

Unsupervised Deep Embedding for Clustering Analysis

Bachelorseminar Data Mining

Lukas Mahr

Ludwig-Maximilians-Universität München

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

Unsupervised Deep Embedding for Clustering
Analysis
Bachelorseminar Data Mining

Lukas Mahr
Ludwig-Maximilians-Universität München

Roadmap

- 1 Clustern von Daten mit hohen Dimensionen
- 2 Einleitung zu Neuronalen Netzen
 - Idee
 - Künstliches Neuron
 - Layer/Schicht
 - Aktivierungsfunktion
 - Loss/Kostenfunktion
 - Backpropagation mit Gradient descent
- 3 Autoencoders
 - Idee
 - Aufbau
- 4 Unsupervised Deep Embedding for Clustering Analysis
 - Stacked Autoencoders
- 5 Deep embedded clustering
 - Kullback Leibler Divergenz
- 6 Referenzen

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Roadmap

Roadmap

- Clustern von Daten mit hohen Dimensionen
- Einleitung zu Neuronalen Netzen
 - Idee
 - Künstliches Neuron
 - Layer/Schicht
 - Aktivierungsfunktion
 - Loss/Kostenfunktion
 - Backpropagation mit Gradient descent
- Autoencoders
 - Idee
 - Aufbau
- Unsupervised Deep Embedding for Clustering Analysis
 - Stacked Autoencoders
- Deep embedded clustering
 - Kullback Leibler Divergenz
- Referenzen

Clustern von Daten mit hohen Dimensionen

■ Problem

- Gaussian Mixture Models, KMeans
 - Abstandsmetriken sind beschränkt auf den ursprünglichen Datendimensionen
 - unwirksam wenn Datendimension hoch sind[1]
- Variationen von KMeans für Daten mit hohen Dimensionen
 - limitiert zu linearen Embeddings[2]
- Spectral clustering
 - Quadratische oder Super-quadratische Komplexität

■ Idee

- Neuronalesnetzwerk zum reduzieren der Dimensionen
 - nicht lineares mapping
- Clustern der reduzierten Daten
 - einfaches Clustering möglich, da Dimensionen reduziert
- Verbessern des NN und der Cluster durch Backpropagation

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Clustern von Daten mit hohen Dimensionen

└ Clustern von Daten mit hohen Dimensionen

viele Daten Punkte viele Distanzen zu berechnen schwierig zu visualisieren
ohne die Dimensionen zu reduzieren Komplexität von Kmeans die exponentiell ansteigt

Clustern von Daten mit hohen Dimensionen

- Problem
 - Gaussian Mixture Models, KMeans
 - Abstandsmetriken sind beschränkt auf den ursprünglichen Datendimensionen
 - unwirksam wenn Datendimension hoch sind[1]
 - Variationen von KMeans für Daten mit hohen Dimensionen
 - limitiert zu linearen Embeddings[2]
 - Spectral clustering
 - Quadratische oder Super-quadratische Komplexität
- Idee
 - Neuronalesnetzwerk zum reduzieren der Dimensionen
 - nicht lineares mapping
 - Clustern der reduzierten Daten
 - einfaches Clustering möglich, da Dimensionen reduziert
 - Verbessern des NN und der Cluster durch Backpropagation

Neuronale Netze

Einleitung zu Neuronalen Netzen

Nicht-lineare statistische Modelle zur Informationsverarbeitung
Informationsverarbeitung umfasst hierbei unter anderem
 Klassifikation
 Prognosenerstellung
Units der Neuronalen Netze angelehnt an Neuronen
 Inputs zusammenfassen
 Mit Schwellenwert vergleichen bzw. aktivieren
Verbindungen zwischen Units angelehnt an Synapsen
 Gewichtung mit verstärkender oder schwächender Wirkung

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis
└─ Einleitung zu Neuronalen Netzen
 └─ Idee
 └─ Einleitung zu Neuronalen Netzen

wofür braucht man neuronale Netze

- Klassifizierung von Daten
- Prognose eines bestimmten Wertes
- Neuronen an Neuronen in unserem Gehirn angelehnt
- Verbindungen zwischen Künstlichen Neuronen an Synapsen angelehnt
- supervised learning = überwachtes lernen
 - man besitzt Daten mit Beschriftungen (labels)
- unsupervised learning nicht überwachtes lernen
 - unbeschriftete Daten, also es sind keine labels vorhanden

