EEP 596: Adv Intro ML || Lecture 13

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 16, 2023

Logistics

Arrhythmia Detection

1. Please pick a team-mate for your project
2. Checkpoint submission for mini-project - Early submission to stay on track!
3. Anything else?

Kaggle Contest

spreadsheet → Teams of 2

Last Time

1. Anomaly Detection Baselines: SMA and EMA

Last Time

1. Anomaly Detection Baselines: SMA and EMA
2. Anomaly Detection: STL

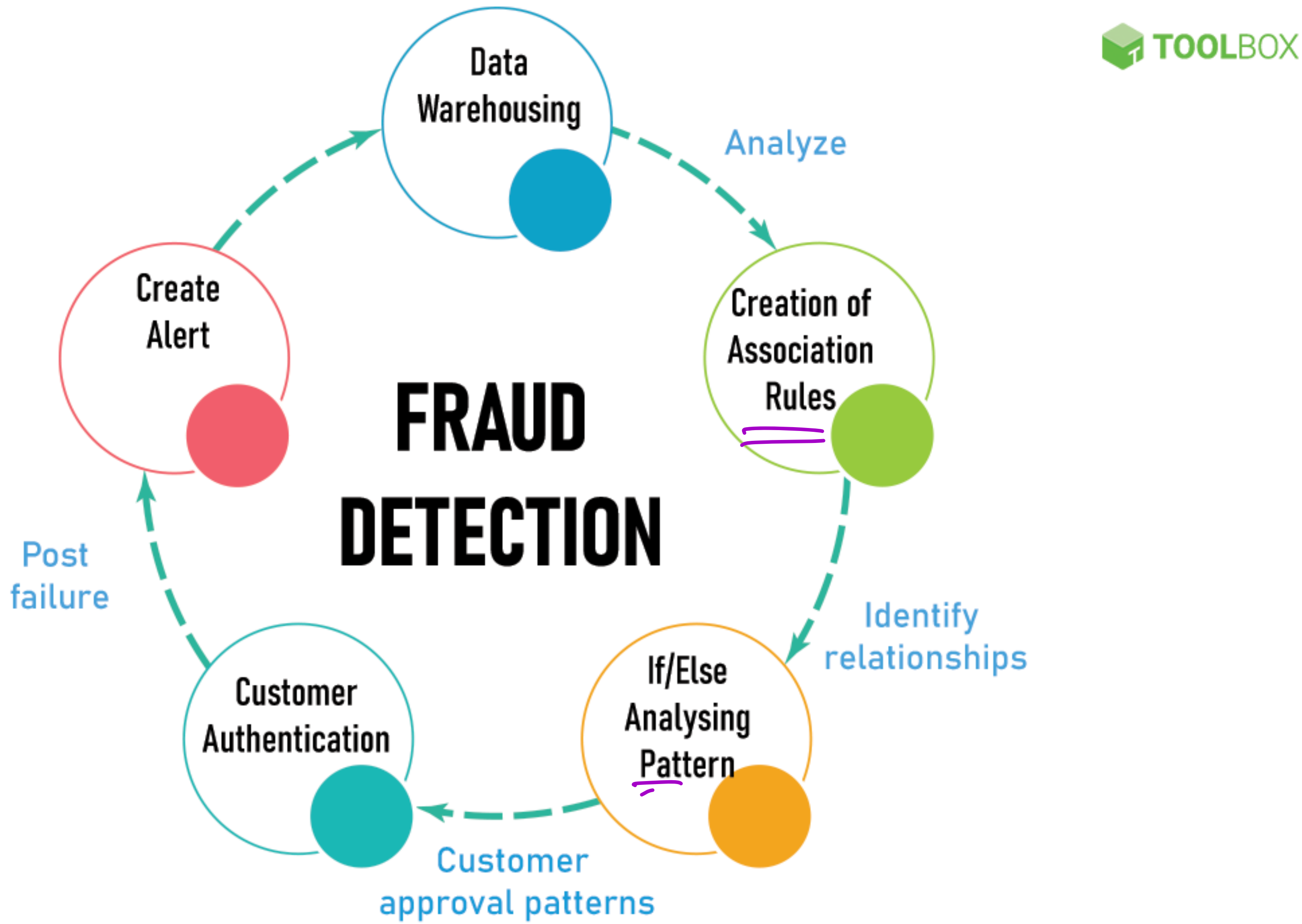
Last Time

a. Anomaly Detection Baselines: SMA and EMA
b. Anomaly Detection: STL
c. Anomaly Detection on Alpaca

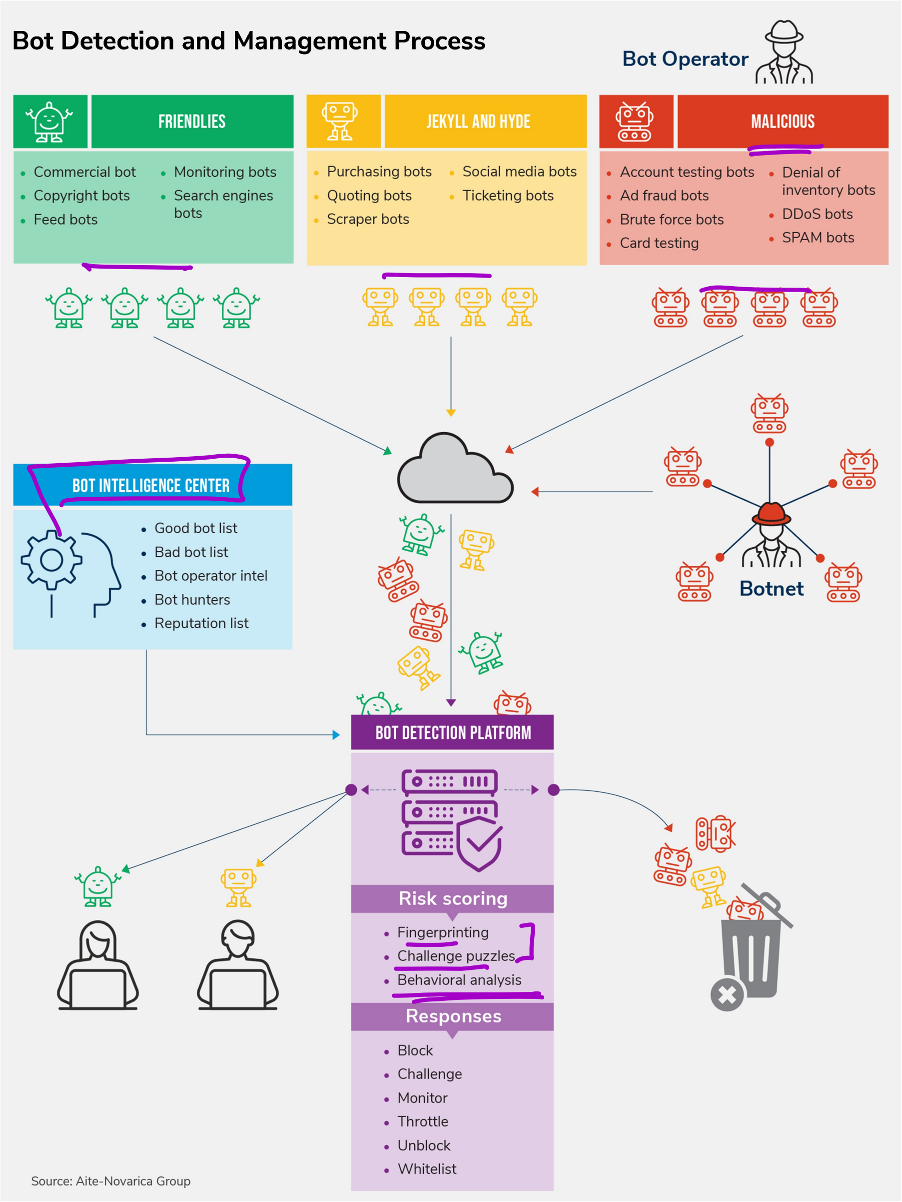
Today

- Anomaly Detection Recap
- Time-series methods for Anomaly Detection
- Introduction to Deep Learning
- Deep Learning Applications
- Deep Learning Theory
- Deep Learning Modeling

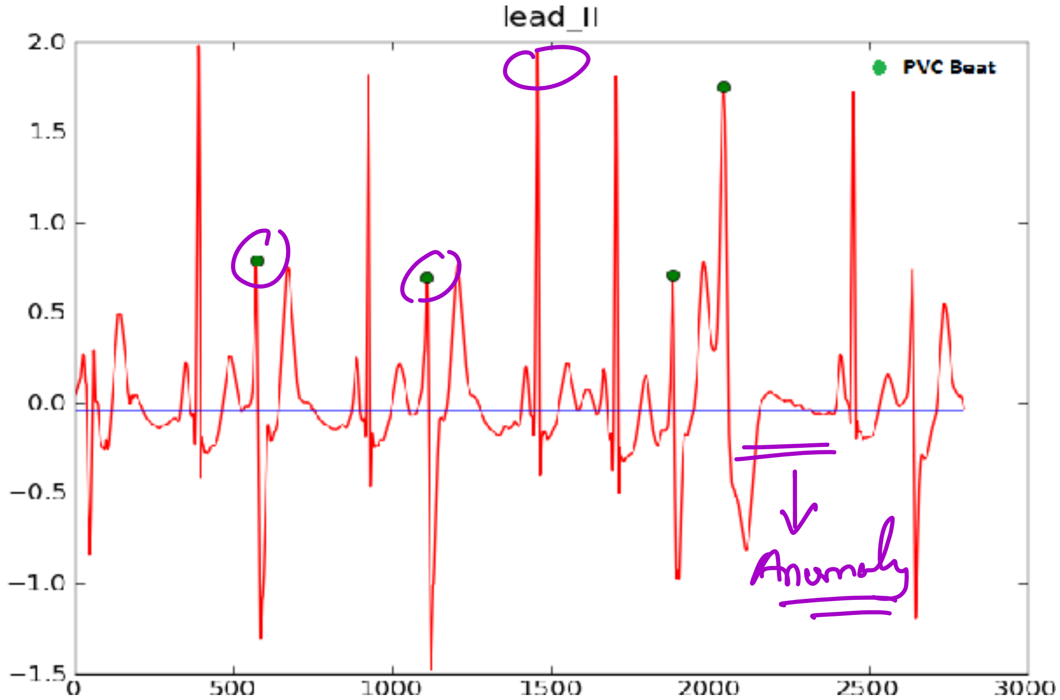
Fraud Detection



Got Bot?

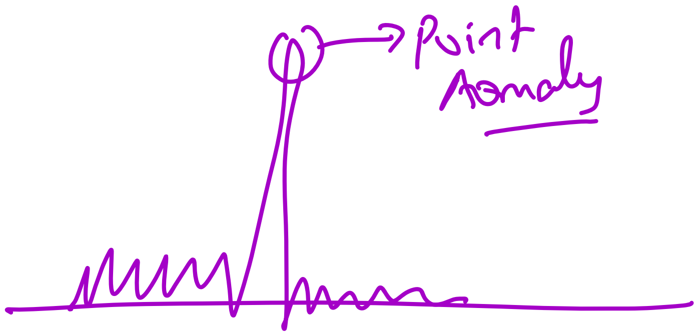

Bot Detection and Management Process
Bot Operator
FRIENDLIES
Commercial bot
Copyright bots
Feed bots
Monitoring bots
Search engines bots
JEKYLL AND HYDE
Purchasing bots
Quoting bots
Scraper bots
Social media bots
Ticketing bots
MALICIOUS
Account testing bots
Ad fraud bots
Brute force bots
Card testing
Denial of inventory bots
DDoS bots
SPAM bots
BOT INTELLIGENCE CENTER
Good bot list
Bad bot list
Bot operator intel
Bot hunters
Reputation list
Botnet
BOT DETECTION PLATFORM
Risk scoring
Fingerprinting
Challenge puzzles
Behavioral analysis
Responses
Block
Challenge
Monitor
Throttle
Unblock
Whitelist
Source: Aite-Novarica Group

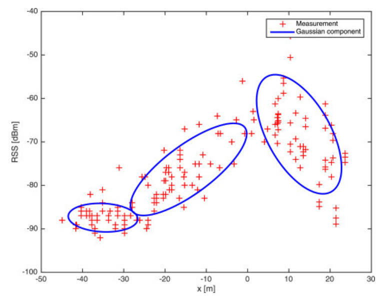
Arrhythmia Detection



Types of Anomalies

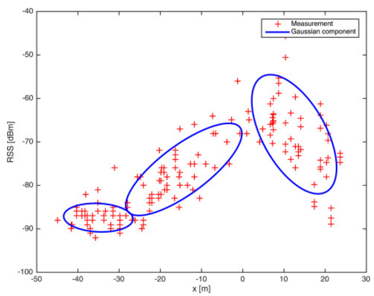
1. **Point Anomaly:** Deviation from a set of data points.



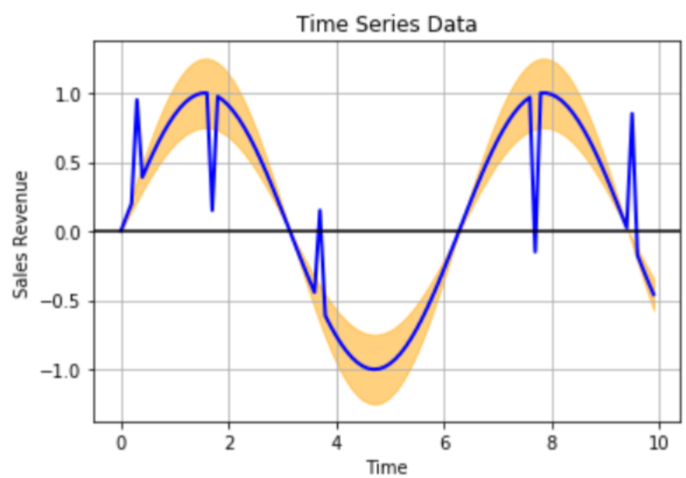


Types of Anomalies

1. **Point Anomaly:** Deviation from a set of data points.

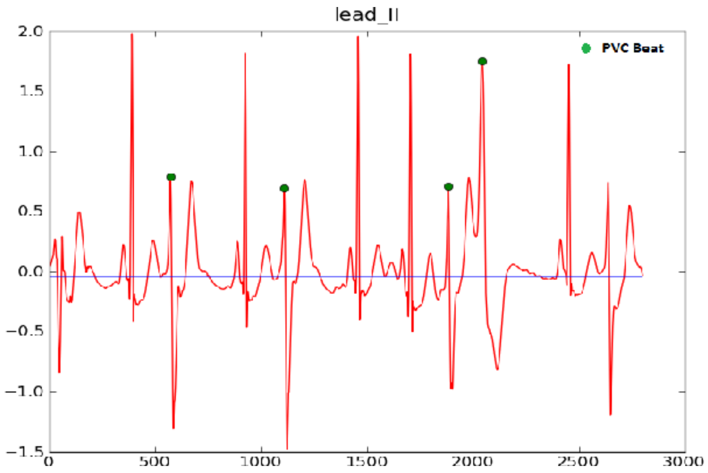


2. **Contextual Anomaly:** Depending on the context, a data point could be an anomaly or not. For instance 35 degrees is not an anomalous temperature for Seattle winter but it is for Seattle summer. Same is true for anomalies in a time-series data e.g. Sales Revenue data.

