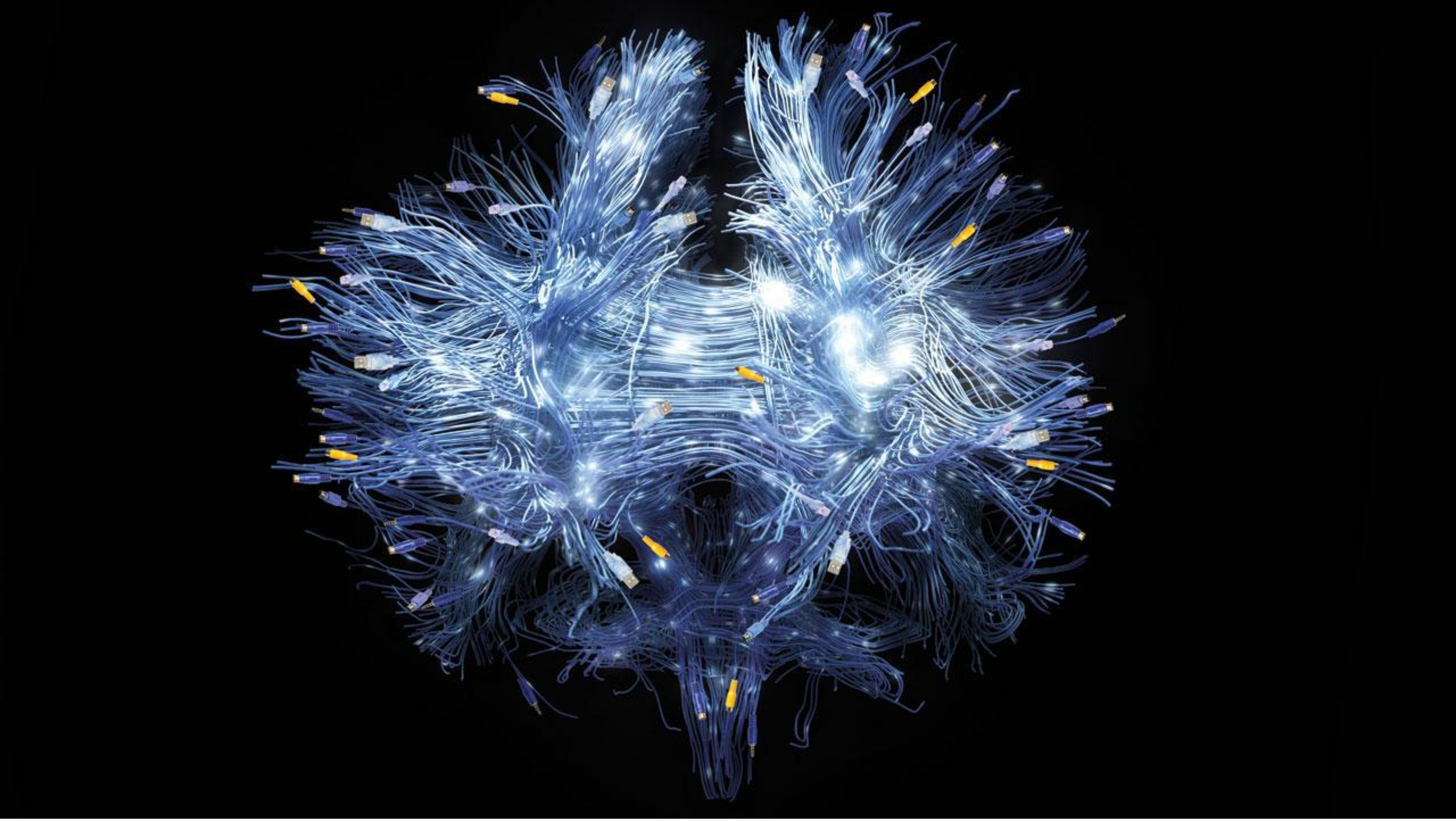


Summer School – Neural Networks with Text

28.09.18, Johanna Binnewitt, Valmir Etemi, Julia Kappes

- Introduction
- Organisation
- Data Science
- Neural Network Basics
- Tools and Frameworks
- Classification with Neural Networks

Introduction

Organisation

Data Science

Neural Network Basics

Tools and Frameworks

Classification with Neural Networks

Introduction

What do you expect?

What are we going to learn?

- Neural Network Basics
- How to use Neural Network Frameworks and Tools
- How to use Neural Networks for Text Classification
- How to use Neural Networks for Text Generation
- Python ☺

Which Technologies are we going to use?

- Jupyter Notebook
- Python
- Keras

Introduction

Organisation

Data Science

Neural Network Basics

Tools and Frameworks

Classification with Neural Networks

Organisation

Schedule

Time	Friday, September 28	Saturday, September 20
9:00	Introduction Workshop / NN Basics	Hands on Natural Language Generation
10:30	*Coffee Break*	
11:00	Tutorial Text Classification	Discussion Natural Language Generation
12:30	*Lunch Break*	Closing
14:00	Discussion Classification	
14:30	Introduction Natural Language Generation	
15:30	*Coffee Break*	
16:00	Hands on Natural Language Generation	
17:00	Closing	

