Introduction to Neural Networks

Wing Yue Geoffrey Louie and Suraj Goyal
ECE 5900, HRI – Fall 2019

Introduction to Computer Vision

- It is a technique used to help computers understand/ visualize the world like humans do.
- Primarily used for images and videos
- Alex-net won the 2012 Image-net LVSRC competition by a tremendous margin using CNN architecture. Network achieved a top 5 error of 15.3% which was around 10% better.
- Computer vision advancements have been mainly due to deep learning
- Deep learning computer vision is helping self-driving cars identify the other cars, pedestrians, traffic signs to be able to maneuver accordingly.
- I-phones already have face recognition functionality to be able to unlock a phone.
- Pictures in phones are getting tagged with emotions to say if you are feeling sad, happy etc.
- Deep learning is even enabling new types of art to be created.
- Therefore, this is a super exciting time to be working in this field.

What is Computer Vision?

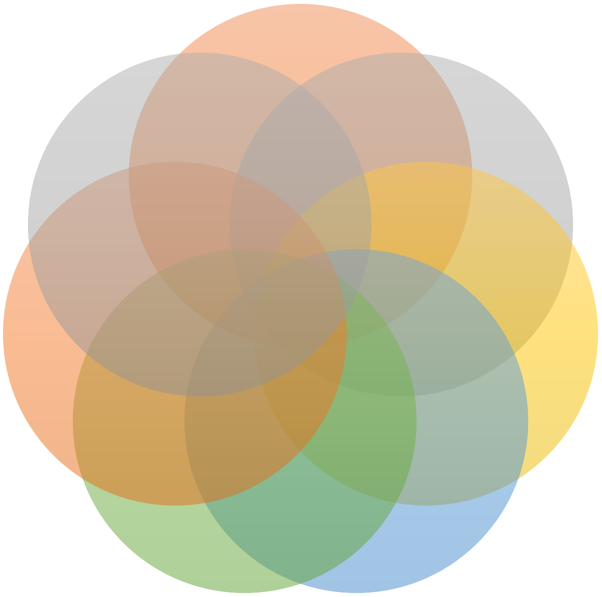


Image Processing

Cognitive Science

Sensors

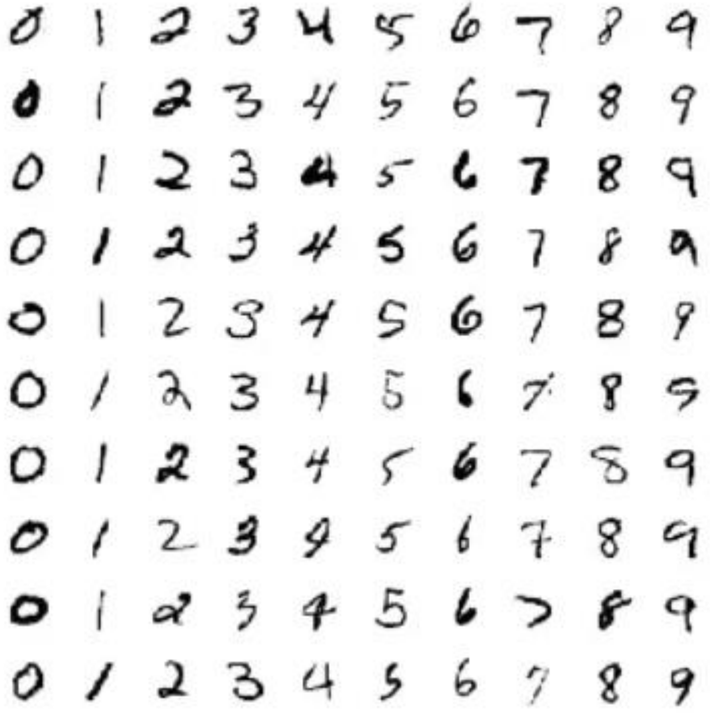
Machine Learning

Artificial Intelligence

Mathematics

Introduction to Computer Vision

Human Brain is truly amazing!



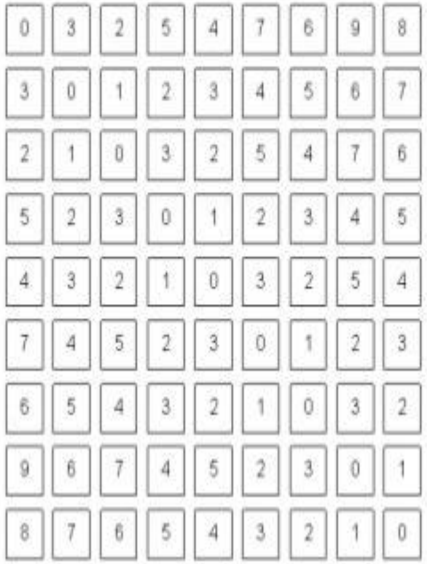
This is where things get complicated.

What seems relatively simple for the human visual cortex is not as simple if you have to program a 28*28 grid to identify digits from 0-9

Reference: http://yann.lecun.com/exdb/mnist/

Human Vision vs Computer Vision

What we see



0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

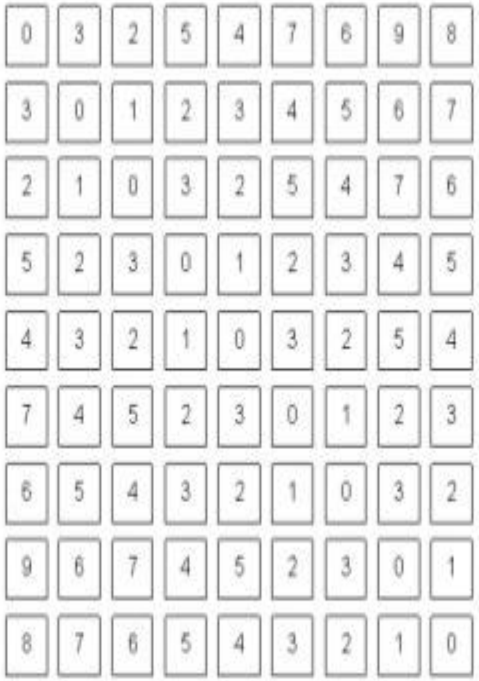
Human vision performs complex visualization easily & efficiently. Visual understanding goes well beyond object recognition. With one glance at an image, we can effortlessly imagine the world beyond the pixels: for instance, we can infer people’s actions, goals, and mental states. While this task is easy for humans, it is tremendously difficult for today’s vision systems, requiring higher-order cognition and commonsense reasoning about the world.

