Data Science and Machine Learning in Python

Stephan Weyers

Fachhochschule Dortmund
University of Applied Sciences and Arts

Topics covered in the online lectures

Fachhochschule Dortmund
University of Applied Sciences and Arts

Part 1: Data Science

	Date	Topics covered
1	Apr 13th	Course introduction Data Science motivation How to use Jupyter Notebook Python types and lists Loops, if/else, functions
2	Apr 20th	Python tuples, lists, dictionaries Functions Numpy basics, operations Image processing
3	Apr 27th	Pandas Series, DataFrame Pandas basic operations Import/export files
4	May 4th	Principles of data visualization Data cleaning and preparation Join, combine and reshape data
5	May 11th	Volkswohl Bund dataset Data visualization in Python How to write Data Science reports Data aggregation and grouping

Part 2: Machine Learning

	Date	Topics covered
6	Jun 1st	Introduction to supervised learning Classification and regression scikit-learn k-Nearest Neighbors Linear regression (ridge and lasso)
7	Jun 8th	Linear classification models Decision trees Random forests and gradient boosting
8	Jun 15th	Kernel support vector machines Neural networks
9	Jun 22nd	Introduction to unsupervised learning Preprocessing and scaling Dimensionality reduction Principal component analysis
10	Jun 29th	k-means clustering Hierarchical clustering DBSCAN
11	Jul 6th	Representing data Engineering features Model evaluation and improvement Text data analysis

Deadlines for Submission and Distribution of Grading

Fachhochschule Dortmund
University of Applied Sciences and Arts

Student task	Deliverables	Deadline	Work	Share of grade
W01 Assignment	Code and results	Apr 26th	Team A	5.0%
W02 Case Study	Code / presentation slides	May 22nd	Team B	18.0%
W02 Case Study	Peer review*	May 31st	Individual	2.0%
W03 Assignment	Code and results	May 29th	Team B	5.0%
W04 Assignment	Code and results	Jun 12th	Team C	10.0%
W05 Assignment	Code and results	Jun 26th	Team D	7.0%
W06 Assignment	Code and results	Jul 8th	Team D	13.0%
W07 Case Study	Code / presentation slides	Jul 17th	Team D	22.0%
W07 Case Study	Peer review*	Jul 31st	Individual	3.0%
DataCamp 1	Finish course	May 9th	Individual	2.5%
DataCamp 2	Finish course	May 30th	Individual	2.5%
DataCamp 3	Finish course	Jun 20th	Individual	2.5%
DataCamp 4	Finish course	Jul 11th	Individual	2.5%

* Peer review is mandatory. Quality of peer review itself is graded. Not providing peer review at all would result in high point deduction

Agenda for online lecture 8

Fachhochschule Dortmund
University of Applied Sciences and Arts

Session	Topic	Mode	Materials used	Minutes	End
14:30-16:00	Organizational questions	Q&A		10	14:40
	Bank Marketing data	Team work in break-out rooms	Lecture 08a notebook	45	15:25
	Kernel SVM	Lecture / Q&A	Lecture slides	15	15:40
	Neural networks	Lecture / Q&A	Lecture slides	15	15:55
16:10-17:40	Telco Customer Churn	Lecture / Q&A	Lecture 08b notebook	30	16:40
	Online retail exercise 1+2	Team work in break-out rooms	Lecture 08c notebook	40	17:20
	Online retail exercise 1	Lecture / Q&A	Lecture 08c notebook	15	17:35
17:50-19:20	Online retail exercise 2+3	Team work in break-out rooms	Lecture 08c notebook	30	18:20
	Online retail exercise 2	Lecture / Q&A	Lecture 08c notebook	10	18:30
	Online retail exercise 3	Team work in break-out rooms	Lecture 08c notebook	30	19:00
	Online retail exercise 3	Lecture / Q&A	Lecture 08c notebook	10	19:10
	Organizational questions	Q&A		10	19:20