Introduction

Organisation

Data Science

Neural Network Basics

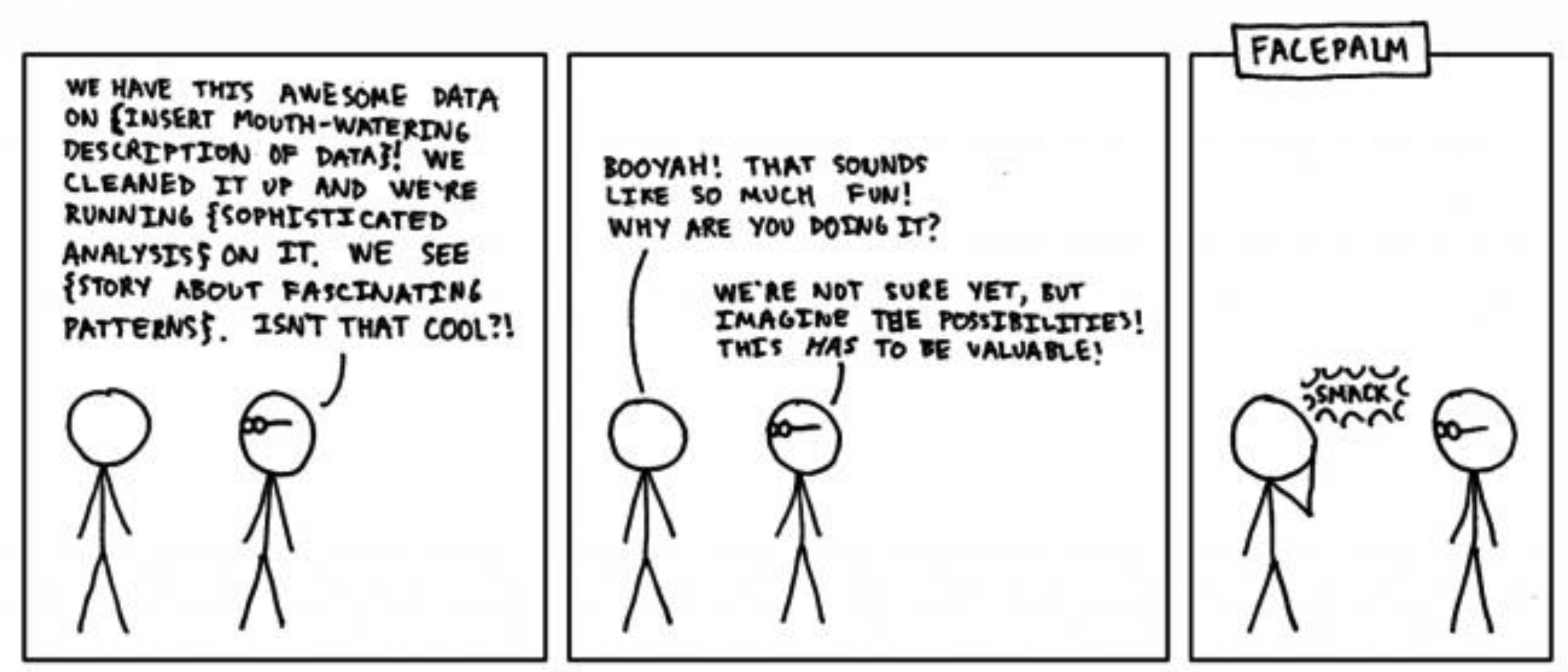
Tools and Frameworks

Classification with Neural Networks

How do Data Scientists work?

1. Define Research Goal
2. Retrieve Data
3. Prepare Data
4. Explore Data
5. Model Data
6. Present and automate Model

1. Define Research Goal



https://medium.com/the-data-experience/building-a-data-pipeline-from-scratch-32b712cfb1db

Retrieve Data

Use open data sites:

> https://www.gutenberg.org/ (for text)

> https://archive.ics.uci.edu/ml/datasets.html

Use internal data if available

Prepare Data

- Data cleansing (remove false, inconsistent or unnecessary data)
- Data integration (enrich data with other sources)
- Data transformation (Transform into suitable format)
- **How to transform text into a model?**

Explore Data

- Understand retrieved data
- How are variables interacting?
- Use of descriptive statistics, plotting techniques and simple modeling

Model Data

- Build a model which suits research goal

- Machine Learning models
- **Neural Networks**
- Etc.

Types of Data

- Structured data (e.g. SQL databases)
- Semi-structured data (e.g. CSV files)
- Unstructured data (e.g. text files)

- Machine generated (e.g. server log files)
- Natural Language
- Audio, video, images
- Streaming

Types of Data

- Structured data (e.g. SQL databases)
- Semi-structured data (e.g. CSV files)
- Unstructured data (e.g. text files)

- Machine generated (e.g. server log files)
- **Natural Language**
- Audio, video, images
- Streaming

