Glossary

Inhalt

[Machine Learning 1](#_Toc148536833)

[Introduction Deep Learning and Foundation Models 1](#_Toc148536834)

[Introduction to Tensorflow: Logistic Regression Classifier: 8](#_Toc148536835)

[Building Blocks 9](#_Toc148536836)

[Unsupervised Learning 14](#_Toc148536837)

[Image Recognition 17](#_Toc148536838)

[Generating Text Sequences 19](#_Toc148536839)

[Sequence to Sequence Models 22](#_Toc148536840)

[Reinforcement Learning 24](#_Toc148536841)

[Generative Modelle 26](#_Toc148536842)

[Index 29](#_Toc148536843)

# Machine Learning

18.10.2023

**G. Keyword**: These keywords are also mentioned at the bottom of the slides, where they are explained. Simply search for them, e.g. “G. Tensor”

## Introduction Deep Learning and Foundation Models

See Slides 01\_Deep-Learning Introduction

| **Konzept (en)** | **Konzept (de)** | **Description** |
| --- | --- | --- |
| G. Scalar | Skalar | Eine reelle Zahl, z.B. |
| G. Vector | Vektor | Eine Spalte von Zahlen, z.B. |
| G. Matrix | Matrix | Eine rechteckige Anordnung von Zahlen, e.g. |
| G. Tensor | Tensor | Eine 1,2,3- oder höherdimensionale Anordnung von reellen Zahlen, z.B.  Alternative Bezeichnung: Multidimensionaler Array |
| G. Pixel | Pixel | Ein Bild ist eine Matrix von Pixeln, welche die Helligkeit oder Farbe jeder Bildposition durch ein oder mehrere Zahlen kennzeichnen. |
| G. Machine Learning | Maschinelles Lernen | Ziel: Extrahiert Informationen, die für den Benutzer relevant sind, aus Daten. |
| G. Data | Daten | Eine Menge von Instanzen (Beobachtungen, Beispiele).   * Jede Instanz enthält Eingaben für maschinelle Lernverfahren, z.B. Bilder, Sprachsignale, Texte, Tensoren von Zahlen. * Die Instanzen können auch Ausgaben enthalten, die prognostiziert werden sollen. Sie werden dann als annotiert oder gelabelt bezeichnet. * Alle Arten von Daten werden zur Verarbeitung in Zahlen, meist Tensoren, konvertiert. |
| G. Instance | Instanz | Eine Beobachtung (Beispiel) aus der Menge der beobachteten Daten. Die Instanzen sind häufig statistisch unabhängig. |
| G. Generalization | Generalisierung | Sinnvolle Übertragung der Zusammenhänge in den vorliegenden Daten auf neue Daten. |
| G. Model | Modell | Eine Funktion, die einer Eingabe (z.B. Bild, Tensor) eine Ausgabe (z.B. Klassenlabel) zuordnet.   * Jede Eingabe wird zuvor in eine numerische Darstellung umgewandelt. * Ein Modell hat i.A. einen Vektor *w* von ‚freien‘ Parametern, der die Ausgabe des Modells beeinflusst. * Die Parameter werden anhand der Trainingsdaten so angepasst, dass das Modell die Ausgaben der Trainingsdaten möglichst gut rekonstruieren kann. |
| G. Modelling | Modellierung | Erstellen eines Modells für eine relevante Fragestellung. |
| G. Model Application | Modell­anwendung­ | Anwendung eines trainierten Modells auf die Prognose/Analyse von Daten, die nicht beim Training verwendet wurden. |
| G. Class, Label | Klasse, Label | Name für eine Gruppe von Instanzen (meist mit ähnlichen Eigenschaften) |
| G. Classification task | Klassifi­ka­tion­s­auf­gabe­­­­­ | Beschreibe durch ein Modell den Zusammenhang von Eingaben (z.B. Tensor) und den zugehörigen Labeln (Klasse) mit endlich vielen möglichen Werten.   * Aufgabe: Modell (d.h. Parameter *w*) so abändern, dass der Unterschied zwischen vorhergesagtem und tatsächlichem (beobachteten) Label klein wird. |
| G. Training, Model training | Training | Ändere den Parametervektor *w* des Modells so ab, dass für jede Eingabe der Unterschied zwischen vorhergesagter Ausgabe (z.B. Klassenlabel) und beobachteter Ausgabe klein wird. |
| G. Parameter vector | Parameter Vektor | Der Parametervektor *w* des Modells bestimmt dessen Abbildungseigenschaften und wird bei Training abgeändert. |
| G. Linear Transformation, weight matrix, bias | Lineare Transformation | Multipliziert einen Eingabevektor mit einer Matrix und addiert einen Bias-Vektor  also  Grundbaustein der meisten Modelle |
| G. Probability Vector | Wahrscheinlich­keits­vektor­­ | Vektor aus reellen Zahlen:   * jedes Element ist . * Die Summe aller Elemente ist 1. |
| G. Softmax Function | Softmaxfunktion | Eine Funktion, die einen beliebigen Vektor in einen Wahrscheinlichkeitsvektor transformiert. |
| G. Logistic Regression Classifier | Logistischer Regressions Klassifikator | Das logistische Regressionsmodell berechnet die Wahrscheinlichkeit von Klassen für einen Input durch  Wählt man die Klasse mit der höchsten Wahrscheinlichkeit, so erhält man den zugehörigen Klassifikator.   * Die logistische Regression kann also als Klassifikationsmodell genutzt werden. * Bestandteil vieler Tiefer Neuraler Netze. |
| G. Probability of Training Set | Wahrscheinlichkeit der Trainingsmenge | (bedingte) Wahrscheinlichkeit der kompletten Trainingsmenge für den aktuellen Parametervektor .   * Sei die Wahrscheinlichkeit der beob­achteten Ausgabe für die Eingabe und den Parameter . * Annahme: Die beobachteten Instanzen der Trainingsmenge sind (statistisch) **unabhängig**, d.h. zufällig erhoben. * Dann ist die Wahrscheinlichkeit der Trainingsmenge das Produkt der Einzelwahrscheinlichkeiten: |
| G. Training set | Trainings­menge­ | Eine Menge von beobachteten / gelabelten Instanzen , die **nur** für das Training verwendet wird. Typischerweise der größte Teil der Daten (60-80%). |
| G. Loss function | Verlust­funktion­ | Reellwertige Funktion , die für die gegebene Trainings­menge und den gegebenen Parametervektor den Unterschied zwischen prognostizierten und beobachteten Ausgaben misst.   * Kleinerer Wert von geringerer Unterschied |
| G. Loss from probability of training set | Verlustfunktion aus der Wahrscheinlichkeit der Trainingsmenge | Verlustfunktion, welche die Wahrscheinlichkeit der Trainingsmenge maximiert.   * Man bildet den Logarithmus der Wahrscheinlichkeit der Trainingsmenge:   + Wandelt Produkt in Summen um   + Kein Overflow, numerisch stabil * Multiplikation mit -1  Minimierung |
| G. Parameter Optimization | Parameter­optimierung­ | Verfahren zur Änderung des Vektors der Modell­parameter, so dass die Verlustfunktion des Modells für die gegebenen Trainingsmenge möglichst klein wird |
| G. Gradient of | Gradient von | Bei gegebener Trainingsmenge *:* Für eine Funktion und einen Wert die Richtung (Vektor)   in der die Funktion den steilsten Anstieg hat.  https://s14-eu5.startpage.com/cgi-bin/serveimage?url=https:%2F%2Fmoodle.capilanou.ca%2Ffile.php%3Ffile%3D%252F11141%252FImages%252Fgradient_vector.JPG&sp=62b312ee2fa9e1f1148d1f1e0336f7f1 |
| G. Gradient descent optimization | Gradient­abstiegs­optimierung­­ | Bei gegebener Trainingsmenge *:* Eine Methode zur Parameteroptimierung für eine Funktion :   * Zuerst wird der Wert des Parameter­vektors initialisiert (z.B. zufällig). * Dann wird der Gradient für den aktuellen Parameter berechnet. Es ergibt sich ein Vektor , der gleich lang ist wie . Dieser Gradient wird mit der Lernrate (kleine positive Zahl) multipliziert und von dem aktuellen Parameter abgezogen.   Diese Schritt wird wiederholt bis sich keine Änderungen mehr im Wert von ergeben.  ist ein Hyperparameter.   * Normalerweise wird der Gradient für die gesamte Trainingsmenge berechnet. |
| G. Learning Rate | Lernrate | Kleine positive reelle Zahl . Der Gradient wird mit der Lernrate multipliziert und von dem aktuellen Parameter abgezogen.   * Zu hohe Lernrate: Der Optimierer spring hin und her über das Optimum. Es ergibt sich eine starke Oszillation und ggf. ein Ansteigen der Verlustfunktion. * Zu niedrige Lernrate: Die Parameter ändern sich nur ganz langsam, ggf. bleibt der Optimierer in lokalem Minimum. |
| G. Test set | Testmenge | Eine Menge von beobachteten / gelabelten Instanzen, die **nicht** für das Training, sondern nur zur Bestimmung der Performanz eines Modells verwendet wird.   * Hierbei darf die Testmenge nur **einmal** am Ende der Modellerstellung verwendet werden, und die Ergebnisse dürfen nicht zur Verbesserung des Modells genutzt werden. * Hat die Testmenge die gleiche Verteilung wie die zukünftigen Anwendungsdaten, so ist die Performanz auf den Testdaten übertragbar auf die Anwendungsdaten (bis auf Zufalls­schwan­kungen). |
| G. Performance measure | Performanz­maß­ | Maß zur Beurteilung der Güte eines trainierten Modells. Dieses Maß muss auf einer separaten Testmenge berechnet werden, die nicht zum Training verwendet wurde. |
| G. Accuracy | Genauigkeit | Ein Performanzmaß der Klassifikation:  Anzahl der korrekt klassifizierten Instanzen dividiert durch Anzahl aller Instanzen. Wert zwischen 0.0 (schlecht) und 1.0 (gut) |
| G. Precision | Präzision | Ein Performanzmaß der Klassifikation bezüglich eine bestimmten Klasse :  Anzahl der korrekt zugeordneten Instanzen, die der Klasse angehören, dividiert durch Anzahl aller zugeordneten Instanzen, die zu der Klasse gehören.   * Wert zwischen 0.0 (schlecht) und 1.0 (gut) |
| G. Recall | Trefferquote, Recall | Ein Performanzmaß der Klassifikation bezüglich eine bestimmten Klasse :  Anzahl der korrekt zugeordneten Instanzen, die der Klasse angehören,  dividiert durch Anzahl aller Instanzen, die tatsächlich zu der Klasse gehören.   * Wert zwischen 0.0 (schlecht) und 1.0 (gut) |
