



Hintergrund: CERN

KÜNSTLICHE INTELLIGENZ

Deep Learning

Erkenntnisgewinn durch moderne datengetriebene Methoden

Martin Erdmann

Die Entwicklung neuer Technologien ist entscheidend für Erkenntnisgewinne in der Grundlagen- und angewandten Forschung. Fortschritten bei modernen Datenanalyseverfahren kommt dabei besondere Bedeutung zu: Sie sind der Schlüssel, um neue Erkenntnisse aus Messdaten zu schöpfen und entscheidenden Mehrwert aus Messapparaturen zu erwirken.

In der Physik werden aktuell Deep Learning-Verfahren bei Fragestellungen erprobt, bei denen sich ein passender Algorithmus nicht so einfach programmieren lässt. Hier weisen Deep Learning-Verfahren häufig Vorteile gegenüber bisher verwendeten Analyseverfahren auf, was zu präziseren wissenschaftlichen Resultaten führt, Erkenntnisprozesse zumindest beschleunigt oder eventuell erst ermöglicht.

Deep Learning beruht auf neuronalen Netzwerken (vgl. dazu die Artikel von Ulrich Eberl und Marco F. Huber sowie die **Infoseite** „Neuronale Netzwerke kurz erklärt“) [1].

Mit ihnen lassen sich flexible Modelle für beliebige funktionale Zusammenhänge realisieren (**Abb. 1**).

Nüchtern betrachtet und sehr verkürzt gesprochen verlangt ein Netzwerk eine Anpassung (Fit) von sehr vielen Parametern, die alle zu linearen Abbildungen vom Typ $y = Wx + b$ gehören, wie sie als Steigung W einer Geraden und Achsenabschnitt b den meisten aus der Schule in Erinnerung ist. Genauer gesagt finden an jedem Netzwerk-knoten affine Abbildungen (d. h. lineare Abbildung plus Verschiebung) mit Vektoren (\vec{x}, \vec{b}) und Matrizen W statt. Die nichtlinearen Abbildungseigenschaften des Netzwerks entstehen durch Anwendung einer nichtlinearen Abbildung $\sigma(y)$ (Aktivierungsfunktion) auf jedes Element des Ergebnisvektors $\vec{y} = W\vec{y} + \vec{b}$.

Durch die Anpassung der Netzwerk-Parameter werden charakteristische Verzweigungen im Netzwerk festgelegt. Wie die vielen Parameter angepasst werden, ist eine der zentralen Herausforderungen für die Verwendung von

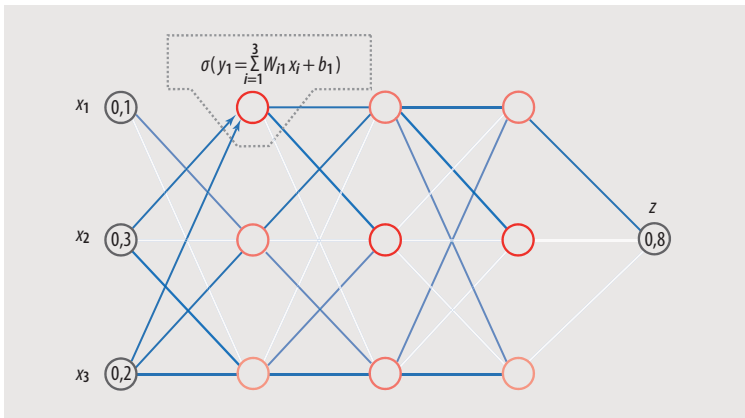


Abb. 1 Prinzip eines neuronalen Netzwerks hier mit 3 Eingabewerten, 3 versteckten Ebenen mit je 3 Netzwerkknoten und 1 Ausgabewert. An jedem Netzwerkknoten – z. B. in der oberen Markierung – werden die Informationen der vorherigen Netzwerklage in 2 Schritten ausgewertet: 1) Lineare Abbildung plus Verschiebung, 2) nichtlineare Abbildung (Aktivierungsfunktion). Die blauen Verbindungen symbolisieren die Gewichtsparameter, die roten Netzwerkknoten die Stärke der Aktivierung aufgrund der Eingabewerte. Die Eingabe- und Ausgabewerte, Farbgebung der Gewichtsparameter und der Aktivierung dienen hier nur der Veranschaulichung.

Netzwerken in der Datenanalyse. Das Anpassungsprinzip ist im folgenden Abschnitt über die Optimierung von Netzwerken beschrieben. Welche Verzweigungen bei einer Berechnung durch das Netzwerk wie stark beitragen, hängt von den Eingabewerten ab. Somit entsteht eine Abbildung von Eingangswerten auf den Ausgabewert.

„Das ist alles?“, mögen Sie fragen: Deep Learning besteht aus der Anpassung von Parametern vieler linearer Abbildungen mit Verschiebungen und mit jeweils nachgeschalteter nichtlinearer Abbildung? Ja. Allerdings ist Gegenstand aktueller Forschung (Mathematik, Informatik, Physik, ...), warum das Konzert aus den vielen Abbildungen an den Netzwerkknoten so herausragend funktioniert.