Einleitung zu Neuronalen Netzen

Nicht-lineare statistische Modelle zur Informationsverarbeitung umfasst hierbei unter anderem
Klassifikation
Prognosenerstellung
Units der Neuronalen Netze angelehnt an Neuronen
Inputs zusammenfassen
Mit Schwellenwert vergleichen bzw. aktivieren
Verbindungen zwischen Units angelehnt an Synapsen
Gewichtung mit verstärkender oder schwächender Wirkung

Einleitung zu Neuronalen Netzen

Künstlichen Neurons

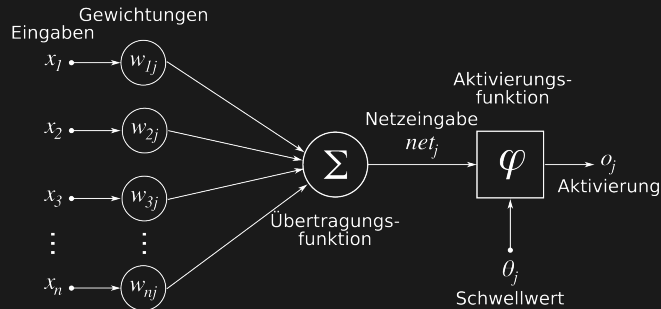
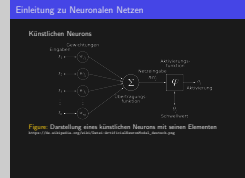


Figure: Darstellung eines künstlichen Neurons mit seinen Elementen
https://de.wikipedia.org/wiki/Datei:ArtificialNeuronModel_deutsch.png

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

- └ Einleitung zu Neuronalen Netzen
 - └ Künstliches Neuron
 - └ Einleitung zu Neuronalen Netzen



x_1, \dots, x_n sind die input variablen, jede der Eingabe variablen besitzt ein Gewicht, w_{1j}, \dots, w_{nj} . Diese werden Multipliziert und davon dann die summe berechnet. Hier die Übertragungsfunktion. Dazu wird ein Bias, in dem Fall der Schwellenwert gerechnet. Als letztes gibt es noch die Aktivierungsfunktion die meistens einen Wert zwischen 0 und 1 zurückgibt. Das ist dann der input für das nächste Neuron.

Einleitung zu Neuronalen Netzen

Layer/Schichten

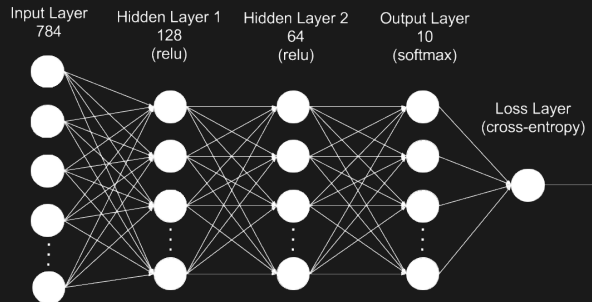
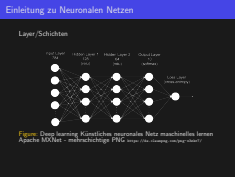


Figure: Deep learning Künstliches neuronales Netz maschinelles lernen
Apache MXNet - mehrschichtige PNG <https://de.cleanpng.com/png-x3zkr7/>

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

- └ Einleitung zu Neuronalen Netzen
 - └ Layer/Schicht
 - └ Einleitung zu Neuronalen Netzen



Layer/Schicht sind mehrere Neuronen die mit allen Neuronen des nächsten Layer/Schicht verbunden sind. Alle Neuronen in einem Layer haben die gleiche Aktivierungsfunktion. Hidden Layer haben meistens die Aktivierungsfunktion rectified linear, da diese recht einfach und schnell zu berechnen ist. Das outputlayer hat meistens eine etwas kompliziertere Funktion wie softmax oder sigmoid. Abhängig von der Aufgabe des Netzwerkes. Letztes Layer hier direkt mit dem Loss

Aktivierungsfunktionen



Figure: Rectifier-Aktivierungsfunktion

https://de.wikipedia.org/wiki/Datei:Activation_rectified_linear.svg

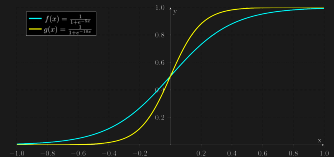


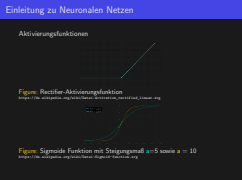
Figure: Sigmoide Funktion mit Steigungsmaß $a=5$ sowie $a = 10$