Reference: https://steemit.com/science/@stormblaze/how-does-a-computer-detect-a-face

How does CV work?

- Computers interpret images as a series of pixels with their own respective color values.
- Each pixel in an image can range from numbers 0-255.
- Colored images have 3 color channels as Red, Green and Blue (RGB)

What we see



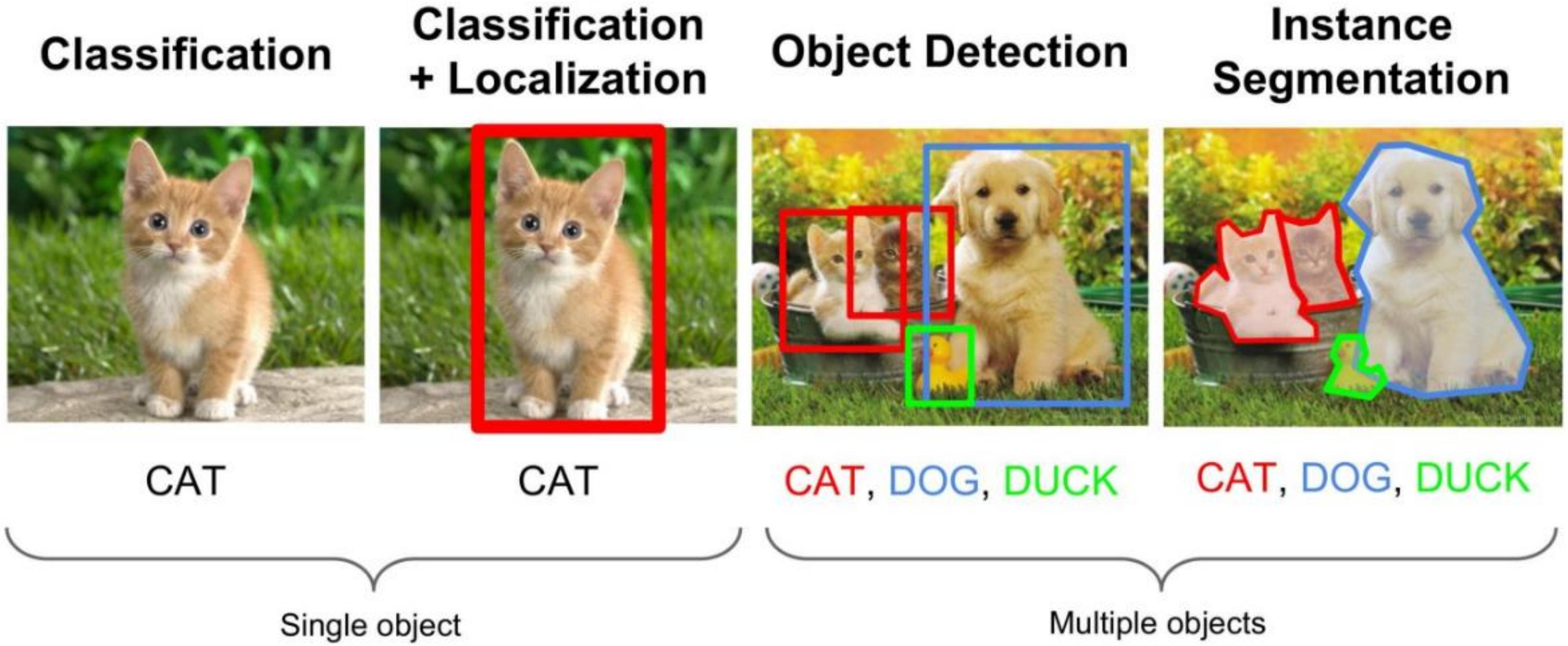
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

References: https://steemit.com/science/@stormblaze/how-does-a-computer-detect-a-face