In der Informatik wurden verschiedene Architekturkonzepte entwickelt, die auf die Struktur des zu lösenden Problems zugeschnitten sind. Einige der Architekturkonzepte wurden bereits in der Physik aufgenommen und zeigen das Potenzial dieser Verfahren für Fortschritte in der Grundlagen- und Anwendungsforschung. Entscheidend bei den Übertragungen der Verfahren ist, die neuen Konzepte an die spezifischen Fragestellungen in der Physik anzupassen, sie gegebenenfalls zu erweitern und die Verlässlichkeit der Netzwerke zu evaluieren. Es gibt bereits mehrere Hundert Publikationen zu wissenschaftlichen Anwendungen von Deep Learning in der Physik.

Optimierung eines Netzwerks

Damit ein neuronales Netzwerk die gewünschte Aufgabe erfüllen kann, müssen Tausende oder sogar Millionen Gewichtsparameter W und Verschiebungen b des Netzwerks im sogenannten Netzwerktraining passend eingestellt werden.

Ein Beispiel dieses Vorgangs ist in der **Abbildung 2** skizziert: Der Verlauf der Punkte (Datenpaare (x, t)) soll durch ein Netzwerk interpoliert werden, das aus sieben Ebenen mit jeweils 200 Knoten besteht. Die Ausgabe des Netzwerks ist mit $f(x)$ bezeichnet. Vor dem Training wurden

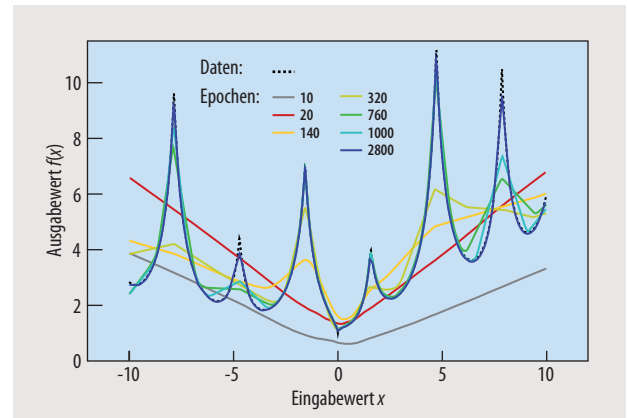


Abb. 2 Optimierung (Training) eines Netzwerks mit dem Ziel, die schwarzen Punktepaare (x, t) durch eine Funktion zu interpolieren. Die Funktion wird hier durch ein Netzwerk mit 1 Eingabewert x und 1 Ausgabewert $f(x)$ realisiert. Bei der Optimierung der Netzwerkparameter wird der Mittelwert der quadrierten Differenzen der Ausgabewerte $f(x)$ und der t -Werte der Punktepaare minimiert. Mit mehr Trainingszyklen verbessern sich die Netzwerkparameter und damit die Interpolation der Punktepaare durch das Netzwerk.

die Verschiebungen auf $b = 0$ und die Gewichtungsfaktoren W anhand einer Gauß-Verteilung zufällig eingestellt, daher hat die anfängliche Ausgabe nichts mit der Interpolation der Punkte zu tun.

Für ein erfolgreiches Training wird eine Zielfunktion benötigt, die Physikerinnen und Physiker anhand ihrer wissenschaftlichen Fragestellung wählen. Bewährt hat sich z. B. der Mittelwert der quadrierten Residuen (Differenz zwischen dem Wert t_i des Datenpaares und dem Ausgabewert $f(x_i)$ des Netzwerks): $Z = \frac{1}{n} \sum_{i=1}^n [t_i - f(x_i)]^2$. Die Optimierung der Gewichtungsfaktoren des Netzwerks geschieht hier durch Minimierung dieser Zielfunktion Z .

Für die Optimierung des Netzwerks werden hier wenige Tausend Iterationen benötigt, bei denen jeweils die partiellen Ableitungen $\partial Z / \partial W$ der Zielfunktion nach den einzelnen Gewichtungsfaktoren W numerisch bestimmt werden (Gradientenverfahren). Damit lassen sich optimierte Werte für die Gewichtungsfaktoren W relativ schnell herausfinden. Genauso werden die Verschiebungsparameter b optimiert (vgl. auch den Artikel von Marco F. Huber).

In **Abbildung 2** kann man das verfolgen. Die Trainingsdaten werden mehrfach genutzt: Mit dem Begriff Epoche wird ein Trainingszyklus bezeichnet, bei dem alle verfügbaren Daten für die Einstellung der Gewichte einmal genutzt werden. Nach den ersten Zyklen ist die Ausgabe $f(x)$ des Netzwerks noch nicht zufriedenstellend, aber bereits nach 2800 Epochen entspricht $f(x)$ recht gut einer Interpolation der Punkte: Die Netzwerk-Parameter sind jetzt gut eingestellt.

Dieses Beispiel mit einer einzigen Eingangsvariablen x und einer Ausgabe $f(x)$ ist sehr einfach gehalten. Bei typischen Fragestellungen haben die Netzwerke n Eingangsvariable und m Ausgaben. Das Prinzip des Netzwerktrainings mit einer Zielfunktion und einem Gradienten-basierten Anpassungsverfahren der Gewichtungsfaktoren wird auch bei anspruchsvollen Netzwerken verfolgt.