| G. F-value | F-Maß | Ein Performanzmaß der Klassifikation bezüglich eine bestimmten Klasse :  Harmonisches Mittel von und .   * Wert zwischen 0.0 (schlecht) und 1.0 (gut) |
| G. Machine Learning Tasks | Lernaufgaben des Maschinellen Lernens | Die wichtigsten Lernaufgaben des maschinellen Lernens sind   * überwachtes Lernen * unüberwachtes Lernen * selbstüberwachtes Lernen * Dynamische Steuerung |
| G. Supervised learning | Überwachtes Lernen | Überwachtes Lernen: Daten sind gelabelt.   * Es gibt eine ausgewiesene Zielvariable, deren Werte anhand der Eingabevariablen (z.B. Tensor) vorherzusagen sind. * Gelernt wird der Zusammenhang zwischen Eingabevariable und der Zielvariable. * Im Training liegen Werte für die Eingabevariable und die Zielvariable vor. * Bei der Anwendung des Modells liegen nur die Werte für die Eingabevariable vor. |
| G. Unsupervised learning | Unüberwachtes Lernen | Unüberwachtes Lernen: Daten sind nicht gelabelt.   * Es gibt keine Unterscheidung zwischen Eingabe- und Zielvariable. * Gelernt wird der statistische Zusammenhang zwischen den Variablen. |
| G. Self-supervised learning | Selbst-überwachtes Lernen | Die Daten werden nicht annotiert. Die Aufgabe besteht darin, einen Teil einer Instanz aus anderen Teilen der Instanz zu prognostizieren. Damit kann die interne Struktur der Daten erfasst werden. |
| G. Dynamic control | Dynamische Steuerung | In vielen Situationen gibt es eine Interaktion eines Agenten mit dem System. Zu jedem Zeitschritt spezifiziert der Agent eine Aktion und erhält von der Systemumgebung einen neuen Zustand und eine Belohnung.   * Ziel ist die Maximierung der Belohnungssumme. * Es kann sehr lange dauern, bis der Agent die finale Belohnung erhält. |
| G. Classification | Klassifikation | Eine Art des überwachten Lernens, bei der die Zielvariable nur endlich viele diskrete Werte annehmen kann. |
| G. Regression | Regression | Eine Art des überwachten Lernens, bei der die Zielvariable kontinuierlich ist. Sie kann ein Skalar oder ein Tensor sein. |
| G. Clustering | Clusteranalyse | Die Clusteranalyse ist ein unüberwachtes Lernverfahren. Sie unterteilt eine Menge von Instanzen anhand ihrer Ähnlichkeit in Gruppen. |
| G. Representa­tional power of logistic regression classifier | Repräsenta­tionsfähigkeit des Logistischen Regressions­klassifikators | Eine Logistischer Regressionsklassifikator kann nur Trenn**ebenen** im Eingaberaum erzeugen, um Klassen zu trennen.   * Daher kann er keine gebogenen Trennflächen erzeugen und viele Klassifikationsaufgaben nicht lösen. * Dies gilt für alle linearen Klassifikationsfunktionen. * Es werden nichtlineare Teilfunktionen (Operatoren, Layer) in einer Klassifikationsfunktion benötigt, um gebogene Trennflächen zu erzeugen. |
| G. Nonlinear Function | Nichtlineare Funktion | Eine Funktion , die keine lineare Funktion  ist |
| G. Sigmoid function | Sigmoid­funktion­ | Die nichtlineare Funktion  https://upload.wikimedia.org/wikipedia/commons/thumb/5/53/Sigmoid-function-2.svg/256px-Sigmoid-function-2.svg.png |
| G. Rectified linear unit | Abgeschnittene Linearfunktion | Die nichtlineare Funktion Relu(x) = max(x,0.0) |
| G. Hyperbolic Tangent | Hyperbolischer Tangens | Die nichtlineare Funktion . Sie ist in Python als Bibliotheksfunktion verfügbar.  https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Hyperbolic_Tangent.svg/220px-Hyperbolic_Tangent.svg.png |
| G. Multilayer Neural Network | Mehrebenen Neuronales Netzwerk | Ein neuronales Netzwerk, welches aus mehreren hintereinandergeschalteten Ebenen (Layern) besteht.   * Jede Ebene bildet eine Funktion , welche einen Tensor mit der Funktion in einen anderen Tensor transformiert, z.B. . * Das Netzwerk führt diese Funktionen nacheinander aus. * Es kann zur Klassifikation (mit Softmax im letzten Layer) oder zur Regression (mit linearem letzten Layer) verwendet werden. |
| G. Hidden Vector | Verborgener Vektor, Hidden Vektor | Interner Vektor in einem Mehrebenen Neuronalen Netzwerk. Hierfür liegen keine Beobachtungswerte vor. |
| G. Representation Power of Multilayer Neural Networks | Repräsentationsfähigkeiten des Mehrebenen-Neuronalen Netzwerks | Ein Neuronales Netzwerk mit zwei Ebenen und einer Nichtlinearität kann beliebige Ein-/Ausgaberelationen beliebig genau approximieren.   * Die Dimension des Hidden Vektors muss genügend groß sein. * Gilt für alle gebräuchlichen Arten von Nichtlinearitäten. * Gilt auch für Neuronale Netze mit zusätzlichen Ebenen. |
| G. Hidden Vector as new Features | Hidden Vektor als neuer Merkmalsvektor | Ein Hidden Vektor ist eine neue Repräsentation des Eingabevektors   * Die Optimierung bewirkt, dass aus dem Hidden Vektor die nachfolgenden Schichten des Netzwerkes leichter die gewünschte Ausgabe berechnen können. * Die Elemente der Hidden Vektoren kann man als neue „optimierte“ Merkmale interpretieren. |
| G. Deep Neural Network | Tiefes Neuronales Netz | Ein Mehrebenen Neuronales Netzwerk mit vielen Ebenen.   * Verallgemeinerung:  Netzwerk mit beliebig verbundenen Operatoren (Verbindungsgraph). Die Operatoren berechnen aus einem oder mehreren Eingabe-Tensoren einen Ausgabetensor. |
| G. Advantage of Deep Neural Networks | Vorteil von Tiefen Neuronalen Netzen | Gegenüber Neuronalen Netzen mit wenigen Schichten können tiefe Neuronale Netze mit gleichvielen Parametern komplexere Zusammenhänge (z.B. Klassifikationstrennflächen, Ausgabefunktionen) darstellen.   * Die Hidden Vektoren werden gerade so optimiert, dass sich die Prognose / Klassifikation genauer berechnen lässt. |
| G. Origin of the Term Neural Networks | Herkunft des Begriffs Neuronale Netze | Der Begriff der Neuronalen Netze orientiert sich an der Organisation des Gehirns. Die Verarbeitung der Seheindrücke des Auges im Gehirn erfolgt zum Beispiel schichtweise. |
| G. Filterkernel, Convolutional layer | Filterkernel | Bei einem Convolutional Layer wird ein kleines Netzwerk (Matrix des Filterkernels) über einen großen Tensor (z.B. Matrix) von Eingaben bewegt.   * Erzeugt für jede Position einen Ausgabewert. * Extrahiert ein kleinräumiges Merkmal an verschiedenen Eingabepositionen |
| G. Pooling Layer | Pooling Layer, Pooling-Ebene | Eine Komponente in einem Convolutional Neural Network, welche eine Aggregation (z.B. Mittelwert) benachbarter Elemente in einem Tensor berechnet. Der Ausgabetensor enthält also weniger Elemente. |
| G. Convolutional Neural Network, CNN | Convolutional Neural Network, CNN | Netzwerk, welches meist auf 2- oder 3-dimensionale Eingabetensoren angewendet wird (z.B. Pixelmatrix eines Bildes).   * Convolutional Layer: Es wird ein kleines Netzwerk (Filterkernel) über die Eingabe bewegt   + berechnet für jede Eingabe eine Wert. * Pooling-Layer: lokale Summierung von Werten   Diese Layerpaare können mehrfach wiederholt werden.   * Das letzte Layer ist meist eine vollverbundene Schicht mit einer Softmax-Aktivierungsfunktion. |
| G. Word Embedding, Embedding Vector | Wort-Einbettung, Embedding | Vektor, welcher ein Wort einer Sprache repräsentiert. Die Einbettungen semantisch ähnliche Worte sollen einen geringen Vektor-Abstand (z.B. Euklidische Distanz) haben.   * Es können auch andere Objekte eingebettet werden (z.B. Sätze). |
| G. Embedding Network | Einbettungs­netzwerk­ | Berechnet ein Embedding (Einbettung) für Worte in einer Sequenz. Ein einfaches Beispielnetzwerk:   * Das Embedding eines Wortes ist der Input. * Aus diesem Input werden die Worte in der Nachbarschaft des Wortes möglichst gut prognostiziert. |
| G. Recurrent Neural Network, RNN | Rekurrentes Neuronales Netzwerk | Neuronales Netzwerk, welches sukzessive das jeweils nächste Element (z.B. Wort) einer Sequenz prognostiziert.   * Kann auch Eigenschaften für ein Sequenzelement prognostizieren, z.B. ob es sich um einen Namen handelt. * Verwendet einen langen Hidden Vektor, um die relevanten Informationen aus den vorherigen Elementen der Sequenz zu speichern. |
| G. Reinforcement Learning Task | Aufgabe des Verstärkungs­lernens­ | Eine überwachte Lernaufgabe, bei dem das Lernsignal (Label) erst nach einer Reihe von Zeitpunkten verfügbar ist (Dynamische Steuerung). Beispiel: Schachspiel   * In jedem Zeitpunkt muss der Agent (der Algorithmus) für einen Zustand eine Aktion auswählen. Diese Aktion beeinflusst den weiteren Verlauf. * Hieraus ergibt sich ein neuer Zustand und eine Belohnung (reelle Zahl).   Ziel ist es, abhängig vom Zustand die Aktion so auszuwählen, dass Summe aller zukünftigen Belohnungen maximiert wird.   * Es wird ein neuronales Netz konstruiert, welches für jedem Zustand die optimale Aktion bestimmen soll. |
| G. Machine Learning Steps | Schritte beim Maschinellen Lernen | Die Erstellung eines Modells des Maschinellen Lernens hat folgende Schritte:   * Festlegung einer Modellfunktion. * Definition einer Verlustfunktion für die Daten, z.B. die Differenz zwischen den prognostizierten und beobachteten Labeln. * Minimierung des Verlusts für die Trainingsdaten durch Optimierung. * Anwendung des geschätzten Modells auf neue Daten. |