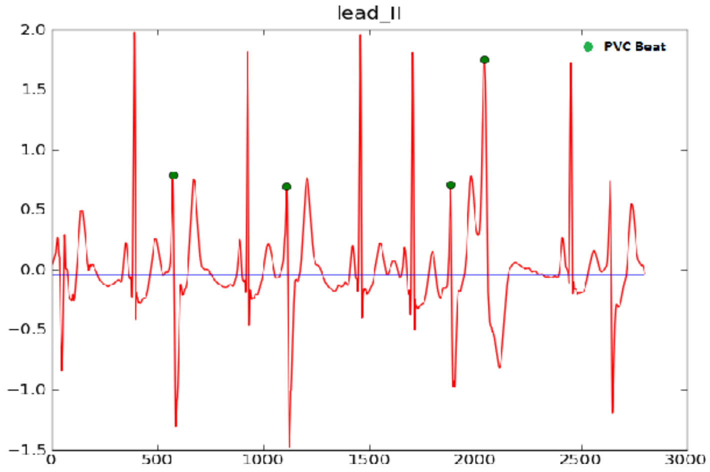


Types of Anomalies

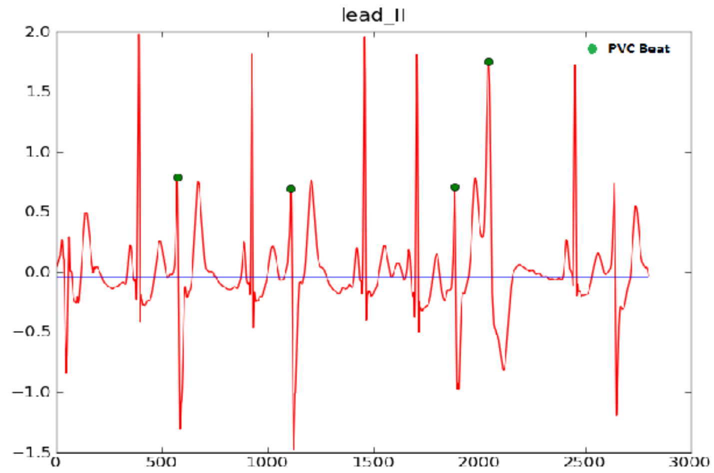
3. **Collective Anomalies:** No one data point is anomalous but a collection of them become anomalous. E.g. the Arrhythmia time series.



Arrythmia detection



Arrythmia detection



Next Assignment: Automated Arrhythmia Detection

You want to build an automated algorithm for Arrhythmia detection from time-series data on heart beats. What would be a baseline un-supervised learning algorithm you can think of Arrhythmia detection? If you wanted to do supervised learning for arrhythmia detection, what features would you use? How would you cast it as a machine learning problem? How would you evaluate the performance of your automated algorithm? What would be the metrics you would use? Discuss in groups - We will implement this as part of the next programming assignment.

Deep Learning for Anomaly Detection

Deep Learning

Deep Learning can provide powerful non-linear supervised models for anomaly detection provided there is enough data (both positive and negative examples) and we account for over-fitting. On the upcoming assignment, can also try out deep learning!

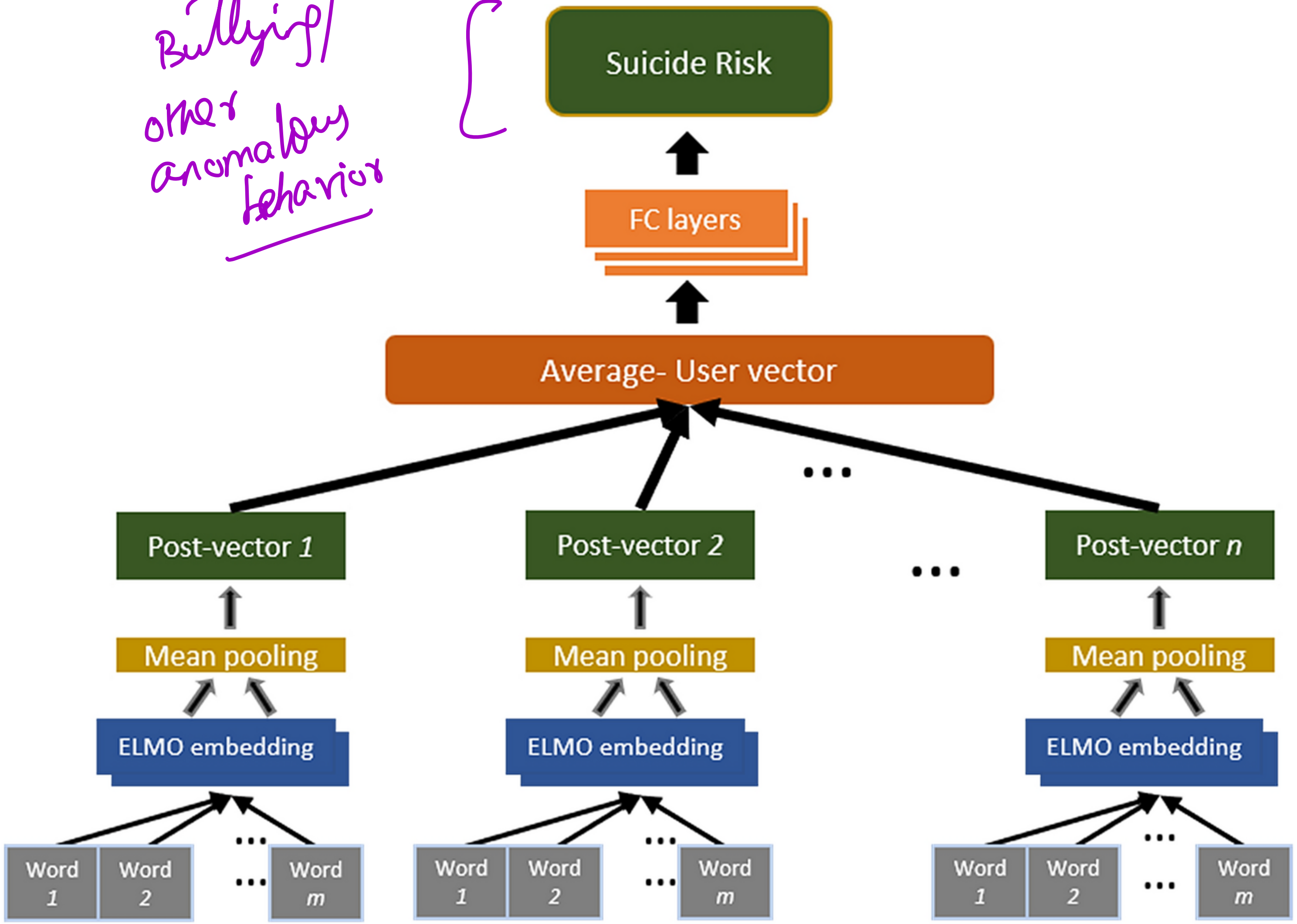
Detecting bots/in-appropriate posts

Biasing the metrics

Do you bias more towards high precision or high recall? Is there a middle ground? Can we have higher recall (i.e. detect the bots/in-appopriate posts) without pissing people off with incorrect flags?

Suicide detection from Social Media posts

Bullying/
other anomalous
behavior



Suicide Detection Reference

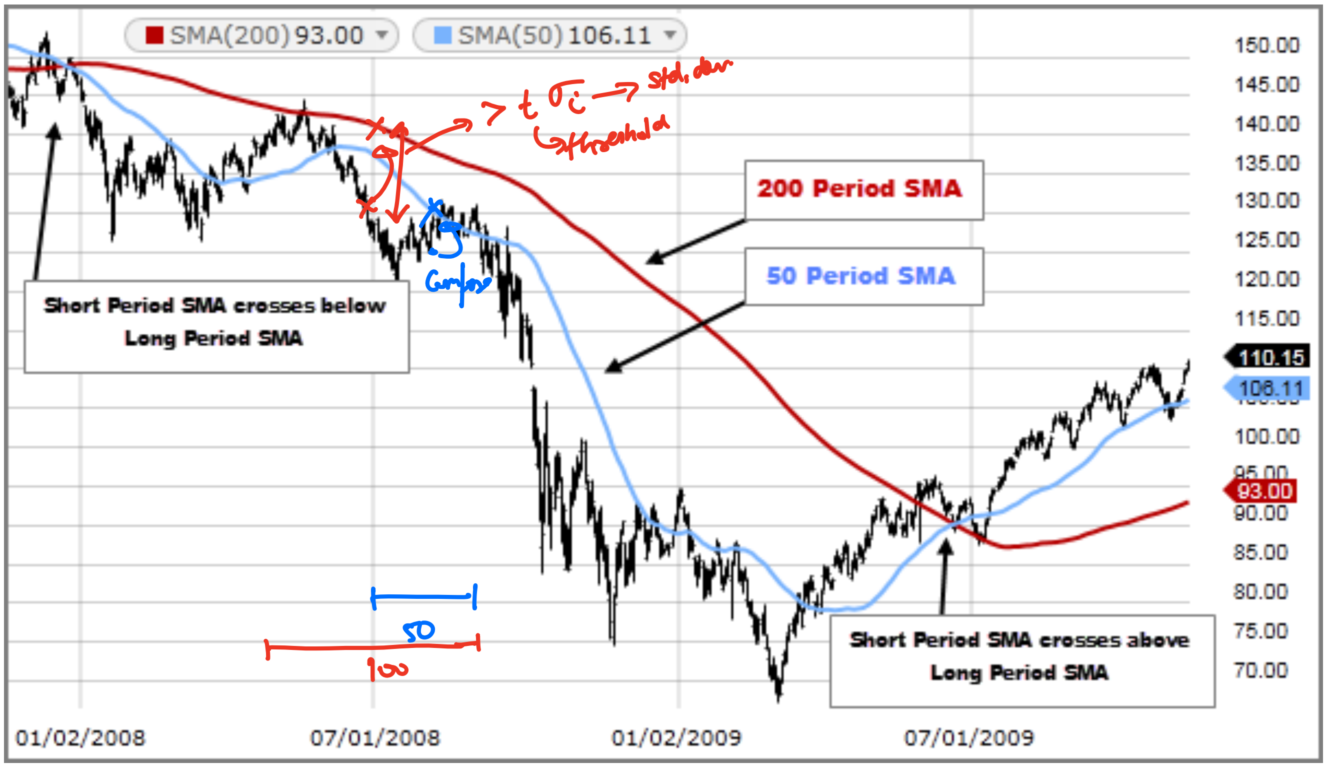
Anomaly Detection Methods so far

	Method	Pros	Cons
1	Mean/Std Deviation	Identifies some anomalies	False positives
2	Supervised Learning	Precise detection	As good as features

Anomaly Detection Methods - Coming up

	Method	Pros
1	Mean/Std Deviation	Identifies some anomalies
2	Supervised Learning	Precise detection
3	**Simple Moving Average (SMA)**	Improves on mean/std deviat
4	**Exponential Moving Average (EMA)**	More sensitive then SMA
5	**STL**	Accounts for seasonality

Moving Averages - Simple Moving Average



Quantity

Stock price

Dates

$x_t = [\quad] \in \mathbb{R}^d$

At time t, have a d-dimensional datapoint

Simple Moving Average and Anomalies

SMA

1. There is a window size 50? that helps you track the **moving** average.
2. 50-SMA is a 50 day moving average
3. 50-SMA$(i) = \frac{1}{50}\sum_{j=i-50}^{i-1} x_j$

Simple Moving Average and Anomalies

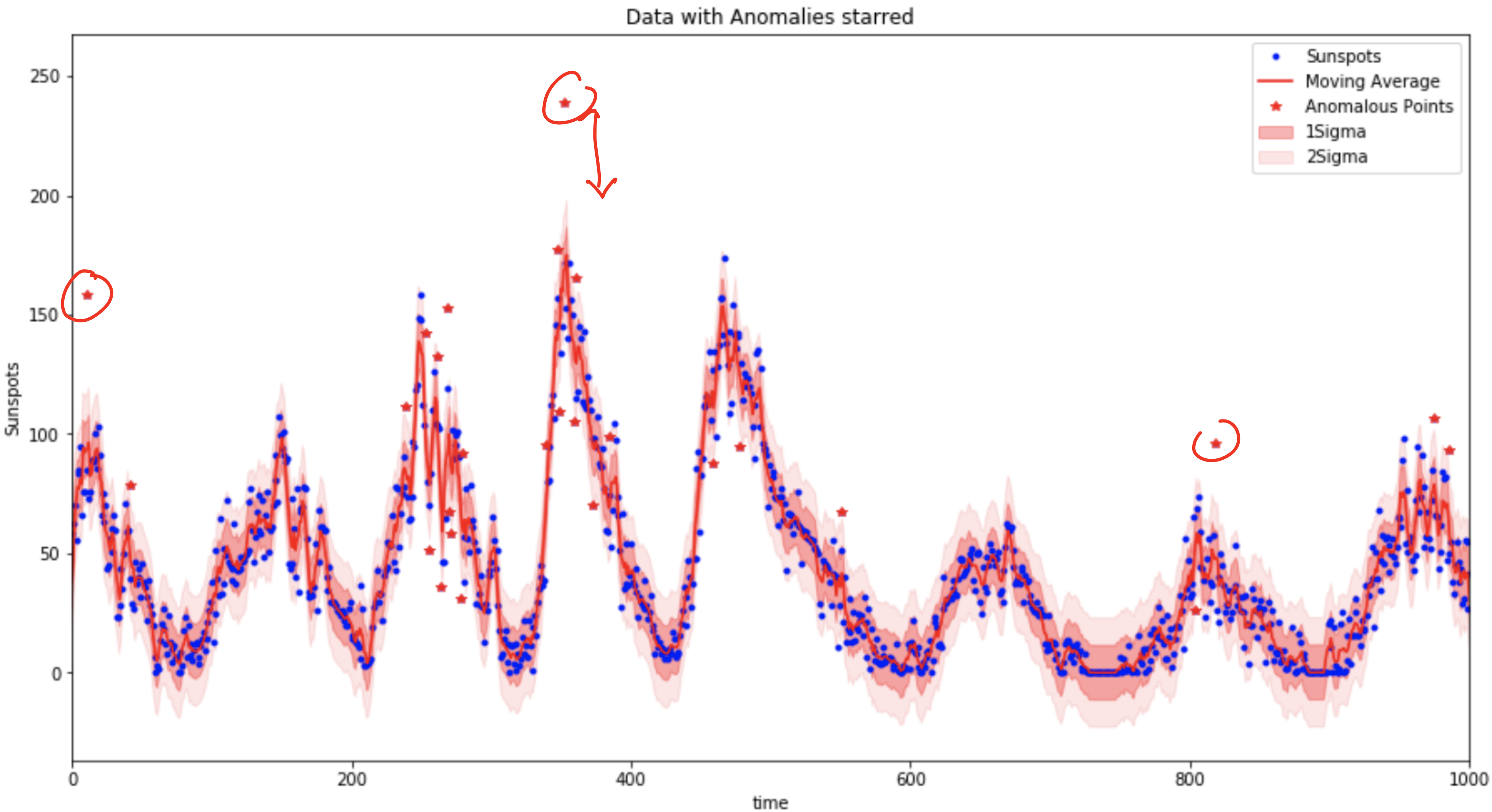
SMA

1. There is a window size that helps you track the **moving** average.
2. 50-SMA is a 50 day moving average
3. $\text{50-SMA}(i) = \frac{1}{50} \sum_{j=i-50}^{i-1} x_j$

Anomaly detection

1. x_i is an anomaly if $\|SMA(i) - x_i\|$ deviates above a $t \times SD(i)$ where $SD(i)$ is the standard deviation and N is the size of the window, t is the threshold.

SMA example



Github Library to try!

Moving mean computation

Let's say you wanted to implement SMA yourself. Let the window size be 100. You have $SMA(i-1)$. How do you compute it from $SMA(i-1)$?