Die für ein Netzwerktraining benötigten Trainingsdaten sind bei vielen Fragestellungen der Physik durch Simula-

tionen von physikalischen Vorgängen fast unbegrenzt erhältlich. Zielwerte, die das Netzwerk lernen soll, sind in den Simulationen in der Regel bekannt.

Datenanalyse

In der Physik untersuchte Datenobjekte sind u. a. Aufnahme-sequenzen der Signale von Himmelskörpern mit Teleskopen oder Kollisionen von Teilchen an Beschleunigern mit allen ihren zugehörigen Informationen. Gewöhnlich gehören die Datenobjekte zu verschiedenen Rubriken und zeigen Untergrundeffekte wie Rauschen, sodass eine Nachbearbeitung der Datenobjekte erforderlich ist. Typische Fragestellungen, die derzeit mit Netzwerken bearbeitet werden, sind Zuordnung von Objekten zu Rubriken (Klassifizierung), Eichungen (Regression) und Simulation physikalischer Vorgänge.

In der klassischen Datenanalyse wird eine Klassifizierung von Signal und Untergrund durch Auswahlkriterien in von Physikerinnen und Physikern ausgewählten Variablen erreicht. Ziel ist eine hohe Effizienz in der Trennung von Signal und Untergrund bei möglichst großer Reinheit der Klassifizierung. Auch automatisierte Verfahren, z. B. sogenannte Boosted Decision Trees, kommen zum Einsatz, bei denen nach bestimmten Kriterien aus allen zur Verfügung stehenden Eingangsvariablen eine Zielvariable gebildet wird, die Signal oder Untergrund anzeigt. Solche maschinellen Verfahren wurden bei der Entdeckung des Higgs-Bosons eingesetzt.

Mit Netzwerken sind solche Klassifizierungen ebenfalls möglich. In der ersten Netzwerkebene werden aus den zur Verfügung stehenden n Eingangsvariablen verschiedene Kombinationen der Eingangsvariablen gebildet (siehe **Abb. 1**). Mit diesen Variablenkombinationen werden in der folgenden Netzwerkebene weitere kombinierte Variable gebildet, was mehrfach wiederholt wird. Damit wird eine Hierarchie von Variablengruppen gebildet, die mit tieferer Netzwerkebene zunehmend abstraktere Eigenschaften beschreiben können. Welche Variablen im Netzwerk gebildet werden, hängt von der Netzwerkarchitektur, von den Trainingsdaten und von der Zielfunktion ab. Nach der letzten Netzwerkebene werden m Ausgabewerte ausgegeben, die Wahrscheinlichkeiten für die Zugehörigkeit eines Ereignisses zu m verschiedenen Klassen angeben. Der Hauptunterschied zu bisherigen Methoden besteht darin, dass die Klassifizierung nicht auf Basis bereits ausgewählter Variablen entschieden wird, sondern im Netzwerk **autonom** optimale Variablen gebildet werden.

Netzwerk-Architekturen

Mit einer Netzwerkarchitektur wird definiert, welche Verbindungen innerhalb eines Netzwerks möglich sind. Dieser Abschnitt zeigt eine Auswahl aktueller Konzepte von Netzwerkarchitekturen, die im Rahmen der Informatik entwickelt und auf Fragestellungen der Physik übertragen und weiterentwickelt wurden.

■ **Vollvernetzte Netzwerke:** Als Gedankenexperiment dienen hier Schwarzweißbilder von Spiralgalaxien (**Abb. 3**). Von Interesse ist die Blickrichtung auf die Galaxien, im einfachsten Fall die Rotationsrichtung der Spiralen. Die Bilder

besitzen eine Anordnung von 32×32 Pixel mit jeweils einem Intensitätswert. Die 32 Zeilen jedes Bildes werden hintereinander in einem Vektor mit insgesamt $32 \times 32 = 1024$ Plätzen angeordnet. Der Vektor jedes Bildes wird an die 1024 Eingangsknoten eines Netzwerks gegeben. In der klassischen vollvernetzten Architektur wird jeder Knoten einer Ebene mit jedem Knoten der folgenden Ebene verbunden. Durch Training mit rechts- bzw. linksorientierten Galaxienbildern werden die Gewichte des Netzwerks so eingestellt, dass die Orientierungen von Galaxien erkannt werden.

■ **Convolutional-Netzwerke:** In modernen Netzwerkarchitekturen wird das Bild nicht als Ganzes, sondern abschnittsweise in das Netzwerk eingegeben (**Abb. 4**) [2]. Dabei werden zunächst lokale Zusammenhänge im Bild analysiert. Bei den sogenannten Convolutional-Netzwerken werden mehrere Filter (Kernel) von z. B. 3×3 Pixeln über das Gitter der Pixel geschoben und in jeder Position $x_{k,l}$ ein Wert aus den neun Gewichten W des Kernels bestimmt. Welche Filter gebildet werden und wie die Gewichte angepasst werden, hängt von den Trainingsdaten und der Zielfunktion ab. Der weitergegebene Wert des Filters p kann beispielsweise der Mittelwert der neun gewichteten Intensitätswerte sein: $\Omega_{k,l}^p = \frac{1}{9} \sum_{i,j=1,2,3} W_{ij}^p x_{k+i-2,l+j-2}$. Die Gewichte eines Filters p können z. B. so angepasst sein, dass der Filter für eine vertikale Linienstruktur besonders große Werte liefert $W_{ij}^p = 0$, $W_{2j}^p = 1$, $W_{3j}^p = 0$ mit $j = 1, 2, 3$. Durch verschiedene Filter kann das Bild auf verschiedene Eigenschaften untersucht werden.

