

# **Recurrent Neural Networks und ihre Anwendung bei der Tau Leptonen Spur-Rekonstruktion und Identifikation im ATLAS-Detektor am CERN**

Paul Dinkler

Matrikelnummer: 15127903  
Georg-August-Universität  
Göttingen

9. August 2023

## **Zusammenfassung**

Der folgende Bericht soll dem Leser ein näheres Verständnis von den Algorithmen geben, die verwendet werden, um eine möglichst performante Spurklassifikation und Identifikation von Tau Leptonen in Run 3 des ATLAS-Experiments zu erreichen. Dem zu Grunde liegt eine Beschreibung von Recurrent Neural Networks, die für das Verständnis der Algorithmen unabdingbar sind und sowohl bei der Spurklassifikation als auch der Tau Leptonen Identifikation eingesetzt werden. Dabei werden einige Nachteile wie Vorteile dieser Netze diskutiert und mögliche Lösungen geschildert.

# Inhaltsverzeichnis

<b>1</b>	<b>Motivation und Überblick</b>	<b>3</b>
<b>2</b>	<b>Aufbau und Funktionsweise des ATLAS-Detektors</b>	<b>3</b>
<b>3</b>	<b>Monte Carlo Simulationen</b>	<b>4</b>
<b>4</b>	<b>Recurrent Neuronal Networks</b>	<b>4</b>
4.1	Architektur und Training . . . . .	4
4.2	Unterschied zu herkömmlichen neuronalen Netzen . . . . .	5
4.3	Probleme von RNNs . . . . .	5
4.4	RNN basierte Lösung . . . . .	6
<b>5</b>	<b>Welche Spuren ionisierender Teilchen werden <math>\tau</math> Leptonen zugeordnet?</b>	<b>6</b>
5.1	Kriterien . . . . .	6
5.2	Spurkategorien . . . . .	7
5.3	Architektur am ATLAS . . . . .	7
5.4	Training des RNNs am ATLAS . . . . .	7
<b>6</b>	<b>Wie werden Tau Leptonen korrekt identifiziert?</b>	<b>8</b>
6.1	Architektur und Training . . . . .	8
6.2	Performance . . . . .	9
<b>7</b>	<b>Fazit</b>	<b>9</b>

# 1 Motivation und Überblick

Die Teilchenphysik ist ein Fachgebiet, welches sich mit der Frage beschäftigt, woraus die Welt im Kleinsten besteht. Um diese Frage zu beantworten, werden Teilchen mit enormen Geschwindigkeiten in Teilchenbeschleunigern zur Kollision gebracht, mit dem Ziel neu entstehende Teilchen detailliert zu untersuchen. Der ATLAS-Detektor im derzeit größten Teilchenbeschleuniger, dem Large Hadron Collider, erzeugt bei jeder einzelnen Kollision Petabyte an Daten, die irgendwie verarbeitet, untersucht und gespeichert werden müssen, was mit traditionellen Analysemethoden praktisch unmöglich ist. Die Data Science ermöglicht es, diese Probleme mit fortgeschrittenen Methoden und Analysetechniken anzugehen. Um also hilfreiche Informationen und Erkenntnisse aus den Teilchenkollisionen zu ziehen, ist es essenziell das volle Potential von Simulationen, neuronalen Netzen und weiteren Tools der Data Science auszunutzen und in den Analysebereich des ATLAS-Detektors zu integrieren. Aufgrund der Relevanz dieser Thematik ist Ziel des folgenden Berichts einen Überblick über den in Run 3 des ATLAS Experiments verwendeten Algorithmus zu schaffen, der hadronisch zerfallende  $\tau$  Leptonen rekonstruiert und identifiziert. Das  $\tau$  Lepton ist mit einer Masse von 1777 MeV das schwerste der drei Leptonenarten im Standardmodell der Teilchenphysik [1]. Es besitzt eine Halbwertszeit von  $2.9 \times 10^{-13}$  s und ist somit sehr viel instabiler als das Myon [9]. Dies ist auch der Grund, warum eine direkte Messung von  $\tau$  Leptonen nur schwer möglich ist und weswegen auf andere Methoden, wie die Spuranalyse von  $\tau$  Leptonen zurückgegriffen wird.

Der Schwerpunkt dieses Berichts liegt auf der  $\tau$  Leptonen Identifikation und Spurklassifikation, weil  $\tau$  Leptonen uns sehr viele Informationen über das Standardmodell der Teilchenphysik liefern können. Zum Einen hilft die genauere Untersuchung von Tau Leptonen, die bisherigen Vorhersagen des Standardmodells zu verifizieren oder zu falsifizieren [3]. Zum Anderen ist der Zerfall des Higgs Bosons in ein Tau und ein Anti-Tau Teilchen ein wichtiger Zerfall, um einige Charakteristika des Higgs Bosons auf ihre Richtigkeit hin zu überprüfen [3]. Einige Theorien der Teilchenphysik postulieren außerdem weitere Teilchen jenseits des Standardmodells, von denen einige ebenfalls mit Hilfe des Tau Leptons bzw. seines Zerfalls entdeckt werden könnten [3]. Tau Leptonen spielen also eine sehr wichtige Rolle bei Teilchenkollisionen, weswegen es um so wichtiger ist die Algorithmen, die für die Analyse von Proton-Proton-Kollision (PP-Kollision) verwendet werden, stetig zu optimieren und weiter zu entwickeln.

