

Applied Deep Learning

Nitish Bhardwaj



Overview of Session :

- Deep Learning Introduction
- Application
- Image Classification
- Object Detection



Deep Learning : Introduction



Compiled By Nitish Bhardwaj

<https://www.linkedin.com/in/nitish-bhardwaj/>

Why is everyone talking about Deep Learning?

- DL is working really well (with awesome *accuracy numbers*) to solve some big problems like Image classification, Face Recognition, Recommendation system, Sentiment Analysis and a lot more.
- **These problems are solved in 90s, aren't they ??** Then why Deep Learning?
Because accuracy matters a lot and really a lot.
- DL has transformed the industry by solving these *already solved problems* with ***great accuracy numbers***.
 - For example, detection of hand-written digits, accuracy on test data
 - **88%** (LeCun et al. 1998) to
 - **99.7%** (Ciresan et al. CVPR 2012).
- DL has also brought solutions to many unsolved problems.
- Open-source + Academic Industry collaboration + Fight for Accuracy

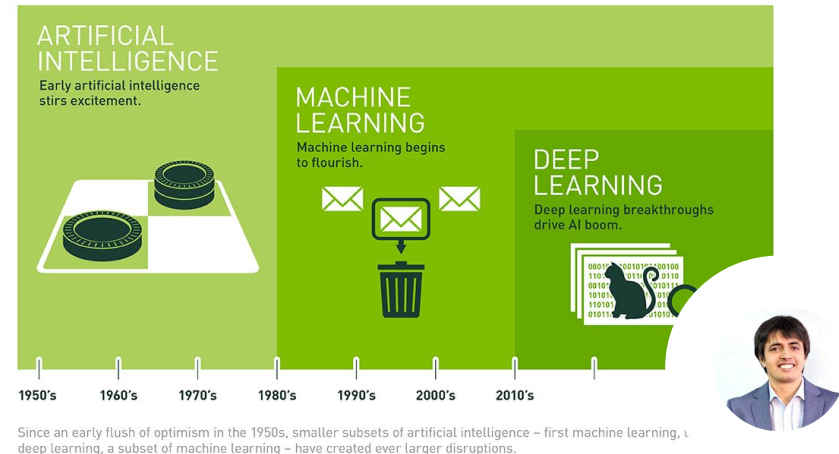
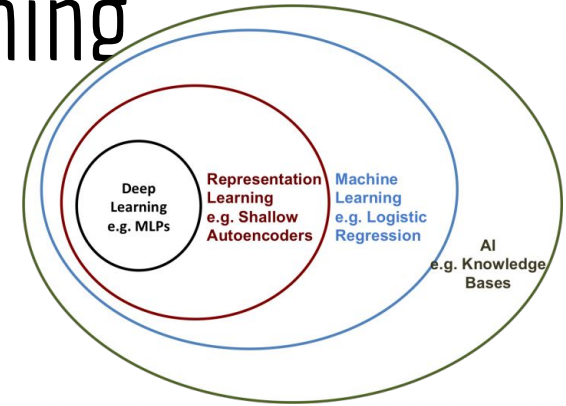


AI vs Machine learning vs Deep learning

AI (technique to enable machine to mimic humans) is super-set of ML

ML (program machines to learn and improve with experience) and **ML** is the super-set of **DL**.

DL (achieve great power and flexibility by learning to **represent the data as nested hierarchy of concepts**, i.e, more depth features with multiple processing layers, hence “deep”).

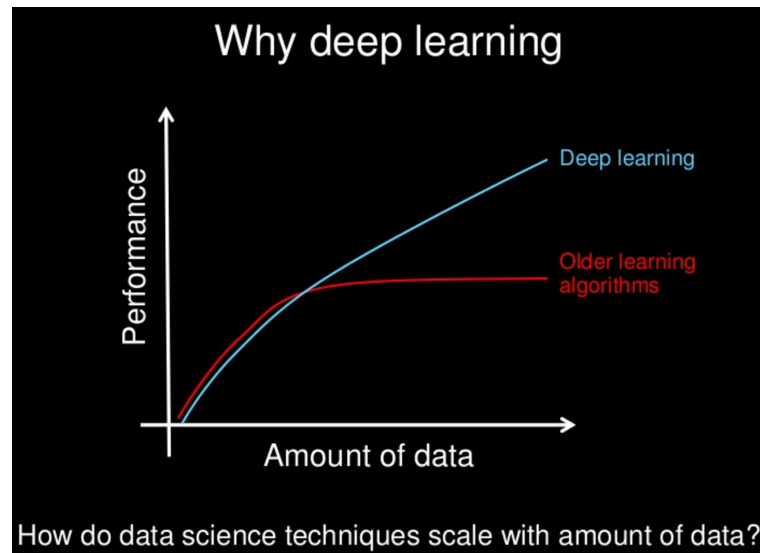
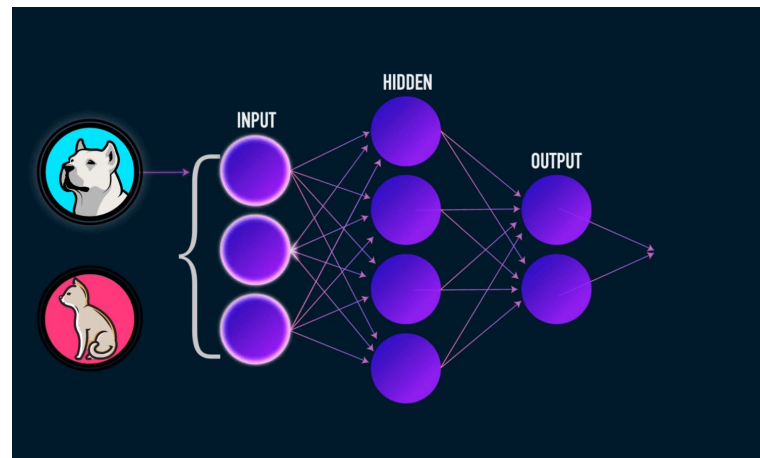


Compiled By Nitish Bhardwaj

<https://www.linkedin.com/in/nitish-bhardwaj/>

Deep Learning

- Based on Concept on Artificial Neural Networks
- *"The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw **a graph showing how these concepts are built on top of each other**, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning."* - Ian GoodFellow



Applications



MACHINE INTELLIGENCE 3.0

ENTERPRISE INTELLIGENCE

VISUAL Orbit Insight Clarifai Cortica Space Know Netra Deepomatic	AUDIO Gridspace Nvidia Capiio Clover QirouAI	SENSOR Predix Sentient Uptake Thingworx	INTERNAL DATA Palantir Alation Digital Reasoning	MARKET Mattermark Boonless QTracks
--	--	--	--	--

ENTERPRISE FUNCTIONS

CUSTOMER SUPPORT Digital Genius Kasisto Eloquent ActionIQ Zendesk Pheed Clarabridge	SALES Collective Fuse Salesforce Zeeleaf	MARKETING Lattice Brightfunnel Cobound	SECURITY Darktrace Zimperium Sentinel Graphistry	RECRUITING Textio Wade & Wendy Univio Gigster
---	---	--	---	--