1. $SMA(i) = SMA(i-1) + (x_i - x_{i-\cancel{1}N})/N$
2. $SMA(i) = SMA(i-1) + x_i/N$
3. $SMA(i) = SMA(i-1)$
4. $SMA(i) = SMA(i-1) - x_{i-\cancel{1}N}/N$

$$SMA(i) = \frac{x_i + x_{i-1} + \dots + x_{i-N+1}}{N}$$

$$SMA(i-1) = \frac{x_{i-1} + \dots + x_{i-N+1} + x_{i-N}}{N}$$

ICE #2

Moving mean computation

Based on the previous question, what's the computational complexity and memory/storage complexity of SMA at point i, i.e. $SMA(i)$?

a. $O(N), O(N)$
b. $O(1), O(N)$
c. $O(N), O(1)$
d. $O(1), O(1)$

$$SMA(i) = \underbrace{SMA(i-1)}_{O(1)\ \text{storage}} + \frac{(x_i - x_{i-N})}{N}$$

$\to O(1)$

O(1) Storage

O(1) time!

Moving Variance computation for SMA

Moving Variance

Same principle as computing the moving mean for SMA.

Exponential Moving Average and Anomalies

EMA

1. Similar to SMA - Except the moving window is **soft**
2. Weight more of the recent terms than before and weight it exponentially.
3. $\boxed{EMA(i) = (1-\beta) * EMA(i-1) + \beta * x_i}$ where $0 \leq \beta \leq 1$
4. $EMA(i) = \beta x_i + \beta(1-\beta)x_{i-1} + \beta(1-\beta)^2 x_{i-2} + \ldots$
5. EMA has a hyper-parameter β instead of window size N as in SMA.

SMA:

sliding window

More weight

EMA:

Less weight/no weight

Exponential Moving Average and Anomalies

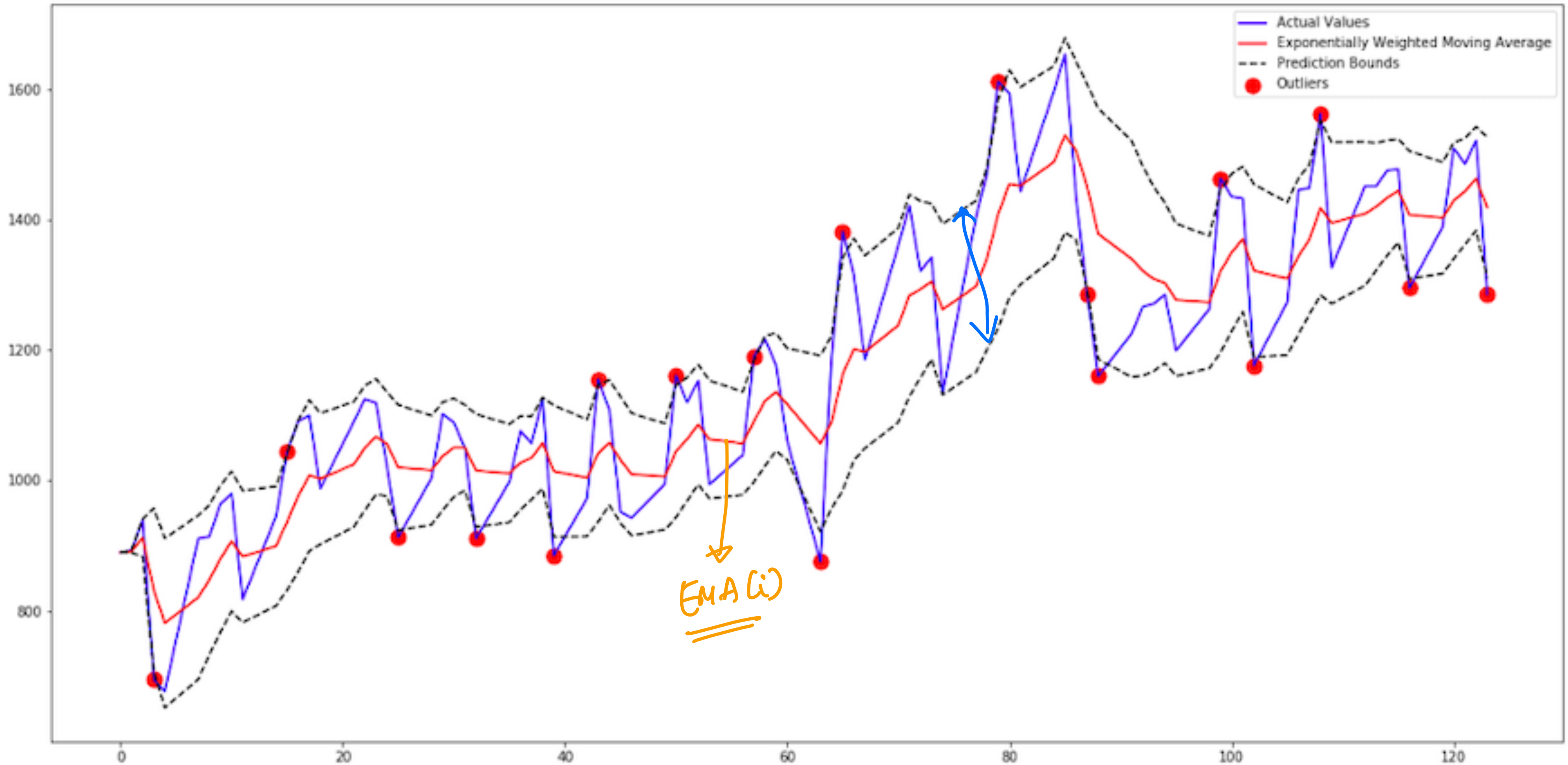
EMA

1. Similar to SMA - Except the moving window is **soft**
2. Weight more of the recent terms than before and weight it exponentially.
3. $EMA(i) = (1-\beta) * EMA(i-1) + \beta * x_i$ where $0 \leq \beta \leq 1$
4. $EMA(i) = \beta x_i + \beta(1-\beta)x_{i-1} + \beta(1-\beta)^2 x_{i-2} + \ldots$
5. EMA has a hyper-parameter β instead of window size N as in SMA.

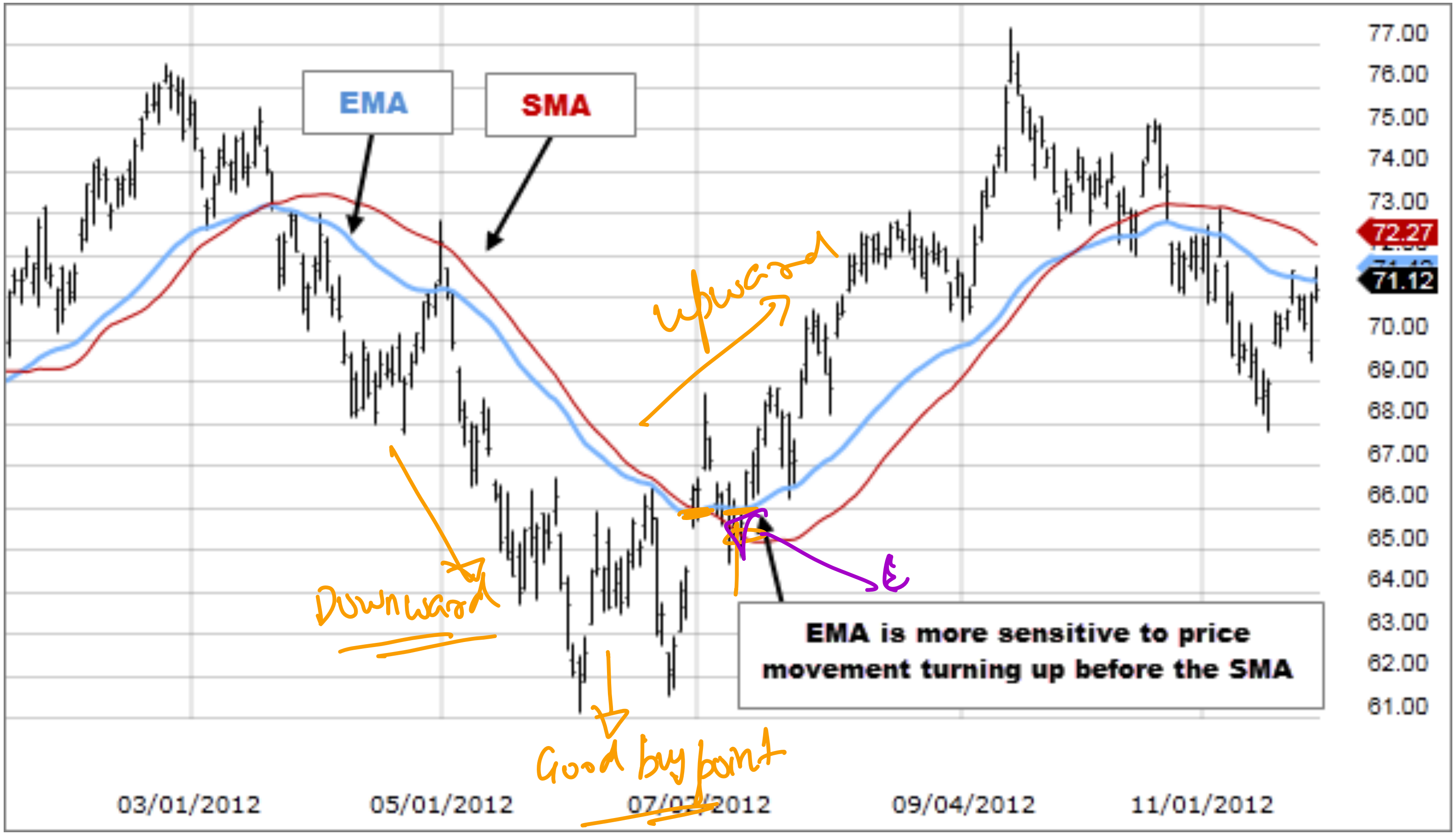
Anomaly detection

1. x_i is an anomaly if $\|EMA(i) - x_i\|$ deviates above a $t \times SD(i)$ where $SD(i)$ is the standard deviation of the deviation.

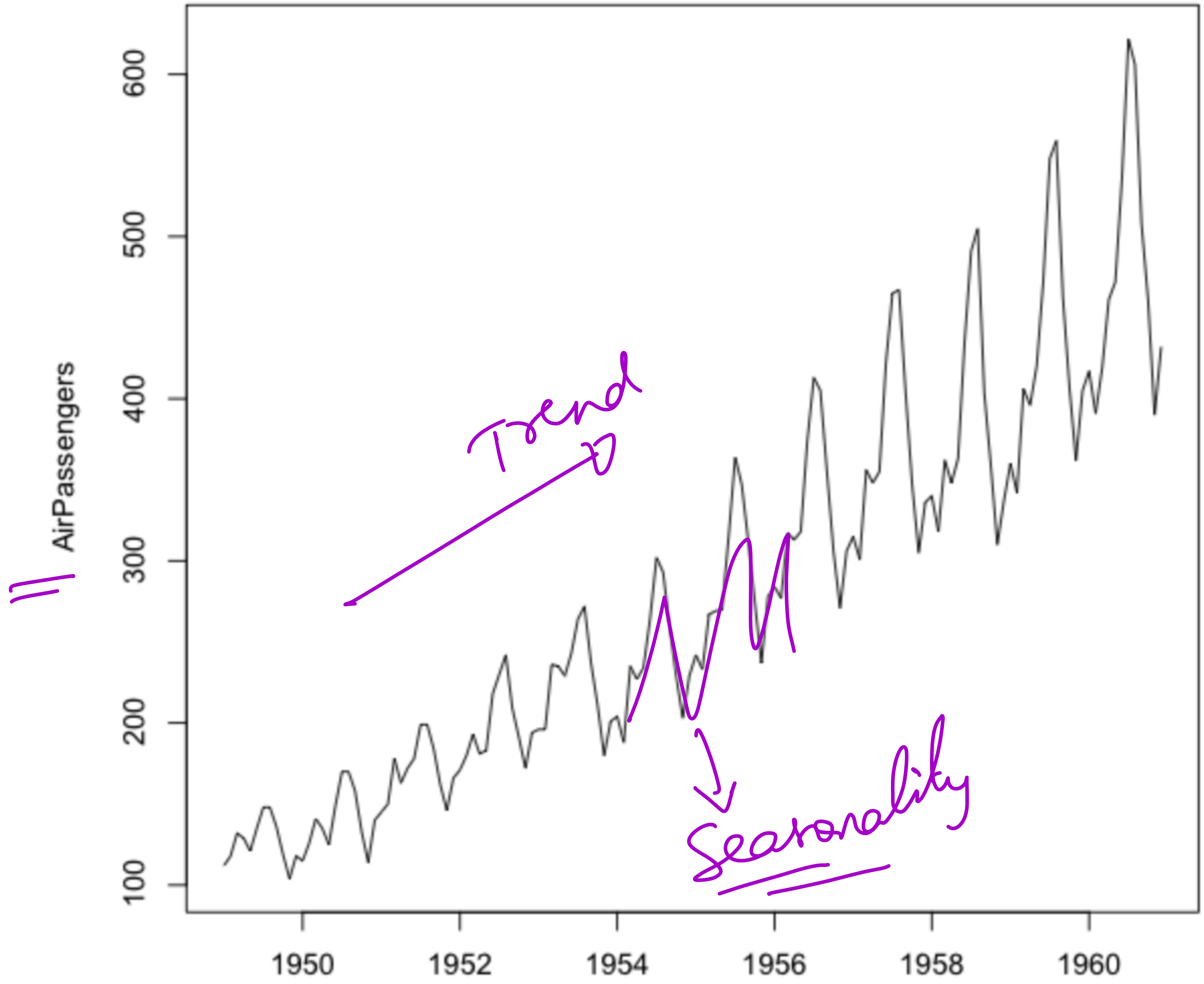
Exponential Moving Average



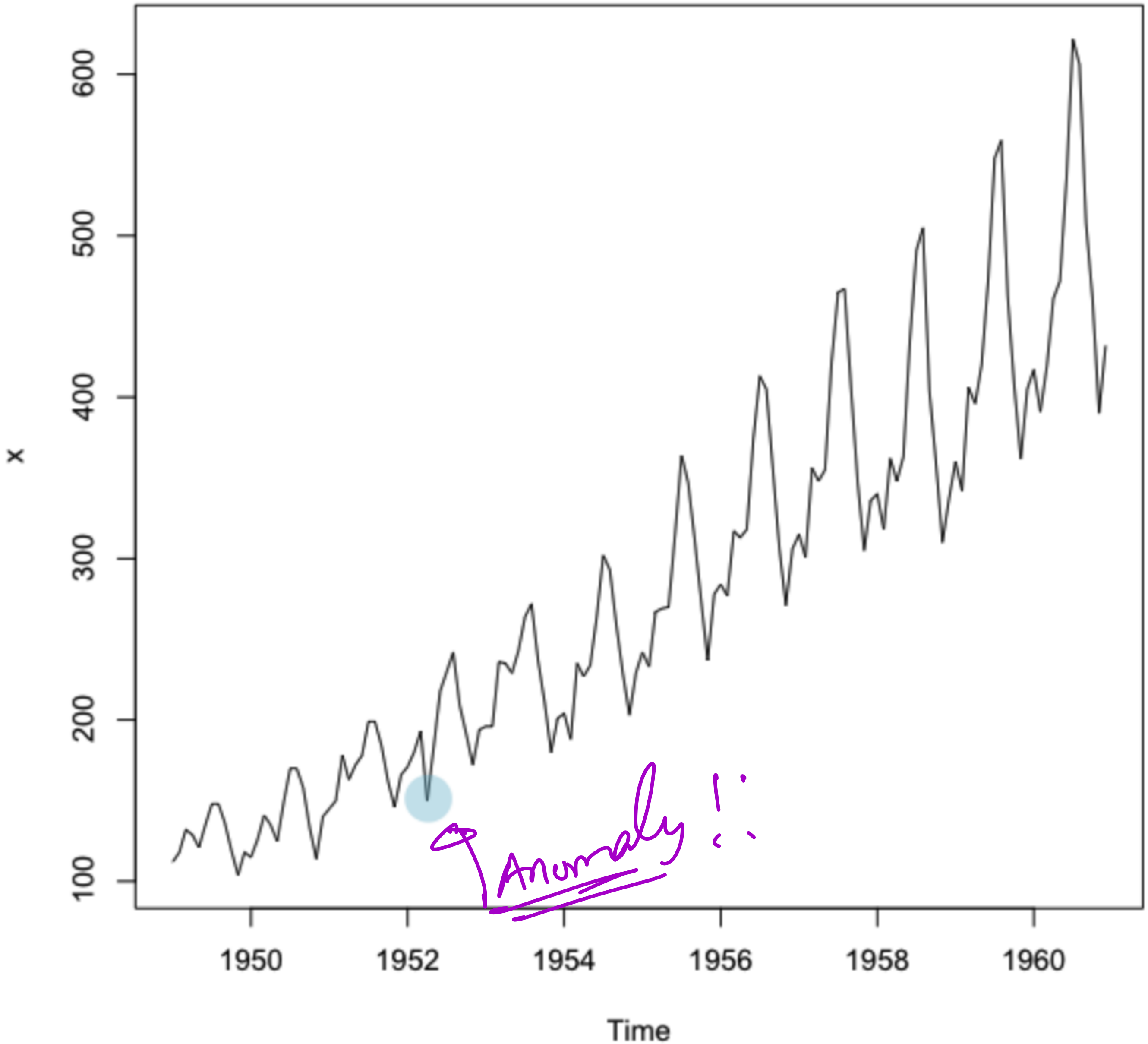
SMA vs EMA


EMA
SMA
77.00
76.00
75.00
74.00
73.00
72.27
71.12
70.00
69.00
68.00
67.00
66.00
65.00
64.00
63.00
62.00
61.00
03/01/2012
05/01/2012
07/02/2012
09/04/2012
11/01/2012
EMA is more sensitive to price movement turning up before the SMA
upward
Downward
Good buy point
E