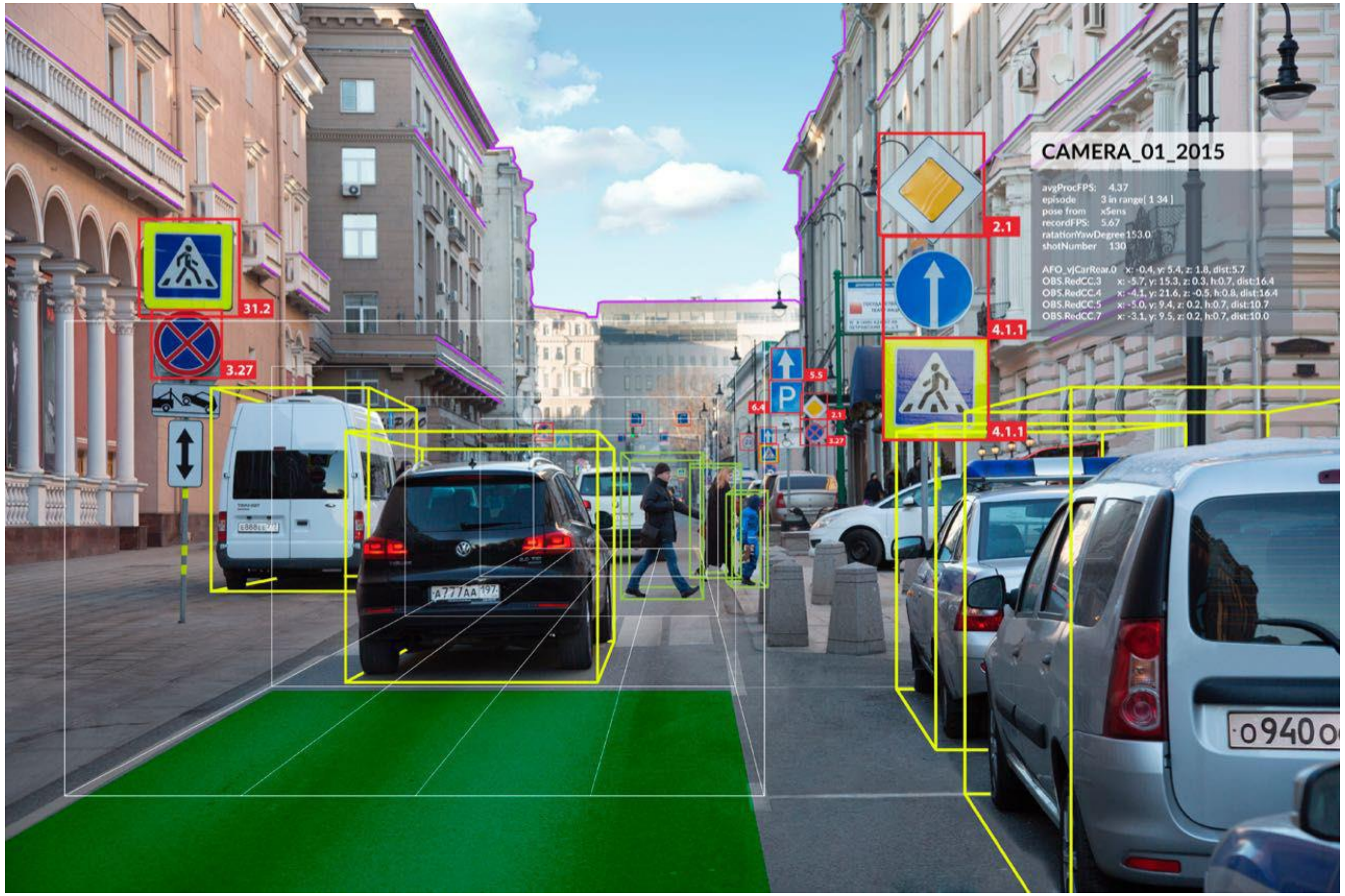
Object Detection, Localization and Segmentation

- Detection or localization involves an object in an image and localizing with a bounding box.



Reference: https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Object Detection is not enough

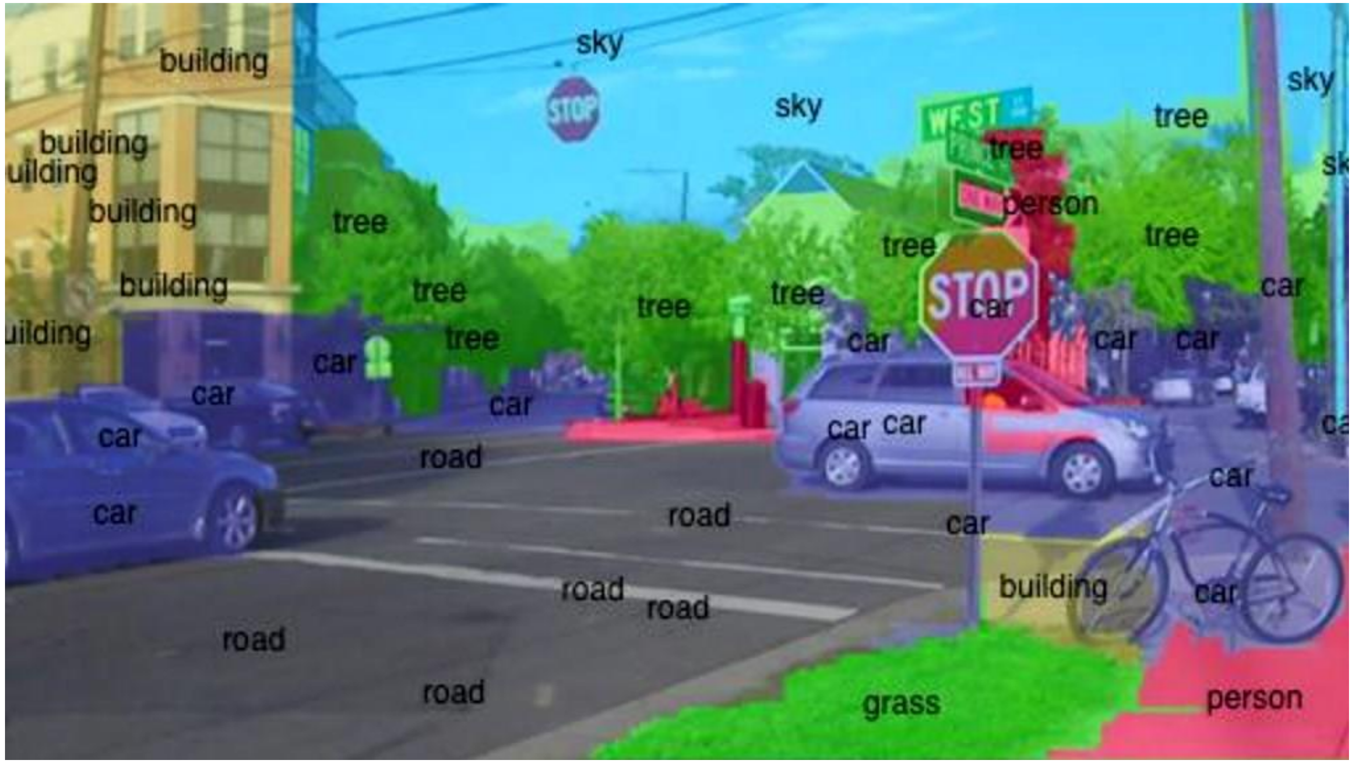


```
CAMERA_01_2015

avgProcFPS:     4.37
episode         3 in range( 1 34 )
pose from       xSens
recordFPS:      5.67
ratationYawDegree 153.0
shotNumber      130

AFO_vjCarRear.0   x: -0.4, y: 5.4, z: 1.8, dist:5.7
OBS.RedCC.3       x: -5.7, y: 15.3, z: 0.3, h:0.7, dist:16.4
OBS.RedCC.4       x: -4.1, y: 21.6, z: -0.5, h:0.8, dist:16.4
OBS.RedCC.5       x: -5.0, y: 9.4, z: 0.2, h:0.7, dist:10.7
OBS.RedCC.7       x: -3.1, y: 9.5, z: 0.2, h:0.7, dist:10.0
```