Types of problems

Fachhochschule Dortmund
University of Applied Sciences and Arts

Supervised Approaches

- Labeled data
- Target values known

Classification

- Predict category

Regression

- Predict numeric value

Unsupervised Approaches

- Unlabeled data
- No target value provided

Cluster Analysis

- Organize similar cases into segments

Dimensionality reduction

- Reduce number of features

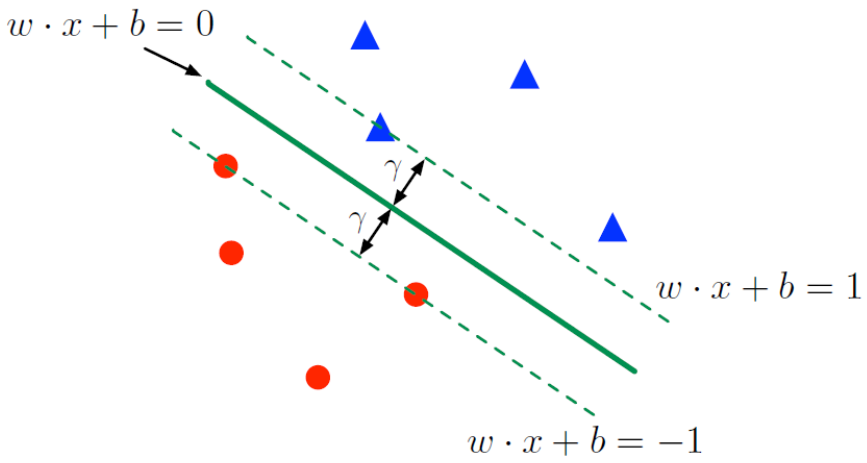
Question for discussion

- Find examples for each of the 4 categories

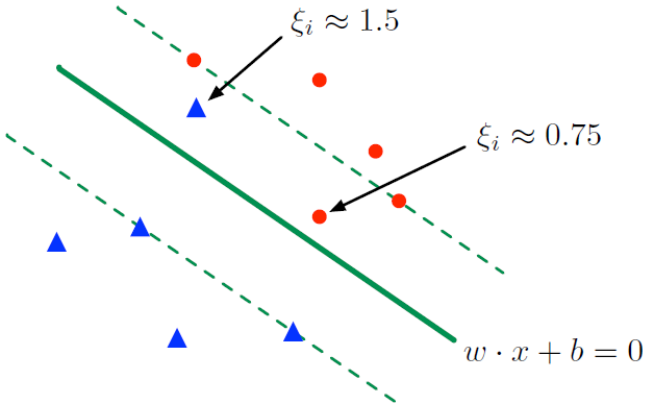
Linear Support Vector Machines

Fachhochschule Dortmund
University of Applied Sciences and Arts

Maximize margin $\gamma = \frac{1}{\|w\|}$



Minimize slack ξ

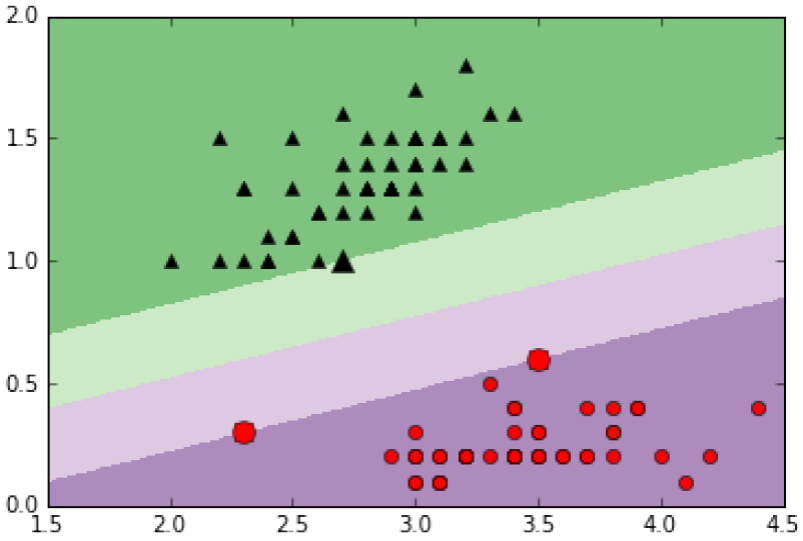


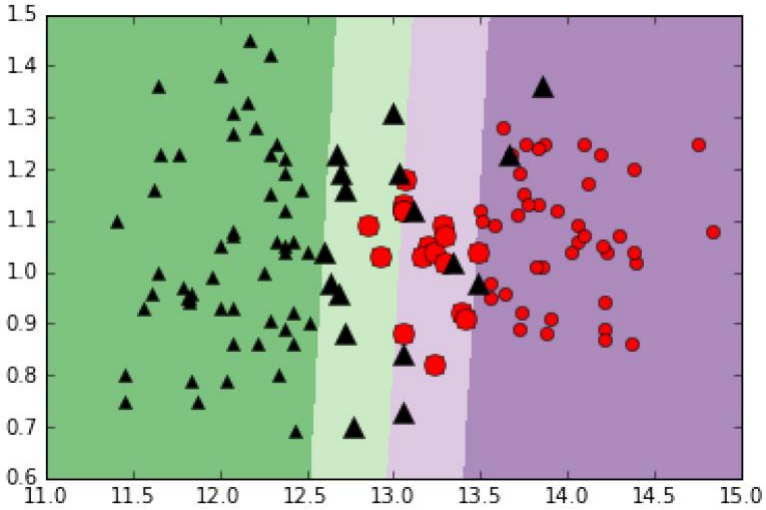
Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Support Vector Machines

Fachhochschule Dortmund
University of Applied Sciences and Arts

Support vectors





Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Binary Classification – Formulas

Fachhochschule Dortmund
University of Applied Sciences and Arts

Given training input data $\boldsymbol{x^{(1)}, \ldots, x^{(n)}} \in \mathbb{R}^{\boldsymbol{d}}$ with labels $\boldsymbol{y^{(1)}, \ldots, y^{(n)}} \in \{-\mathbf{1}, \mathbf{1}\}$.