Accounting for Seasonality



Accounting for Seasonality



STL: Accounting for Seasonality

Data =

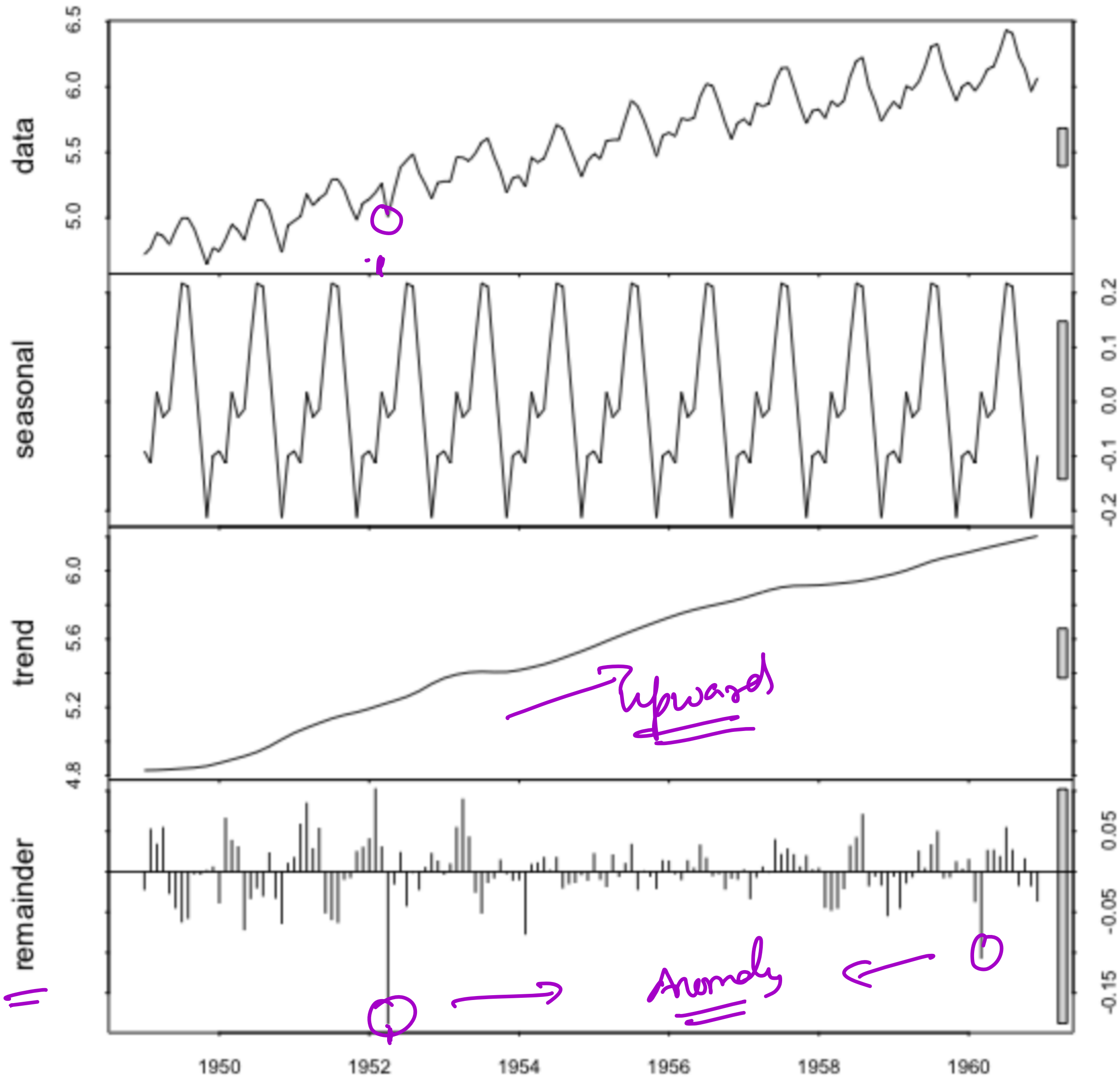
Seasonality

+

Trend

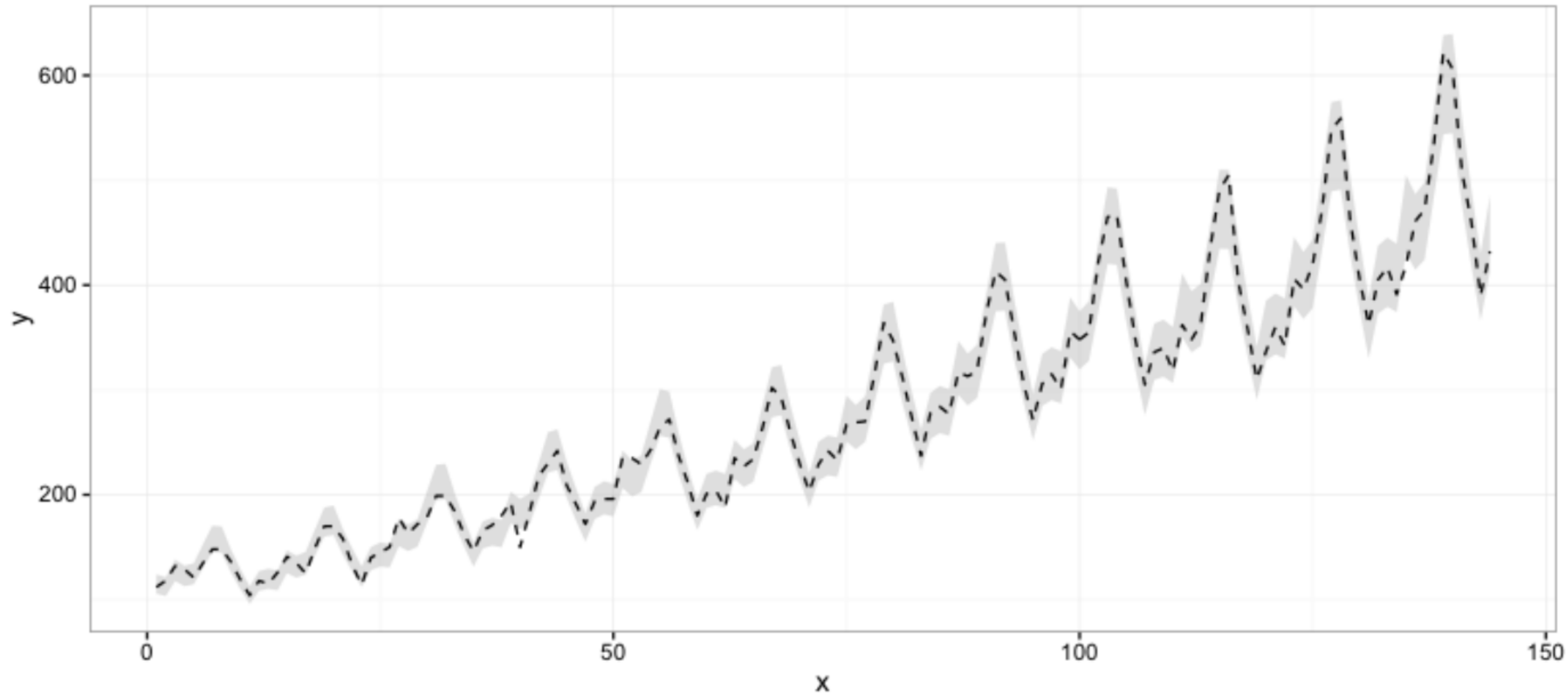
+

Residual



$\mu = 0$

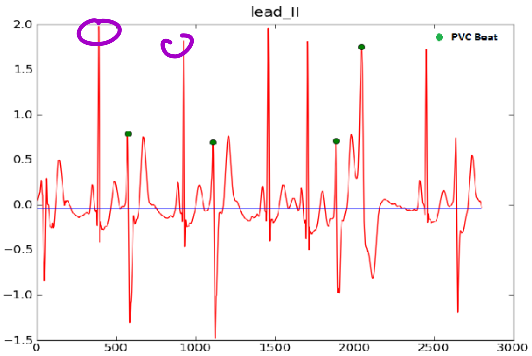
STL: Accounting for Seasonality



STL Library

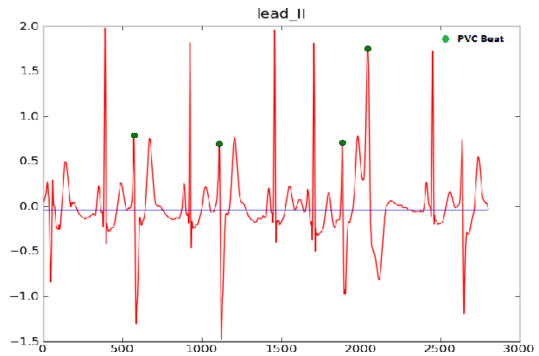
Prophet Anomaly Detection

For the upcoming assignment on Arrhythmia Detection



1. Try SMA/EMA (unsupervised baseline)

For the upcoming assignment on Arrhythmia Detection



1. Try SMA/EMA (unsupervised baseline)
2. Try a supervised linear-model baseline like Logistic Regression/SVM

For the upcoming assignment on Arrhythmia Detection

1. Try SMA/EMA (unsupervised baseline)
2. Try a supervised linear-model baseline like Logistic Regression
3. Try a supervised non-linear model like Random Forest (non-deep learning)

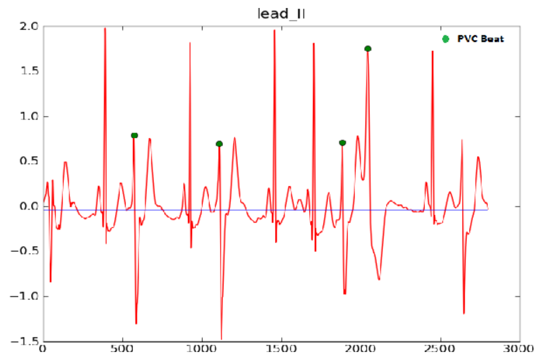
For the upcoming assignment on Arrhythmia Detection

1. Try SMA/EMA (unsupervised baseline)
2. Try a supervised linear-model baseline like Logistic Regression
3. Try a supervised non-linear model like Random Forest
4. Try a deep learning model

For the upcoming assignment on Arrhythmia Detection

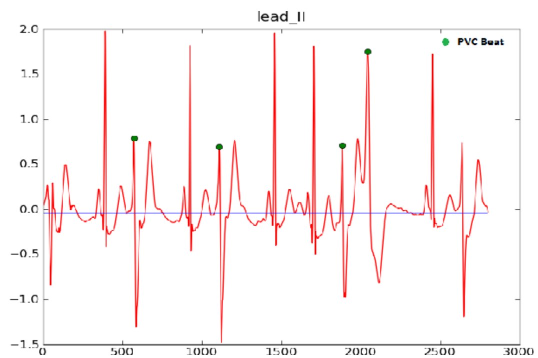
1. Try SMA/EMA (unsupervised baseline) 1st
2. Try a supervised linear-model baseline like Logistic Regression
3. Try a supervised non-linear model like Random Forest
4. Try a deep learning model
5. Benchmark offline results in a tabular format with algorithms and metrics.

For the upcoming assignment on Arrhythmia Detection



1. Try SMA/EMA (unsupervised baseline)
2. Try a supervised linear-model baseline like Logistic Regression
3. Try a supervised non-linear model like Random Forest
4. Try a deep learning model
5. Benchmark offline results in a tabular format with algorithms and metrics.
6. Kaggle competition benchmarks performance on held-out and unseen test set

For the upcoming assignment on Arrhythmia Detection



1. Try SMA/EMA (unsupervised baseline)
2. Try a supervised linear-model baseline like Logistic Regression
3. Try a supervised non-linear model like Random Forest
4. Try a deep learning model
5. Benchmark offline results in a tabular format with algorithms and metrics.
6. Kaggle competition benchmarks performance on held-out and unseen test set

   why??
7. Share your insights in the process - Pros/cons of different approaches and what changes give you a boost in performance and why?

Next Topic: Deep Learning

Introduction to Deep Learning

Deep Learning

1. Lot of buzz around Deep Learning in the past decade!

Deep Learning

1. Lot of buzz around Deep Learning in the past decade!
2. Deep Learning refers to Neural Networks that is a loose approximation of how the brain works

Why was NN not popular in 2000's?
- Not enough data
- not enough compute power
- No cloud infra
- overfitting issues

Revolution in
NLP & CV
DL

Applications of Deep Learning

Applications

1. Self-driving cars ICV

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis | NLP

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization NLP

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course! (TS)

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course!
5. Image to text generation. Caption images automatically. CV/NLP

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course!
5. Image to text generation. Caption images automatically.
6. Machine Translation. Translate a French sentence to English sentence. Sequence to sequence architecture NLP

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course!
5. Image to text generation. Caption images automatically.
6. Machine Translation. Translate a French sentence to English sentence. Sequence to sequence architecture Transformers
7. Auto-complete sentence in Emails. How many of us use this?

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course!
5. Image to text generation. Caption images automatically.
6. Machine Translation. Translate a French sentence to English sentence. Sequence to sequence architecture
7. Auto-complete sentence in Emails. How many of us use this?
8. Auto-complete search results.