<u>Needs Classification!</u>

Reference: https://www.linkedin.com/pulse/eat-pray-computer-vision-what-does-mean-big-mac-iulia-feroli/

Semantic Segmentation

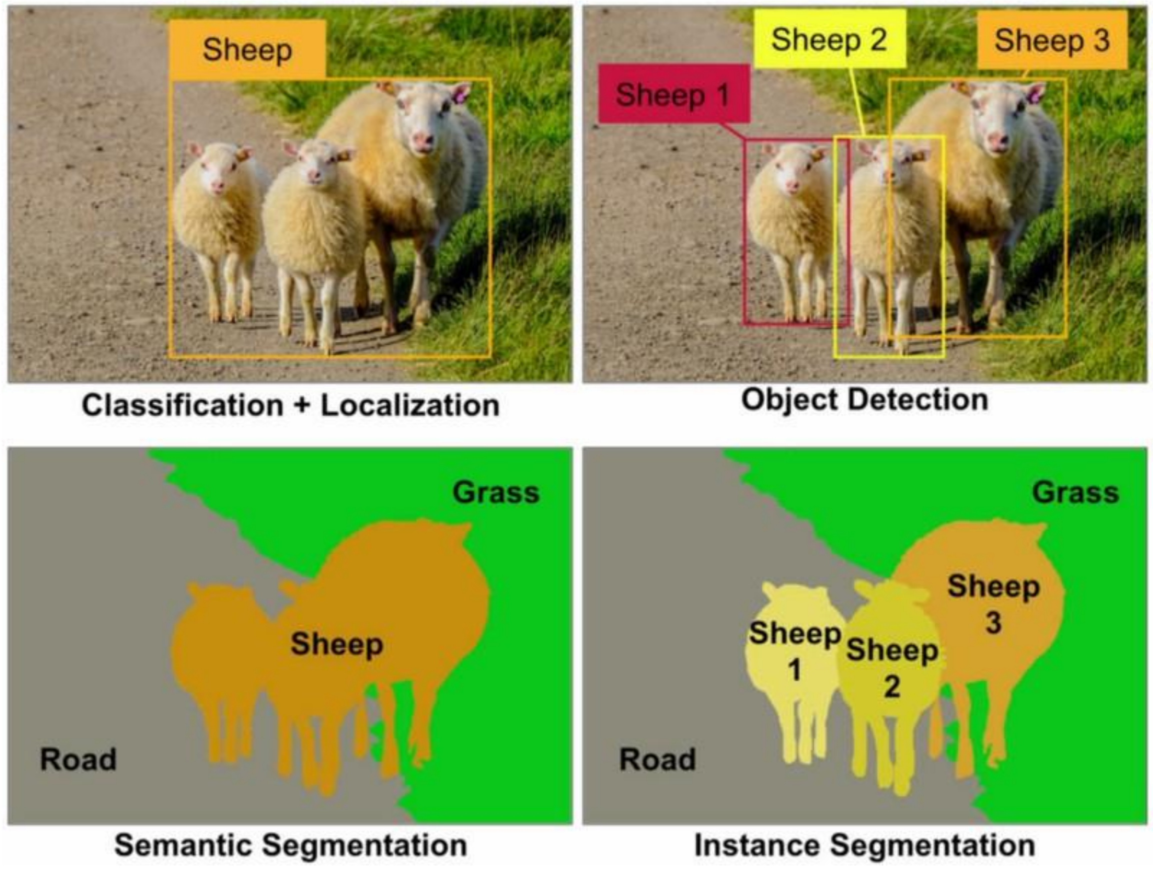


- This involves pixelwise classification based on which different objects are categorized.

Note: Semantic segmentation doesn't distinguish between the same category of object.

Reference: https://blog.algorithmia.com/introduction-to-computer-vision/

Semantic Segmentation



Reference: https://towardsdatascience.com/detection-and-segmentation-through-convnets-47aa42de27ea

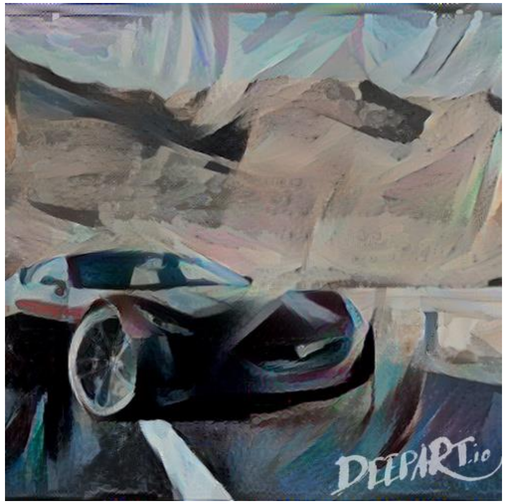
Neural Talking Head Models

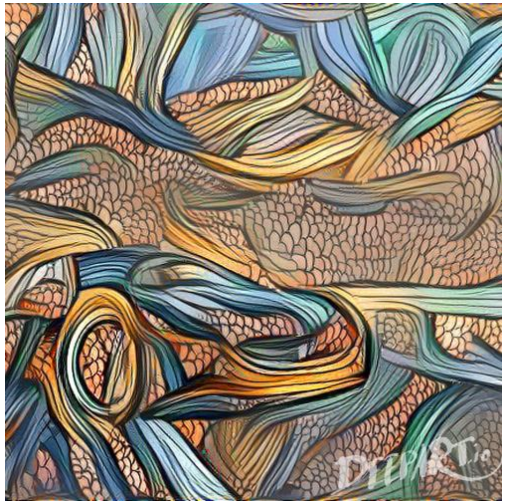
Reference: https://arxiv.org/abs/1905.08233