<https://de.wikipedia.org/wiki/Datei:Sigmoid-function.svg>

2022-02-04

- Unsupervised Deep Embedding for Clustering Analysis
 - Einleitung zu Neuronalen Netzen
 - Aktivierungsfunktion
 - Einleitung zu Neuronalen Netzen

alles negativ ist wird bei relu zu 0 während bei sigmoid, abhängig von der Steigung Werte zwischen -1 und 1 möglich sind



Loss/Kostenfunktion

Mean Squared Error

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

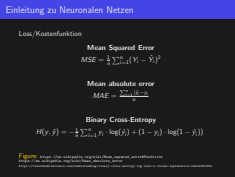
Mean absolute error

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n}$$

Binary Cross-Entropy

$$H(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

Figure: https://en.wikipedia.org/wiki/Mean_squared_error#Predictor
https://en.wikipedia.org/wiki/Mean_absolute_error
<https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>



Man berechnet immer den unterschied zwischen den wahren labeln und den predicteden labeln um zu erkenne wie weit diese auseinander liegen. Es wird immer versucht den Loss zu minimieren. Also ein Minimum der Kostenfunktion zu finden. Die Parameter der Funktion, welche angepasst werden müssen sind alle weights und biases der einzelnen Neuronen und den Layern.

Backpropagation mit Gradient descent

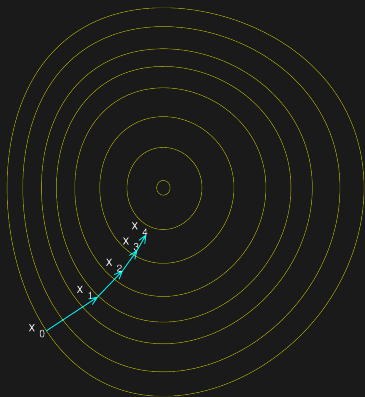


Figure: Illustration of gradient descent on a series of level sets
https://en.wikipedia.org/wiki/File:Gradient_descent.svg

Über Backpropagation wird hier mit z.B Gradient Descent die Loass funktion minimiert. Der Gradient der Loss/ Kostenfunktion wird für alle wights and biases gleichzeitig berechnet. Man kann sich das vorstellen, wie eine Kugel die man einen in einer Hügellandschaft rollen lässt ein kleinen schritten und zwischen den schritten immer nach der Steigung des Abhanges schaut und dabei versucht die Kugel in das tiefste Tal zu bekommen.

Autoencoder

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis
└ Autoencoders
 └ Idee
 └ Autoencoders

Autoencoders

Autoencoder

Ein Autoencoder ist ein feedforward Neural network, was versucht den input zu Kopieren. Das hört sich im ersten Moment nutzlos an, hat aber doch ein paar Anwendungsmöglichkeiten. Dazu gehört z.B Denosing oder Reduktion des Features Spaces

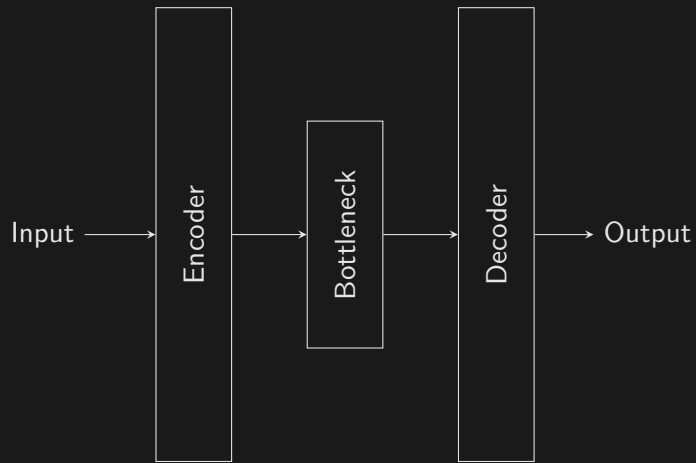
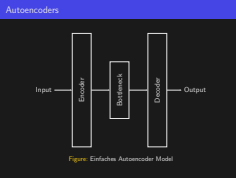


Figure: Einfaches Autoencoder Model

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

- └ Autoencoders
 - └ Aufbau
 - └ Autoencoders



- Aufbau, Das Netzwerk besteht im Grunde aus 2 Teilen.
- Encoder und Decoder
- Encoder Transformiert die Input Daten in eine kleinere gewünschte Dimension
- Decoder Transformiert die Daten aus der Kleinen Dimension zurück in die Original Dimensionen.
- Hoffnung das der Encoder die Daten auf die Wichtigsten Features reduziert.