Die Ausgaben der Filter werden auf Gittern angeordnet und werden in der folgenden Netzwerkebene durch weitere Filter analysiert. Mit jeder Netzwerkebene vergrößert sich die Reichweite der Analyse lokaler Zusammenhänge im ursprünglichen Bild und es werden komplexere Muster erkannt. Das Konzept dieser Netzwerkarchitektur impliziert die Symmetrie der Translation: Der Filter für vertikale Linienstrukturen wird im Bild vorhandene Linien an jeder Position finden. Convolutional-Netzwerke zeigen herausragende Ergebnisse in der Bilderkennung.

Für hexagonale Anordnungen von Sensoren wie z. B. in Teleskopkameras der Astronomie eignen sich Filter mit 7 Gewichten (**Abb. 4**). Sie werden über die Sensorebene bewegt und an jeder Position rotiert, sodass jeweils 6 Bewertungen entstehen [3]. Damit wird eine Rotationssymmetrie eingeführt, mit der schräg auf die Sensorebene treffende

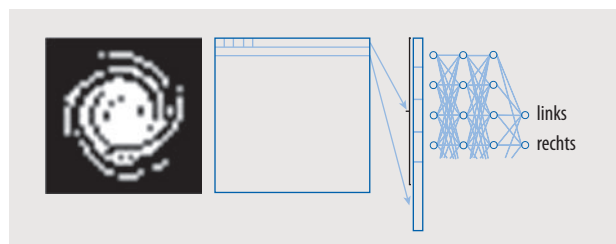


Abb. 3 Bilderkennung mit einem vollvernetzten Netzwerk: Das Bild einer Spiralgalaxie – hier reduziert auf 32×32 Pixel – wird als Ganzes verarbeitet. Die Intensitäten der Pixel werden zeilenweise in die 1024 Eingangswerte des Netzwerks eingefüllt und dann verarbeitet. Mit Trainingsdaten und einer geeigneten Zielfunktion werden die Netzwerkparameter so optimiert, dass die Links- oder Rechtsorientierung einer Spiralgalaxie erkannt wird.

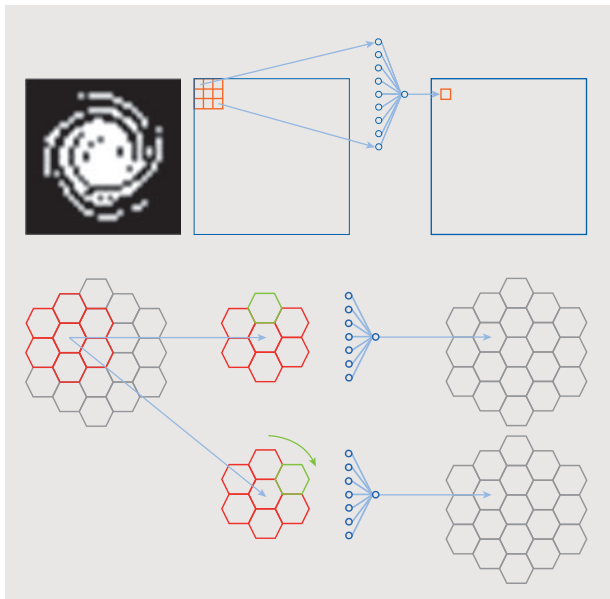


Abb. 4 Bei der Bilderkennung mit einem Convolutional-Netzwerk wird das Bild segmentweise verarbeitet (oben). In der ersten Netzwerklage werden verschiedene Filter über das Bild bewegt, die an jeder Position die Intensitäten der 9 benachbarten Pixel bewerten. Hier gezeigt ist die Ablage der Ergebnisse eines Filters auf einem Gitter entsprechend der Bildpixel. In der zweiten Netzwerklage werden die Ergebnisse aller Filter durch weitere Filter bewertet. Wie in **Abb. 3** beim vollvernetzten Netzwerk werden die gesammelten Informationen in der Ausgabe des Netzwerks zusammengeführt (hier nicht gezeigt). Für Bilder von Teleskopkameras mit hexagonal angeordneten Sensoren werden die Signale nicht mit quadratischen Filtern, sondern mit hexagonal-förmigen Filtern analysiert, die an jeder Position gedreht und für jede Drehung ausgelesen werden (unten).

Signale für alle Richtungen mit demselben Einfallswinkel zur Flächennormalen des Sensors gleichbehandelt werden.

■ **Recurrent-Netzwerke:** Für Zeitspuren von Sensoren wie z. B. Antennen- oder Photomultipliersignale eignen sich sogenannte Recurrent-Netzwerke, die u. a. für Spracherkennung verwendet werden. Bei diesen Netzwerken wird das Signal jedes Zeitintervalls einzeln in das Netzwerk eingegeben und dort mit den zuvor eingegebenen Signalen der vorherigen Zeitintervalle kombiniert (**Abb. 5**).