Style Transfer

New image generated by design adaptation of another image while preserving the original features of the image

Gatys algorithm using Convolutional Neural Networks (CNN)





References: https://deepart.io/
https://arxiv.org/pdf/1508.06576.pdf

Image Processing

- **Image processing** is the process of creating a new image from an existing image, by enhancement or modification.
- A given computer vision system may require image processing to be applied to raw input, e.g. pre-processing images.

- Examples of image processing include:
  - Normalizing photometric properties of the image, such as brightness or color.
  - Cropping the bounds of the image, such as centering an object in a photograph.
  - Removing digital noise from an image, such as digital artifacts from low light levels.

Typical pipeline for CV

- **Image acquisition** – A digital image is produced by a single or multiple image sensors. Based on the sensor there can be different measurement characteristics such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.
- **Pre-processing** – Before feature extraction, it is usually necessary to process the data to satisfy certain assumptions.

  Examples: Contrast enhancement, Re-sampling (coordinates), Noise reduction, and Re-scaling
- **Feature extraction** – Image features at various levels of complexity are extracted from the image data. Typical examples of such features are Lines, edges and ridges.

  Localized interest points such as corners, blobs, points or texture.
- **Detection/segmentation** – Regions of interest (ROI) in the image for further processing. Segmentation of image into foreground, object groups, single objects or salient object parts.
- **Classification and Verification** – Classifying an object into different categories and verifying that the data satisfies the model assumptions.
- Final output analysis

Reference: Davies, E. R. (2005). Machine vision : theory, algorithms, practicalities. Boston: Elsevier Science

Features?

- Features can be specific structures in the image such as points, edges or objects. Features are the information extracted from images in terms of numerical values that are difficult to understand by humans as computers do. Features can be categorized as:
- Local features: Features are also referred as descriptors. Local features are used for object recognition/identification. Detection is identifying if an object is present in an image/ video where as recognition is finding the object type. They identify the key-points in an image and represent the texture in an image patch. SIFT and SURF are some examples of local descriptors.
- Global features: Global features are used in image retrieval, object detection and classification. They describe the image as a whole. Global features include contour representations, shape descriptors etc. Histogram Oriented Gradients (HOG) and Co-HOG are some examples of global descriptors.

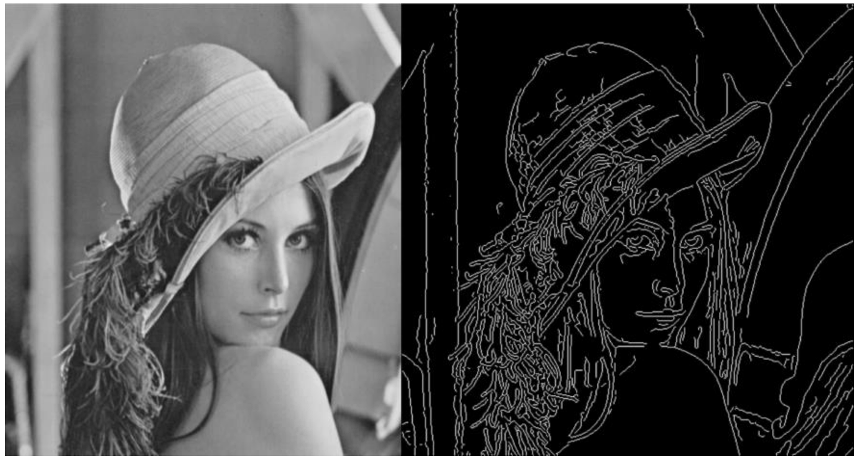
Features?

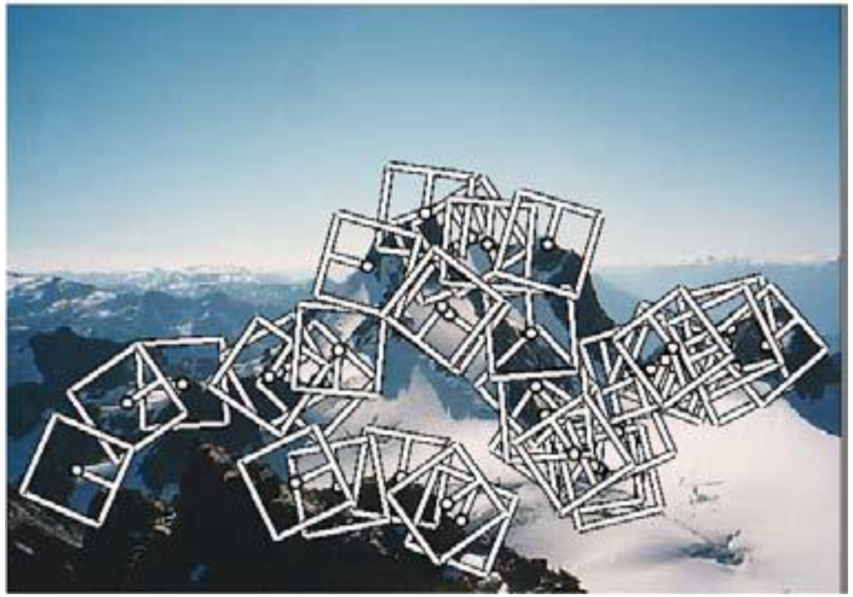
- Combination of global and local features can help improve the accuracy of recognition.
- Based on the use case it might not be sufficient to extract only one type of feature to obtain the relevant information from the image data.
- Multiple features are extracted, resulting in multiple feature descriptors at each image point.
- Then these descriptors as the elements of one single vector, commonly referred to as a feature vector.
- The set of all possible feature vectors constitutes a feature space.

