Machine Learning Techniques for Text

Module 8: Detecting Hateful and Offensive Language

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

- The dramatic increase in inflammatory language, urged companies to regulate or even remove extreme posts
- On the other hand, there are raising concerns that attempts to curb inappropriate language could also lead to the restraint of free speech
- The current module aims to identify hateful and offensive language in tweets
  - In this module we will reuse and tune third-party models to minimize the effort of a new deployment
  - Using an open source dataset we will examine the steps to build a state-of-the-art language model and use it for classification
  - The presented algorithms have been in the spotlight recently due to their usage in winning prestigious competitions in the field
  - We will also utilize a validation test to adjust the model's parameters and avoid certain pitfalls
  - Finally, we will examine the strategies for dealing with imbalanced data

Module objectives

After completing this module, you should be able to:

- Implementing state-of-the-art language models
- Building more complex neural architectures
- Applying new algorithms for text classification
- Understanding the need for validation sets
- Treating imbalanced datasets

Machine Learning Techniques for Text

Section 1: Introducing social networks

Social networks

- In the late 1960s, the famous psychologist Stanley Milgram decided to investigate the small-world concept, which states that the entire world is connected through short chains of acquaintances
- Performing an ingenious experiment, Milgram asked a few hundred people from various locations to get a letter to a stranger in Boston
- The participants were given information about the target recipient and instructed to send the letter to someone they knew that would more likely know that individual

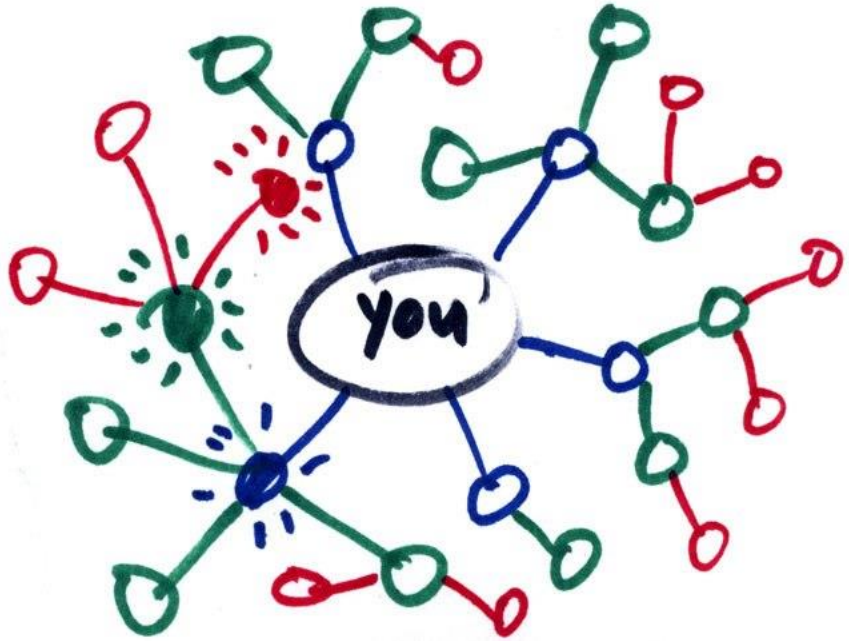
Social networks

- The following person in the chain had to repeat the same task and send the letter to someone even closer
- When Milgram examined the letters that reached the target, he realized they had changed hands about six times on average
- The result demonstrated that, on average, any two individuals in the US are separated by five connections, known by the phrase ***six degrees of separation***

*[Is the idea that all living things and everything else in the world is six or fewer steps away from each other so that a chain of "a friend of a friend" statements can be made to connect any two people in a maximum of **six steps**.]*
Wikipedia

Social networks

*[Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of **three and a half** other people.]*
Facebook Research - 2016



Hateful and offensive language

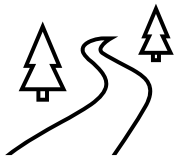
- The difference between ***hate*** and ***offensive*** speech can be subtle, with difficult-to-discern boundaries
- We can define hate speech as language that expresses hatred toward a targeted group or is intended to be derogatory, and is used to humiliate, or insult the group's members
- On the other hand, sexist tweets are generally classified as offensive
- So, let's consider three examples:
  - **Hate speech**: *I hate the ghetto trash at the special school across the street from my building. All of them will grow up to be criminals*
  - **Offensive speech**: *God, my tweets are so ghetto*
  - **Neither offensive nor non-offensive speech**: *So many weird people in the ghetto at this time*

Machine Learning Techniques for Text

Section 2: Understanding BERT

Contextualized representations of words

- Imagine you're reading a book, and you encounter the word ***bank***
- It could refer to a financial institution or the side of a river



- Traditional ***word embeddings*** might assign the same representation to both meanings, missing the nuance
- A highly perceptive reader does not only understand each word but also grasps the subtle differences based on the entire sentence
- We need models that consider bidirectional context mirrors the way our minds comprehend language in context, providing a more nuanced and accurate understanding of words based on their surroundings

