Klasse ist dargestellt durch ein Histogramm, wobei der linke Peak alle Datenpunkte der Klasse A und der Rechte alle Datenpunkte der Klasse B enthält. Die Klassifikationsgrenze ist durch einen vertikalen Strich symbolisiert, wobei alle Punkte links davon vom Netzwerk der Klasse A und allen Punkten rechts davon der Klasse B zugeordnet wurden. True Positive (TP) bezeichnet alle Datenpunkte, die tatsächlich der Klasse B angehören und die vom Netzwerk auch richtig klassifiziert wurden. True Negative (TN) sind alle Datenpunkte, die tatsächlich der Klasse A angehören und auch richtig vom Netzwerk erkannt wurden. False Positive (FP) sind die Ereignisse, die Klasse A sind, aber fälschlicherweise als Klasse B klassifiziert wurde. False Negative (FN) sind die Ereignisse, die der Klasse B angehören und der Klasse A zugeordnet wurden. Durch eine Verschiebung der Klassifikationsgrenze verschiebt sich die Anzahl innerhalb dieser definierten Klassen. Es werden zwei weitere Begriffe definiert, welche die Achsen der ROC-Kurve darstellen (vgl. Abbildung 20).

- Die True-Positive-Rate (TPR): $\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
- Die False-Positive-Rate (FPR): $\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$

Die ideale Klassifikation hat eine ROC-Kurve, die zunächst senkrecht ansteigt. Hier liegt die Trefferquote bei 100%, während die Falschklassifikation bei 0% liegt [22].

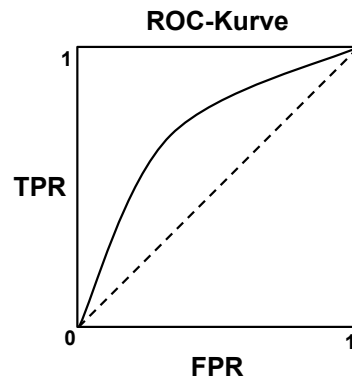


Abbildung 20: Darstellung einer ROC-Kurve. Es ist die TRP über die FPR aufgetragen, wobei beide einen Wertebereich von $[0,1]$ haben. Die Diagonale ist gestrichelt. Jede Klassifikationsgrenze stellt einen Punkt in der ROC-Kurve dar.

3 Motivation für die Arbeit

Wie bereits in der Einleitung erwähnt, ist das Ziel der Arbeit die Klassifikation der Primärteilchen der kosmischer Strahlung mit Hilfe eines neuronalen Netzes. Dabei wird ein sehr direkter Ansatz gewählt, indem man als Eingangsinformation nur die Rohdaten des IceCube Arrays verwendet. Mit Rohdaten sind die detektierten Photonen, die mittels PMTs in Ladung umgewandelt werden, gemeint. Dabei werden auf die Rohdaten keine Algorithmen zu Energie- oder Teilchenbahnrekonstruktion angewendet, die dem Netzwerk als Input zur Verfügung stehen könnten. Diese Rekonstruktionen soll das Netzwerk selbstständig erlernen, um eine Klassifikation durchzuführen.

Für die verschiedenen Elemente der Primärteilchen werden im Detektor unterschiedliche Signale erwartet. Entsprechend dem Superpositionsmodell (vgl. Abschnitt 2.2.2) ist die Entwicklung der Luftschauer abhängig von der Masse des Primärteilchens. Die Zahl der erzeugten Myonen sinkt mit abnehmender Masse des Primärteilchens, unter der Voraussetzung gleicher Primärenergien. Da die Primärenergie auf weniger Teilchen verteilt wird, werden mit abnehmender Masse des Primärteilchens hochenergetischere Myonen erwartet. Bei den Interaktionen hochenergetischer Teilchen im Eis werden stochastischere Energieverluste vorausgesagt. Die Detektion, welche mit dem IceCube Detektor durchgeführt wird, als auch die Entstehung der kosmischen Strahlung sowie deren Entwicklung in der Atmosphäre hin zu Luftschauern, wird simuliert. Dabei werden, wie in Abschnitt 2.2.4 erwähnt, fünf verschiedene Klassen von kosmischen Primärteilchen erzeugt (H, He, N, Al, Fe). Durch die Massenabhängigkeit der Luftschauer erwartet man bei Wasserstoff zu Eisen die signifikantesten Unterschiede in der Detektorsignatur, also dem Signal, welches innerhalb des Detektors gemessen wird. Aus diesem Grund werden alle detektierten Luftschauer, ausgelöst von den Primärteilchen He, N, Al, vernachlässigt und eine Klassifikation nur nach Wasserstoff und Eisen durchgeführt. Diese Klassifikation soll ein neuronales Netz übernehmen, welches in der Lage sein sollte, die eben beschriebene Effekte im Detektorsignal zu unterscheiden. Um das eben beschriebene Ziel zu erreichen, wurde im Rahmen dieser Arbeit zunächst der Datensatz näher betrachtet (vgl. Abschnitt 4). Dieser setzt sich aus dem Trainingsset und dem Validierungsset zusammen. Es wurde die Verteilung der Klassen, das Energiespektrum der Primärteilchen und das der detektierten Sekundärteilchen untersucht. Außerdem wurde näher auf die Energieverteilung innerhalb eines Myonenpakets und die Herkunftsrichtung der Primärteilchen eingegangen. Durch die Betrachtung des Datensatzes soll ausgeschlossen werden, dass darin unerwartete Strukturen auftauchen, die unphysikalisch sind.

Anschließend an die Betrachtung der Datensätze, wurden mehrere neuronale Netze implementiert und mit Hilfe der Datensätze trainiert. Anhand deren Ergebnisse wurde die Netzwerkstruktur verändert und optimiert. Der finale Aufbau des Netzwerks wird im Kapitel 5 beschrieben und diskutiert. Anschließend wurde das verwendete Netzwerk genauer untersucht. In Kapitel Abschnitt 6 wird die Klassifikationsrate, der Verlust und die Anpassung der Lernrate betrachtet und diskutiert. Anschließend wurde versucht die Strukturen zu extrahieren, anhand derer das Netzwerk die Klassifikationsentscheidung trifft.

4 Datensatz

Der Datensatz enthält Simulationen von einzelnen, sogenannte Events. Jedes Event umfasst einen Luftschaer, ausgelöst von einem Primärteilchen (hier H oder Fe), dessen Ladungssignatur dann im Detektor sichtbar ist.

Im Datensatz sind für jeden Luftschaer alle verwendeten Parameter zur Erzeugung und zur Entwicklung gespeichert. Dazu zählen die Art und Energie der Primär- und Sekundärteilchen, alle räumlichen und zeitlichen Informationen über die Wechselwirkungen und Zerfälle dieser Teilchen, sowie deren Ausbreitungsrichtungen in Form von Zenit und Azimut, etc. Außerdem werden alle Informationen über die Photonen, erzeugt durch Cherenkov-Strahlung, gespeichert. Die Photonen werden in den DOMS mittels PMTs in Ladung umgewandelt, welche ebenfalls gespeichert wird. Auf die genaue Datenstruktur der Ladung, welche den Input des Netzwerks darstellt, wird in Unterabschnitt 5.1 genauer eingegangen.

Der Datensatz ist zweigeteilt, in ein Trainingsset mit 2.440.702 Events und in ein Validierungsset, welches 22.711 Events enthält. Das Verhältnis des Validierungssets zum Trainingsset ist somit bei circa 1:100. Im Folgenden werden diese beiden Datensätze genauer beleuchtet. Einige Aspekte werden auf dem Trainings- und einige auf dem Validierungsset betrachtet. Der Grund dafür ist, dass die Daten so präprozessiert wurden, dass das Training des neuronalen Netzes möglichst schnell geht. Weshalb im Trainingsset nur fürs Training relevante Informationen der Simulation gespeichert wurden. Dadurch wurde eine Verkleinerung der Dateien erreicht und somit die Verarbeitung der Daten und daraus folgend der Trainingsprozess beschleunigt. Im Validierungsset hingegen sind alle Informationen gespeichert. Deshalb war es in einigen Fällen nur möglich die Aspekte auf dem kleineren Validierungsset zu untersuchen. Da die Abweichungen zwischen den beiden Datensätzen im Mittel sehr gering sind, ist die einseitige Betrachtung vertretbar.

Zunächst wird die Teilchenart des Primärteilchens und die Verteilung auf die beiden Klassen innerhalb des Datensatzes analysiert. 82% davon sind der Klasse Wasserstoff und die restlichen 18% der Klasse Eisen zugeordnet. Um diese ungleichmäßige Verteilung während des Trainings zu berücksichtigen, werden Klassengewichte verwendet (vgl. Abschnitt 5).

