# Deep Learning

## Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

21-12-2022

# Recap

- Embeddings
  - mapping of words or documents to vectors
  - ›embedded‹ in a vector space
  - allows calculating similarity between words
- Implementing Embeddings in Keras
  - Two relevant new layers
    - `tensorflow.python.keras.layers.Embedding()`
    - `tensorflow.python.keras.layers.Flatten()`
  - Preperations
    - `tensorflow.keras.preprocessing.text.Tokenizer()`
    - `tensorflow.keras.preprocessing.text.text_to_word_sequence()`
    - `tensorflow.keras.preprocessing.sequence.pad_sequences()`

# Recap

- Embeddings
  - mapping of words or documents to vectors
  - ›embedded‹ in a vector space
  - allows calculating similarity between words
- Implementing Embeddings in Keras
  - Two relevant new layers
    - `tensorflow.python.keras.layers.Embedding()`
    - `tensorflow.python.keras.layers.Flatten()`
  - Preperations
    - `tensorflow.keras.preprocessing.text.Tokenizer()`
    - `tensorflow.keras.preprocessing.text.text_to_word_sequence()`
    - `tensorflow.keras.preprocessing.sequence.pad_sequences()`
- Exercise 8

# Today

What we have learned so far...

- Python
- NLP and Deep Learning
- Hypothese, Loss, Optimizer
- Feed-forward Neural Networks
- Training (Overfitting, Drop Out, Regularization)
- Input Representation
- In der Praxis...

Section 1

# What we have learned so far...

Subsection 1

Python

# Python Functionalities

- List Comprehension
  - Define lists by specifying a pattern
    - `[x*2 for x in l1 if x < 10]`
- Functions
  - Named arguments, default values
  - Return values, `None` and `NoneType`
- Input/Output
  - Stream-oriented
  - Open file, work with stream, close file

# Python Functionalities

- Exceptions
  - Handle all kinds of runtime errors
  - `raise` to throw errors
  - `try: ... except:` to catch them
- Python Packages
  - Use pip for installing python packages

# Literatur, Webseiten und Dokus

- `https://docs.python.org/3/`
- Library reference: `https://docs.python.org/3/library/index.html`
- Tutorial
  - Al Sweigart: `https://automatetheboringstuff.com`
  - Also available as printed book and YouTube series
- IO:
  - `https://docs.python.org/3/library/functions.html#open`
  - `https://docs.python.org/3/library/io.html#module-io`
- Error Handling:
  - `https://docs.python.org/3/tutorial/errors.html`
  - `https://docs.python.org/3/library/exceptions.html`
- Central repository for python libraries: `https://pypi.org`

Subsection 2

## NLP and Deep Learning

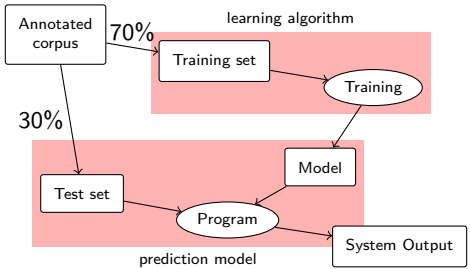
# DL Tasks

» Types of DL tasks for Natural Language Processing

- Summerization, Sentiment Analysis, Question Answering, ...
- Classification
  - Text classification: An entire text is classified (e.g., genre, sentiment, . . . )
  - Sequence labeling: Each individual word is classified (e.g., pos-tagging, . . . )
  - binary and multi-class possible

# Prediction Model and Learning Algorithm

# Prediction Model and Learning Algorithm



Subsection 3

## Hypothese, Loss, Optimizer

# 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**)

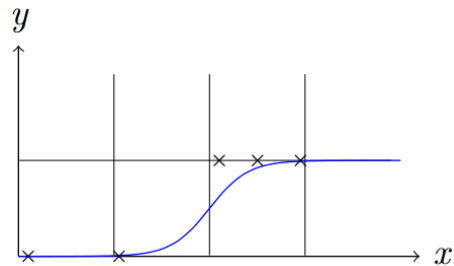
# Linear regression

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

# Logistic regression

- Assigning *classes* to *objects/instances/items*
- Binary (0 or 1, yes or no, A or B ...) and multi-class classification possible
- Method for predicting **categorical values** (dependent variables) using a set of independent variables

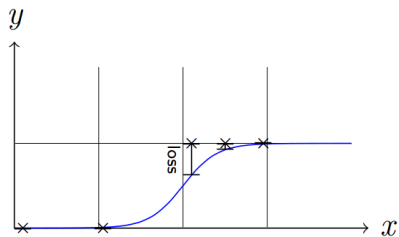
$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}}$$