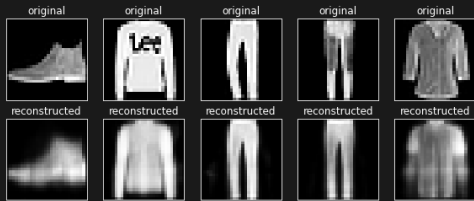


Figure: Original und Decoded Bilder

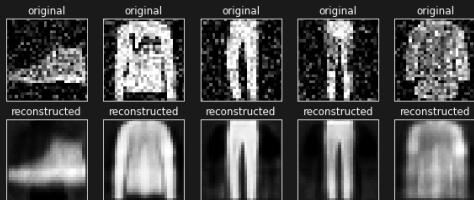


Figure: Noisy und Decoded Bilder

<https://github.com/Plutokezk/dec/blob/main/Autoencoders.ipynb>



Figure: Original und Decoded Bilder



Figure: Noisy und Decoded Bilder

Images: 11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66/67/68/69/70/71/72/73/74/75/76/77/78/79/80/81/82/83/84/85/86/87/88/89/90/91/92/93/94/95/96/97/98/99/100

- 784 input zu 10 zu 784 (28*28)
- Bottleneck size = 10 oder feature redutktion auf 10
- Random noise

DEC

Unsupervised Deep Embedding for Clustering Analysis

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis
└ Unsupervised Deep Embedding for Clustering Analysis
└ Unsupervised Deep Embedding for Clustering Analysis

- Besteht aus 2 teilen
 - embedding
 - Stacked Autoencoders
 - clustering
 - Clustern der embddeden Daten — Embeddings verbessern so wie die Cluster anpassen

DEC

Unsupervised Deep Embedding for Clustering Analysis

Stacked Autoencoders

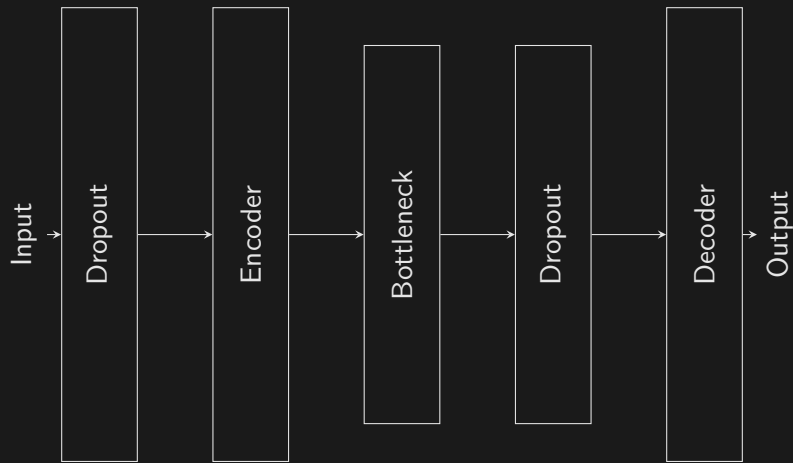
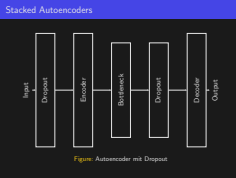


Figure: Autoencoder mit Dropout

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis
└ Unsupervised Deep Embedding for Clustering Analysis
└ Stacked Autoencoders
└ Stacked Autoencoders



Stacked Autoencoders ist genau das was der name sagt.
mehrere Autoencoder hintereinander.
man startet mit einem Autoencoder, den man trainiert.
dann entfernt man den decoder Teil
und setzt an diese stelle den nächsten Autoencoder, wird wieder trainiert
bis die gewünschte anzahl der stacks erreicht ist.
Besonderheit man trainiert die folgenden Autoencoder mit dem Output
des vorigen encoders.