Im Folgenden wird das Energiespektrum der Primärteilchen betrachtet. Mit Hilfe von CORSIKA werden Primärteilchen mit einer Energie zwischen 1×10^5 GeV und 1×10^{11} GeV simuliert. Die Häufigkeit der erzeugten Teilchen hängt von deren Energie ab und nimmt mit E^{-2} ab. Das bedeutet niederenergetische Teilchen treten häufiger auf als Hochenergetische. Im Datensatz sind nicht alle aus Primärteilchen resultierenden Events enthalten. Nur falls der Detektor mindestens 1×10^4 Photonen nachweist, wird das Event gespeichert. Dadurch wird sichergestellt, dass nur Events berücksichtigt werden, die ein starkes Signal im Detektor hinterlassen, wodurch eine Klassifikation vereinfacht wird. In Abbildung 21a sind die Energiehistogramme aller Primärteilchen des Validierungssets aufgetragen, separat für Wasserstoff und Eisen. Zunächst fällt auf, dass der Eisen-Peak etwas niederenergetischer ist als der des Wasserstoffs. Außerdem haben die Eisen-Events zum Großteil eine Energie zwischen 5.5×10^5 GeV und

1×10^9 GeV. Die untere Grenze lässt sich folgendermaßen erklären: Myonen benötigen in etwa 500 GeV um das Eis zu durchqueren und im Detektor sichtbar zu sein [13]. Bei Eisen wird die Energie E_0 eines Primärteilchens nach dem Superpositionsmodell auf 56 Teilchen mit jeweils einer Energie von $E_0 / 56$ aufgeteilt. Ein Myon, welches aus einem Primärteilchen mit der Energie E_0 entsteht trägt in etwa $\frac{1}{20} E_0$ an Energie [12]. Berechnet man die Mindestenergie, die ein Eisen-Primärteilchen haben muss, damit die erzeugten Myonen im Detektor sichtbar sind, so erhält man aus Gleichung 14 $E_{min} = 5.6 \times 10^5$ GeV.

$$E_{min} = 500 \text{ GeV} \cdot 20 \cdot 56 = 5.6 \times 10^5 \text{ GeV} \quad (14)$$

Dazu kommt, dass nur Events mit einer bestimmten Mindestzahl an detektierten Photonen gespeichert werden, wodurch das Spektrum zur unteren Grenze hin abfällt. Die obere Grenze ergibt sich durch die abfallende Häufigkeit der erzeugten Primärteilchen mit zunehmender Energie. Der Verlauf des Wasserstoffspektrums lässt sich analog erklären.

In Abbildung 21b sind ebenfalls zwei Energiehistogramme aufgetragen, wieder für Wasserstoff und Eisen. Diese Energie ist dabei die Summe aller Myonenenergien, die bei einem Event im Detektor nachgewiesen wurden und wird im Folgenden als E_{mea} bezeichnet. Die Peaks sind entsprechend den Erwartungen niederenergetischer als zuvor, da die Teilchen in der Atmosphäre und im Eis Energie verlieren. Die qualitativen Verläufe lassen sich analog zum Primärenergiespektrum erklären.

Wie im Superpositionsmodell (vgl. Abschnitt 2.2.2) beschrieben, werden bei massereicheren Primärteilchen mehr Myonen, mit jeweils weniger Energie erzeugt, verglichen mit leichteren Primärteilchen mit der gleichen Primärenergie. Es wird zunächst das hochenergetischste Myon $E_{\mu,max}$ aller erzeugten Myonen im Luftschauer betrachtet. Dividiert man seine Energie durch die Summe der Einzelenergien $E_{\mu,i}$ aller Myonen, so erhält man den Bruchteil des hochenergetischsten Myon an der Gesamtheit aller n erzeugten Myonen.

$$\mu_{frac} = \frac{E_{\mu,max}}{\sum_i^n E_{\mu,i}} \quad (15)$$

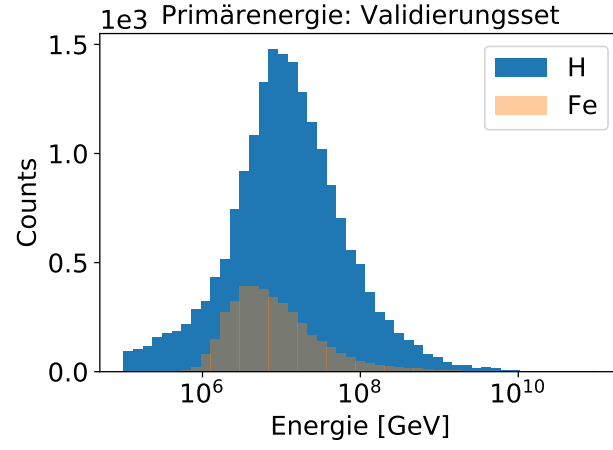
Bei massereicheren Teilchen ist dieser Bruchteil kleiner als bei Masseärmeren.

Dies lässt sich auch im Datensatz beobachten. Trägt man diese Bruchteile separat für H und Eisen in einem Histogramm für alle Events im Validierungsset auf, so erhält man Abbildung 22. Es ist erkennbar, dass bei Eisen-Events der Bruchteil μ_{frac} nur in wenigen Fällen größer als 0.2 ist. Wohingegen die Wasserstoff-Events über den kompletten Bereich $[0,1]$ streuen.

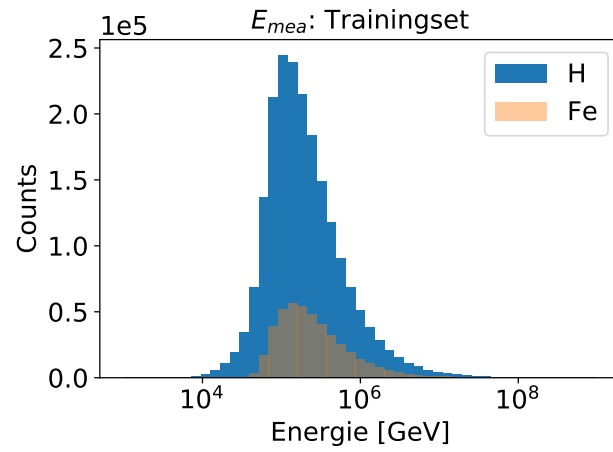
Im Folgenden wird die Winkelverteilung der Herkunftsrichtung der Primärteilchen betrachtet. Diese lässt sich mit Hilfe des Zenit und des Azimut Winkels eindeutig beschreiben, welche in Abbildung 23 eingezeichnet sind. Der Azimut liegt im Bereich $[0,2\pi]$ und der Zenit im Bereich $[0,\pi]$. In Abbildung 24a sieht man das Histogramm des Azimutwinkels für Wasserstoff als auch für Eisen. Damit ein Vergleich möglich ist, wurden beide normiert. Erkennbar ist eine gleichmäßige Verteilung, sowohl für Wasserstoff als auch für Eisen, aus allen Richtungen die statistischen Schwankungen

unterliegt. Keine Klasse besitzt eine Vorzugsrichtung.

Der Zenit-Winkel liegt im Bereich $[0, \pi]$. In Abbildung 24b sieht man ein Histogramm der Verteilung, ebenfalls normiert. Dabei wird der Kosinus des Zenit-Winkels verwendet. Da die Ausbreitung der kosmischen Strahlung ausschließlich nach unten, also zur Erde hin, stattfindet, ist zu erwarten, dass der Zenit-Winkel zwischen 0 und $\pi/2$ liegt. 0 (in 24b: $\cos(0) = 1$) entspricht einer Ausbreitungsrichtung zur Erde hin und $\pi/2$ (in 24b: $\cos(\pi/2) = 0$) einer Ausbreitung parallel zur Erdoberfläche. Weil der Kosinus dieses Winkels aufgetragen ist, liegt der Wert zwischen $[0, 1]$. Diese Erwartung ist auch in den Daten sichtbar. Zu sehen ist, dass die Verteilung gegen $\cos(\pi/2) = 0$ abfällt. Das liegt an der Absorption der Myonen in der Atmosphäre. Je schräger die Teilchen eintreffen, desto länger ist ihr Weg durch die Atmosphäre. Dadurch werden sie mit einer höheren Wahrscheinlichkeit absorbiert. Die leichten Abweichungen zwischen der H- und Fe-Verteilung resultieren vermutlich aus einer leicht unterschiedlichen Absorption. Diese Abweichungen sind sehr gering und machen sich deshalb im Entscheidungsprozess des Netzwerks vermutlich kaum bemerkbar.



(a)



(b)

Abbildung 21: Energiehistogramme des Datensatzes. In (a) ist das ungewichtete Energiehistogramm für alle Primärteilchen des Validierungssets, separat für Wasserstoff (blau)- und Eisen-Events (orange) abgebildet. In (b) das Histogramm von E_{mea} auf dem Trainingsset, ebenfalls für Wasserstoff und Eisen.