## 2 Aufbau und Funktionsweise des ATLAS-Detektors

Die detaillierte Beschreibung des ATLAS-Detektors und der Monte Carlo Simulationen ist nicht Ziel dieses Berichts, sodass sie nur grob beschrieben werden. Eine genauere Veranschaulichung beider Themen finden sie in [1]. Der ATLAS-Detektor besteht aus vier Systemen, die zwiebelschalenartig angeordnet sind, und ermöglicht geladene Teilchen bis zu einer Pseudorapidität von  $|\eta| < 2.5$  zu verfolgen. Der Ausdruck Pseudorapidität ist eine Größe in der Teilchenphysik, um die Richtung eines Vektors relativ zur Strahlenachse zu beschreiben. Die Pseudorapidität ist definiert als  $\eta = -\ln[\tan(\theta/2)]$ . Dies gelingt, da geladene Teilchen einen ionisierten Raum zurücklassen. Dieser wird als Spur betrachtet und ist Grundlage für die Tau Leptonen Identifikation.

- Das Magnetsystem ist die Ursache für das 2T starke, nahezu komplett homogene Magnetfeld im Beschleuniger. Es lenkt die geladenen Teilchen im Inneren ab, wodurch deren Impuls bestimmt werden kann.
- Der Innere Detektor besteht aus drei weiteren Subdetektoren, die für den Spurnachweis der Teilchen verantwortlich sind. Der innerste Subdetektor ist der Pixeldetektor (IBL), der vier Spurpunkte pro ionisierendem Teilchen bereitstellt. Um diesen herum liegt der Silizium-Streifen-Detektor (SCT), der acht Spurpunkte bereitstellt. Die äußerste Schicht bildet der Übergangsstrahlungsspurdetektor (TRT), welcher etwa 30 Spurpunkte pro durchgehendem ionisierendem Teilchen registriert. Außerdem kann mit dem Nachweis der Übergangsstrahlung zwischen Elektronen und Hadronen unterschieden werden.
- Das Kalorimetersystem besteht aus den zwei Komponenten Elektromagnetisches Kalorimeter und Hadronisches Kalorimeter und zeichnet Spuren bis zu einer Pseudorapidität von  $|\eta| < 4.9$  auf. Beide Kalorimeter benutzen flüssiges Argon als Detektormaterial. Das Hadronische Kalorimeter bestimmt die Energie der Hadronen und das Elektromagnetische Kalorimeter bestimmt Energie und Impuls von elektromagnetisch wechselwirkenden Teilchen.

- Das Myonen-Spektrometer besteht aus einer Triggerkomponente, die physikalisch interessante Signaturen (im Zusammenhang mit Myonen) markieren soll und einer weiteren hochauflösenden Komponente, die die Spuren und den Impuls der Myonen misst, indem die Ablenkung der Myonen im magnetischen Feld beobachtet wird.

### 3 Monte Carlo Simulationen

Eine genaue Beschreibung liegt in [1] vor. Monte Carlo Simulationen spielen in der Teilchenphysik eine wichtige Rolle und werden unter anderem am ATLAS-Detektor eingesetzt, um Datensätze zu generieren, mit denen die Recurrent Neural Network (RNN) basierte  $\tau$ -Identifikations- und Spur-Rekonstruktions-Algorithmen trainiert werden. Der simulierte Signal-Datensatz wird mit der Software Pythia generiert und besteht aus  $\gamma^* \rightarrow \tau\tau$  Zerfällen. Pythia simuliert den  $\tau$  Leptonen Zerfall, die Hadronisierung, Partonen-Schauer und Untergrund-Ereignisse, um tatsächliche Kollisionen so genau wie möglich zu beschreiben. Alle generierten  $\tau$  Leptonen zerfallen in der Simulation hadronisch, da der Algorithmus speziell auf diesen Zerfall trainiert werden soll. Pythia wird außerdem dafür benutzt, Hintergrundereignisse aus "dijet"-Ereignissen zu simulieren, damit der verwendete Algorithmus optimiert werden kann. "Dijet"-Ereignisse beschreiben Kollisionen, bei denen zwei Jets bzw. kollimierte Strahlen von Teilchen entstehen. Geant4, ein weiteres Softwarepaket, wird dafür verwendet die Antwort vom Detektor zu reproduzieren, welcher die Energieabgabe von bestimmten Teilchen und Zerfällen kalkuliert. Zusammenfassend werden MC-Simulationen gebraucht, um das Verhalten von  $\tau$  Leptonen zu modellieren, Multivariate-Analyse-Algorithmen zu trainieren und ihre Performance zu evaluieren. Diese Algorithmen werden anschließend am Detektor zur Analyse von echten Datensätzen eingesetzt.

### 4 Recurrent Neuronal Networks

Die folgenden Abschnitte beschäftigen sich näher mit Recurrent Neural Networks (RNNs), weil sie eine essenzielle Rolle am ATLAS-Detektor spielen. Hier folgt zunächst eine allgemeine Beschreibung. Die Anwendung am ATLAS ist in späteren Kapiteln beschrieben.

