Machine Learning Techniques for Text

Module 4: Extracting Sentiments from Product Reviews

Dr. Nikos Tsourakis

Course outline

- Module 0: Python Crash Course
- Module 1: Intro to Machine Learning
- Module 2: Detecting Spam Emails
- Module 3: Classifying Topics of Newsgroup Posts
- **Module 4: Extracting Sentiments from Product Reviews**
- Module 5: Recommending Music Titles
- Module 6: Teaching Machines to Translate
- Module 7: Summarizing Wikipedia Articles
- Module 8: Detecting Hateful and Offensive Language
- Module 9: Generating Text in Chatbots
- Module 10: Clustering Speech-to-Text Transcriptions

Overview

- Deciphering the emotional tone behind a sequence of words finds extensive utility in analyzing survey responses, customer feedback, or product reviews
- The advent of social networks offered new possibilities for people to instantly express their opinions on various issues
- We focus on another typical problem in ***natural language processing*** (NLP): the extraction of sentiment from a piece of text using an open-source dataset with reviews from the Amazon
  - EDA is again the first task in the pipeline, which helps us discuss important findings on the input data
  - We create different visualizations and enhance our plot construction skills with Python
  - Next, with have a deeper look at how the model's parameters are estimated
  - Then, we introduce a state-of-the-art architecture that is nature-inspired
  - Finally, we contrast two classifiers for the same task while discussing different implications

Module objectives

After completing this module, you should be able to:

- Creating models for predicting continuous values
- Acquiring a better understanding of how algorithms learn from data
- Examining optimizations techniques
- Learning how to avoid overfitting
- Introducing state-of-the-art machine learning architectures
- Creating different classification models

Machine Learning Techniques for Text

Section 1: Understanding sentiment analysis

Sentiment analysis

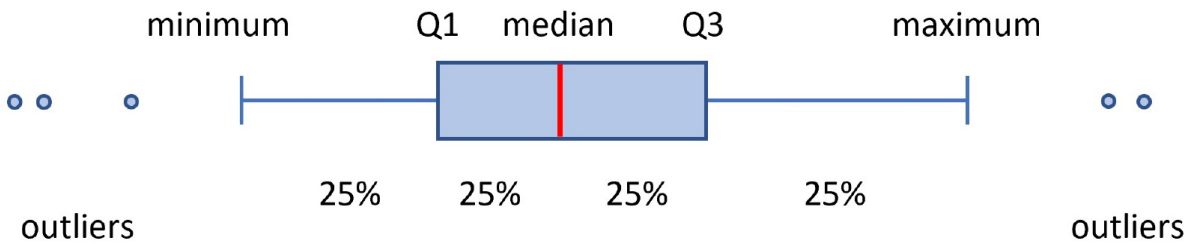
- You are running for public office, and to increase the chances of being elected, you must perform a substantial effort to persuade the voters
  - A possible strategy is to focus on less favorable regions to your candidacy, which can be identified from the sentiment expressed in social media posts in this area
- Similarly, suppose you are the CEO of a company that recently deployed a new product
  - This time, you are interested in knowing how your customers perceive it and in understanding their opinions
- All these issues can be addressed by performing ***sentiment analysis***: assigning a sentiment label to a piece of text

Machine Learning Techniques for Text

Section 2: Performing exploratory data analysis

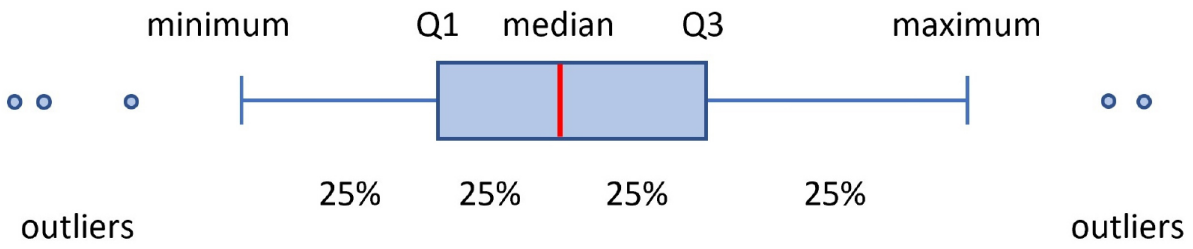
Boxplots

- Creating a ***boxplot***—also known as a box and whisker plot—is an elegant way to present condensed information about the data
- It provides a visual five-number summary of the underlying data and is frequently encountered in EDA
- For example, we can check whether the product scores are symmetric (roughly the same on each median side)
- ***Q1*** is the median value of the first half of the dataset, whereas ***Q3*** is the median value of the second half



Boxplots

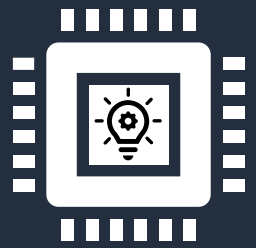
- ***Outliers*** are data points significantly different from the other samples and may indicate some sort of abnormality
- For example, an age field with a negative value is a sign of bad data and can distort the analysis
- On the other hand, outliers can help detect anomalies in the data and find patterns that do not conform to the expected behavior
  - Examples are the detection of fraud, faults in safety-critical systems, or intrusion



Boxplots

- Only the ***Logos Bible*** and ***TOPO! Nation*** products have symmetric scores
- Additionally, there are outliers for the ***Encyclopedia*** and ***Pajama Sam 2*** cases

Let’s practice!



Tasks

- Exploratory data analysis

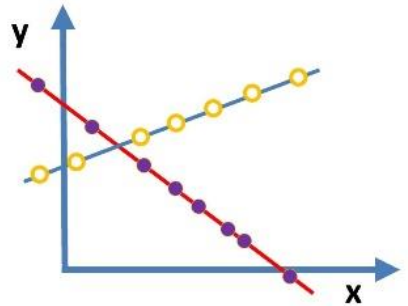
https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-04/sentiment-analysis.ipynb

Machine Learning Techniques for Text

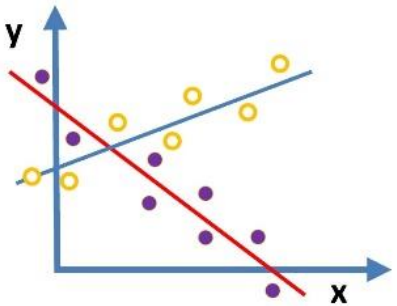
Section 3: Introducing linear & logistic regression

Relationship between variables

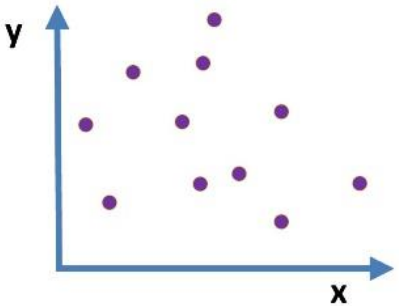
- Let's consider three plots that show the relationship between two variables: ***x*** and ***y***
  A. The points of both datasets reside on their line, which defines a clear deterministic relationship between the two variables. As ***x*** changes its value, we can precisely calculate the value of ***y*** using one of the line equations
  B. We cannot predict the exact value of ***y***, but we can obtain a good approximation based again on the line equations
  C. The relationship is random, and we cannot find any function to infer ***y*** based on the values of ***x***



(A)



(B)



