O'REILLY®

Advanced Machine Learning: Create Real-World ML Projects

with Noureddin Sadawi

About the Speaker

Name: Dr Noureddin Sadawi

Qualification: PhD Computer Science

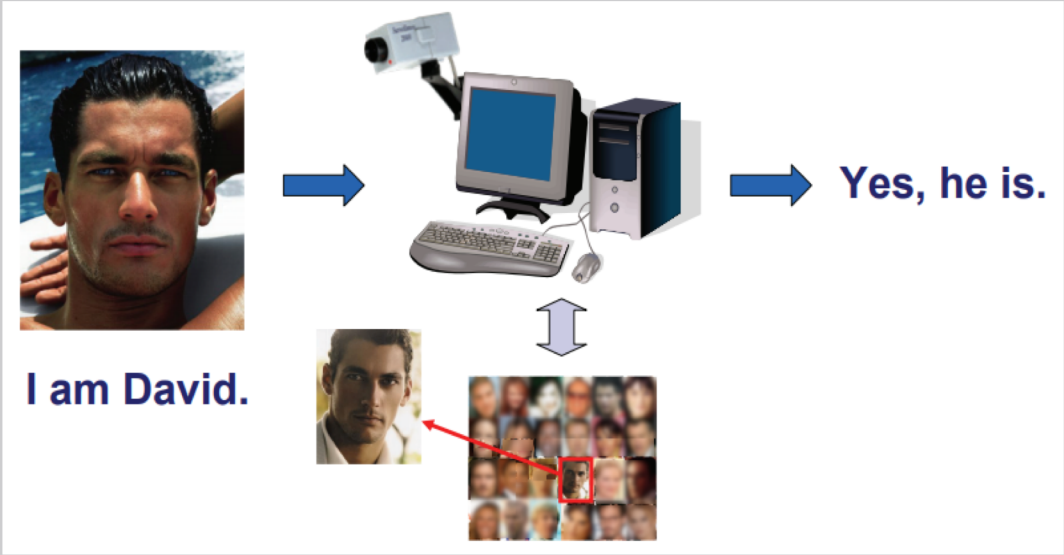
Experience: Several areas including but not limited to Docker, Machine Learning and Data Science, Python and much more

What is Face Recognition

- Who is this person?
- Is s/he who he claims to be?
- We would like to use the computer to decide
- Do it automatically and in real time

How Can we do this?

- Capture a face image
- Compare it to an existing database of face images



Example Application Areas

- Civil applications:
  - National ID, passport, driver's license, border control
  - Surveillance of public places (airports, metro stations, etc)
  - Forensic applications
- Security applications for electronic transactions and access control
  - Physical access
  - Secure access to networks and infrastructures
  - e-health, e-commerce, e-banking (and now mobile…)

Example Application Areas

- Ambient Intelligence
  - Smart homes
  - Natural human-machine interaction
- Wearable systems
  - Memory aids and context-aware systems
- Entertainment
  - Interactive movies, computer games
- Search
  - Picasa 3.5 face recognition application for finding and managing photos

Face Recognition in Humans

- The human visual system starts with a preference for face-like patterns
- The human visual system devotes special neural mechanisms for face perception
- Facial identity and expression might be processed separately
- Humans can recognize faces in very low dimensional images
- Tolerance to image degradation increases with familiarity
- Color and texture are as important as shape
- Illumination changes influence generalization

"Face recognition by humans: 19 results all computer vision researchers should know about"

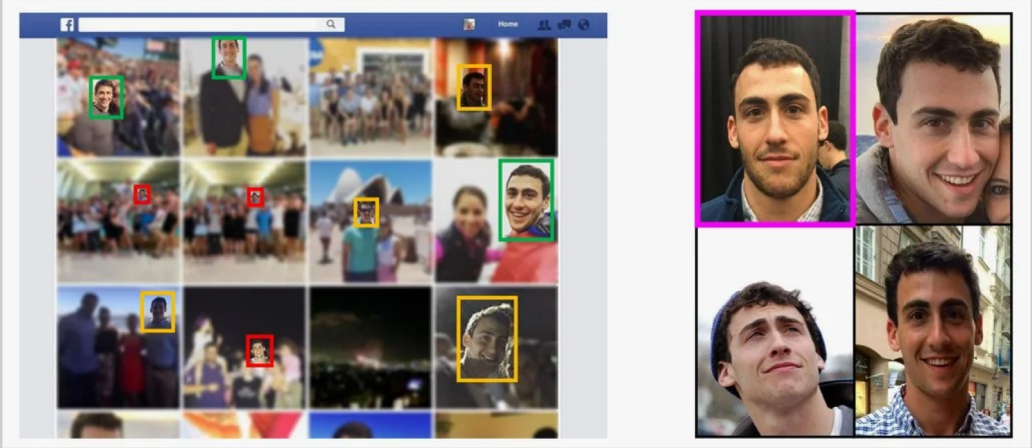
http://web.mit.edu/sinhalab/Papers/19results_sinha_etal.pdf

Challenges: Intrapersonal variations

- If people can do it so easily, why can't computers?
- Intrapersonal (intra-class) variations are variations of the appearance of the same face caused by:
  - Illumination variations
  - Pose variations
  - Facial expressions
  - Use of cosmetics and accessories, hairstyle changes
  - Temporal variations (aging, etc)

Challenges: Tricks

- [Hackers Trick Facial-Recognition Logins With Photos From Facebook (What Else?)]



Face Detection

- But .. before attempting to guess whose face it is .. the system needs to make sure it is a face
- Face detection: find all faces in an image/video (if any) regardless of their position, scale, in plane rotation, pose, illumination, facial expressions, occlusions
  - First step to every face recognition system
- Face localization: find the exact location of a detected face
  - Detection of salient facial features such as eyes, nose, nostrils, eyebrows, mouth, etc
- Face tracking: detect (“follow”) a face in a video sequence

Example Face Detection Techniques

- Knowledge-based: Translate knowledge about typical face to a set of rules
- Structural matching: Statistical models of shape appearance based on a set of landmarks
- Template matching: Standard patterns stored to describing the face or facial features
- Feature invariant: Find features of the face invariant to appearance variations (facial features, edges, shape, texture, skin color)
- https://paperswithcode.com/task/face-detection/codeless

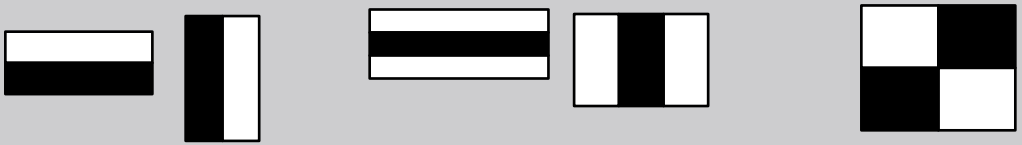
Haar Features 1/2

- Haar features consist of two or more rectangles and encode intensity differences between neighboring areas
- A cascade of boosted classifiers working with Haar features used to classify image regions as face or non-face
  - Classifiers at earlier stages use fewer Haar features
  - Feature selection is based on the Adaboost algorithm → features sorted in order of importance
  - Fast and robust, but time-consuming training (days...)