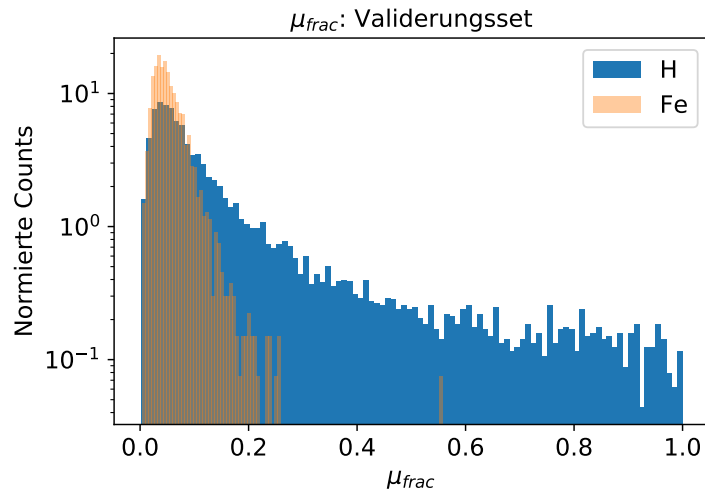


Abbildung 22: Normiertes Histogramm aller Events des Validierungssets aufgetragen für den Bruchteil des hochenergetischsten Myons an der Gesamtenergie aller Myonen, hier bezeichnet als μ_{frac} , getrennt für Wasserstoff- und Eisen-Events

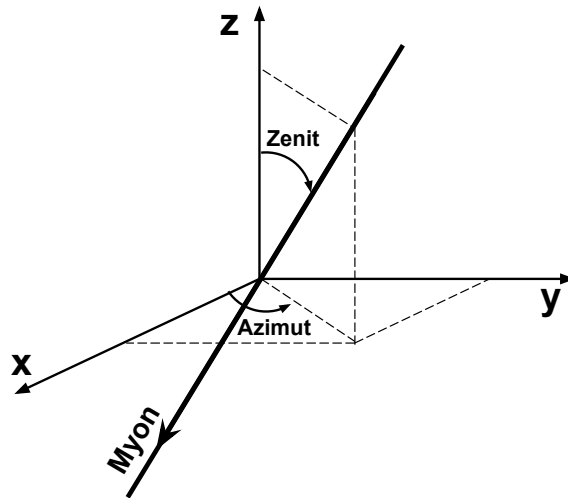
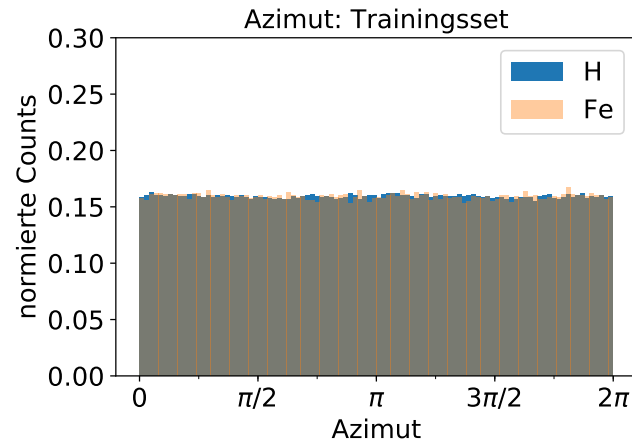
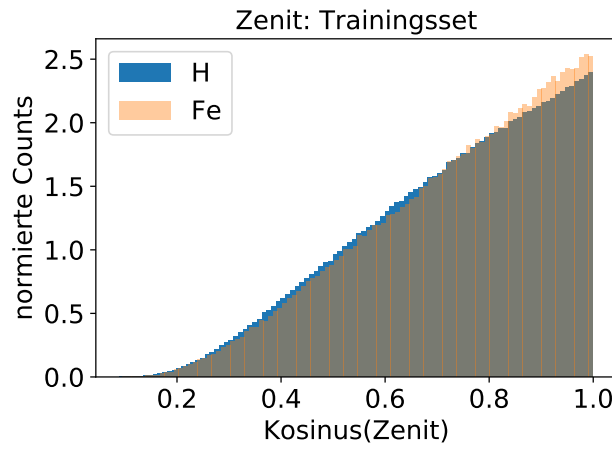


Abbildung 23: Skizze zur Herkunftsrichtung der Myonen mit den eingezeichneten Winkeln Azimut und Zenit [18].



(a)



(b)

Abbildung 24: Normierte Winkelverteilung der Ausbreitungsrichtung der Primärteilchen des Trainingsdatensatzes, dargestellt als Histogramme. In (a) ist der Azimut, aufgetragen für alle H und Fe-Events des Trainingssets. Die Teilchen kommen gleichmäßig aus allen Richtungen, wobei die Verteilung statistischen Schwankungen unterliegt. In (b) sind die Histogramme über den Zenit-Winkel aufgetragen für H und Fe. Dabei ist auf der x-Achse der $\cos(\text{Zenit})$ aufgetragen. Durch die Absorption der Myonen in der Atmosphäre fällt die Verteilung nach links ab. Außerdem existiert eine leichte Abweichung zwischen H und Fe.

5 Implementierung des Netzwerks

Im Folgenden wird auf die innere Struktur des verwendeten neuronalen Netzes als auch auf die verwendeten Einstellungen näher eingegangen. Das neuronale Netz ist in Abbildung 25 grafisch dargestellt. Der Code zum Erstellen und Trainieren des Netzes wurde von einem Energierekonstruktionsnetz übernommen und verändert [26]. Im Rahmen dieser Arbeit wurde dieses neuronale Netz umgebaut zu einem Klassifikationsnetzwerk und durch Anpassung der Hyperparameter für die Problemstellung optimiert.

Insgesamt verwendet das Netzwerk 2.501.666 trainierbare Parameter, die während des Trainings angepasst werden. Der verwendete Optimierungsalgorithmus zu Anpassung ist ADAM [21]. Die Lernrate wird zu Beginn auf 2.5×10^{-4} gesetzt, wobei diese während des Trainings angepasst wird. Zum Trainieren wird, wie in Abschnitt 2.3 beschrieben, die Input-Daten in Gruppen, sogenannte Batches, zusammengefügt. Die Größe eines Batches ist 32.

Wie in Unterunterabschnitt 2.3.3 erklärt, kann die Verwendung von Klassengewichten (vgl. Gleichung 13) zu einem verbesserten Training auf Datensätzen mit unausgewogener Klassenverteilung führen. Da in dem hier verwendeten Datensatz circa 82% Wasserstoff- und nur 18% Eisen-Events sind, werden die Klassengewichte aus Tabelle 1 verwendet. In den nachfolgenden Unterkapiteln wird auf jedes verwendete Layer näher eingegangen.

Element	H	Fe
Klassengewicht	1	4.82

Tabelle 1: Verwendete Klassengewichte für das Training des neuronalen Netzes

5.1 Input

Der Input des hier verwendeten Netzwerks ist die Ladung, welche detektiert wird, ausgelöst von jeweils einem Primärteilchen. Die Ladung für jedes Event ist in einem dreidimensionalen Array der Größe $86 \times 60 \times 210$ gespeichert (vgl. Abbildung 26). Die ersten beiden Zahlen repräsentieren die räumliche Information der Events. Es gibt 86 vertikale Strings, die in das Eis eingelassen sind. An jedem dieser Strings sind 60 DOMs befestigt (vgl. Unterabschnitt 2.1). Jeder DOM wird durch einen Eintrag in dieses zweidimensionale Array 86×60 dargestellt.

Da der Detektor von oben betrachtet eine zweidimensionale, hexagonale Form bildet, die Daten zu jedem String aber nur in einer Dimension gespeichert werden (86), wird die Information der horizontalen Anordnung der Strings zueinander bei der Speicherung nicht berücksichtigt. Die senkrechte Information bleibt jedoch bestehen. Das bedeutet, die erste horizontale Schicht des Detektors wird im Array ebenfalls durch die ersten horizontalen Einträge der Matrix repräsentiert. Zu jedem DOM existiert zusätzlich eine zeitliche Information. Dies ist die dritte Dimension. Das gesamte Event, also die Ausbreitung der Myonen im Detektor, wird in 210 Zeitschritte mit je 50 ns eingeteilt. Innerhalb jedes Zeitschritts wird die gesamte gemessene Ladung eines DOMs

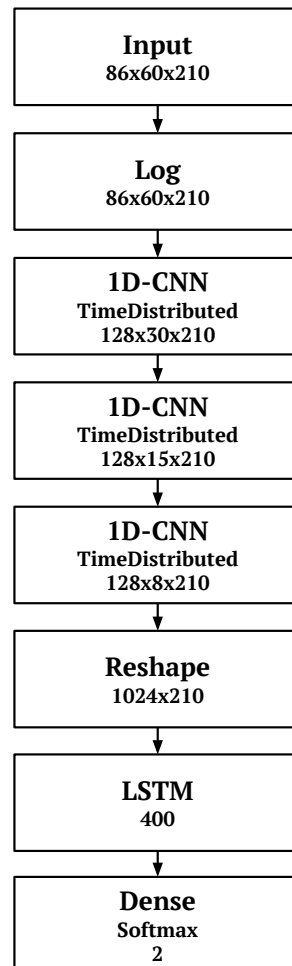


Abbildung 25: Grafische Darstellung des Aufbaus des neuronalen Netzes. Dabei sind die einzelnen Layer als Rechtecke dargestellt. Sie enthalten die Bezeichnungen der Layer, sowie besondere Einstellungen. Die unterste Zeile ist die Form des Outputs jedes Layers.

