

BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Beuth Hochschule für Technik

Seminararbeit

Einführung in die wissenschaftliche Projektarbeit

Thema: Generative Modelle im maschinellen Lernen

Autor: Frank Trojanowski <info@franktrojanowski.com>
Matrikelnummer: 881001

Version vom: 14. Juni 2020

Betreuer: Dr. Hussein Hussein

Zusammenfassung

Hier steht der Text, welcher den Inhalte der Seminararbeit zusammenfasst...

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Abkürzungsverzeichnis	iii
1 Machine Learning	1
1.1 Deep Learning	1
1.2 Representation Learning	1
1.3 Vergleich diskriminativer und generativer Modelle	2
2 Generative Modelle	3
2.1 PixelRNN	3
2.2 Variational Autoencoders (VAE)	3
2.3 Generative Adversarial Networks (GAN)	5
3 Anwendungsmöglichkeiten generativer Modelle	6
4 Ausblick	7
5 Fazit	8
Quellenverzeichnis	9

Abbildungsverzeichnis

1	Autoencoder	4
2	VAE	5

Abkürzungsverzeichnis

AI	artificial intelligence
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
KI	künstliche Intelligenz
KNN	künstliches neurales Netz
ML	machine learning
NN	Neural Network
RNN	Recurrent Neural Network
VAE	Variational Autoencoder

1 Machine Learning

Machine Learning (ML) ist ein interdisziplinäres Teilgebiet der künstlichen Intelligenz, welches die Generierung von »Wissen« aus »Erfahrung« bezweckt. Mithilfe von ML werden IT-Systeme in die Lage versetzt, auf Basis vorhandener Datenbestände und Algorithmen Muster und Gesetzmäßigkeiten zu erkennen und ein komplexes Modell der Daten zu entwickeln. Das Modell, und damit die automatisch erworbene Wissensrepräsentation, kann dann verallgemeinert für neue Problemlösungen oder für die Analyse von bisher unbekannten Daten verwendet werden. Dazu verwendet ML zahlreiche Methoden, wobei in vorliegender Seminararbeit nur jene erläutert werden, welche für das allgemeine Verständnis generativer Modelle nötig sind.

1.1 Deep Learning

Deep Learning umfasst eine Reihe von Algorithmen, deren Architektur auf mehreren nacheinander angeordneten Verarbeitungsschichten basiert und deren Aufgabe es ist, aus unstrukturierten Daten übergeordnete Repräsentationen zu lernen. Deep-Learning-Modelle können auf strukturierte Daten angewendet werden, aber ihre eigentliche Stärke, insbesondere im Hinblick auf die generative Modellierung, ergibt sich aus ihrer Fähigkeit, unstrukturierte Daten verarbeiten zu können. In den meisten Fällen liegt es in der Natur der Aufgabe, unstrukturierte Daten wie neue Bilder oder echte Textzeilen zu erzeugen. Daher verwundert es wenig, dass vor allem die Entwicklungen im Deep Learning einen bedeutenden Einfluss auf die generative Modellierung hatten. Bei der Mehrheit aller Deep-Learning-Modelle handelt es sich um künstliche neuronale Netze (KNNs, kurz »neuronale Netze«) mit mehreren hintereinanderliegenden verborgenen Schichten. Aus diesem Grund ist Deep Learning inzwischen fast zum Synonym für tiefe neuronale Netzwerke geworden. Es ist jedoch wichtig darauf hinzuweisen, dass jedes System, das mehrere Schichten verwendet, um übergeordnete Darstellungen der Eingangsdaten zu lernen, auch eine Form des Deep Learning ist (z. B. Deep-Believe-Netzwerke und Deep-Boltzmann-Maschinen). Um Annahmen nicht bereits im Voraus treffen zu müssen, wird ein Modell benötigt, das die relevanten Strukturen aus den Daten selbst ableiten kann. Hierbei zeichnet sich Deep Learning aus. Die Tatsache, dass Deep Learning seine eigenen Merkmale in einem niederdimensionalen Raum finden kann, bedeutet eine Form des Representation Learnings.