Applications of Deep Learning

Applications

1. Self-driving cars
2. Sentiment analysis
3. Text Summarization
4. Arrythmia detection - Possible assignment for this course!
5. Image to text generation. Caption images automatically.
6. Machine Translation. Translate a French sentence to English sentence. Sequence to sequence architecture
7. Auto-complete sentence in Emails. How many of us use this?
8. Auto-complete search results. Bard Google
9. Chat bots - Like ChatGPT/Sparrow/Anthropic, etc NLP

OpenAI

Email auto-complete

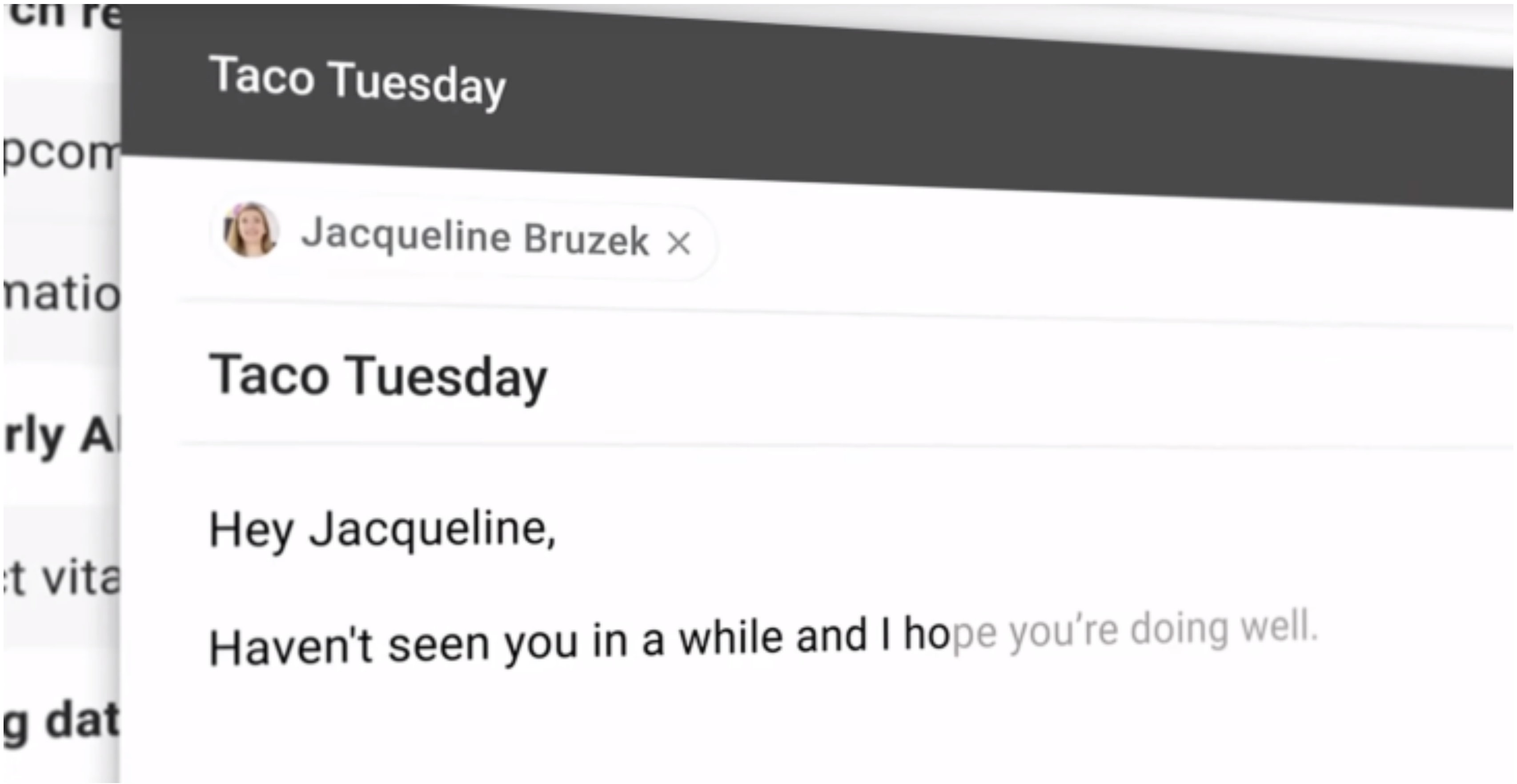
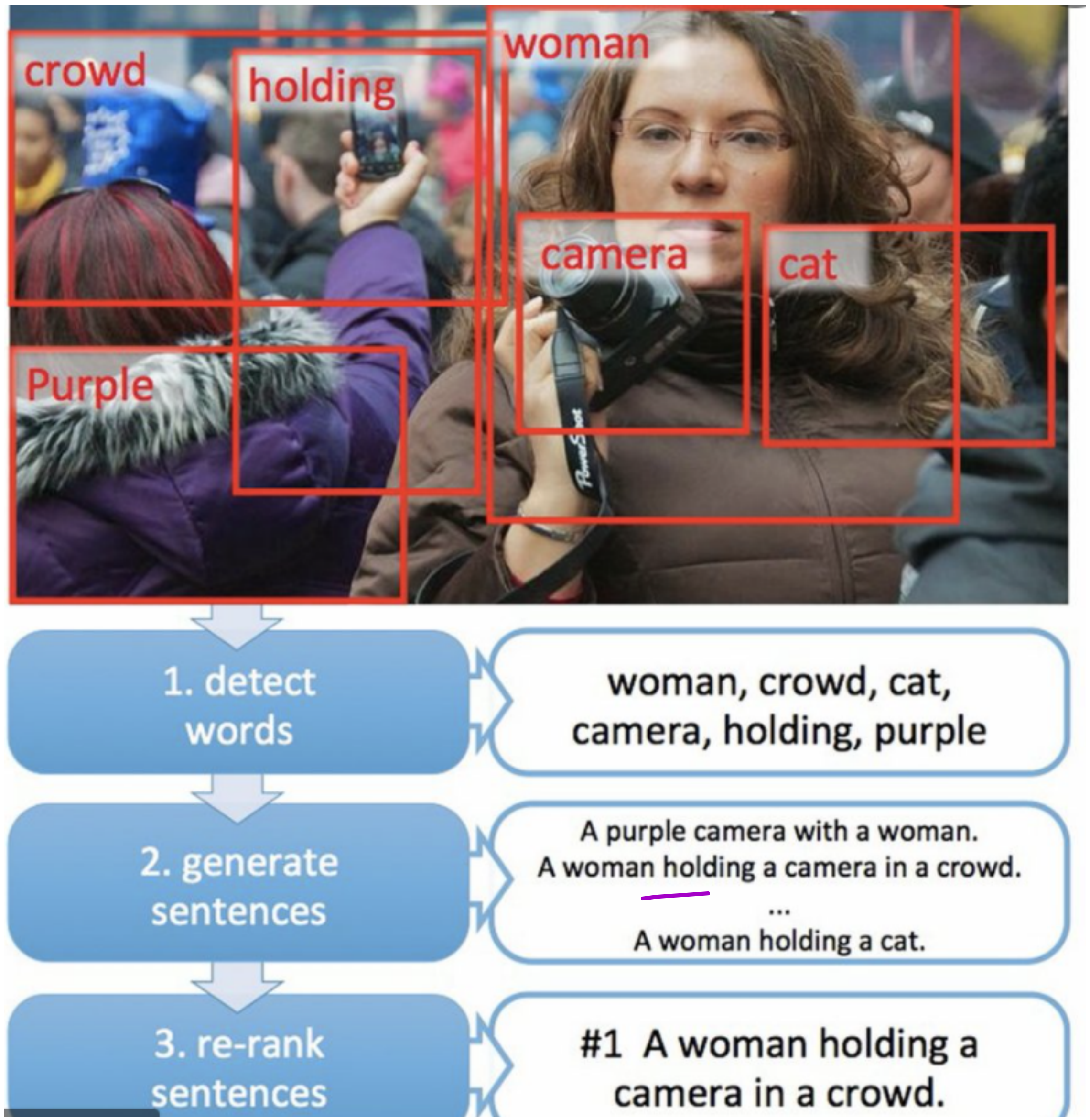


Image to Text!

CNN model →

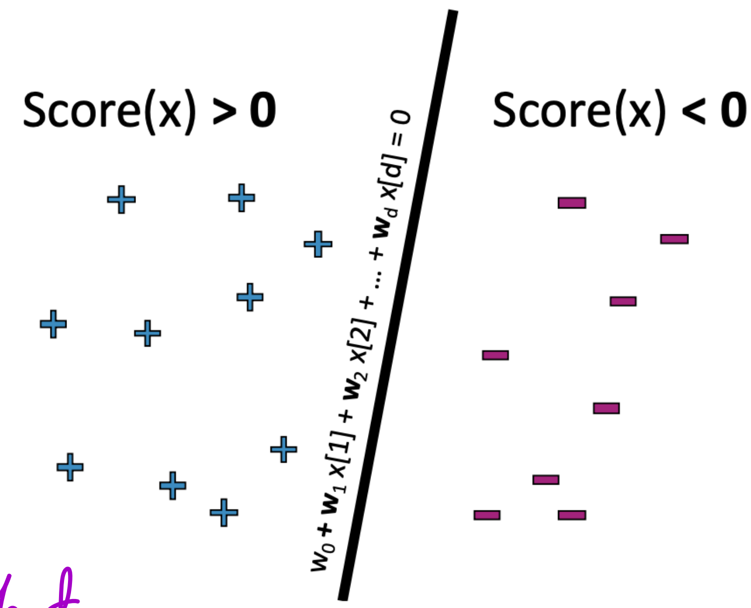
+

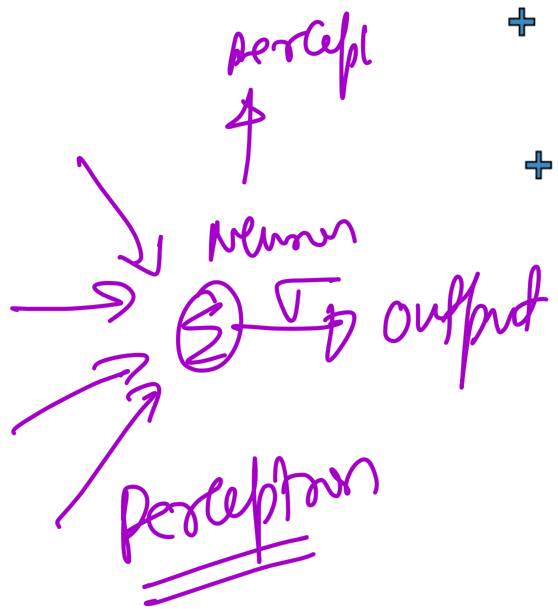
Transformer/ LSTM model



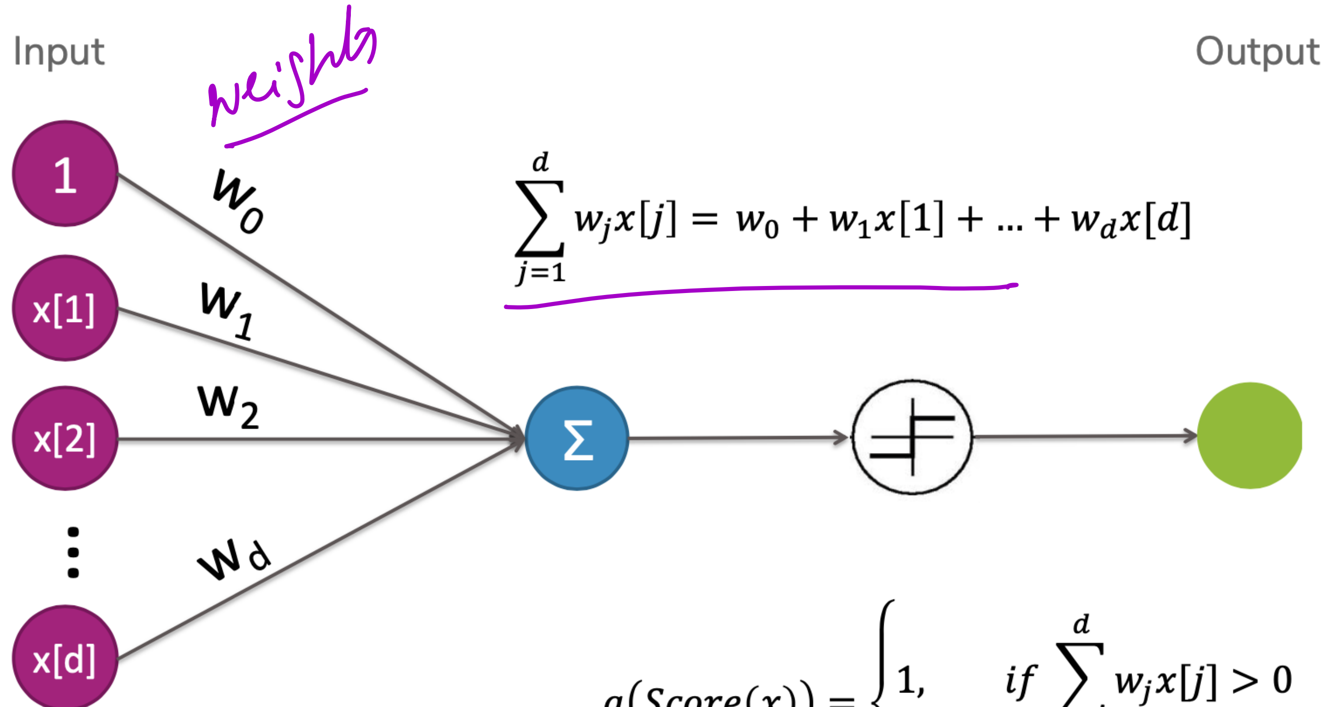
Perceptron

$\text{Score}(x) = w_0 + \mathbf{w}_1\, x[1] + \mathbf{w}_2\, x[2] + \ldots + \mathbf{w}_d\, x[d]$





Perceptron



$$\sum_{j=1}^{d} w_j x[j] = w_0 + w_1 x[1] + \ldots + w_d x[d]$$

$$g(Score(x)) = \begin{cases} 1, & if \sum_{j=1}^{d} w_j x[j] > 0 \\ 0, & otherwise \end{cases}$$

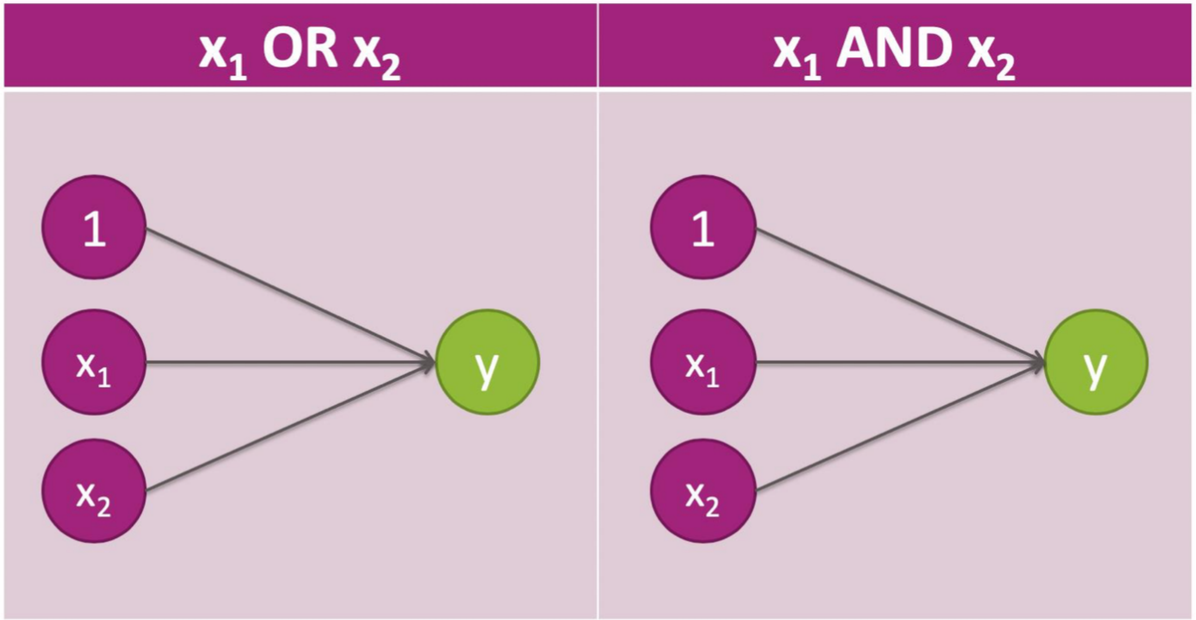
→ Perceptron

→ Not differentiable

RELU

OR and AND Functions

What can a perceptrons represent?



