Machine Learning

Mit Python

Inhalte:

* Grober Überblick
* Ein Beispiel, dass sich ohne Machine Learning so nicht lösen lassen würde
* Tutorial numpy und pandas
* Modellevaluation
* Sklearn API und ein Beispiel für Feature Extraction

Setup

https://github.com/ephes/data_science_tutorial

„The first rule of Python is you don't use system Python“

–Barry Warsaw

Wir verwenden conda, wie seit langem in der pydata-Community üblich. In letzter Zeit sieht das mit conda aber nicht mehr so toll aus :(.

„The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.“

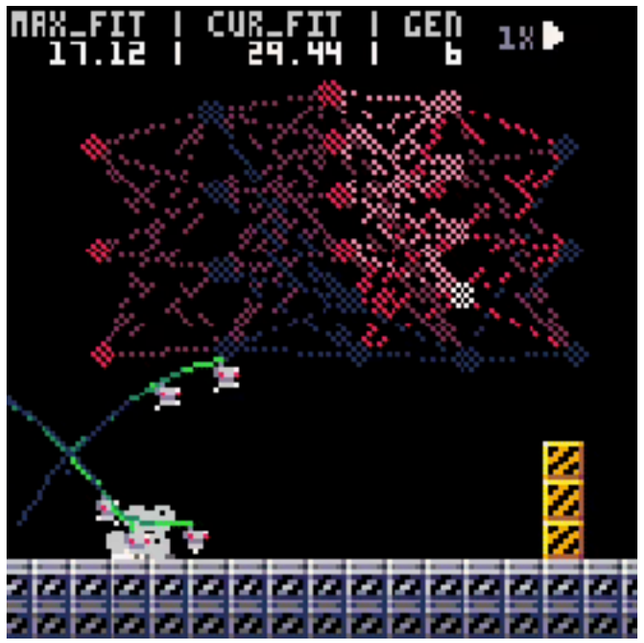
–Tom Mitchell 1997

Dabei interessiert uns vor allem die Generalisierungsperformance und nicht so sehr wie gut das Programm am Schluss ist. Man kann recht leicht Erfahrung und Energie gegen Ergebnis tauschen (siehe Schach), aber das ist eigentlich uninteressant.

„A computer program is said to **learn** from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.“

–Tom Mitchell

Ein neuronales Netz spielt ein pico-8 Spiel



https://github.com/fartenko/pico-nn

Google IO 2018 Duplex Demo

Warum ML?

- Warum überhaupt?
  - Es gibt Probleme, für die es sehr schwer ist, Algorithmen anzugeben (Spam vs Nichtspam, Hunde vs. Katzenbilder)
- Warum gerade jetzt?
  - Die Hardware ist enorm viel besser geworden
  - Bessere Datasets und Benchmarks (Internet)
  - Algorithmen: relu, RMSProp, Adam

Convnets und Backpropagation waren 1989 schon weitgehend verstanden.
LSTMs wurden 1997 vorgestellt.

Warum Erfolge erst nach 2012? -> Hardware und Daten

Bereiche in denen ML verwendet wird

- Suchmaschinen
- Spracherkennung
- Übersetzung
- Autonome Fahrzeuge
- Software, die auf den User zugeschneidert wird

DeepL

Wofür habe ich ML schon verwendet?

- Angebote in einen Kategorienbaum sortieren
- Angebote zu Produkten zusammenfassen
- Besseres Ranking der Suchmaschine
- Profileigenschaften von Webseitenbesuchern vorhersagen
- Einkaufen in Onlineshops
- Automatisches Bieten auf Anzeigenplätze
- Optimierung von Hotelpreisen
- Kaggle/Competitions
  - Yahoo learning to rank challenge
  - Predict closed questions on stackoverflow
  - Homedepot product search relevance

Wo funktioniert ML (noch) nicht?

- Es gibt einfache Spiele, bei denen ML grandios scheitert: Alles, was abstrakte Planung über einen längeren Zeitraum erfordert (Schlüssel holen und irgendwo hin bringen etc.)
- Aus Programmspezifikationen Implementationen generieren - finde ich ja besonders ärgerlich
- Was passiert im nächsten Frame eines Videos (wie sollte da überhaupt eine Verlustfunktion aussehen im Vergleich zum echten Frame?) - Menschen und Tiere haben damit erstaunlich wenig Probleme

Welche Arten ML gibt es?