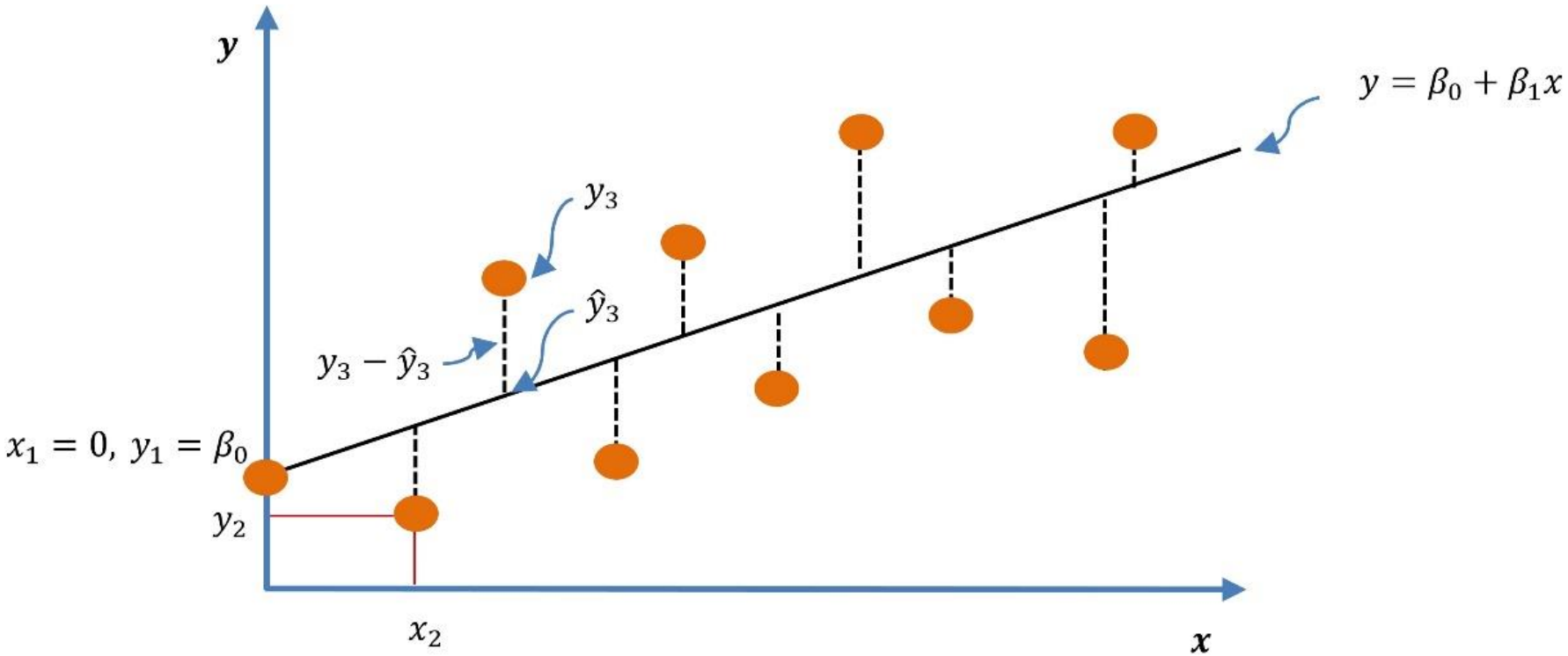
(C)

Linear regression

- One of the most well-known algorithms to elicit the best relationship between an independent variable ***x*** and a dependent variable ***y*** is ***linear regression***
- In the case of a single independent variable, the method is referred to as ***simple linear regression***, and for multiple ones, it is called ***multiple linear regression***
- The core idea is to obtain a ***regression line*** that best fits the data, exhibiting the lowest prediction error for all data points

Linear regression

- The most popular method for estimating the line of best fit is called ***ordinary least squares*** (OLS)



Linear regression

- 10 observations for which we want to find the line of best fit
- The independent variable represents the ***GDP per capita*** value for each country, and the dependent variable is the corresponding ***Happiness score*** value

GDP per capita	Happiness score	Option A $y = 1.6715 + 3.8141x$			Option B $y = 1.7638 + 3.7829x$		
x_i	y_i	$\hat{y}_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$	$\hat{y}_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$
1.34	7.769	6.782394	0.986606	0.973391	6.832886	0.936114	0.876309
1.376	7.246	6.919702	0.326298	0.106471	6.96907	0.27693	0.07669
1.269	6.852	6.511593	0.340407	0.115877	6.5643	0.2877	0.082771
1.286	6.354	6.576433	-0.22243	0.049476	6.628609	-0.27461	0.07541
1.206	6.182	6.271305	-0.0893	0.007975	6.325977	-0.14398	0.020729
0.912	6.028	5.149959	0.878041	0.770956	5.213805	0.814195	0.662914
1.173	5.809	6.145439	-0.33644	0.113191	6.201142	-0.39214	0.153775
1.004	5.603	5.500856	0.102144	0.010433	5.561832	0.041168	0.001695
1.221	5.339	6.328516	-0.98952	0.979142	6.382721	-1.04372	1.089353
1.043	5.208	5.649606	-0.44161	0.195016	5.709365	-0.50136	0.251367
			0.554197	**3.321929**		**0.000293**	**3.291014**

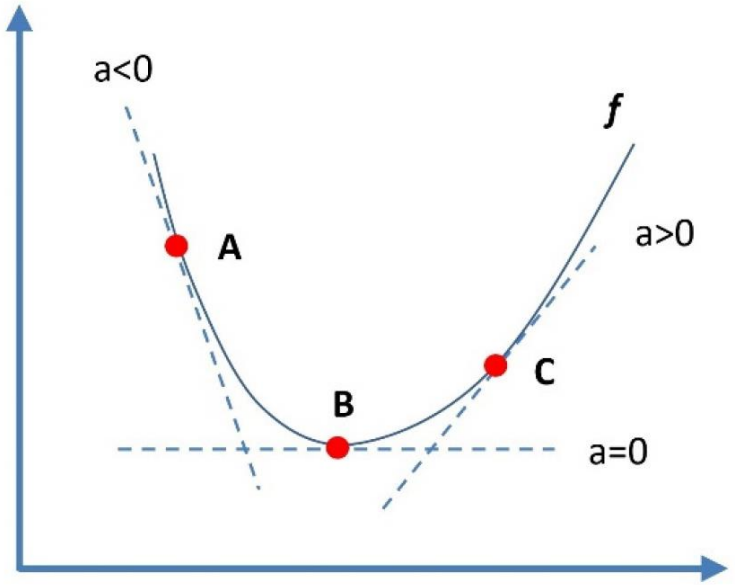
Linear regression

- 10 observations for which we want to find the line of best fit
- The independent variable represents the ***GDP per capita*** value for each country, and the dependent variable is the corresponding ***Happiness score*** value

GDP per capita	Happiness score	Option A $y = 1.6715 + 3.8141x$			Option B $y = 1.7638 + 3.7829x$		
x_i	y_i	$\hat{y}_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$	$\hat{y}_i$	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$
1.34	7.769	6.782394	0.986606	0.973391	6.832886	0.936114	0.876309
1.376	7.246	6.919702	0.326298	0.106471	6.96907	0.27693	0.07669
1.269	6.852	6.511593	0.340407	0.115877	6.5643	0.2877	0.082771
1.286	6.354	6.576433	-0.22243	0.049476	6.628609	-0.27461	0.07541
1.206	6.182	6.271305	-0.0893	0.007975	6.325977	-0.14398	0.020729
0.912	6.028	5.149959	0.878041	0.770956	5.213805	0.814195	0.662914
1.173	5.809	6.145439	-0.33644	0.113191	6.201142	-0.39214	0.153775
1.004	5.603	5.500856	0.102144	0.010433	5.561832	0.041168	0.001695
1.221	5.339	6.328516	-0.98952	0.979142	6.382721	-1.04372	1.089353
1.043	5.208	5.649606	-0.44161	0.195016	5.709365	-0.50136	0.251367
			0.554197	**3.321929**		**0.000293**	**3.291014**

Linear regression

- In ***optimization problems*** we seek the best solution from all feasible solutions
- A common way to attack this situation is to use calculus to obtain the coefficients that minimize the value of a loss function*

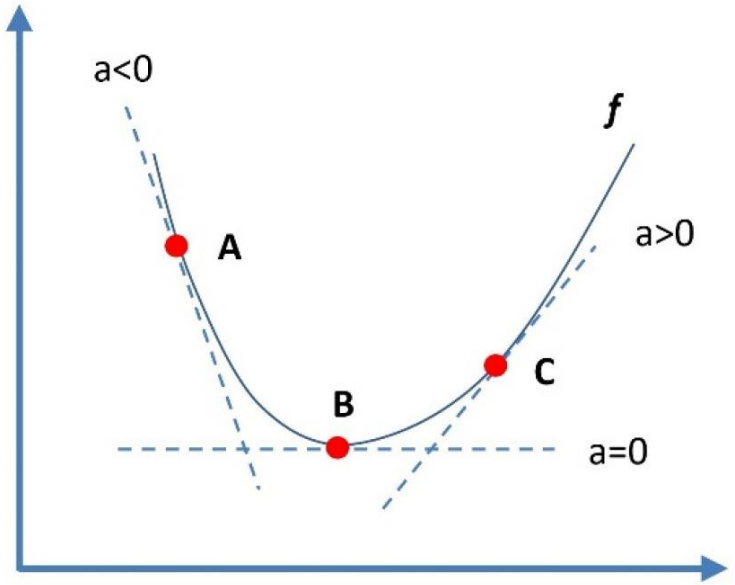


a=slope of the line

*In our case the loss functions that increases when the regression line doesn't fit the data well