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Learning XOR

XOR through Multi-layer perceptron

Perceptron 1
Perceptron 3
Perceptron 2

This is a 2-layer neural network

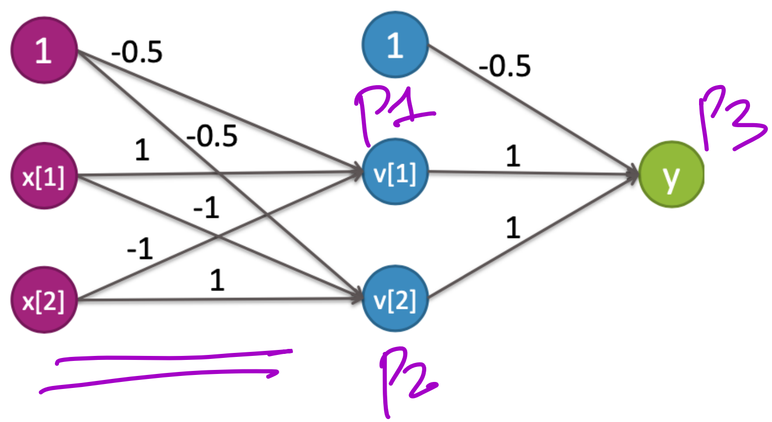
$y = x[1]\ XOR\ x[2] = (x[1]\ AND\ !\,x[2])\ OR\ (!\,x[1]\ AND\ x[2])$

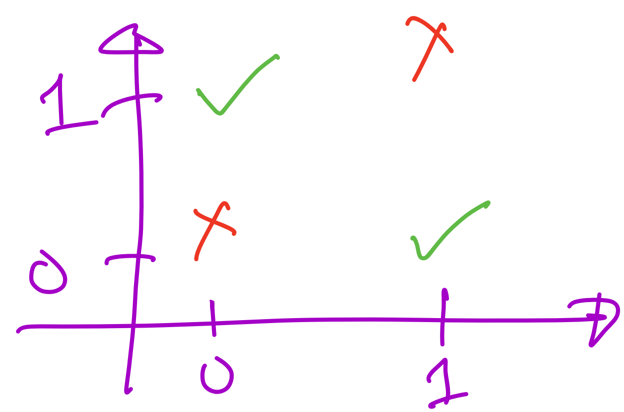
$v[1] = (x[1]\ AND\ !\,x[2])$
$\quad = g(-0.5 + x[1] - x[2])$

$v[2] = (!\,x[1]\ AND\ x[2])$
$\quad = g(-0.5 - x[1] + x[2])$

$y = v[1]\ OR\ v[2]$
$\quad = g(-0.5 + v[1] + v[2])$

MLP = Extension of Single perceptron ↕ Logistic Regression



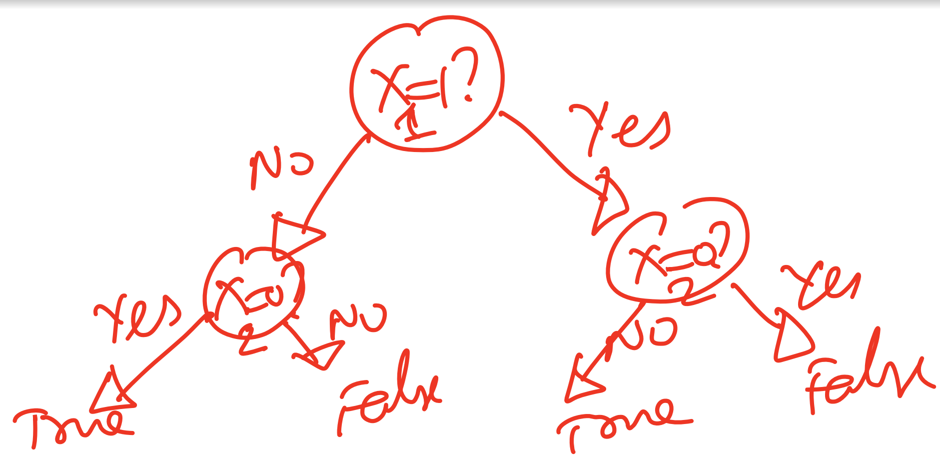


Which methods can learn the XOR function?

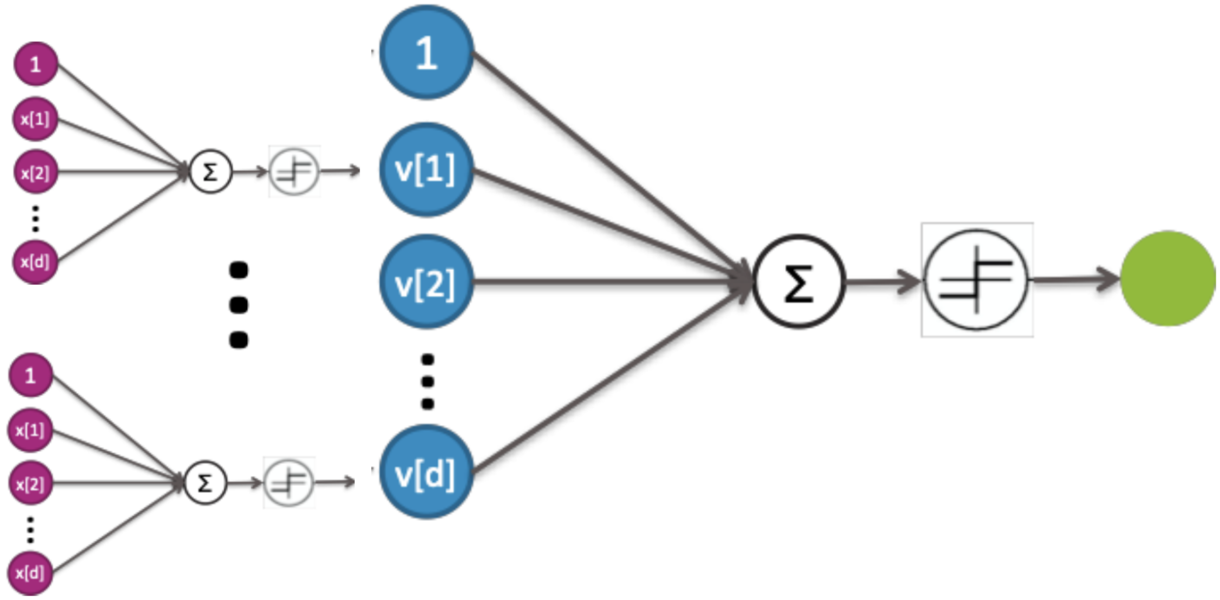
1. Logistics Regression
2. Naive Bayes Classifier → Baye Rule
3. Decision Trees
4. Support Vector Machines

Goal

$$P(Y/X) = \frac{P(X/Y)\,P(Y)}{P(X)}$$



Multi-Layer Perceptron (MLP)

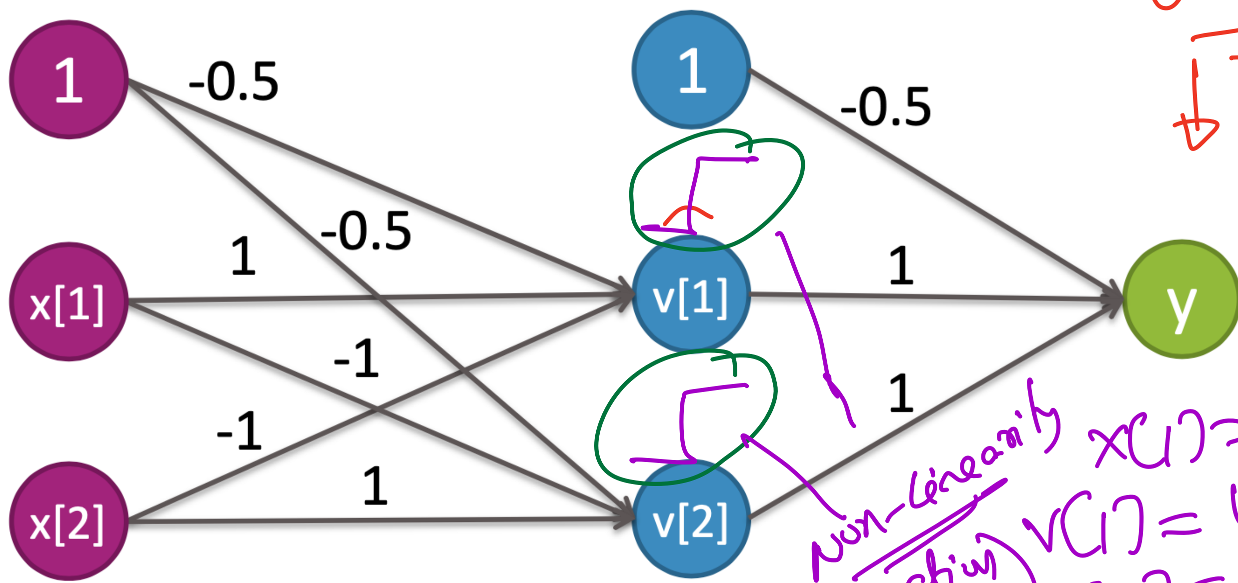


Multi-Layer Perceptron (MLP)

Input Layer (x)

Hidden Layer

output (y)



Non-Linearity (Activation function)

$x[1]=1, x[2]=1$

$v[1] = 1-1-0.5 = -0.5 \to 0$

$v[2] = -1+1-0.5 = -0.5 \to 0$

$y = -0.5 < 0 \Rightarrow$ False

X → Can Learn XOR

$x[1]=1, x[2]=0$

$v[1] = 1-0.5 = 0.5 \to 1$

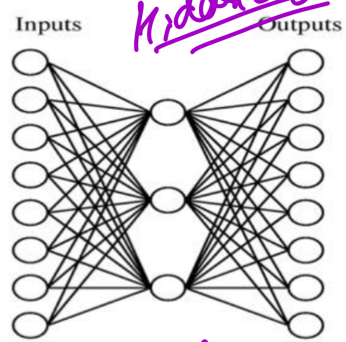
$v[2] = -1-0.5 = -1.5 \to 0$

$y = v[1]+v[2]-0.5 = 1+0-0.5 = 0.5 > 0 \Rightarrow$ True

Two layer neural network (alt. one hidden-layer neural network)

Hidden Layer

Feed Forward NN



Sigmoid/Softmax

Logistic Regression

Single

(Linear model)

y

$$out(x) = g\left(w_0 + \sum_j w_j x[j]\right)$$

(one-Layer NN)

Second Layer

1-hidden layer

2 layer NN

$$out(x) = g\left(w_0 + \sum_k w_k g\left(w_0^{(k)} + \sum_j w_j^{(k)} x[j]\right)\right)$$

y 2 1

Non-linear model

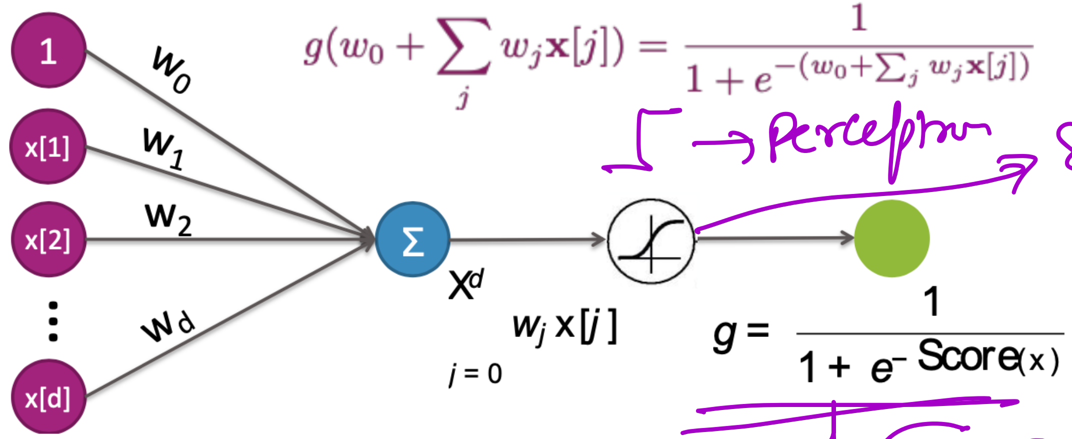
Activation Function

First Layer

Perceptron to Logistic Regression

one-layer NN

$$g(w_0 + \sum_j w_j \mathbf{x}[j]) = \frac{1}{1 + e^{-(w_0 + \sum_j w_j \mathbf{x}[j])}}$$



$\int$ → Perceptron

Sigmoid → LR

Sigmoid

Choices for Non-Linear Activation Function

useful for getting a prob

- Sigmoid
  - Historically popular, but (mostly) fallen out of favor
- Neuron's activation saturates
  (weights get very large -> gradients get small)
- Not zero-centered -> other issues in the gradient steps
  - When put on the output layer, called "softmax" because interpreted as class probability (soft assignment)

- Hyperbolic tangent $g(x) = \tanh(x)$
  - Saturates like sigmoid unit, but zero-centered

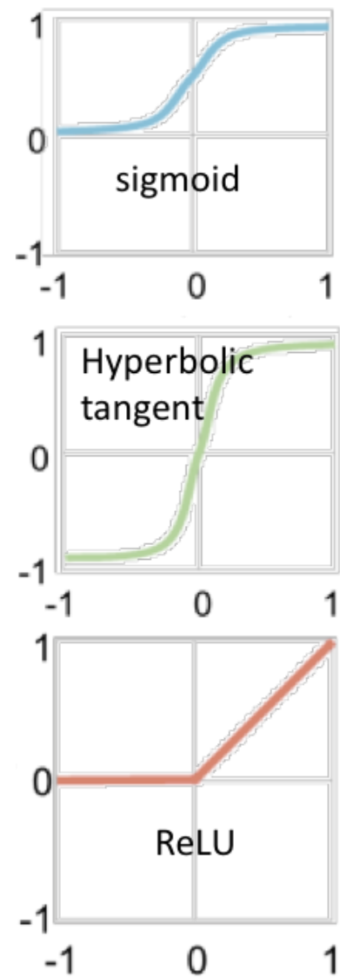
f linear

- Rectified linear unit (ReLU) $g(x) = x^{+} = \max(0,x)$
  - Most popular choice these days
  - Fragile during training and neurons can "die off"… be careful about learning rates
  - "Noisy" or "leaky" variants