P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, Proc. Int. Conf. on Computer Vision and Pattern Recognition, 2001

Haar Features 2/2

- The main feature types are:
  - Edge Features Line Features Four-rectangle Features



- Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle
- Calculating these sums for the entire image would be very computationally expensive
- The Viola-Jones algorithm solves this by using the **integral** image.
- Resulting in an O(1) running time of the algorithm

OpenCV

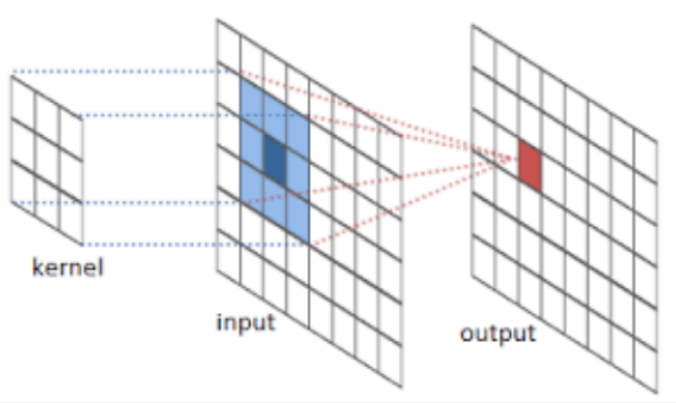
- OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.
- Created by Intel in 1999, it is written in C++ (we will be using the Python bindings)
- It contains many popular algorithms for computer vision, including object detection and tracking algorithms built-in
- It is a big library with a large number of functionalities for image and video processing and analysis
- Deserves a full course on it!

Face Detection -> Face Recognition

- In this course we are going to use OpenCV to connect to our webcam and detect our faces in real time
- We will use the Haar algorithm for face detection
- We will draw a rectangle around each face
- For face recognition we will develop our own deep learning approach using convolutional neural networks (CNNs)
- So let us have a quick overview of what CNNs are and how they work
- For implementation we will use Keras + Tensorflow

CNNs

- CNNs are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers



- Convolution is a mathematical operation having a linear form

Input and Output 1/2

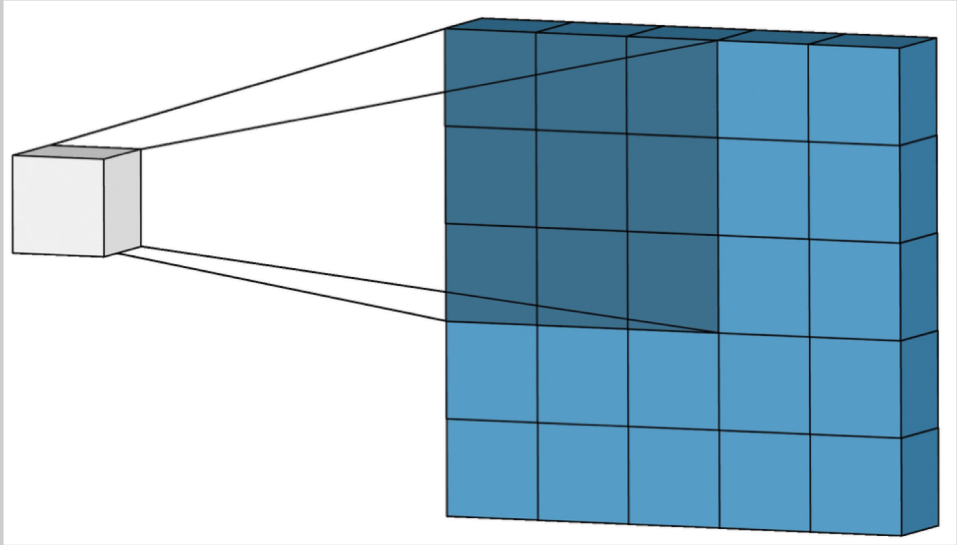
- When a computer sees an image (takes an image as input), it will see an array (or matrix) of pixel values
- Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values)
  - Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480
  - The representative matrix will be 480 x 480 x 3
  - Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point

Input and Output 2/2

- These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer
- The idea is that we give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class
    - 0.80 for cat, 0.15 for dog, 0.05 for bird, etc

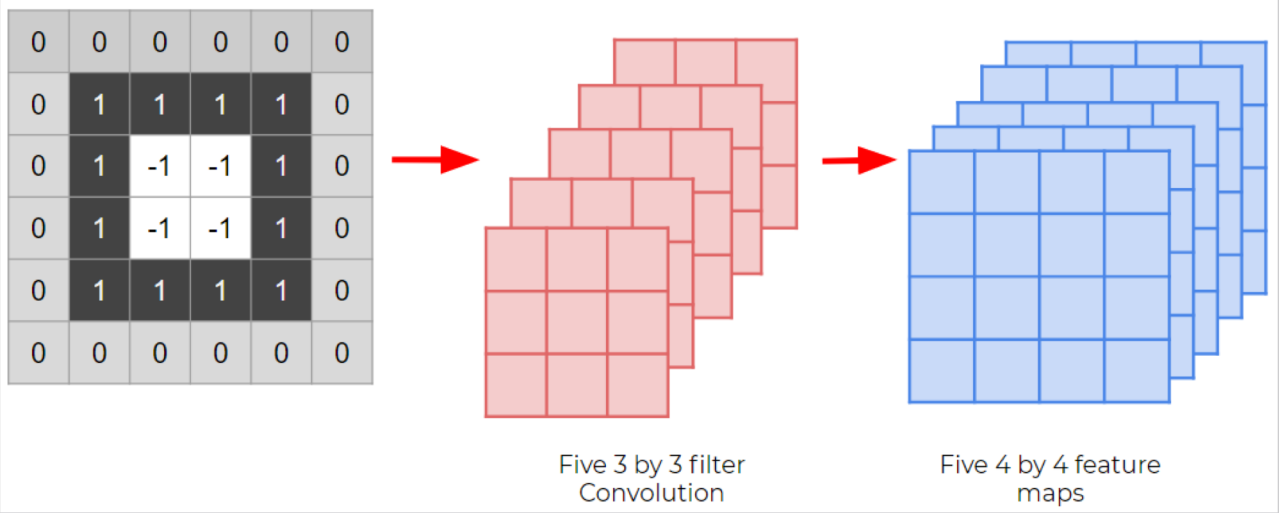
Useful 3 part intro/tutorial:
https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

Convolution



Nice presentation on CNNs with intuition and examples:
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture05.pdf

How Filters are applied



- Representation of Multiple Filters
- Each Filter generates a feature map

Filters in CNNs

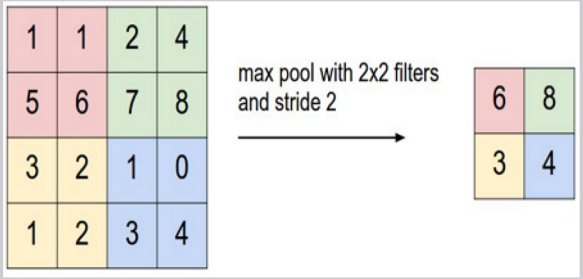
- In CNNs, filters are **learnt**
- We do not have to have a fixed value of filter elements
    - Like we do when we sharpen or find edges etc
- Each filter generates a different feature map
- The number of resulting feature maps is the same as the number of filters

Feature Maps (aka Activation Maps)

- They result after conv layers
- The more filters, the greater the depth of the activation map, and the more information we have about the input image
- As you go through the network and go through more conv layers, you get feature maps that represent more and more complex features

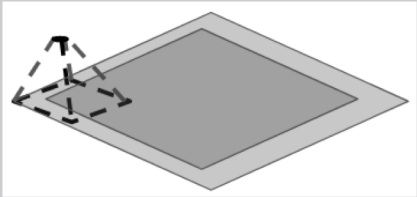
Pooling

- Subsamples the input signal, which reduces the memory use and computer load as well as reducing the number of parameters
- This pooling layer will end up removing a lot of information, even a small pooling “kernel” of 2 by 2 with a stride of 2 will remove 75% of the input data
- Stride can be controlled
- You can apply **max** pooling, **average** pooling and others