#### 4.1 Architektur und Training

Recurrent Neural Networks sind eine bestimmte Art von neuronalen Netzen, die in der Lage sind Informationen aus früheren Eingaben und Ausgaben zu speichern. Des eignet sie hervorragend für sequentielle Datensätze, bei denen die Reihenfolge der Daten ein Rolle für die Ausgabe spielt [7]. Gespeichert werden die Ein- und Ausgaben in dem "hidden state", in der verdeckten Schicht. Traditionelle RNNs sind nur in der Lage unidirektional auf Informationen zuzugreifen, was bedeutet, dass sie dahingehend beschränkt sind nur vergangene Informationen mit einzubeziehen. Wie später noch erläutert wird, existieren auch Variationen von RNNs, die bidirektional arbeiten, also auch zukünftige Datenpunkte in die Analyse einbeziehen. Allgemein betrachtet besteht ein RNN aus den drei Schichten: Eingabeschicht (engl. input layer), verdeckte Schicht (engl. hidden layer) und Ausgabeschicht (engl. output layer), siehe Bild 1.

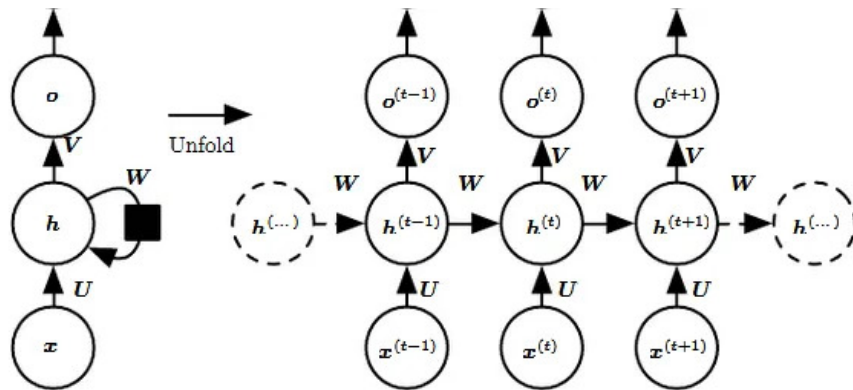


Abbildung 1: Beispielarchitektur eines RNNs [5]

Funktionsweise, siehe [5]:

Die Eingabeschicht des RNNs erhält eine Eingabe  $x(t)$ , die normalerweise als Vektor repräsentiert wird, zu einem bestimmten Zeitpunkt  $t$ .  $x(t)$  wird anschließend an die verdeckte Schicht des Netzwerks weitergeleitet, wo  $x(t)$  zu  $h(t)$  verrechnet wird.  $h(t)$  errechnet sich mit  $f(b + U \times x(t) + W \times h(t - 1))$ .  $U$  und  $W$  sind Gewichte, bzw. Matrizen, die  $x(t)$  und  $h(t - 1)$  skalieren, und  $b$  ist ein Bias-Term, der die Ausgabe reguliert, indem beispielsweise auch dann ein Ergebnis erzeugt wird, wenn alle Parameter null sind.  $f$  ist eine nichtlineare Transformation wie z.B.  $\tanh$ . Die verdeckte Schicht ist der entscheidende Teil eines RNNs, der die Fähigkeit zur Verarbeitung von Sequenzen bietet. Diese Schicht besteht aus mehreren Zellen, die allerdings in verschiedenen Formen auftreten. Ein Beispiel sind "Long Short Term Memory"-Zellen, siehe unten. Anschließend wird in der Ausgabeschicht  $h(t)$  erneut mit einer Matrix  $V$  multipliziert und einem Bias-Term  $c$  addiert. Die Funktion wird mit  $o(t)$  dargestellt, siehe 1.

Nimmt das RNN mehrere Klassenklassifikation vor, wird anschließend eine softmax-Funktion verwendet, die die Ausgabe jedes Neurons so skaliert, dass die Summe der Ausgaben aller Neuronen 1 ist. Somit stellt die Ausgabe eines einzelnen Neurons die Wahrscheinlichkeit dafür dar, dass der Fall einer spezifischen Klasse eintritt. Dieser Prozess wird "Vorwärts-Propagation" (engl. Forward propagation) genannt. Nachdem die Vorwärts-Propagation abgeschlossen ist, wird der Fehler des prognostizierten Wertes errechnet. Dieser Fehler wird anschließend in der verdeckten Schicht, durch die Zeit zurück propagiert (engl. Backward Pass). Dies geschieht, indem das Modell beim letzten Schritt beginnt und sich dann schrittweise zurück zum ersten Zeitschritt bewegt. Dabei wird der Fehler zuerst in der Ausgabeschicht berechnet und anschließend in den versteckten Zustand und die jeweilige rekurrente Zelle zurück propagiert. Während dessen werden die Gradienten des Fehlers berechnet, indem die partiellen Ableitungen der Fehlerfunktion bezüglich der Gewichte und Schwellenwerte gebildet werden. Diese werden verwendet um die Gewichte zu aktualisieren. Dies geschieht mit einem Optimierungs-Algorithmus über mehrere Zeitschritte. Dieser Prozess wird für jede Sequenz in der Trainingsdatenmenge wiederholt. Um Überanpassung (engl. overfitting) zu vermeiden wird beim Training der Datensatz in Trainings-, Validierungs- und Testdatensätze aufgeteilt. Dabei wird mithilfe des Validierungsdatensatzes die Leistung des Modells überwacht um Überanpassung zu vermeiden, nachdem das Netzwerk mit den Trainingsdaten trainiert wurde. Mit Hilfe des Testdatensatzes wird die Performance des RNNs evaluiert.