BERT representation

- ***Bidirectional Encoder Representation from Transformers*** (BERT) is a state-of-the-art transformer-based technique to generate language representation models
- BERT generates word representations that are sensitive to the context in which the words appear in a sentence
- BERT incorporates a stack of transformer encoders to understand the language better
- Similarly to word embedding, the method does not require human-annotated observation labels
- Therefore, BERT can be utilized in various tasks, such as machine translation, sentiment analysis, text summarization, and so forth

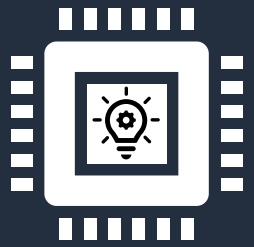
BERT representation

- A typical step when incorporating BERT is to pre-train the model to understand the language and adjust it for specific applications
- This way, the knowledge that is extracted during the pre-training phase, which takes place once, can be transferred to several applications without much effort
- This feature is essentially the basic idea behind ***transfer learning***, where we first pre-train a model using a large dataset and then fine-tune it for a specific task using a smaller one

Transfer learning

- Consider, for example, applications for recognizing human faces
- All these applications share a common step – extracting the human face's basic characteristics, such as the eyes, nose, and mouth, and their differences rely mainly on the setting
- For example, a video surveillance system aims to identify a human face from a distance, while for an access control system, the person is usually close to the camera
- A robust pre-trained model can be tuned with a smaller context-dependent dataset in both cases
- In NLP we can resort to powerful language models and adapt them the peculiarities of a specific task without starting from scratch

Let’s practice!



Tasks

- BERT

https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-08/social-networks-bert.ipynb

Machine Learning Techniques for Text

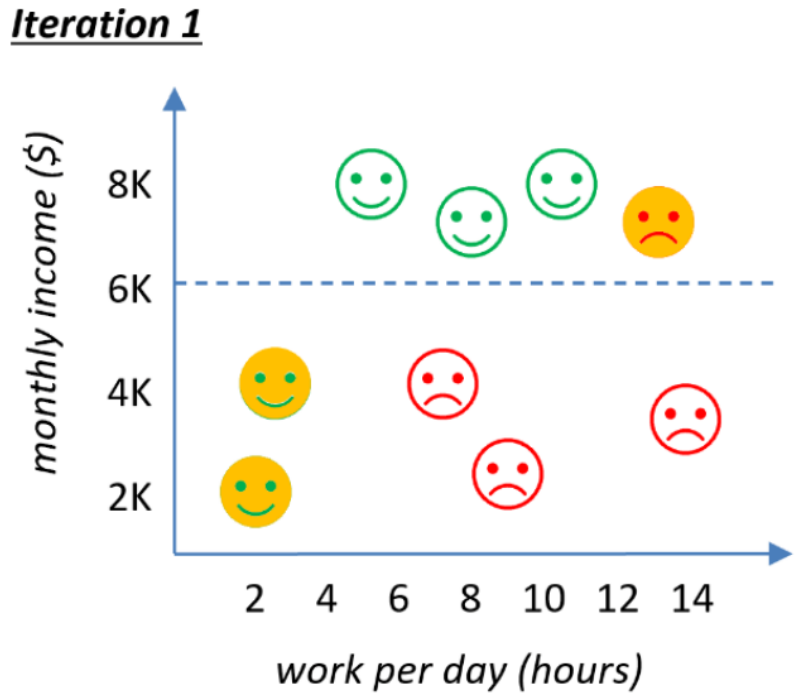
Section 3: Introducing boosting algorithms

Introduction

- The term ***boosting*** refers to a family of algorithms that use ensemble learning to build a collectively robust classifier from several weak classifiers
- The difference with other ensemble techniques is that in boosting, we build a series of trees, where every other tree tries to fix the mistakes made by its predecessor
- With the random forest classifier, multiple trees are constructed in parallel using the ***bagging*** technique
- Another distinctive characteristic of boosting algorithms is their ability to deal with the ***bias-variance*** trade-off

AdaBoost

- The algorithm creates a very shallow tree (**T1**) from the training data during the first iteration
- The decision based on this stump* is depicted with the horizontal line, ***y=6***, so that a person receiving more than $6K per month is labeled with a happy face
- The **F1** model consists only of the specific decision tree