Introduction

Organisation

Data Science

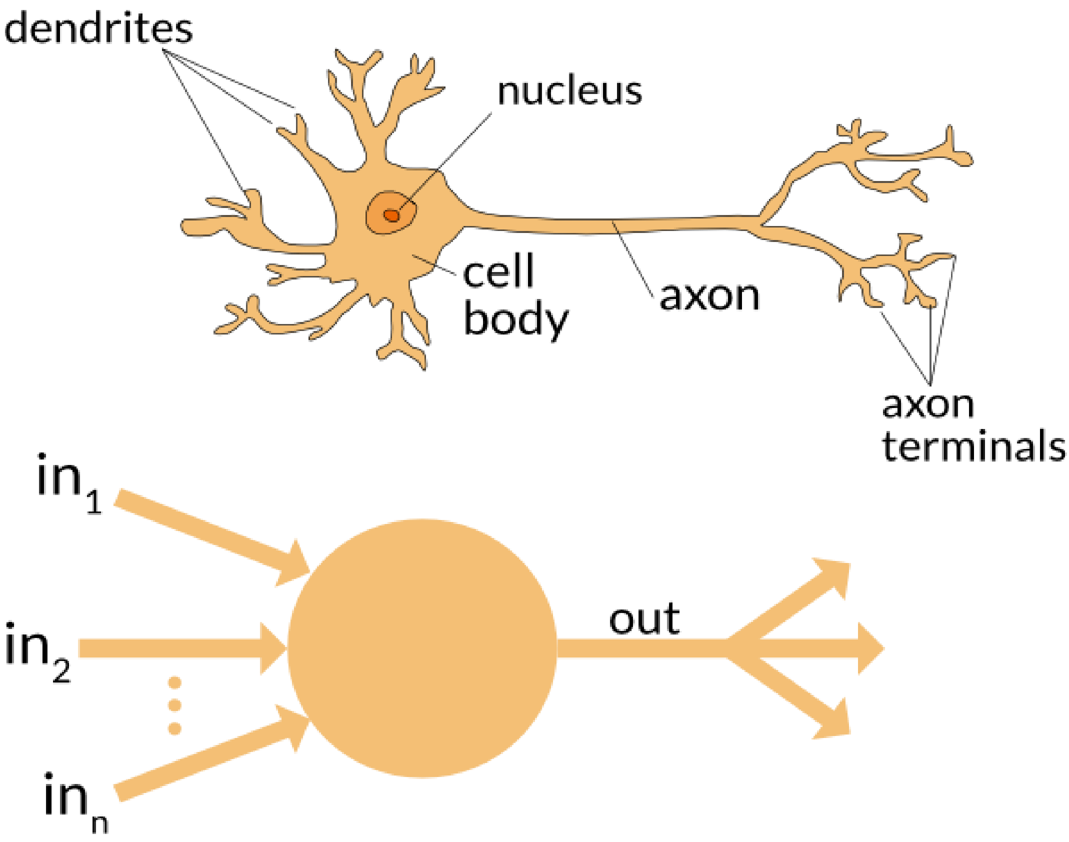
Neural Network Basics

Tools and Frameworks

Classification with Neural Networks

Neural Network Basics

Biological Inspiration



Source: https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

Perceptron (1)

Input data:

Numerical values, e.g. blood values of patients:

in1 (Amount of iron) = 1

in2 (Amount of white blood cells) = 5

in3 (Something else) = 0

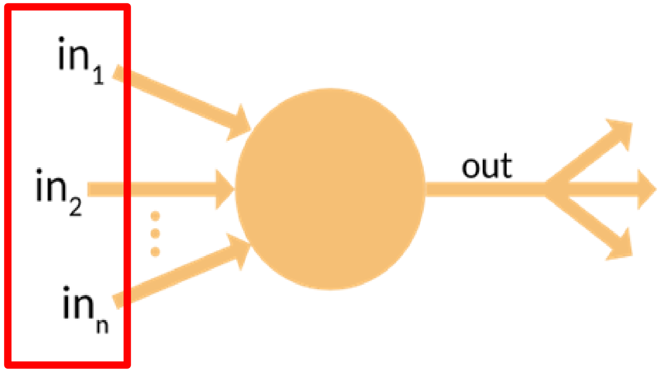
=> Vector: { 1, 5, 0 }

Input data consists of values and its class, e.g.

patient1 = {1, 0, 1} → Klasse 1 (krank)

patient2 = {1, 1, 1} → Klasse 0 (nicht krank)

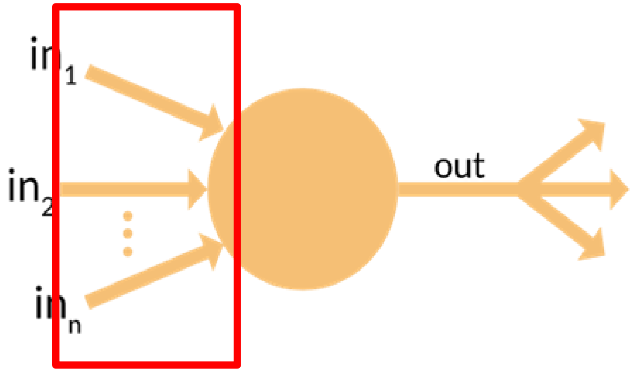
…



Perceptron (2)

Weights:

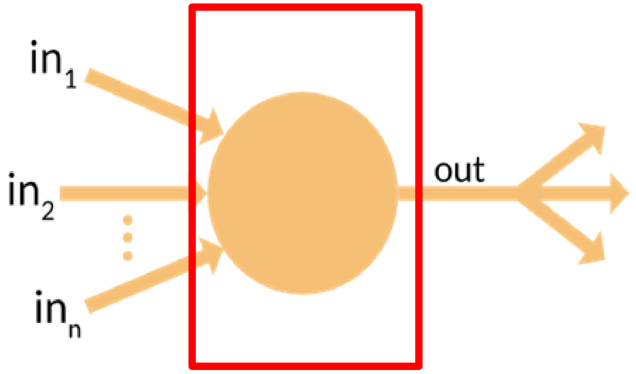
- That is our model
- Weights will be adjusted during the training in order to calculate the right class for the given training data.



Perceptron (3)

Prediction:

> The calculation is based on the dot product of the vector of the input data and the vector of the weights



Example:

> Input Data: {2, 3, 2},

> Weights: {0.5, -0.2, 0.1}

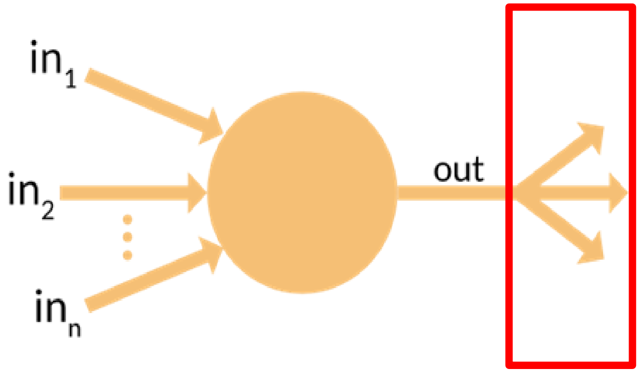
> *Calcuation:*

> 2 * 0.5 + 3 * (-0.2) + 2 * 0.1 = 0.6

Perceptron (4)

Output:

- The output is the result of the dot product and an activation function.
- Determines the class
- Activation functions can be super simple functions, e.g. thresholds or more complex ones, e.g. Sigmoid