# Loss Function

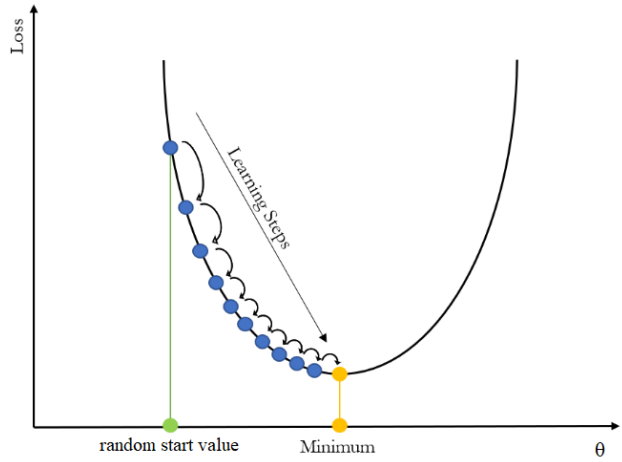
Learning algorithm: How to select the parameters $a, b$ such that the hypothesis function describes the data points as best as possible?

- How big is the gap between a hypothesis and the data?
- Loss should be as small as possible
- Total loss can be calculated for given parameters $\theta = (a, b)$
- Loss function $J$
  Calculates ›wrongness‹ of $h$, given parameter values $\theta$ (and a data set)
  - In reality, $\theta$ represents more than two parameters



# Gradient Descent

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



Subsection 4

## Feed-forward Neural Networks

# What is a neural network?

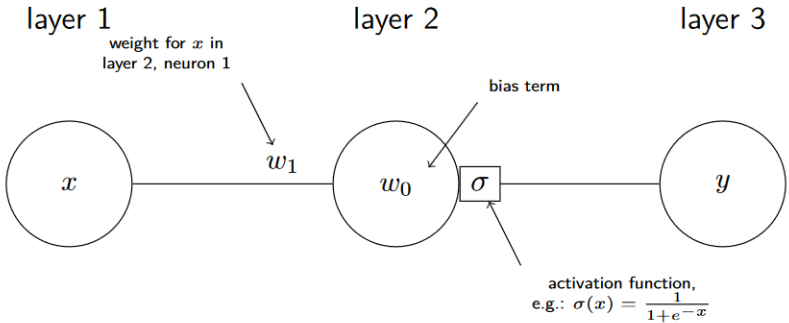



Figure: 1 neuron (with logistic activation) = logistic regression (with 1 feature)

# What is a neural network?

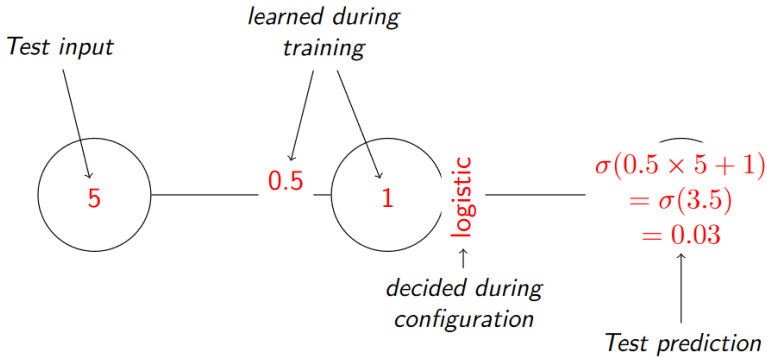
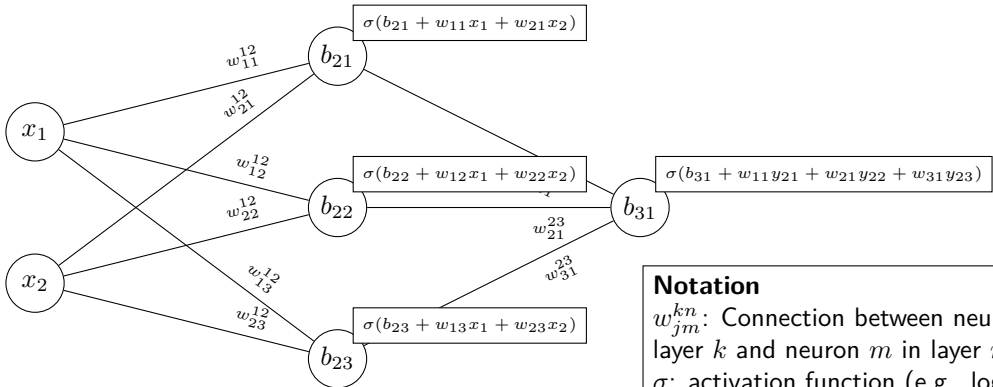