Linear regression

- In ***optimization problems*** we seek the best solution from all feasible solutions
- A common way to attack this situation is to use calculus to obtain the coefficients that minimize the value of a loss function*



a=slope of the line

$$\frac{\partial S}{\partial \beta_1} = \frac{\partial \sum_i^n (y_i - \beta_0 - \beta_1 x_i)^2}{\partial \beta_1}$$

$$\frac{\partial S}{\partial \beta_0} = \frac{\partial \sum_i^n (y_i - \beta_0 - \beta_1 x_i)^2}{\partial \beta_0}$$

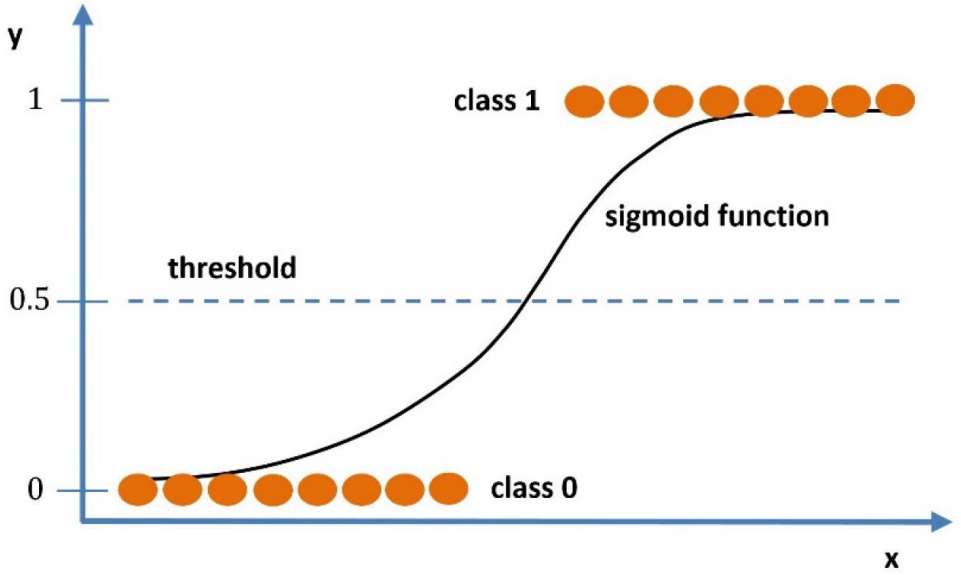
partial derivatives

*In our case the loss functions that increases when the regression line doesn't fit the data well

Logistic regression

- ***Logistic regression*** is a supervised learning algorithm which is suitable for binary classification problems
- It is a parametric learning algorithm that outputs a probability that an input belongs to a particular class
- Instead of fitting a straight line to the data, the effort is to fit an ***S-shaped*** curve called the ***sigmoid function***:
  - squeezes any real number in the interval (0, 1) that is essentially a range for probabilities
  - Given by:

$$S(x) = \frac{1}{1 + e^{-x}}$$

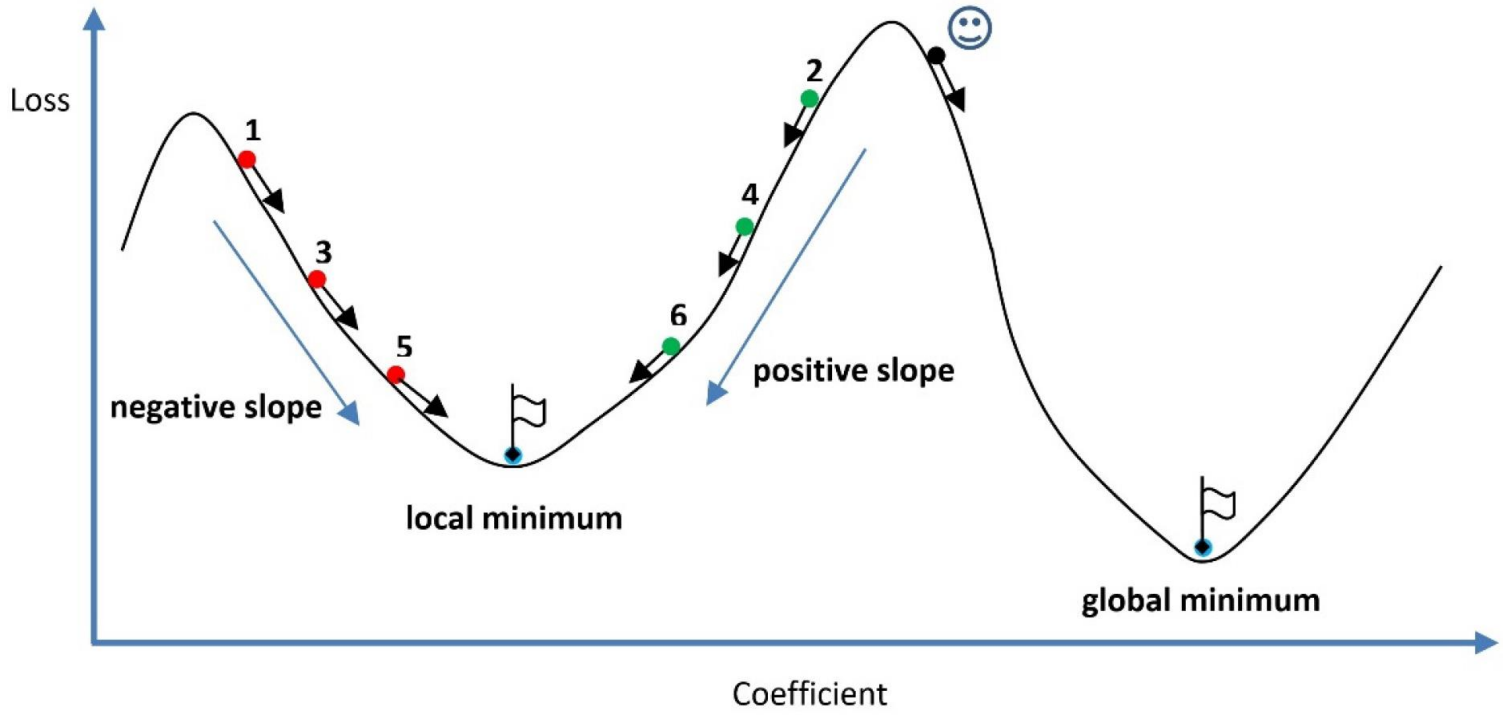


Logistic regression

- As in the case of linear regression, the aim is to estimate the coefficients that reduce the difference between the observed and the predicted value
- This difference can be expressed with a loss function such as the ***OLS*** we have seen before
- However, this method is not suitable for logistic regression
- This is because the form of the ***OLS*** loss function for logistic regression typically contains many local minima
- Therefore, minimizing the loss based on zeroing the partial derivatives often fails to find the optimal solution