## 4.2 Unterschied zu herkömmlichen neuronalen Netzen

Herkömmliche neuronale Netze oder auch Feedforward-Neuronale Netze (hier FNNs) und RNNs unterscheiden sich in ihrer Architektur und ihrer Fähigkeit zeitabhängige oder sequenzielle Daten zu verarbeiten, siehe [6]. FNNs sind gerichtete Graphen, die Informationen nur in eine Richtung fließen lassen, wodurch Rückkopplungen nicht möglich sind. Das bedeutet, dass Informationen aus vorherigen Schritten nicht wiederverwendet werden können. RNNs hingegen besitzen diese Fähigkeit, da sie mithilfe von Schleifen die Informationen beidseitig weiterleiten und verarbeiten können. Aufgrund dessen ist es ihnen möglich kontextabhängige Muster in den Eingabesequenzen zu erkennen. Es ist sogar möglich, dass die Ausgabe eines Neurons als Eingabe desselben Neurons oder eines anderen Neurons in derselben Schicht dient. Somit hängt die Ausgabe eines RNNs nicht nur von der aktuellen Eingabe, sondern auch von seinem vorherigen Zustand ab. Aufgrund dessen werden RNNs vor allem bei der zeitabhängigen Verarbeitung von Daten verwendet und so auch am ATLAS-Detektor. FNNs werden hauptsächlich für statische Datensätze benutzt, bei denen die Reihenfolge der Datenpunkte keine Rolle spielt, siehe [1].

## 4.3 Probleme von RNNs

Wie bereits angesprochen werden RNNs meist für sequenzielle Daten verwendet, um Vorhersagen zu treffen. Bestimmte Szenarien stellen allerdings Schwierigkeiten bereit, wenn RNNs Abhängigkeiten über große Distanzen einer Sequenz erlernen, siehe [4]. Hier kommt es häufig vor, dass während des Rückkopplungs-Prozesses (engl. Backpropagation) die Gradienten, also die Werte, die die Gewichte der Knoten im Netzwerk updaten, entweder gegen null laufen oder explodieren. Dies macht den Lernprozess des Modells über große Distanzen sehr schwer, weil sich während der Rückkopplung die Werte entweder nicht ändern oder eine Multiplikation mit so einer großen Zahl zu einem irrationalen Ergebnis führt. Des Weiteren besitzen standardisierte RNNs ein limitiertes Kurzzeitgedächtnis, weshalb der Zugriff auf weit zurückliegende Informationen sehr schwer sein kann. Ist der Speicher voll, wird die älteste Information gelöscht und mit einer neuen Information ersetzt. LSTM-Zellen versuchen dieses Problem zu lösen.

Ein weiterer Nachteil von RNNs ist, dass RNNs wenige Optionen zur Parallelisierung von Aufgaben zulassen, was zu einer signifikanten Verminderung der Trainings- und Verarbeitungsgeschwindigkeit von Informationen führt. Da die Ausgabe des RNNs zu einem bestimmten Zeitpunkt auf der Ausgabe zu einem anderen Zeitpunkt basiert, ist ein Parallelisieren von Aufgaben nur sehr beschränkt möglich [4].

## 4.4 RNN basierte Lösung

Eine sehr verbreitete Methode, um das Problem des Verschwindens oder Explodierens der Gradienten zu beheben, ist das Implementieren von “Long Short Term Memory“-Zellen [4]. LSTM-Zellen sind praktisch eine weitere Schicht an Zellen im Netzwerk, die in ihrem Zellen Status wichtige Informationen permanent speichern. Mit Hilfe dieser Informationen können langfristige Abhängigkeiten in Sequenzen besser erfasst werden, weswegen sie auch am ATLAS bei der  $\tau$  Spurklassifikation und Identifikation eingesetzt werden. Die Funktionsweise lässt sich in folgende Komponenten unterteilen, siehe [8]:

- Das “Input-Gate“ implementiert eine Sigmoid-Funktion, mit dessen Hilfe es steuert, welche Informationen aus der Eingabe in den Zellzustand übernommen werden sollen, um irrelevante Information zu eliminieren.
- Das “Forget-Gate“ wird ebenfalls mit einer Sigmoid-Funktion gesteuert und entscheidet, welche Informationen im aktuellen Zellzustand vergessen werden sollen.
- Das “Output-Gate“ steuert ebenfalls auf Grundlage einer Sigmoid-Funktion die Selektion der relevanten Information, die zur Ausgabe bzw. zur weiteren Verwendung im nächsten Schritt verwendet werden sollen.
- Der Zellzustand ist der interne Speicher der LSTM-Zelle und speichert langfristig Informationen aus der Vergangenheit, die durch das “Input-Gate“ und das “Forget-Gate“ gesteuert werden. Er wird stetig aktualisiert.