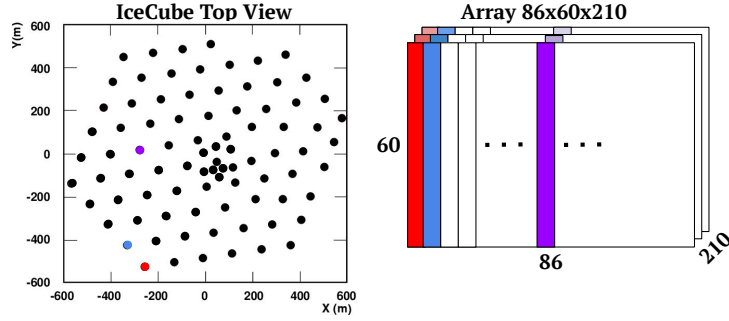


Abbildung 26: Links ist der IceCube Detektor von oben betrachtet [3]. Dabei stellt jeder Punkt einen String dar. Rechts ist das Array eines Events mit der Dimension $86 \times 60 \times 210$. Die Farben markieren einzelne Strings mit 60 Einträgen und deren Speicherung im Array. Hierbei wird die horizontale räumliche Anordnung der Strings zueinander vernachlässigt, die vertikale Information bleibt bestehen.

aufsummiert und erhält dann einen Eintrag im Array. Dabei ist die Ladung der DOMs proportional zur Anzahl der detektierten Photonen.

5.2 Skalierung

Der Input, also die Ladung, hat einen Wertebereich zwischen 0 und circa 5000. Daher ist der erste Schritt die Anwendung der Logarithmusfunktion auf jeden Datenpunkt des Inputs. Dadurch wird der Input reskaliert. Dies ermöglicht dem Lern-Algorithmus des Netzwerks eine schnellere Lern-Kurve und verbessert die Stabilität [8]. Da während eines Events nicht alle DOMs Ladung messen, enthält das Input-Array Nullen. Die Logarithmus-Funktion ist jedoch nur für den Bereich $]0, +\infty]$ definiert, weshalb auf jeden Eintrag innerhalb des Arrays vor der Anwendung der Log-Funktion eine 1 addiert wird.

5.3 CNNs

Die hier verwendete Netzwerkstruktur enthält drei aufeinanderfolgende 1D-CNN-Layer. Die Verwendung dieser Layer zielt auf zwei Punkte ab. Der erste ist die Ausnutzung der senkrechten Translationsinvarianz des Detektors. Alle drei CNN-Layer haben die gleichen Einstellungen. Die Anzahl der verwendeten Filter n_{Filter} ist 128. Die Aktivierungsfunktion ist die *Rectified Linear Unit*-Funktion (ReLU). Diese ist definiert als: $a(x) = \max(0, x)$. Die Größe k des Kernels ist fünf und die Schrittgröße s ist zwei. Die Ausbreitungsrichtung geht anschaulich betrachtet entlang der senkrechten Höhenachse des Detektors. Diese wird so gewählt, da dadurch die Muster zum Eintritt und Austritt des Events in bzw. aus dem Detektor erlernt werden können.

Der Input des ersten Layers ist die skalierte Ladung ($86 \times 60 \times 210$). Wie in Abschnitt

2.3.1 erwähnt, muss der Input zu 1D-CNNs zweidimensional sein. Um dies zu erreichen, wird hier jeder der 210 Zeitschritte separat behandelt. Dadurch erhält man pro Event 210-mal Arrays mit einer Größe von 86×60 , wobei jedes dieser zweidimensionalen Arrays unabhängig von den anderen Zeitschritten die CNN-Layer durchläuft. Dieser Prozess wird hier als *Time Distributed* bezeichnet. Dadurch ist es für die CNN-Layer nicht möglich zeitabhängige Informationen zu extrahieren. Dafür werden dann im Folgenden LSTM-Layer verwendet. Die Größe des ersten Outputs ist entsprechend $128 \times 30 \times 210$. Pro Layer verkleinert sich dann die Größe des Outputs weiter, wodurch am Ende ein Array der Größe $128 \times 8 \times 210$ ausgegeben wird.

5.4 Anpassung der Output-Form

Dieser Layer, in Abbildung 25 als *Reshape* bezeichnet, verändert nur die Form des Arrays. Es erhält ein Array der Größe $128 \times 8 \times 210$. Dabei beinhalten die ersten zwei Dimensionen die Informationen über die Ladung für jeden Zeitschritt. Die Informationen der Ladung (128×8) werden hier in ein eindimensionales Array konvertiert, woraus sich für alle Zeitschritte ein zweidimensionales Array der Größe 1024×210 ergibt. Dabei gehen keine Informationen verloren, aber die Verarbeitung durch die LSTM-Layer wird ermöglicht.

5.5 LSTM

Der verwendete LSTM-Layer (vgl. Abschnitt 2.3.1) besitzt für jeden der 210 Zeitschritte ein Modul. Die eigenständigen neuronalen Netze, realisiert durch die Gates innerhalb jedes Moduls, haben 400 Neuronen. Die Ausgabewerte der einzelnen Module h_t werden an das nächste Modul weitergereicht. Nur der Ausgabewert $h_{t=210}$ des letzten Moduls wird als Output des LSTM-Layers verwendet. Dieser besitzt somit die Dimension 400.

5.6 Softmax

Der letzte Layer besitzt zwei Neuronen, welches die Softmax-Funktion (vgl. Abschnitt 2.3.2) als Aktivierungsfunktion verwendet. Dieses gibt für jede Klasse eine Wahrscheinlichkeit aus, mit der ein Event dieser Klasse angehört. In diesem konkreten Fall bedeutet das anschaulich betrachtet, dass eine Wahrscheinlichkeit $p(H)$ ausgegeben wird, mit der das Event der Klasse Wasserstoff zugeordnet ist. Da nur zwei Klassen verwendet werden, ist die Wahrscheinlichkeit für Eisen: $1-p(H)$.

6 Ergebnisse

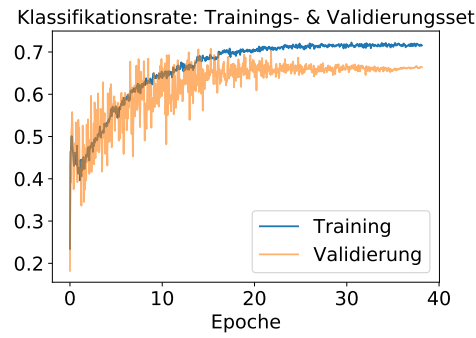
6.1 Trainingsprozess

In diesem Kapitel wird zunächst das Verhalten des neuronalen Netzes während des Trainings betrachtet. Dazu ist in Abbildung 27a die Klassifikationsrate des verwendeten Netzes über die Epochen aufgetragen. Diese wurde auf dem Trainings- als auch auf dem Validierungsset ermittelt. Die verwendete Klassifikationsgrenze (vgl. Unterunterabschnitt 2.3.4) liegt hier bei 0,5. Dieser Plot veranschaulicht den Lernprozess des Netzwerks während des Trainings in Bezug auf die Zahl der richtig klassifizierten Events. Die Klassifikationsrate startet bei 0,25. Dies lässt sich dadurch erklären, dass zu Beginn die Gewichte innerhalb des Netzwerks zufällig initialisiert werden, wodurch keine Klassifikation möglich ist. Dann steigt der Verlauf sprunghaft auf circa 0,5 und fällt anschließend wieder auf circa 0,4 ab. Danach nimmt der Verlauf verstärkt zu, wobei die Steigung des Anstiegs sich langsam verringert, bis bei circa 14 Trainingsepochen ein Wert von circa 0,7 erreicht wird. Bis zum Ende des Trainings bleibt die Klassifikationsrate dann annähernd konstant. Auf diesen Verlauf wird gleich noch einmal näher eingegangen. Die Performance auf dem Validierungsset verhält sich fast analog zum Trainingsset, wobei zwei Unterschiede deutlich sichtbar sind. Der Erste sind die starken Fluktuationen, welche daher kommen, dass das Validierungsset deutlich weniger Events als das Trainingsset enthält. Dadurch streuen die berechneten Werte stärker um den Mittelwert. Die Fluktuationen nehmen mit steigender Epochenzahl deutlich ab, worauf auch gleich nochmal eingegangen wird. Der zweite Unterschied ist, dass die Klassifikationsrate nur bis circa 0,65 ansteigt. Diese Differenz, also die bessere Performance auf dem Trainingsset, lässt sich durch das sogenannte *Overfitting* erklären. Das bedeutet, das Netzwerk erlernt Muster, die spezifisch für das Trainingsset sind und sich nicht auf unbekannte Daten verallgemeinern lassen.