AUTONOMOUS SYSTEMS

GROUND NAVIGATION Drive Zoox Uber Autonomy	AERIAL Skydio Airware Pilot AI	INDUSTRIAL Jaybridge Clearpath Kinross Fetch Robotics	PERSONAL Amazon Alexa Cortana Facebook Siri Replica	PROFESSIONAL Butter.ai Clara Talla Zoom Sudo
---	--	--	---	--

INDUSTRIES

AGRICULTURE Blue River Tule Zipprian Udacity	EDUCATION Knewton CTI Udacity	INVESTMENT Bloomberg Sentient iSentium AlphaSense Generellum Capital	LEGAL Blue J Everlaw Seal Legal Robot	LOGISTICS Nauto Priteck Routific Marble Pitstop
---	---	--	--	---

INDUSTRIES CONT'D

MATERIALS Zymergen Eigen Innovations Bright Machine Amgen Calculario	RETAIL FINANCE Tala Lendy Affirm Wealthfront	PATIENT Pulse Zephyr Health Onco Atomwise	IMAGE Butterfly Arterys Baylabs Google DeepMind	BIOLOGICAL CarbonX Color Grail Recursion Luminate Atomwise
--	---	--	--	---

TECHNOLOGY STACK

AGENT ENABLERS

OCTANE.AI
 OpenAI Gym
 Kasisto
 KITT.AI
 AUTOMAT
 semantic machines

DATA SCIENCE

DOM NO
 SPARKBEYOND
 kaggle
 DataRobot
 yhat
 AYASDI
 data iku
 seldon
 yseop
 big ml

MACHINE LEARNING

CognitiveScale
 Google ML
 context relevant
 Qyber
 HyperScience
 NORO
 minds.ai
 H2O AI
 SCALED INFERENCE
 sparkcognition
 loop
 GEOMETRIC INTELLIGENCE
 deepsense.io
 reactive
 sky mind
 bonsai

NATURAL LANGUAGE

ogolo
 PYLIEN
 LEXALYTICS
 Narrative Science
 spaCy
 LUMINOSO
 cortical.io
 MonkeyLearn

DEVELOPMENT

SIGOPT
 HyperOpt
 fuzzyo
 pkite
 rainforest
 lobe
 Anodot
 Signifai
 LAYER
 bonsai

DATA CAPTURE

CrowdFlower
 diffbot
 CrowdAI
 import
 Paxata
 DATASIFT
 amazon mechanical turk
 enigma
 WorkFusion
 DATALOGIX
 TRIFACTA
 parsehub

OPEN SOURCE LIBRARIES

Keras
 Chainer
 CNTK
 TensorFlow
 Caffe
 H2O
 DEEPLARNING.AI
 theano
 torch
 DSSTNE
 scikit-learn
 Azure ML
 neon
 MXNet
 DMTK
 Spark
 PaddlePaddle
 WEKA

HARDWARE

KNUPATH
 TENSORTORRENT
 Cinnascale
 NVIDIA
 nervana
 Movidius
 tenSilica
 Google TPU
 IO Labs
 Cerebras
 Issemi

RESEARCH

OpenAI
 Knoggin
 Numenta
 Kimera Systems
 ELEMENT
 vicarious
 Cogsci

Deep learning with Images : Computer Vision

Classification



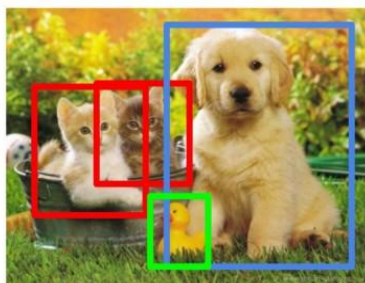
CAT

**Classification
+ Localization**



CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**



CAT, DOG, DUCK

Single object

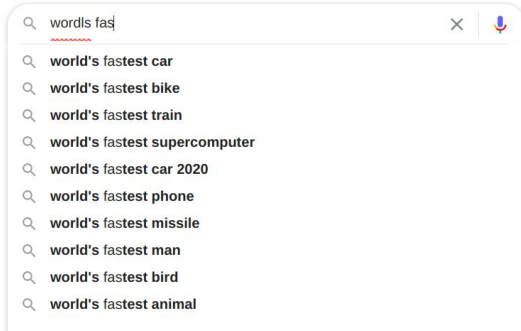
Multiple objects

- Classification
- Detection
- Segmentation
- Generation
- And Lot more

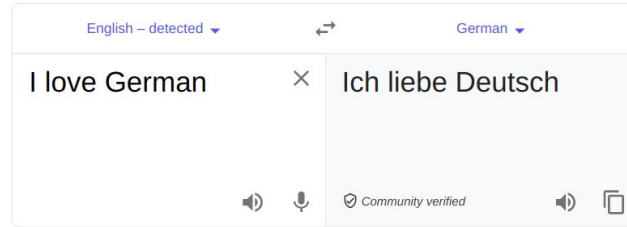


Deep Learning with Text : Natural Language Processing

Search Autocorrect and Autocomplete



Language Translation



- Text Classification
- Chatbots
- Voice Assistants
- Social Media Analytics
- Advertisements
- Summarization
- And lot more



Deep Learning with Number data :

Stock Market Prediction

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

- House Price Prediction
- Recommendation system
- Disease Identification
- Medical Reporting
- And lot more

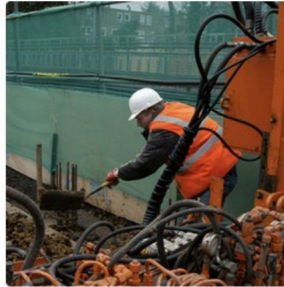


Complex Deep Learning

Image Captioning



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



Images

Text




Deep Neural Networks



Dataset

Inputs

Output



x1	x2	x3	y
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

PROBLEM SOLVING IN 7 STEPS

1 Classification Problem

2 Data Preparation

1. Problem Statement

2. Data Preparation

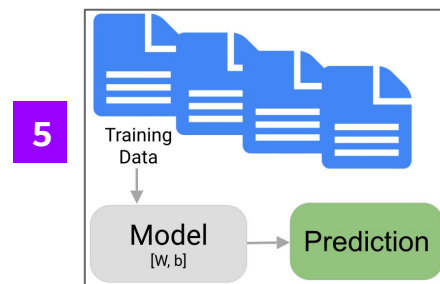
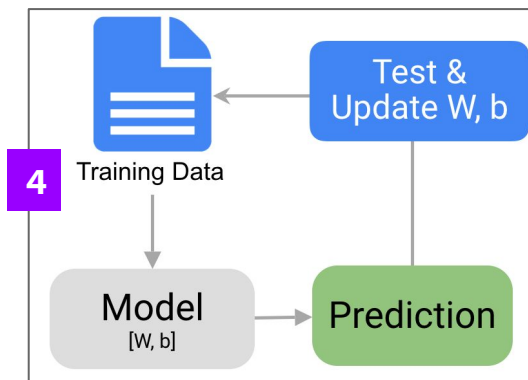
3. Choosing a **model, $y=f(x)$**