## Introduction to Tensorflow: Logistic Regression Classifier:

See Slides 03\_IntroTensorflow and Notebook 03.1-Intro\_Tensorflow+keras+gpu\_v2

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Graphical Processing Unit, GPU | Grafikprozessor, GPU | Ein auf die Berechnung von Grafiken spezialisierter und optimierter Prozessor.   * Enthält bis zu mehreren tausend Rechenwerken und hat eine sehr hohe Rechenkapazität.   Die aktuellen Frameworks für Neuronale Netze (z.B. TensorFlow) können den Grafikprozessor für parallele Berechnungen nutzen. |
| G. Parallelization of computations | Parallelisierung der Berechnungen | Als Standard werden alle geeigneten Berechnungen auf der GPU / den Multiprozessor- Cores ausgeführt.   * Jede Operation kann in TensorFlow einem Rechenknoten zugeordnet werden.   Den Datentransport besorgt TensorFlow. |
| G. Deep Learning Toolkit | Programmsysteme für Deep Learning | Systeme zur Formulierung und Ausführung von Deep Learning Modellen: CNTK, Pytorch, Tensorflow, ... |
| G. Tensorflow | Tensorflow | TensorFlow ist ein Framework zur datenstromorientierten Programmierung. Es wird von Python-Programmen heraus benutzt und ist in Python und C++ implementiert. |
| G. Flowgraph | Flußgraph | Eine Darstellung der Berechnungsreihenfolge in einem Tiefen Neuronalen Netz:   * Die mathematischen Operationen sind Knoten des Graphen   die Kanten entsprechen den erzeugten Daten: Tensoren |
| G. Keras | Keras | Library on top of Tensorflow generating Tensorflow commands: simplified specification of neural networks |
| G. Jupyter Notebook | Jupyter Notebook | Ein Jupyter Notebook kann in unterschiedlichen „Zellen“ erklärende Texte und ablauffähige Python-Scripts enthalten.   * Die Texte können Formeln, Grafiken, Videos, etc. enthalten. * Die Zellen mit Programmcode können einzeln ausgeführt werden. |
| G. Model Training Steps | Schritte zum Training eines Modells | 1. Lies die Trainings- und Testdaten. 2. Vorverarbeitung der Trainings und Testdaten. 3. Definiere das Modell. 4. Trainiere das Modell durch Optimierung auf den Trainingsdaten, speichere das trainierte Modell. 5. Validiere das Modell auf den Testdaten. |
| G. Model Application Steps | Schritte zur Anwendung eines Models | 1. Lies die Anwendungsdaten. 2. Vorverarbeitung der Anwendungsdaten. 3. Lies das trainierte Modell ein. 4. Wende das Modell auf den Anwendungsdaten an. |

## Building Blocks

See Slides 04\_DeepLearning\_Building\_Blocks

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Mapping between multidimensional spaces | Abbildung zwischen hochdimensionalen Räumen | Ein tiefes Neuronale Netz definiert eine Abbildung zwischen i.A. hochdimensionalen Räumen:   * Eingaben können Texte, Bilder, Videos, Tonsignale etc. sein, alle repräsentiert als Tensoren * Ausgaben können Klassen, Texte, Bilder, Videos, Tonsignale etc. sein, alle repräsentiert als Tensoren |
| G. Invariant mapping by Deep Neural Network | Invariante Abbildung durch tiefes Neuronales Netz | Die Abbildungsfunktion eines tiefen Neuronalen Netzes muss invariant sein gegenüber einer Reihe von Transformationen. Z.B. für Objekterkennung in Bildern:   * Verschiebung, Skalierung, Verdeckung, Beleuchtungsänderung, Deformation, Hintergrundänderungen * Unterschiedliche Varianten innerhalb einer Klasse |
| G. Deep neural networks extract new Features | Tiefe Neuronale Netze extrahieren neue Merkmale | Ein durch eine Ebene berechneter Hidden Vektor ist eine neue Repräsentation des Eingabevektors   * Die Optimierung bewirkt, dass aus dem Hidden Vektor die nachfolgenden Schichten des Netzwerkes leichter die gewünschte Ausgabe berechnen können.   Die Elemente der Hidden Vektoren kann man als neue „optimierte“ Merkmale interpretieren. |
| G. Deep Learning | Tiefes Lernen | Das Training von Tiefen Neuronalen Netzen:   * Die Abfolge der Ebenen / Operatoren wird durch den Nutzer definiert. * Nonlinear activation functions are necessary to capture non-linear decision boundaries * Die letzte Schicht erzeugt die gewünschte Ausgabe oder Klassifikation. * Während des Trainings werden die Parameter aller Ebenen gleichzeitig so angepasst, dass sie möglichst gut die Instanzen der Trainingsmenge rekonstruieren können. |
| G. Shallow Learning | Shallow Lernen | Das Training von Neuronalen Netzen oder anderen Maschinellen Lernverfahren, welche keine Tiefen Neuronalen Netze sind. |
| G. Invariant representation | Invariante Repräsentation | Repräsentation von Inhalten (e.g. in Bildern), die sich nur wenig ändern, wenn die Inhalte transformiert oder variiert werden |
| G. Activation function | Aktivierungs­funktion­ | Eine nichtlineare Funktion als Operator in einem Netzwerk, die eine Nichtlinearität in das Netzwerk einbringt.   * Notwendig in einem Mehrebenen-Netz * Oft Bestandteil einer vollverbundene Schicht (Fully Connected Layer). |
| G. Fully Connected Layer | Vollverbun­dene Schicht­ | Eine Schicht eines Neuronalen Netzes, welches eine lineare Transformation und anschließend eine nichtlineare Funktion ausführt:  Sie multipliziert einen Eingabevektor mit einer Matrix und addiert einen Bias-Vektor . |
| G. Fully Connected Network (FCN) | Vollverbun­denes Netz­ | Ein vollverbundenes Netz besteht aus einer Reihe von vollverbundenen Schichten. |
| G. Likelihood | Likelihood | Gemeinsame Wahrscheinlichkeit/Dichte aller Instanzen der Trainingsmenge betrachtet als Funktion des Vektors der Parameter   * Meist das Produkt der Wahrscheinlichkeiten aller Instanzen der Trainingsmenge, wenn diese als statistisch unabhängig angenommen werden können |
| G. Maximum Likelihood principle | Maximum Likelihood Prinzip | Der Parametervektor soll so geändert werden, dass die Likelihood maximal wird |
| G. Maximum Likelihood loss function | Maximum Likelihood Verlustfunktion | Verlustfunktion, die genau dann minimal, ist wenn die Likelihood maximal wird |
| G. Network operator | Netzwerk­operator­ | Bestandteil in einem Neuronalen Netz, welches Berechnungen durchführt.   * Historisch als Ebene (Layer) bezeichnet. |
| G. Directed Graph | Gerichteter Graph | Neuronale Netze haben oft nicht aufeinander folgende Ebenen, sondern bilden einen gerichteten Graphen von Operatoren.   * Die Knoten des Graphen (Operatoren) modellieren einzelne Berechnungen. * Die gerichteten Kanten des Graphen entsprechen Daten, welche von einem Operator zu einem anderen fließen. * Die Kanten können parallel verlaufen.   Meist handelt es sich um gerichtete azyklische Graphen.  Diese Modellstrukturen sind in TensorFlow darstellbar. |
| G. Gradient computation | Gradientenberechnung | Die Berechnung des Gradienten der Verlustfunktion erfolgt nach den Regeln der Analysis   * Z.B. Summenregel, Produktregel, Kettenregel * Diese Regeln lassen sich leicht auf multidimensionale Funktionen verallgemeinern * Die Gradienten werden von den Toolboxen automatisch berechnet. |
| G. Chain Rule | Kettenregel | Regel, nach der man die Ableitung geschachtelter Funktionen berechnen kann. Dies wird bei der automatischen Berechnung von Ableitungen verwendet. |
| G. Backpropa­gation ­ | Backpropa­gation | Verfahren zu Berechnung des Gradienten einer Verlustfunktion bei Neuronalen Netzen. Wegen der einzelnen Schichten eines Netzwerks ist die Verlustfunktion eine geschachtete Funktion:  Deren Ableitung wird mit der Kettenregel berechnet.   * Zunächst wird die Prognose des Netzwerkes berechnet und alle Zwischenwerte gespeichert. * Es wird der Wert der Verlustfunktion berechnet. * Dann werden die Ableitungen von der Ausgabe- zur Eingabeschicht unter Verwendung der Zwischenergebnisse „zurückpropagiert“ (Verwendung der Kettenregel). * In aktuellen Frameworks wird die Ableitung automatisch berechnet. |
| G. Automatic Gradient Computation | Automatische Gradientenberechnung | Wenn die Struktur des Tiefen Neuronalen Netzes festgelegt ist, so kann der Gradient des Netzes automatisch berechnet werden   * Für gegebene Trainingsmenge und Parametervektor wird eine Formel zu Berechnung des Gradienten bestimmt. * Der resultierende Gradient hat die gleiche Länge wie . * Dies wird von den aktuellen Toolboxen automatisch durchgeführt. |
| G. Minibatch, Batch | Minibatch | Eine kleine Teilmenge (Batch) der beobachteten Instanzen. Wird meist zufällig aus der Trainingsmenge ausgesucht. |
| G. Stochastic Gradient Descent | Stochastischer Gradienten­abstieg­ | Variante der Gradientenabstiegsoptimierung, bei der der Gradient nur für einen Minibatch von Daten (und nicht für alle Trainingsdaten) berechnet wird.   * Der Minibatch wird für jede Gradientenberechnung zufällig neu ausgewählt.   Vorteile gegenüber der Gradientenberechnung für die gesamte Trainingsmenge:   * Insgesamt geringerer Rechenaufwand, um in die Nähe des Optimums zu kommen. * Gradient schwankt zufällig um den korrekten Wert. Daher kann das Verfahren oft lokale Minima überwinden.   Hyperparameter:   * Die Größe des Minibatchs. * Lernrate |
| G. Epoch | Epoche | Ein Durchgang durch sämtliche Trainingsdaten bei den Iterationen für die Optimierung. |
| G. Batchsize | Batchgröße | Die Batchgröße bestimmt, wie viele Instanzen zur Berechnung des Gradienten für einen Optimierungsschritt genutzt werden. Die Batchgröße ist ein wichtiger Hyperparameter.   * Je kleiner die Batchsize, desto ungenauer die Approximation des Gradienten. * In der Regel erreicht man mit kleineren Batchgrößen das Minimum früher. * Bei kleiner Batchgröße verlässt man eher ein lokales Optimum. * Die optimale Batchgröße hängt auch von der Lernrate ab.   . |
| G. Variants of stochastic Gradient Descent Optimizers | Varianten des stochastischen Gradienten­abstiegs­ | Es gibt eine Reihe von Varianten:   * Momentum: Speichert die vorigen Gradienten und addiert einen Bruchteil hiervon zum aktuellen Gradienten. * Averaging: Nutzt einen Mittelwert der letzten Parameterwerte. * AdaGrad: hat für jede Komponente des Parametervektors eine eigene Lernrate. Diese Lernraten werden automatisch adaptiert. * RMSProp: Automatische Adaption der Lernraten für jeden Parameter. Verwendet einen Durchschnitt der letzten Gradienten. * Adam: Erweiterung von RMSProp, verwendet den Durchschnitt der letzten Gradienten und den Durchschnitt der letzten Quadrate von Gradienten. |
| G. Validation set | Validations­menge­ | Eine Menge von beobachteten / gelabelten Instanzen, die nur für die Messung der Modellgüte während des Trainings und das sukzessive Auswählen von Hyperparametern (z.B. Anzahl der Layer) verwendet wird. |
| G. Parameter initialization | Initialisierung der Parameter | Die Parameter eines tiefen neuronalen Netzes werden im Allgemeinen nach dem Zufallsprinzip initialisiert.   * Notwendig, um Nullableitungen zu vermeiden * Initialize so, dass die Ausgabewerte im Durchschnitt eine Standard-Varianz (1.0) haben. |
| G. Underfitting | Unteran­passung­ | Ein Modell ist ungeeignet für die Modellierungsaufgabe:   * Beim Training hat das Modell einen hohen Fehler auf Trainings und Validations- / Testmenge. * Möglicherweise ist die Lernrate zu hoch. * Oder die Modellstruktur ist zu einfach bzw. hat zu wenige Parameter. Ggf. zusätzliche Layer und/oder größere Hidden Vektoren verwenden. |
| G. Overfitting | Über­anpassung­ | Ein Modell kann zu stark an die vorhandenen Daten angepasst werden und damit nicht-systematische, zufällige Einflüsse in den Daten reproduzieren.   * Beim Training hat das Modelle einen höheren Prognosefehler auf der Validations- / Testmenge und einen kleineren Fehler auf der Trainingsmenge. * Ein solches Modell hat im allgemeinen höhere Prognosefehler auf neuen Daten und kann daher nur schlecht generalisieren. * Als Gegenmaßnahme muss eine Regularisierung verwendet werden. |
| G. Regularization | Regu­larisierung­ | Maßnahme, um eine Überanpassung zu vermeiden   * Reduktion der Modellkomplexität (weniger Layer, weniger Parameter) * Zusätzlicher Term der Verlustfunktion, der die Werte der Parameter in Richtung 0.0 “zieht”.   + L2-Regularisierung reduziert Summe der Quadrate der Parameterwerte.   + L1-Regularisierung reduziert Summe der Absolutwerte der Parameterwerte. * Dropout * Batch Normalization, usw.. |
| G. Dropout | Dropout | Ein Ansatz zur Regularisierung.   * Er geht aus von einem Hidden Vektor innerhalb eines Modells. * In einen Dropout-Layer wird ein bestimmter Prozentsatz (z.B. 50%) der Komponenten von zufällig ausgewählt und auf 0 gesetzt. |
| G. Ensemble Methods | Ensemble­methoden­ | Ensemblemethoden nutzen eine Menge gleichartiger Modelle, um eine Prognoseaufgabe zu lösen.   * Die Modelle weisen zufällige bzw. systematische Unterschiede auf. * Bildet man eine Prognose aus den Mittelwerten der Modellprognosen, so ist der Prognosevarianz meist **geringer** ist als bei Einzelmodellen. Die Methoden lassen sich für beliebige Maschinelle Lernverfahren anwenden, d.h. auch für Neuronale Netze. |
| G. Batch Normalization | Batch Norma­lisierung­ | Die Batch Normalisierung macht folgende Transformation:   * Die Ausgabewerte werden für jeden Batch normiert, z.B. auf Mittelwert 0 und Varianz 1. * Bewirkt eine Regularisierung, reduziert Überanpassung.   Damit kann das Problem der verschwindenden/ explodierenden Gradienten reduziert werden. |
| G. Hyperparameter ­ | Hyperpara­meter | Ein Hyperparameter ist ein Parameter, dessen Wert gesetzt wird bevor der Lernprozess beginnt, z.B. die Anzahl der Layer in einem neuronalen Netz.   * Im Gegensatz dazu werden die Werte der anderen Parameter durch das Training bestimmt. |
| G. Hyperpara­meter Optimization ­ | Hyperpara­meteroptimie­rung­­ | Verfahren zur Auswahl der Werte von Hyperparametern.   * Ein einfacher Ansatz trainiert ein Modell auf den Trainingsdaten mit unterschiedlichen Werten der Hyperparameter und vergleicht die Modellperformanz auf einer zusätzlichen Validationsmenge von annotierten Instanzen. * Durch die Optimierung der Hyperparameter wird die Performanz auf den Validationsdaten systematisch zu hoch.   Zur erwartungstreuen Schätzung der Performanz wird die Testmenge ein einziges Mal nach Auswahl des finalen Modells genutzt. |
| G. Number of parameters vs. training set size | Anzahl der Parameter vs. Anzahl der Datenelemente | Neuronale Netze mit mehr Parametern (N) und einer größeren Trainingsmenge D haben eine höhere Genauigkeit.   * Für optimale Ergebnisse sollten beide Größen in einem bestimmten Verhältnis (nahezu gleich) wachsen. * Bei beschränkter Rechenzeit sollte man eher größere Modelle nehmen. |