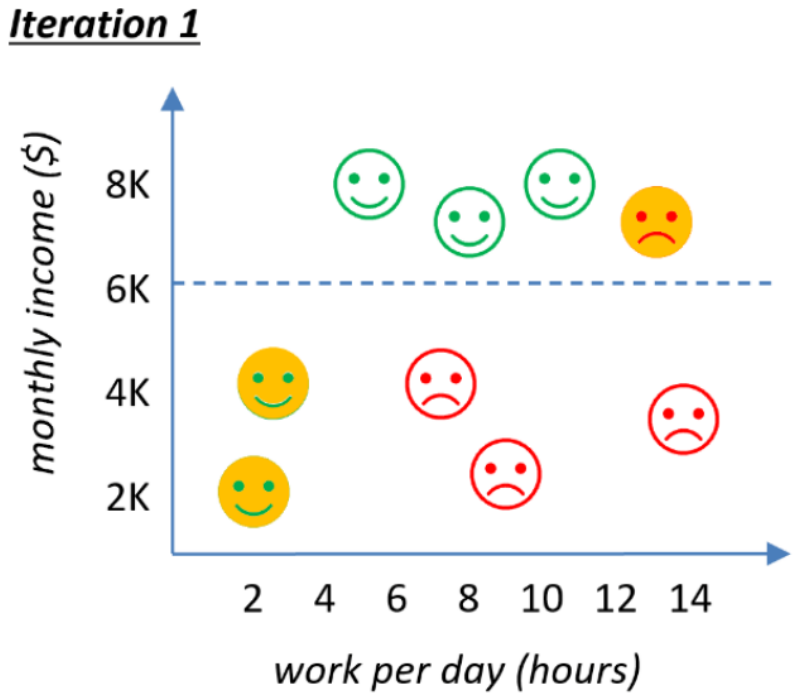


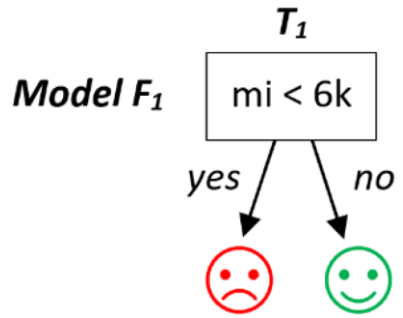
stump = a decision tree with a single split

AdaBoost

- The algorithm creates a very shallow tree (**T1**) from the training data during the first iteration
- The decision based on this stump* is depicted with the horizontal line, ***y=6***, so that a person receiving more than $6K per month is labeled with a happy face
- The **F1** model consists only of the specific decision tree

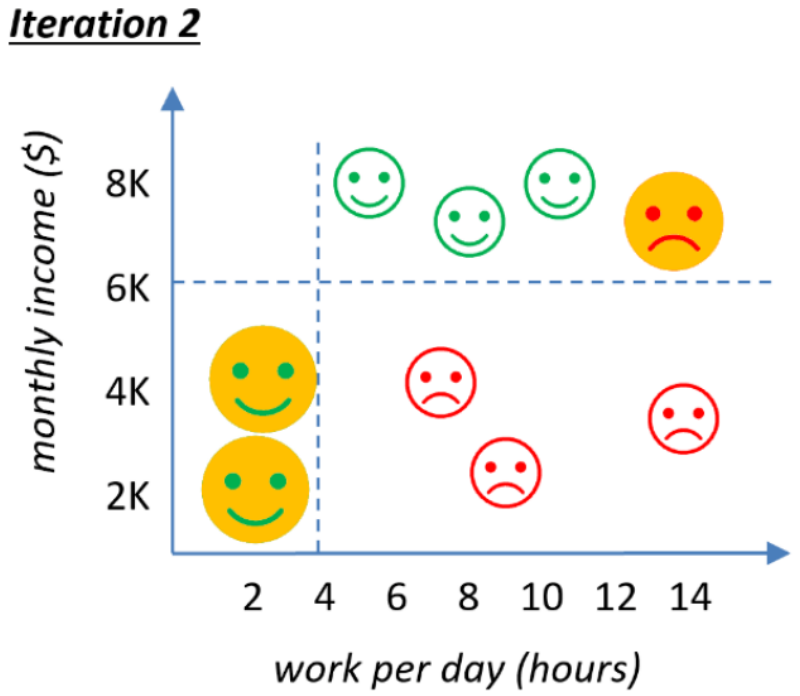
*stump = a decision tree with a single split