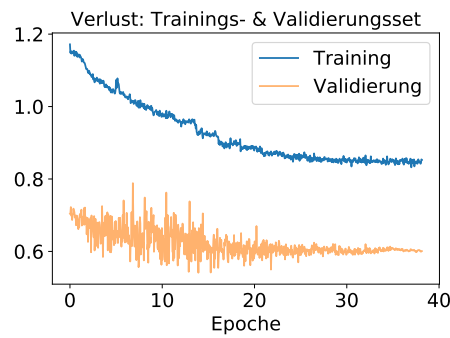
In Abbildung 27b ist der Verlust, also der mittels Verlustfunktion errechnete Wert, für die beiden Datensätze über die Epochen aufgetragen. Allerdings ist nur der Verlust auf dem Trainingsset und nicht der auf dem Validierungsset entsprechend den Klassen gewichtet. Deshalb ist ein Vergleich schwer, wobei dennoch der qualitative Verlauf betrachtet werden kann. Die einseitige Gewichtung des Trainingsverlustes ergibt sich aus der verwendeten Python Bibliothek *Keras* [20]. Der gewichtete Trainingsverlust startet bei circa 1,15. Dies lässt sich erklären, wenn man die Klassenverteilung und die Formel zur Berechnung des Verlust (vgl. Gleichung 12) näher betrachtet. Zu Beginn des Trainings hat das Netzwerk noch keine Struktur der Daten erlernt und ordnet somit beiden Klassen in etwa die gleiche Wahrscheinlichkeit von 0,5 zu. Außerdem sind im Datenset circa 82% Wasserstoff-Events und entsprechend nur 18% Eisen-Events. Berechnet man nun den Verlust, und gewichtet jedes Eisen-Event entsprechend den Klassengewichten (vgl. Abschnitt 5), hier mit 4,82, so erhält man folgenden Wert für den gemittelten Verlust auf dem Datensatz:

$$L = -1 \cdot 0.82 \cdot \ln(0.5) - 4.82 \cdot 0.18 \cdot \ln(0.5) \approx 1.15 \quad (16)$$

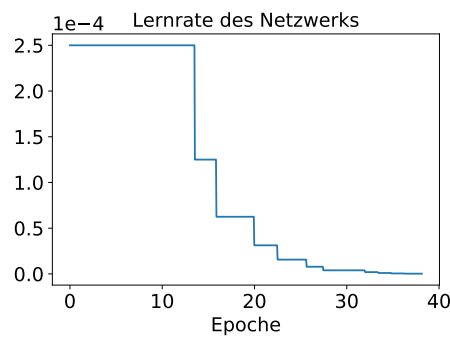
Innerhalb der ersten 15 Epochen nimmt der Verlust dann stark ab, von circa 1,15 auf 0,9. Anschließend sinkt dieser weiter, wobei sich der Verlust nur noch auf circa 0,85



(a)



(b)



(c)

Abbildung 27: In diesen Abbildungen wird das Verhalten des neuronalen Netzes während des Trainings dargestellt. In (a) ist die Klassifikationsrate zu sehen. In (b) der Verlust, wobei nur der Verlust auf dem Trainingsset gewichtet ist, wohingegen der auf dem Validierungsset ungewichtet ermittelt wurde. In (c) ist die Lernrate des Netzwerks während des Trainings. Alle Verläufe sind über die Zahl der Trainingsepochen aufgetragen. In (a) und (b) ist der jeweilige Verlauf, ermittelt auf dem Trainingsset in orange und der auf dem Validerungsset in blau abgebildet.

verringert. Der Validierungsverlust startet bei circa 0,7. Da dieser ungewichtet ist, ergibt sich das aus $L = -1 \cdot \ln(0.5) = 0.69$. Anschließend fällt der Verlauf sehr gering ab, bis er bei circa 13 Epochen anfängt bei einem Wert von circa 0.6 zu stagnieren. Dass der Validierungsverlust konstant bleibt und nicht beginnt zu steigen, lässt darauf schließen, dass das Overfitting sehr gering ist. Auch hier, auf dem Verlust des Validierungssets, sind starke Fluktuationen zu beobachten, die sich mit zunehmender Epochenzahl verringern.

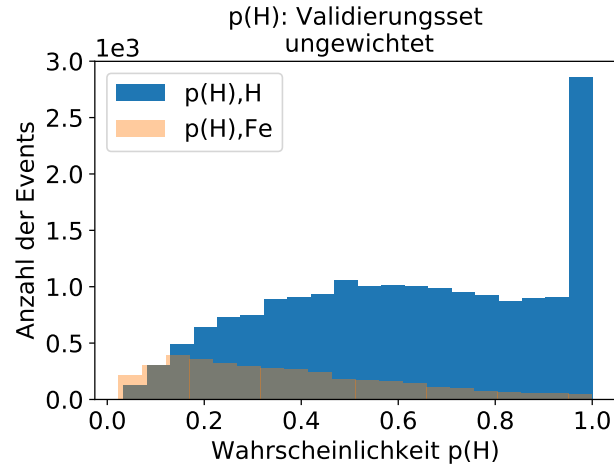
Um die Verläufe des Verlusts und der Klassifikationsrate zu erklären, muss zunächst die Entwicklung der Lernrate des Netzwerks während des Trainings betrachtet werden. Diese ist in Abbildung 27c abgebildet. Die Lernrate wurde zu Beginn des Trainings auf 2.5×10^{-4} gesetzt. Diese wird jedoch während des Trainingsprozesses verkleinert, und zwar dann, wenn der Verlust auf dem Validierungsset über einige Zeit stagniert. Dies ist ein übliches Verfahren. Durch eine kleinere Lernrate werden die Änderungen an den Gewichten auch kleiner. Dies kann dazu führen, dass der Verlust auf dem Validierungsset wieder anfängt etwas zu sinken. Außerdem wird durch dieses Verfahren der Overfitting-Effekt verringert. In Abbildung 27c lässt sich dieses Verhalten der Lernrate beobachten. Die Lernrate halbiert sich nach circa 13 Epochen das erste Mal, da der Validierungsverlust stagniert (vgl. Abbildung 27b). Anschließend lässt sich diese Halbierung in etwa alle drei Epochen beobachten. Die starke Reduktion der Lernrate erklärt mehrere Effekte. Zum einen ist die abnehmende Steigung der Klassifikationsrate, als auch des Verlustes nach circa 13 Epochen darauf zurück zu führen. Aber auch die abnehmenden Fluktuationen ergeben sich durch die verkleinerten Schritte in der Anpassung der Gewichte.

6.2 Evaluation der ungewichteten Verteilung

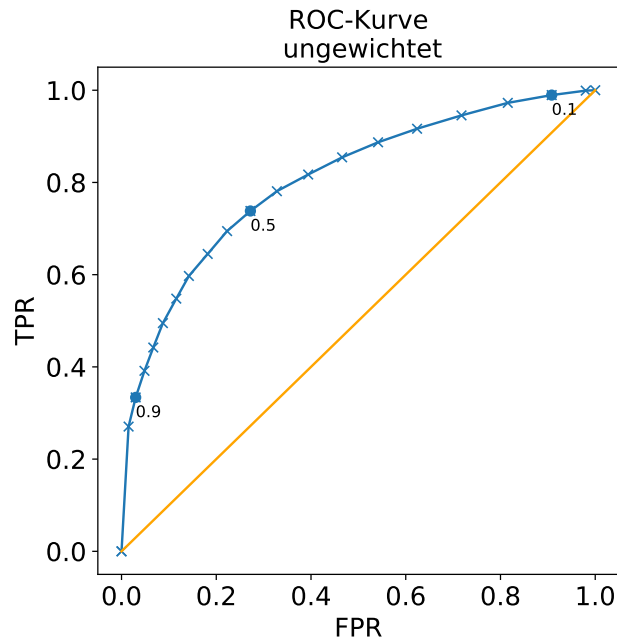
Im Folgenden werden die vom Netzwerk vergebenen Wahrscheinlichkeiten $p(H)$ auf dem Validierungsset näher betrachtet. Die Wahrscheinlichkeit gibt an, wie sicher sich das Netzwerk ist, dass der Luftschauer des Events von einem Wasserstoffkern ausgelöst wurde. Außerdem wird mit Hilfe einer Metrik, der ROC-Kurve, das Verhalten bei verschiedenen Klassifikationsgrenzen analysiert.