## Unsupervised Learning

See Slides 05\_Unsupervised

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Word context and meaning | Wort-Kontext und -Bedeutung | Hypothese, dass die Bedeutung eines Wortes durch die Worte in der Nachbarschaft charakterisiert wird (Firth).   * Erklärung für die Ergebnisse von Word2vec. |
| G. Embedding Neighbor­hood ­ | Embedding Nachbarschaft | Worte, deren Embeddings eine kleine Euklidische Distanz zur dem Embedding eines Zielwortes haben   * Sollten semantisch ähnlich zum Zielwort sein, * Sollten im Text öfter in der Nähe des Zielwortes auftauchen |
| G. Word2Vec | Word2Vec | Verfahren zur Bestimmung von Embeddingvektoren für Worte von Sätzen.   * Prognostiziert mit einem logistischen Regressionsmodell die Wahrscheinlichkeit der Nachbarworte aus dem Embedding für ein zentrales Wort. * Simultanes Training der Embeddings und des Regressionsmodells. |
| G. Negative sampling | Negative Sampling | Approximation der Softmax-Funktion bei sehr vielen Alternativen. Im Nenner wird nur eine zufällige Stichprobe von Alternativen verwendet. |
| G. 2d-Represen­ta­tion of Embeddings ­­ | 2D-Darstellung von Word Embeddings | Darstellung semantisch ähnlicher Worte mit kleinen Euklidischen Distanzen in 2 Dimensionen.   * Die Darstellung wird so berechnet, das Abstände und Richtungen aus den hohen Dimensionen möglichst erhalten bleiben. * Dient zur Visualisierung von Embeddings. |
| G. Embedding Difference | Embedding­differenz­ | Die Differenz zwischen Embeddings ist oft interpretierbar:   * z.B. König – Mann ist der Unterschied zwischen einer einfachen Person und einer Person mit „Königseigenschaft“ * d.h. gleich zu Königin – Frau   Kann als Analogie interpretiert werden.  Gilt nur in einem Teil der Fälle. |
| G. Same word with several meanings | Mehrere Bedeutungen eines Wortes | Je nach Kontext kann ein Wort mehrere Bedeutungen haben. Beispiel „Bank“: Ich gehe zur Bank, um Geld abzuheben.  Ich gehe zur Bank und setze mich. |
| G. Problems of Word2Vec | Probleme von Word2Vec | Die von Word2Vec abgeleiteten Embeddings sind nicht genügend aussagekräftig weil:   * Worte abhängig von den Worten im Kontext unterschiedliche Bedeutungen haben können: Bank (Möbel), Bank (Geld) * Word2Vec ignoriert die Reihenfolge der Worte: Mann tötet Löwen vs. Löwe tötet Mann.   è Die Wort-Einbettung muss den Kontext berücksichtigen. |
| G. Token | Token | Token sind Elemente eines relativ kleinen Vokabulars aus häufigen Wörtern, Wortteilen und Buchstaben.   * Da beliebige Wörter durch Token dargestellt werden können, vermeiden sie das Problem fehlender Wörter bei der Anwendung von NLP-Modellen. * Sie werden so erzeugt, dass häufige Wörter in einem Trainingsdatensatz Token sind  è jeder Datensatz hat andere Token * Sie können mit speziellen Modulen generiert werden, z. B. Byte-Pair Encoding. |
| G. Attention | Attention, Aufmerksamkeit | Berechnung einer Assoziation zwischen einem Referenzwort und allen Worten einer Sequenz.   1. Berechne die Assoziation zwischen und (gewichtet durch Parameter) 2. Normalisiere die Assoziationen von zu allen auf einen Wahrscheinlichkeitsvektor. 3. Transformiere die durch eine Transformation und summiere sie gewichtet mit den Assoziationen  è neues Embedding |
| G. Self-Attention | Self-Attention | Berechnung einer Assoziation zwischen allen Worten einer Sequenz und einem Referenzwort in der Sequenz. |
| G. Self-Attention Computations | Self-Attention Berechnungen | 1. Transformiere Embeddings durch Matrixmultiplikation Query: , Key: , Value: 2. Berechne modifiziertes Skalarprodukte (Assoziationen)    ist eine Skalierung mit der Länge der Embeddings 3. Normalisiere Skalarprodukte zu Wahrscheinlichkeitsvektor: 4. Summiere die Value Vektoren gewichtet mit den normalisierten Skalarprodukten è neue Einbettungen |
| G. Position embedding | Positions-Embedding | Algorithmus benötigt Informationen über die Position eines Tokens im Text.   * Für jede mögliche Position wird ein Positions-Embeddingvektor trainiert, der zu dem Tokenembedding addiert wird. |
| G. Masked language modelling | Masked language Modelling | Trainingkriterium für BERT   * Etwa 15% der Eingabetoken werden mit [MASK] maskiert * Prognostiziere die Wahrscheinlichkeit beobachteten maskierte Tokens aus den berechneten Einbettungsvektoren für [MASK]   BERT ist gezwungen, die Struktur der Sprache zu erlernen |
| G. Multihead Self-Attention | Multihead Self-Attention | In jeder Schicht werden Attention-Berechnungen mit unterschiedlichen Matrizen durchgeführt (heads)   * Die Länge der Einbettungen wird dabei auf reduziert, wenn die ursprüngliche Länge der Einbettungen ist. * Die resultierenden Einbettungen werden anschließend aneinandergehängt. * Jede Attention berechnet Assoziationen zu unterschiedlichen Aspekten * Kein höherer Rechenaufwand |
| G. Attention regularisation | Attention Regularisierung | Das Training der Attention-Modelle in vielen Ebenen erfordert:   * Regularisierung durch Layer Normalization * Residuale Verbindungen, welche einen „Bypass“ um die Attention-Berechnungen bereitstellen |
| G. BERT architecture | Architektur von BERT | BERT besteht aus   * Einer Anzahl von Self-Attention Blöcken bestehend aus   + Eine Multihead Self-Attention Ebene   + Eine vollverbundene Ebene mit nichtlinearer Aktivierung   + Layer-Normalisierung und Residuale Verbindungen (Bypass) * Eine letzte Klassifikationsschicht mit einem logistischen Klassifikator |
| G. Advantages of BERT | Vorteile von BERT | Durch seine Architektur kann BERT viel über Sprache lernen   * Die Einbettungen sind kontext-sensitiv und hängen von den restlichen Token des Textes ab. * BERT kann Informationen von Token vor und hinter dem Zieltoken verwenden, um ein Token zu prognostizieren. * Durch die Self-Attention kann BERT die Informationen von weitentfernten Token nutzen * BERT kann durch die Prognose maskierter Token viele inhaltliche Relationen lernen,  z.B:. die Hauptstadt von Frankreich ist [MASK] |
| G. Pretraining | Vortraining | BERT wird zunächst auf einem sehr großen Textkorpus unüberwacht trainiert.   * Kriterium ist dabei die Prognose der maskierten Token. * Dabei lernt es die Struktur der Sprache und viele inhaltliche Zusammenhänge |
| G. Finetuning | Feintuning | Nach dem Pretraining kann BERT auf eine neue Aufgabe angepasst werden   * Die Finetuningdaten sind mit Klassen annotiert. * Die Wahrscheinlichkeit der Klassen wird aus dem Embedding von dem Starttoken [CLS] vorhergesagt * Die Klassifikations-Genauigkeiten sind höher als bei herkömmlichen Modellen. |
| G. Transfer Learning | Transferlernen | Ein Modell wird auf einer ersten großen Trainingsmenge trainiert.   * Anschließend wird es auf einer zweiten kleineren Trainingsmenge trainiert. * Hat das Modell genügend viele Parameter, so bleibt Wissen aus der erste Trainingsmenge weitgehen erhalten und wird mit dem der zweiten Trainingsmenge kombiniert. |
| G. Information extraction | Informationsextraktion | Extrahieren von Fakten aus einem Text: Entitäten, Namen, Beziehungen, Relationen, Eigenschaften, Objekte, usw. |
| G. Fine tuning tasks for text mining, G. Named entity recognition, G. Sentiment analysis, G. grammatical correctness, G. Entailment | Feintuning Aufgaben für Text Mining | * Named entity recognition markiert alle Eigennamen in einem Text und bestimmt die Art der Eigennamen (person, Ort, etc.) * Sentiment Analyse klassifiziert die Stimmung in einem Text als negativ, neutral oder positiv. * Prüfung auf grammatische Korrektheit: Klassifiziere einen Text als grammatisch korrekt oder nicht. * Prüfung einer Folgerung: Bestimme, ob ein Text B eine Konsequenz des Textes A ist oder nicht. |
| G. GLUE Benchmark | GLUE Benchmark | Acht Test, die das Verständnis der Englische Sprache überprüfen:   * Z.B Folgt eine Aussage logisch aus einer anderen, worauf bezieht sich ein Pronomen, ist ein Satz grammatisch korrekt, etc. * Kann von Muttersprachlern mit hoher Schulbildung bewältigt werden. * BERT verbessert de State-of-the-Art auf 82%, aber immer noch schlechter als die menschliche Genauigkeit von 87%   Aktuelle Modell viel besser: 91% |