4. **Training**

5. Evaluation / **Validation**

6. Hyperparameter tuning

7. Prediction/**Testing**



3

6

$$y = m * x + b$$

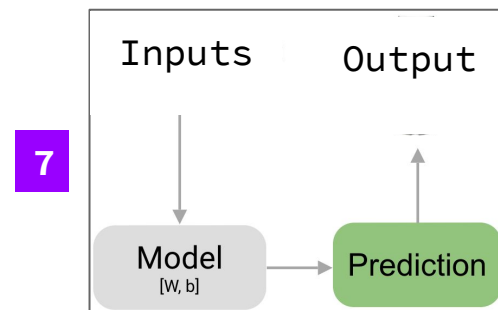
OUTPUT SLOPE INPUT Y-INTERCEPT

WEIGHTS =

$$\begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \\ m_{3,1} & m_{3,2} \end{bmatrix}$$

BIASES =

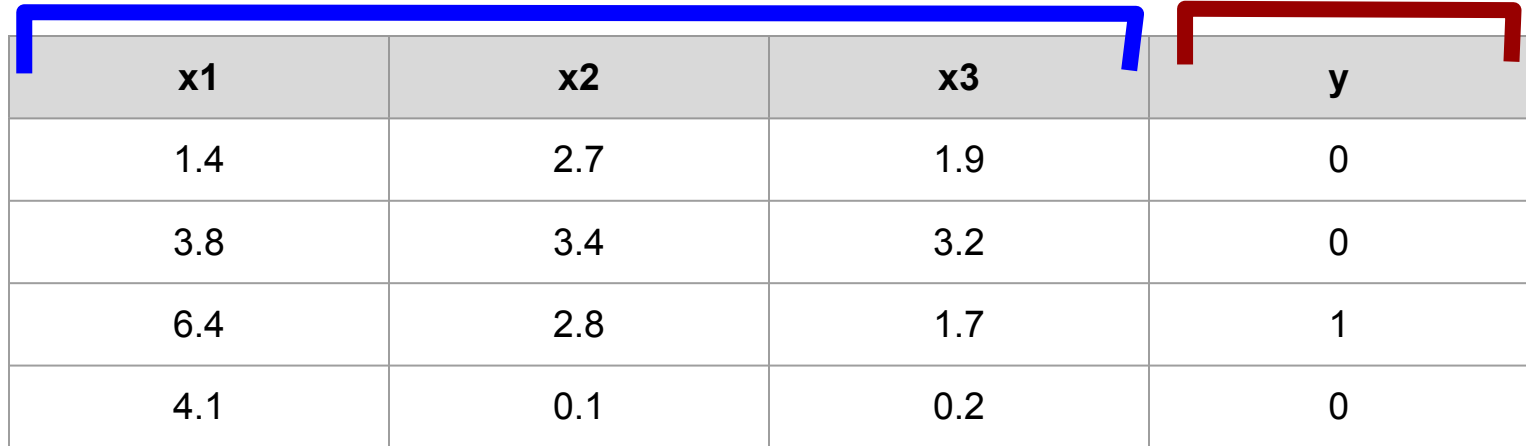
$$\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix}$$



Problem Solving :

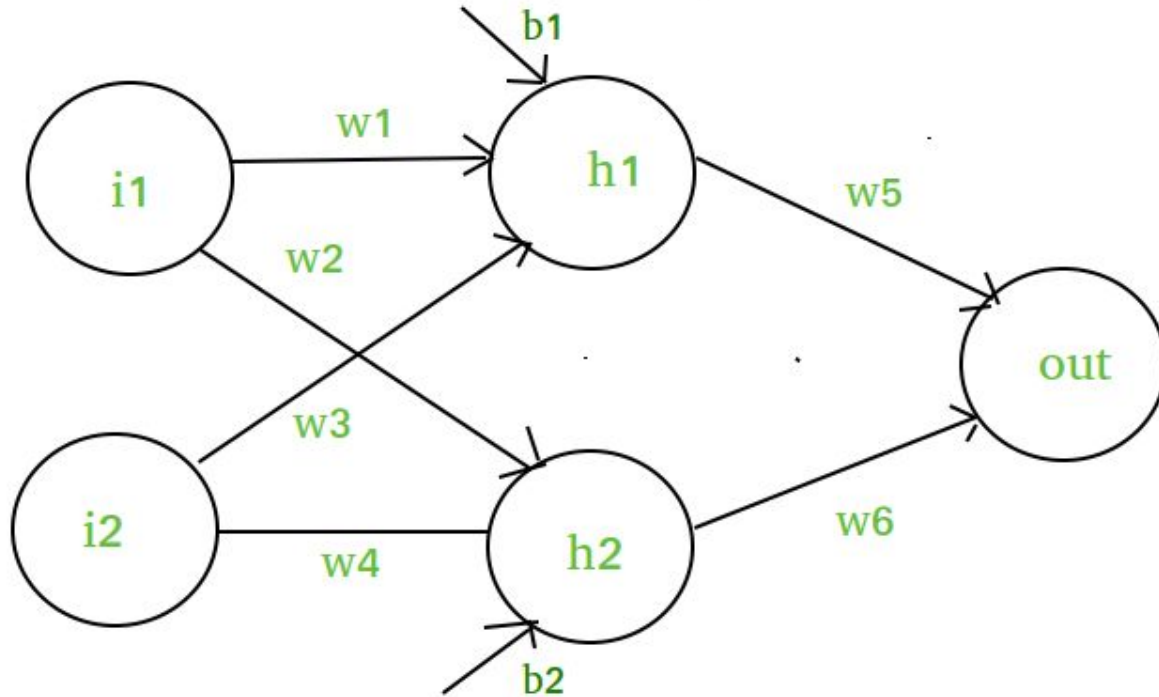
Inputs

Output



x1	x2	x3	y
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

Weights and Biases in Neural network



A dataset

<i>Fields</i>	<i>class</i>
---------------	--------------

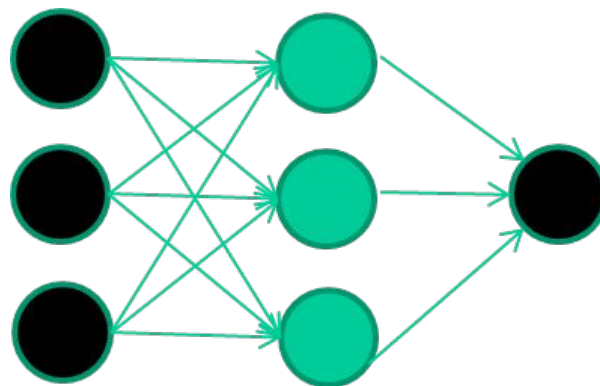
1.4 2.7 1.9	0
-------------	---

3.8 3.4 3.2	0
-------------	---

6.4 2.8 1.7	1
-------------	---

4.1 0.1 0.2	0
-------------	---

etc ...