Weiterentwicklungen dieser Architektur sind Zellen, die einen internen Speicherbereich besitzen [4]. Damit kann die Zelle wichtige Informationen über größere Zeitabstände behalten. Diese Zellen werden Long-Short-Term-Memory (LSTM) genannt. Liegt eine Zeitspur als Ganzes bereits vor, kann die Form der Zeitspur ebenso mit Convolutional-Netzwerken analysiert werden. Je nach Anwendungsfall lohnen sich Vergleiche zwischen den alternativen Konzepten.

■ **Graph-Netzwerke:** Bei Teilchenkollisionen am LHC entstehen viele Teilchen, deren Richtungen und Impulse kontinuierlich verteilt sind. In anfänglichen Übertragungen der Convolutional-Netzwerke auf Fragestellungen der Teilchenphysik wurden die Energie- und Impulsinformationen der Teilchen auf ein kartesisches Raster projiziert und ähnlich wie Bilder analysiert. Durch die Projektion der Teilchen auf feste Intervalle kann jedoch Information verloren gehen.

Alternativ werden neuerdings sogenannte Graph-Netzwerke verwendet, in denen für jedes Teilchen die näch-

sten Nachbarn gesucht und Filter auf diese Nachbarschaft angewendet werden. Die Reihenfolge der Teilchen spielt dabei keine Rolle, sodass hier die Symmetrie der Permutation zum Tragen kommt. In einer Vergleichsstudie zur Erkennung von Top-Quarks mit 15 teilnehmenden Gruppen siegte eine spezielle Form der Graph-Netzwerke, sogenannte Dynamic-Graph-Convolutional-Netzwerke (**Abb. 6**) [5].

■ **Autoencoder-Netzwerke:** Rauschunterdrückung ist nicht nur bei Experimenten der Physik, sondern auch in der Bilderkennung eine bekannte Herausforderung. Hierzu wurden sogenannte Autoencoder-Netzwerke entwickelt. Eine häufig verwendete Architektur enthält eine Engstelle (**Abb. 7**): Im ersten Teil der Netzwerkarchitektur wird die Eingangsinformation (das verrauschte Bild) auf die wichtigste Information komprimiert. Im zweiten Netzwerkteil wird aus dieser komprimierten Information das Bild so originalgetreu wie möglich wiederhergestellt. Aufgrund der gezielt starken Komprimierung kann die Bildinformation nicht vollständig erhalten bleiben: Verloren gehen die zufällig entstandenen Anteile im Bild, das sind die Rauschteile. In der Physik gibt es verschiedene Anwendungen dazu wie die Bereinigung von Antennensignalen oder von Gravitationswellensignalen [6].

Neuerdings werden auch Suchen nach neuen physikalischen Phänomenen in der Teilchenphysik mit Autoencoder-Netzwerken durchgeführt. Bei diesen Suchen werden in der komprimierten Information die Erwartungen des Standardmodells der Teilchenphysik gespeichert. Kann das Netzwerk ein Ereignis nicht annähernd gut reproduzieren, könnte es eine Abweichung vom Standardmodell geben und damit neue Physik jenseits des Standardmodells [7].

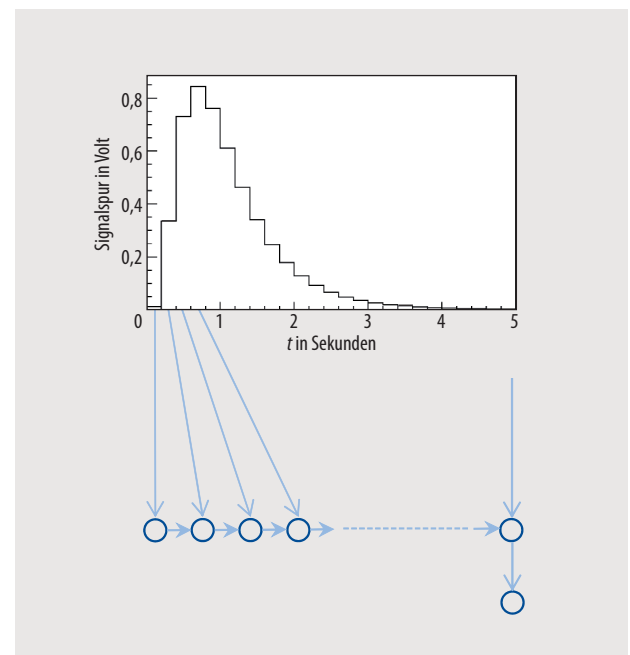


Abb. 5 Prinzip-Beispiel eines Recurrent-Netzwerks. Die Amplituden jedes Zeitintervalls werden einzeln in das Netzwerk eingegeben und dort mit der vorherigen Amplitude zusammen verarbeitet. Im Beispiel wird die Zeitspur durch 1 Ausgabewert charakterisiert.