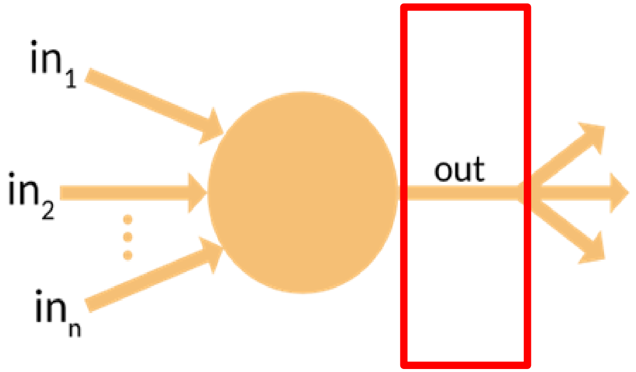
Example:

- Result: 0.6,
- Threshold: 0.5
- Class → 1

Perceptron (5)

Training:

> If calculation of prediction outputs a wrong class, then the weights have to be adjusted in order to calcuate the correct class



Example:

> Result: 0.6,

> Threshold: 0.5

> Target class = „0“ (Actual class „1“)

→ *Error = 0 (Target) – 1 (Result) = -1*

Perceptron (6)

Optimizer:

> Our weights have to be adjusted so that the output is less then the treshold (0.5)

> Optimizer functions can also be simple, e.g. Hebbian learn rule or very complex

Example (Hebbian learn rule):

Adjust = *Learning rate * Input value * Error*

Adjust1 = 0.02 * 2 * (-1) = -0.04

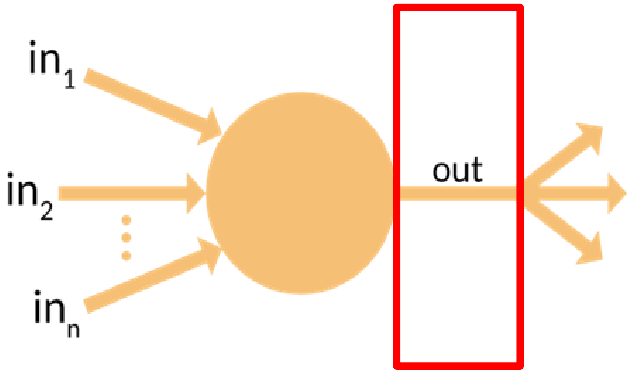
New weight 1 = *Old weight + Adjustment*

0.5 – 0.04 = 0.46

Weights:

Weights before: { 0.5, -0.2, 0.1 }

Weights after: { 0.46, -0.26, 0.06 }



Multi Class Problem

- Previously only two classes: „0“ or „1“ (Spam, not spam, etc.)

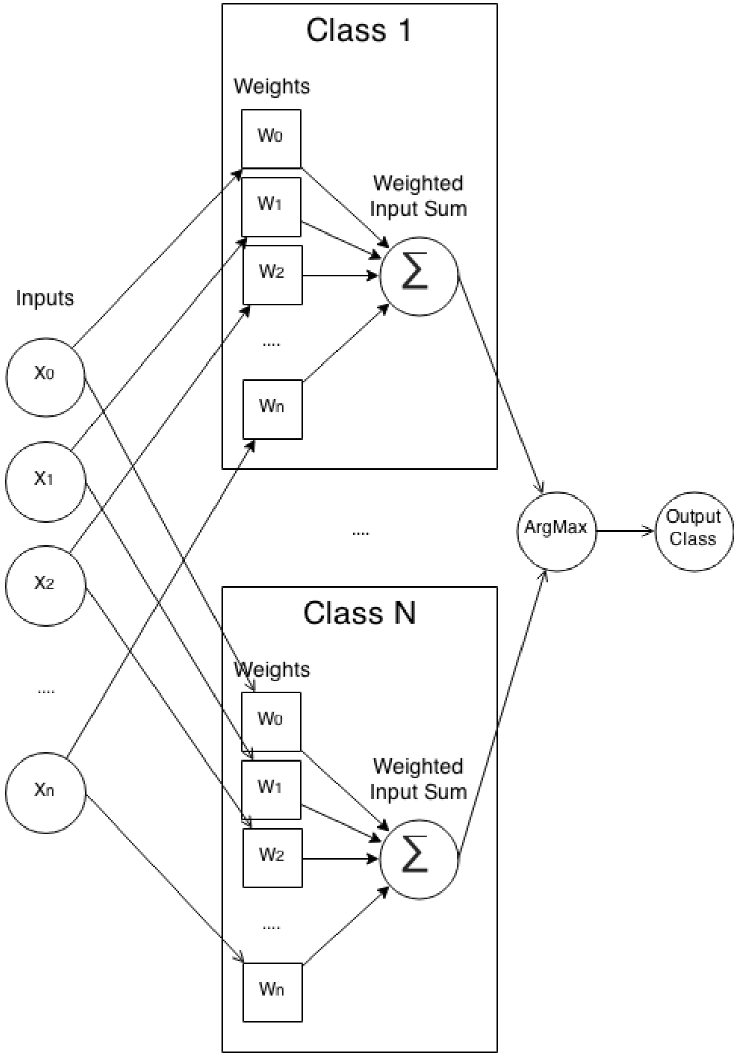
Example for Multi Class Problem:

- Image Recognition
- Each picture (dog, cat, mous, etc.) equals to one class

Multi Class Classification – Perceptron?

- Is it possible to use perceptrons for multiple classes?
- If not, what could we change to make it work?