## 

## Image Recognition

See Slides 06\_Image\_Recognition

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Image recognition tasks | Aufgaben der Bilderkennung | Erkennen der Objekte in einem Bild   * Erkenne ihren Typ: Klassifikation * Finde ihre Position:  begrenzendes Rechteck, Pixel des Objekts   Bildunterschrift: Beschreibung des Inhaltes eines Bildes |
| G. ImageNet | ImageNet | ImageNet ist eine Bilddatenbank mit Millionen von Fotos zur Bildklassifizierung.   * Hierbei wird jedem Bild ein entsprechendes Label zugewiesen. * Es gibt einen Wettbewerb, bei dem Bilder in 1000 Klassen eingeordnet werden müssen.   Dieser Wettbewerbs hat die Entwicklung neuronaler Netze massiv vorangetrieben. |
| G. Convolution | Faltung | Die Faltung ist der wichtigste Baustein vieler neuronaler Netze.   * Der Filterkernel (kleine Matrix) wird hierzu über die Eingabe (große Matrix, z.B. Pixelmatrix) bewegt (Schrittweite 1) und erzeugt durch Multiplikation für jede Position einen Ausgabewert. Beispielrechnung: Eingabe: Filterkernel: Ergebnis: * Die dazugehörigen Filter-Gewichte sind zu lernende Parameter des neuronalen Netzes. |
| G. MaxPooling | MaxPooling | Fasst benachbarte Felder einer Matrix durch Maximumsbildung zusammen:  Beispielrechnung: Eingabe:  MaxPool mit 2x2 Filter und Schrittweite 2:  Resultat: |
| G. AlexNet | AlexNet | AlexNet war einer der ersten Bildklassifikatoren mit hoher Genauigkeit   * 5 Layer: Faltung und MaxPooling * 3 vollverbundenen Layer * Nutzte GPUs |
| G. Inception module | Inception Modul | Das Inception Modul ist der wichtigste Baustein der InceptionNets (vier Versionen).   * Die Grundidee ist Faltungsoperationen mit verschiedenen Filtergrößen parallel anzuwenden, z.B. 1x1, 3x3 und 5x5. – dadurch werden die Netze „breiter“. * Die Ausgabe aller Layer wird dann konkateniert bevor sie dem nächsten Layer übergeben wird. |
| G. Residual Connection | Bypass­verbindung­ | Die Residual Connection überbrückt eine Ebene eines tiefen neuronalen Netzes. Dabei wird der Output des vorherigen Layers direkt zum Output eines nachfolgenden Layers hinzuaddiert.   * der Effekt verschwindender bzw. explodierenden Gradienten wird reduziert.   Dadurch können Netze beliebiger Tiefe trainiert werden. |
| G. ResNet | ResNet | ResNet dient zu Bildklassifikation und stellt einen Meilenstein in der Entwicklung des tiefen Lernens dar.   * Es verwendet Residual Connections, die den Output der vorherigen Ebene direkt zum Output der nachfolgenden Ebene hinzuaddieren.   Es gibt ResNet-Varianten mit sehr vielen Ebenen (>100). |
| G. Vision Transformer | Vision Transformer | Ein Modell mit BERT-Architektur zur Klassifikation von Bildern. Ein Bild wird in 16x16 Bildbereiche zerlegt.   * Jeder Bildbereich wird linear in ein Embedding transformiert. * Darauf wird ein Standard BERT-Modell angewendet, welches kontextsensitive Embeddings erzeugt. * Pre-Training: entweder maskierte Token vorhersagen oder direkt ImageNet Klassen prognostizieren   Bei sehr vielen Daten:   * Bessere Genauigkeit als CNN-Modelle auf ImageNet |
| G. Segmentation | Segmentie­rung­ | Segmentierung bezeichnet die Unterteilung von Daten oder eines Bildes in Teilbereiche, z.B. alle Pixel, die zu einem Objekt gehören. |
| G. U-Net | U-Net | Die U-Net Architektur ist wichtigste Netzarchitektur zur Bildsegmentierung.   * Der Encoder des U-Nets verringert die Bildgröße immer weiter, bis eine Klassifikation stattfinden kann. * Der Decoder erhöht die Bildgröße durch Upconvolutions immer weiter bis die ursprüngliche Bildgröße nahezu erreicht ist. Hierdurch wird die Klasseninformation auf die Pixel „verteilt“. * Skip-connections, übergeben die Layer-Ausgaben des Encoder an die Schichten des Decoders. |