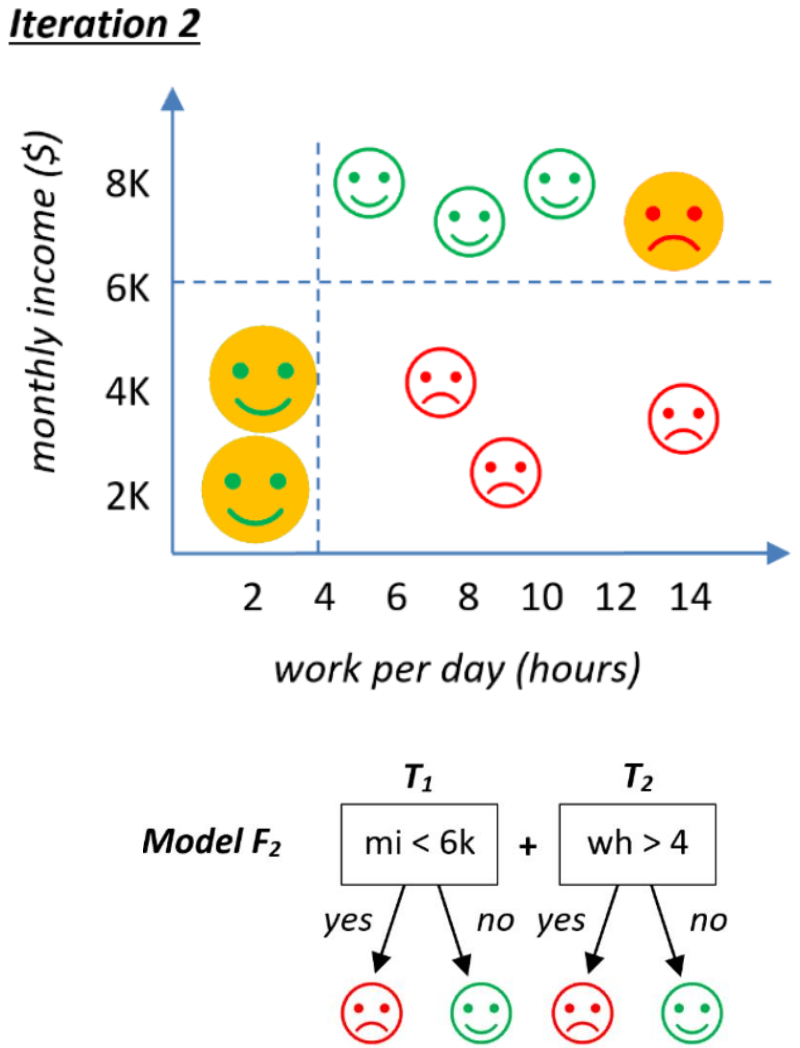
AdaBoost

- In the second iteration, a new stump is created (**T2**) that tries to rectify the previous errors
- The incorrectly classified observations now carry more weight than the observations that were correctly classified
- For this reason, the corresponding icons appear larger than the others
- The vertical line, ***x=4*** (the stump's outcome), corrects two errors



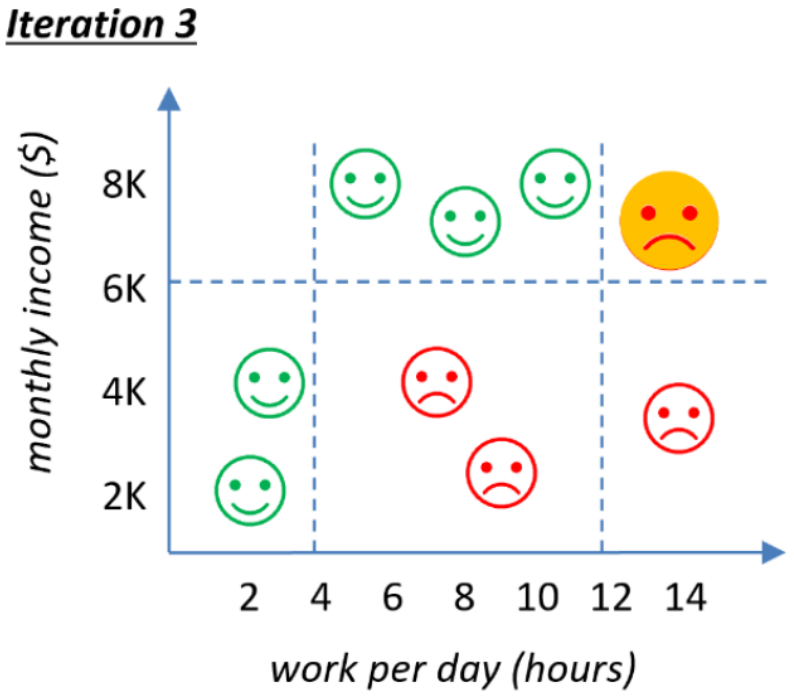
AdaBoost

- In the second iteration, a new stump is created (**T2**) that tries to rectify the previous errors
- The incorrectly classified observations now carry more weight than the observations that were correctly classified
- For this reason, the corresponding icons appear larger than the others
- The vertical line, ***x=4*** (the stump's outcome), corrects two errors