Training the neural network

Fields ***class***

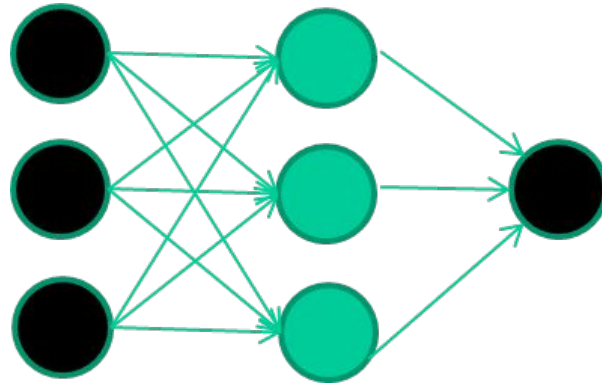
1.4 2.7 1.9 0

3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...



Training data

Fields ***class***

1.4 2.7 1.9 0

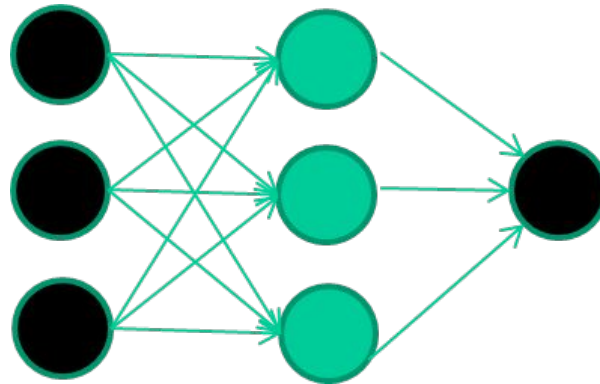
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Initialise with random weights and bias



Training data

Fields *class*

1.4 2.7 1.9 0

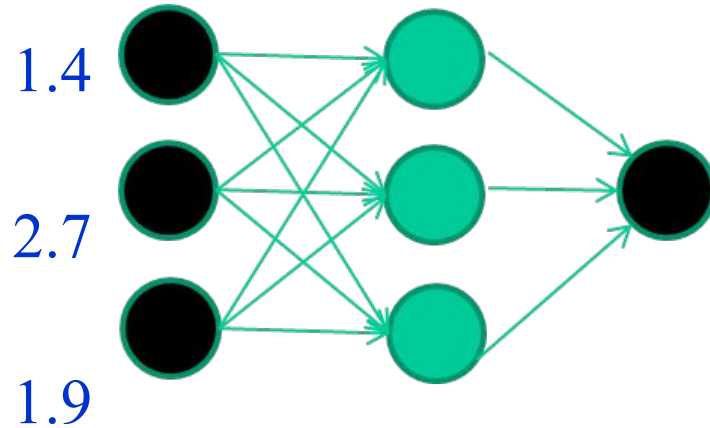
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Present a Training Pattern



Training data

Fields *class*

1.4 2.7 1.9 0

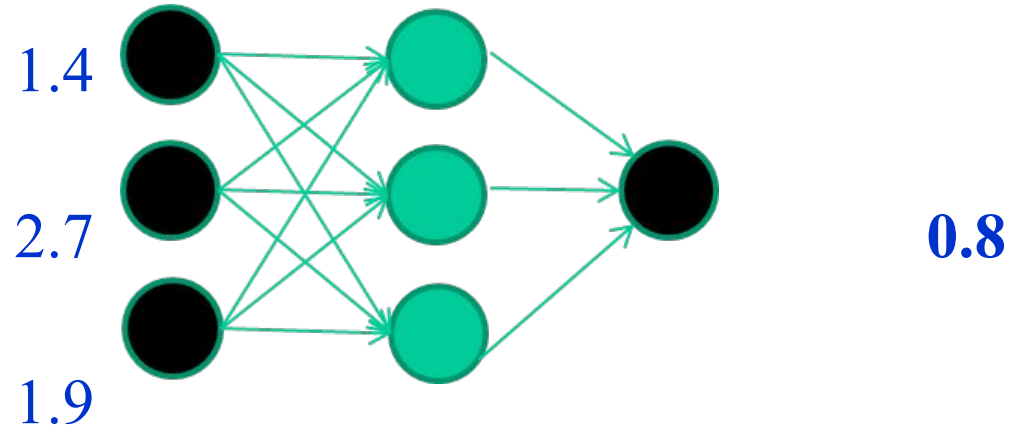
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Feed it through the output



Training data

Fields *class*

1.4 2.7 1.9 0

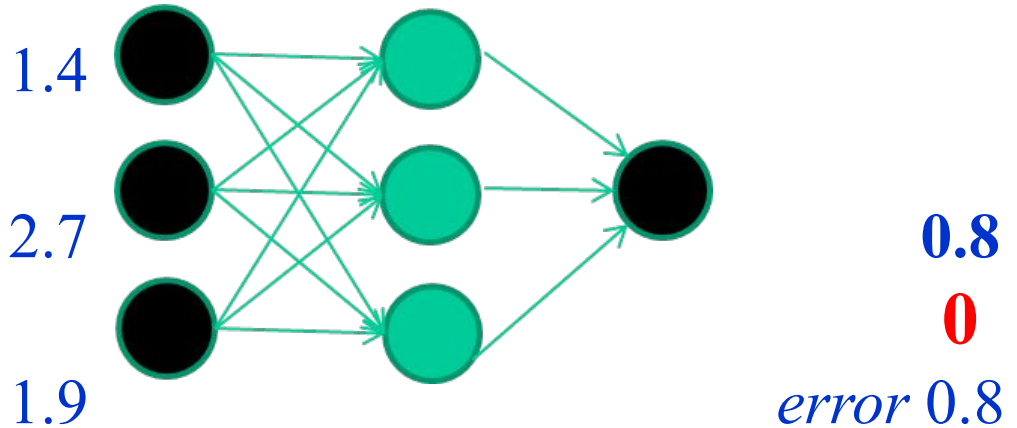
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Compare with target output



Training data

Fields *class*

1.4 2.7 1.9 0

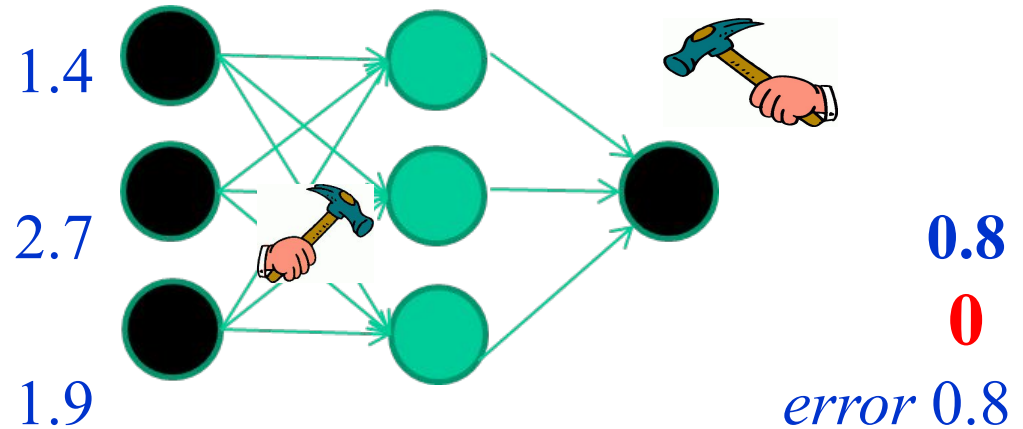
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Adjust weights based on error



