Deep Learning

Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

23-11-2023

Recap

- Exceptions
  - Handle all kinds of runtime errors
  - `raise` to throw errors
  - `try: ... except:` to catch them
- Python Packages
  - Use pip for installing python packages
- Types of DL tasks
  - Summerization, Sentiment Analysis, Question Answering, ...
  - Text classification, sequence labeling
- Classification
  - Different algorithms (Naive Bayes, Decision Trees, ...)
- Learning algorithm and prediction model

Today

Regression Tasks and Linear Regression

Classification Tasks and Logistic Regression

Loss Function

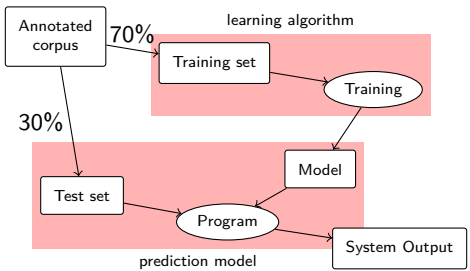
Gradient Descent

Scikit-Learn

Exercise

Recap Prediction Model and Learning Algorithm

Recap Prediction Model and Learning Algorithm



Blackbox problem

- We only know input (data) and output (i.e. f-score) of system
- We have only limited insight into the exact processes within the neural network

Blackbox problem

- We only know input (data) and output (i.e. f-score) of system
- We have only limited insight into the exact processes within the neural network
- ⚠ How then do we know if the system output is "good" or "bad"?

Blackbox problem

- We only know input (data) and output (i.e. f-score) of system
- We have only limited insight into the exact processes within the neural network
- ⚠ How then do we know if the system output is "good" or "bad"?
- Where we start: Understanding Learning Algorithms

Supervised learning

- The correct result/label is known
- System produces its own result/label ($\hat{y}$) (**hypothesis function**)
- We want the produced result ($\hat{y}$) to be as close as possible to the real result (y)
- Difference (loss) between y and $\hat{y}$ is determined (**loss function**)
- Loss is minimized as much as possible (**optimization algorithm**)

Section 1

Regression Tasks and Linear Regression

Regression

- Predicting a set or quantity
- Continuous variable –> Infinite number of possible values
  - e.g. age, distance, price, sales figures ...

Regression

- Predicting a set or quantity
- Continuous variable –> Infinite number of possible values
  - e.g. age, distance, price, sales figures ...

Example (Stock Price Prediction)

Can past stock prices be used to predict future stock prices?

Regression

- Predicting a set or quantity
- Continuous variable –> Infinite number of possible values
  - e.g. age, distance, price, sales figures ...

Example (Stock Price Prediction)

Can past stock prices be used to predict future stock prices?

- dependent variable –> stock prices
  - always continuous for regression tasks
- independent variable –> time
  - is used to predict dependent variable
- linear dependency between variables

Lineare Regression

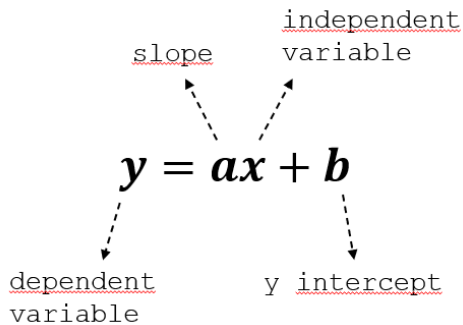
» Method for predicting **continuous values** (dependent variables) using independent variables

Lineare Regression

» Method for predicting **continuous values** (dependent variables) using independent variables

Lineare Regression

» Method for predicting **continuous values** (dependent variables) using independent variables



Section 2

Classification Tasks and Logistic Regression

Classification

- Assigning *classes* to *objects/instances/items*
- Binary (0 or 1, yes or no, A or B ...) and multi-class classification possible

Classification

- Assigning *classes* to *objects/instances/items*
- Binary (0 or 1, yes or no, A or B ...) and multi-class classification possible

Example (Nobel Prize Prediction)

Given the number of characters in a narrative text, will a book win the Nobel prize for literature?