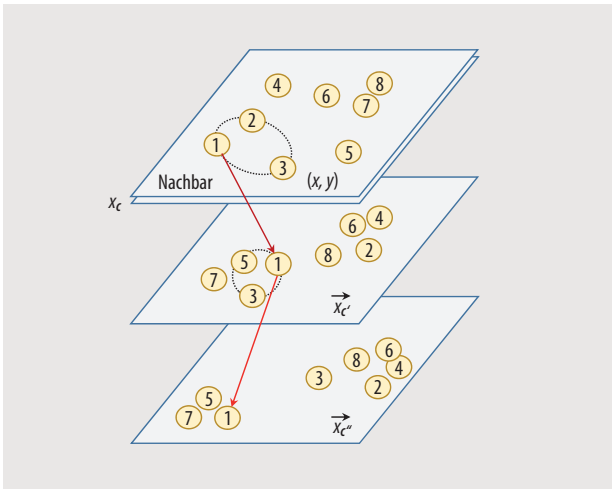


Abb. 6 Prinzip eines *dynamic graph convolutional* Netzwerks. Ohne Erfordernis einer symmetrischen Anordnung werden räumlich benachbarte Objekte – z. B. Teilchen 1, 2, ..., 8 aus einer Kollision – mithilfe von Filtern analysiert (angedeutet durch rote Pfeile; vergleiche Convolutional-Netzwerke **Abb. 4, 5**). In der zweiten Netzwerklage ist Nachbarschaft von Objekten nicht auf räumliche Dimensionen x, y oder Objekteigenschaften x_c beschränkt, sondern kann durch Eigenschaften $x_{c'}$ bestimmt sein, die sich erst durch Nachbarschaften der Objekte untereinander ergeben. Eingeebene Objekte bleiben nach jeder Netzwerklage auffindbar.

■ **Adversarial-Netzwerke:** Zu den neusten Entwicklungen gehören sogenannte Adversarial-Netzwerke, bei denen zwei Netzwerke gegeneinander zur Lösung einer Aufgabe eingesetzt werden [8]. Ein Anwendungsgebiet ist die Erzeugung von Bildern bzw. Detektorsignalen (**Abb. 8**). Das eine Netzwerk wird dazu eingesetzt, ein Bild oder gewünschte Daten zu erzeugen. Es wird als Generatornetzwerk bezeichnet und erhält als Eingangsinformation mehrere Zufallszahlen. Die Zufallszahlen bewirken, dass alle produzierten Bilder des Generators unterschiedlich sind. Das zweite Netzwerk dient ausschließlich dem Trainingsprozess des Generators und wird als Diskriminator bezeichnet. Mithilfe typischer echter Bilder wird das Diskriminatornetzwerk dafür angepasst, Bilder des Generators als Fälschungen zu entlarven. Das Generatornetzwerk wird darauf angepasst, möglichst Bilder zu produzieren, die das Diskriminatornetzwerk von echten Bildern nicht unterscheiden kann.

Abbildung 8 zeigt eine Abfolge von Verbesserungen beim Netzwerktraining zur Simulation von handgeschriebenen Zahlen. Anfangs nach ca. 1000 Trainingszyklen reproduziert der Generator noch das Rauschen der Zufallszahlen am Eingang des Netzwerks. Aber nach genügend vielen Trainingszyklen lernt der Generator, die Außenbereiche der Bilder frei zu lassen und immer bessere Zahlen zu generieren.

Das Prinzip wird auch für ultraschnelle Detektorsimulationen verwendet. Die Erzeugung von Energiedepositionen beim Eindringen eines Teilchens in Materie ist um Größenordnungen schneller als bei einer sukzessiven Simulation von Schauerteilchen: Pro Sekunde werden für mehrere 1000 Teilchenschauer entsprechende Energiedepositionen mit guter Qualität erzeugt (**Abb. 8**) [9].

Systematische Unsicherheiten

In der Physik sind die Abschätzungen statistischer und systematischer Unsicherheiten obligatorisch. Bei Datenanalysen mit neuronalen Netzwerken ist das Vorgehen nicht anders als bei sonst üblichen Analyseverfahren. Neuronale Netzwerke produzieren eine Abbildung von n Eingangsvariablen auf m Ausgaben. Statistische Unsicherheiten, die z. B. in der Anzahl von Ereignissen begründet sind, ergeben sich wie üblich über Poisson-Statistik. Der Einfluss der Unsicherheiten von Eichgrößen (Kalibration) kann durch Variation der Eichung direkt an den verschobenen Ausgabewerten des Netzwerks abgelesen werden.

Weiterhin wird untersucht, welche Unsicherheit durch das Netzwerktraining entsteht. Dadurch, dass alle Gewichte des Netzwerks anfänglich durch Zufallszahlen initialisiert werden, kann jedes Training in einem etwas anderen Optimum münden. Häufig werden daher Ensembles von Netzwerken angepasst und die Ergebnisse aller Netzwerke gemittelt und ihre Varianz bei der Ermittlung der Unsicherheiten berücksichtigt. Ebenso werden als Netzwerkausgaben nicht nur der Wert selbst, sondern zusätzlich eine Vorhersageunsicherheit vom Netzwerk selbst produziert.