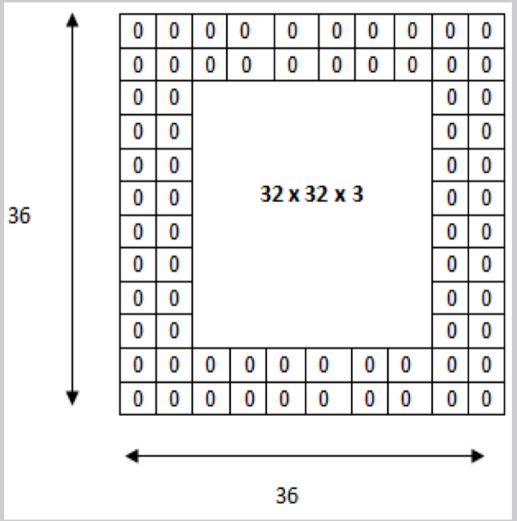
Padding 1/2

- When we apply filters, the spatial dimensions decrease
  - As we keep applying conv layers, the size of the volume will decrease faster than we would like
- In the early layers of our network, we want to preserve as much information about the original input volume so that we can extract those low level features
- Let's say we want to apply the same conv layer but we want the output volume to remain the same (i.e. 32 x 32 x 3)



Padding 2/2

- To do this, we can apply a zero padding of size 2 to that layer
- Zero padding pads the input volume with zeros around the border
- If we think about a zero padding of two, then this would result in a 36 x 36 x 3 input volume



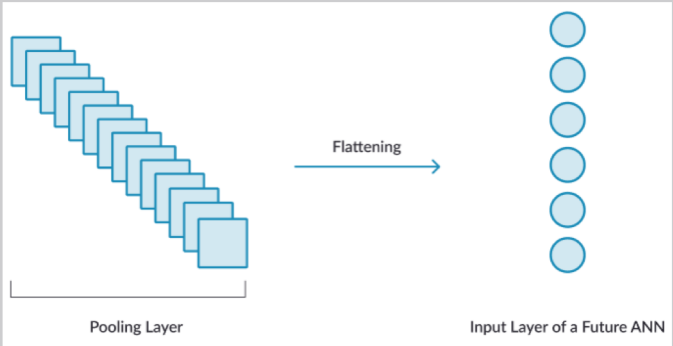
The input volume is 32 x 32 x 3. If we imagine two borders of zeros around the volume, this gives us a 36 x 36 x 3 volume. Then, when we apply our conv layer with our three 5 x 5 x 3 filters and a stride of 1, then we will also get a 32 x 32 x3 output volume.

Dropout

- Another common technique deployed with CNN is called “Dropout”
- Dropout can be thought of as a form of regularization to help prevent overfitting
- During training, units are randomly dropped, along with their connections
- This helps prevent units from “co-adapting” too much

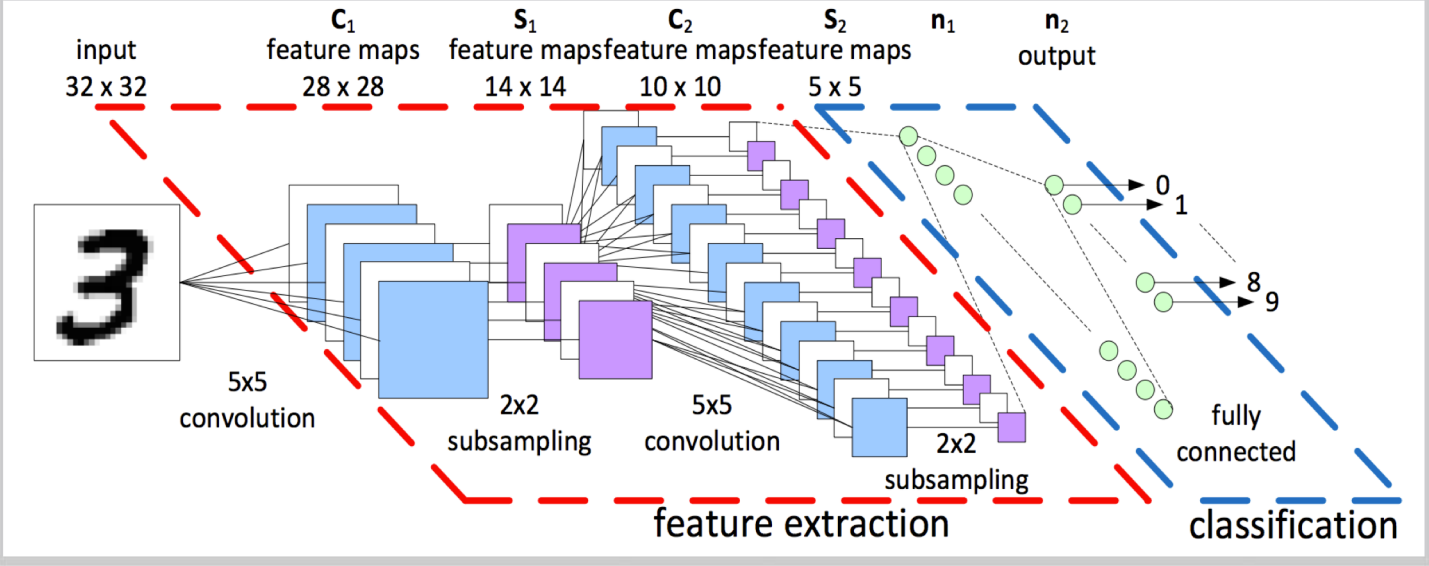
Flatten and Fully Connected (Dense)

- Towards the end of our network, we flatten the feature maps into one long vector
- We feed this vector into a normal ANN to make predictions
- This ANN is a fully connected layer (aka the dense layer)



Example CNN

- When learning about CNNs you'll often see a diagram like this:



Download Face Recognition Data

Several Datasets: https://www.face-rec.org/databases/

https://cswww.essex.ac.uk/mv/allfaces/faces95.html

Code and Sample Data for this course are Available here:
https://drive.google.com/open?id=1D9qB78B5xTnbYgsbuSib3Dx0K7Og5m0v

Enough Talking ..
Let’s do it!


thank you
danke
謝謝
gracias
merci
obrigado
dziękuję
bedankt
teşekkür ederim
ngiyabonga
спасибо
Баярлалаа
рахмат
dank je
tapadh leat
mochchakkeram
go raibh maith agat
dakujem
terima kasih
sukriya
kop khun krap
grazie
arigatô
takk
tack
barka
mersi
spas
welalin
misaotra
matondo
paldies
grazzi
faafetai lava
vinaka
blagodaram
спасибі
kiitos
dankie
dhanyavad
nanni
nandri
hvala
maurruru
köszönöm
gràcie
bayarlalaa
enkosi
akun
dankon
aciu
djiere dieuf
tau
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
dèkuji
sagolun
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
rahmet
tanemirt
ありがとう
diolch
dhanyavadagalu
shukriya
mercè
мерси
trugarez
mahalo
хвала
asante
manana
obrigada
tenki
chokrane
murakoze
mamnun
дякую
ευχαριστώ
xièxie
감사합니다
তোমাকে ধন্যবাদ

About the Speaker

Name: Dr Noureddin Sadawi

Qualification: PhD Computer Science

Experience: Several areas including but not limited to Docker, Machine Learning and Data Science, Python and much more

Text Mining

- “Text mining, also referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text.” -wikipedia
- “Another way to view text data mining is as a process of exploratory data analysis that leads to heretofore unknown information, or to answers for questions for which the answer is not currently known.” - Hearst, 1999