Training data

Fields ***class***

1.4 2.7 1.9 0

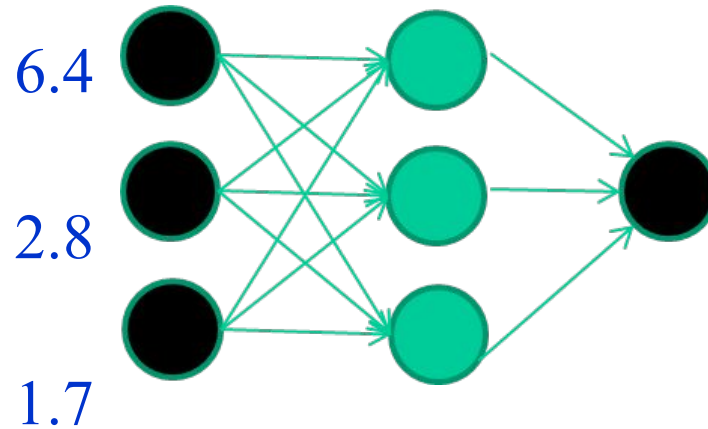
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Present a Training Pattern



Training data

Fields ***class***

1.4 2.7 1.9 0

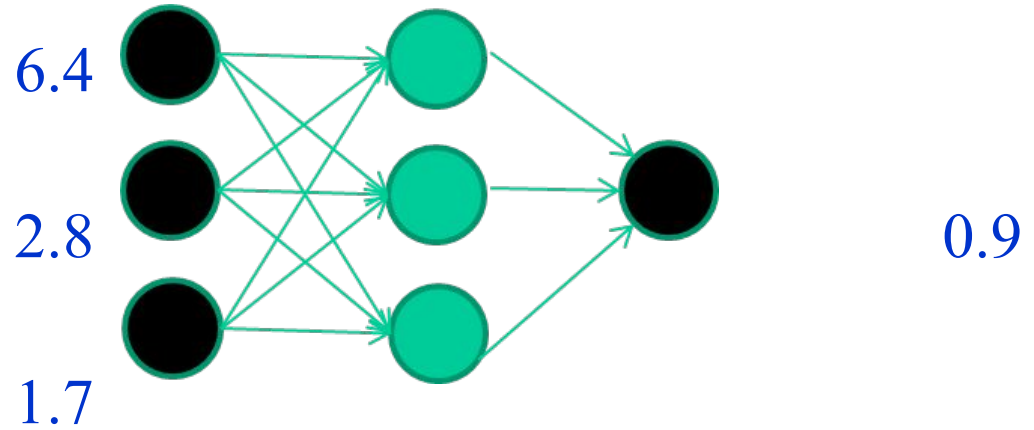
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Feed it through the output



Training data

Fields ***class***

1.4 2.7 1.9 0

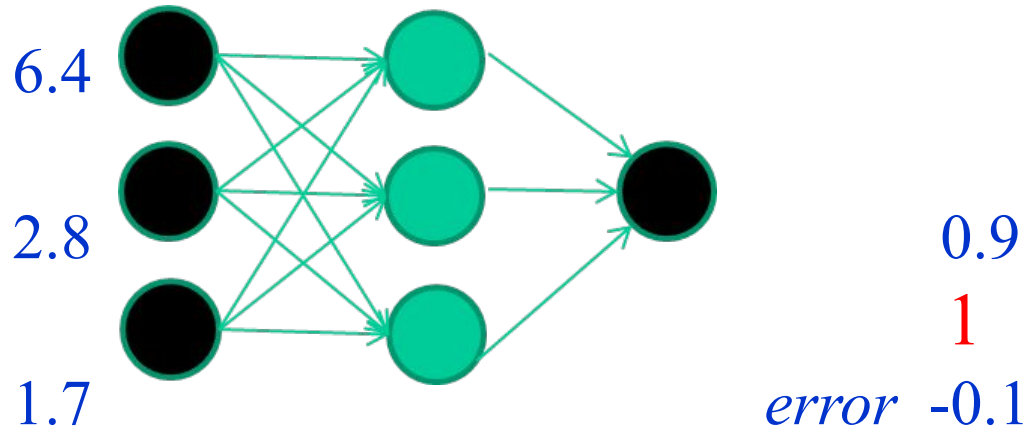
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Compare with target output



Training data

Fields ***class***

1.4 2.7 1.9 0

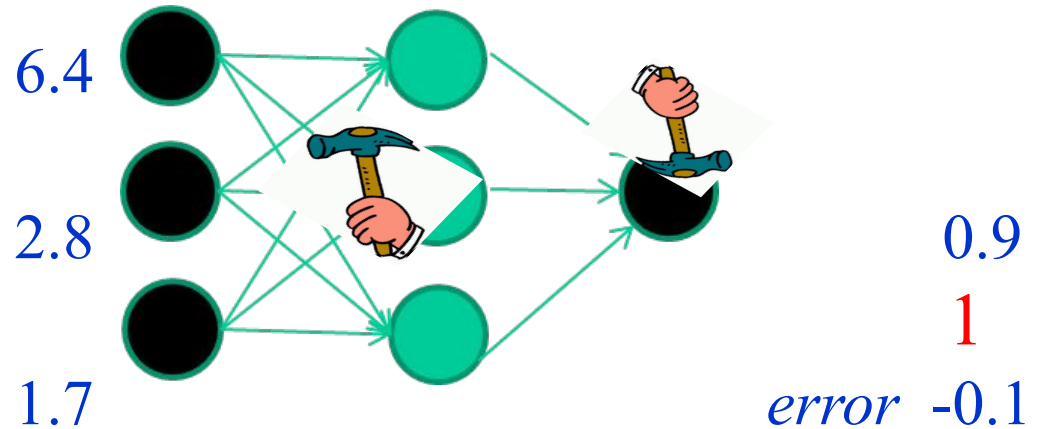
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

Adjust weights based on error



Training data

Fields ***class***

1.4 2.7 1.9 0

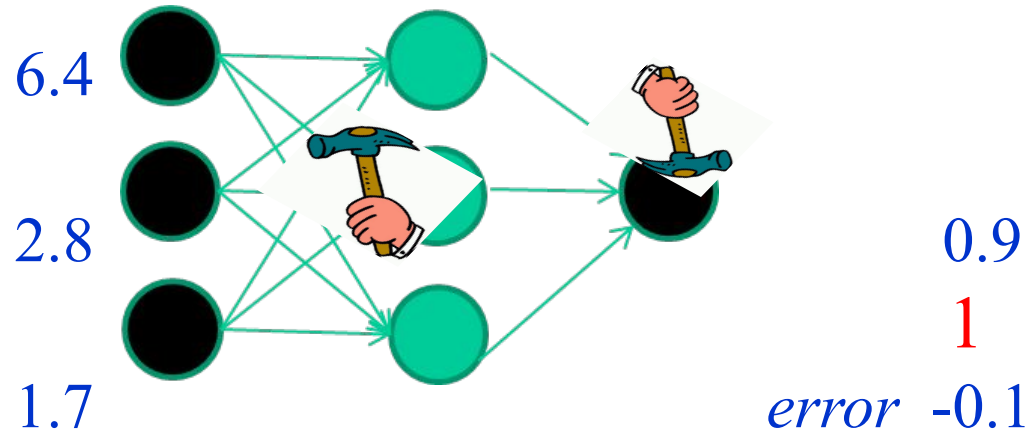
3.8 3.4 3.2 0

6.4 2.8 1.7 1

4.1 0.1 0.2 0

etc ...