Die Speichermöglichkeit von langfristigen Informationen ist auch der Grund, weswegen Abhängigkeiten flexibel und aktuell erkannt werden können.

Eine Erweiterung dieser Architektur stellen “Bi-directional Long Short Term Memory“-Zellen dar. Sie sind ebenfalls am ATLAS-Detektor implementiert und haben den Vorteil, dass Informationen aus beiden Richtungen der Eingabesequenz genutzt werden können. Das bedeutet, dass sie in der Lage sind Informationen aus der Vergangenheit und aus der Zukunft für die Analyse zu nutzen [4].

## 5 Welche Spuren ionisierender Teilchen werden $\tau$ Leptonen zugeordnet?

Da  $\tau$  Leptonen eine sehr geringe Halbwertszeit haben, ist es schwer möglich sie direkt am Detektor zu messen. Allerdings ist es möglich die Spuren ihrer Zerfallsprodukte zu messen. Der folgende Abschnitt beschäftigt sich mit der Frage, welche Spuren als  $\tau_{had-vis}$  Kandidat kategorisiert werden, siehe [1].

### 5.1 Kriterien

Es gibt verschiedene Kriterien, die erfüllt sein müssen, damit eine Detektorspur zu einem  $\tau_{had-vis}$  (hadronisch zerfallendem  $\tau$  Lepton) zugeordnet werden kann. Zunächst einmal müssen sich die Spuren des Detektors innerhalb einer Kernregion (engl. core region) von  $\delta R < 0.25$  befinden. Spuren deuten in diesem Zusammenhang auf die Flugbahn geladener Teilchen bzw. Hadronen hin, da wir uns auf den hadronischen Zerfall des  $\tau$  Teilchens beziehen. Die Kernregion ist ein bestimmter Bereich um die Richtung des Tau jets (Ansammlung von  $\tau$  Leptonen mit einer Richtung und einer Energie) herum, die mit dem Impuls und der Energie bestimmt wird.  $\delta R$  ist der Winkelabstand zwischen dem  $\tau_{had-vis}$  Kandidaten und den Spuren. Er wird unter Verwendung der Pseudorapidität und des Azimutwinkels beider Objekte berechnet. Außerdem muss der gemessene transversale Impuls  $p_T$  größer als 1 GeV sein, es muss mindestens zwei Messungen auf der Pixel Ebene des Inneren Detektors und mindestens sieben Messungen insgesamt geben. Darüber hinaus darf die senkrechte Abweichung der Spur zum Tau Produktions-Vertex  $|z_0^{TJV A} \sin\theta|$ , also dem Vertex in dem das  $\tau$  Teilchen nach dem TJVA Algorithmus entsteht, nicht größer als 1.0 mm und longitudinal  $|d_0^{TJV A} \sin\theta|$  nicht größer als 1.5 mm sein. Sind diese Kriterien erfüllt, wird die Spur einem Tau Teilchen zugeordnet. Liegt  $\delta R$  zwischen 0.25 und 0.4, können die Spuren ebenfalls zu einem  $\tau$  Teilchen zugeordnet werden, falls der dichteste Jet zu der Spur der “seeding jet“ ist oder falls die Spuren durch eine Geisterteilchen-Assoziierungstechnik (engl. ghostparticle association technique) mit dem “seeding jet“ in Verbindung gebracht werden können. Der “seeding jet“ ist ein Ansatz für die Spurrekonstruktion, bei dem als Ausgangspunkt ein jet verwendet wird, um die Spur zu erhalten. Er dient als Schätzung für den Startpunkt der Spur. Die Geisterteilchen-Assoziierungstechnik ermöglicht es, durch Annahme von vorläufigen Spuren (Geisterteilchen) eine Schätzung der Flugbahn der Teilchen zu erhalten und anschließend mit den gemessenen Werten im Detektor abzugleichen, um die tatsächliche Flugbahn zu erhalten.

## 5.2 Spurkategorien

Die Spuren, die diese Kriterien erfüllt haben, werden anschließend mit Hilfe eines RNNs in vier Kategorien aufgeteilt, damit die Zerfallsprodukte des  $\tau$  Teilchens korrekt von anderen Zerfallsprodukten separiert werden können. Die vier Spurkategorien sind: “Tau Tracks“, “Conversion Tracks“, “Isolation Tracks“ und “Fake Tracks“.

“Tau Tracks“ repräsentieren die Spuren, die direkt aus den Zerfallsprodukten des  $\tau$  Teilchens entstehen. Sie werden verwendet, um die Ladung des Tau Teilchens und die Anzahl seiner Zerfallsprodukte herauszufinden.

“Conversion Tracks“ beziehen sich auf die Elektronen und Positronen Spuren, die durch Photonen-Umwandlungen im Detektor entstehen. Sie werden sowohl für die Rekonstruktion und Identifikation von Pionen als auch die Ablehnung von Elektronen verwendet, die fälschlicherweise als Tau Kandidat kategorisiert wurden.