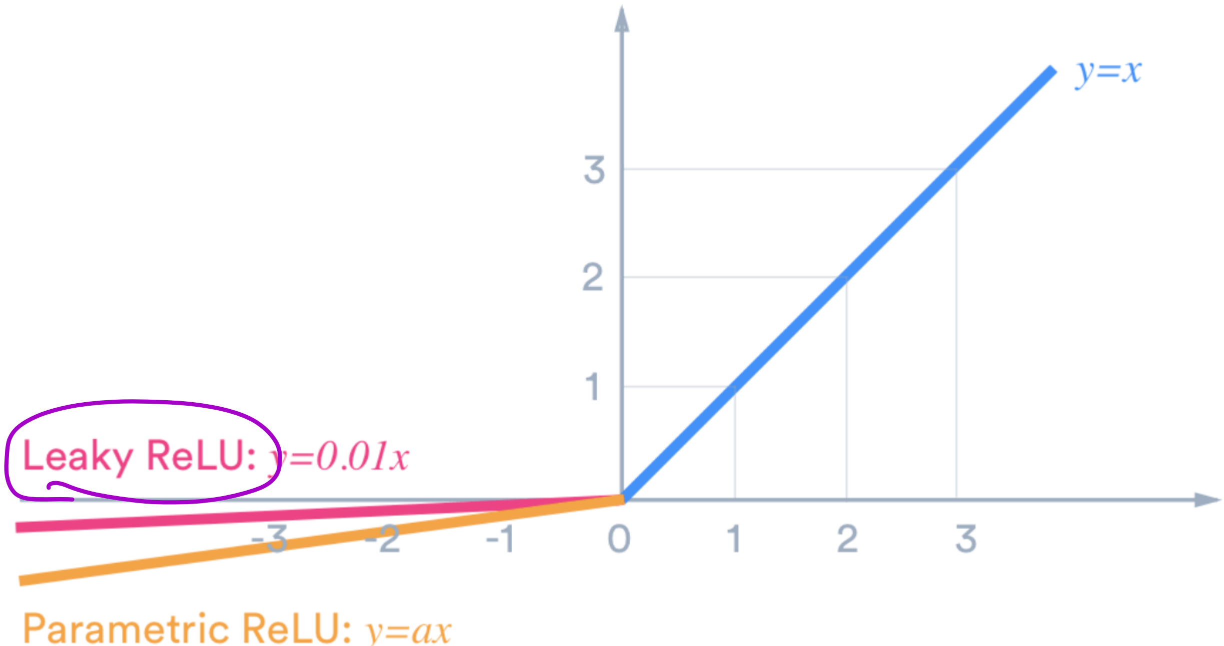
f rectifi

RELU

- Softplus $g(x) = \log(1+\exp(x))$
  - Smooth approximation to rectifier activation