## Generating Text Sequences

See Slides 07\_Generate\_Text\_Sequences

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Language Model | Sprach­modell­ | Prognostiziert in einem Satz das laufende Word aus allen vorhergehenden Wörtern. Dabei wird die Wahrscheinlichkeiten von Wörtern an dieser Position durch ein Modell abgeschätzt. |
| G. Ngram | N-Gramm | Sequenzen von der festen Länge   * Buchstaben N-Gramm: Abschnitte von aufeinanderfolgenden Buchstaben aus einen Text   Wort N-Gramm: Abschnitte von aufeinanderfolgenden Worten aus einen Text |
| G. n-gram Language Model | N-Gramm Sprach­modell­ | Prognostiziert in einem Satz das laufende Word aus (z.B. ) vorhergehenden Worten.   * Ungenau, erfasst keine weitreichenden Abhängigkeiten. |
| G. Recurrent Neural Network, RNN | Rekurrentes Neuro­nales Netz­, RNN | Modell zur Prognose des nächsten Elementes (Token) einer Sequenz :   * Die Token des Vokabulars werden durch ein Embedding kodiert * Eingabe: Embedding von vorigem Token und voriger Hidden Vektor * Berechnung: oft vollverbundene Schicht mit Aktivierung * Ausgabe: nächster Hidden Vektor und Wahrscheinlichkeit des nächsten Tokens . * Verlustfunktion: gleich wie bei logistischen Regressionsmodellen, wird summiert über die gesamte Sequenz.   Kann Sequenzen variabler Länge mit einem Netzwerk mit fester Ein-/Ausgabedimension verarbeiten. Dabei speichert der Hidden Vektor die Informationen aus den vorigen Sequenzelementen. Training mit SGD |
| G. Exploding Gradient | Explo­die­render Gradient­­ | Bei der wiederholten Anwendung eines Modells (z.B. RNN) geht ein Parameter nach *k* Zeitschritten als in den Gradienten ein.   * : explodierender Gradient,  sehr großer Wert * Optimierer konvergiert nicht |
| G. Vanishing Gradient | Verschwin­dender Gradient­ | Bei der wiederholten Anwendung eines Modells (z.B. RNN) geht ein Parameter nach *k* Zeitschritten als in den Gradienten ein.   * : verschwindender Gradient, sehr kleiner Wert * Optimierer berücksichtigt den über ausgedrückten weitreichenden Zusammenhang nicht |
| G. Gradient Clipping | Gradienten­beschränkung­ | Hat der Gradientenvektor eine Länge größer als eine Schwelle , so wird seine Länge auf reduziere.   * Die Richtung des Gradienten bleibt gleich * löst das Problem des Explodierenden Gradienten |
| G. Long Short Term Memory, G. LSTM | Long Short Term Memory, LSTM | Für jeden Zeitschritt wird die LSTM-Zelle ausgeführt. Sie besteht aus drei Komponenten:  Inputs: Input , Memory Vektor und Hidden Vektor   * Speicher Gate (store gate): addiert modifizierte Inputs zum Memory Vektor, gesteuert durch transformierte Inputs * Vergessen Gate (forget gate): reduziert Memory Vektor Komponenten, gesteuert durch transformierten Inputs * Output Gate (output gate): liest Memory Vektor Komponenten aus gesteuert durch transformierte Inputs . Die ausgelesenen Komponenten werden transformiert und bilden den neuen Hidden Vektor ,   Outputs: Memory Vektor und Hidden Vektor  Hat 5 lineare Transformationen mit trainierbaren Parametern. |
| G. Multilayer RNN | Mehr­ebenen-RNN­ | RNN mit mehreren Ebenen:   * Input der höheren Ebenen sind die ausgegebenen Hidden Vektoren der darunterliegenden Ebene. * Aus dem Hidden Vektor der höchsten Ebene wird das nächste Element (Wort) durch das logistische Modell prognostiziert. |
| G. Disadvantages of RNN | Nachteile von RNN | * Explodierende und verschwindende Gradienten * Es ist sehr schwierig, Korrelationen zwischen weit entfernten Wörtern zu rekonstruieren. * Jedes Wort hat eine einzige Einbettung: kontextabhängige Bedeutungen werden vermischt |
| G. GPT Language Model | GPT Sprachmodell | Prognostiziert das nächste Token in einem Text mit einem BERT-Modell.  Für ,   * Berechne Self-Attention der Embeddings * Aus dem obersten Embedding von prognostiziere Token-Wahrscheinlichkeiten von mit einem logistischen Regressionsmodell. |
| G. GPT Training | GPT Training | Für einen großen Textkorpus prognostiziert GPT sukzessive die bedingte Wahrscheinlichkeiten für die nächsten Token   * Daraus lässt sich die Wahrscheinlichkeit des gesamten Textes als Produkt berechnen. Beispiel: * Die Maximierung der Wahrscheinlichkeit entspricht der Minimierung der Summe der negativen Logarithmen. loss * Ändere die Parameter so mit SGD, dass die Wahrscheinlichkeit des gesamten Textes immer höher wird. * Das Modell kann sehr viel über die Sprache lernen: Syntax und Semantik |
| G. Perplexity | Perplexity | Maßzahl für die Güte des Wortmodells:  Inverse Likelihood pro Wort  Kleiner ist besser |
| G. GPT-2 | GPT-2 | GPT-2 hat 1,5 Milliarden Parameter,   * Eingabelänge 1024 Token * Anzahl der Ebenen: 48 * trainiert auf 40 GB Text * Perplexity auf Penn Tree Bank: 35.8 |
| G. Generating new text | Neuen Text erzeugen | Der Benutzer gibt einen Starttext (Prompt) vor. Der erzeugte Text soll eine passende Fortsetzung des prompts sein.  Wiederhole:   * Berechne die kontext-sensitiven Embeddings für die vorherigen Token. * Vorhersage von Token-Wahrscheinlichkeiten für die nächste Position aus der letzten Einbettung mittels logistischer Regression. * Der nächste Token wird zufällig entsprechend diesen Wahrscheinlichkeiten ausgewählt   Bei der Generierung von Text wird vermieden, Token mit einer geringen Wahrscheinlichkeit zu generieren, weil diese ungenau sind   * Top- sampling: wähle unter den wahrscheinlichsten Token gemäß deren Wahrscheinlichkeit * Top- sampling: Berücksicht token bis zu einer kumulierten Wahrscheinlichkeit von , e. |
| G. GPT-3 | GPT-3 | GPT-3 hat 175 Milliarde Parameter,   * Eingabelänge 2048 Token * Anzahl der Ebenen: 96 * trainiert auf 570 GB Text |
| G. Few-shot learning | Few-shot learning | Große GPT-Modelle können durch Beispiele dazu gebracht werden, eine Aufgabe zu lösen.  Beispiel einer Eingabe:  This is awesome! // Negative  This is bad! // Positive  What a horrible show! //  Antwort: Negative  Die Genauigkeit ist z.T. besser als bei Modellen mit FeintuninG. |
| G. Instruction following | Befolgen von Instruktionen | Große GPT-Modelle können durch Anweisungen dazu gebracht werden, eine Aufgabe zu lösen.  Beispiel einer Eingabe: „translate to German: Peter went to Paris.“  Antwort: Peter fuhr nach Paris. |
| G. Time Series application scenarios | Anwendungsbeispiel für Zeitreihenanalyse | * Vorhersage von Temperatur, Windgeschwindigkeit und Luftfeuchtigkeit in 10-Minuten-Intervallen * Vorhersage von Börsenkursen für ausgewählte Aktien * Klassifizierung der Messwerte einer Windkraftanlage (normal oder nicht normal): Temperatur, Windgeschwindigkeit, Vibration, Luftfeuchtigkeit, erzeugter Strom,... * Klassifizierung von Messwerten für einen Intensivpatienten (normal/alarm): Puls, Blutdruck, Sauerstoffsättigung, ... |
| G. Time series analysis task groups | Aufgaben der Zeitreihenanalyse | Es gibt vier wesentliche Aufgaben der Zeitreihenanalyse:   * Prognose zukünftiger Zeitreihenwerte * Interpolation zwischen beobachteten Zeitreihenwerten * Endeckung anomaler Zeitreihenwerte * Klassifikation von Zeitreihen in unterschiedliche Verlaufsarten |
| G. Methods for time series analysis | Methoden der Zeitreihenanalyse | * Klassische Verfahren wie ARIMA * Rekurrente neuronale Netze * Self-Attention Modelle wie Transformer und GPT * Convolutional neural networks * Direkte Mehrschrittprognose |

## 

## Sequence to Sequence Models

See Slides 08\_seq2seq

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Machine Translation | Maschinelle Übersetzung | Computergestützte Übersetzung von Text aus einer Sprache in eine andere. |
| G. Machine Translation Challenges | Probleme bei der Maschinellen Übersetzung | * Oft keine direkte Entsprechung eines Wortes in einer anderen Sprache. * Worte müssen meist umgeordnet werden. * Viele spezifische Regeln. |
| G. RNN Sequence-to-Sequence Model | RNN Sequenz-nach-Sequenz Modell | Ein Modell zur Übersetzung einer Eingabesequenz in eine entsprechende Ausgabesequenz. Eine wichtige Anwendung ist die Übersetzung eines Satzes aus einer Sprache in eine andere.   * Die Eingabesequenz wird durch ein Encoder-RNN in einen Hidden Vektor kodiert. * Der Hidden Vektor wird durch ein Decoder-RNN in die Ausgabesequenz übertragen. Ein Logistisches Modell berechnet Wahrscheinlichkeit der Ausgabeworte.   Trainingskriterium: Erzeuge die Worte der Ausgabesequenz mit möglichst hoher Wahrscheinlichkeit. |
| G. Sentence embedding | Satz-Embedding | Durch das Encoder RNN erzeugter Hidden Vektor , der den Inhalt des Eingabesatzes kodiert. |
| G. BLEU | BLEU | Performanzmaß zur Beurteilung einer Übersetzung. Es wird das Übersetzungsresultat mit Referenzübersetzungen verglichen. Dabei wird die Anzahl von gleichen Worten, 2-Grammen, …,4-Grammen berücksichtigt. Höherer Wert -> bessere Übersetzung |
| G. Generating a Translation | Erzeugung einer Übersetzung | * Der Eingabesatz wird mit dem Encoder-Modell transformiert und erzeugt einen Hidden Vektor. * Der Decoder berechnet aus dem Hidden Vektor sukzessive die Wahrscheinlichkeit des nächsten Wortes. * Im einfachsten Fall wird das Wort mit der höchsten Wahrscheinlichkeit gewählt. Es kann als Eingabe für die Prognose des nächsten Wortes genutzt werden. |
| G. Beam Search | Beam-Search | Ein heuristischer Suchalgorithmus. Er berücksichtigt die die bisherigen besten Teilergebnisse. Bei der maschinellen Übersetzung wird er bei der Erzeugung des Ausgabesatzes verwendet. |
| G. Transformer | Transformer | Ein Übersetzungsmodell, welches eine Sequenz in eine andere übersetzt. Besteht aus   * Einem BERT encoder, der ein Embedding für jedes Token berechnet. * Einem Decoder (GPT-Sprachmodell), welcher für die nächste Position die Wahrscheinlichkeit der Token berechnet. Er nutzt Encoder-Decoder-Attention, um informationen aus den Encoder-Einbettungen zu nutzen.   Die Ausgabesequenz wird Token für Token erzeugt. |
| G. Encoder-Decoder attention | Encoder-Decoder attention | Eine Attentionberechnung zwischen den Einbettungen der Eingabesequenz und den Einbettungen der aktuellen Ausgabesequenz.   * Wird wie die übliche Self-Attention berechnet, wobei die queries and den berechnet werden und die keys und values aus den |
| G. Multilingual Translation Models | Multilinguale Übersetzungsmodelle | Wenn Transformer genügend viele Parameter haben, können sie mehrere Sprachen gleichzeitig übersetzen   * Nutze einsprachige Texte durch Rückübersetzung: Ein Übersetzer erzeugt die Zielsprache, ein zweiter übersetzt zurück. Die Rückübersetzung sollte gleich dem Ausgangstext sein. * Die Genauigkeit der Übersetzung von Sprachen mit wenig parallelen Daten wird bei multilingualen Modellen besser. * Teilweise erreichen multilinguale Modelle auch bei Sprachpaaren mit vielen Daten eine höhere Genauigkeit als bilinguale Modelle. |
| G. Relation between number of parameters and number of tokens | Relation zwischen Parameteranzahl und Token | Es wurden umfangreiche empirische Versuche unternommen, um die optimale Anzahl der Parameter eines Modells für eine gegebene Anzahl von Token in den Trainingsdaten zu bestimmen.   * Wichtigstes Ergebnis:  Wird die Anzahl der Parameter verdoppelt, so sollte sich die Anzahl der Trainingsdaten verdoppeln |
| G. Finetuning of dialog models | Feintuning von Dialogmodellen | Damit Sprachmodelle für den Dialog mit Nutzern brauchbar sind muss man sie feintunen:   * Anpassung an Beispiele (Frage + Antwort), bei denen das Modell spezifische Fragen beantworten soll |