Wichtig ist auch, sich einen Einblick darüber zu verschaffen, welche intermediären Variablen das Netzwerk bildet und auf welche Eingangsvariablen das Netzwerk besonders reagiert. Dafür gibt es verschiedene Verfahren wie die sogenannte saliency Darstellung, bei der der Einfluss jeder Eingangsvariablen auf die Netzwerkausgabe durch Variation der Variablen untersucht wird (vgl. auch den Artikel von Marco F. Huber). Eine fortgeschrittene Methode ist die sogenannte Layer-wise-Relevance Propagation (LRP), bei der die Aktivierungen in jeder Netzwerkebene zusammen mit den Netzwerk-Gewichten zusammengefasst werden [10].

Schließlich gibt es Untersuchungen zu dem Problem des Netzwerktrainings mit Simulationsdaten, die nicht exakt den Messdaten entsprechen. Diese Unterschiede können bei Anwendung der Netzwerke auf Messdaten systematische Verschiebungen in der Netzwerkausgabe ergeben. Mithilfe der Adversarial-Netzwerke gibt es Lösungsansätze, in denen Simulationsdaten zuerst an die Eigenschaften der Messdaten angepasst werden und erst danach für das

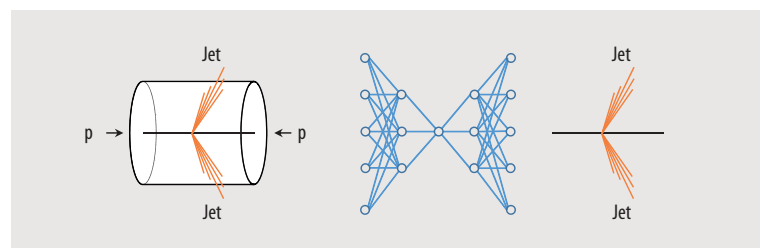


Abb. 7 Netzwerk mit einer Engstelle, wie sie häufig bei *autoencoder* Netzwerken verwendet wird. Aufgrund der Engstelle werden nur wesentliche Informationen der Eingabedaten am Netzwerkausgang reproduziert. Dies wird z. B. zur Reduktion von Rauscheffekten genutzt. Eine neue Anwendung ist die Suche nach bislang unbekanntem physikalischen Phänomenen in der Teilchenphysik: Das Netzwerk wird so trainiert, dass die Reproduktion der Eingabedaten für bereits bekannte Prozesse gelingt. Ereignisse mit neuen physikalischen Phänomenen werden durch Unterschiede zwischen den Eingabe- und Ausgabedaten angezeigt.

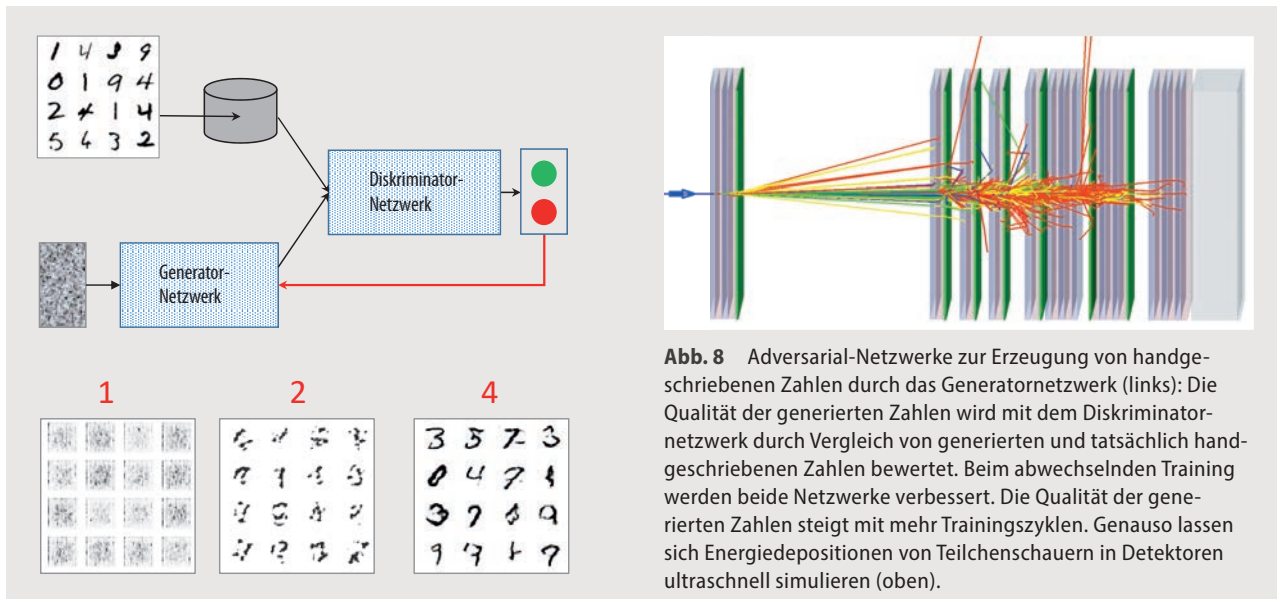


Abb. 8 Adversarial-Netzwerke zur Erzeugung von handgeschriebenen Zahlen durch das Generatorkernetzwerk (links): Die Qualität der generierten Zahlen wird mit dem Diskriminatornetzwerk durch Vergleich von generierten und tatsächlich handgeschriebenen Zahlen bewertet. Beim abwechselnden Training werden beide Netzwerke verbessert. Die Qualität der generierten Zahlen steigt mit mehr Trainingszyklen. Genauso lassen sich Energiedepositionen von Teilchenschauern in Detektoren ultraschnell simulieren (oben).