Mining

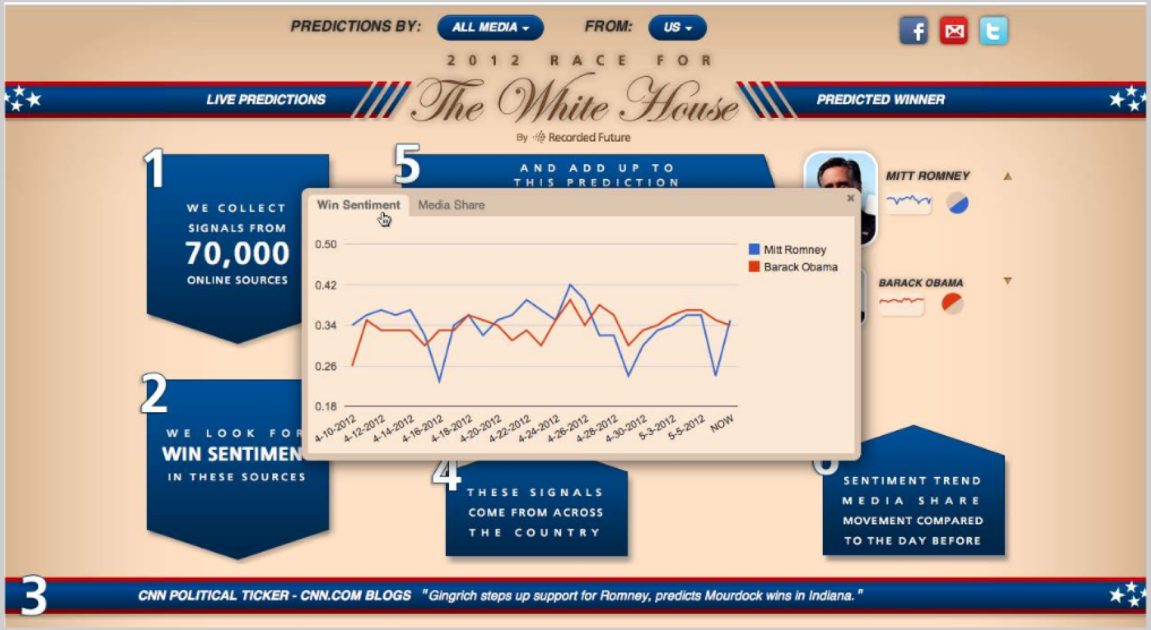
- Goal-oriented (effectiveness driven)
  - Any process that generates useful results that are non-obvious is called “mining”.
  - Keywords: “useful” + “non-obvious”
  - Data isn’t necessarily massive
- Method-oriented (efficiency driven)
  - Any process that involves extracting information from massive data is called “mining”
  - Keywords: “massive” + “pattern”
  - Patterns aren’t necessarily useful

Text mining around us

Sentiment Analysis

Text mining around us

Sentiment Analysis



Text mining around us

Document summarization

Text mining around us

- Movie recommendation
- Restaurant/hotel recommendation
- News recommendation
- Text analytics in financial services
- Text analytics in healthcare

How do we do it?

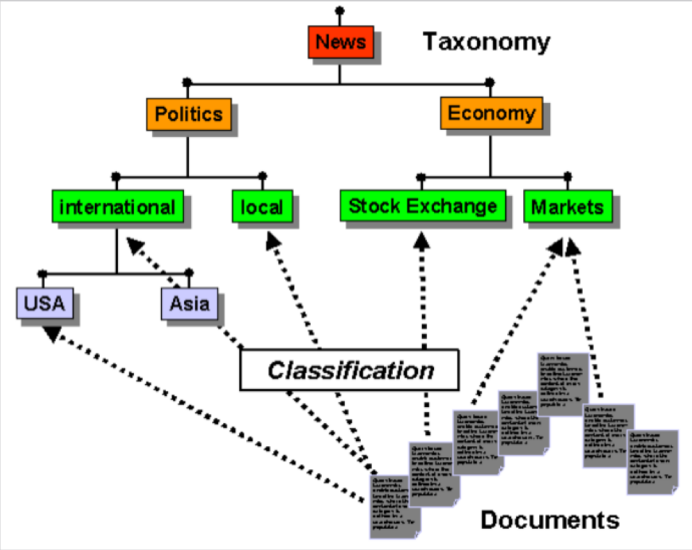
- We view it as a combination of:

Data Mining + Text Data

- Data Mining:
  - Information Retrieval
  - Natural Language Processing
  - Applied Machine Learning
- Text Data:
  - Emails
  - Tweets
  - News Articles
  - Medical Docs
  - Blogs

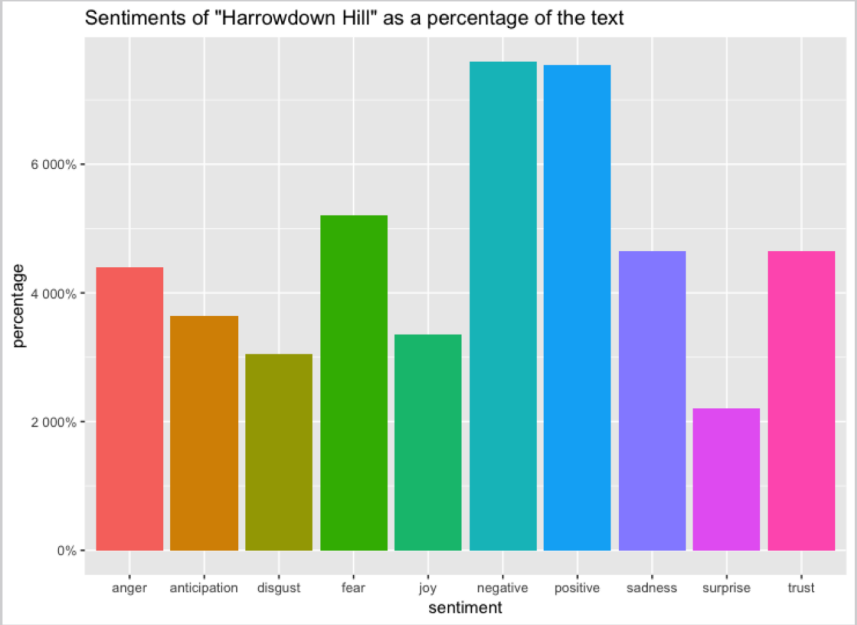
Problems we will deal with

- Document/Text Categorization
- Given a text document, which class does it belong to?



Problems we will deal with

- Social media and social network analysis (Twitter)



Necessary Steps

- To be able to automatically perform the classification task, a preliminary phase of text preprocessing and feature extraction is essential
- We want to transform each text in a vector form, in which each document is represented by the presence (or frequency) of some important terms
- These terms are the ones contained in the *collection vocabulary*
- To build the vocabulary, various operations are typically performed (many of which are language-specific)

Text Classification (Sentiment Analysis)

Text Input

- Data Collection
- Data Organisation (in a table or directory structure)
- Table: Your input can be in the form of a table where each row contains the text in a document and the class/category of this text
- Directory: Your input can be in a directory with subdirectories inside it, each subdirectory contains text documents of the same category, the subdirectory name is the category
  - This needs to be transformed into a table

Word Parsing and Tokenization

- In this phase, each document is analyzed with the purpose of extracting the terms
- Separator characters must be defined, along with a tokenization strategy for particular cases such as accented words, hyphenated words, acronyms, etc
- Tokenization is the term used to describe the process of converting the normal text strings in to a list of tokens (words or sentences that we actually want)

Stopword Removal/Filtering

- A very common technique is the elimination of frequent usage words: conjunctions, prepositions, base verbs, etc
- This kind of terms should be filtered as they have a poor characterizing power, making them useless for the text classification
- Usually we use a predefined list of stopwords

Negation Handling

Lemmatization and Stemming

- The lemmatization of a word is the process of determining its lemma
- The lemma can be thought of as the “common root” of various related inflectional forms
  - for instance, the words *walk*, *walking* and *walked* all derive from the lemma *walk*
- A simple technique for approximated lemmatization is called *stemming*
- Stemming algorithms work by removing the suffix of the word, according to some grammatical rules

Term Selection and Feature Extraction

- The term set resulting from the previous phases has still to be filtered, since we need to remove the terms that have poor prediction ability (w.r.t the document class) or are strongly correlated to other terms
- This term selection task also leads to a simpler and more efficient classification
- A common powerful technique for feature extraction based on word importance is the Term Frequency - Inverse Document Frequency, or TF-IDF for short

Term Frequency

- In TF we just count the number of words occurred in each document
- The main issue with this Term Frequency is that it will give more weight to longer documents
- Term frequency is basically the output of the BoW model
- Essentially it is a frequency table showing how many times each word appears in each document/tweet