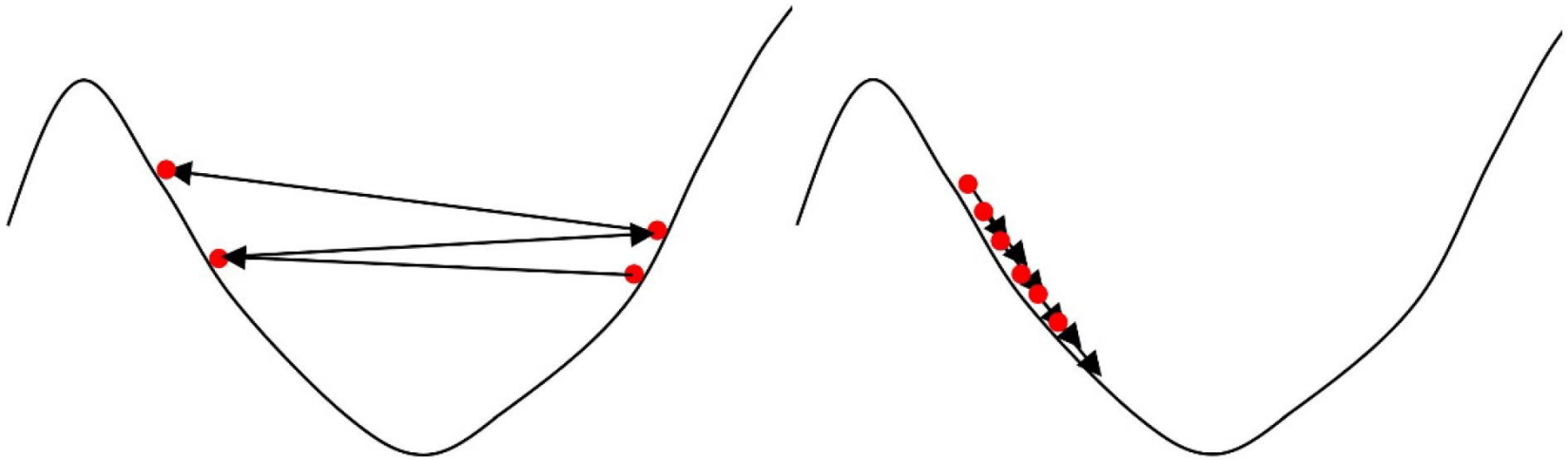
Gradient descent

- Another technique to extract the minimum of a function is called ***gradient descent***
- The basic idea behind the algorithm is the iterative update of the coefficients to be estimated until we reach convergence



Gradient descent

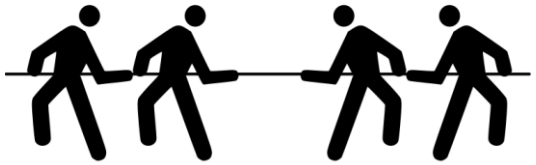
- Another tricky situation is determining how small or big steps we take in every iteration
- Too large steps may inhibit the gradient descent algorithm from reaching the minimum and bounce between the two sides of the curve
- Small steps might take too long for the algorithm to converge
- The ***learning rate*** (hyperparameter) dictates the size of steps



Regularization

- ***Occam's razor*** concept: between two competing explanations (models), the simplest one should be preferred

__Complex__ models tend to __overfit__ and do not generalize well



Too __simple__ models may result in solutions that __underfit__ the data

- There are different strategies for addressing both situations
  - We can add more features or try alternative ML algorithms for underfitting
  - For overfitting we can add more data as the population of the training observations might not representative enough
  - We can also attack overfitting using ***regularization***, which penalizes the complexity of a model

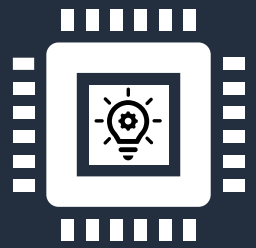
Regularization

- Minimization of the ***loss function*** has been the primary way of eliciting the best model
- Using ***regularization***, however, we also need to minimize a second factor: the ***complexity*** of the model
- Thus, the minimization task becomes twofold:

minimize(Loss + model_complexity)

- How can we quantify complexity in this case?
  - A common approach is to penalize models with high-weight values
  - The assumption is that smaller values result in simpler models
  - The model overfits as the weights grow in size to handle the specifics of the training observations

Let’s practice!



Tasks

- Exploratory data analysis
- Linear & logistic regression

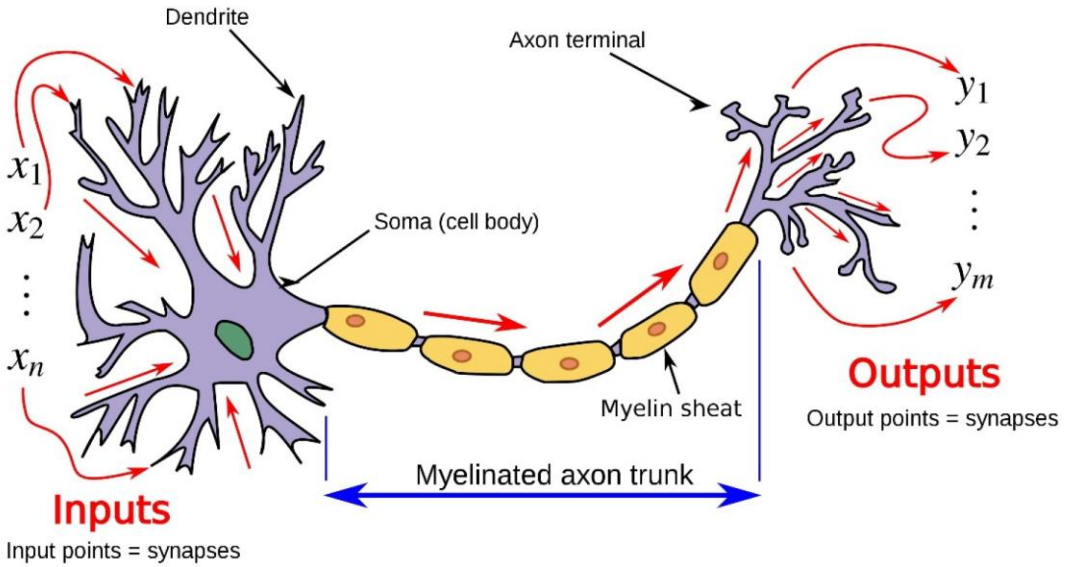
https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-04/sentiment-analysis.ipynb

Machine Learning Techniques for Text

Section 4: Introducing deep neural networks

Biological vs artificial neurons

What is more natural than to think that emulating the human brain and its functionalities can enhance artificial cognition?

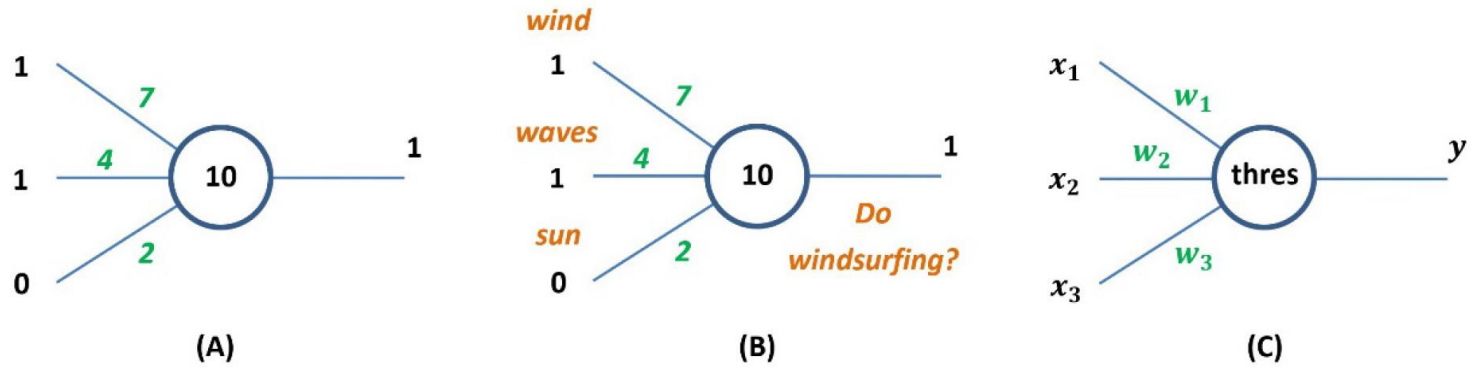


Source: https://en.wikipedia.org/wiki/Biological_neuron_model#/media/File:Neuron3.png

- ***Dendrites*** act as inputs to the neuron, which come from other interconnected neurons
- Then, they transfer each input (with a specific weight) to the ***soma***, which works as a summation function
- The ***axon*** receives the result, and once it reaches a specific electrical potential, it emits a signal pulse
- Finally, the pulse is transferred to the ***terminals***