Features

Features can also be categorized as:

- Edges
- Interest Points/ Corners
- Blobs



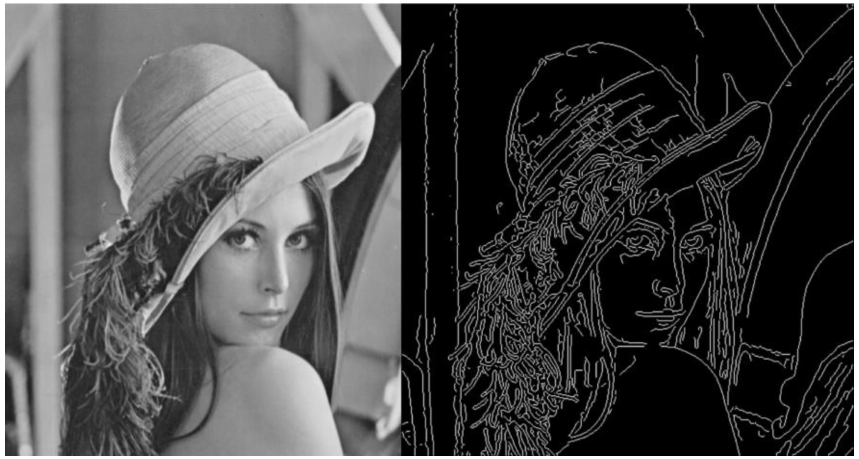


Reference: Computer Vision:Algorithms and Applications, Richard Szeliski

Image Filtering - Edges

Edges are sets of points in the image which have a strong gradient magnitude. Based on the strong gradient points edges are formed but may impact the properties of an edge, such as shape, smoothness, and gradient value. Locally, edges have a one-dimensional structure.

Applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image Example: Canny or Sobel



Reference: Computer Vision: Algorithms and Applications, Richard Szeliski

Image Filtering - Canny

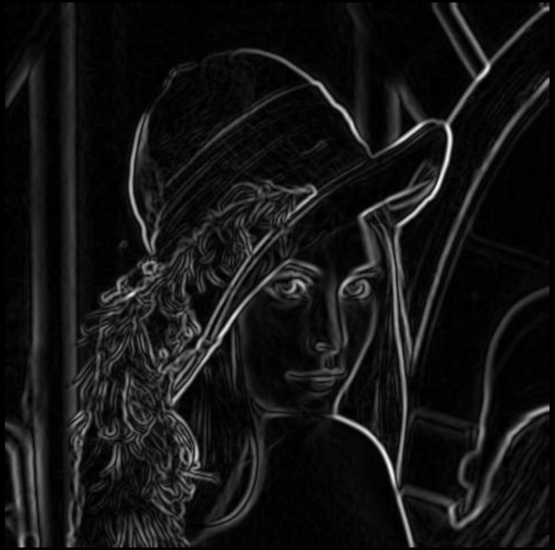
- Canny edge detection is a simple multi-step algorithm that can detect edges with noise suppressed at the same time. It consists of following steps:
- Apply the x and y derivatives of a Gaussian (convolve or low pass) filter to the image to eliminate noise, improve localization and have single response.
- Find the magnitude and orientation of the gradient at each pixel.
- Perform non-maximum suppression, which thins the edges down to a single pixel in width, since the extracted edge from the gradient after step 2 is quite blurry.
- Thresholding and linking, also known as hysteresis, to create connected edges where high to low threshold ratio of 1.5 is the recommended starting point.