1.2 Representation Learning

Die Grundidee beim Lernen von Repräsentationen besteht darin, nicht den hochdimensionalen Stichprobenraum direkt zu modellieren, sondern jede Beobachtung im Trainingsdatensatz mittels eines niederdimensionalen latenten Raums zu erfassen. Das Modell muss dann eine Mapping-Funktion lernen, die einen Punkt im latenten Raum

nehmen und ihn auf einen Punkt in dem ursprünglichen Raum abbilden kann. Mit anderen Worten: Jeder Punkt im latenten Raum ist die Darstellung einer hochdimensionalen Repräsentation der Eingabedaten, beispielsweise eines Bilds.

1.3 Vergleich diskriminativer und generativer Modelle

Um zu verstehen, was generative Modelle leisten sollen und warum sie wichtig sind, ist es sinnvoll, sie mit ihrem Gegenstück, den diskriminativen Modellen, zu vergleichen. Die meisten Fragestellungen im ML sind diskriminativer Natur.

Ein wesentlicher Unterschied besteht darin, dass bei diskriminativen Modellen jede Beobachtung in den Trainingsdaten mit einem Label versehen ist. Im Fall einer binären Klassifikationsaufgabe wären beispielsweise Gemälde von Picasso mit einer 1 und alle anderen mit einer 0 gekennzeichnet. Das Modell lernt dann, beide Gruppen zu unterscheiden, und gibt die Wahrscheinlichkeit aus, dass eine neue Beobachtung das Label 1 trägt – gleichbedeutend damit, dass sie von Picasso gemalt wurde. Deshalb ist diskriminative Modellierung gleichbedeutend mit überwachtem Lernen beziehungsweise dem Erlernen einer Funktion, die eine Eingabe mithilfe eines mit Labeln versehenen Datensatzes auf eine Ausgabe abbildet. Für generative Modelle bedarf es für gewöhnlich nur ungelabelter Datensätze (als Form des unüberwachten Lernens). Sie kann jedoch auch auf einen mit Labeln versehenen Datensatz angewendet werden, um zu lernen, wie man Beobachtungen und Merkmale aus den einzelnen Kategorien erzeugt.

2 Generative Modelle

Ein generatives Modell beschreibt, wie ein Datensatz im Rahmen eines Wahrscheinlichkeitsmodells erzeugt wird. Die aus diesem Modell gewonnenen Stichproben ermöglichen es, neue Daten zu generieren.

Gegeben sei ein Datensatz mit Bildern von bestimmten Objekten. Es soll ein Modell entwickelt werden, das ein neues Bild von einem dieser Objekte erzeugen kann, das nie existiert hat, aber trotzdem echt aussieht, weil das Modell die allgemeinen Prinzipien gelernt hat, die das Aussehen dieses Objektes bestimmen. Exakt das ist die Art von Aufgabenstellung, die sich mithilfe eines generativen Modells im Machine Learning lösen lässt. Zuerst benötigt man einen Datensatz, der aus zahlreichen Beispielen der zu erzeugenden Entität besteht. Man spricht in diesem Zusammenhang vom Trainingsdatensatz, bei dem jeder dieser Datenpunkte als Beobachtung bezeichnet wird. Jede Beobachtung besteht aus vielen Merkmalen – bei einer Bilderzeugungsaufgabe sind die Merkmale üblicherweise die jeweiligen Werte der einzelnen Pixel. Das Ziel ist es, ein Modell zu erstellen, das neue Merkmale erzeugen kann, die so aussehen, als wären sie nach den gleichen Regeln wie die Originaldaten erstellt worden. Rein konzeptionell ist dies für die Bilderzeugung eine unglaublich schwierige Aufgabe – bedenkt man, wie vielfältig die einzelnen Pixelwerte zugeordnet werden können und wie vergleichsweise gering dagegen die Anzahl der Anordnungen ist, die ein Bild der Entität erzeugen, die reproduziert werden soll.

Ein generatives Modell sollte zudem probabilistisch und nicht deterministisch sein. Wenn das Modell nur eine fest vorgegebene Berechnung umfasst, wie z. B. die Berechnung des Durchschnittswerts jedes Pixels im Datensatz, ist es nicht generativ, da das Modell jedes Mal die gleiche Ausgabe erzeugt. Folglich muss das Modell ein stochastisches (zufälliges) Element beinhalten, das die von ihm erzeugten Ausgaben beeinflusst. Es gibt also eine unbekannte Wahrscheinlichkeitsverteilung, die erklärt, warum einige Bilder wahrscheinlich im Trainingsdatensatz zu finden sind und andere Bilder hingegen nicht. Es muss ein Modell entwickelt werden, das diese Verteilung so genau wie möglich nachahmt, um daraus neue, einzigartige Beobachtungen zu generieren, die so aussehen, als würden sie dem ursprünglichen Trainingsdatensatz entstammen.