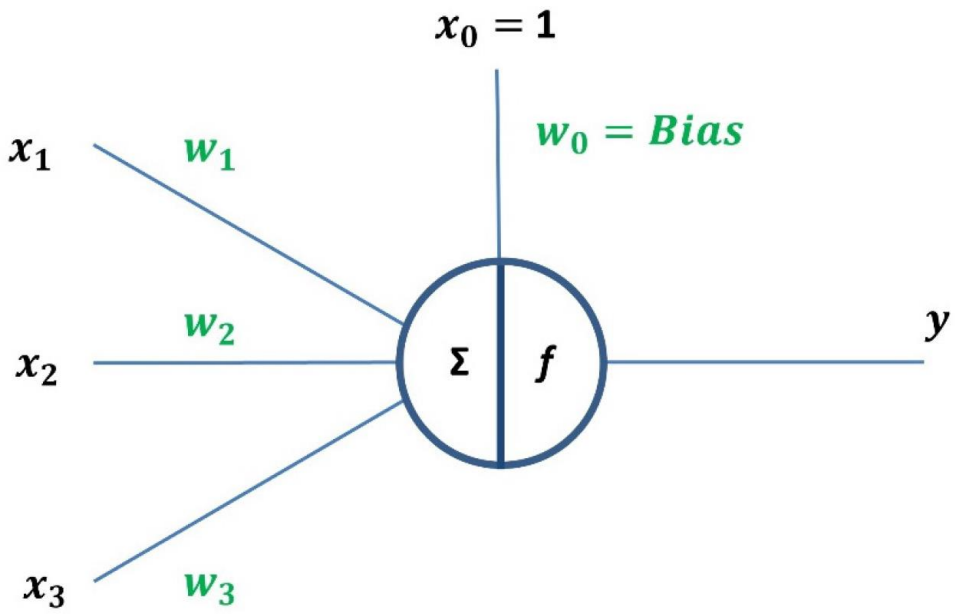
Biological vs artificial neurons

- The ***perceptron*** is a fundamental element that has been available since the 1950s but started to get significant hype in recent years
- A perceptron takes several binary inputs and emits a single binary output
- Each input is multiplied by a weight coefficient and then added all together
- The result is examined against a certain threshold, determining whether the preceptor emits ***0*** or ***1***



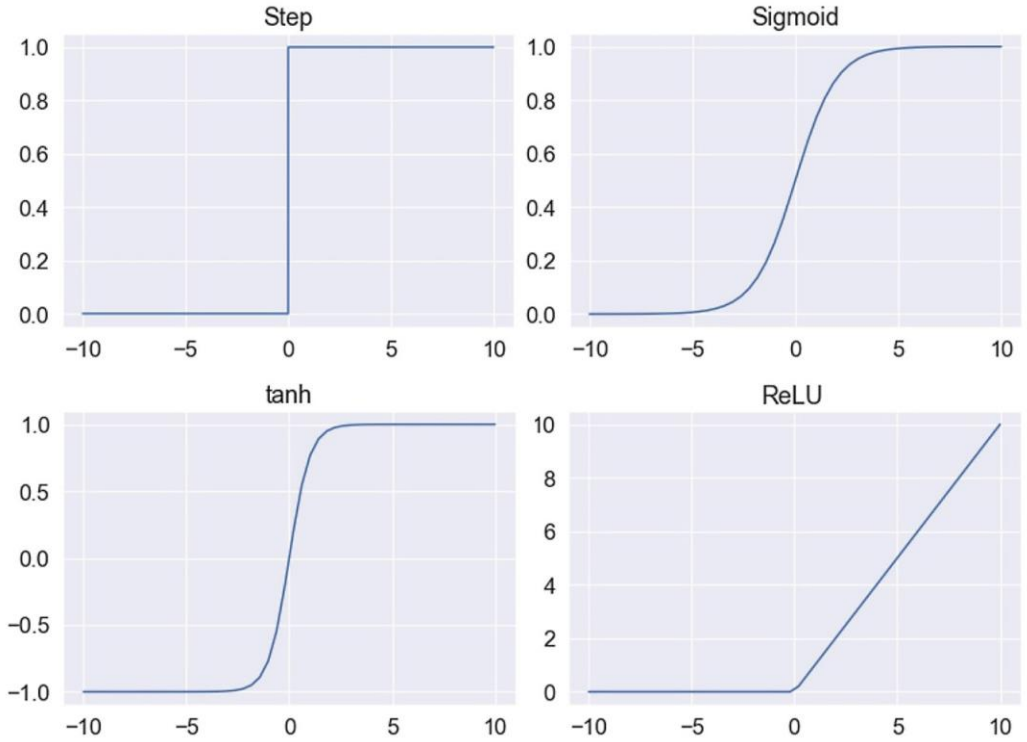
Artificial neurons

- A limitation of a perceptron is that it receives and offers only binary values
- In many problems, however, we would like to make predictions from inputs of continuous value, so, perceptrons experience limited practical utility
- Another more suitable fundamental block is called ***artificial neuron***

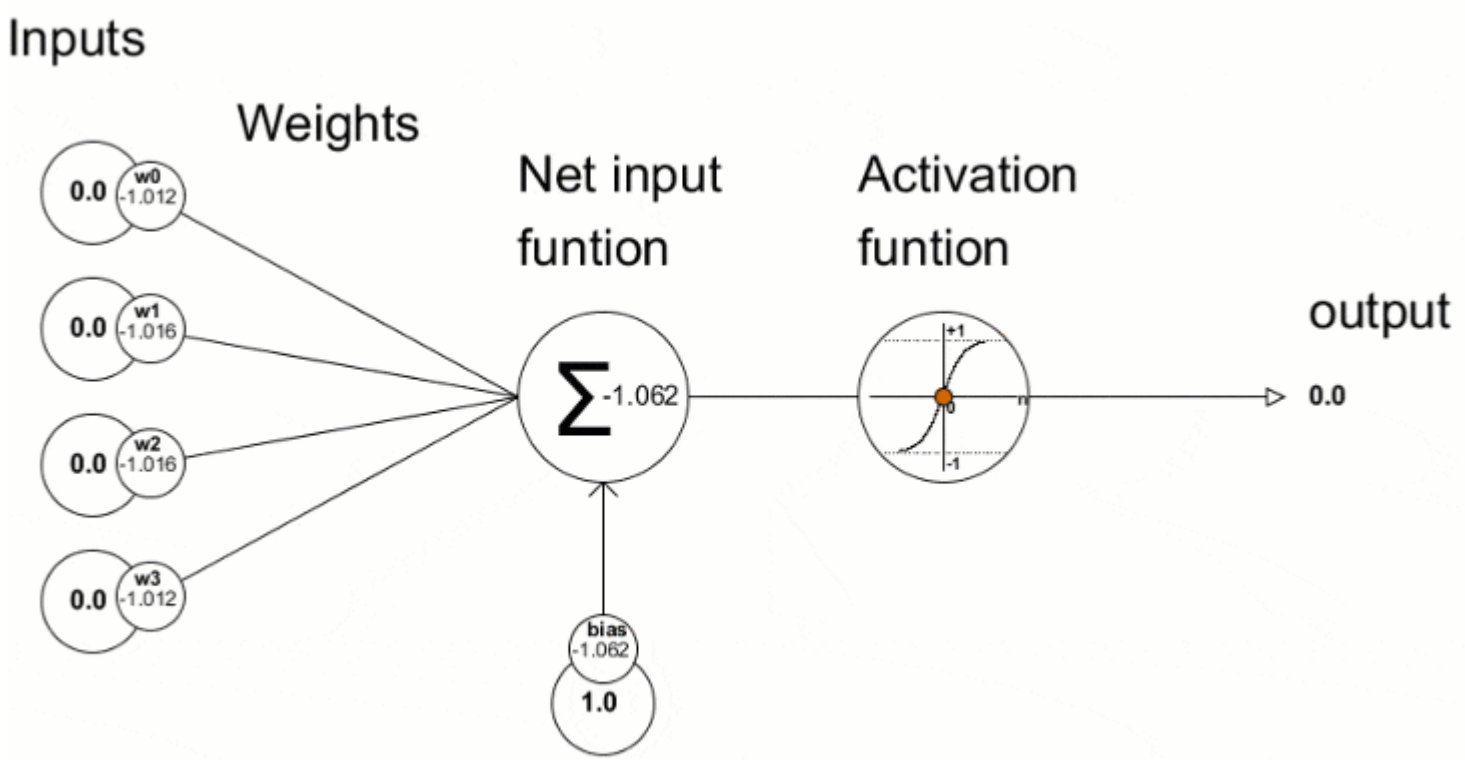


Activation functions

- Artificial neurons can be activated at different levels, making them more similar to their biological counterparts
- For this reason, f is known as the neuron's activation function