“Isolation Tracks“ sind Spuren, die aus Quark- oder Gluonenjets entstehen. Diese Spuren bestehen tendenziell aus mehr Teilchen mit einem geringeren  $p_T$  Wert als die Spuren aus  $\tau_{had-vis}$  Zerfällen. Außerdem ist die Streuung der Teilchen größer, was auf einen breiteren Öffnungswinkel schließen lässt.

“Fake Tracks“ sind Spuren, die aufgrund von falscher Rekonstruktion oder “Pile-up Tracks“, also Spuren aus sekundären PP-Kollisionen entstehen. Sie sind keiner der vorherigen Kategorien zuzuordnen.

## 5.3 Architektur am ATLAS

In Run 2 wurde ein “Boosted Decision Tree“ für die Tau Leptonen Spur Klassifikation verwendet. Dieser ist allerdings nicht optimal, weswegen man sich in Run 3 für ein RNN mit BLSTM-Zellen entschieden hat. Diese haben zum Einen den Vorteil, dass die relevanten Spurinformatoren nicht nur unidirektional in Richtung Ausgabe-Schicht weitergetragen werden, sondern auch rückwärts in Richtung Eingabe-Schicht, weswegen die Informationen über die Zeit hinweg zurück propagiert werden können. Dadurch kann das Modell Korrelationen bidirektional erfassen. Zum Anderen sind RNNs mit BLSTM Zellen bei der Eingabe nicht auf normierte Sequenzen, mit gleicher Größe limitiert, weswegen die Spuren dynamisch als variate Variable behandelt werden können.

Die RNN Architektur am ATLAS-Detektor besteht aus einer Eingabe-Schicht, drei vollständig verbundenen Schichten (engl. dense layers), drei BLSTM-Schichten, die sich zwischen den ersten beiden “dense layers“ befinden, und einer Ausgabe-Schicht. Die Eingabe-Schicht erhält als Variablen die jeweiligen Spuren, wobei maximal 30 Spuren pro Tau Kandidat betrachtet werden, und ist verbunden mit der ersten vollständig verbundenen Schicht. Diese besteht aus 80 Knoten, die zweite aus 25 Knoten und die dritte aus 4 Knoten. In der Ausgabe-Schicht werden die jeweiligen Wahrscheinlichkeiten der 4 Spurenklassen ausgegeben. Bis auf den letzten “dense layer“ sind alle Schichten mit der “Rectified Linear Unit“-Aktivierungsfunktion (ReLU) verbunden. Sie ist sehr simple als  $Relu(x) = \max(0, x)$  definiert und ändert alle negativen Werte zu 0, um Nichtlinearitäten in das Modell einzuführen. Diese helfen komplexe Zusammenhänge zwischen Eingabe und Ausgabe zu erkennen.

## 5.4 Training des RNNs am ATLAS

Trainiert wird das RNN mit den Bibliotheken Keras und Tensorflow und mit Hilfe von Monte Carlo simulierten Datensätzen. Die Signal-Trainingssätze bestehen aus  $\gamma * - > \tau\tau$  Reaktionen, wobei nur die  $\tau_{had-vis}$  Kandidaten mit  $|p_T| > 5.5$  in Erwägung gezogen werden, und die Hintergrund-Trainingssätze aus “dijet“-Events. Die “dijet“-Events sind in 8  $p_T$  Regionen und zusätzlich in “hard Scatter“ (Interaktion von hochenergetisierten Quarks oder Gluonen) und “pile-up“-jets aufgeteilt. Beide Datensätze werden in Trainingsdaten, Validierungsdaten und Testdaten aufgeteilt, wobei es 1 Millionen Signal und 200000 Hintergrunddaten gibt.

Die folgenden Variablen beschreiben ausschlaggebende Charakteristika einer Spur, wie die kinematische oder geometrische Verteilung der Spur am Detektor, und werden ausgewählt, da sie entweder das Netzwerk parametrisieren oder die Differenz der Spurkategorien leichter erkennbar machen:

$\ln(p_T^{track})$ : Natürlicher Logarithmus des transversalen Impulses der Spur.

$\ln(p_T^{seed-jet})$ : Natürlicher Logarithmus des transversalen Impulses des “seed jets“.

$\frac{p_T^{track}}{p_T^{seed-jet}}$ : Quotient zwischen dem transversalem Impuls der Spur und des “seed jets“.

$\eta^{track}$ : Pseudorapidity der Spur.

$z_0^{TJVA} \sin\theta$ : longitudinaler Tau Vertex Parameter.

$\ln(r^{Conv})$ : Natürlicher Logarithmus des Radius der Spur-Umwandlung.

$\tanh(\frac{r^{ConvII}}{500})$ : Tangens Hyperbolicus des Radius der Spur-Umwandlung.

$\delta R(track, seed - jet)$ : Radiale Distanz der Spur und "seed jet" Achse.

$\tanh(\frac{d_0^{TJVA}}{10})$ : Tangens Hyperbolicus des transversalen Einfluss-Parameters.

$\frac{q^{track}}{p_T^{track}}$ : Quotient von transversalen Impuls und Ladung der Spur.