## 

## Reinforcement Learning

See Slides 09\_Reinforcement

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Reinforcement Learning | bestär­kendes Lernen­ | Der Agent kennt den laufenden Zustand zum Zeitpunkt und soll eine Aktion auswählen.   * beschreibt den Zustand (oft ein Vektor). * Der Agent erhält zu jedem Zeitpunkt eine Belohnung (Reward) , eine reelle Zahl.   Aufgabe ist, eine Strategie zur formulieren, die jedem Zustand eine Aktion zuordnet, so dass die Summe der Belohnungen über die Zeitpunkte maximiert wird. |
| G. Reinforcement Learning Applications | Anwendun­gen des bestär­kenden Lernens­­ | Reinforcement Learning lässt sich anwenden, wenn das endgültige Ergebnis erst nach mehreren Schritten verfügbar ist. Anwendungsbereiche:   * Videospiele, Brettspiele, z.B. Schach * Selbstfahrende Autos * Robotersteuerung * Aktienkauf und -verkauf |
| G. Reward | Beloh­nung­ | Für jeden Zeitpunkt eine reelle Zahl , z.B. ein Geldbetrag. Ziel ist es, die Summe der Belohnungen zu maximieren. |
| G. Policy | Strategie | Eine Funktion , die jedem Zustand eine Aktion zuordnet   * Die Strategie kann auch zufällig sein, d.h. eine Aktion gemäß einer Wahr­schein­lich­keits­ver­tei­lung auswählen. Dies erfordert spezielle Methoden: z.B. Policy Model (Strategiemodell) |
| G. Episode | Episode | Eine Sequenz von beobachteten Tripeln (Zustand, Belohnung, Aktion): |
| G. Deterministic Environment | Deter­ministische Umgebung­ | Der nächste Zustand / Belohnung wird deterministisch vom aktuellen Zustand / Aktion bestimmt. |
| G. Stochastic Environment | Stochas­tische Umgebung­ | Der nächste Zustand / Belohnung ist vom aktuellen Zustand / Aktion und von Zufallseinflüssen abhängig   * Aufgabe ist dann, eine Strategie zur formulieren, die jedem Zustand eine Aktion zuordnet, so dass der **Mittelwert** der Summe der Belohnungen über die Zeitpunkte maximiert wird. * Der Mittelwert wird über die Zufallseinflüsse genommen. |
| G. Discounted Future Reward | Diskon­tierte Zukünftige Beloh­nung­­ | Oft ist der Wert weit in der Zukunft liegender Belohnungen kleiner.   * Daher werden die zukünftigen Belohnungen mit einem Faktor mit reduziert. * Daraus ergibt sich die Diskontierte Zukünftige Belohnung:   Ziel: Finde Strategie mit maximaler Diskontierter Zukünftiger Belohnung. |
| G. Q-Function | Q-Funktion | Ordnet jedem Zustand-Aktionspaar einen Wert zu:   * die maximale Diskontierte Zukünftige Belohnung, die für mit einer optimalen Strategie erreichbar ist. |
| G. Bellman Equation | Bellman Gleichung | Definiert eine Beziehung zwischen und nachfolgenden .  Falls beobachtet wurde gilt für deterministische Umgebungen  Dies erzeugt ein „Gleichungssystem“ zwischen verschiedenen Werten von .  Wichtiger Punkt:   * Die Bellmanngleichung berechnet den Q-Wert für den laufenden Zeitpunkt aus den Q-Werten für den nächsten Zeitpunkt. |
| G. Deep Q-Network | Tiefes Q-Netzwerk | Q-Funktion muss komplexe Zusammenhänge abbilden  approximiere Q-Funktion durch ein tiefes neuronales Netz . |
| G. Deep Q-Network Training | Training des Tiefen Q-Netzwerks | Training des Tiefen Q-Netzwerks bei einer generierten Episode :   * Gemäß der Bellman-Gleichung muss gelten: . * **Verlustfunktion**: Quadratischer Abstand der rechten von der linken Seite * Training mit stochastischem Gradien­ten­abstieg (SGD) * Anschließend wird mit dem Q-Netz und den neuen Parameter eine neue Episode erzeugt, die als neue Trainingsdaten genutzt wird.   Wichtiger Punkt:   * Das Training bringt die Unterschiede zwischen beiden Seiten auf Null. * Die Trainingsdaten werden vom Modell mit den aktuellen Parametern während des Trainings erzeugt. |
| G. Exploration | Explo­ration­ | Um die beste Strategie zu finden ist es notwendig, dass im Training die gesamte Zustands-Aktions-Menge evaluiert wird. Daher sollen neue Episoden nicht immer mit dem optimalen Aktionen gemäß bestimmt werden sondern Aktionen auch manchmal zufällig gewählt werden |
| G. Stochastic Policy | Stochas­tische Strategie­ | Eine nicht-deterministische Strategie, welche in einem Zustand eine Aktion zufällig gemäß einer Wahrscheinlichkeitsverteilung über den Aktionen auswählt.   * Oft besser als eine deterministische Strategie:  z.B. beim Spiel Stein-Papier-Schere * generiert bei gegebenem Zustand zufällig eine Aktion . |
| G. Policy Model | Strategie­modell | Ein Modell, welches eine Wahrscheinlichkeitsverteilung über den Aktionen ausgibt.   * Ist eine stochastische Strategie.   Wird durch ein Neuronales Netz mit Parameter realisiert. |
| G. Episode Distribution of a Stochastic Policy | Episoden­verteilung einer stochas­tischen Strategie | Eine stochastische Strategie: Startzustand .   * generiert bei gegebenem Zustand eine Aktion * Environment generiert zum Zustand-Aktionspaar einen neuen Zustand und einen Reward . * Kann Episoden erzeugen   unterschiedlich durch Zufallseinfluss   Einer stochastischen Strategie entspricht damit eine Verteilung über den Episoden . |
| G. Expected Value of a Policy | Erwartungs­wert einer Strategie | Sei der Parameter der Strategie .   * Strategie erzeugt eine Verteilung über den Episoden. * ist die Summe der diskontierten Belohnungen für die Episode .   Erwartungswert einer Strategie:   * Erzeuge zufällig Episoden mit aktueller Strategie . * Bilde den Mittelwert der diskontierten Belohnungen * Für großes : Erwartungswert der zukünftigen diskontierten Belohnungen   Optimierungskriterium |
| G. Policy Gradient Optimization | Optimierung des Strategie­gradienten | Bei einem Strategiemodell (Policy Model) wird der des Erwartungswertes einer Strategie maximiert.  Iteration:   * Erzeuge eine Episode mit und dem aktuellen Parameter * Für jedes ändere gemäß dem Gradienten   Konvergiert gegen die optimale Strategie. |

## Generative Modelle

See Slides 10\_Generative-Models

| **Konzept (en)** | **Konzept (de)** | **Definition** |
| --- | --- | --- |
| G. Generative Adversarial Networks, G. GAN | Generative Adversarial Networks (GAN) | Generative Adversarial Networks bestehen aus zwei neuronalen Netzwerken, die gegeneinander arbeiten.   * Das Generator-Netzwerk erstellt Objekte, z.B. Bilder. * Das Diskriminator-Netzwerk erhält in zufälliger Reihenfolge echte Bilder oder erzeugte Bilder. Es muss entscheiden, ob ein es sich um echte oder erzeugte Bilder handelt. * Typischerweise bildet das Generator-Netzwerk einen Zufallsvektor auf den gewünschten Ergebnisraum ab. Das Ziel des Generators ist es zu lernen, Bilder zu erzeugen, die vom Diskriminator nicht abgelehnt werden. * Der Diskriminator wird darauf trainiert, die Bilder des Generators von den Bildern aus der Trainingsmenge zu unterscheiden.   Im Ergebnis kann der Generator Bilder erzeugen, die der Diskriminator nicht mehr von den echten Bildern der Trainingsmenge unterscheiden kann. |
| G. Transposed Convolution | Aufwärtsfal­tung­ | Die transposed Convolution (auch Upconvolution) verteilt die Klasseninformation von einer kleinen Matrix auf eine größere Matrix. |
| G. GAN Optimization criterion | GAN Optimie­rungskrite­rium ­­ | Generator und Diskriminator haben unterschiedliche Optimierungskriterien.:   * Der Generator ändert , so dass die Wahrscheinlichkeit steigt, dass der Diskriminator die synthetischen Bilder als echt klassifiziert * Der Diskriminator ändert , so dass mit möglichst hoher Wahrscheinlichkeit die echten Bilder als echt klassifiziert werden und mit möglichst geringer Wahrscheinlichkeit die synthetischen Bilder als echt klassifiziert werden.   Beide Optimierungen werden abwechselnd durchgeführt. |
| G. GAN applications | GAN Anwendungen | GANs können verwendet werden für:   * Erzeugung synthetischer Trainingsdaten, z.B. um andere Neuronale Netze zu trainieren. * Unüberwachte Merkmalsextraktion. * Erzeugung von Testbeispielen, die eine Modell in die Irre führen (adversarial). * Erhöhung der Auflösung von Bildern (Super-Resolution). * Stil-Übertragung: z.B. in den Stil eines speziellen Malers |
| G. Autoencoder | Autoen­coder | Ein Autoencoder ist ein künstliches neuronales Netz, das dazu genutzt wird, effiziente Kodierungen zu lernen.  Es besteht aus mehreren Schichten.   * Der Ein- und Ausgabevektor ist gleich groß. * Die mittlere Schicht hat einen Hidden Vektor, der **kleiner** ist als der Eingabevektor. Ziel ist es, den Wert des Eingabevektors durch den Ausgabevektor zu rekonstruieren.   Dadurch ist das Netzwerk gezwungen, eine effektive Komprimierung der Eingaben in dem Hidden Vektor zu erzeugen. |
| G. GPT model for image generation, DALL E | DALL E | Die erste Version des Bilderzeugungsprograms DALL E verwendete eine GPT-Architektur   * Text-Token wurden zu Embeddings transformiert * 8x8 Bildbereiche wurden zu Embeddings transformiert * Das Modell lernte die Sequenz der Text- und Bildembeddings zu rekonstruieren   Es wurde auf 250 Millionen Text-Bildpaaren trainiert. |
| G. Generating an image with a GPT model | Erzeugung eines Bildes mit einem GPT-Modell | Das GPT-Modell erhält einen Text als Prompt.   * Das GPT-Modell generiert daraus sukzessive Bild-Token * Die Bildtoken werden in Pixel transformiert und zu einem  Bild zusammengesetzt. |
| G. Diffusion model | Diffusionsmodell | Ein Diffusionsmodelle degradiert ein Bild schrittweise durch die Addition von Zufallsrauschen   * Es kann ein Modell trainiert werden, welches diese Degradierung schrittweise umkehrt * Kann die Genauigkeit und Auflösung von generierten Bildern erhöhen |
| G. Speech-to-text translation | Übersetzung gesprochener Sprache in Text | Gesprochene Sprache wird vorverarbeitet   * Für jede 10 msec werden 80 Messwerte zu Token transformiert * Ein Tranformer Encoder-Decoder übersetzt diese in Texttoken * State-of-the-Art Genauigkeit auf störungsfreien Daten und starke Verbesserung auf verrauschten Daten |
| G. Multipurpose model for image analysis | Modell für viele Aufgaben der Bildverarbeitung | Als Eingabe erhält das Modell ein Bild, ein Kommando und einen Beschreibung (geweils tokenisiert)   * Transformer Encoder-Decoder * Gesteuert durch das Kommando kann das Modell eine Vielzahl von Bild-Text Aufgaben erledigen: Objekte erkennen, Bildunterschrift, Fragen zum Bild beantworten, etc. |
| G. Multipurpose model video processing | Modell für viele Aufgaben der Bildverarbeitung | Als Eingabe erhält das Modell Text und ein Video, repräsentiert durch kurze Folgen von Bildbereichen   * Kann Videos nach einer Textbeschreibung erzeugen * Neue Modelle nutzen Diffusionmodelle für eine bessere Auflösung |
| G. Foundation model | Foundation Modell | LLMs können andere Medien verarbeiten, indem diese Medien als Token kodiert werden   * Die gleiche Architektur kann für sehr viele Anwendungen und Medien genutzt werden * In vielen Bereichen Verbesserung des State-of-the-Art * Spezielle Aufgaben können oft durch Instruktionen mit einigen Beispiellösungen (few-shot) gelöst werden. |
| G. Important models and their application | Wichtige Modelle und ihre Anwendungen | * Logistic regression: classification * multilayer perceptron MLP: classification, regression * Word2Vec: semantic similarity * BERT: prediction of masked words, natural language understanding * Convolutional neural network CNN: image classification, image segmentation and analysis * Vision transformer: image classification * Recurrent neural network RNN: text generation, translation, Zeitreihen * GPT language model: text generation, Zeitreihen, few-shot learning * Transformer Encoder-Decoder: translation, solve special tasks * Deep Q-Learning: solve control problems * Generative adversarial network GAN: image generation * DALL E image generation from text * Imagen video: video generation from text |