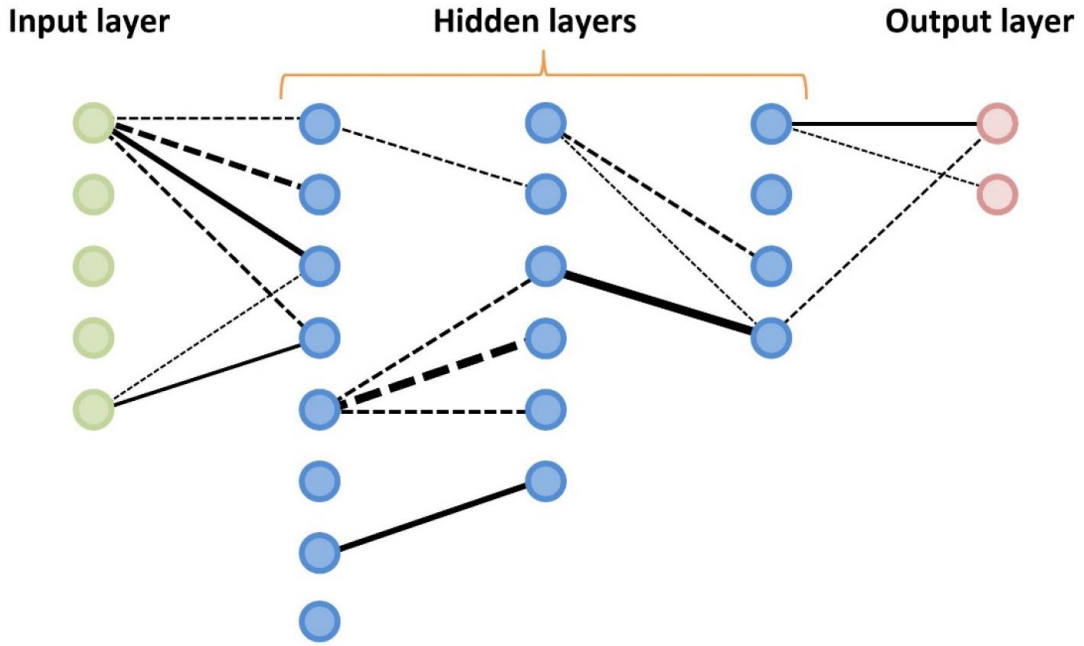


Artificial neuron



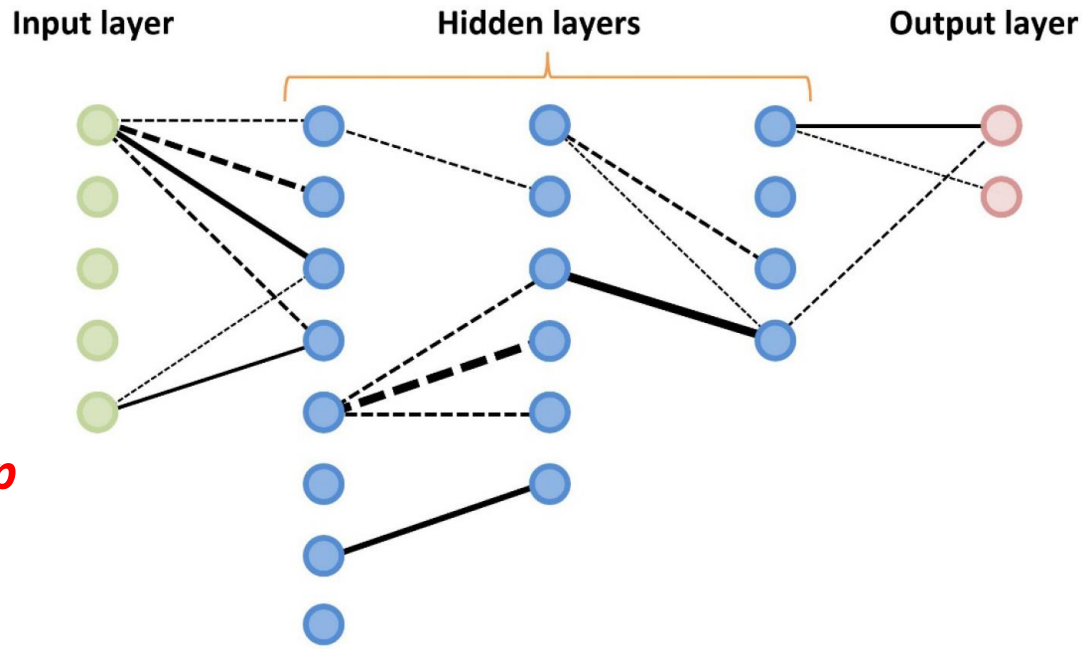
Artificial neural networks

- The real power of artificial neurons emerges when they are networked together to learn features from the data and inference from unseen instances
- An ***artificial neural network*** (ANN) is a collection of connected nodes (artificial neurons) stacked in layers



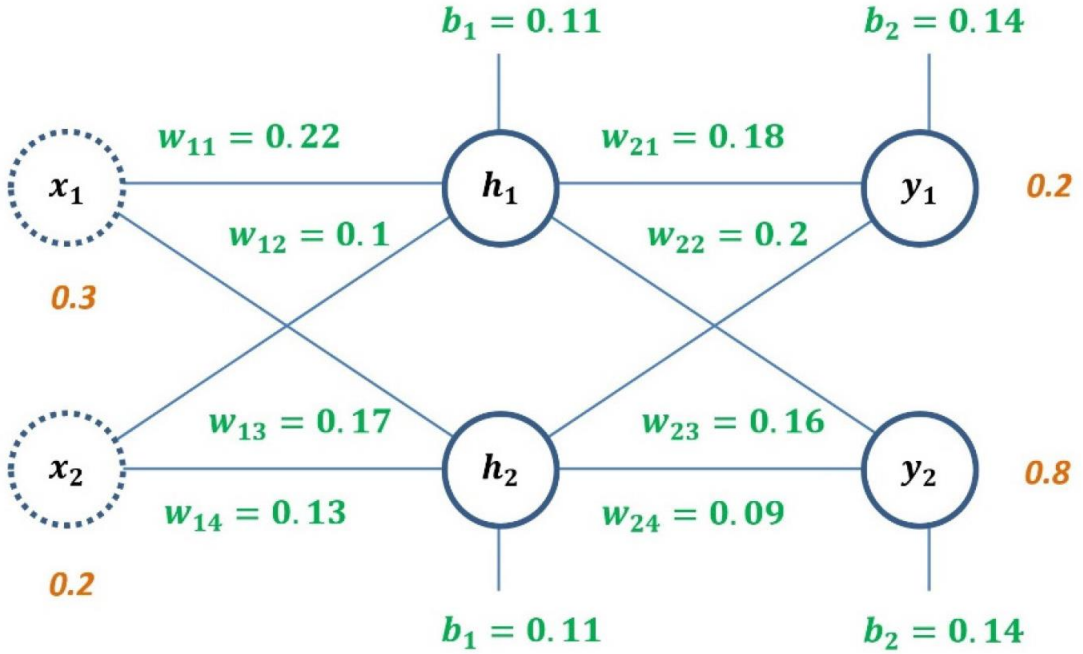
Artificial neural networks

- The ***input layer*** receives data for the ANN, and its size is restricted by the number of features in each input sample
  - E.g., an embedding vector with 256 elements
- The network includes a series of ***hidden layers***
  - They are the secret sauce of an ANN and provide its special power
  - Networks with many hidden layers are called ***deep neural networks*** (DNNs)
- The ***output layer*** is the final layer of the network and determines the result of the ANN processing
- A fully connected layer is called a ***dense layer***
  - Line width shows the strength of the connection



Training artificial neural networks

- The training process consists of a ***forward*** and a ***backward*** pass
- We feed an input sample to the ANN and calculate the error in the first pass
- In the backward one, we carry the information about the error in the reverse order and adjust the ***weights*** and ***biases*** of the network
- As we need to go through multiple layers to adjust the parameters, ***gradient descent*** is paired with another technique called ***backpropagation***