Dazu sind in Abbildung 28a zwei Histogramme abgebildet, wobei jedes Event im Validierungsset über die vom Netzwerk vergebene Wahrscheinlichkeit $p(H)$ aufgetragen ist. Die Wahrscheinlichkeiten wurden vom fertig trainierten Netzwerk, also nach circa 38 Epochen, berechnet. Sehr auffällig ist der Peak am rechten Rand der Wasserstoffverteilung. Demnach gibt es viele Wasserstoff-Events, die vom Netzwerk eine sehr hohe Wahrscheinlichkeit $p(H)$ zugeordnet bekommen. Außerdem gibt es im Bereich $[0.5, 0.95]$ ein Plateau, wobei dieses im Bereich $[0, 0.5]$ abfällt. Zudem lässt sich beobachten, dass die Wahrscheinlichkeitsverteilung der Eisen-Events stark nach links verschoben ist. Das bedeutet, dass beim Großteil dieser Events ebenfalls die Klasse richtig erkannt wird, wobei auch diese Verteilung nur sehr langsam abebbt.

In diesen Plot kann man nun eine Klassifikationsgrenze K_{sep} legen, vorstellbar durch eine vertikale Trennlinie. Diese Plots sind demnach so zu verstehen, dass alle Events, die links dieser Klassifikationsgrenze liegen ($p(H) < K_{sep}$) vom Netzwerk als Eisen



(a)



(b)

Abbildung 28: In (a) sind zwei ungewichtete Histogramme, die jedes Event im Validierungsset über die vom Netzwerk vergebene Wahrscheinlichkeit $p(H)$ auftragen, für Wasserstoff (blau) und Eisen (orange). In (b) ist die zugehörige ROC-Kurve in blau. Dabei stellt jedes Kreuz eine Klassifikationsgrenze zwischen $[0,1]$ dar, wobei die Schrittweite 0,05 ist. Außerdem sind drei blaue Punkte markiert, bei denen die verwendete Klassifikationsgrenze dabei steht. Die orangefarbene Linie ist die Diagonale.

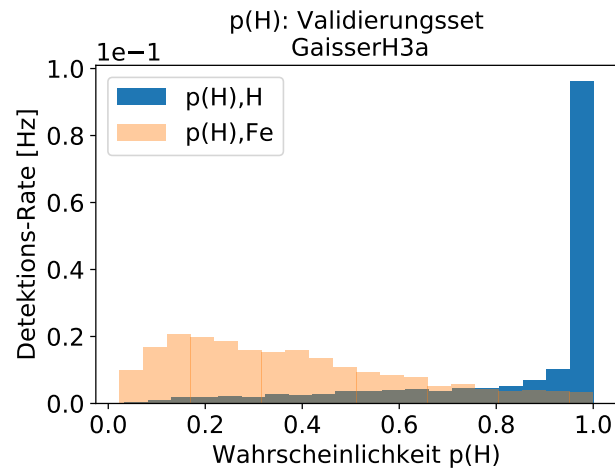
klassifiziert werden und alle Events rechts davon ($p(H) > K_{sep}$) als Wasserstoff. In Abbildung 28b ist die zu Abbildung 28a gehörige ROC-Kurve aufgetragen. Die Kurve steigt zu Beginn auf einen Wert von 0,2 für die TPR (vgl. Abbildung 19), wobei die FPR noch knapp über Null liegt. Das ist folgendermaßen zu erklären: Obwohl der Peak in Abbildung 28a sich sehr deutlich von der Verteilung abhebt, enthält das Plateau im Bereich $[0.5, 0.95]$ etwa fünf mal so viele Events, die bei einer hohen Klassifikationsrate dann falsch klassifiziert werden. Allerdings werden in diesem Fall anteilig nur sehr wenige Eisen-Events falsch klassifiziert. Der weitere Verlauf der ROC-Kurve liegt deutlich über der Diagonalen. Bei der Klassifikationsgrenze von 0,5 ergibt sich eine TPR von circa 0,65, bei einer Fehlerrate von nur 0,25.

6.3 Evaluation der gewichteten Verteilung

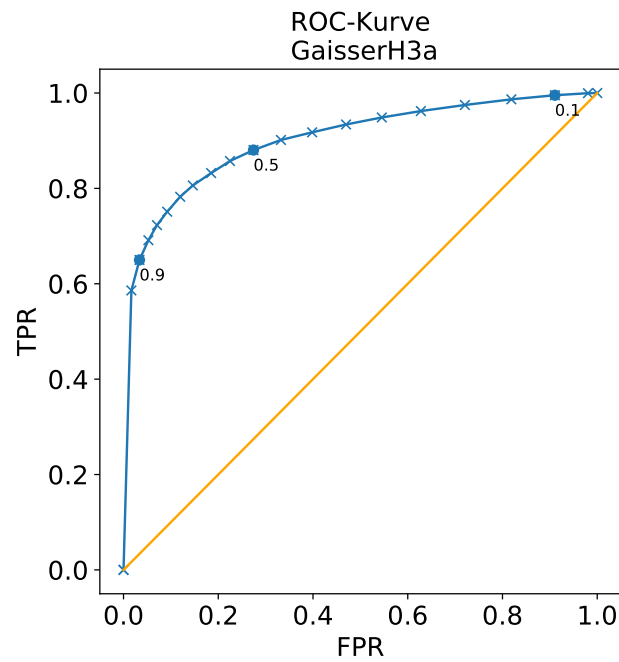
Die Verteilung auf die beiden Klassen während des Trainings des Netzwerks entspricht nicht der Realität. Aus diesem Grund wird nun eine Ungewichtung mit Hilfe des GaisserH3a-Modells (vgl. Abschnitt 2.2.4) vorgenommen. Zuvor war in Abbildung 28a die absolute Zahl der Events aufgetragen. Durch die Gewichtung ergibt sich im Histogramm eine Detektionsrate pro Sekunde. Dabei bleibt der qualitative Verlauf gleich. In Abbildung 29a lässt sich erkennen, dass die Separation zwischen den beiden Klassen deutlich besser funktioniert als im ungewichteten Fall (vgl. Abbildung 28a). Das liegt vor allem daran, dass schlecht klassifizierbare Wasserstoff-Events aus den ungewichteten Daten in den GaisserH3a gewichteten Daten viel seltener auftreten. Das Plateau in der Wasserstoff-Verteilung im Bereich $[0, 0.95]$ ist stark reduziert. Diese Veränderung lässt sich auch sehr gut in der ROC-Kurve in Abbildung 29b beobachten. Bei einer Klassifikationsgrenze von 0,9 liegt die TPR nun bei circa 0,65, wobei die FPR hier nur knapp über der Null liegt. Das ergibt eine Differenz von circa 0,45 zum ungewichteten Fall. Danach flacht die Kurve etwas ab. Bei der Klassifikationsgrenze von 0,5 liegt die TPR bei fast 0,9, wohingegen die Fehlerrate, wie in der ungewichteten ROC-Kurve vorher, bei circa 0,25 liegt.

6.4 Physikalische Eigenschaften der Klassifikation

Im Folgenden wird nun untersucht, welche physikalischen Eigenschaften das Netzwerk für die Klassifikation verwendet. Das Erste ist die Primärenergie. Dazu ist in Abbildung 30 das nach GaisserH3a gewichtete Spektrum der Primärenergie für Wasserstoff und Eisen aufgetragen. Wie schon im ungewichteten Fall lässt sich erkennen, dass Eisen eine untere Grenze bei circa 1×10^6 GeV besitzt, wohingegen Wasserstoff-Events für den Simulationsbereich bis 1×10^5 GeV auftreten. Dadurch ergibt sich eine Zone links der Grenze, in der fast ausschließlich Wasserstoff-Events sind. Plottet man nur Events, deren Primärenergie größer als diese Grenze von 1×10^6 GeV ist, erhält man Abbildung 31a und Abbildung 31b. Erkennbar ist, dass der Peak in der Wahrscheinlichkeitsverteilung des Wasserstoffs von circa 0,1 auf 0,02 auf etwa ein Fünftel abnimmt. Das bedeutet, das neuronale Netz ist in der Lage die Primärenergie zu rekon-



(a)



(b)

Abbildung 29: In (a) sind die Histogramme der vom Netzwerk vergebenen Wahrscheinlichkeit $p(H)$ für Wasserstoff (blau) und Eisen (orange) auf den nach GaisserH3a gewichteten Daten abgebildet. In (b) ist die zugehörige ROC-Kurve in blau. Die markierten Punkte in der Kurve sind Klassifikationsgrenzen.