# Index

2D-Darstellung von Word Embeddings 15

Abgeschnittene Linearfunktion 6

Accuracy 4

Activation function 10

AdaGrad, Stochastic Gradient Descent 12

Adam, Stochastic Gradient Descent 12

Advantage of Deep Neural Networks 7

Advantages of BERT 16

Aktivierungsfunktion 10

AlexNet 18

annotiert 2

Anwendungen des Verstärkenden Lernens 24

Attention 15

Attention regularisation 16

Aufgabe des Verstärkungslernens 8

Aufwärtsfaltung 27

Autoencoder 27

Automatic Gradient Computation 11

Averaging , Stochastic Gradient Descent 12

Backpropagation 11

Batch 12

Batch Normalisierung 14

Batch Normalization 14

Batchgröße 12

Batchsize 12

Beam Search 23

Beam-Search 23

Bellman Equation 25

Bellman Gleichung 25

Belohnung 24

BERT architecture 16

BLEU 23

Bypassverbindung 18

Chain Rule 11

Classification 6

Classification task 2

Clusteranalyse 6

Clustering 6

CNN 7

CNTK 9

Convolution 18

Convolutional layer 7

Convolutional Neural Network 7

DALL E 27

Data 2

Daten 2

Decoder-RNN 23

Deep Learning 10

Deep Learning Toolkits 9

Deep Neural Network 7

Deep Neural Networks, Advantage of 7

Deep Q-Network 25

Deep Q-Network Training 25

Dense Layer 10

Deterministic Environment 24

Deterministische Umgebung 24

Diffusion model 28

Directed Graph 11

Discounted Future Reward 25

Diskontierte Zukünftige Belohnung 25

Diskriminator-Netzwerk 27

Dropout 13

Dynamic control 5

Einbettungsnetzwerk 8

Embedding 8

Embedding Difference 15

Embedding Nachbarschaft 14

Embedding Neighborhood 14

Embedding Network 8

Embedding Vector 8

Embeddingdifferenz 15

Embeddings, 2d-Representation 15

Encoder-Decoder attention 23

Encoder-RNN 23

Ensemble Methods 13

Ensemblemethoden 13

Entailment 17

Episode 24

Episode Distribution of a Stochastic Policy 26

Epoch 12

Epoche 12

Erzeugung einer Übersetzung 23

Expected Value of a Policy 26

Explodierender Gradient 20

Exploding Gradient 20

Exploration 25

Faltung 18

Few-shot learning 22

Filterkernel 7

Fine tuning tasks for text mining 17

Finetuning 17

Finetuning of dialog models 24

Flowgraph 9

Flußgraph 9

F-Maß 5

forget gate 20

Foundation model 28

Fully Connected Layer 10

Fully Connected Network 10

F-value 5

GAN 26

GAN Anwendungen 27

GAN applications 27

GAN Optimierungskriterium 27

GAN Optimization criterion 27

gelabelt 2

Generalisierung 2

Generalization 2

Generating a Translation 23

Generating an image with a GPT model 28

Generating new text 21

Generative Adversarial Networks 26

Generator-Netzwerk 26

Gerichteter Graph 11

GLUE Benchmark 17

GPT Language Model 21

GPT model for image generation 27

GPT Training 21

GPT-2 21

GPT-3 21

GPU 8

Gradient Clipping 20

Gradient computation 11

Gradient Descent Optimization 4

Gradient of 3

Gradient von 3

Gradientabstiegsoptimierung 4

Gradientenbeschränkung 20

Grafikprozessor 8

grammatical correctness 17

Graphical Processing Unit 8

Herkunft des Begriffs Neuronale Netze 7

Hidden Vector 7

Hidden Vector as new Features 7, 10

Hidden Vektor 7

Hidden Vektor als neuer Merkmalsvektor 7, 10

Hyperbolic tangens 6

Hyperbolischer Tangens 6

Hyperparameter 14

Hyperparameter Optimization 14

Hyperparameteroptimierung 14

Image recognition tasks 18

ImageNet 18

Important models and their application 28

Inception module 18

Information extraction 17

Instance 2

Instanz 2

Invariant mapping by Deep Neural Network 9

Invariant representation 10

Jupyter Notebook 9

Keras 9

Kettenregel 11

Klassifikation 6

Klassifikationsaufgabe 2

Korrektklassifikationsrate 4

Language Model 19

Layer, Dense 10

Layer, Fully Connected 10

Learning Rate 4

Learning, Deep 10

Learning, Shallow 10

Lernaufgaben des Maschinellen Lernens 5

Lernrate 4

Likelihood 10

Linear Transformation 2, 3

Lineare Transformation 2

Logistic Regression Classifier 3

Logistischer Regressions Klassifikator 3

Long Short Term Memory 20

Loss from Probability of Training Set 3

Loss function 3

LSTM 20

LSTM-Zelle 20

Machine Learning 1

Machine Learning Steps 8

Machine Learning Tasks 5

Machine Translation 22

Machine Translation Challenges 22

Mapping between multidimensional spaces 9

Maschinelle Übersetzung 22

Maschinelles Lernen 1

Masked language modelling 16

Matrix 1

Maximum Likelihood loss function 10

Maximum Likelihood principle 10

MaxPooling 18

Mehrebenen Neuronales Netzwerk 6

Mehrebenen-RNN 20

Methods for time series analysis 22

Minibatch 12

Model 2

Model Application 1, 2

Model Application Steps 9

Model training 2

Model Training Steps 9

Modell 2

Modellanwendung 2

Modellierung task 2

Modelling task 2

Momentum, Stochastic Gradient Descent 12

Multi-Head Attention 16

Multilayer Neural Network 6

Multilayer Neural Networks, Representation Power of 7

Multilayer RNN 20

Multilingual Translation Models 23

Multipurpose model for image analysis 28

Multipurpose model video processing 28

Named entity recognition 17

Negative sampling 14

Network Operator 11

Netzwerkoperator 11

new Features, Hidden Vector as 10

Ngram 19

n-gram Language Model 19

N-Gramm 19

N-Gramm Sprachmodell 19

Nichtlineare Funktion 6

Noise-contrastive estimation 14

Nonlinear Function 6

Number of parameters vs. training set size 14

Optimizer, Stochastic Gradient Descent 12

Origin of the Term Neural Networks 7

output gate 20

Overfitting 13

Parallelisierung der Berechnungen 9

Parallelization of computations 9

Parameter initialization 13

Parameter Optimization 3

Parameter vector 2

Parameteroptimierung 3

Performance measure 4

Performanzmaß 4

Perplexity 21

Policy 24

Policy Gradient Optimization 26

Policy Model 26

Policy, Expected Value of a 26

Pooling Layer 7

Pooling-Ebene 7

Position embedding 16

Präzision 5

Precision 5

Pretraining 17

Probability of Training Set 3

Probability of Training Set, Loss from 3

Probability Vector 2

Probleme bei der Maschinellen Übersetzung 22

Problems of Word2Vec 15

Programmsysteme für Deep Learning 9

Pytorch 9

Q-Function 25

Q-Funktion 25

Q-Network Training, Deep 25

Q-Network, Deep 25

Rausch-Kontrastive Schätzung 14

Recall 5

Rectified linear unit 6

Recurrent Neural Network 8, 19

Regression 6

Regularisierung 13

Regularization 13

Reinforcement Learning 24

Reinforcement Learning Applications 24

Reinforcement Learning task 8

Rekurrentes Neuronales Netz 19

Rekurrentes Neuronales Netzwerk 8

Relation between number of parameters and number of tokens 24

Repräsentationsfähigkeit des Logistischen Regressionsklassifikators 6

Representation Power of Multilayer Neural Networks 7

Representational Power of Logistic Regression Classifier 6

Residual Connection 18

ResNet 18

Reward 24

RMSProp, Stochastic Gradient Descent 12

RNN 8, 19

RNN, Decoder 23

RNN, Encoder 23

Same word with several meanings 15

Satz-Embedding 23

Scalar 1

Schritte beim Maschinellen Lernen 8

Schritte zum Training eines Modells 9

Schritte zur Anwendung eines Models 9

Segmentation 19

Segmentierung 19

Self-Attention 15

Self-Attention Computations 15

Self-supervised learning 5

Sentence embedding 23

Sentiment analysis 17

Sequence-to-Sequence Model 22

Sequenz-nach-Sequenz Modell 22

Shallow Learning 10

Shallow Lernen 10

Sigmoid function 6

Sigmoidfunktion 6

Skalar 1

Softmax Function 3

Softmaxfunktion 3

Speech-to-text translation 28

Sprachmodell 19

Stochastic Environment 24

Stochastic Gradient Descent 12

Stochastic Gradient Descent AdaGrad 12

Stochastic Gradient Descent Adam 12

Stochastic Gradient Descent Averaging 12

Stochastic Gradient Descent Momentum 12

Stochastic Gradient Descent Optimizers, Variants of 12

Stochastic Gradient Descent RMSProp 12

Stochastic Policy 26

Stochastische Strategie 26

Stochastische Umgebung 24

Stochastischer Gradientenabstieg 12

store gate 20

Strategie 24

Supervised learning 5

Tensor 1

Tensorflow 9

Test set 4

Testmenge 4

Tiefes Lernen 10

Tiefes Neuronales Netz 7

Tiefes Q-Netzwerk 25

Time series analysis task groups 22

Time Series application scenarios 22

Token 15

Training 2

Training des Tiefen Q-Netzwerks 25

Training set 3

Training Set, Probability of 3

Trainingsmenge 3

Transfer Learning 17

Transformer 23

Translation, Generating a 23

Transposed Convolution 27

Überanpassung 13

Überwachtes Lernen 5

Underfitting 13

U-Net 19

Unsupervised learning 5

Unteranpassung 13

Unüberwachtes Lernen 5

Validation set 12

Validationsmenge 12

Vanishing Gradient 20

Varianten des stochastischen Gradientenabstiegs 12

Variants of stochastic Gradient Descent Optimizers 12

Vector 1

Vektor 1

Verborgener Vektor 7

Verlustfunktion 3

Verlustfunktion aus der Wahrscheinlichkeit der Trainingsmenge 3

Verschwindender Gradient 20

Verstärkendes Lernen 24

Vision Transformer 19

Vollverbundene Schicht 10

Vollverbundenes Netz 10

Vorteil von Tiefen Neuronalen Netzen 7

Wahrscheinlichkeit der Trainingsmenge 3

Wahrscheinlichkeitsvektor 2

Word context and meaning 14

Word Embedding 8

Word2Vec 14

Wort-Einbettung 8

Wort-Kontext und -Bedeutung 14