Machine and Deep Learning

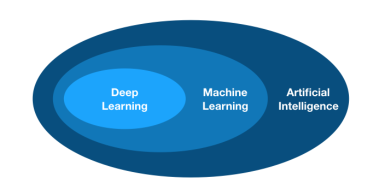
Holger Schmidt

April 18, 2020

Inhaltsverzeichnis

Grundglegende Begrifflichkeiten

Das Machine Learning ist Teilgebiet der Artificial Intelligence. Das Deep Learning ist wiederum ein Teilgebiet des Machine Learning.



Supervised und Unsupervised Machine Learning

Das Machine Learning gliedert sich grob in zwei Bereiche

- Supervised Learning = Gelabelte Daten (die Label werden zum Trainieren benutzt)
- Undsupervised Learning =ungelabelte Daten (bzw. die Label werden zum Trainieren nicht benutzt)

Training Examples

Für die Training Examples wird folgende Nomenklatur eingeführt

$$(\vec{x}^{(i)}, y^{(i)}), \quad i = 1, \ldots, m$$

Es gibt m Training Examples.

- Jeder der m **Inputs** enthält n features.

$$\vec{x}^{(i)} \in \mathbb{R}^n$$

- Für die m **Targets** gilt
  - Klassifikation mit K Klassen:

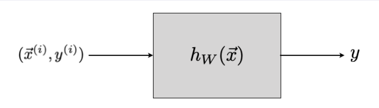
$$y^{(i)} \in \{0, 1, 2, \ldots, K-1\}$$

  - Regression

$$y^{(i)} \in \mathbb{R}$$

Ziel des Supervised Learing

Ausgehend von der Training Examples versucht, der Learning Algorithmus eine Hypothese $h_W(\vec{x})$ zu finden, um ein Target y möglichst gut vorherzusagen.



Linear Regression

Linear Regression

Bei der linearen Regression ist

$$(\vec{x}^{(i)}, y^{(i)}), \quad \vec{x}^{(i)} \in \mathbb{R}^n, \; y^{(i)} \in \mathbb{R}$$

aus. Unterscheide die beiden Fälle

- $n = 1$: Linearen Regression
- $n \geq 2$ Multivariate Lineare Regression

Hypothese und Costfunction

Die Hypothese wird durch eine mehrdimensionale lineare Funktion gebildet

$$h_{\vec{w}}(\vec{x}) = \vec{w}^{\,T} \cdot \vec{x} = w_0 + w_1 x_1 + w_2 x_2 + \ldots w_n x_n$$

wobei

$$\vec{x} = \begin{pmatrix} 1 & x_1 & x_2 & \ldots & x_n \end{pmatrix}^T, \vec{w} = \begin{pmatrix} w_0 & w_1 & w_2 & \ldots & w_n \end{pmatrix}^T.$$

Die Costfunction berechnet sich als mittlere quadratische Abweichung der Hypthose zum Target für jedes Training Example

$$J(\vec{w}) = J(w_0, w_1, \ldots, w_n) = \frac{1}{2} \sum_{i=1}^{m} \left(h_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)} \right)^2 .$$

Gradient Descent

Zur Bestimmung des optimalen $\vec{w}$ (=Minimum der Costfunction) wird die Gradient Descent Methode verwendet. Der Gradient errechnet sich zu

$$\vec{\nabla} J(\vec{w}) = \frac{1}{2} \sum_{i=1}^{m} \left(\vec{w}^{\,T} \cdot \vec{x}^{(i)} - y^{(i)} \right) \vec{x}^{(i)}$$

Folgende Iteration wird bis nun bis zur Konvergenz durchgeführt

$$\vec{w} \quad \to \vec{w} - \alpha \vec{\nabla} J(\vec{w})$$

Der Parameter α heisst Learning Rate.

R^2-Wert

R^2-Wert/ Bestimmtheitsmaß = Anteil der durch die Regression erklärten Quadratsumme an der zu erklärenden totalen Quadratsumme= wie viel Streuung in den Daten kann durch ein lineares Regressionsmodell erklärt werden

$$R^2 = \frac{\sum_{i=1}^{m} \left(h_{\vec{w}}(\vec{x}^{(i)}) - \bar{y} \right)^2}{\sum_{i=1}^{m} \left(y^{(i)} - \bar{y} \right)^2}$$

Linear Regression in sklearn

```
# Lineare Regression durchführen
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X,y)   # X mit shape (m,n) und y mit shape m

# Lineare Regression evaluieren
lr.predict(...)   # X Hypothese auswerten
lr.score(X,y)     # R^2 berechnen
```

Linear Regression - Fragen

- Wann wird LR angewendet?
- Wie sieht die Hypothese aus?
- Was bewirkt der Bias Therm?
- Was kann man tun, wenn die Daten nicht linear sind?
- Was versteht man unter Overfitting/Underfitting?
- Wie ist die Costfunction aufgebaut? Von was hängt sie ab?
- Was wird minimiert beim Gradient Descent Verfahren? Beschreiben Sie dieses Verfahren? Welche Funktion hat die Learning Rate α?
- Geben Sie ein Beispiel für eine lineare/multivariate Regression an.
- Wie kann eine Linear Regression in sklearn durchgeführt/evaluiert werden? In welcher Form müssen die Daten hier vorliegen?