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Term Frequency

- This matrix is using a single word
- It can be a combination of two or more words
- It is called a bigram or trigram model and the general approach is called the n-gram model (a tokenization approach)

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Inverse Document Frequency

- IDF measures the amount of information a given word provides across the document
- It is the logarithmically scaled inverse ratio of the number of documents that contain the word and the total number of documents
- https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Recommended tf–idf weighting schemes

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Inverse Document Frequency

- TF-IDF normalizes the document term matrix
- It is the product of TF and IDF
- If word has a high tf-idf in a document, it appears most of the time in this document .. and and much less times in the other documents
- So the word must be a signature word

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

Classification

- This is a supervised learning task
- The idea is to train a classifier on existing labelled data and then assign a class label to unclassified input
- It is important to bear in mind that the same preprocessing steps used to prepare the training data must be applied to the new data
- After this you can use your favourite classifier
- NaiveBayes usually works well in text classification

Example Data

- There are several freely available datasets
- A common one is the 20 Newsgroups Dataset: http://qwone.com/~jason/20Newsgroups/
- Movie Sentiment Data: https://github.com/adamwulf/movie-review-sentiment-data

Course Notebooks and Data

Available here:

https://drive.google.com/open?id=1nF9HkZTlRE1fylyZA4oMOt0Ib_OGcQLz


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
go raibh maith agat
mochchakkeram
tapadh leat
dank je
спасибо
рахмат
Баярлалаа
faafetai lava
vinaka
спасибі
blagodaram
kiitos
dankie
dhanyavad
hvala
mauruuru
köszönöm
bayarlalaa
gračje
enkosi
nanni
nandri
dankon
akun
aĉiŭ
misaotra
matondo
paldies
grazzi
tapadh leat
хвала
asante
manana
obrigada
tenki
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
sukriya
kop khun krap
tạikụ
arigatô
takk
dakujem
trugarez
sobodi
dėkuji
sagolun
mèsi
didi madloba
kam sah hamnida
najis tuke
rahmat
terima kasih
ありがとう
tanemirt
rahmet
grazie
diolch
dhanyavadagalu
shukriya
mercè
мерси
ευχαριστώ
xièxie
감사합니다
তোমাকে ধন্যবাদ
barka
kia ora
welalin
tack
spas
mahalo
chokrane
murakoze
tenki

About the Speaker

Name: Dr Noureddin Sadawi

Qualification: PhD Computer Science

Experience: Several areas including but not limited to Docker, Machine Learning and Data Science, Python and much more

Malicious Software (Malware)

- Network attacks prevent a business from operating
- Malicious software (Malware) includes:
  - Virus
  - Worms
  - Trojan horses
- Goals:
  - Destroy data
  - Corrupt data
  - Shutdown a network or system

Viruses

- Virus attaches itself to an executable file
- Can replicate itself through an executable program:
  - Needs a host program to replicate
- No foolproof method of preventing them

Antivirus Software

- Detects and removes viruses
- Detection based on virus signatures
- Must update signature database periodically
- Use automatic update feature

Ransomware

- Encrypts files, demands ransom for the key
- Doesn't need to be reported as a breach, because no data was stolen

JANUARY 10, 2017, 9:35 AM

A community college in the San Fernando Valley has become one of the latest institutions to pay ransom to hackers who took control of its computer system.

Los Angeles Valley College in Valley Glen said it paid $28,000 in bitcoins to the hackers, who had used malicious software to commandeer a variety of systems, including key computers and emails.

Worms

- Worm:
  - Replicates and propagates without a host, often through email
- Infamous examples:
  - Code Red
  - Nimda
- Can infect every computer in the world in a short time:
  - At least in theory

Trojan Programs

- Insidious attack against networks
- Disguise themselves as useful programs
  - Hide malicious content in program
    - Backdoors
    - Rootkits
  - Allow attackers remote access

Firewalls

- Identify traffic on uncommon ports
- Can block this type of attack, if your firewall filters outgoing traffic:
  - Windows Firewall in XP SP2, Vista, and Win 7 does not filter outgoing traffic by default
- Trojan programs can use known ports to get through firewalls
  - HTTP (TCP 80) or DNS (UDP 53)

Spyware

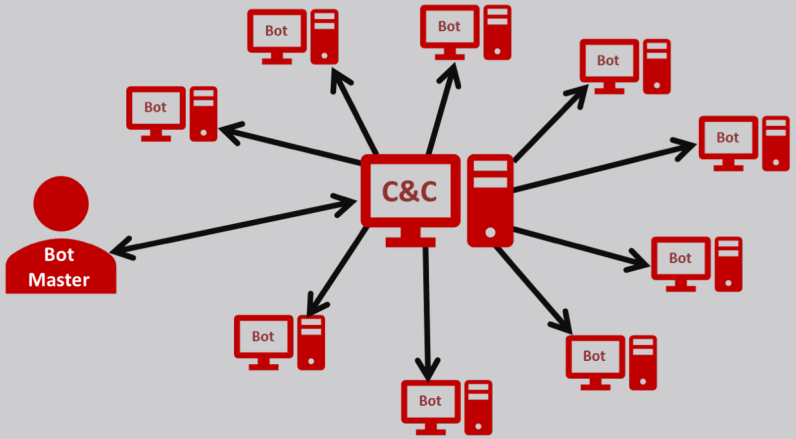
- Sends information from the infected computer to the attacker
  - Confidential financial data
  - Passwords
  - PINs
  - Any other stored data
- Can register each keystroke entered (keylogger)
- Prevalent technology

Adware

- Similar to spyware
  - Can be installed without the user being aware
- Sometimes displays a banner
- Main goals:
  - Determine user's online purchasing habits
  - Tailored advertisement
- Main problem
  - Slows down computers

Botnets

- A Botnet is a network of Internet devices (i.e. computers, mobile etc) that have been manipulated to carry out malicious activities.
- The owners of such devices are often unaware of this!
- These Bots serve the wishes of some master known as Command & Control



Botnet Examples

- Neris | IRC
- Rbot | IRC
- Menti | IRC
- Sogou | HTTP
- Murlo | IRC
- Virut | HTTP
- NSIS | P2P
- Zeus | P2P
- SMTP Spam | P2P
- UDP Storm | P2P
- Tbot | IRC |
- Zero Access | P2P
- Weasel | P2P
- Smoke Bot | P2P
- Zeus Control (C&C) | P2P

Obtaining Data

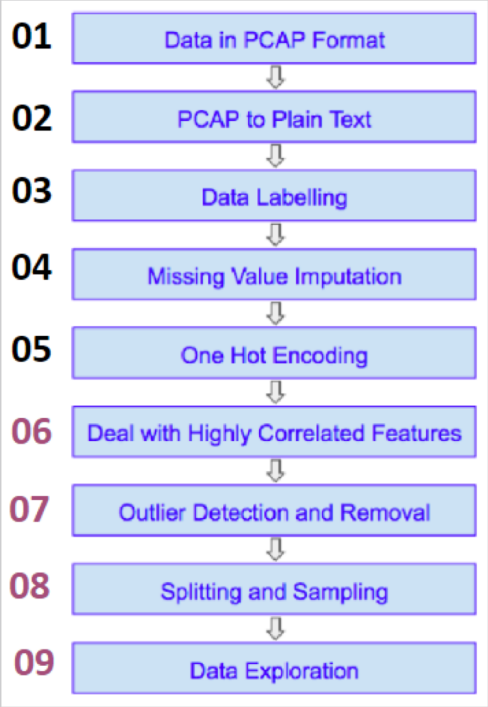
- Computer Network data can be obtained by using capture tools such as WireShark, Snort, tcpdump … etc
- Also there are freely available datasets (for research purposes)
  - Some useful examples are here (labelling info is provided): https://www.unb.ca/cic/datasets/botnet.html
- The data is normally in binary format (PCAP for Packet Capture)
- In order to transform this data into a format understandable by machine learning tool, several steps should be performed

Data Preparation (PCAP -> CSV)

- Convert PCAP Data into a suitable format for Machine Learning (CSV)
- Use open source Java tool (works best on Ubuntu Linux):
  - https://github.com/ahlashkari/ISCXFlowMeter
  - Requires JNetPcap Package https://sourceforge.net/projects/jnetpcap/
- This tool generates several numeric attributes (features) such as Source Port, Destination Port, Protocol, Flow Duration, Flow Bytes per second and Flow Packets per second
- Data needs to be somehow labelled (Domain knowledge is required .. ISCX provides labelling info)

How?