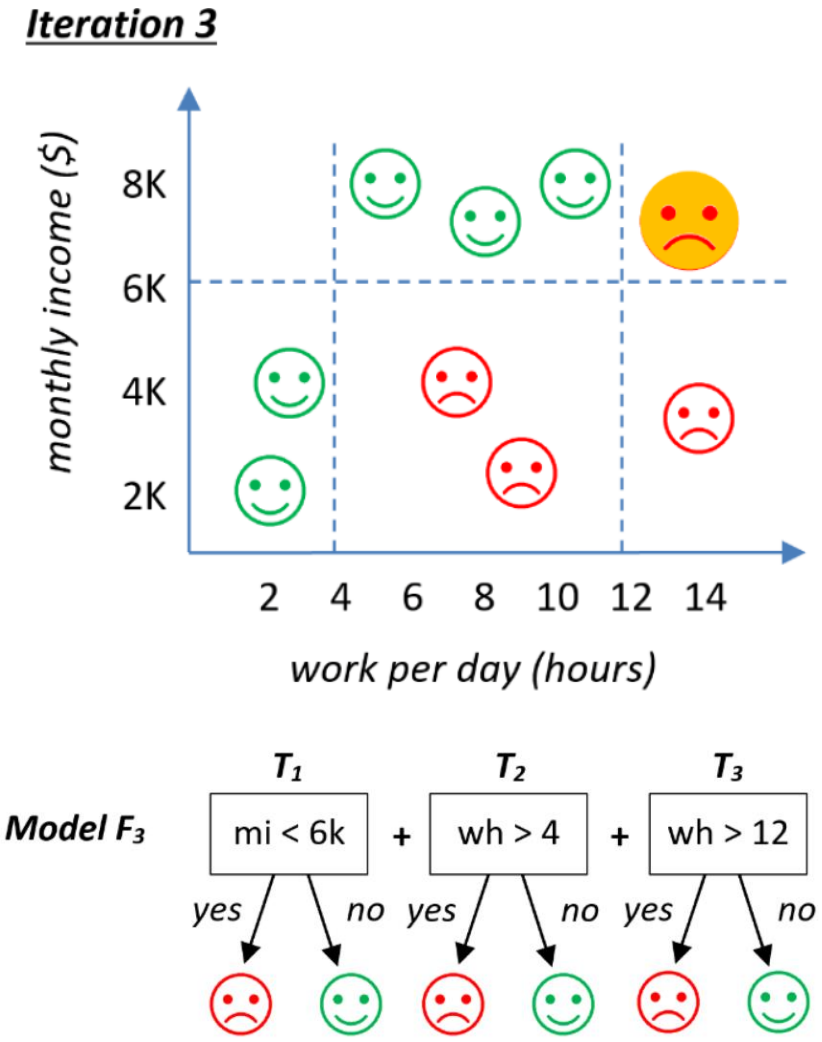
AdaBoost

- Only one error remains unresolved during the third iteration, treated by the **T3** stump
- Another vertical line, ***x=12***, separates the sad smiley faces from the happy ones
- Finally, the new **F3** model is the sum of **T1**, **T2**, and **T3** that correctly classifies all the data points



AdaBoost

- Only one error remains unresolved during the third iteration, treated by the **T3** stump
- Another vertical line, ***x=12***, separates the sad smiley faces from the happy ones
- Finally, the new **F3** model is the sum of **T1**, **T2**, and **T3** that correctly classifies all the data points



Gradient boosting

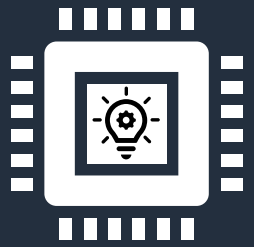
- ***Gradient boosting*** is an extension of boosting, where gradient descent is used to boost weak models
- A primary component of the algorithm is the loss function, such as the ***mean squared error*** for regression or the ***logarithmic loss function*** for classification
- The only prerequisite is that the function is differentiable to apply gradient descent
- Although many models can be used for weak learners, decision trees are almost always incorporated in practice

XGBoost

- ***eXtreme Gradient Boosting*** (XGBoost) is a popular and efficient open source implementation of the gradient boosting tree algorithm
- In recent years, it has been a favored choice in various **Kaggle*** competitions, helping people win significant prizes
- XGBoost can be used for regression, classification, and ranking problems
- The method builds upon the concepts of supervised machine learning, decision trees, ensemble learning, and gradient boosting already presented in the previous modules

**Kaggle is an online community of data scientists and machine learning enthusiasts*

Let’s practice!



Tasks

- BERT
- BERT + XGBoost

https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-08/social-networks-bert-xgboost.ipynb