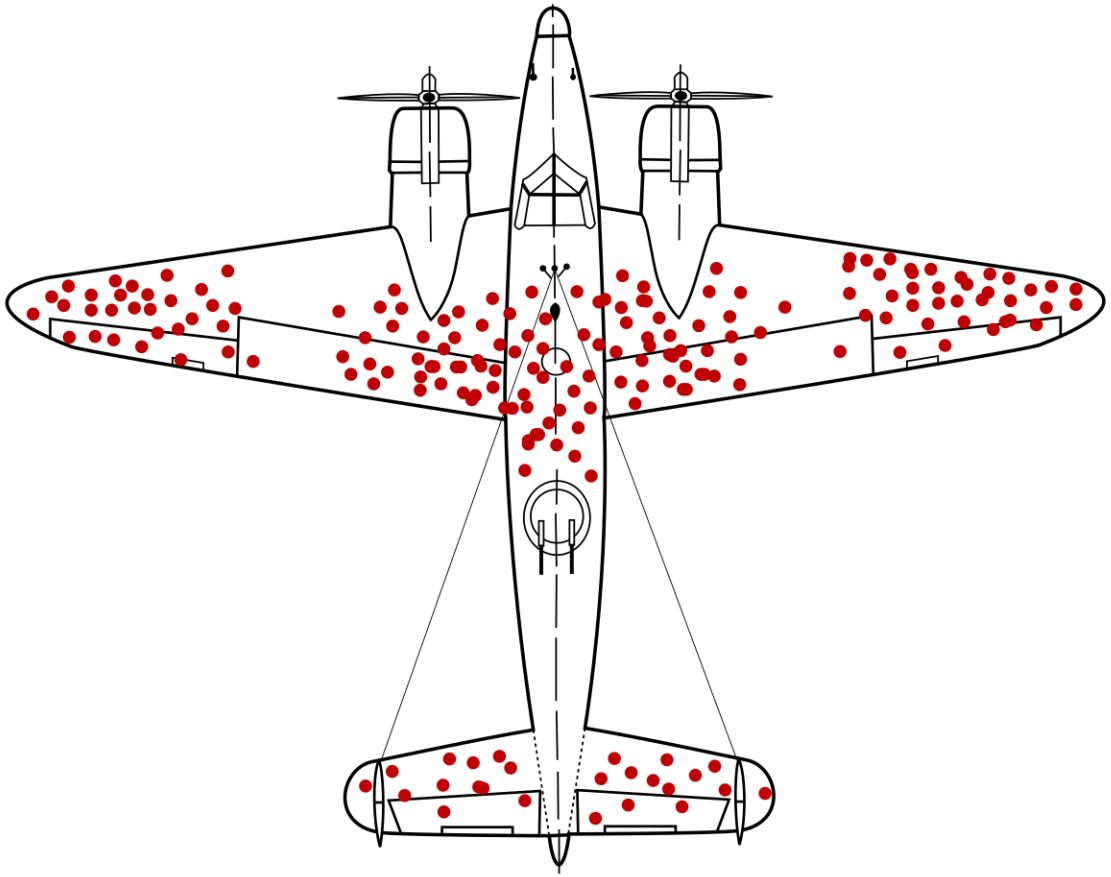


Bias

- ***Biases*** are systematic errors in ML models due to incorrect assumptions in the ML process
- ***Voluntary response bias***
  - There is an inherent problem when TV or radio shows solicit their audience to participate in online polls, especially on controversial issues
  - Responses are given by self-selected people who often have a firm opinion on the issue
- ***Survivorship bias***
  - Bill Gates, Steve Jobs, and Mark Zuckerberg are famous university dropouts that became multibillionaires
  - So, it's logical to think dropping out of university is a prerequisite to phenomenal success
  - However, this ignores the far more significant set of dropouts who never got anywhere

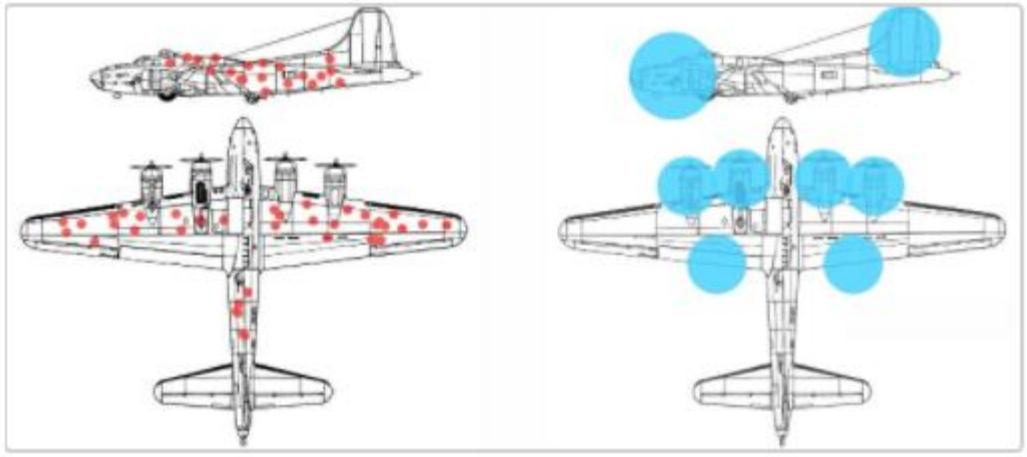
Survivorship bias

- During World War II we observed this damage pattern of surviving aircraft.
- How would you reinforce the aircrafts based on this pattern?



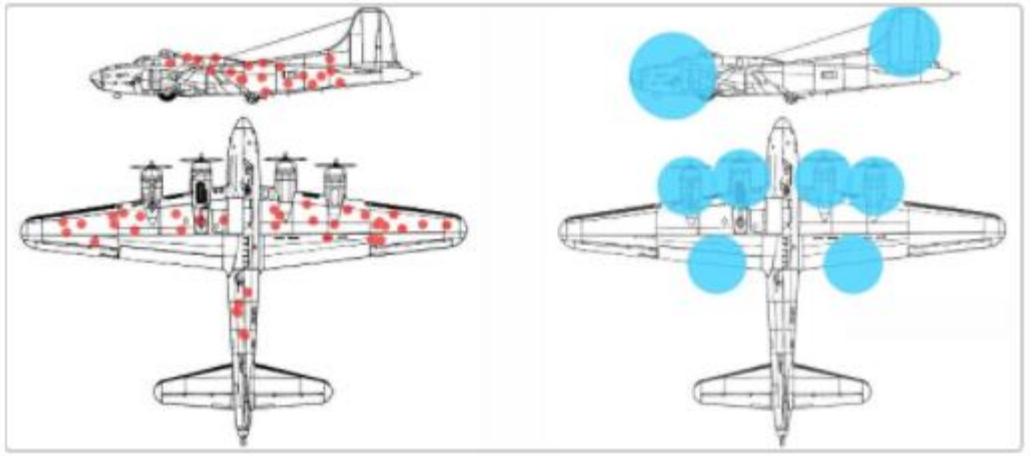
Survivorship bias

- During World War II we observed this damage pattern of surviving aircraft.
- How would you reinforce the aircrafts based on this pattern?