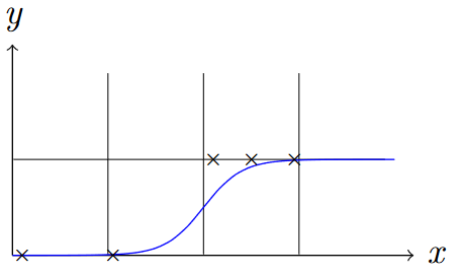
- dependent variable –> WIN or LOSE, YES or NO ...
  - categorical value (labels/classes)
- independent variable –> number of characters in a narrative text
  - is used to predict dependent variable

Logistic Regression

» Method for predicting **categorical values** (dependent variables) using a set of independent variables

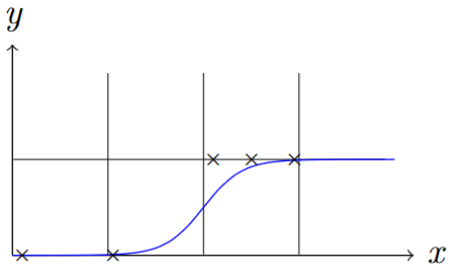
Logistic Regression

» Method for predicting **categorical values** (dependent variables) using a set of independent variables



Logistic Regression

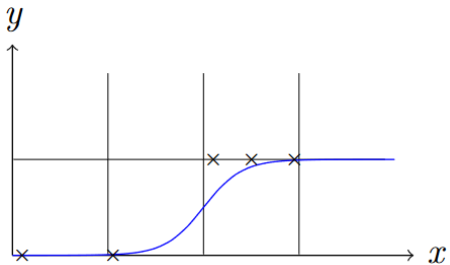
» Method for predicting **categorical values** (dependent variables) using a set of independent variables



$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}}$$

Logistic Regression

» Method for predicting **categorical values** (dependent variables) using a set of independent variables



$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}}$$

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 2.71828$$

Parameter Fitting

Linear Regression

$$h(x) = ax + b$$

Logistic Regression

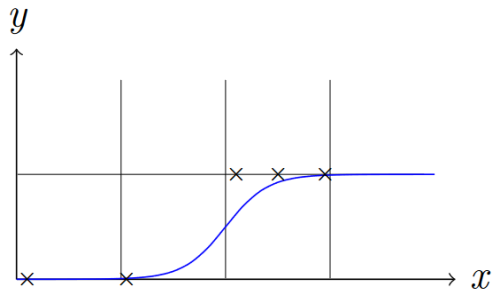
$$h(x) = \frac{1}{1 + e^{-(ax+b)}}$$

Learning algorithm: How to select the parameters a, b such that the hypothesis function describes the data points as best as possible?

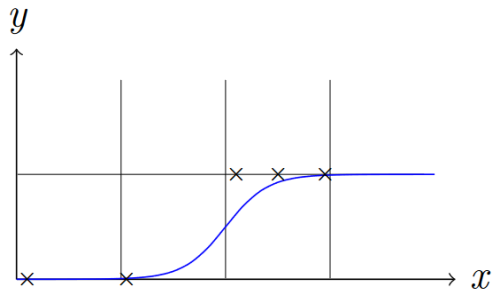
Section 3

Loss Function

Loss: Intuition

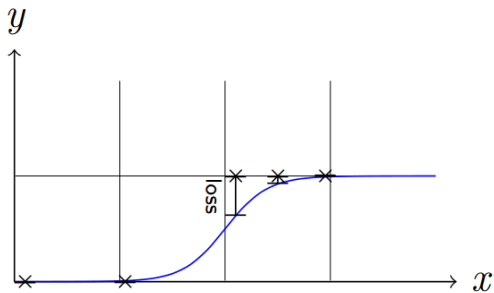


Loss: Intuition



- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss: Intuition



- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss function: Intuition

- Loss should be as small as possible
- Total loss can be calculated for given parameters $\theta = (a, b)$
- Hypothesis function h
  Calculates outcomes, given feature values x
- Loss function J
  Calculates ›wrongness‹ of h, given parameter values θ (and a data set)
  - In reality, θ represents more than two parameters

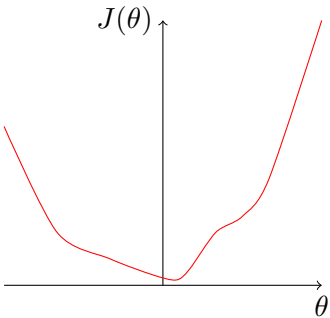
Loss function

Loss function depends on hypothesis function

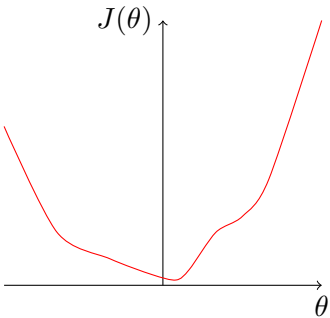
Linear
- » $h(x) = ax + b$
- » Loss: Mean squared error