Machine Learning Techniques for Text

Section 4: Creating validation sets

The need for validation set

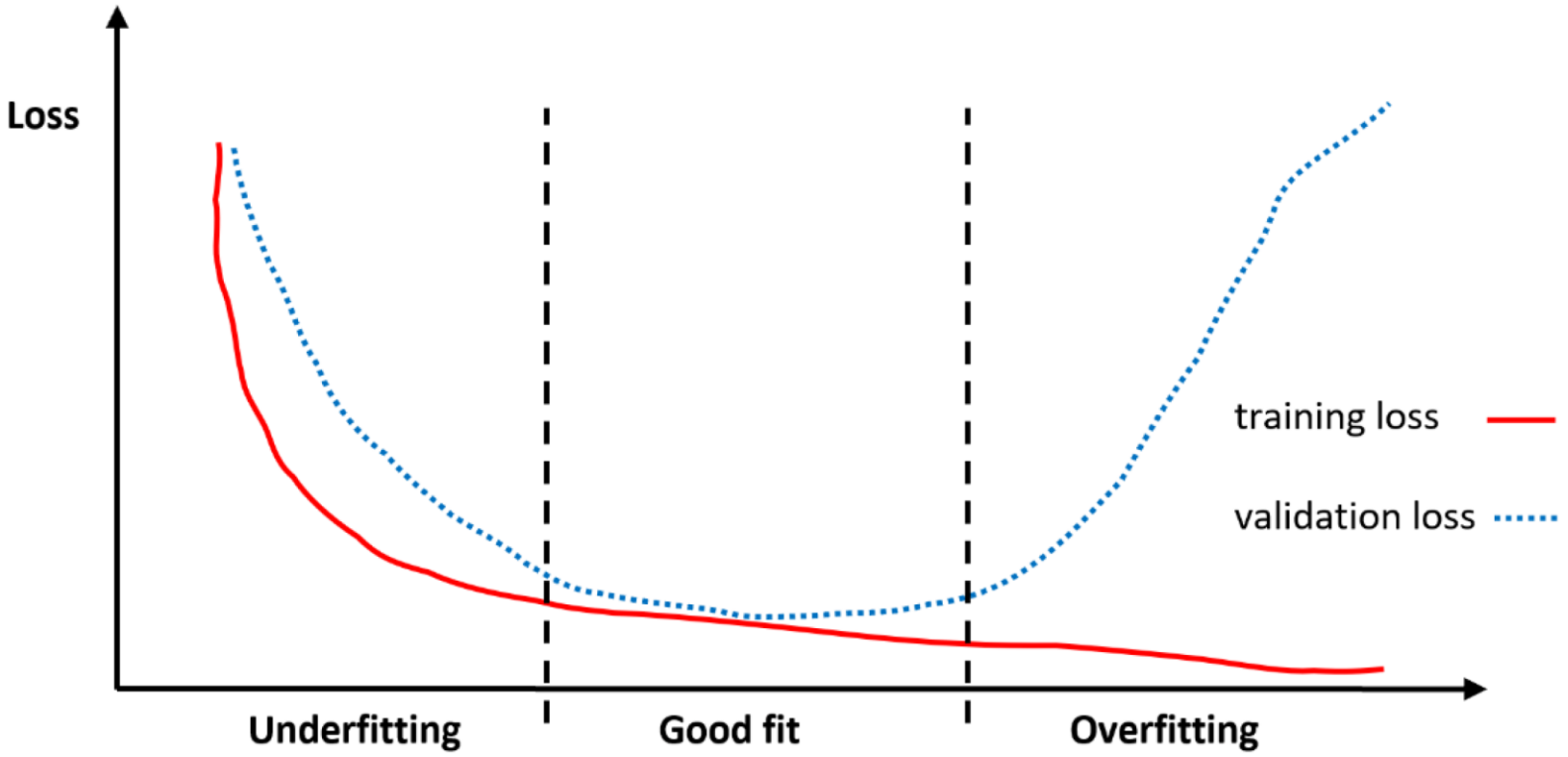
- During training we need to experiment with multiple configurations of the models to find the optimal one
- Typically, we need to adjust the hyperparameters and the topology of deep learning architecture, training on a set of samples, and testing on another set
- For that reason, machine learning is a highly ***iterative process***
- This strategy engenders a particular risk, however:
  - Evaluating different model configurations with a given test set over multiple rounds leads to a model tuned to work well with the specific set
  - As the number of epochs increases, we implicitly fit the model to the peculiarities of the test set and consequently get a too-optimistic performance in the end

The need for validation set

- We need a way to validate our model performance during training while leaving the test set for the final evaluation
- This role is undertaken by the ***validation set*** that helps us tune the model's hyperparameters and configurations accordingly
- As a result, the model learns the patterns from the training samples without overfitting
- On the other hand, the validation set creates a model bias and is unsuitable for evaluating its generalization performance
- The test set keeps unseen samples in the model until the very end, which is why it can offer an honest assessment of the final model

Early stopping

- When should we stop the training process?