Figure: 1 neuron (with logistic activation) = logistic regression (with 1 feature)

# Feed-forward neural network




**Notation**

$w_{jm}^{kn}$: Connection between neuron $j$ in layer $k$ and neuron $m$ in layer $n$

$\sigma$: activation function (e.g., logistic)

Figure: A feed-foward neural network with 1 hidden layer

# Feed-forward neural networks

- The above is called a »feedforward neural network«
  - Information is fed only in forward direction
- Configuration choices
  - Activation function
  - Layer size: Number of neurons in each layer
  - Number of layers
  - Loss function
  - Optimizer
- Training choices
  - Epochs/batches
  - Training status displays

- All neurons of one layer have the same activation function
- Popular choices for activation functions:

logistic $y = \sigma(x) = \frac{1}{1+e^{-x}}$ – ›squashes‹ everything to a value between 0 and 1

relu $y = \max(0, x)$ – Makes everything negative to 0

softmax Scales a vector such that values sum to 1 (probability distribution)

Subsection 5

## Training (Overfitting, Drop Out, Regularization)

# Overfitting

- ›Fitting‹: Train a model on data (= »fit« it to the data)
  - Underfitting: The model is not well fitted to the data, i.e., accuracy is low
  - Overfitting: The model is fitted too well to the data, i.e., accuracy is high

## Why is overfitting a problem?

- We want to the model to behave well »in the wild«
- It needs to generalize from training data
- If it is overfitted, it works very well on training data, and very badly on test data

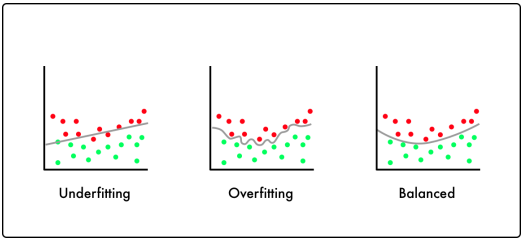



Figure: Towards Data Science

# Overfitting

There is no one solution for overfitting!

## Techniques against overfitting

- Regularization (numerical)
- Dropout (structurally)

# Regularization

» Formally, regularization is a parameter added to the loss

$$J(\vec{w}) = J_{\text{original}}(\vec{w}) + R$$

# Regularization

## $L^2$

- »

$$(||\vec{w}||_2)^2 = \sum_{i=0}^{n} w_i^2$$

- » Regularization rate $\lambda$: Factor that expresses how much we want (another hyperparameter)
- » What does it do?
  - If weights $\vec{w}$ are large: Loss is increased more
  - Large weights are only considered if the increased loss is »worth it«, i.e., if it is counterbalanced by a real error reduction
  - Small weights are preferred

## $L^1$

- »

$$L^1(\vec{x}) = \sum_{i=0}^{n} |x_i|$$

### $L^1$ or $L^2$?

- » Skansi 2018:
  - In most cases: $L^2$ is better
  - Use $L^1$ if data is very noisy or sparse

# Dropout

» Structurally combatting overfitting
- Hinton et al. (2012)
- A new hyperparameter $\pi = [0; 1]$
- In each epoch, every weight is set to zero with a probability of $\pi$

# Dropout

## Example

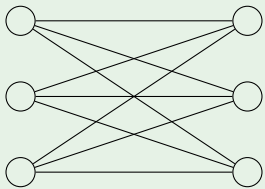


Figure: Dropout $\pi = 0.5$, visualized

# Dropout

## Example

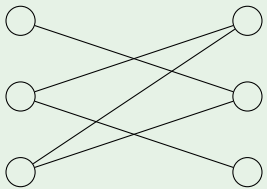


Figure: Dropout $\pi = 0.5$, visualized, Epoch 0

# Dropout

## Example

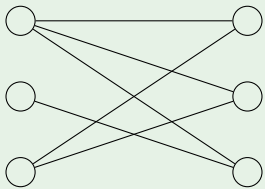


Figure: Dropout $\pi = 0.5$, visualized, Epoch 1

# Dropout

## Example

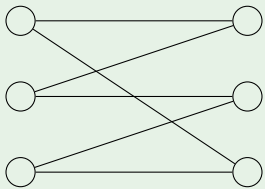


Figure: Dropout $\pi = 0.5$, visualized, Epoch 2

Subsection 6

## Input Representation

# Structured Data - Tables

- i.e. Titanic data set
- Objects (passengers) are described with the help of various properties (name, sex, ticket, age, cabin, ...)
- Number of features gives us the input shape
- Input that is not an integer is converted to an integer
- Input is a vector with the size of the feature count

| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22 | 1 | 0 | A/5 2117 |
| 2 | 1 | 1 | Cumings, Mrs. John Bradley (Flo | female | 38 | 1 | 0 | PC 17599 |
| 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26 | 0 | 0 | STON/O2 |
| 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lil | female | 35 | 1 | 0 | |
| 5 | 0 | 3 | Allen, Mr. William Henry | male | 35 | 0 | 0 | |
| 6 | 0 | 3 | Moran, Mr. James | male | | 0 | 0 | |
| 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54 | 0 | 0 | |
| 8 | 0 | 3 | Palsson, Master. Gosta Leonard | male | 2 | 3 | 1 | |
| 9 | 1 | 3 | Johnson, Mrs. Oscar W (Elisabet | female | 27 | 0 | 2 | |
| 10 | 1 | 2 | Nasser, Mrs. Nicholas (Adele Ac | female | 14 | 1 | 0 | |

• • •

• • •

# Bag of Words

- Disregards word order and semantic
- A vocabulary is established
- Words are counted in order of vocabulary in a text
- Size of vocabulary gives us the input shape
- Input is a vector with the size of the vocabulary

| Abend | Adresse | also | auf | ... | bei | beugen | Blume | Brief | ... | und | Urlaub | ... | Zaun | zeigen |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 4 | | 3 | 0 | 0 | 1 | | 7 | 2 | | 0 | 3 |

...

# Embeddings

- An embedding is a mapping of words or documents to vectors
  - Things are ›embedded‹ in a vector space
  - Representing a word as a vector allows calculating similarity between words
- Word vectors: Weights learned by a simple neural network with a classification target
  - word2vec: Given word $w_i$, how likely is it that $w_j$ appears in its context?
- Idea
  - Embeddings are learned using a neural network
  - Classification task: Given a word, predict its context words
  - Use learned weights as embeddings

# Embeddings and neural networks

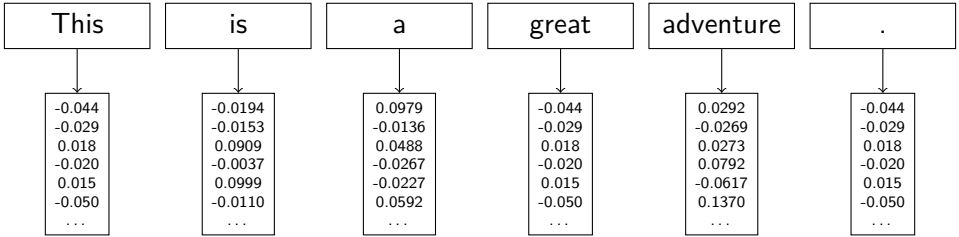
- Existing (pre-trained) embeddings can be plugged in
- Specific embeddings can be trained, just like all other weights

## Pre-trained embeddings

- Glove (Stanford University): `https://nlp.stanford.edu/projects/glove/`
- FastText (facebook research): `https://fasttext.cc` (multiple languages)

```
adventure 0.0292 -0.0269 0.0273 0.0792 -0.0617 0.1370 -0.0628 0.0420
0.0743 0.0979 -0.0136 0.0488 -0.0267 -0.0227 0.0592 0.0410 0.0314 0.0378
-0.0455 0.0616 -0.0380 0.0232 -0.0218 0.0000 -0.0699 -0.1327 -0.0393
0.0467 0.0413 0.0089 -0.0046 0.0372 -0.0590 0.0740 0.0214 0.0625 0.0067
-0.0063 0.0218 -0.0447 -0.0298 0.0186 -0.0207 0.0158 -0.0508 -0.0297
-0.0807 -0.0619 -0.0194 -0.0153 0.0909 -0.0037 0.0999 -0.0110 ...
```

# Embeddings and neural networks




- Our input now consists of a matrix (per instance)
- Matrix size needs to be predefined
  - Embedding dimension: Parameter we can set freely
  - Length: To be set on training data
- Input length
  - This parameter controls how long sentences (or texts) can be
- Padding
  - Extend shorter inputs so that they have the same length
  - Truncate longer inputs

Subsection 7

## In der Praxis...

# Libraries

| | |
|---|---|
| Tensorflow | python machine learning platform (`https://www.tensorflow.org/`) |
| Keras | deep learning library (part of Tensorflow) (`https://keras.io/`) |
| Sci-kit learn | python machine learning library for predictive data analysis (`https://scikit-learn.org/stable/`) |
| NumPy | python library for scientific computing (`https://numpy.org/`) |