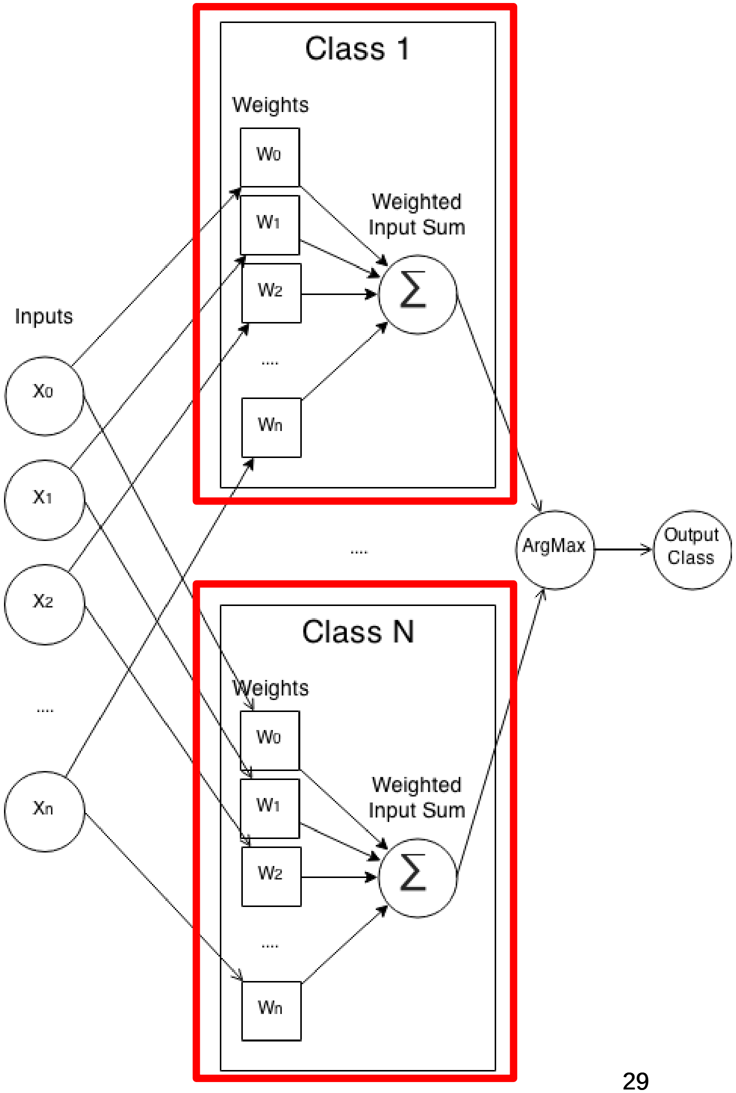
Multi Class Classification – Perceptron! (1)



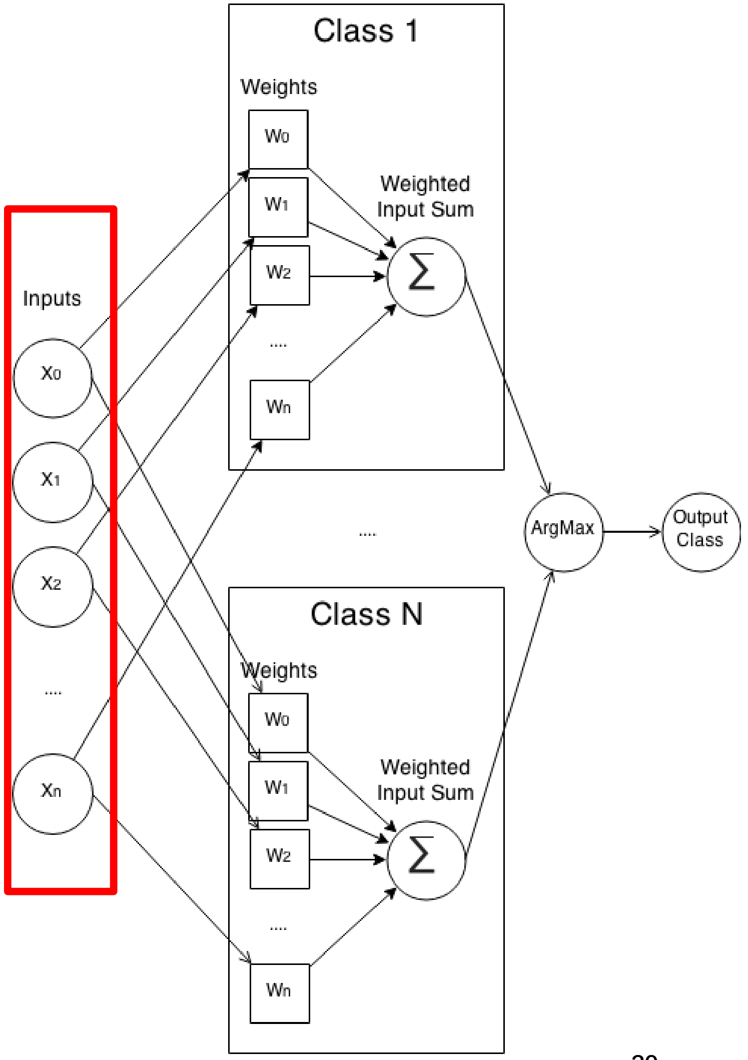
Multi Class Classification – Perceptron! (2)

Calculation:

- Each class is one perceptron and will be trained separately
- That is, each class/perceptron has its own set of weights to train
- Calculation remains the same



Multi Class Classification – Perceptron! (3)