Logistic
- » $h(x) = \frac{1}{e^{-(b+ax)}}$
- » Loss: (Binary) cross-entropy loss
  - Not the only choice

Loss function

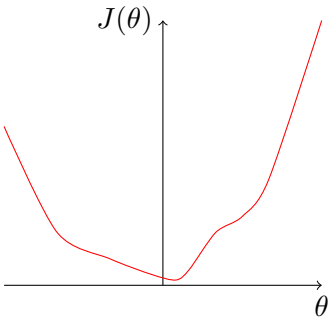


Loss function



$$J(a,b) = -\frac{1}{m}\sum_{i=0}^{m} y_i \log h_{a,b}(x_i) + (1-y_i)\log(1-h_{a,b}(x_i))$$

Loss function



$$J(a,b) = -\frac{1}{m}\sum_{i=0}^{m} y_i \log h_{a,b}(x_i) + (1-y_i)\log(1-h_{a,b}(x_i))$$

Now that we know the loss we want to minimize it!

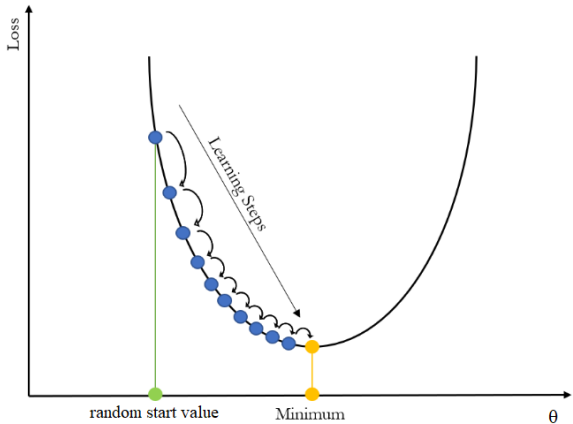
Section 4

Gradient Descent

Gradient Descent

- Initialise θ with random values (e.g., 0)
- Repeat:
  - Find the direction to the minimum by taking the derivative
  - Change θ accordingly, using a learning rate η
  - Stop when θ don't change anymore

Gradient Descent



Section 5

Scikit-Learn

Introduction

- Generic machine learning library for Python
- Classification algorithms: Naive Bayes, support vector machines, ...
- Clustering algorithms: KMeans, agglomerative clustering, ...

Introduction

- Generic machine learning library for Python
- Classification algorithms: Naive Bayes, support vector machines, ...
- Clustering algorithms: KMeans, agglomerative clustering, ...
- Utility functions
  - Cross-validation, training and test splits
  - Evaluation (precision/recall/f-score)

Workflow

Preprocessing and Preparations

» Preprocessing

- Read in data files
- Remove columns that we cannot handle or columns that we feel are irrelevant
- Convert features into numeric representations
- Split into train and test data
- Deal with missing values
- Add additional features from other resources
- ...

Workflow

Preprocessing and Preparations

- Preprocessing
  - Read in data files
  - Remove columns that we cannot handle or columns that we feel are irrelevant
  - Convert features into numeric representations
  - Split into train and test data
  - Deal with missing values
  - Add additional features from other resources
  - ...
- Preparations
  - Split into x, y
  - Split into train and test
- Machine learning $\rightarrow$ next slide

Workflow

Machine Learning

- Initialize object `cls = Classifier(...)`
- Call `cls.fit(x_train, y_train)` to train
- Call `y_pred = cls.predict(x_test)` to get predictions on the test set
- Call `*_score(y_test, y_pred)` to calculate evaluation scores

Workflow

Machine Learning

- Initialize object `cls = Classifier(...)`
- Call `cls.fit(x_train, y_train)` to train
- Call `y_pred = cls.predict(x_test)` to get predictions on the test set
- Call `*_score(y_test, y_pred)` to calculate evaluation scores
- »Initialize–fit–predict« pattern

Section 6

Exercise

Exercise 04

https://github.com/IDH-Cologne-Deep-Learning-Uebung/exercise-05