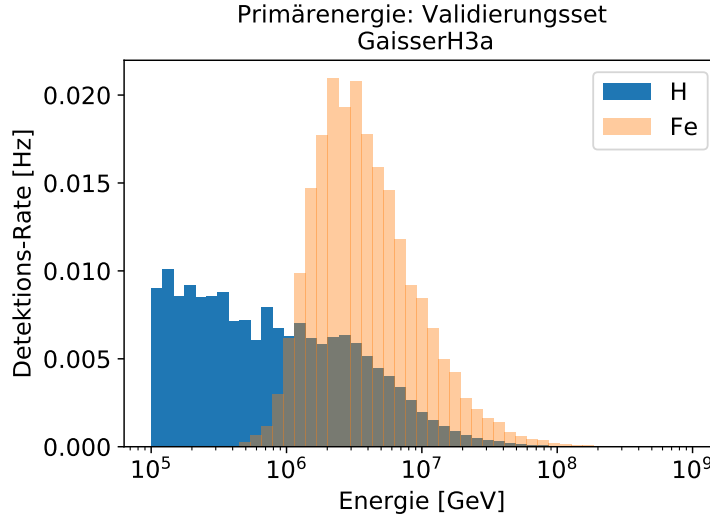
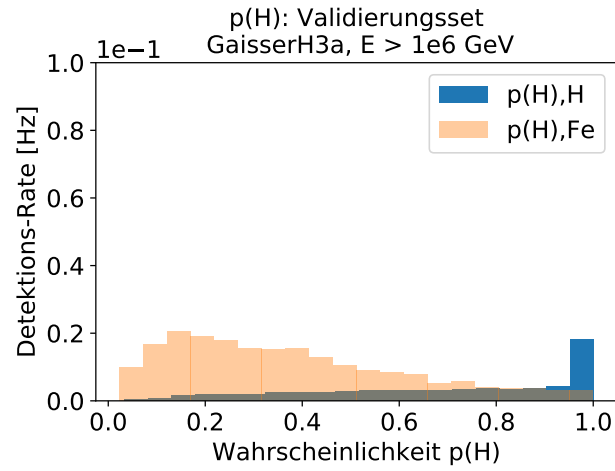


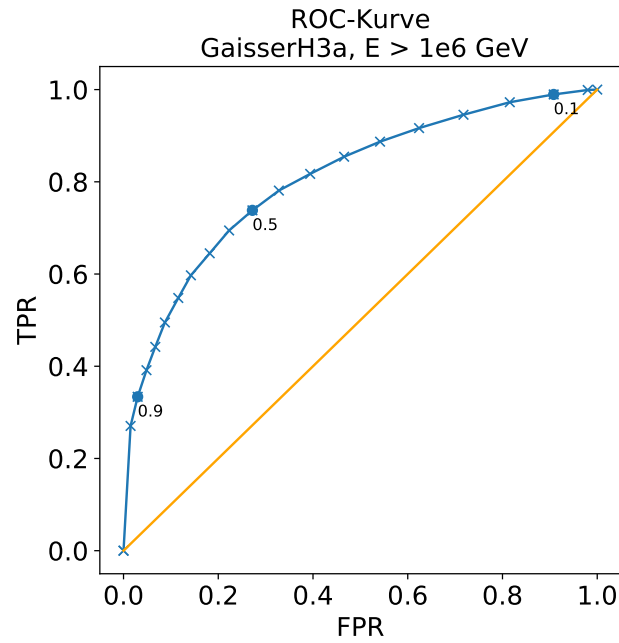
Abbildung 30: Nach dem GaisserH3a-Modell gewichtetes Energiehistogramm der Primärenergie für Wasserstoff (blau) und Eisen (orange)

struieren und daher niederenergetische Wasserstoff-Events eindeutig zu identifizieren. Daraus lässt sich ableiten, dass die Primärenergie ein entscheidendes und physikalisch sinnvolles Klassifikationskriterium ist.

Das zweite Klassifikationskriterium ist μ_{frac} (vgl. Gleichung 15). Im Folgenden werden die Events der gewichteten Wahrscheinlichkeitsverteilung aus Abbildung 29a dahingehend näher betrachtet. Dafür werden drei Wahrscheinlichkeits-Intervalle definiert, aus denen für alle enthaltenen Events der berechnete Bruchteil μ_{frac} in einem Histogramm aufgetragen wird. Der erste Bereich ist $[0,0.2]$ und enthält alle Eisen-Events, denen das Netzwerk eine sehr hohe Wahrscheinlichkeit beimisst, dass sie tatsächlich Eisen sind. Außerdem einige wenige stark fehlklassifizierte Wasserstoff-Events. Die zugehörigen Histogramme für Wasserstoff und Eisen sind in Abbildung 32a abgebildet. Erkennbar ist, dass für alle Eisen-Events μ_{frac} kleiner als 0,2 ist. Dies entspricht auch den Erwartungen, (vgl. Abbildung 22). Außerdem fällt auf, dass die falsch klassifizierten Wasserstoff-Events auch zum größten Teil in diesem Bereich liegen, obwohl es auch wenige mit deutlich höherer μ_{frac} gibt. Das zweite Intervall ist $[0.4,0.6]$. Die Histogramme zum Intervall werden in Abbildung 32b gezeigt. Betrachtet man die vom Netzwerk vergebene Wahrscheinlichkeit als ein Maß dafür, wie sicher sich das Netzwerk bei einer Klassifikation ist, so enthält dieses Intervall alle Events, bei denen das Netzwerk relativ unsicher ist, da die Signatur im Detektor keine großen Unterschiede aufweist. Die Verteilung der Eisen-Event ist nach wie vor im vorderen Bereich. Aber auch die Verteilung der Wasserstoff-Events erreicht das Maximum wieder dort, ähnlich zum ersten Intervall. Das dritte Intervall ist $[0.8,1]$ und enthält alle sehr gut klassifizierbaren Wasserstoff-Events, sowie ein paar wenige stark fehlklassifizierte Eisen-Events. Diese



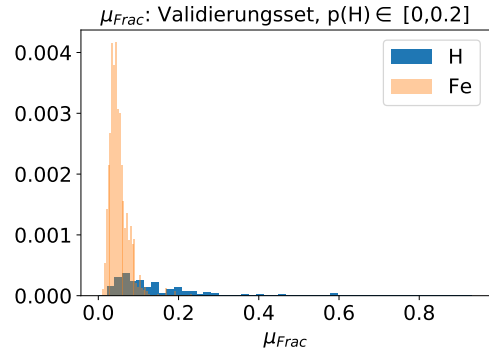
(a)



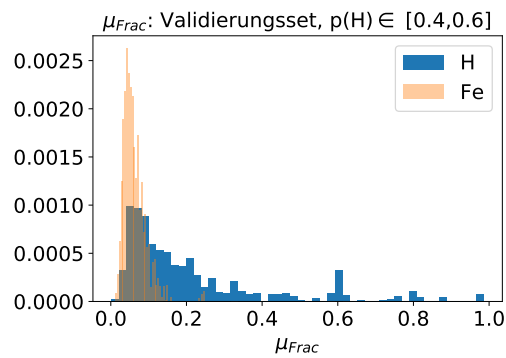
(b)

Abbildung 31: In (a) sind die beiden Histogramme für $p(H)$ aufgetragen. Dabei werden nur Events berücksichtigt, deren Primärenergie größer $1 \times 10^6 \text{ GeV}$ ist. Die H-Events sind dabei in blau, die Fe-Events in orange dargestellt. In (b) ist die entsprechende ROC-Kurve abgebildet.

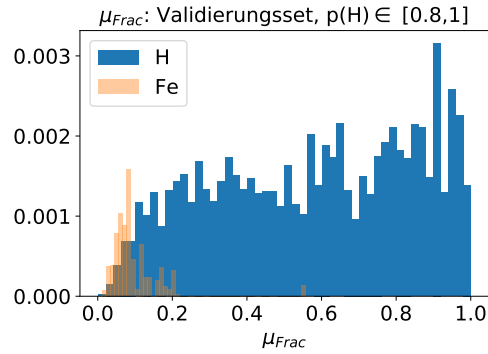
Events sind in Abbildung 32c abgebildet. Die wenigen Eisen-Events sind wie erwartet im Bereich μ_{frac} kleiner 0,2. In der Verteilung des Wasserstoffs ist auffällig, dass fast alle gut klassifizierbaren Wasserstoff-Events eine höhere μ_{frac} als 0,2 tragen. Das bedeutet, gut klassifizierbare Wasserstoff-Events haben häufig nur sehr wenige Myonen, die aber hohe Energieanteile von der Gesamtenergie dieses Myonpakets tragen. Außerdem gilt, je größer μ_{frac} , desto höher ist die vom Netzwerk vergebene Wahrscheinlichkeit $p(H)$.