Netzwerktraining verwendet werden. Systematische Verschiebungen bei Anwendung des Netzwerks auf Messdaten können so reduziert werden [11].

Abschließende Bemerkung

In vielen aktuellen Studien mit tiefen neuronalen Netzwerken wird gezeigt, dass Netzwerke die in Daten vorhandenen Informationen noch umfassender evaluieren, als es bisher möglich war. Aktuell liegt ein besonderes Augenmerk auf autonomen Suchen der Netzwerke nach optimalen Variablen für die jeweils zu lösende Fragestellung (**Abb. 3–6**) und autonomen Suchen nach neuen physikalischen Phänomenen (**Abb. 7**). Weiterhin sind im Fokus Adversarial-Netzwerke (**Abb. 8**), die ultraschnelle Simulation von physikalischen Prozessen, Adaption von Simulationsdaten an Messdaten und Reduktion von systematischen Unsicherheiten einer Messung ermöglichen.

Bei der derzeitigen, berechtigten Aufbruchstimmung ist es wichtig, daran zu erinnern, dass die Netzwerke nur wenig von allein machen. Es braucht Physikerinnen und Physiker, die Netzwerkarchitekturen den Fragestellungen entsprechend entwickeln, die passende Zielfunktionen formulieren, denen die Anpassung von Millionen Netzwerkparametern gelingt, und die die statistischen und systematischen Unsicherheiten der Netzwerkausgaben quantitativ erfassen.

Das Forschungsfeld ist weit offen für alle, die sich für Erkenntnisgewinne durch moderne datengetriebene Methoden interessieren und engagieren möchten. An vielen Universitäten und im World Wide Web werden Kurse zu maschinellem Lernen sowie Deep Learning-Verfahren angeboten. Mindestvoraussetzung für deren professionelle Anwendung sind solide Grundkenntnisse in linearer Algebra, statistische Verfahren zur Datenanalyse und Bestimmung systematischer Unsicherheiten sowie Programmierkenntnisse, z. B. in der Sprache Python. Durch anwenderfreundliche Software-Bibliotheken sinkt die Einstiegsschwelle für die Gestaltung und Optimierung von

Netzwerken. Zum schnell zugänglichen Ausprobieren von Netzwerken ohne aufwändige Softwareinstallationen gibt es online-Plattformen [12, 13].

Literatur

- [1] I. Goodfellow, Y. Bengio und A. Courville, Deep Learning, MIT Press, Cambridge (MA) (2016), www.deeplearningbook.org
- [2] Y. LeCun et al., Proc. IEEE, **86**(11), 2278 (1998)
- [3] E. Hoogeboom et al., arXiv:1803.02108 (2018)
- [4] S. Hochreiter und J. Schmidhuber, Neural Computation. **9**, 1735 (1997)
- [5] G. Kasieczka, T. Plehn et al, SciPost Phys. **7**, 014 (2019)
- [6] H. Shen et al., arXiv:1711.09919 (2017)
- [7] T. Heimel et al., SciPost Phys. **6**, 030 (2019)
- [8] I. Goodfellow et al, arXiv:1406.2661 (2014)
- [9] M. Erdmann, J. Glombitza und T. Quast, Comput. Softw. Big. Sci. **3**, 4 (2019)
- [10] G. Montavon, W. Samek und K.-R. Müller, Digital Signal Processing **73**, 1 (2018)
- [11] M. Erdmann, L. Geiger, J. Glombitza und D. Schmidt, Comput. Softw. Big. Sci. **2**, 4 (2018)
- [12] Tensorflow Playground: playground.tensorflow.org
- [13] VISPA-Plattform für Datenanalysen: vispa.physik.rwth-aachen.de

Der Autor



Martin Erdmann (FV Teilchenphysik) ist Universitätsprofessor für Experimentalphysik an der RWTH Aachen. Seine Arbeitsgruppe forscht mit dem CMS-Experiment am CERN zu Themen der Top-Quark- und Higgs-Physik sowie mit dem Pierre-Auger-Observatorium in Argentinien zu Themen der ultra-hoch-energetischen kosmischen Strahlung. Ein besonderes Anliegen ist ihm die Entwicklung sowie der Einsatz aktueller Analysetechnologien in der Beantwortung sehr anspruchsvoller Fragestellungen. Er engagiert sich für die Vermittlung der Möglichkeiten moderner Digitalisierung durch Vorlesungen, Lehrbücher und Workshops sowie als Vorsitzender des DPG-Arbeitskreises Physik, moderne Informationstechnologie und Künstliche Intelligenz (AKPI).

Prof. Dr. Martin Erdmann, Physikalisches Institut IIIA, RWTH Aachen, Otto-Blumenthal-Str. 16, 52074 Aachen