Reference: Computer Vision: Algorithms and Applications, Richard Szeliski

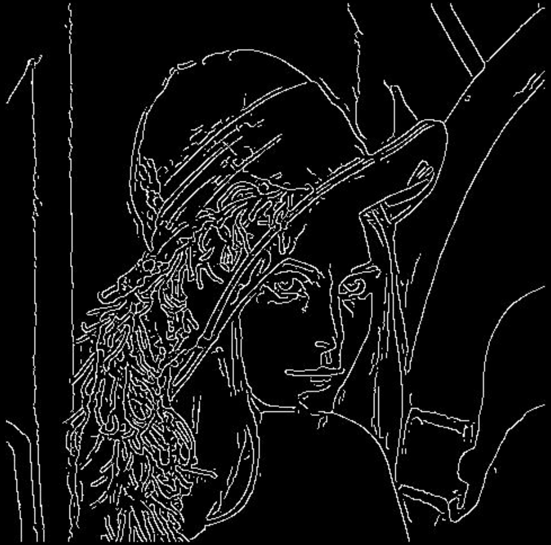
Original

Gray

Smoothing



Gradient Magnitude

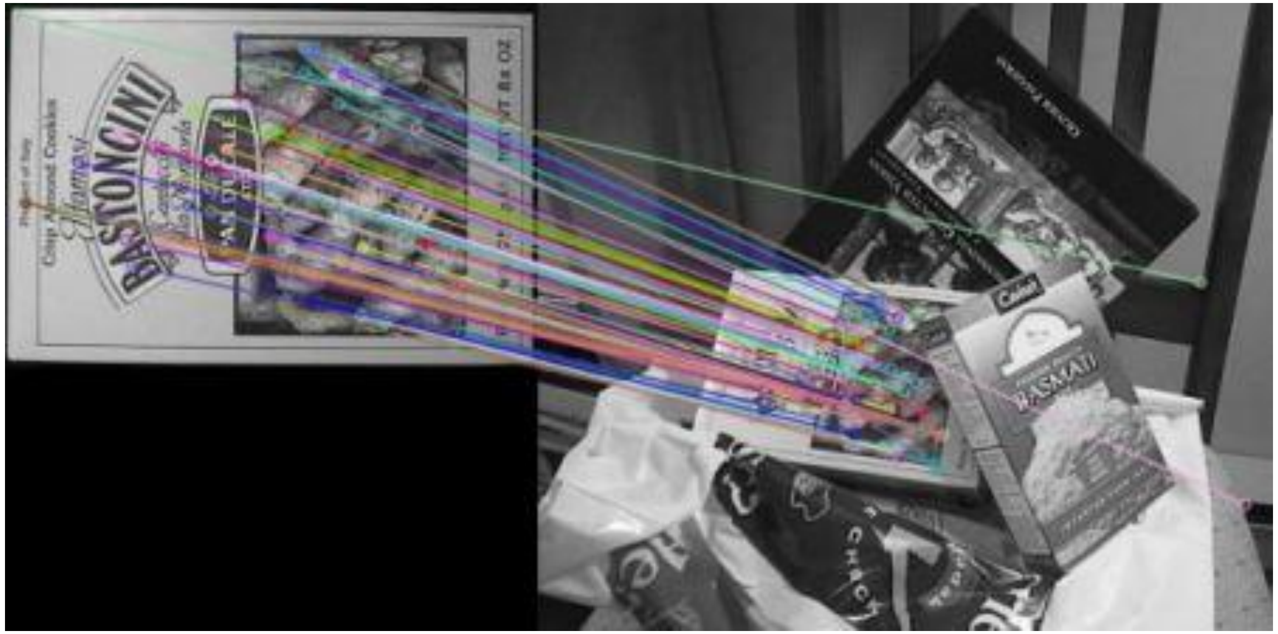


Canny Edge Output

Image Matching - Key Points

- Interest points are two dimensional structures in an image. They are specific locations in the images such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. These kinds of localized feature are often called key-point features or interest points or corners.
- Key points are normalized to find a local descriptor for it. The local descriptor is a vector of numbers that can be used to compare and match key points across different images.

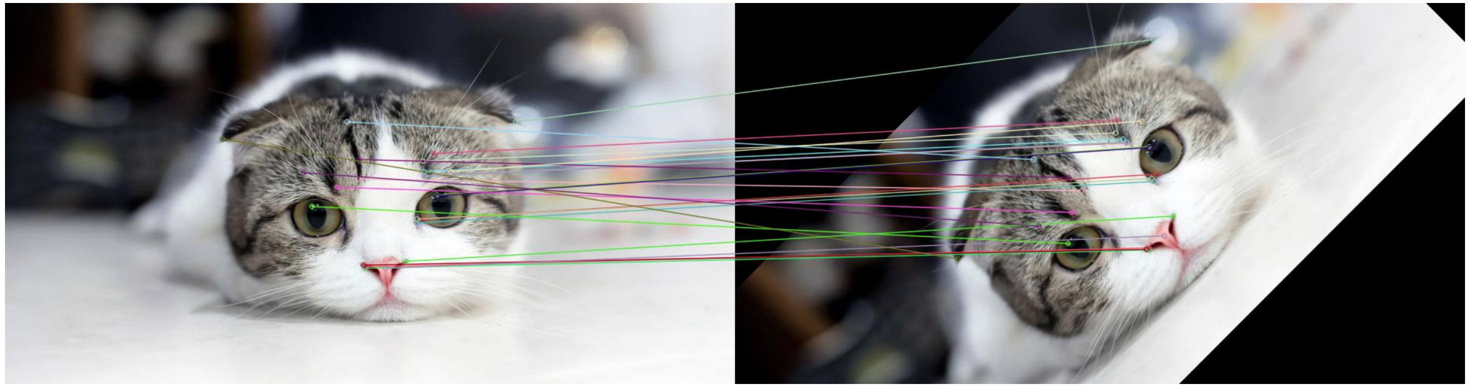
Reference: https://www.cs.ubc.ca/~lowe/keypoints/



OpenCV SIFT Docs

Image Matching - ORB

- ORB (Oriented FAST and Rotated BRIEF) is a replacement for SIFT and SURF in OpenCV.
- Uses a multiscale image pyramid, where a single image consists of sequences of images all of which are versions of the image at different resolutions. Each level in the pyramid contains the down sampled version of the previous image.
- Once orb has created a pyramid it uses the FAST algorithm to detect keypoints in the image independent of the scale



FAST: For a pixel x in an array fast compares the brightness of that pixel to 16 neighboring pixels in a circular region. They are then sorted into three classes: lighter than x, darker than x or similar to x. If more than 8 pixels are darker or brighter than x than it is selected as a keypoint.

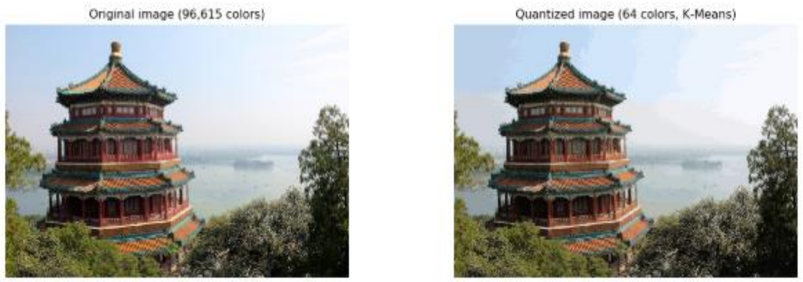
Reference: https://www.cs.ubc.ca/~lowe/keypoints/