And so on



Repeat this thousands, maybe millions of times – each time taking a random training instance, and making slight weight adjustments

Algorithms for weight adjustment are designed to make changes that will reduce the error

Coding Session

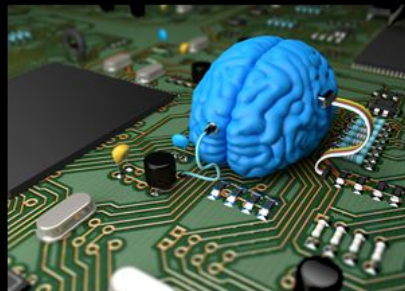


Let's Deep Dive

Deep Learning



What society thinks I do



What my friends think I do



What other computer
scientists think I do



What mathematicians think I do



What I think I do

```
In [1]:  
import keras  
Using TensorFlow backend.
```

What I actually do

LET'S CODE...



Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.

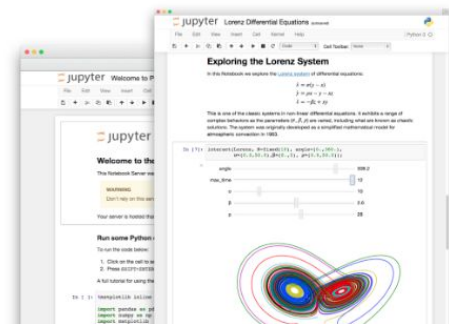


Image Classification



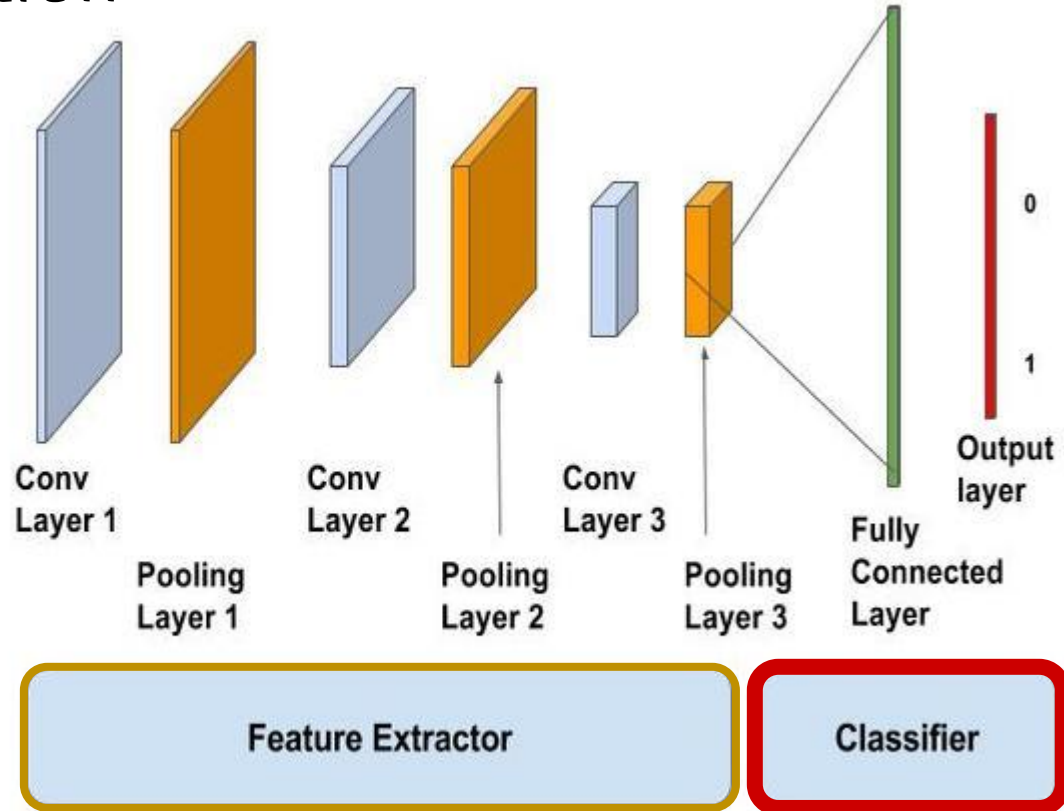
Image classification

Dataset

- Images
- Classes



Input



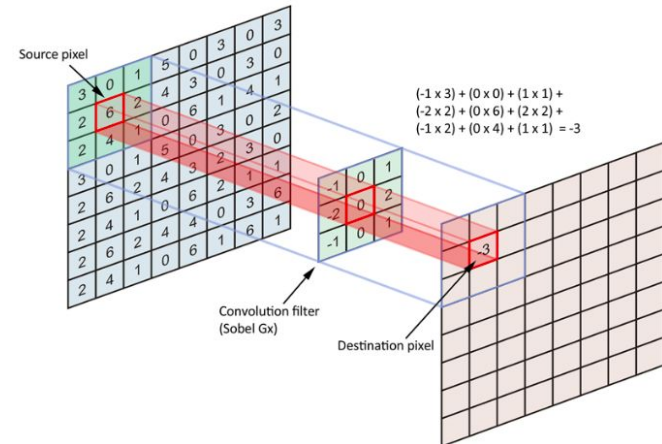
Feature extraction

- HOG: Histogram of Oriented Gradients
- SIFT: Scale Invariant Feature Transform
- SURF: Speeded-Up Robust Feature
- **CNN: Convolution Neural Network**