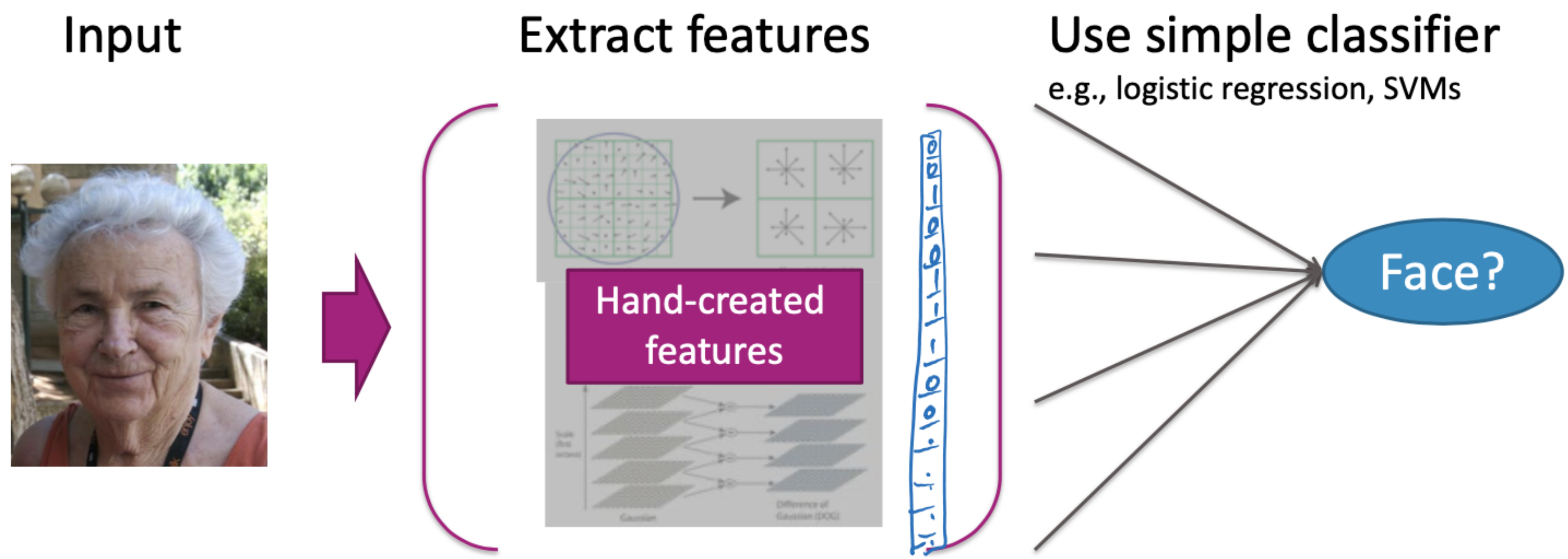


RELU vs Leaky RELU

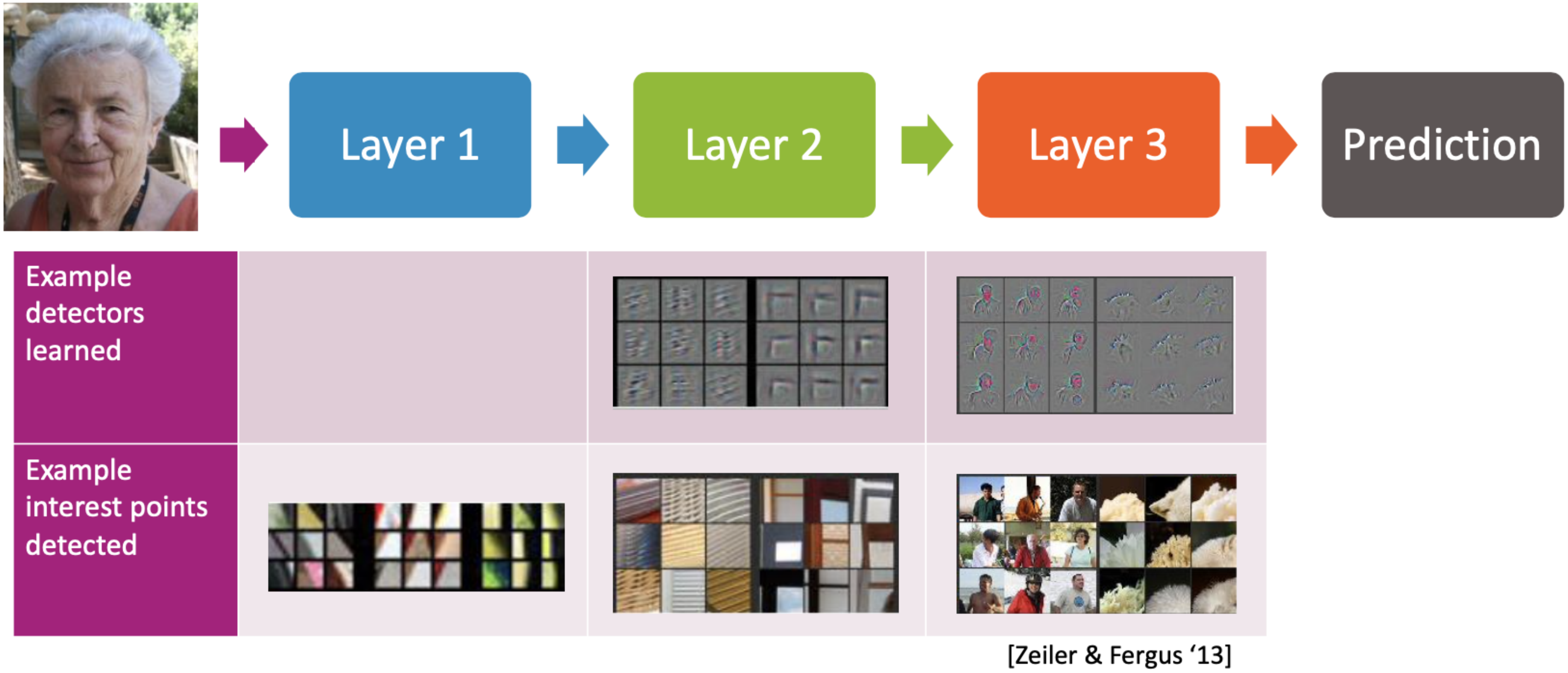


Leaky
when values are -ve

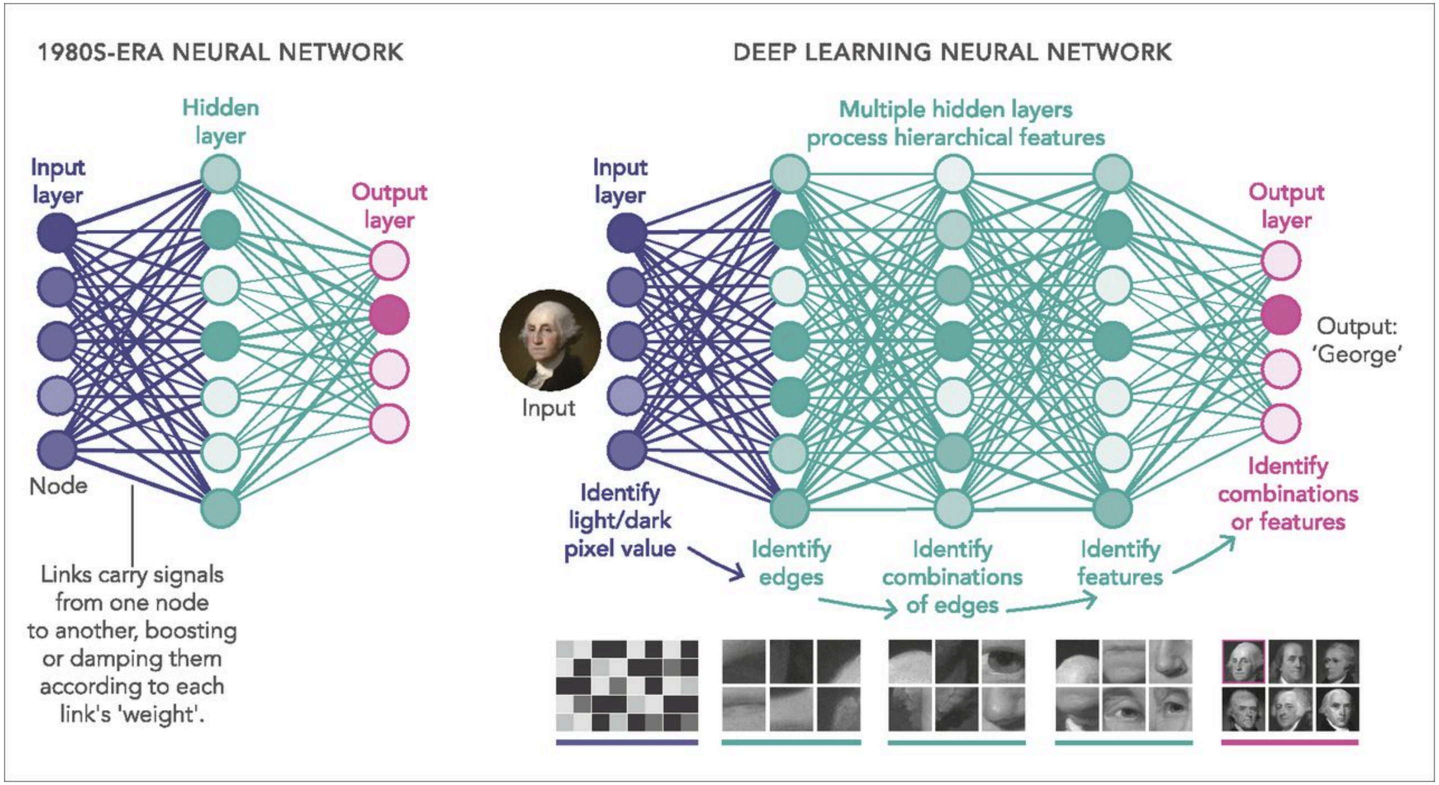
Computer vision before deep learning



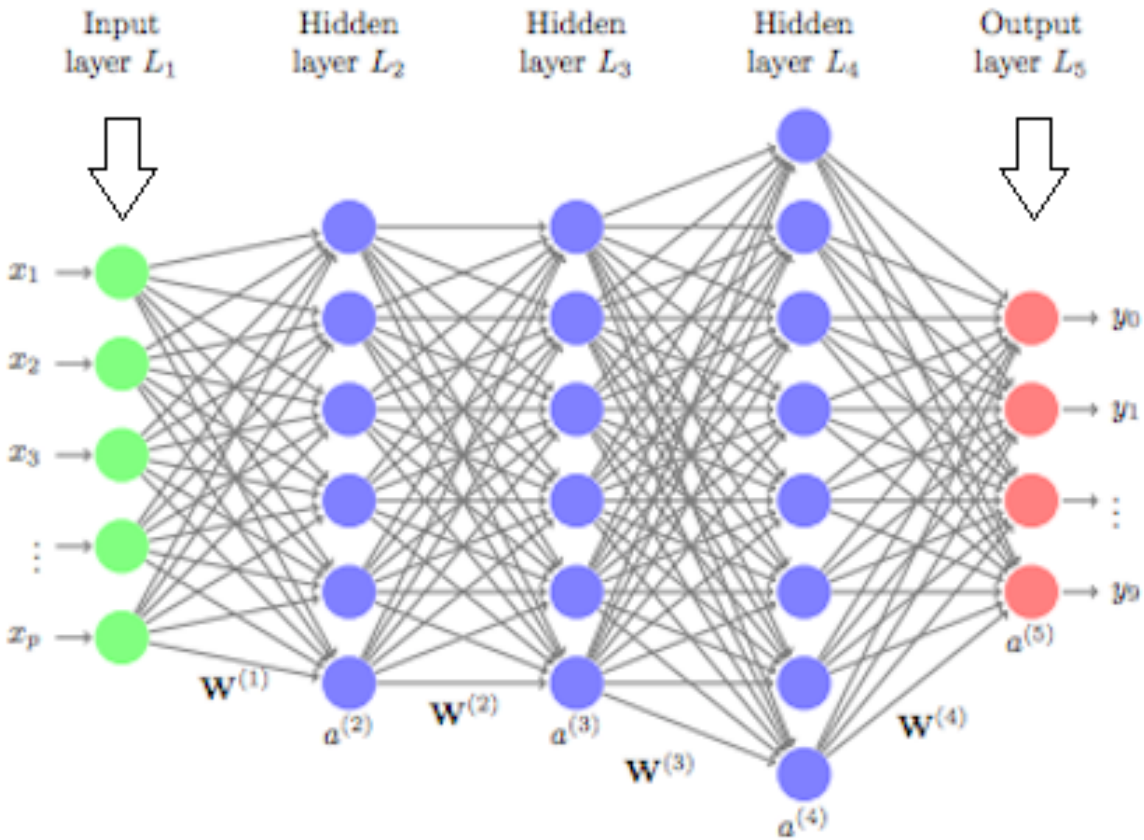
Computer vision after deep learning



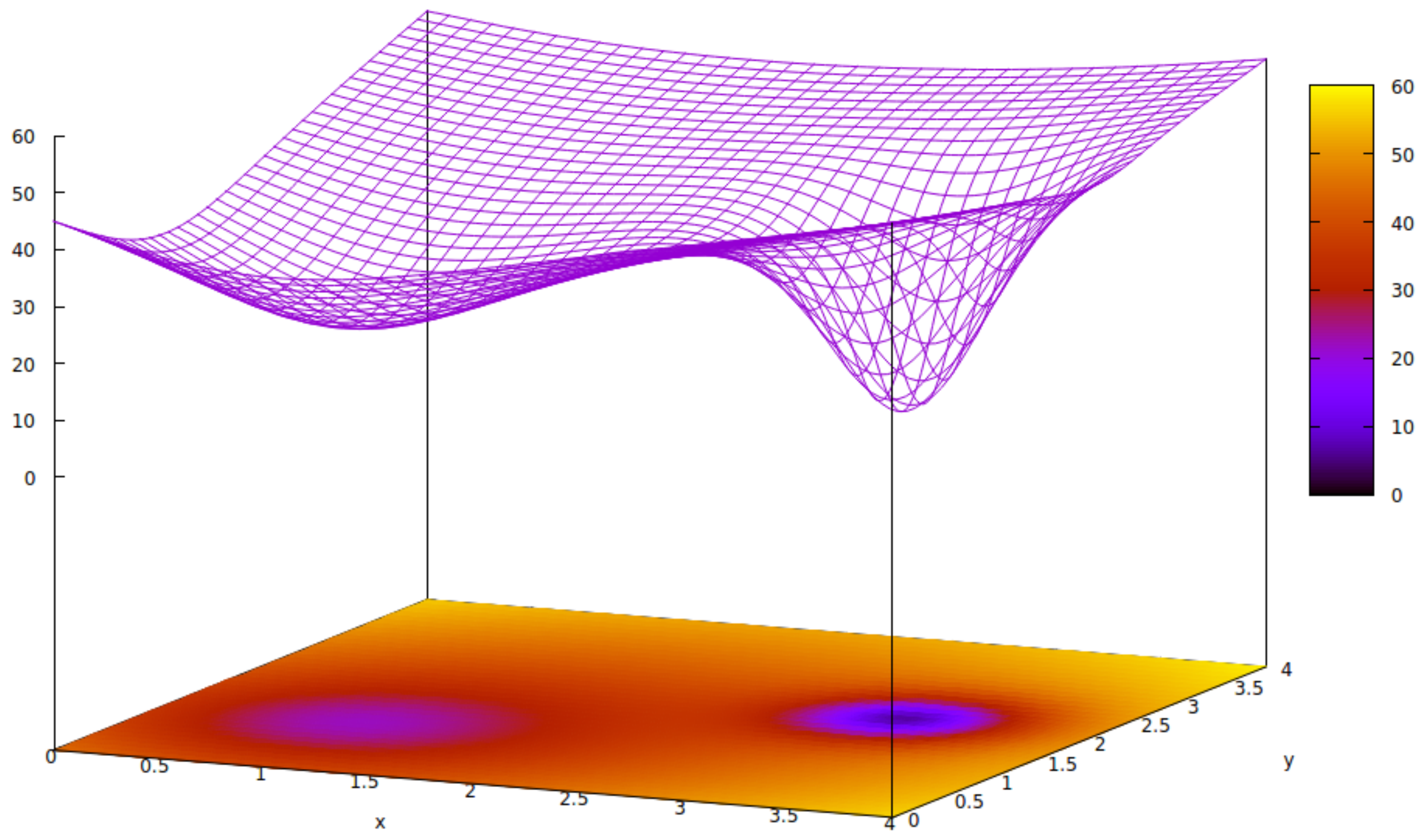
Feed-forward Deep Learning Architecture Example



Feed-forward Deep Learning Architecture Example



Good vs Bad Local minima



Hyper-parameters in Deep Learning

ICE #4: Which of the following is not a hyper-parameter in deep learning?

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. None of the above
5. All of the above

Hyper-parameters in Deep Learning

Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer

Hyper-parameters in Deep Learning

Hyper-parameters

1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used

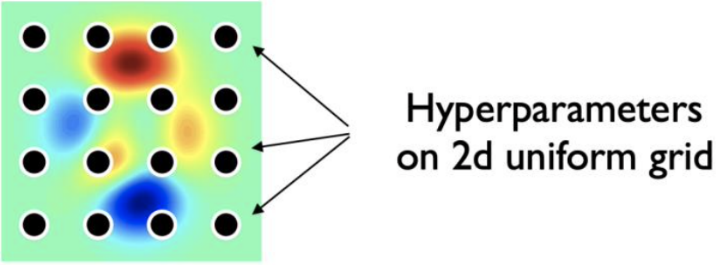
Hyper-parameters in Deep Learning

Hyper-parameters

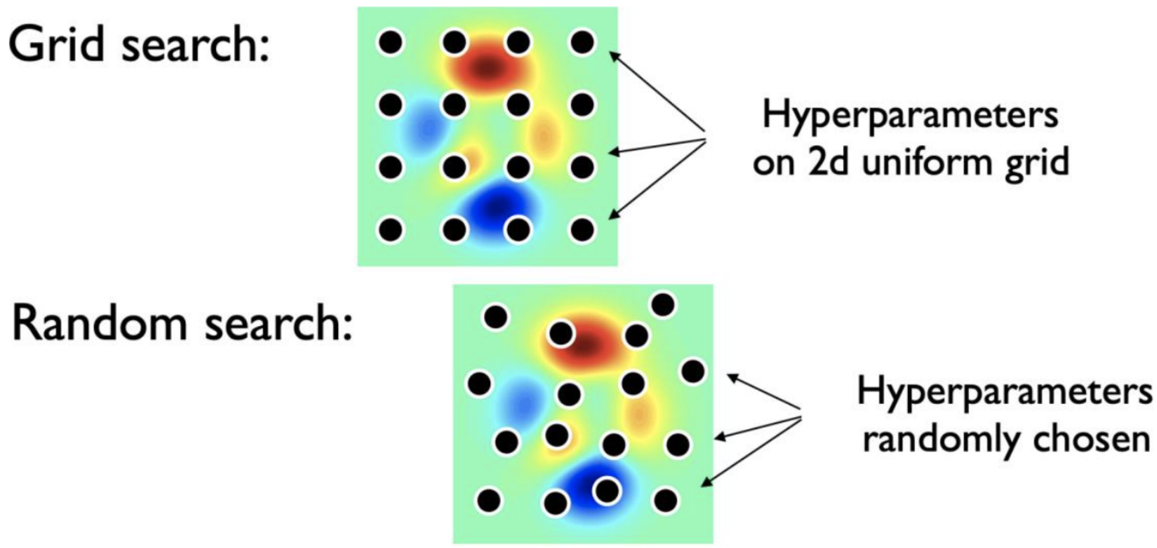
1. Learning rate
2. Number of Hidden Layers
3. Number of neurons per hidden layer
4. Type of non-linear activation function used
5. Anything else?

Hyper-parameter tuning methods

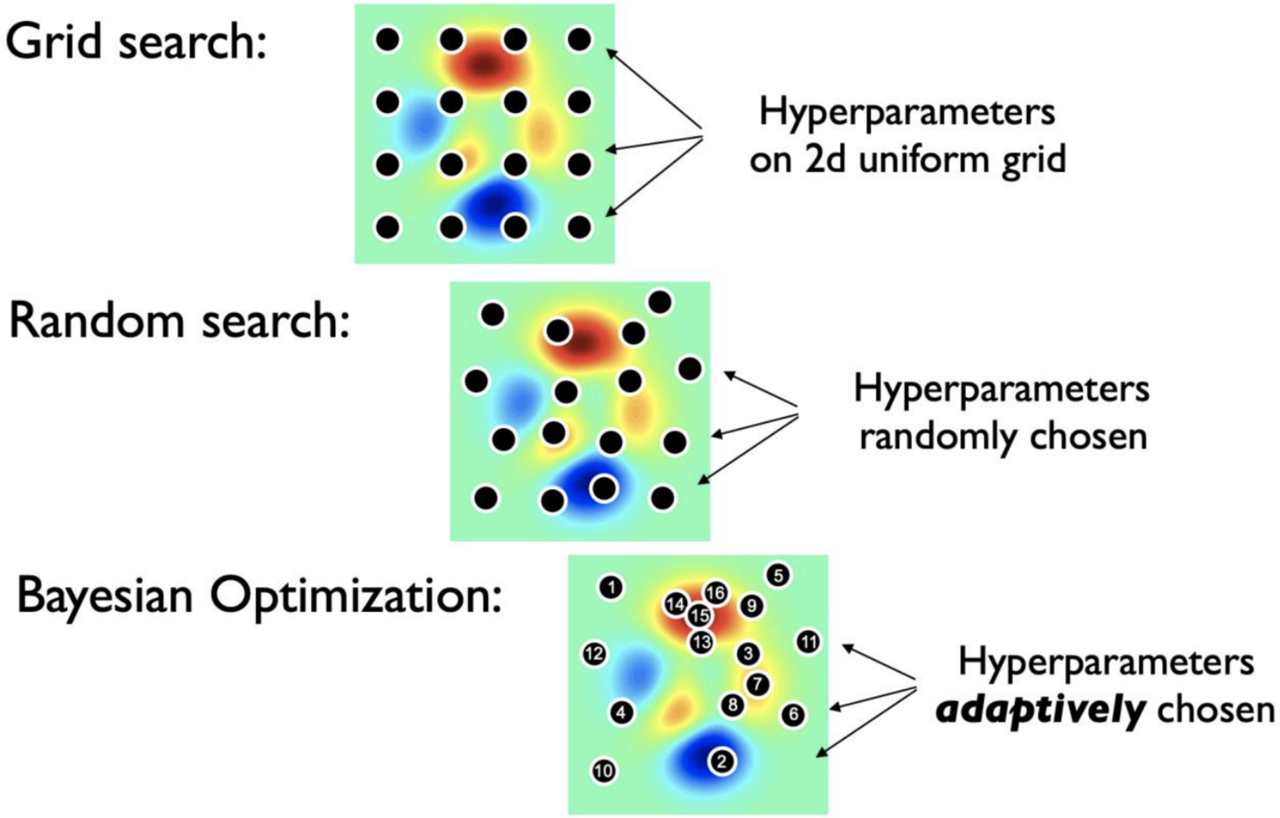
Grid search:



Hyper-parameter tuning methods



Hyper-parameter tuning methods



ICE #5

Compute the number of parameters in DNN model

Consider a DNN model with 3 hidden layers where each hidden layer has 1000 neurons. Let the input layer be raw pixels from a 100x100 image and the output layer has 10 dimensions, let's say for a 10 class image classification example. How many total parameters exist in the DNN model?

1. 10 million parameters
2. 11 million parameters
3. 12 million parameters
4. 13 million parameters

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?

Over-fitting in DNNs

How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!

Over-fitting in DNNs

How to handle over-fitting in DNNs

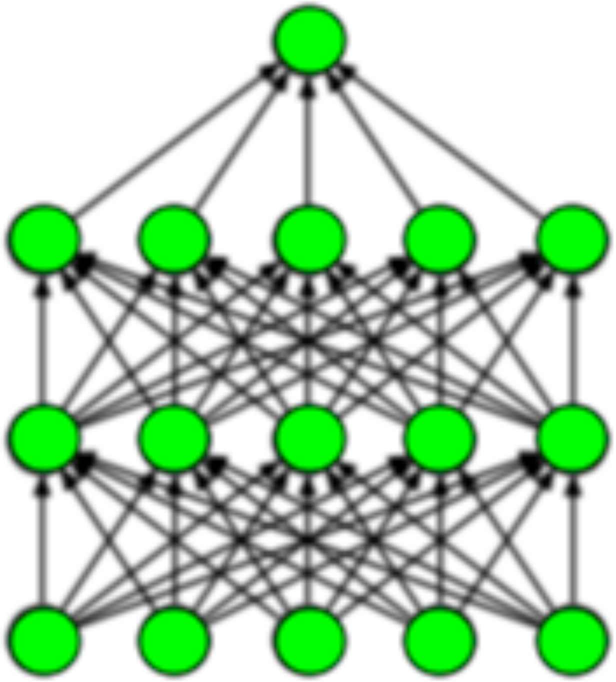
1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??

Over-fitting in DNNs

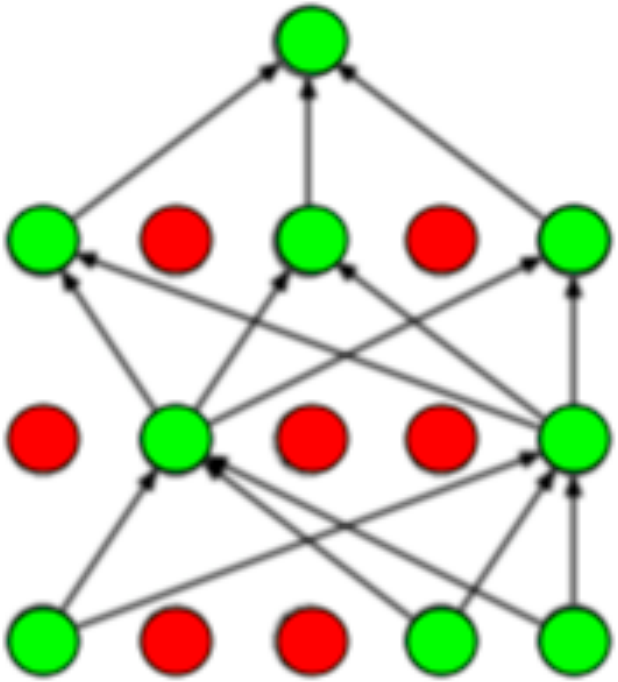
How to handle over-fitting in DNNs

1. A DNN model with 100 million parameters and only 100k data points or even a million data points will overfit unless we take care of over-fitting.
2. Weight regularization can help - ℓ_1, ℓ_2
3. More common over-fitting strategy for DL?
4. Dropouts!
5. Early stopping is also a great strategy! Stop training the DL model when the validation error starts increasing. How's this different from regular validation we were doing earlier??
6. Book by Yoshua Bengio has tons of details and great reference for Deep Learning!

Taking care of Over-fitting: Dropouts

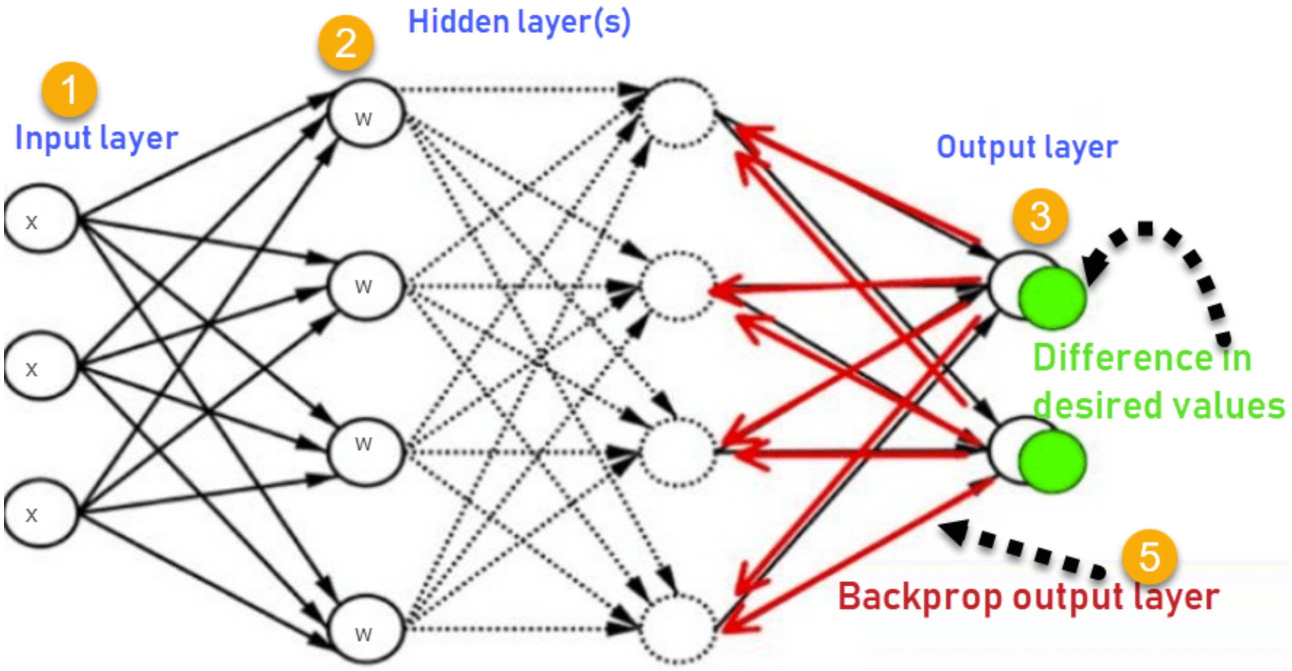


(a) Standard Neural Net



(b) After applying dropout.

Forward Propagation vs Back-propagation



Back Propagation explained