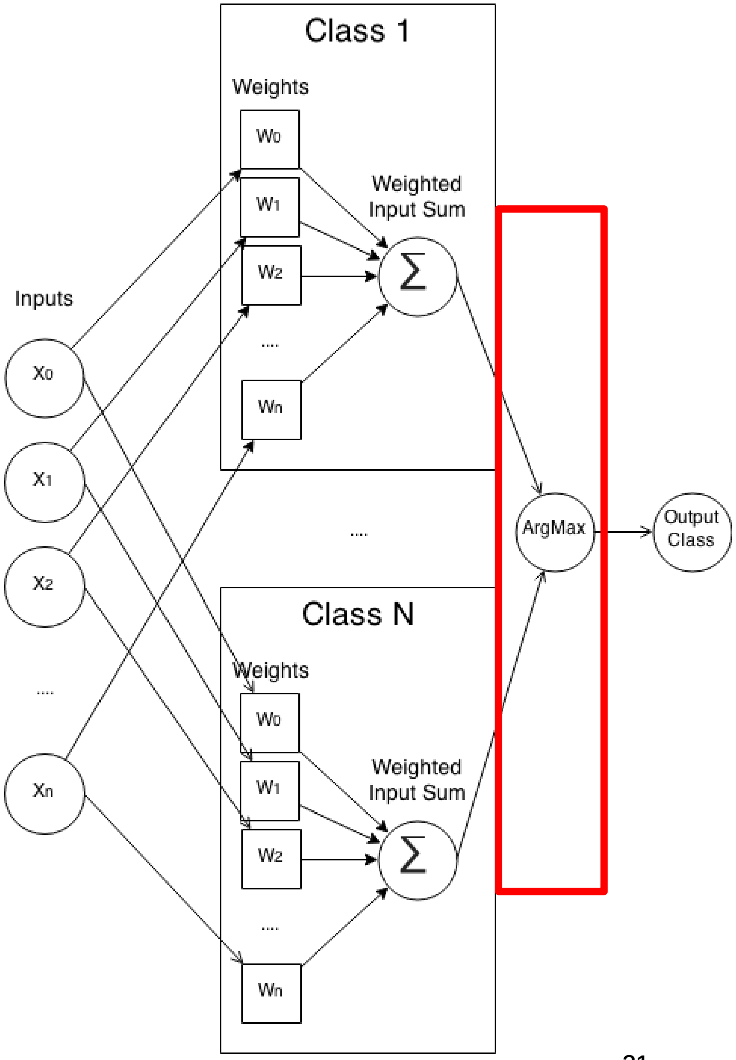
Input:

> The whole input will be trained with each class weight set

Multi Class Classification – Perceptron! (4)

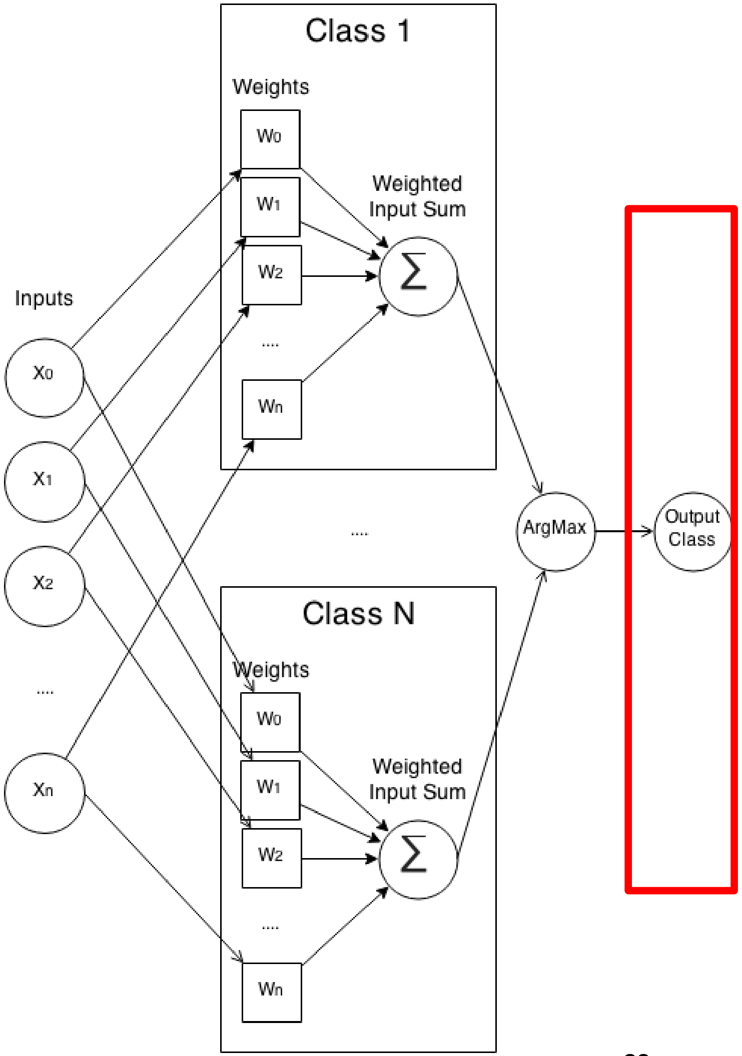
Prediction:

- Results of each class weight calculation will be compared
- The class with the highest calculation is the class of prediction (ArgMax Function)



Multi Class Classification – Perceptron! (5)

> But how do we train now?