Try to find $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}$ and $\boldsymbol{b} \in \mathbb{R}$, so that

$$\mathbf{sign}\left(\boldsymbol{w} \cdot \boldsymbol{x^{(i)}} + \boldsymbol{b}\right) = \boldsymbol{y^{(i)}}$$

as often as possible.

Logistic regression

Minimize

$$\boldsymbol{L(w, b)} = -\ln\left(\prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{Pr_{w,b}}\left(\boldsymbol{y^{(i)}} \mid \boldsymbol{x^{(i)}}\right)\right) + \boldsymbol{\lambda}\|\boldsymbol{w}\|_{\mathbf{2}}^{\mathbf{2}} = -\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \ln\left(\frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{y^{(i)}}\left(\boldsymbol{w} \cdot \boldsymbol{x^{(i)}} + \boldsymbol{b}\right)}}\right) + \boldsymbol{\lambda}\|\boldsymbol{w}\|_{\mathbf{2}}^{\mathbf{2}}$$

Linear support vector machines

Hard margin

$$\min_{\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}, \boldsymbol{b} \in \mathbb{R}} \|\boldsymbol{w}\|_{\mathbf{2}}^{\mathbf{2}} \quad \text{such that} \quad \boldsymbol{y^{(i)}}\left(\boldsymbol{w} \cdot \boldsymbol{x^{(i)}} + \boldsymbol{b}\right) \geq 1$$

Soft margin

$$\min_{\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}, \boldsymbol{b} \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^{\boldsymbol{n}}} \|\boldsymbol{w}\|_{\mathbf{2}}^{\mathbf{2}} + \boldsymbol{K}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\xi_i} \quad \text{such that} \quad \boldsymbol{y^{(i)}}\left(\boldsymbol{w} \cdot \boldsymbol{x^{(i)}} + \boldsymbol{b}\right) \geq \mathbf{1} - \boldsymbol{\xi_i}, \quad \boldsymbol{\xi} \geq \mathbf{0}$$

Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Linear Models – Summary

Parameters

- Regularization parameter alpha and C
- Model type lasso vs. ridge for regression / logistic vs. SVM for classification

Strengths

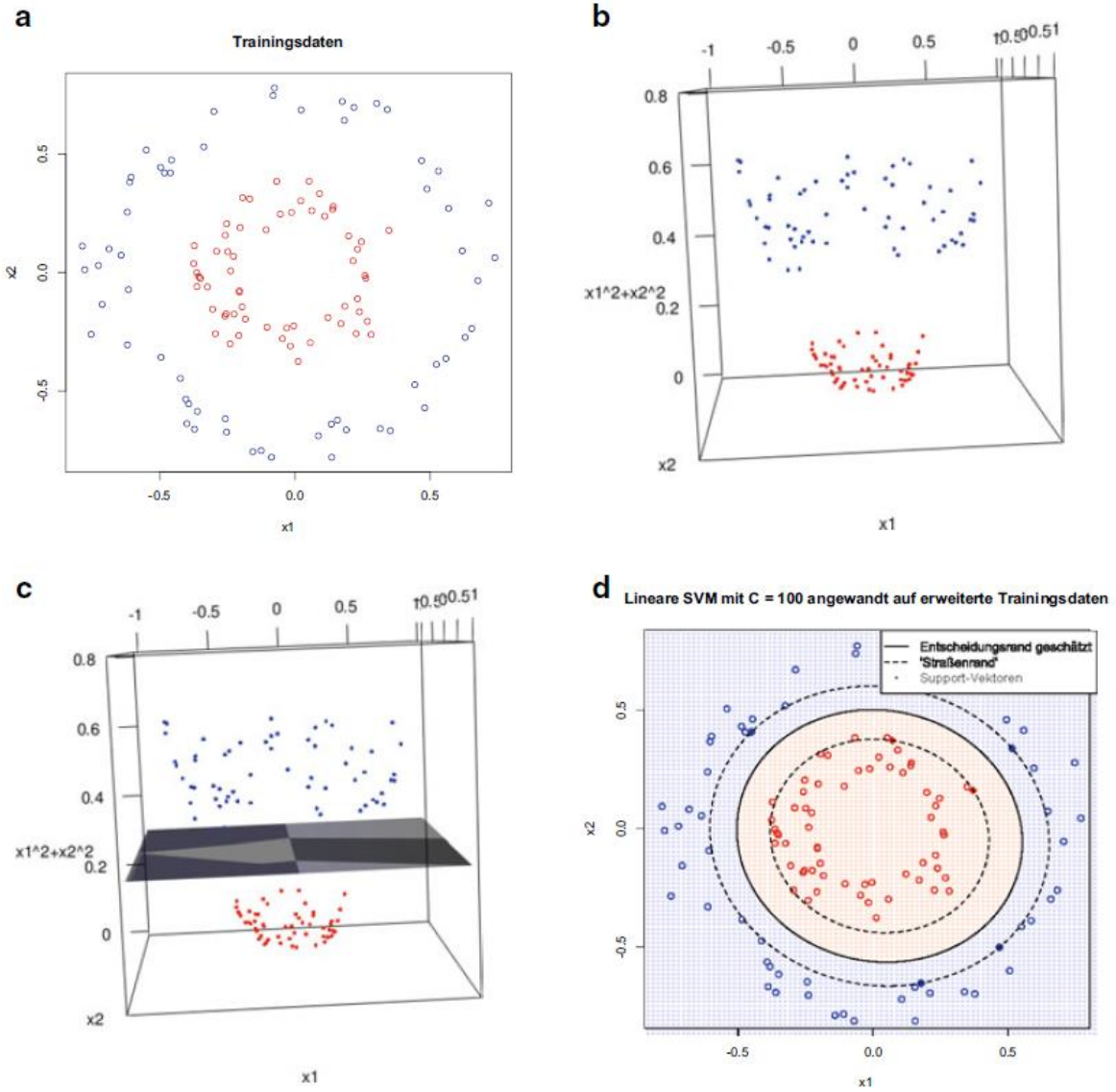
- Fast to train, fast to predict
- Work well with sparse data
- Relatively easy to understand how predictions are made
- Work well with large number of features