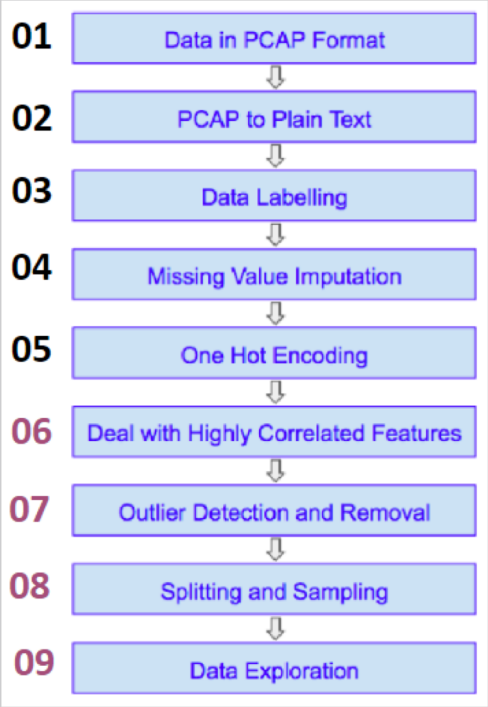
Pre-processing 1/3



1. Raw Network Traffic Data can be obtained via capture tools such as WireShark (usually in PCAP format)
2. PCAP format can be transformed into CSV using tools such as FlowMeter (generates several useful features)
3. Resulting CSV data should be labelled (e.g. when generating training data)
4. Data should be checked for missing values, if any exist, these values should be imputed (several techniques)

Points in purple are optional

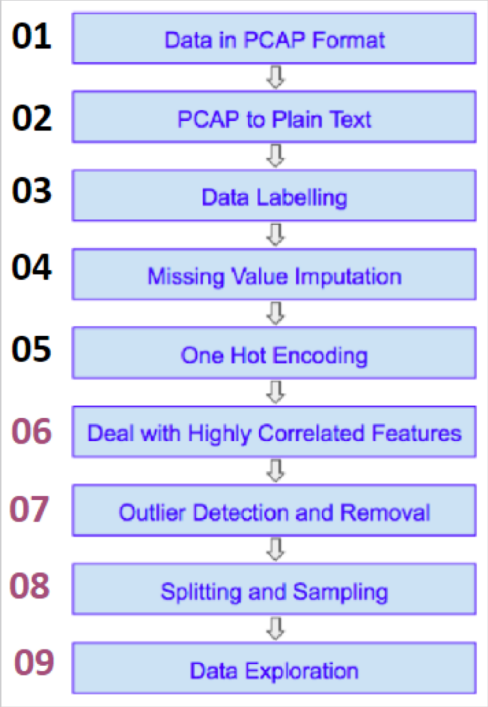
Pre-processing 2/3



5. Sometimes categorical features are represented numerically, this is not recommended and one-hot encoding can be used to represent such data correctly
6. Some of the features in the generated data can be highly correlated, this case must be dealt with appropriately to achieve reliable results
7. Data should be checked for outliers, if any exist, they should be dealt with appropriately (depends on the purpose of analysis).

Points in purple are optional

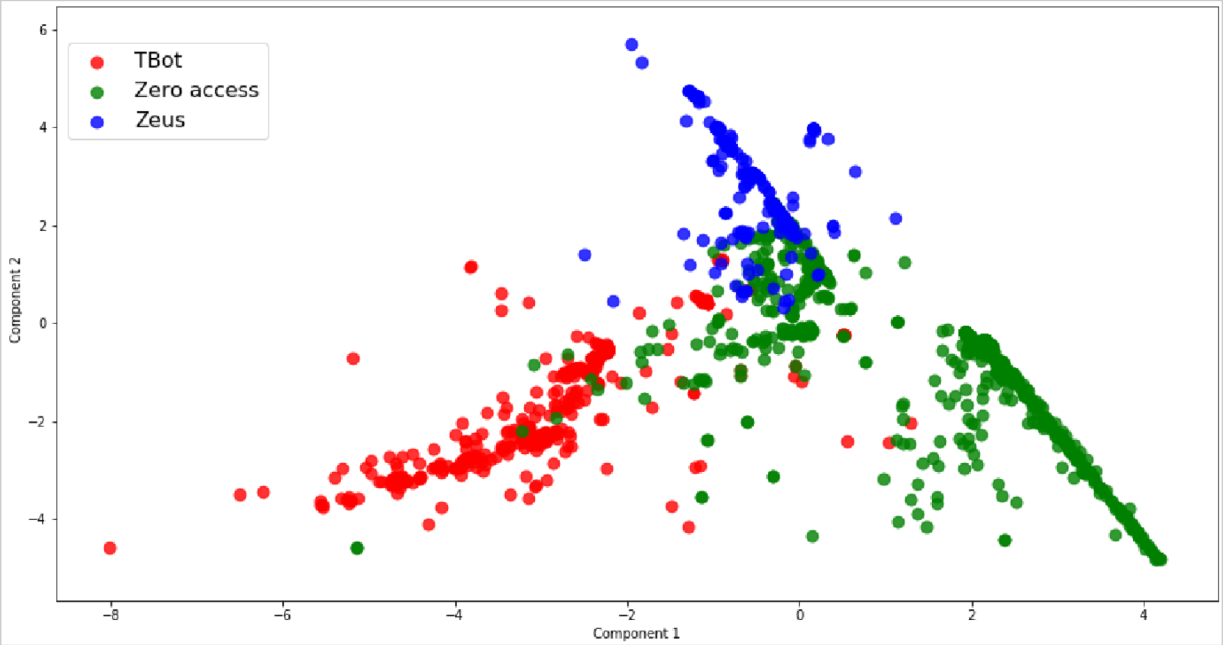
Pre-processing 3/3



8. If the data contains multiple categories (e.g. Normal, DDoS, Worm … etc), it might be useful to have a separate dataset for each category (depends on the purpose of analysis)
9. Data exploration techniques can be used to inspect the data distribution

Points in purple are optional

Data Visualization



- Use PCA or PLS to explore data

Course Notebooks and Data

Available here:
https://drive.google.com/open?id=1PpqJ9WfXVB5kYASVHMBY-8HlEDQYXPpm

Enough Talking ..
Let’s do it!