1. Überwacht
2. Unüberwacht
3. Irgendwas dazwischen: Reinforcement Learning

Überwacht

Es gibt manuell erstellte Annotationen

- Klassisch
  - Klassifikation
  - Regression
- Structured Output
  - Sequence generation
  - Learning to rank
  - Syntax tree prediction
  - …

Selbstüberwacht

Es gibt keine menschlichen Annotationen

- Autoencoder (Target == Input)
- Die Wahrheit kommt aus der Zukunft (Aktienkurse, Conversion/Clickthrough rates..)
- Nächstes Wort eines Satzes generieren

Autoencoder nehmen Daten, beispielsweise ein Bild, generieren daraus einen möglichst kleinen Code und können aus dem kleinen Code wieder ein Bild erzeugen, dass bezüglich einer Verlustfunktion möglichst ähnlich zu dem Originalbild ist. In gewisser Weise ist das eine Form der Dimensionsreduktion.

Unüberwacht

- Clustering
- Dimensionsreduktion

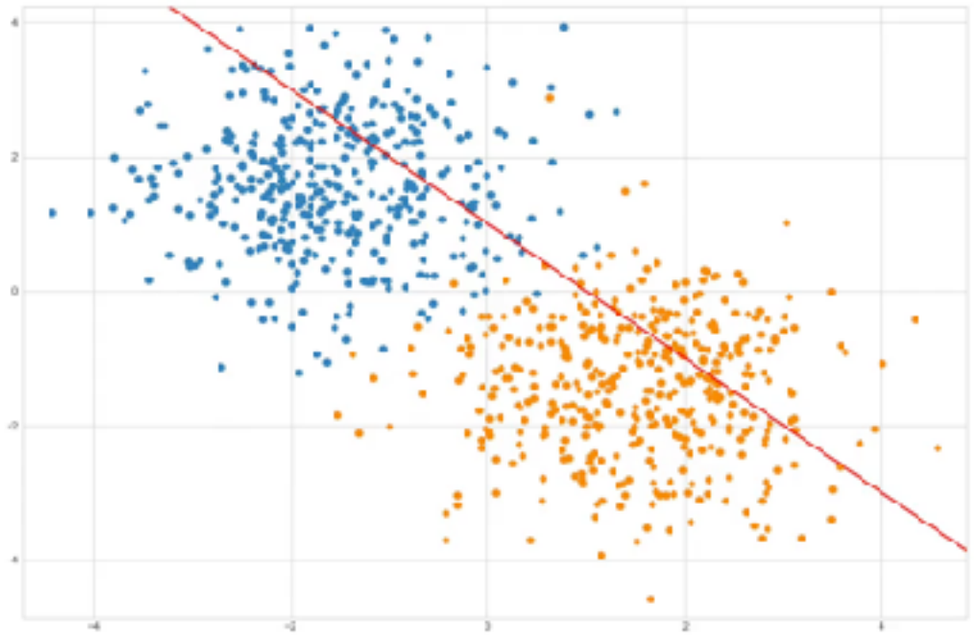
Dimensionsreduktion ist quasi Kompression und Kompression hat viel mit Verständnis zu tun, weil man Wesentliches von Unwesentlichem unterscheiden können muss, wenn man alles Unwesentliche weglassen möchte.

Kunst, Humor.

Reinforcement Learning

- Die Label sind sparse (man weiß beispielsweise erst am Ende eines Spiels, ob man gewonnen oder verloren hat, muss aber Belohnung/Bestrafung auf einzelne Züge aufteilen)
- Selbstfahrende Autos / Autonome Roboter etc.
- Hängt oft davon ab, ob sich eine Umgebung simulieren lässt (Lager voller Arme, optimal für Spiele, AlphaGoZero)

Perceptron



Mein Modell ist die rote Linie. Es soll lernen blaue von gelben Punkten zu unterscheiden.

Perceptron Implementation

```
def perceptron(features, targets, w):
    for x, target in zip(features, targets):
        if np.dot(w, x) <= 0:
            w += x * target
    return w
```

$target \in \{1, -1\}$

$\overrightarrow{w_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \overrightarrow{x} = \begin{bmatrix} x_0 \\ x_1 \\ 1 \end{bmatrix}$