Weaknesses

- Coefficients hard to interpret, especially if features are highly correlated
- Sometimes fail with small datasets
- Perform bad with non-linear features and datasets that are not linearly separable

Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

SVM: Motivation for Polynomial Kernels



Source: Richter, S. (2019). Statistisches und maschinelles Lernen. Springer Spektrum.

SVM: Polynomial basis expansion

Input variables

$$x = (x_1, x_2, x_3)$$

Extended input variables (degree p=2)

$$\phi(x) = \left(x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3\right)$$

Extended input variables (degree p=3)

$$\phi(x) = \left(x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, x_1^3, x_2^3, x_3^3, x_1^2x_2, x_1^2x_3, x_2^2x_1, x_2^2x_3, x_3^2x_1, x_3^2x_2, x_1x_2x_3\right)$$

Kernel trick

$$\phi(x) \cdot \phi(y) = (1 + x \cdot y)^p$$

SVM: Gauß kernel

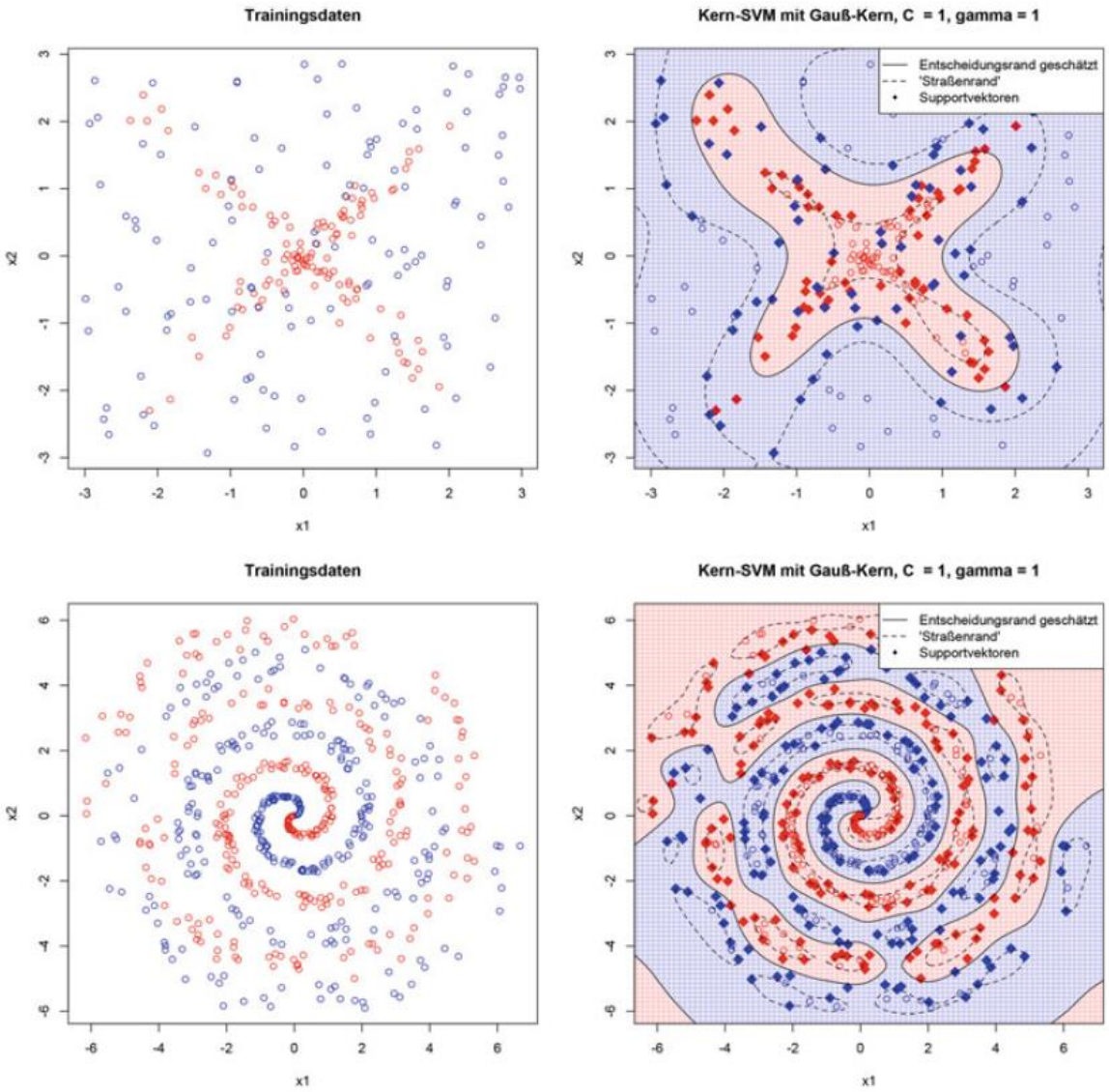
Polynomial kernel

$$\boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{y}) = (\mathbf{1} + \boldsymbol{x} \cdot \boldsymbol{y})^{\boldsymbol{p}}$$

Gaussian (RBF) kernel

$$\boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{e}^{-\boldsymbol{\gamma}\|\boldsymbol{x}-\boldsymbol{y}\|^2}$$

SVM: Gauß Kernel Graphics



Source: Richter, S. (2019). Statistisches und maschinelles Lernen. Springer Spektrum.

Kernelized Support Vector Machines – Summary

Fachhochschule Dortmund
University of Applied Sciences and Arts

Parameters

- Regularization parameter C
- Kernel “poly” or “rbf”
- Kernel specific parameters “degree” or “gamma”

Strengths

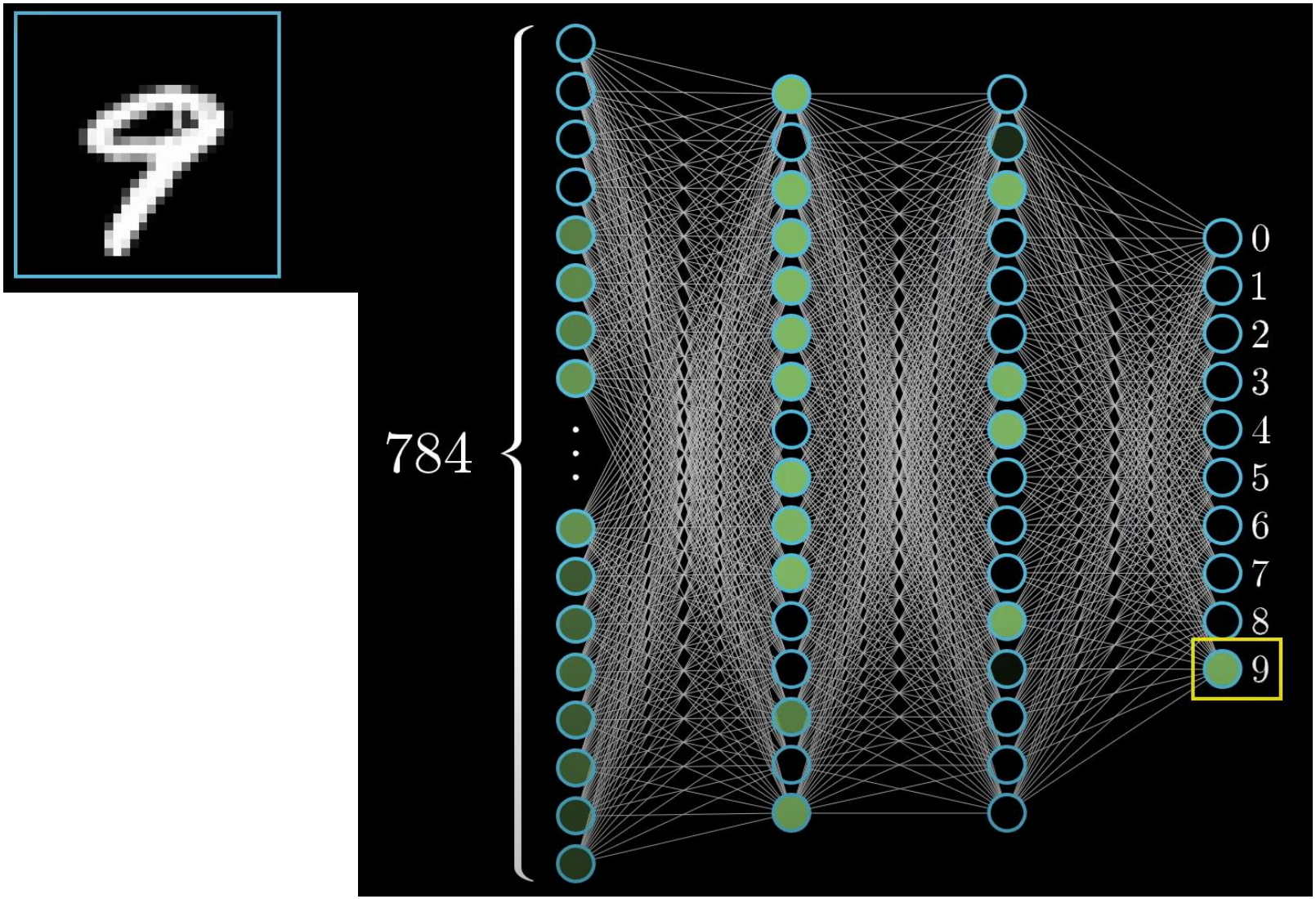
- Complex decision boundaries
- Work well with few and also many features
- Work well, if all features are on same scale / in same units