Die Präzision des RNNs wird hauptsächlich anhand der  $\tau_{had-vis}$  Rekonstruktions-Effizienz und der Fehlermatrix der Klassifikationen bemessen. Letztere ist eine quadratische Matrix, die angibt, wie präzise ein Klassifikator (hier das RNN) eine bestimmte Klasse vorhersagt.

## 6 Wie werden Tau Leptonen korrekt identifiziert?

Die Spur-Rekonstruktion liefert alle möglichen  $\tau$  Kandidaten. Zu Beginn von Run 2 des ATLAS Experiments, wurden die wahren  $\tau_{had-vis}$  Kandidaten von den falsch identifizierten mit einem "Boosted Decision Tree" Ansatz getrennt. Es hat sich allerdings gezeigt, dass ein RNN die falsch identifizierten Kandidaten um 75% bis 100% besser zuordnen konnte, weswegen man sich noch in Run 2 dazu entschieden hat, ein RNN für die  $\tau$  Identifizierung einzusetzen. Der folgende Abschnitt beschäftigt sich sowohl mit dem Training des RNNs als auch mit seiner Performance, siehe [1] und [2].

### 6.1 Architektur und Training

Die Architektur des verwendeten RNNs für die Identifikation von  $\tau$  Leptonen entspricht der des RNNs für die Spurklassifikation, wobei die Eingabeschicht auf drei verschiedene Äste aufgeteilt ist. Einer für Spur-Variablen, einer für "Cluster"-Variablen und einer für "High level jet"-Variablen. "Cluster"-Variablen beschreiben Charakteristika von Teilchengruppen wie beispielsweise ihre Energiesumme oder die Anzahl der vorhandenen Teilchen. "High level jet"-Variablen sind Eigenschaften von jets, die auf einer fortgeschrittenen Analyseebene betrachtet werden.

Das RNN wird mit Datensätzen von  $\tau_{had-vis}$  Kandidaten trainiert, die durch Monte Carlo Simulationen entstehen. Damit eine Spur aus einem  $\gamma^* \rightarrow \tau\tau$  Zerfall als  $\tau$  Spur identifiziert werden kann, muss die Spur des rekonstruierten  $\tau$  Kandidaten mit den im Training simulierten Spuren übereinstimmen. Außerdem muss der rekonstruierte  $\tau_{had-vis}$  Kandidat korrekt als "1-prong"-oder "3-prong"-Zerfall eingestuft werden. Das bedeutet, dass die Anzahl der sichtbaren Spuren entweder 1 oder 3 sein muss. Beide weisen jeweils verschiedene Signaturen am Detektor auf, weswegen das Training für beide getrennt durchgeführt wird.  $\tau_{had-vis}$  Kandidaten, die nicht aus  $\tau_{had}$  Zerfällen entstehen, werden aus simulierten "dijet"-Datensätzen gezogen und müssen ebenfalls als "1-prong"-oder "3-prong"-Ereignisse festgestellt werden.

Damit das Signal effizient vom Hintergrundrauschen getrennt werden kann, werden die  $\tau$  Leptonen-Spuren aus den Hintergrundprozessen neu gewichtet, indem die beobachtete  $p_T$  Verteilung der wahren  $p_T$  Verteilung aus den simulierten  $\gamma^* \rightarrow \tau\tau$  Zerfällen angepasst wird. Die Anpassung hat zur Folge, dass das RNN nicht anhand von Verteilungen mit einem zusätzlichen Bias-Term trainiert wird, da die Hintergrunddaten tendenziell geringere  $p_T$  Werte als die Signaldaten aufweisen. Der Trainingsdatensatz besteht aus ca 14 Millionen Signal- und 83 Millionen Hintergrund  $\tau_{had-vis}$ -Kandidaten nach der Selektion und wird in Trainingsdaten(40%), Testdaten(10%) und Daten zur Fortschrittsvalidierung(50%) aufgeteilt.

Genau wie bei der Spurklassifikation, wird das RNN mit Keras und Tensorflow trainiert.

Bei einer tatsächlichen Analyse ordnet das RNN jedem  $\tau$  Kandidaten einen Wert zu und transformiert ihn, sodass er zwischen 0 und 1 liegt. Dieser Wert repräsentiert die Likelihood, dass der Kandidat einem echten  $\tau$  Lepton entspricht und stellt eine konstante Skalierung der  $p_T$  und  $< \mu >$  Werte sicher. Die "Working points" bzw. "thresholds" werden anhand der transformierten Werte angepasst.

Das Modell wird mit der Minimierung der "binary cross-entropy loss" Funktion trainiert, welche den Fehler zwischen der Vorhersage der binären Klassifikation von "ist Tau Teilchen" und "ist kein Tau Teilchen" be-



rechnet. Die Minimierung erfolgt mit dem Optimierungsalgorithmus “stochastic gradient descent(SGD)“. Dieser passt alle Parameter des Modells schrittweise an, indem der negative Gradient der Verlustfunktion, der die Richtung angibt, in der die Verlustfunktion am schnellsten abnimmt, in Bezug auf die Parameter verwendet wird. Das Modell wird solange trainiert, bis sich der Verlust innerhalb von 10 Durchläufen nicht weiter verbessert.