(a)



(b)



(c)

Abbildung 32: In diesen drei Abbildungen sind Histogramme der μ_{frac} , aufgetragen für H und Fe. Dabei sind alle Events aus drei verschiedenen Intervallen der Wahrscheinlichkeitsverteilung aus Abbildung 29a abgebildet. In (a) für das Intervall $[0,0.2]$, in (b) für das Intervall $[0.4,0.6]$ und in (c) für $[0.8,1]$.

7 Zusammenfassung und Ausblick

In dieser Arbeit sollte anhand des Detektorsignals eines Luftschauers im IceCube Array das auslösende Primärteilchen bestimmt werden. Diese Klassifikation wurde auf Wasserstoff- und Eisen-Events beschränkt. Um das Ziel zu erreichen, wurde das verwendete Datenset hinsichtlich der Teilchenart, der Winkelverteilung, der Primärenergie und der Energieverteilung im Myonenpaket überprüft. Anschließend wurde ein neuronales Netz implementiert und unter Verwendung dieses Datensatzes trainiert. Im ungewichteten Fall wurde eine Klassifikationsrate von 66% erreicht. Nach der Gewichtung mit dem GaisserH3a-Modell auf eine realitätsnahe Verteilung, konnte das Netzwerk knapp 80% richtig klassifizieren. Das Netzwerk konnte die Primärenergie der Teilchen rekonstruieren, sowie die Energieverteilung innerhalb der Myonenpakete als physikalisch sinnvolles Klassifikationskriterium extrahieren. Die im Rahmen dieser Arbeit durchgeführte Klassifikation ist die erste Separation basierend auf den Rohdaten des IceCube Arrays. Vorherige Ansätze verwendeten durch Algorithmen hergeleitete Observablen als Eingangsinformation. Durch diese Arbeit haben sich einige Möglichkeiten ergeben, in dieser Richtung weiter zu forschen. Eine Verbesserung der Klassifikation könnte möglicherweise durch das Training auf gewichteten Daten erreicht werden. Außerdem wurden hier alle weiteren Elemente der 5-Klassen-Simulation, wie Helium, Stickstoff und Aluminium ignoriert. Eine Fortsetzung der Arbeit wäre es, die Klassifikation auf alle fünf Klassen auszuweiten.

Literatur

- [1] A. Haungs et al. *The KASCADE Cosmic-ray Data Centre KCDC: Granting Open Access to Astroparticle Physics Research Data*. Juni 2018. (Besucht am 07.10.2019).
- [2] D. Heck et al. *CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers*. 1998. URL: https://web.ikp.kit.edu/corsika/physics_description/corsika_phys.pdf (besucht am 07.10.2019).
- [3] M. Aartsen et al. „An IceCube Search for Dark Matter Annihilation in Nearby Galaxies and Galaxy Clusters“. In: *Physical Review D* 88 (Dez. 2013), S. 122001. DOI: 10.1103/PhysRevD.88.122001.
- [4] M. Tanabashi et al. *PARTICLE PHYSICS BOOKLET*. California, 2018.
- [5] M.G. Aartsen et al. *IceCube-Gen2: A Vision for the Future of Neutrino Astronomy in Antarctica*. 2014. URL: <http://inspirehep.net/record/1334495/plots#0> (besucht am 07.10.2019).
- [6] Y. LeCun et al. „Object Recognition with Gradient Based Learning“. In: (1998). URL: <http://yann.lecun.com/exdb/publis/pdf/lecun-99.pdf> (besucht am 07.10.2019).
- [7] D. Chirkin und W. Rhode. *Muon Monte Carlo: a high-precision tool for muon propagation through matter*. URL: <https://icecube.wisc.edu/~dima/work/BKP/DCS/MMC/muon.pdf> (besucht am 07.10.2019).
- [8] F. Chollet. *Deep Learning mit Python und Keras*. Frechen: mitp, 2018.
- [9] *Construction*. URL: <https://icecube.wisc.edu/science/icecube/construction> (besucht am 16.10.2019).
- [10] *Cosmic rays discovered 100 years ago*. 2012. URL: <https://home.cern/news/news/physics/cosmic-rays-discovered-100-years-ago> (besucht am 16.10.2019).
- [11] R. Engel, D. Heck und T. Pierog. *Extensive Air Showers and Hadronic Interactions at High Energy*. 2011. URL: <https://www.annualreviews.org/doi/pdf/10.1146/annurev.nucl.012809.104544> (besucht am 07.10.2019).
- [12] T. K. Gaisser. *Calculation of Muon Yields, Response Functions, and Sea Level Integral Energy Spectrum Using Recent Accelerator Data and Feynman Scaling*. 1974. URL: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JA079i016p02281> (besucht am 07.10.2019).
- [13] T. K. Gaisser. „Primary spectrum and composition with IceCube/IceTop“. In: *Nuclear and Particle Physics Proceedings* 279-281 (2016). Proceedings of the 9th Cosmic Ray International Seminar, S. 47–55. DOI: <https://doi.org/10.1016/j.nuclphysbps.2016.10.008>. URL: <http://www.sciencedirect.com/science/article/pii/S2405601416301894>.

- [14] T. K. Gaisser. „Spectrum of cosmic-ray nucleons, kaon production, and the atmospheric muon charge ratio“. In: *Astroparticle Physics* 35.12 (2012), S. 801–806. DOI: <https://doi.org/10.1016/j.astropartphys.2012.02.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0927650512000497>.
- [15] T.K. Gaisser, R. Engel und E. Resconi. *Cosmic Rays and Particle Physics*. Cambridge: Cambridge University Press, 2016.
- [16] I. Goodfellow, Y. Bengio und A. Courville. *Deep Learning*. MIT Press, 2016. URL: <http://www.deeplearningbook.org>.
- [17] S. Hochreiter und J. Schmidhuber. „Long Short-Term Memory“. In: (1997). URL: <https://www.bioinf.jku.at/publications/older/2604.pdf> (besucht am 07.10.2019).
- [18] *I3Direction*. 2019. URL: <http://software.icecube.wisc.edu/documentation/projects/dataclasses/i3direction.html> (besucht am 07.10.2019).
- [19] *IceCube - Detector*. URL: <https://icecube.wisc.edu/science/icecube/detector> (besucht am 05.10.2019).
- [20] *Keras: The Python Deep Learning library*. URL: <https://keras.io/> (besucht am 07.10.2019).
- [21] D. Kingma und J. Ba. „Adam: A Method for Stochastic Optimization“. In: *International Conference on Learning Representations* (Dez. 2014).
- [22] *Machine Learning Crash Course with TensorFlow APIs*. 2019. URL: <https://developers.google.com/machine-learning/crash-course/classification/> (besucht am 07.10.2019).
- [23] C. Olah. *Understanding LSTM Networks*. 2015. URL: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (besucht am 07.10.2019).
- [24] Wikipedia. *ROC-Kurve* — *Wikipedia, Die freie Enzyklopädie*. 2019. URL: <https://de.wikipedia.org/w/index.php?title=ROC-Kurve&oldid=191786642> (besucht am 07.10.2019).
- [25] Wikipedia. *Tscherenkow-Strahlung* — *Wikipedia, Die freie Enzyklopädie*. 2019. URL: <https://de.wikipedia.org/w/index.php?title=Tscherenkow-Strahlung&oldid=192314799> (besucht am 19.10.2019).
- [26] G. Wrede. Private Kommunikation.

Danksagung

An dieser Stelle möchte ich mich bei Allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben. Dabei gilt folgenden Personen besonderer Dank:

- Dr. Thorsten Glüsenkamp für die Bereitstellung dieser Arbeit, sowie für die fachliche Unterstützung
- Prof. Dr. Gisela Anton für die hilfreichen Tipps in den IceCube Meetings
- Gerrit Wrede, der sehr viel Zeit investiert hat und immer ein offenes Ohr für all meine Fragen hatte
- Allen aus dem Büro 341 für die schöne Zeit und die unterhaltsamen Spaziergänge zur Kaffeemaschine
- Laura Bartl, Felix Beier, Tobias Latzo, Malin Stiefler und Olaf Tönsing für das Korrekturlesen der Arbeit und die Unterstützung während des gesamten Studiums

Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe.

Erlangen, den 21. Oktober 2019

Johannes Bartl