2.1 PixelRNN

*** Unterkapitel wird möglicherweise verworfen ***

2.2 Variational Autoencoders (VAE)

Im Vergleich zu den diskriminativen Aufgaben von KNN als Regressoren und Klassifikatoren haben VAEs starke generative Modelle, die mittlerweile diverse Anwendungsmöglichkeiten haben, von der Erzeugung menschlicher Gesichter bis zur Produk-

tion reinsynthetischer Musik. Dabei möchte man aber meist bereits vorhandene Daten nicht auf zufällige Weise verändern, sondern die erzeugten Ausgaben in eine spezifische Richtung steuern. VAEs sind in dieser Hinsicht allen anderen momentan bekannten Methoden überlegen. Um die Funktionsweise von VAEs zu verstehen, ist es hilfreich, sich erst einmal einen Standard-Autoencoder anzuschauen:

Autoencoder

Ein Autoencoder-Netzwerk ist eigentlich ein in Encoder und Decoder geteiltes Netzwerk. Der Encoder nimmt Eingabedaten und konvertiert diese in eine kleinere, dichtere Repräsentation. Das Encoding beschreibt die komprimierten Daten im latenten Raum. Der Decoder kann die ursprünglichen Daten fast originalgetreu aus dieser Repräsentation wiederherstellen. Dies ist als eine Art Datenkompression zu sehen, da der Encoder

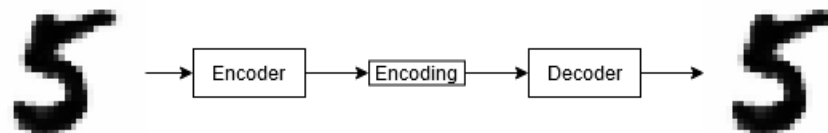


Abbildung 1: Funktionsweise eines Autoencoders

spezifische Encodings generiert, die für die spätere Generierung des dem originalen Input sehr ähnlichen Ausgabedaten nötig ist. Als Verlustfunktion wird normalerweise entweder »mean-squared error« oder »cross-entropy« benutzt, um das Autoencoder-Netzwerk für Ausgaben zu bestrafen, die nicht den Eingaben entsprechen. Da das Encoding bzw. der latente sehr viel weniger Speicherplatz bedarf, als die Originaleingabe, muss das Netzwerk Informationen verwerfen. Der Encoder lernt so viel wie nötig, und so wenig wie möglich an relevanten Informationen im Encoding zu behalten, damit das Decodernetzwerk ein möglichst exaktes Abbild der Eingabedaten rekonstruieren kann.

Standard-Autoencoder können also kompakte Datenrepräsentationen lernen und ihre Eingabedaten sehr gut rekonstruieren. Sie können auch sehr gut zur Bildreinigung verrauschter Bilder verwendet werden, da der Encoder lernt, dass die Position des Rauschens innerhalb des latenten Raums nicht relevant ist. Das Problem mit Standard-Autoencodern ist im Hinblick auf generative Modelle, dass sie keine Interpolation im latenten Raum zulassen. Man möchte ja mit generativen Modellen nicht die Originaldaten wiederherstellen, sondern zufällige Stichproben dem latenten Raum entnehmen bzw. Variationen der Eingabedaten aus dem latenten Raum generieren. Wenn diese Stichproben in Raumdiskontinuitäten liegen, werden inkorrekte Ausgabedaten erzeugt, weil der Decoder schlichtweg keine Erfahrung mit Vektoren aus dieser Region des latenten Raumes hat.

VAE

Variational Autoencoder haben hingegen einen kontinuierlichen latenten Raum, welcher

eine Interpolation und zufällige Stichproben erlaubt. Dies wird erreicht, indem der Encoder statt *einem* Vektor der Größe n , *zwei* Vektoren der Größe n , μ und σ , codiert. Dabei ist μ der Mittelwert der Verteilung und σ der Logarithmus der Varianz jeder Dimension.