## 6.2 Performance

Die Performance des RNNs wird im Vergleich zu dem BDT-Ansatz der Tau Identifikation evaluiert. Dies geschieht, indem statistisch unabhängige Tests für  $\gamma * - > \tau\tau$  signal und “dijet“-Hintergrunddatensätze durchgeführt und anschließend mit der Performance des BDTs abgeglichen werden. Dabei ist die Effizienz des RNNs ungefähr doppelt so gut, wie die des BDTs.

Die “Working points“, bzw. “thresholds“ sind für “1-prong“ und “3-prong“ Ereignisse unterschiedlich zu validieren, allerdings sind für beide die “working points“ in “Tight“, “Medium“, “Loose“ und “Very loose“ mit abnehmender Hintergrund-Ablehnung aufgeteilt. Tendenziell sind die “thresholds“ für “3-prong“ Ereignisse ein paar Prozent weiter unten angesetzt, so ist “Tight“ bei “1-prong“ Ereignissen 60% und bei “3-prong“ Ereignissen 45%. Dies liegt daran, dass man hier versucht, die Rate an falsch identifizierten  $\tau_{had}$  Kandidaten von “dijet“-Ereignissen zu reduzieren. Diese Rate hat sich im Vergleich zu dem BDT-Ansatz in dem  $p_T$  Intervall von 30-50 GeV um 75% und bei höheren GeV-Werten sogar um 100% verbessert. Erstaunlich ist ebenfalls, dass die durchschnittliche Spur Assozierungseffizienz von 70% auf über 90% (von 65% auf 75%) für “1-prong“ (“3-prong“) Ereignisse, im Vergleich zu dem anfänglichen Ansatz, stieg. Wichtig zu verstehen ist noch, dass die tatsächliche Ablehnungseffizienz stark von den  $p_T$ ,  $\eta$  und  $\mu$  Verteilungen und der Quark-Gluonenstruktur des Hintergrund-Datensatzes abhängt, wobei die Abhängigkeit zu den Verteilungen signifikant geringer für schwächere “working points“ ist.

## 7 Fazit

Zusammenfassend ist die Verwendung von RNNs gegenüber BDTs ein erheblicher Fortschritt im Hinblick auf die Effizienz der Datenverarbeitung. Das liegt vor allem an der Architektur von RNNs, die es ihnen durch ihre Schleifenstruktur ermöglicht, vergangene Informationen zu speichern und für spätere Analysen wieder zu verwenden. Diese Eigenschaft ist bei der Verwendung von “Time series data“, wie die am ATLAS-Detektor essenziell, da der Zerfall von Teilchen ein zeitlicher Prozess ist und nur durch eine zeitliche Beobachtung korrekt evaluiert werden kann. Zwar lässt sich zeigen, dass im Allgemeinen “Boosted Descision Trees“ sehr effizient bei großen Datensätzen und robust gegenüber Ausreißern sind. Allerdings haben jüngste Ergebnisse gezeigt, dass diese Eigenschaften nicht ausreichen um die Performance von RNNs zu schlagen. Es gibt jedoch Komponenten, die weiterentwickelt werden können und müssen, um die verwendeten Modelle und Algorithmen zu verbessern. So lässt sich zum Beispiel überlegen, das Problem von Abhängigkeiten über sehr lange Sequenzen mit dem Ansatz von Transformern anstatt von LSTM-Zellen zu lösen. Transformer sind eine Kategorie von recht jungen KI-Modellen, die durch ihren Attention-Mechanismus jedes Element einer Sequenz auf alle anderen Elemente aufmerksam machen können. So lassen sich selbst weit zurück liegende Information in die Analyse eines bestimmten Zeitschritts einbeziehen. Transformer sind nur eine der Möglichkeiten die noch nicht implementiert sind und es gilt eine allgemeine Zuversicht, dass in den nächsten Jahren immer neue Modelle hinzukommen die noch effizienter an die Probleme angepasst sind.

Das Feld der Teilchenphysik und die Erkenntnisse, die wir erhalten, sind also nicht weniger stetig als die sich ständig weiterentwickelnden Methoden, mit deren Hilfe wir eine neue Sicht auf die Welt erlangen können.

## Literatur

- [1] <https://cds.cern.ch/record/2827111/files/ATL-PHYS-PUB-2022-044.pdf>
- [2] <http://cds.cern.ch/record/2688062/files/ATL-PHYS-PUB-2019-033.pdf>
- [3] [https://www.fsp103-atlas.de/sites/sites\\_custom/site\\_fsp103-atlas/content/e17637/e25959/e25977/FSP103\\_Newsfla](https://www.fsp103-atlas.de/sites/sites_custom/site_fsp103-atlas/content/e17637/e25959/e25977/FSP103_Newsfla)
- [4] <https://databasecamp.de/en/ml/recurrent-neural-network>
- [5] <https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85>
- [6] <https://stats.stackexchange.com/questions/2213/whats-the-difference-between-feed-forward-and-recurrent-neural-networks>
- [7] <https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/>
- [8] <https://www.bigdata-insider.de/was-ist-ein-long-short-term-memory-a-774848/>
- [9] <https://www.spektrum.de/lexikon/astronomie/myonen/308>