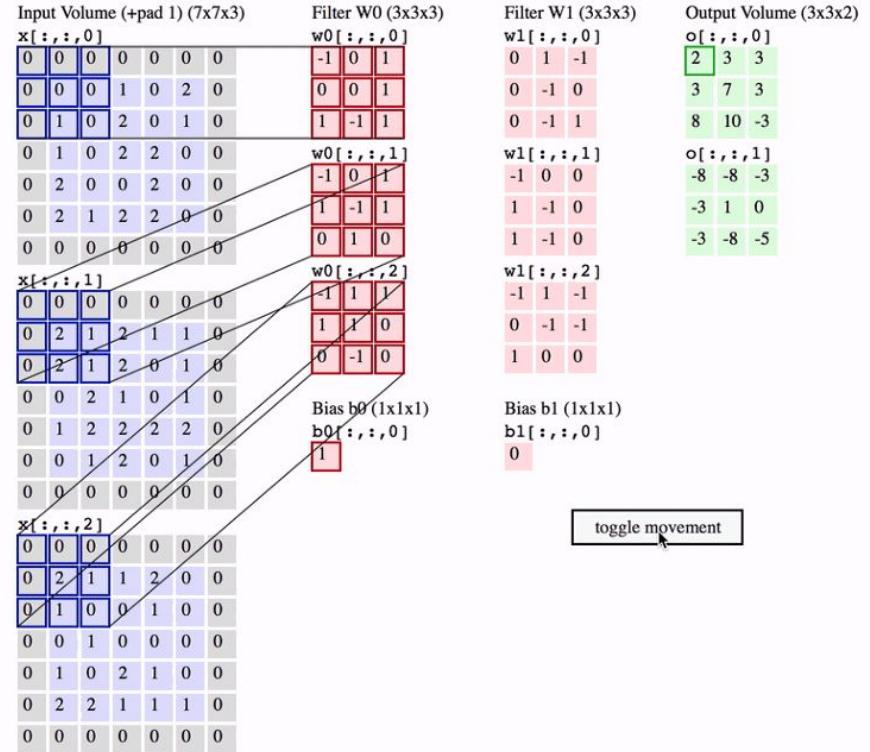


1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		



Convolution neural Network



Object Detection

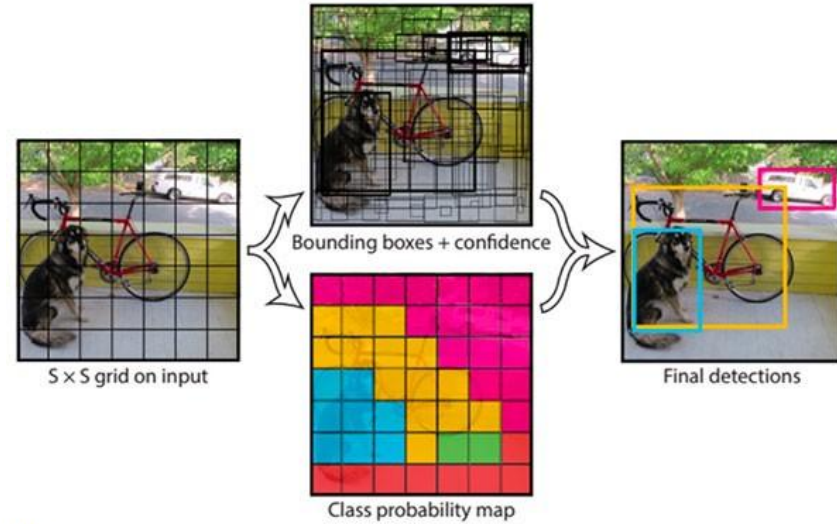
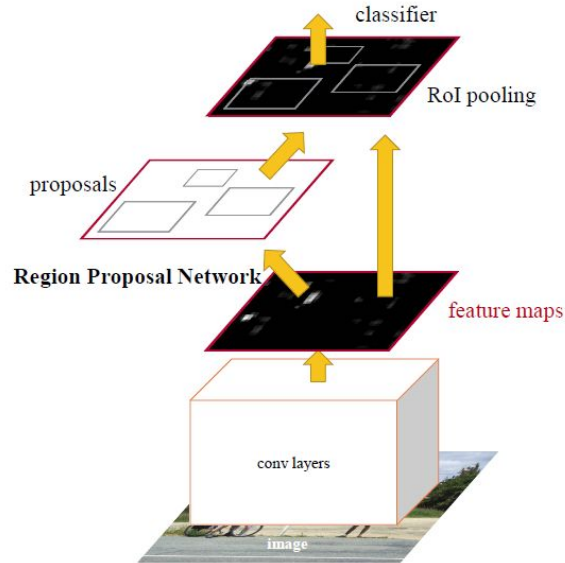


Object detection

Dataset

- Images
- Bounding Boxes(x,y,w,h)
- Classes

Region Proposal Network, providing a number of regions which are then passed to common DL based classification architectures



With the need of real time object detection, many one-step object detection architectures have been proposed, like YOLO, YOLOv2, YOLOv3, SSD, RetinaNet etc. which try to combine the detection and classification step.

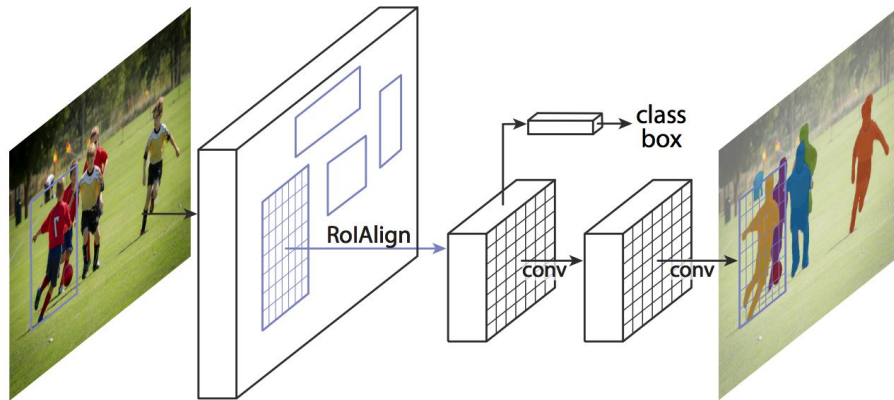
Examples



Segmentation

Dataset

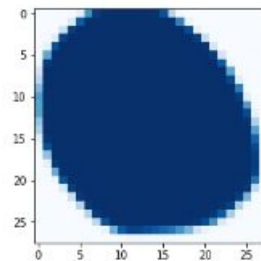
- Images
- Coordinates[list of (x,y)]
- Classes



Segmentation Masks

The mask branch is CNN that takes the positive regions selected by the ROI classifier and generates masks for them.

The generated masks are low resolution: 28x28 pixels. But they are soft masks, represented by float numbers, so they hold more details than binary masks.



28x28 Soft Mask



Resized Binary Mask

Face recognition

Dataset

- Images
- Classes



Facebook automatically tags people in your photos that you have tagged before. I'm not sure if this is helpful or creepy! :P

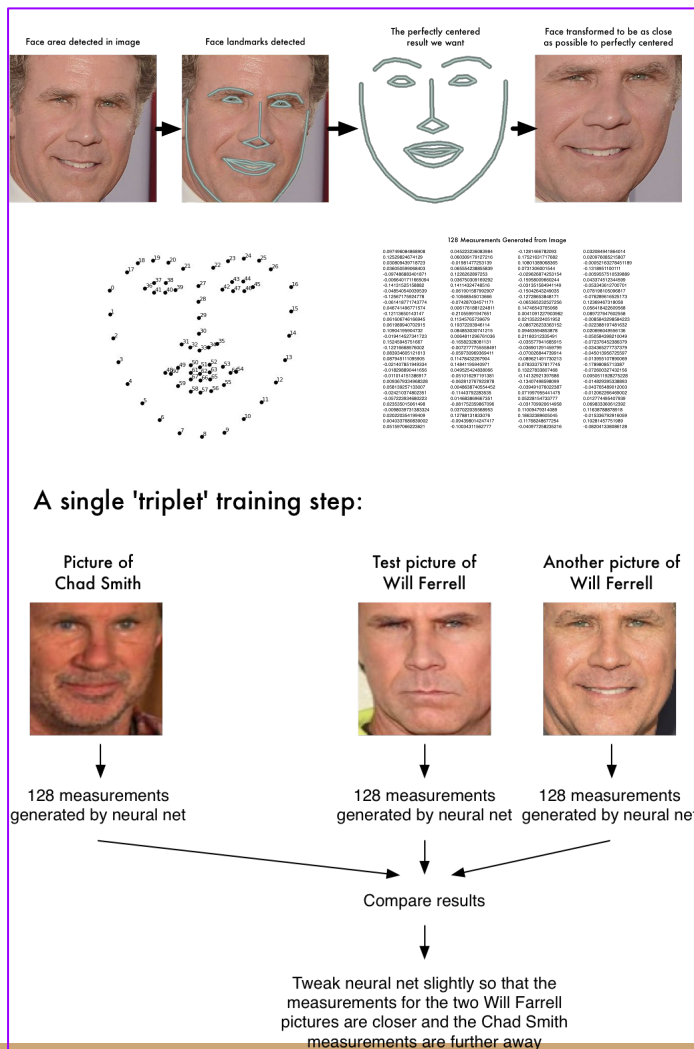
Algorithm:

- Face Detection
- Face Alignment

- DeepNet Model to extract features

Pass the centered face image through a neural network that knows how to measure features of the face. Save those 128 measurements.