$np.dot(w, x) = w_0 \cdot x_0 + w_1 \cdot x_1 + w_2 \cdot x_2$

Überblick über unterschiedliche ML-Verfahren

Naive Bayes

- Training
    1. Mach aus den Mails in deinem Spamordner eine lange Liste von Wörtern
    2. Zähle, wie oft ein Wort vorkommt
    3. Merke dir die Gesamtzahl der Wörter und die Gesamtzahl der Mails
    4. Wiederhole das Ganze für den Ordner mit Nichtspam-Mails
- Klassifikation
    1. Zerlege die zu prüfende Mail in eine Liste von Worten
    2. Mache aus der Wortliste eine Liste von Zahlen, indem du jedes Wort durch seine Kategoriefrequenz ersetzt
    3. Multipliziere alle diese Zahlen und a priori Wahrscheinlichkeit, dass eine Mail Spam oder Nichtspam ist miteinander
    4. Normalisiere das Ergebnis so, dass der Wert für Spam + Nichtspam 1 ergibt

Paul Graham: A plan for spam (2002)

Generative vs discriminative Modelle.

Support Vector Machines

- Lineare SVMs
  - SvmLight
  - liblinear
- SVMs mit Kernel
  - libsvm

Interessant, weil mathematisch gut verstanden. Margin Maximierung optimiert auch die Generalisierungsperformance (gibt es einen Beweis für).

Decision Trees

- CART
- Random Forests
  - scikit-learn
- Gradient boosting machines
  - xgboost
  - lightgbm
  - catboost

Ist das, was bei strukturierten Daten am besten funktioniert. Xgboost ownt kaggle. Random Forests sind fast immer das zweitbeste Modell über viele unterschiedliche Datasets hinweg.

Lineare Modelle

- Logistic Regression
- Linear Regression
- SGD
- Lasso / Elasticnet

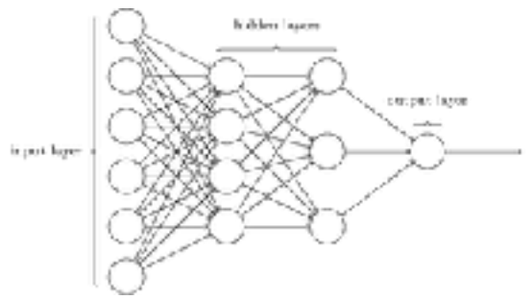
Sind sehr praxisrelevant. Schnell zu trainieren, robust, kommen mit sparsen Eingabevektoren klar, gut interpretierbar.

Man wird oft Richtung deep learning gedrängt. Obschon das, was der Kunde eigentlich braucht logistic regression auf gesäuberten Daten ist.

Künstliche Neuronale Netze

Gestapelte Funktionsapproximatoren

- Perceptron
- Multi Layer Perceptron
- Deep Learning
  - tensorflow, keras, pytorch, cntk, caffe



Perceptron, Rosenblatt XOR.

MLP vieleviele Perceptrons

Trainiert wird mit Backpropagation MLP - 80er Jahre. XOR geht.

AI-Winter

Fortschritte:
Rechner sind viel schneller geworden (danke Gamer). Dropout. Relu.

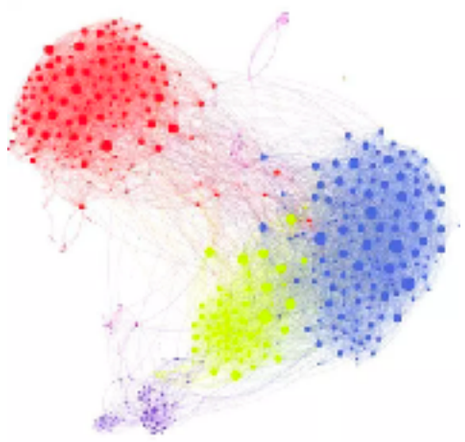
Der heisse Scheiss, z.Z.

Andere Interessante Verfahren

- KNN
- Hidden Markov Models, CRF
- Graphical Models, Bayesian Networks
  - pymc3