Weaknesses

- Runtime and memory usage challenges with high number of samples
- Require preprocessing / scaling of data to get good results
- Results hard to interpret, difficult to explain to non-experts

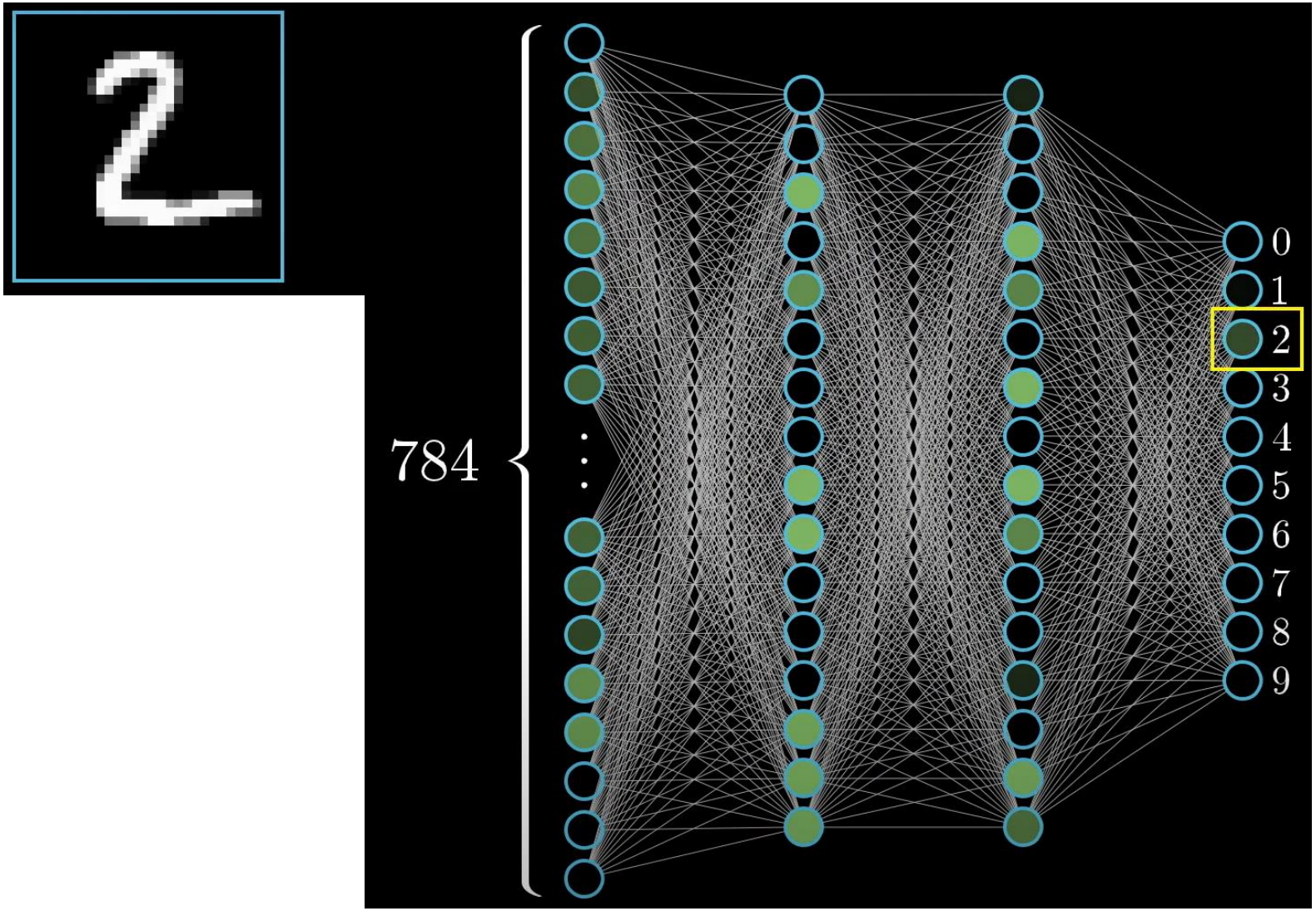
Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media