- *Testing new face : Extract 128 measurements and find the most similar match*



Style transfer

- Problem Statement
 - Given:
 - Content Image
 - Style Image
 - Result:
 - Stylized Image
- Algorithm
 - Feature Extraction of Content (A)
 - Feature Extraction of Style (B)
 - Merge Features(A+B)

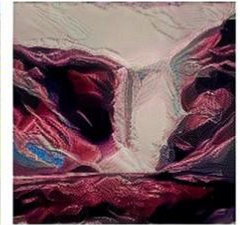
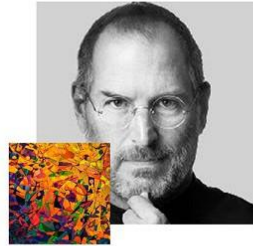
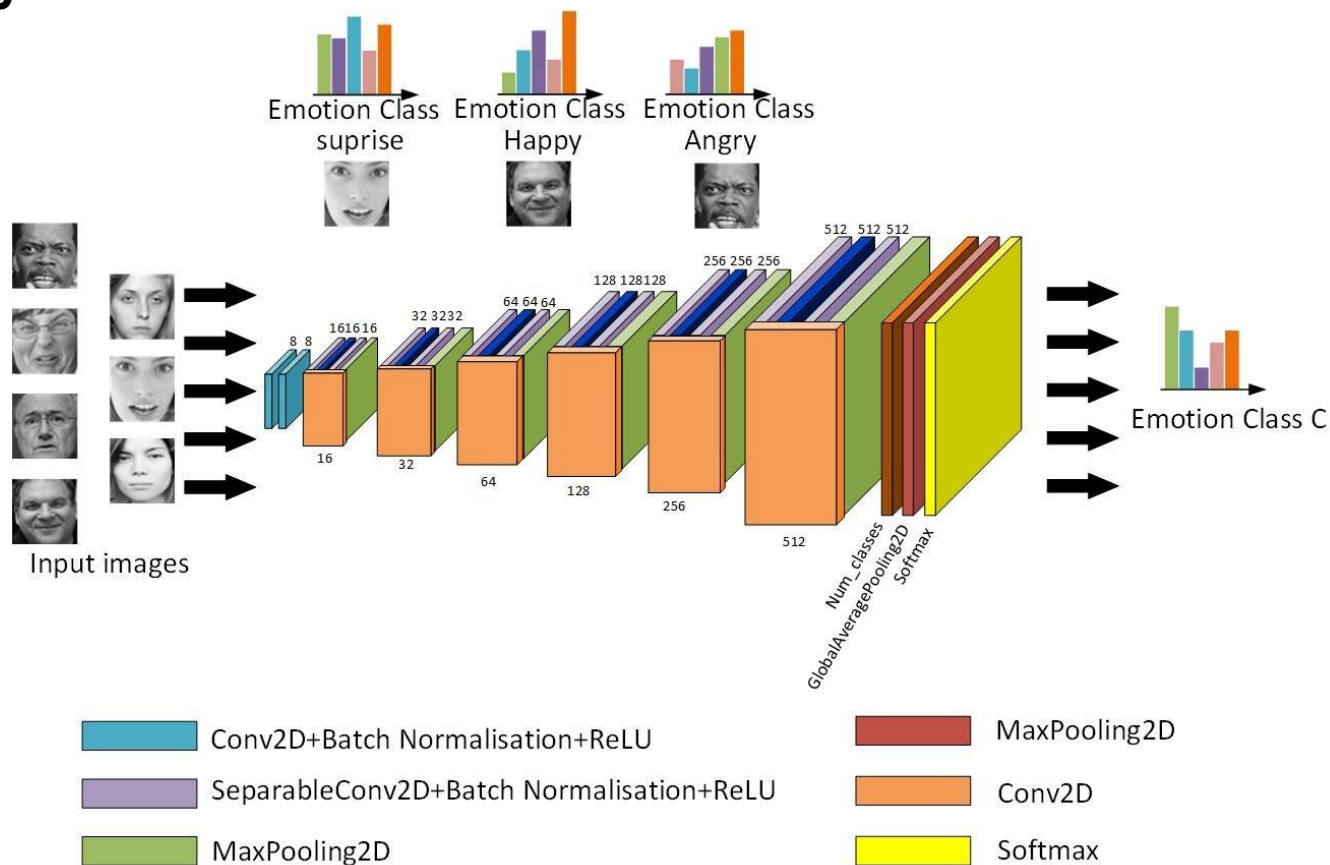


Image analytics

1. Problem Statement
2. Data Preparation
3. Choosing **model(s)**
 - a. Training
 - b. Evaluation / **Validation**
 - c. Hyperparameter tuning
 - d. Prediction/**Testing**
4. Combine Results
5. Presentation



Summary

- AI to Deep Learning
- Neural networks to Deep nets
- Problem Solving
- Coding session
 - Image Classification
 - Convolutional Neural network
 - Object Detection
 - Segmentation
- More examples



References

- <https://setosa.io/ev/image-kernels/>
- <https://poloclub.github.io/cnn-explainer/>
- <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>
- <https://www.youtube.com/watch?v=nKW8Ndu7Mjw>
- <https://www.thispersondoesnotexist.com/>
- <https://playground.tensorflow.org/>
- <https://reiinakano.com/arbitrary-image-stylization-tfjs/>
- <https://blog.insightdatascience.com/generating-custom-photo-realistic-faces-using-ai-d170b1b59255>
- <https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>
- https://github.com/ageitgey/face_recognition
- https://medium.com/@manivannan_data/how-to-train-yolov2-to-detect-custom-objects-9010df784f36

Tutorials

- https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
- <https://keras.io/examples/>
- <https://www.tensorflow.org/learn>
- <https://www.youtube.com/watch?v=CU6bTEClzlw>
- <https://www.youtube.com/c/K%C3%A1rlyZsolnai/videos>
-



Overview of Session :

- Deep Learning Introduction
- Application
- AI Pipeline
- Image Classification
- Object Detection
- Text Classification
- AI Pipeline
- Industrial Requirement
- Research Topics in Deep Learning
- GAN



Thank You