рахмат
danke
謝謝
ngiyabonga
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
blagodaram
dank je
misaotra
paldies
grazzi
kiitos
dankie
спасибі
matondo
gracias
хвала
nanni
nandri
dhanyavad
thank you
asante
manana
bedankt
hvala
maururu
köszönöm
akun
dankon
aĉiŭ
obrigada
tenki
bayarlalaa
gràçie
murakoze
enkosi
djiere dieuf
tau
mochchakkeram
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
chokrane
go raibh maith agat
taiku
sobodi
děkuji
sukriya
kop khun krap
sagolun
obrigado
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
ありがとう
tanemirt
grazie
najis tuke
terima kasih
rahmet
kam sah hamnida
rahmat
diolch
dhanyavadagalu
shukriya
mercê
мерси
xièxie
তোমাকে ধন্যবাদ
감사합니다
ευχαριστώ
merci

About the Speaker

Name: Dr Noureddin Sadawi

Qualification: PhD Computer Science

Experience: Several areas including but not limited to Docker, Machine Learning and Data Science, Python and much more

Definition

- Time Series: *An ordered sequence of values of a variable at equally spaced time intervals*
- The essential difference between modeling data via time series methods or using other methods is:
  - *Time series analysis accounts (or must account) for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for*
  - *If we change the order of points, we have a different signal (unlike many other data types)*

https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm

Applications

The usage of time series models is twofold:

- Obtain an understanding of the underlying forces and structure that produced the observed data
- Fit a model and proceed to forecasting, monitoring or even feedback and feedforward control

Example Application Areas

- Economic Forecasting
- Sales Forecasting
- Budgetary Analysis
- Stock Market Analysis
- Yield Projections
- Process and Quality Control
- Inventory Studies
- Website Traffic
- Competition Position
- Demand of products
- Workload Projections
- Utility Studies
- Census Analysis

Valuable Skill

Being able to deal with time series data is not a very common skill in the data science space

What makes Time Series special?

- A Time Seris is a collection of data points collected at **constant time intervals**
- These are analyzed to determine the long term trend so as to forecast the future or perform some other form of analysis
- It is a regression problem as we are trying to predict a real value

What makes Time Series special?

- There are 2 things that make a TS different from a regular regression problem:
  - It is **time dependent**. So the basic assumption of a linear regression model that the observations are independent doesn't hold in this case
  - Most time series have some form of **seasonality trends** (increasing or decreasing), i.e. variations specific to a particular time frame (e.g. if you see the sales of ice cream, you will invariably find higher sales in the summer)

Techniques

- A number of techniques for the modeling and analysis of time series data exist
- The user's application and preference will decide the selection of the appropriate technique.
- It is beyond the realm and intention of this training to cover all of them
- This training will look:
  - Averaging Methods
  - Exponential Smoothing Techniques
  - A deep learning method for modelling and predictions

Smoothing Techniques

- Inherent in the collection of data taken over time is some form of random variation
- There exist methods for reducing of canceling the effect due to random variation
- An often-used technique in industry is "smoothing"
- This technique, when properly applied, reveals more clearly the underlying trend, seasonal and cyclic components
- There are two distinct groups of smoothing methods:
  - Averaging Methods
  - Exponential Smoothing Methods

Simple Average

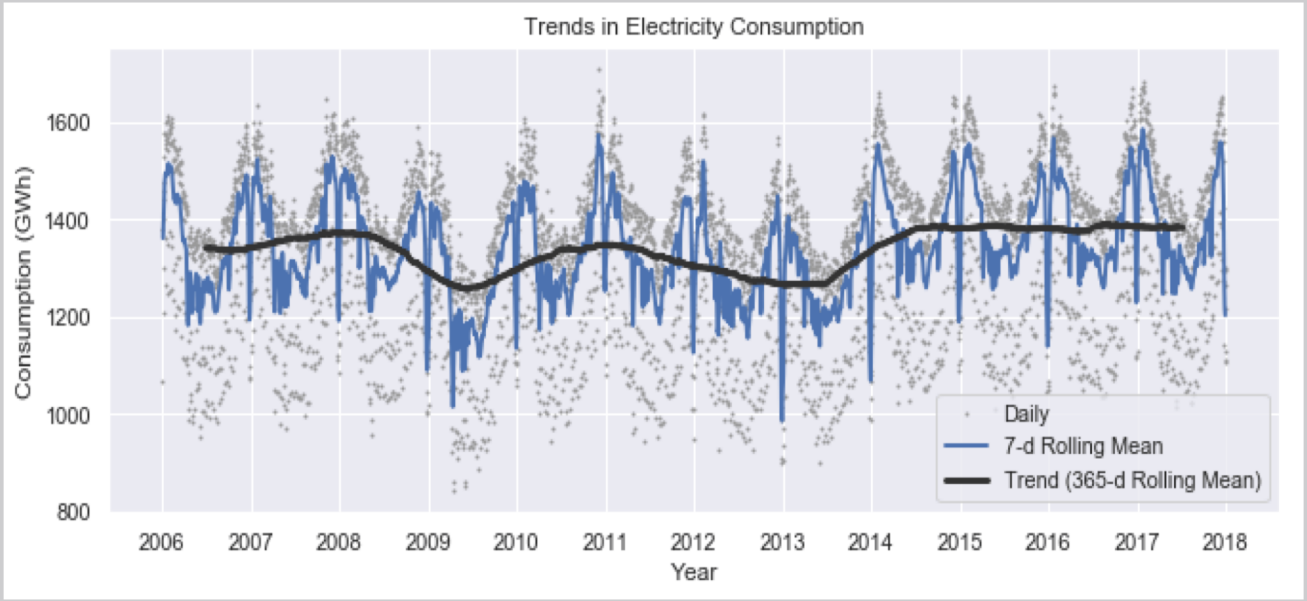
- This method says we compute the simple average (i.e. the mean) of all past data and use it to forecast future data
- The "simple" average or mean of all past observations is only a useful estimate for forecasting when there are no trends
    - If there are trends, a different estimates that takes the trend into account should be used
- The simple average "weighs" all past observations equally
- Perhaps a better method is to use a weighted average where observations are given weights according to their importance
    - For example, most recent observations have higher weights than older observations

Simple Moving Average

- This is a variation of the previous method (MA)
- In this approach, we take average of ‘k’ consecutive values depending on the frequency of time series (i.e. compute the mean of successive smaller sets of values of past data)
- AKA: *Rolling Mean*
- This is called "smoothing" (i.e., some form of averaging)
- This smoothing process is continued by advancing one period and calculating the next average of ‘k’ values, dropping the first value

https://pandas.pydata.org/pandas-docs/stable/user_guide/computation.html#window-functions

Example


Trends in Electricity Consumption
Consumption (GWh)
1600
1400
1200
1000
800
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Year
Daily
7-d Rolling Mean
Trend (365-d Rolling Mean)

Exponential Smoothing

- This is a very popular scheme to produce a smoothed Time Series
- Whereas in Single Moving Averages the past observations are weighted equally, Exponential Smoothing assigns *exponentially decreasing weights* as the observation get older
    - i.e. weights are assigned to all the previous values with a decay factor
- In other words, *recent observations are given relatively more weight in forecasting than the older observations*

https://pandas.pydata.org/pandas-docs/stable/user_guide/computation.html#exponentially-weighted-windows

Univariate vs Multivariate Data

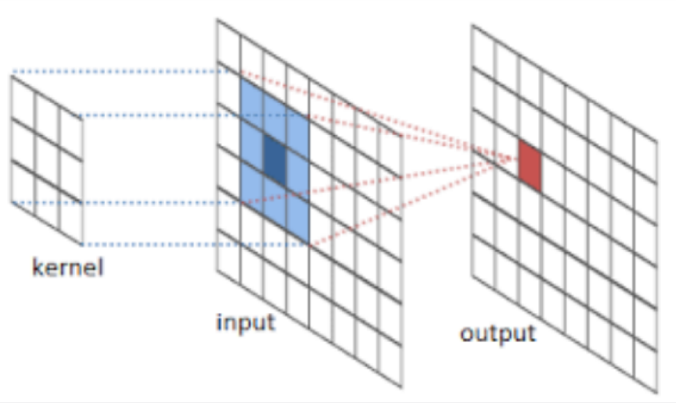
- Univariate time series are datasets comprised of a single series of observations with a temporal ordering and a model is required to learn from the series of past observations to predict the next value in the sequence
- Multivariate time series data means data where there is more than one observation for each time step
  - There can be one output for all of them (AKA Multiple Input Series)
  - There can be one output for each of them (AKA Multiple Parallel Series)
- This training will only cover Univariate time series

Modeling and Future Predictions

- In this training we are going to use the following deep learning techniques:
  a. Convolutional neural networks (CNNs)
  b. Recurrent neural networks (RNNs)
  c. Long short-term memory networks (LSTMs)
  d. Gated recurrent units (GRUs)