Neural Networks



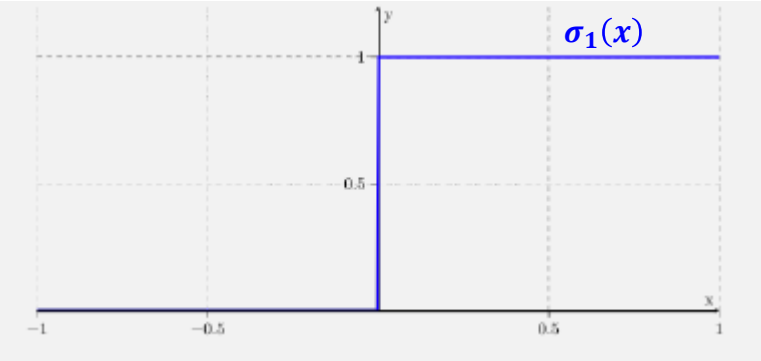
Source: https://ki-campus.org/node/209 / https://www.youtube.com/watch?v=aircAruvnKk

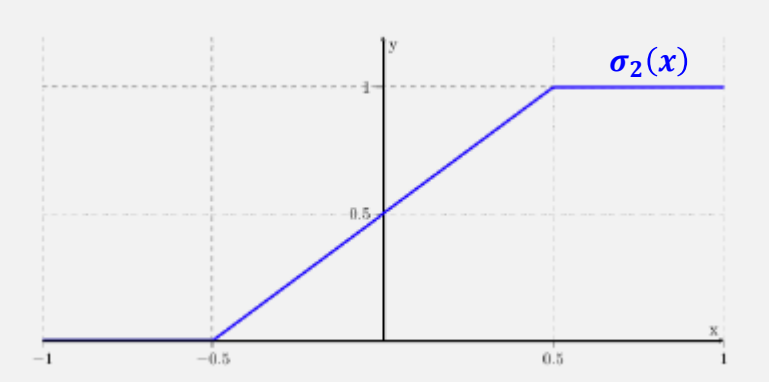
Neural Networks

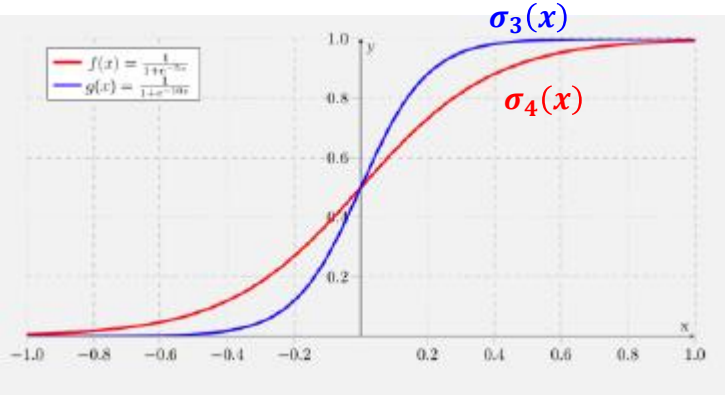


Source: https://ki-campus.org/node/209 / https://www.youtube.com/watch?v=aircAruvnKk