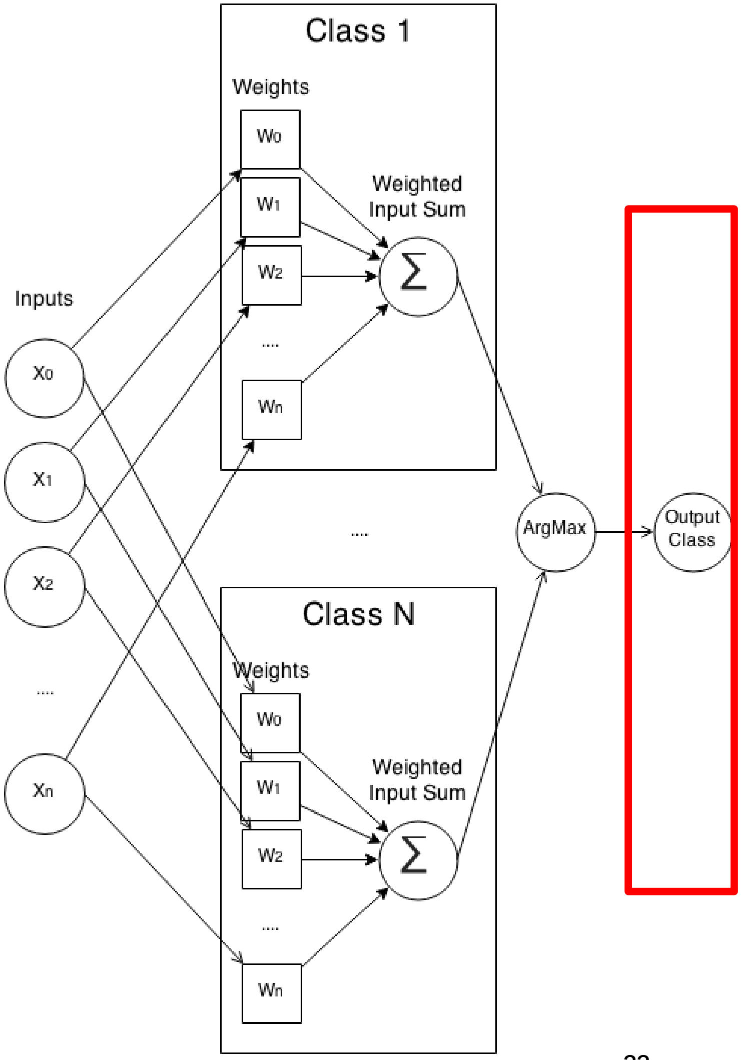


Multi Class Classification – Perceptron! (6)

Training:

> If the prediction is wrong, the weights of the predicted class have to be decreased for the particular input

> Weights of correct (but not predicted) class have to be adjusted upwards

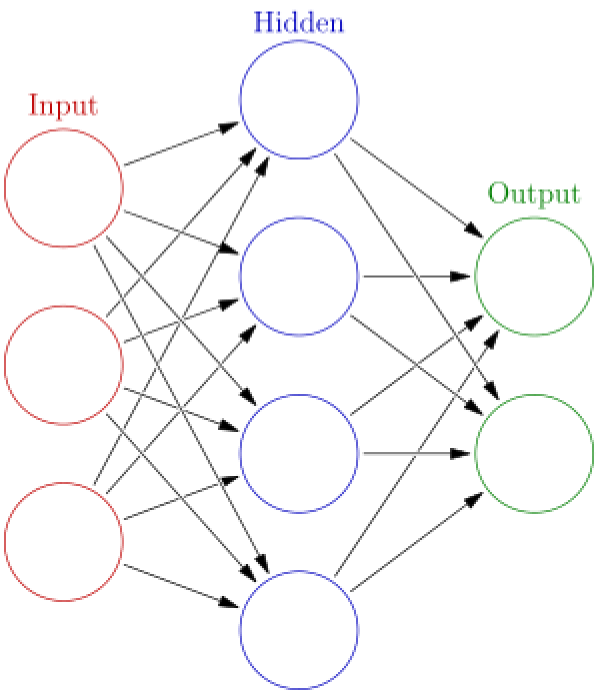


Neural Network Basics

Structure of Multi Layer Perceptrons (aka Neural Networks)

- Multi Class Perceptrons have one hidden layer where all the calculations is done

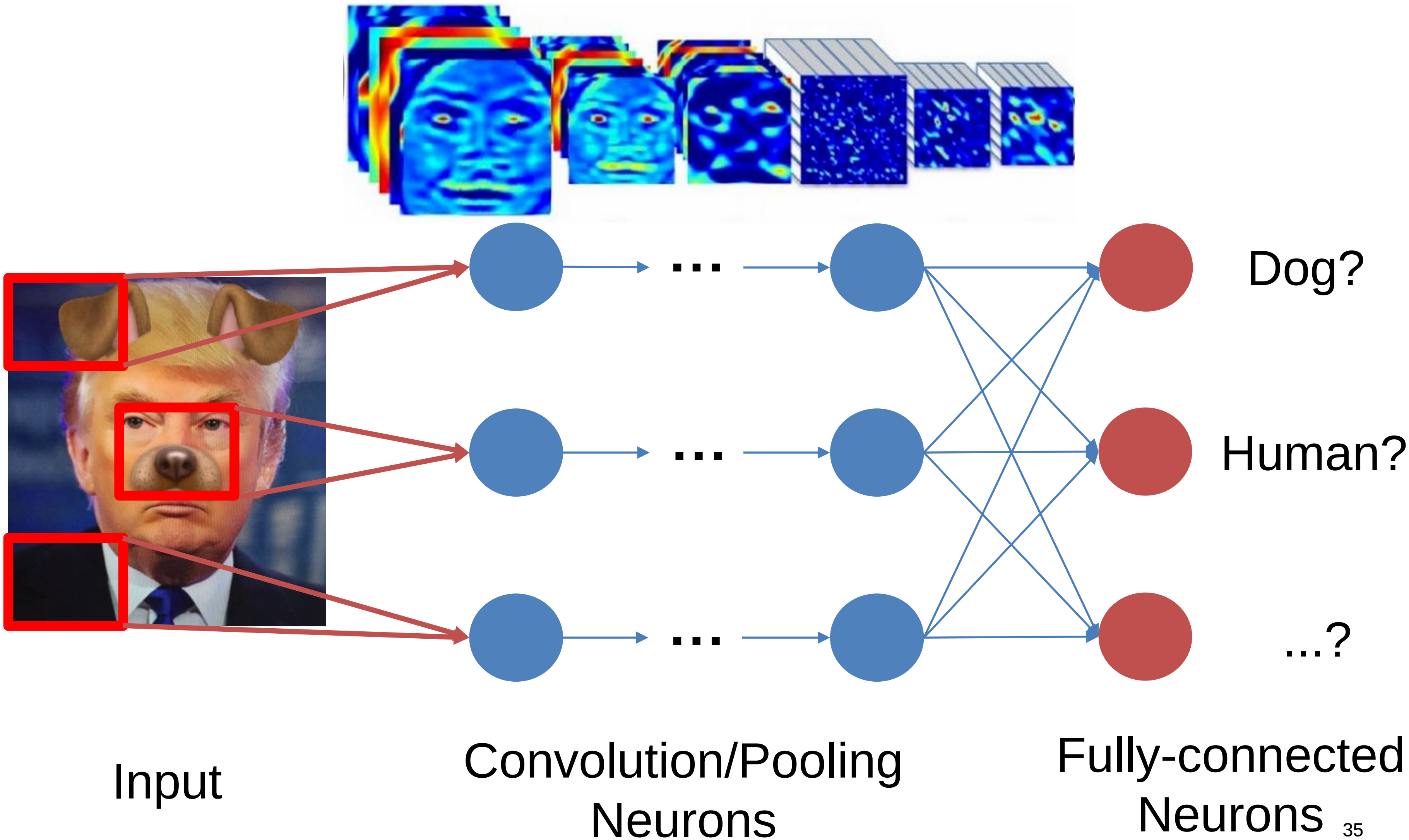
- For more complex problems neural networks with several layers have been developed (and nowadays called „deep“)