Stacked Autoencoders

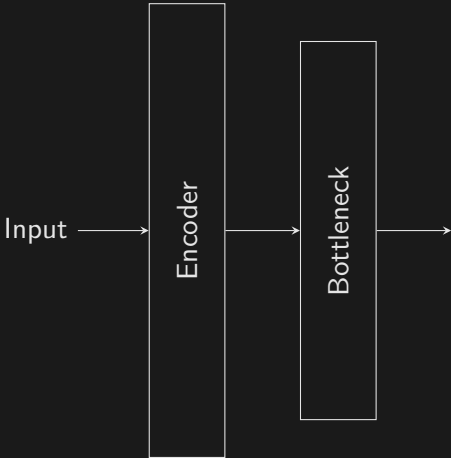


Figure: Encoder ohne Dropout

2022-02-04

- Unsupervised Deep Embedding for Clustering Analysis
 - Unsupervised Deep Embedding for Clustering Analysis
 - Stecked Autoencoders
 - Stacked Autoencoders

Stacked Autoencoders

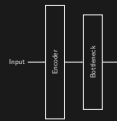


Figure: Encoder ohne Dropout

Encoder teil ohne noise/dropout und ohne dem decoder

Stacked Autoencoders

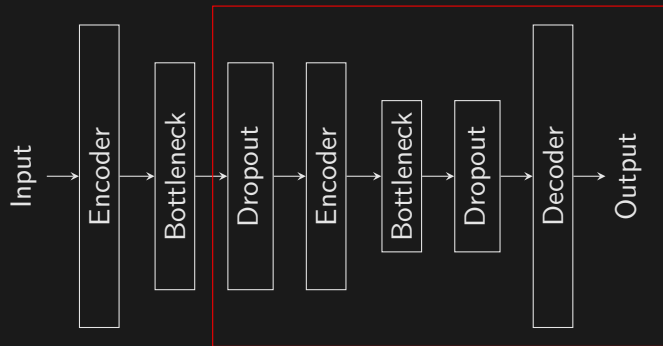
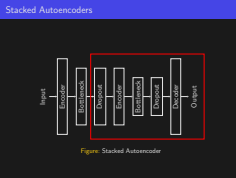


Figure: Stacked Autoencoder

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis
└ Unsupervised Deep Embedding for Clustering Analysis
└ Stacked Autoencoders
└ Stacked Autoencoders



Aktivierungsfunktion relu außer im letzten Decoder und Encoder Layer
sigmoid

für die zero-mean images

Zero-mean images - \bar{z} durchschnitt pro pixel über alle bilder im Datensatz
das bild wird dann von jedem bild abgezogen und damit liegt der durchschnitt bei den Bilder bei null deswegen zero mean images.

loss = least-square

nach dem Training des Autoencoders wird der Encoder als nicht lineares
mapping für die daten genutzt die man Clustern möchte

Deep embedded clustering

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Deep embedded clustering

└└ Kullback Leibler Divergenz

└└└ KL Divergenz

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Deep embedded clustering

└└ Kullback Leibler Divergenz

└└└ Soft Assignment

Minimierung von KI Divergenz

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

- └ Deep embedded clustering
 - └ Kullback Leibler Divergenz
 - └ Minimierung von KI Divergenz

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Deep embedded clustering

└└ Kullback Leibler Divergenz

└└└ Optimierung

Optimierung

andere Clustering algorithmen ? andere
Dimensions-Reduktions-algorithmen

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

- └ Deep embedded clustering
 - └ Kullback Leibler Divergenz
 - └ Vorherige Arbeiten

Von wem ist das Paper

Junyuan Xie
University of Washington

Ross Girshick
Facebook AI Research (FAIR)

Ali Farhadi
University of Washington

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Deep embedded clustering

└└ Kullback Leibler Divergenz

└└└ Von wem ist das Paper

Von wem ist das Paper

Junyuan Xie
University of Washington
Ross Girshick
Facebook AI Research (FAIR)
Ali Farhadi
University of Washington



Steinbach, Michael, Ertöz, Levent, and Kumar, Vipin. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pp. 273–309. Springer, 2004.



Ye, Jieping, Zhao, Zheng, and Wu, Mingrui. Discriminative k-means for clustering. In *NIPS*, 2008.

2022-02-04

Unsupervised Deep Embedding for Clustering Analysis

└ Referenzen

└ Referenzen

- Steinbach, Michael, Ertöz, Levent, and Kumar, Vipin. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pp. 273–309. Springer, 2004.
- Ye, Jieping, Zhao, Zheng, and Wu, Mingrui. Discriminative k-means for clustering. In *NIPS*, 2008.