Machine Learning Techniques for Text

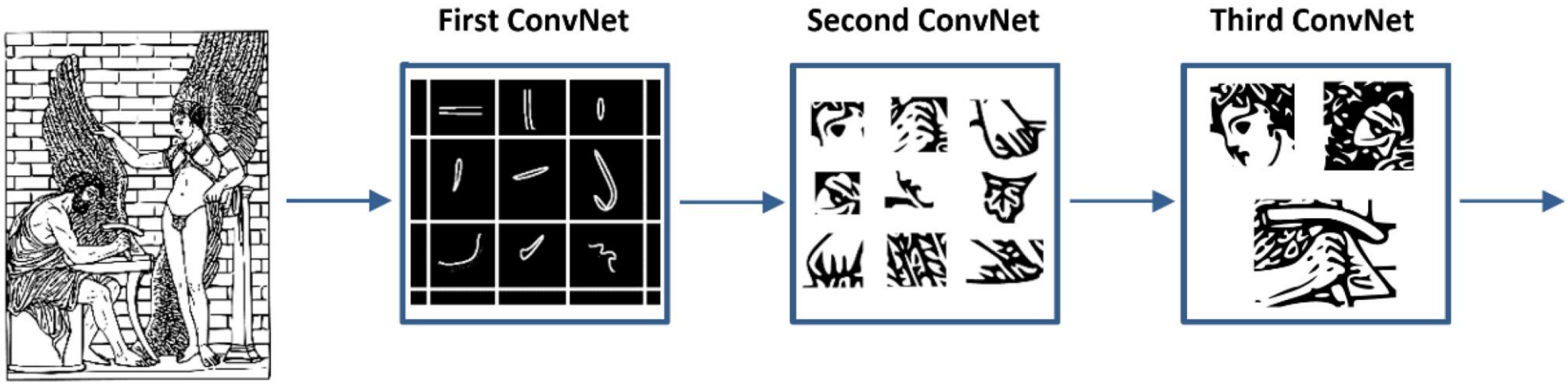
Section 5: Understanding convolutional neural network

Convolutional neural network

- A ***convolutional neural network*** (CNN or ConvNet) is a category of neural network
- It can include one or more convolutional layers capable of efficiently processing spatial patterns in data with a grid-like topology
- Therefore, CNNs find extensive utility in image-processing applications that work with two-dimensional image data
- The layers are arranged in such a way as to detect simpler or more complex patterns

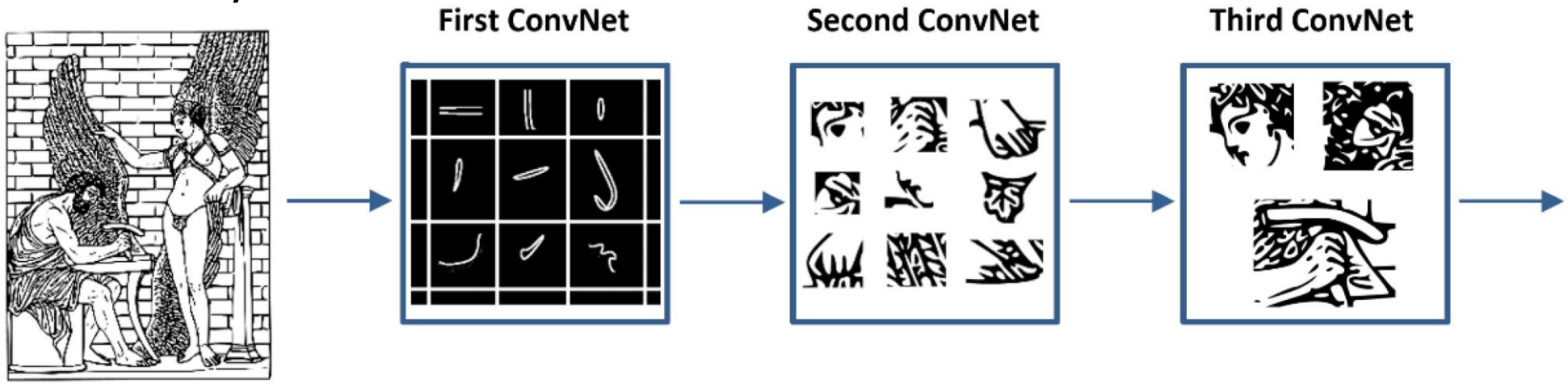
Convolutional neural network

- In an image classification task, the first layers can identify simpler features such as lines and arcs
- In contrast, the layers, further along, can detect patterns such as part of a face or an object
- So, a CNN made of a single layer can only learn low-level features, and in typical applications, we stack more than one



Convolutional neural network

- Each convolutional layer of the network includes a set of ***kernels***, also known as ***filters***, that aim to extract different features from the input
- In **First ConvNet** we observe six filters to detect the different edges of the image
- Their output is fed to **Second ConvNet** to identify higher-level features, such as an eye or fingers
- Finally, the last layer can detect more complex patterns based on the output of the second layer



The convolution operation

- ***Convolution*** is a linear operation, depicted with the * symbol, and involves the multiplication of the set of weights of the filter with the input

Input (I)