Source: https://en.wikipedia.org/wiki/Artificial_neural_network

Neural Network Basics

Example for Deep Learning – Convolutional Neural Network



Neural Network Basics

Summary and terms

- **Input:** Numerical vectors or matrices
- **Activation:** Can be a threshold or other functions to determine if Neuron fires or not („0“ or „1“)
- **Learning rate:** Determines how fast an algorithm should adjust to new data. A high learning rate adjusts better and faster to new data but also discards learned circumstances
- **Optimizer:** Function to optimize model (i.e. weights)
- **Epochs:** An epoch is a whole iteration of than one iteration of all data to have an optimal model for the given problemall input data. Algorithms need more
- **Batches:** Sometimes data is too big to put fully into RAM. Then it needs to be processed into slices (aka batches).
- **Regularization:** Goal is to use as less variables for a model as possible to prevent overfitting, e.g. by dropping out a fixed number of neurons

- Introduction
- Organisation
- Data Science
- Neural Network Basics
- Tools and Frameworks
- Classification with Neural Networks

Tools and Frameworks

Prerequisites

- Git (https://git-scm.com/downloads)
- Sourcecode (https://github.com/spinfo/LSS_DLwithText)
- Anaconda (https://www.anaconda.com/download/)
- Keras (

How do I get started with DL?

...pretty good question

How do I get started with DL?

...pretty good question

Caffe

Chainer

DL4J
Deeplearning4j

K
KERAS

Microsoft
CNTK

MatConvNet

MINERVA

mxnet

Purine

TensorFlow

theano

torch

Best way to learn?

… DIY!

Keras

Why Keras?

- Quickly train and test model

From Standard layer:

- Wraps multiple frameworks
- (Simplified) interface to Theano, CNTK or TensorFlow
- TensorFlow is default API

- Write and debug custom models and layers:

How do I build a Neural Network in Keras?

Pipeline

1. Define Network
2. Compile it
3. Fit it
4. Evaluate it
5. Make Predictions

How do I build a Neural Network in Keras?

Keras

Define Network

- Create an instance of the sequential class
- Define sequence of layers
- Add new lines (each line is a new layer)
- First Layer: number of inputs (can differ depending to the network type)

```
model = Sequential()

model.add(Dense(32, activation='relu', input_shape=(784,)))
model.add(Dense(10, activation='softmax'))

model.summary()
```

How do I build a Neural Network in Keras?

Keras

Compile it

- Transforms a simple sequence of layers into a highly efficient series of matrix transforms
- Intended to be executed on the GPU (depending on the configuration set)
- Optimization: Train the network
- Loss Function: Evaluate the network

```
# For a multi-class classification problem
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# For a binary classification problem
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# For a mean squared error regression problem
model.compile(optimizer='rmsprop',
              loss='mse')
```

How do I build a Neural Network in Keras?

Keras

Fit Network

> Adapting the weights of the training data set

> Input X → Matrix of input patterns

> Output Y → Array of matching output patterns

```
X = count_vectorizer.fit_transform(traindf['text'])
```

```
label_map = {'HillaryClinton':1,'realDonaldTrump':-1}
Y = list(traindf['handle'].apply(lambda x: label_map[x]))
```

```
print(X.shape)
print(len(Y))
```

```
(5600, 49)
5600
```

```
model.fit(X, Y, batch_size=32, epochs=10)
```

How do I build a Neural Network in Keras?

Keras

Evaluate Network and use it to make predictions

- How good does model work? Evaluate it!
  - Input X → Matrix of input patterns (test data)
  - Output Y → Array of matching output patterns (test data)
- Try out trained model with completely new data
- What will it predict?

```
: model.evaluate(
      data,
      labels,
      batch_size=32,
      verbose=1,
      sample_weight=None)

   32/1000 [..............................] - ETA: 0s

: [0.69383435416221617, 0.53900000000000003]
```

```
: model.predict(
      data,
      batch_size=32,
      verbose=1)

   32/1000 [..............................] - ETA: 0s
```

Introduction

Organisation

Data Science

Neural Network Basics

Tools and Frameworks

Classification with Neural Networks

How to transform a text into a model?

- Neural Networks need numerical vectors to compute output
- Transformation from text to model is done via **word embeddings**
- There are plenty algorithms for generating word embeddings

Word embedding

Simplest word embedding is **Bag of Words**:

> *Input*: “My dog has a dog name”, “My dog is cute”

> *Bag of Words*:

Sent1: { “My” : 1, “dog” : 2, “has” : 1, “a” : 1, “name” : 1, “is” : 0, “cute” : 0 }

Sent2: { “My” : 1, “dog” : 1, “has” : 0, “a” : 0, “name” : 0, “is” : 1, “cute” : 1 }

→

Sent1: { 1, 2, 1, 1, 1, 0, 0 }

Sent1: { 1, 1, 0, 0, 0, 1, 0 }

Word embedding

Bag of Words:

- Features: Words
- Values in Vector: Total Word Count in document
- (other features (lemma, ngrams,..) and values (weigthed,…) are possible)

Feature Vector serves as Input Vector for Neural Network

Classification with Neural Networks

And the labels?

- each output node of the Neural Network represents a label
- the higher the output value, the more probable the label is

Classification with Neural Network

Let‘s get our hands dirty!