CNNs in One Slide

- CNNs are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers



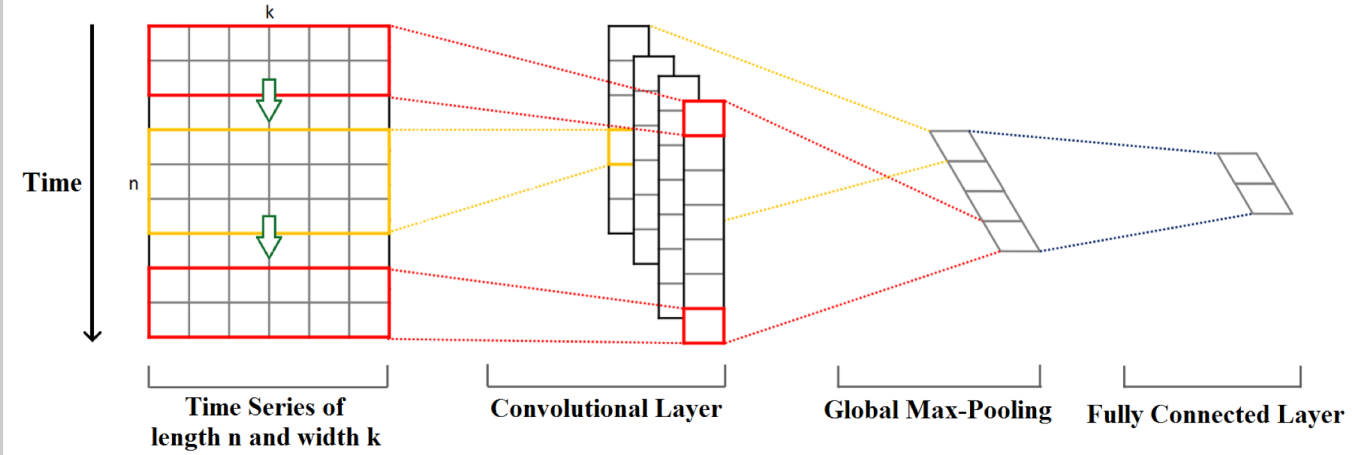
- Convolution is a mathematical operation having a linear form

Types of Inputs

Inputs have a structure:

- Color images are three dimensional and so have a volume
- Time domain speech signals are 1-d while the frequency domain representations (e.g. MFCC vectors) take a 2d form
  - They can also be looked at as a time sequence!
- Medical images (such as CT/MR/etc) are multidimensional
- Videos have the additional temporal dimension compared to stationary images
- Speech signals can be modelled as 2 dimensional
- Variable length sequences and time series data are again multidimensional

1D CNN

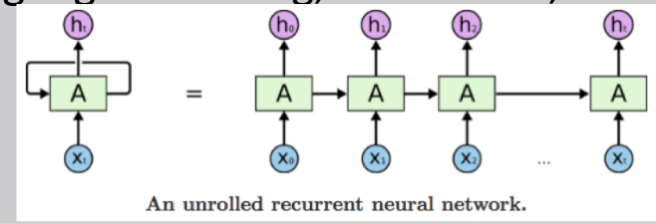

k
Time
n
Time Series of
length n and width k
Convolutional Layer
Global Max-Pooling
Fully Connected Layer

Essential Concepts in Deep Learning

- **Pooling:** subsamples the input signal, which reduces the memory use and computer load as well as reducing the number of parameters
- **Padding:** pad the input volume with values (e.g. zeros) around the border to preserve as much information about the original input volume so that we can extract those low level features
- **Dropout:** during training, units are randomly dropped, along with their connections (this helps prevent units from “co-adapting” too much .. it can be thought of as a form of regularization to help prevent overfitting)

RNNs

- RNNs are networks with loops in them, allowing **information to persist**
- RNNs can be thought of as multiple copies of the same network, each passing a message to a successor
- This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists
- There have been incredible success applying RNNs to a variety of problems: speech recognition, language modeling, translation, image captioning and so on



An unrolled recurrent neural network.

LSTMs

- RNN Issue: when input sequence is long, the data from early time steps have little impact on the output (Vanishing Gradient)
- LSTMs are Units of recurrent neural networks (RNNs)
- Remember values over arbitrary time intervals
- LSTMs are explicitly designed to avoid the long-term dependency problem
- Well-suited to classifying, processing and making predictions based on time series data
- But .. require plenty of data

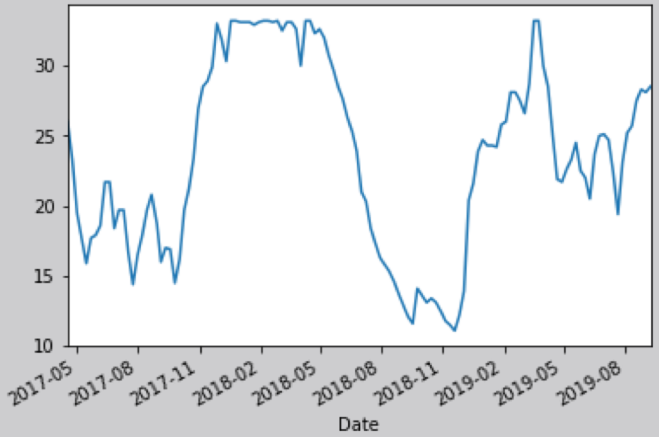
GRUs

- LSTMs work well but they are complicated (lots of parameters)
- GRU is a variant of LSTM
- Approach:
  - Combine the forgetting gate and input gate in LSTM into a single "Update Gate"
  - Combine the Cell State and Hidden State
- Computationally less expensive
  - Less parameters, less complex structure
- Performance is as good as LSTM

Data Preparation .. Very Important

- Before a time series data can be modeled using one of the previous techniques, it must be prepared!
- The models will learn a function that maps a sequence of past observations as input to an output observation
- As such, the sequence of observations must be transformed into multiple examples from which the model can learn
- The idea here is to divide the sequence into multiple input/output patterns called samples, where ‘n’ time steps are used as input and one time step is used as output for the one-step prediction that is being learned!

Data Preparation Example



- Normalize the Data into values between 0 and 1
- Input Seq (5 time steps) [0.67, 0.55, 0.38, 0.29, 0.21] -> Output 0.29
- Input Seq (5 time steps) [0.55, 0.38, 0.29, 0.21, 0.29] -> Output 0.30
- Input Seq (5 time steps) [0.38, 0.29, 0.21, 0.29, 0.30] -> Output 0.00

Download Stock Market Data

Several Methods:

- Use Pandas DataReader https://pandas-datareader.readthedocs.io/en/latest/index.html
- Use Yahoo Finance: https://pypi.org/project/yfinance/
- Use SimFin: https://github.com/simfin/simfin-tutorials/

Course Notebooks and Data

Available here:
https://drive.google.com/open?id=1nZGLTrDFMqSSAS4KGRL9fg2xvqQF_Fg9

Enough Talking ..
Let’s do it!


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
sukriya
kop khun krap
go raibh maith agat
mochchakkeram
dakujem
arigatô
grazie
terima kasih
sagolun
hvala
maurruru
tapadh leat
dank je
спасибо
рахмат
Баярлалаа
mersi
barka
welalin
tack
spas
faafetai lava
vinaka
спасибі
blagodaram
misaotra
matondo
paldies
grazzi
kiitos
dankie
dhanyavad
nanni
nandri
köszönöm
gràcie
bayarlalaa
enkosi
akun
dankon
aèiù
djiere dieuf
tau
хвала
asante
manana
obrigada
tenki
mamnun
дякую
chnorakaloutioun
gratias ago
gràcies
sulpáy
tack
takk
sobodi
dèkuji
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
ありがとう
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
merci
мерси
trugarez
chokrane murakoze
xièxie
ευχαριστώ