Neural Networks – Activation Functions

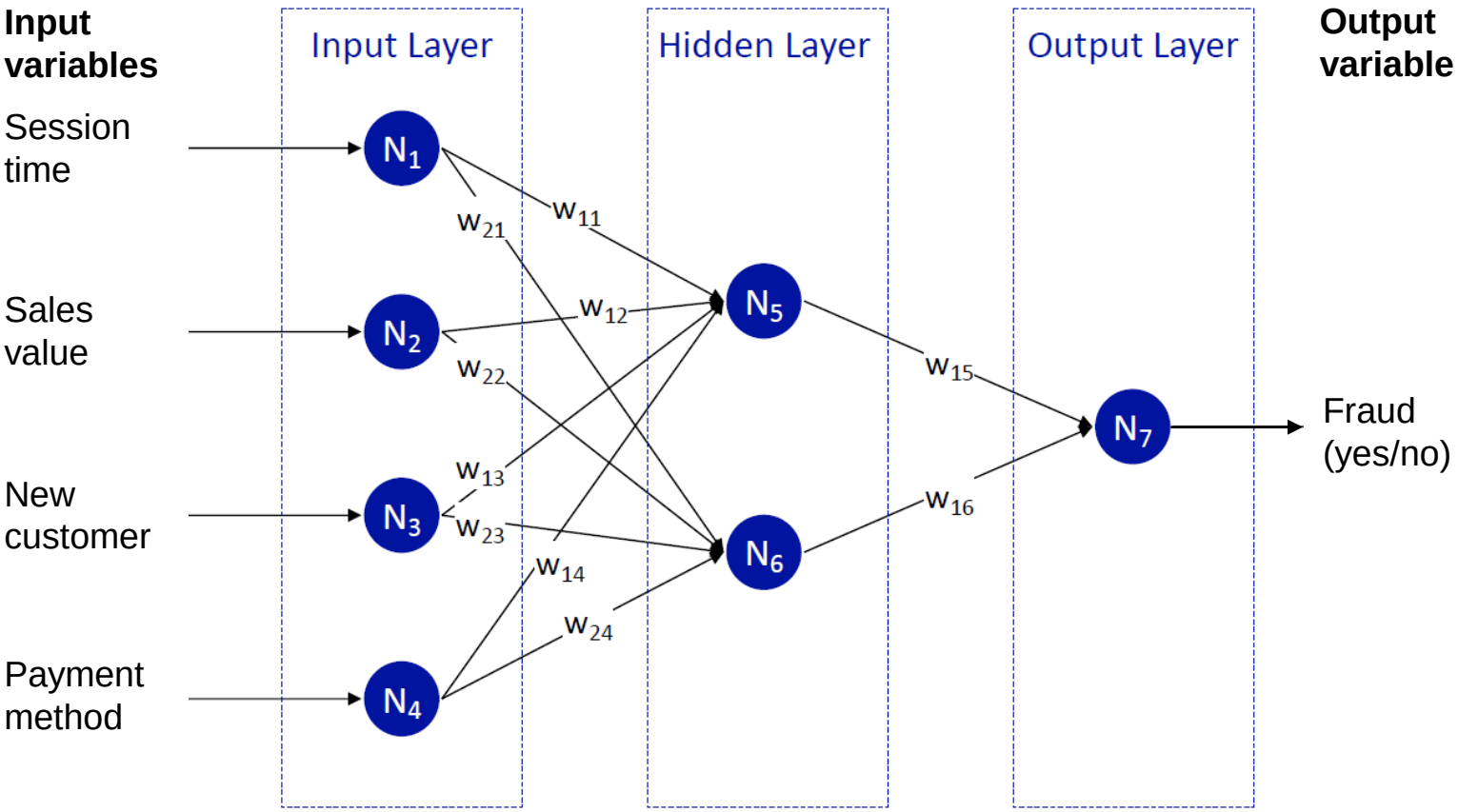






Source: Prof. Dr. Michael Bücker – Data Mining

Neural Networks – Online Retail Fraud Detection



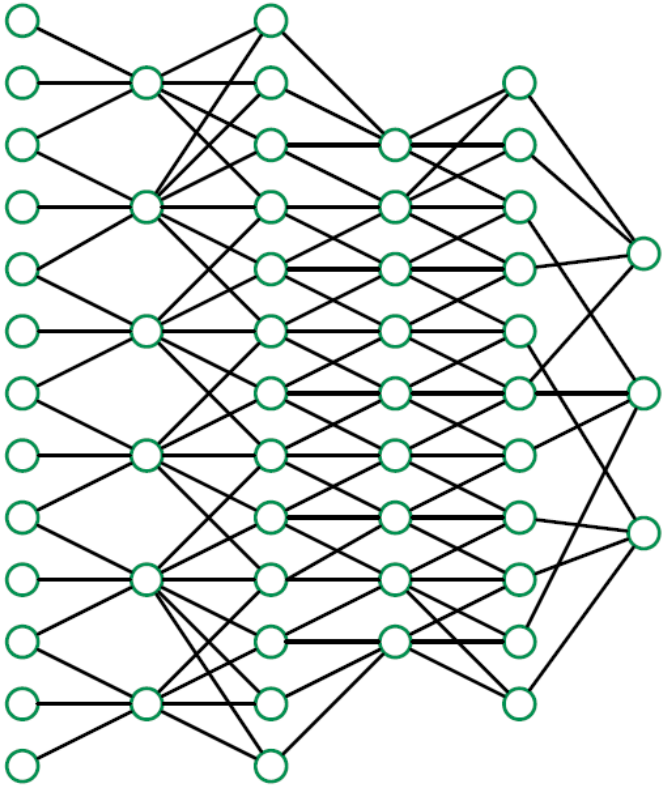
$$N_5 = \sigma(w_{11}N_1 + w_{12}N_2 + w_{13}N_3 + w_{14}N_4 + b_5)$$

$$N_6 = \sigma(w_{21}N_1 + w_{22}N_2 + w_{23}N_3 + w_{24}N_4 + b_6)$$

$$N_7 = \sigma(w_{15}N_5 + w_{16}N_6 + b_7)$$

Source: Prof. Dr. Michael Bücker – Data Mining

Neural Networks – Complexity



Question for discussion

A neural network has 6 layers, each consisting of 100 nodes, and each fully connected to the previous layer. Roughly how many parameters ($\boldsymbol{w_{ij}}$) does this network have?

Source: UCSanDiegoX DSE220x – Machine Learning Fundamentals

Neural networks – Summary

Fachhochschule Dortmund
University of Applied Sciences and Arts

Parameters

- Activation function, number and size of layers
- Many more

Strengths

- State-of-the-art in many applications
- Able to build very complex models

Weaknesses

- Black box – difficult to interpret and explain results
- Careful preprocessing and scaling of data needed
- Very long computation time for large datasets
- Difficulties with mixed kinds of features on different scales

Source: Müller, A. C., Guido, S. (2016) Introduction to Machine Learning with Python : A Guide for Data Scientists, O'Reilly Media