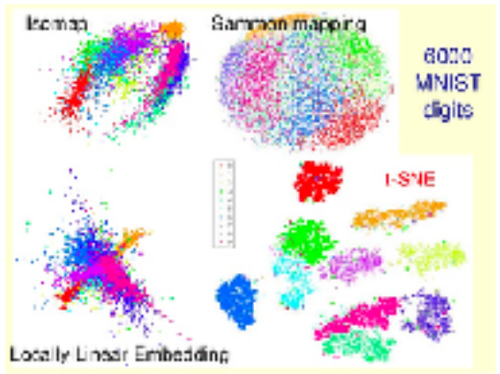
Clustering

- Kmeans
- Hierarchical Clustering
- Spectral Clustering



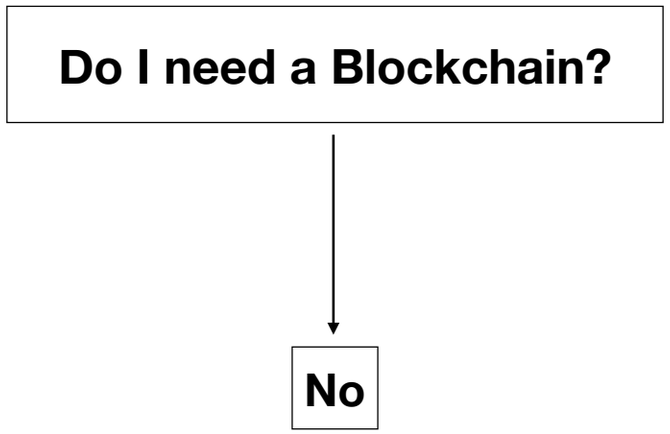
Dimensionsreduktion

- PCA
- SVD
- t-SNE



Trends

Big Data



„Consulting service: you bring your big data problems to me, I say "your data set fits in RAM", you pay me $10,000 for saving you $500,000.“

–Gary Bernhardt

Deep Learning

- Braucht man da nicht wahnsinnig viele Trainingsdaten?
- Sind solche Black-Box Modelle nicht ungeheuer intransparent?
- Deep Learning hat keine gute Theoriegrundlage, funktioniert aber trotzdem

An den vielen Trainingsdaten ist was dran. Wenn man viele Parameter hat, braucht man auch grosse Datenmengen, um die zu fitten. Aber es gibt auch vortrainierte Modelle, die man nutzen kann. Feature Extraction, Finetuning.

Deep Learning ist black box. Aber man kann da durchaus Dinge visualisieren (convnets). Andererseits - sind Menschen nicht auch black-box Modelle? Chickensexer.

https://www.youtube.com/watch?v=4wje7KurLvk&t=35m00s https://www.srf.ch/sendungen/sternstunde-philosophie/john-searle-der-sinn-des-bewusstseins

https://www.zeit.de/2001/21/200121_flugzeug.xml

„Wenn man mit Investoren spricht, nennt man es AI, gegenüber potentiellen Mitarbeitern nennt man es Machine Learning“

–Kann mich nicht mehr erinnern

Unterschiedliche Erwartungen bedienen.

AI oder was ist eigentlich general intelligence?

- Haben wir Menschen eigentlich generelle Intelligenz?
- Angenommen wir haben eine Menge von Karten, die auf der einen Seite Buchstaben und auf der anderen Seite Zahlen hat. Folgende Regel: wenn eine Karte ein „D“ auf der einen Seite hat, hat sie eine „3“ auf der anderen. Wie viele Karten müssen mindestens umgedreht werden, um die Karten unten zu überprüfen?

D F 3 7

https://www.youtube.com/watch?v=trfvI4JGtVg

Confirmation Bias

https://en.wikipedia.org/wiki/Wason_selection_task

Regel: Wenn du Alkohol in einer Bar trinken möchtest, musst du über 21 sein.

Wer muss nach dem Ausweis gefragt werden?

- Da sitzt jemand und trinkt Bier
- Da sitzt jemand, der eine Cola trinkt
- Jemand, von dem du weisst, dass er über 21 ist, sitzt an der Bar und trinkt etwas, von dem du nicht weißt, was es ist
- Und jemand, von dem du weisst, dass er noch keine 21 ist, trinkt ein unbekanntes Getränk

Hardware

GPU (nur Nvidia)

- Gamer GPUs (1080 TI, Titan XP)
- Server GPUs (k80, p100, v100)

TPU (nur Google)

- Schnell, aber es gibt keine guten Schnittstellen
- Alles nicht so einfach zu verwenden

Glossar

- Sample oder input - Ein Datenpunkt/Beobachtung
- Prediction oder output - Was das Modell ausgerechnet hat
- Target - Die Wahrheit. Das, was das Modell hätte ausspucken sollen
- Prediction error oder loss - Ein Mass für den Abstand zwischen prediction und target
- Label - Target im Fall eines Klassifikationsproblems.
- Learning rate - Wie stark wirkt sich der loss auf die Gewichte des Modells aus
- Epoch - Ein Durchgang über das komplette Trainingsset
- Feature - Eine Eigenschaft einer Beobachtung, „enthält das Wort Katze“ beispielsweise
- Metric - Eine Zahl, die man wichtig findet
- Objective - Die Zahl, die das Modell versucht zu optimieren