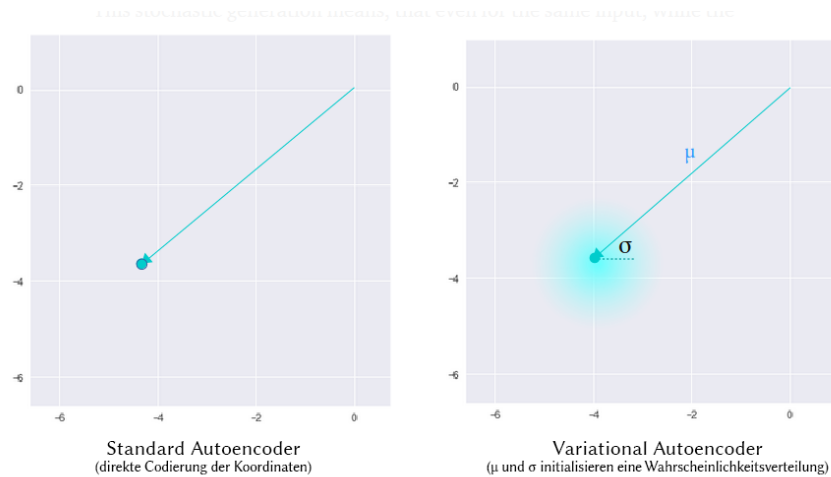


Abbildung 2: Unterschied im Encoding Autoencoder vs. VAE

2.3 Generative Adversarial Networks (GAN)

Dritter Unterabschnitt

3 Anwendungsmöglichkeiten generativer Modelle

Da die meisten von Unternehmen benötigten Lösungen dem Gebiet der diskriminativen Modellierung entstammen.... sie zielen darauf ab, den Einsatz ** in der Industrie durch Automatisierung von Erstellungs-, Validierungs- und Überwachungsprozesse zu kommerzialisieren.

4 Ausblick

Text des Ausblicks... Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5 Fazit

Text des Fazits... Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

[GPAM⁺14] [Doe16] [Goo16] [Roc19] [Ras18] [OKK16] [SKCK17] [Sha18]

Quellenverzeichnis

- [Doe16] DOERSCH, Carl: Tutorial on Variational Autoencoders. In: *arXiv:1606.05908 [cs, stat]* (2016), August. <http://arxiv.org/abs/1606.05908>. – arXiv: 1606.05908
- [Goo16] GOODFELLOW, Ian: NIPS 2016 tutorial: Generative adversarial networks. In: *arXiv preprint arXiv:1701.00160* (2016)
- [GPAM⁺14] GOODFELLOW, Ian ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; XU, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; BENGIO, Yoshua: Generative adversarial nets. In: *Advances in neural information processing systems*, 2014, S. 2672–2680
- [OKK16] OORD, Aäron van d. ; KALCHBRENNER, Nal ; KAVUKCUOGLU, Koray: Pixel Recurrent Neural Networks. In: *CoRR* abs/1601.06759 (2016). <http://arxiv.org/abs/1601.06759>. – [eprint: 1601.06759](http://arxiv.org/abs/1601.06759)
- [Ras18] RASTISLAV, Paluv: Grundlagen des Machine Learning – überwachtes und unüberwachtes Lernen. In: *Plus IT* (2018). <https://plus-it.de/blog/machine-learning-ueberwachtes-vs-unueberwachtes-lernen/>. – Library Catalog: plus-it.de Section: Trends & Innovationen
- [Roc19] ROCCA, Joseph: Understanding Variational Autoencoders (VAEs) - Towards Data Science. (2019). <https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>
- [Sha18] SHAFKAT, Irhum: Intuitively Understanding Variational Autoencoders. In: *Medium* (2018), April. <https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>. – Library Catalog: towardsdatascience.com
- [SKCK17] SALIMANS, Tim ; KARPATY, Andrej ; CHEN, Xi ; KINGMA, Diederik P.: PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In: *arXiv:1701.05517 [cs, stat]* (2017), Januar. <http://arxiv.org/abs/1701.05517>. – arXiv: 1701.05517