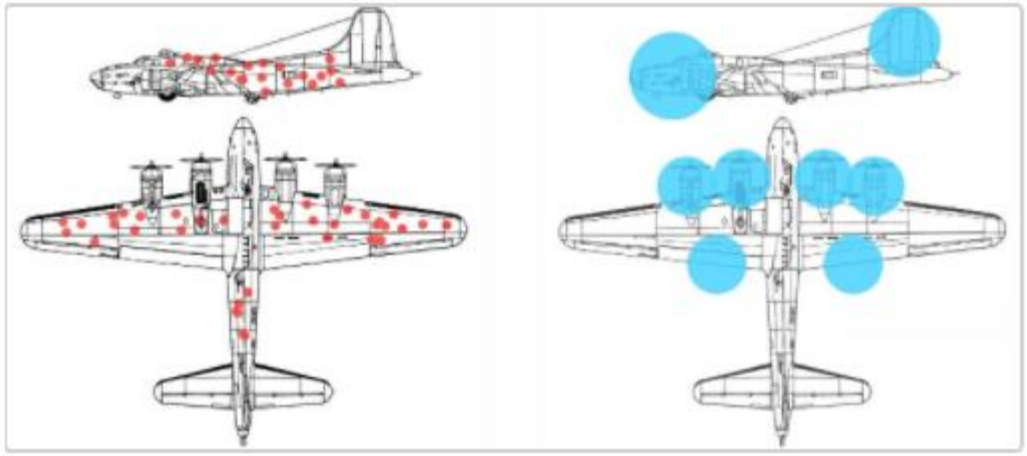
Survivorship bias

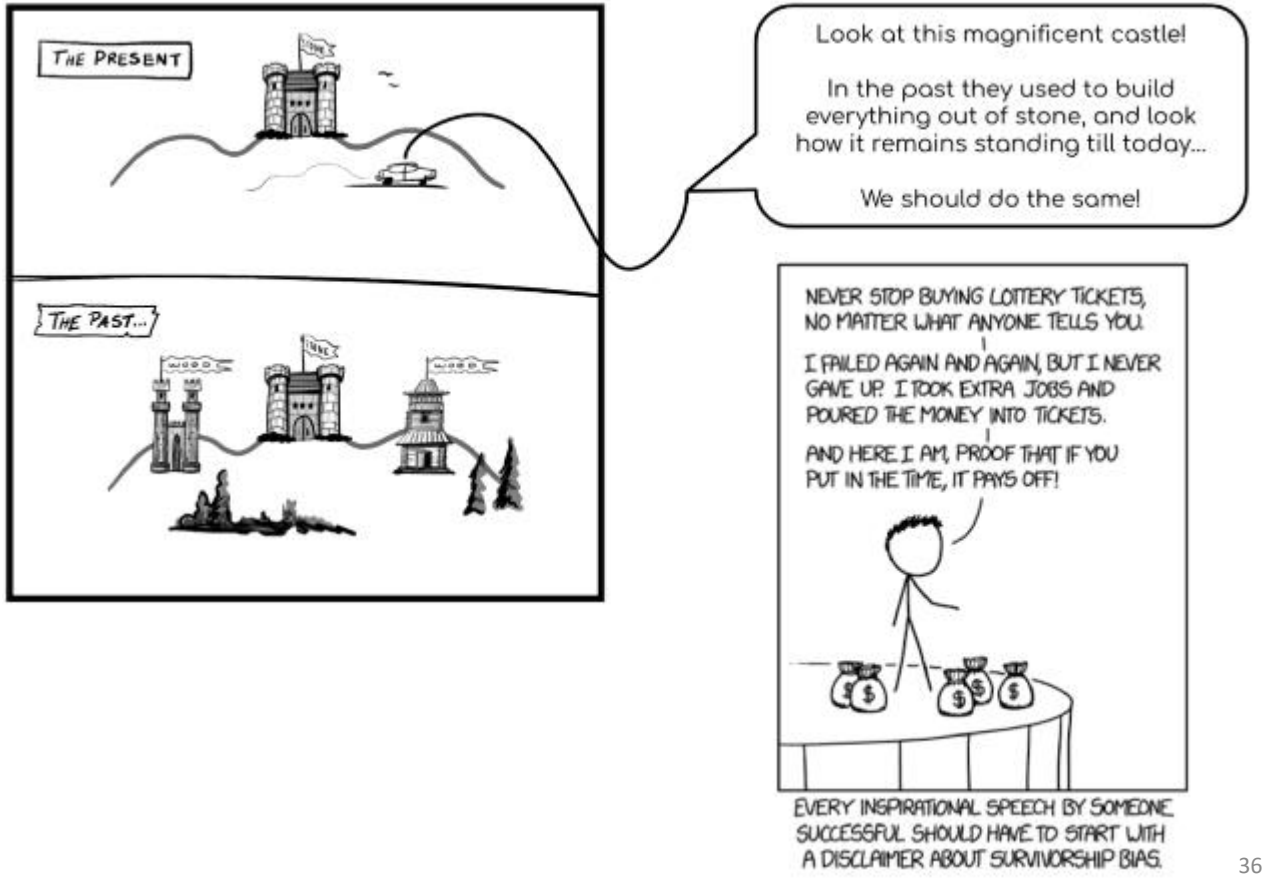
- During World War II we observed this damage pattern of surviving aircraft.
- How would you reinforce the aircrafts based on this pattern?



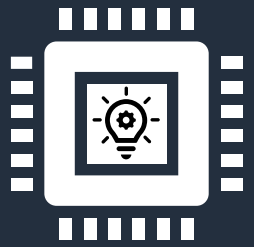
Survivorship bias

- During World War II we observed this damage pattern of surviving aircraft.
- How would you reinforce the aircrafts based on this pattern?



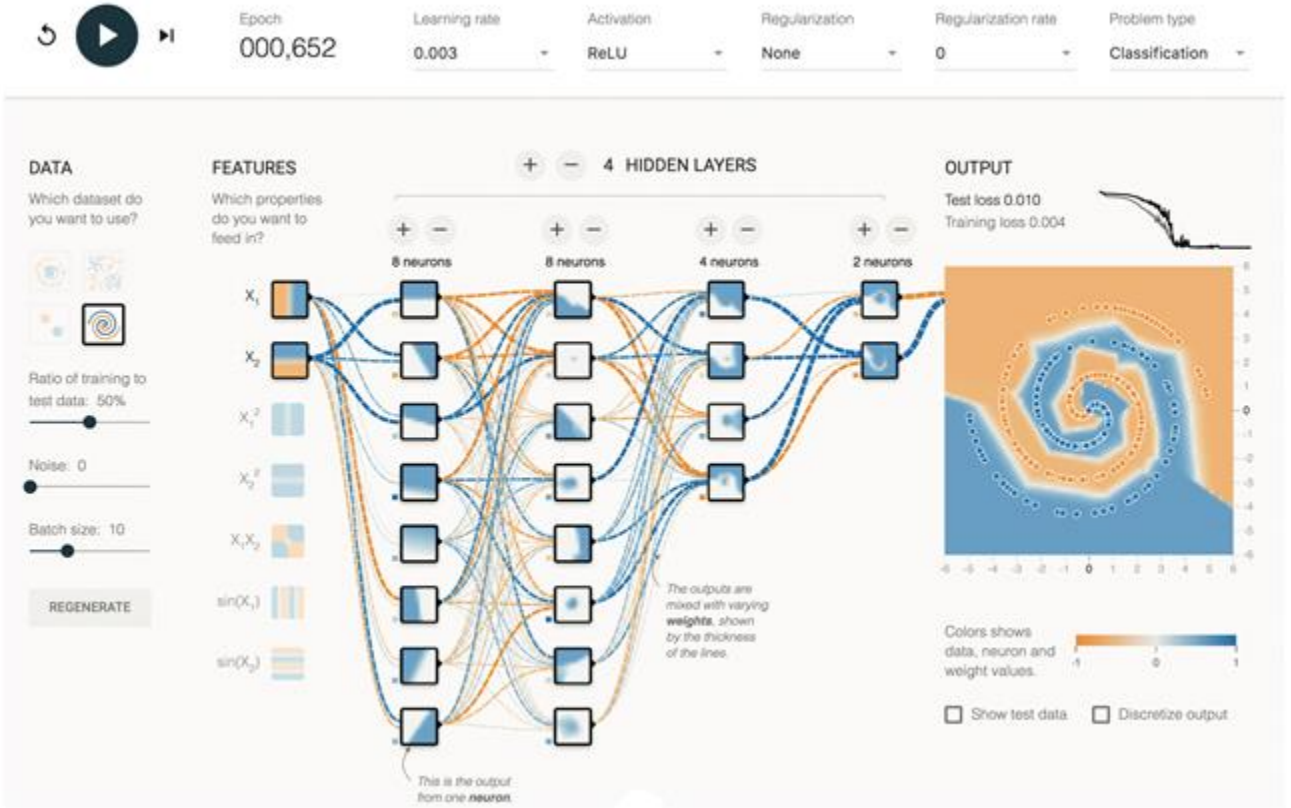


Let’s try!

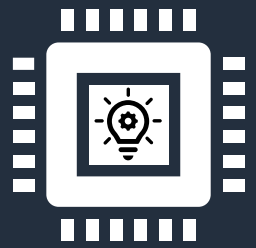


TensorFlow playground

- https://playground.tensorflow.org/



Let’s practice!



Tasks

- Exploratory data analysis
- Linear & logistic regression
- Deep Neural Networks

https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-04/sentiment-analysis.ipynb

Key takeaways

Visualizations

- Bar charts
- Box plots
- Histograms
- Bubble plots
- Scatter plots

ML concepts

- Regression
- Optimization
- Regularization
- Gradient descent

ML algorithms & models

- Linear Regression
- Logistic Regression
- Deep Neural Networks

Performance metrics

- Loss functions
- Principle of least squares

Machine Learning Techniques for Text

Questions?