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

*

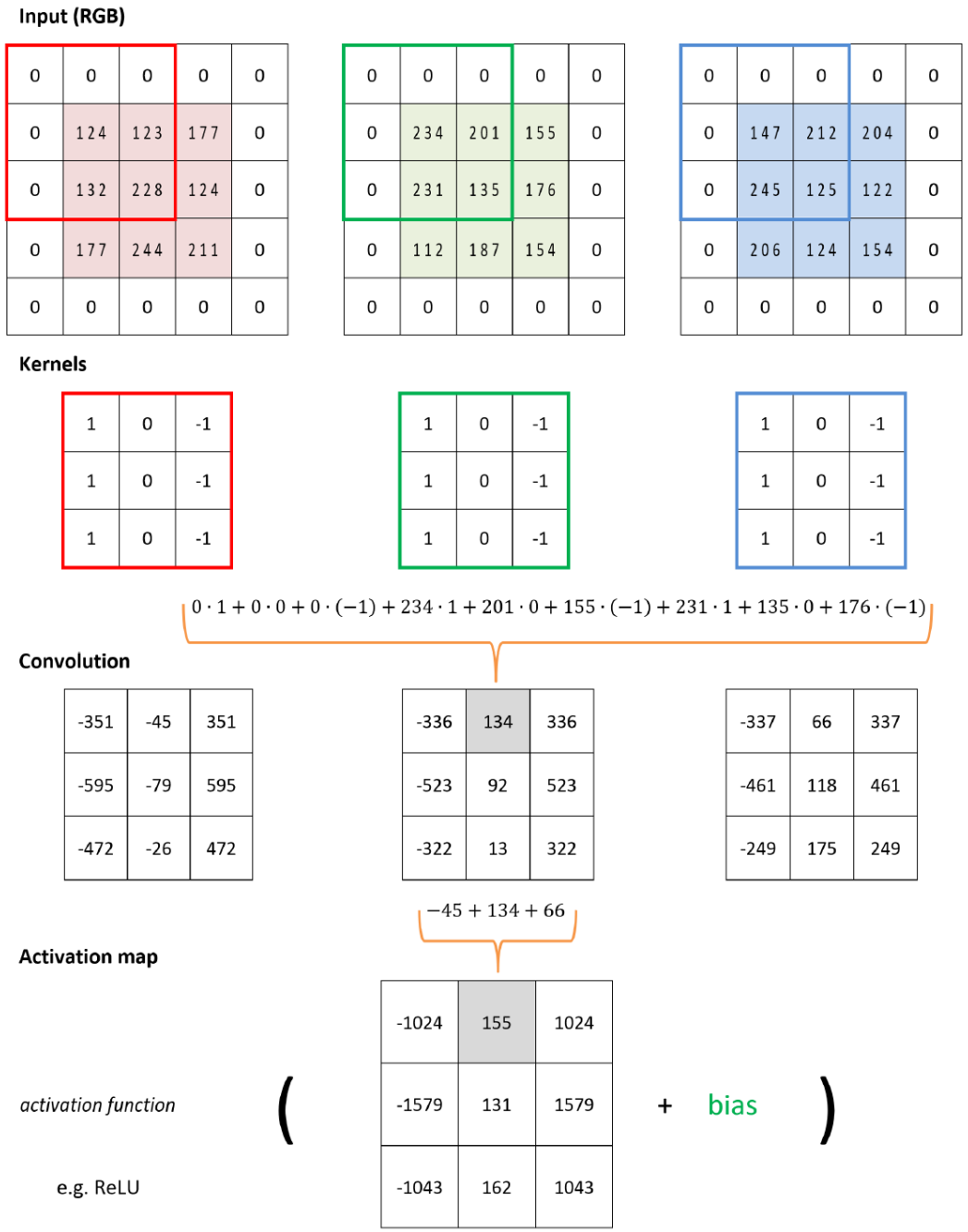
Kernel (K)

w	x
y	z

=

Input * Kernel (I*K)

aw+bx+ ey+fz	bw+cx+ fy+gz	cw+dx+ gy+hz
ew+fx+ iy+jz	fw+gx+ jy+kz	gw+hx+ ky+lz
iw+jx+ my+nz	jw+kx+ ny+oz	kw+lx+ oy+pz

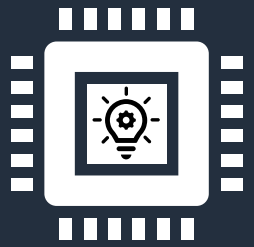


Each input matrix is convolved with the same filter that detects vertical lines in images

The output is summed and passed through an activation function, including a bias term

The result is an ***activation map***, also known as a ***feature map***, that defines which information is passed to the next layer

Let’s practice!



Tasks

- BERT
- BERT + XGBoost
- BERT + CNN

https://colab.research.google.com/github/PacktPublishing/Machine-Learning-Techniques-for-Text/blob/main/chapter-08/social-networks-bert-cnn.ipynb

Key takeaways

Visualizations

- Violin plots

Text representations

- BERT

ML algorithms & models

- Gradient Boosting
- AdaBoost
- XGBoost
- Convolutional Neural Networks

ML concepts

- Transfer learning
- Imbalanced datasets
- Validation sets
- Early stopping
- Pooling

Machine Learning Techniques for Text

Questions?