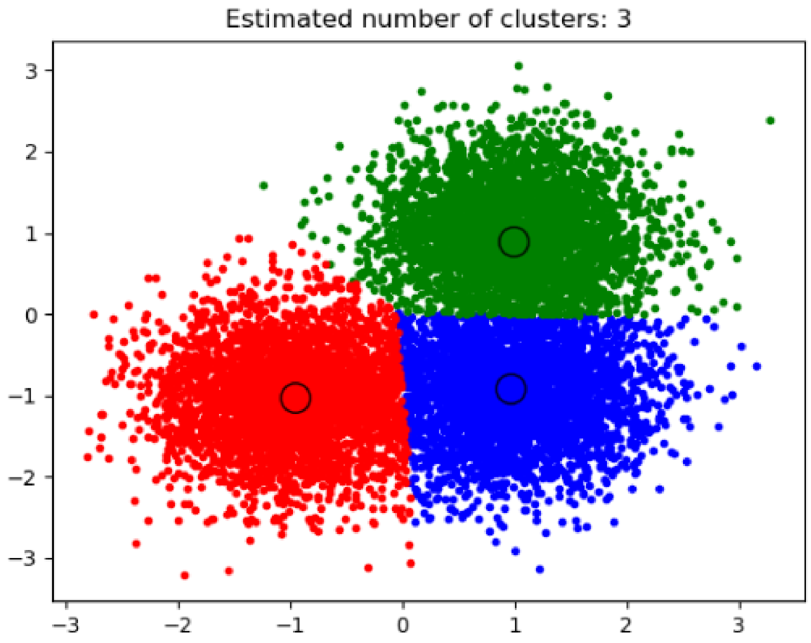
Image Matching - Blobs

- Blobs/ ROI points provide a complementary description of image structures in terms of regions, as opposed to corners that are more point-like. Blob detectors can detect areas in an image which are too smooth to be detected by a corner detector. Examples: DoG or DoH

Image Segmentation - Clustering

- Segmentation is to partition an image into separate regions to achieve a homogenous visual pattern.
- K-means clustering is an unsupervised machine learning technique used to separate data into distinct groups.
- Image compression to compress the images and reduce storage space.
- Pixel-Wise Vector Quantization (VQ) of a high resolution by reducing the number of colors required to show the image, while preserving the overall appearance quality.
- Large computation times and can be expensive


Original image (96,615 colors)
Quantized image (64 colors, K-Means)

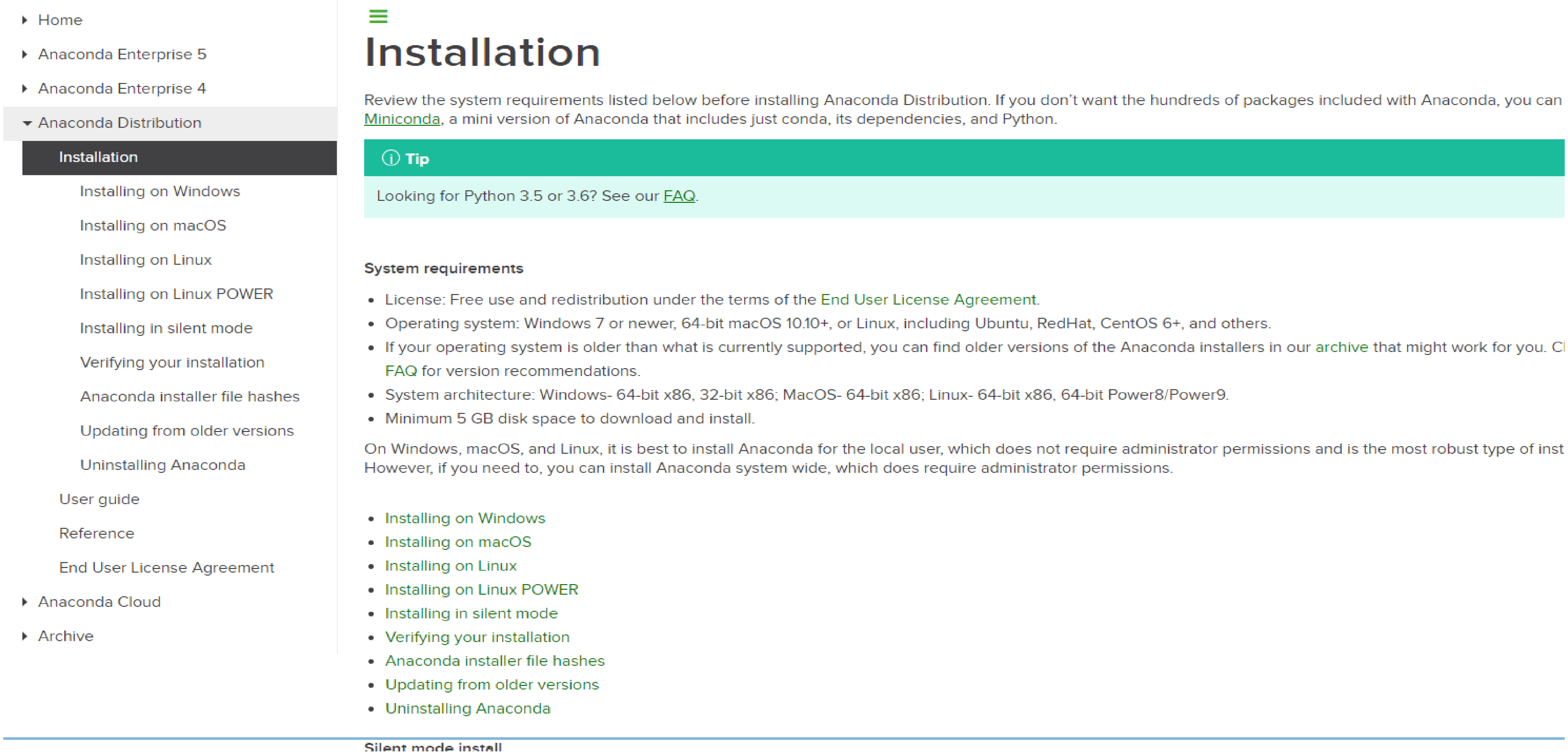


Reference: Scikit Learn

OpenCV: Summary

- Pre-processing: Techniques like noise reduction, contrast enhancement, sharpening, resampling etc.
- Feature Extraction: Object classification or detection
- Image Filters
- Segmentation

Installing Anaconda



- Home
- Anaconda Enterprise 5
- Anaconda Enterprise 4
- Anaconda Distribution
  - Installation
    - Installing on Windows
    - Installing on macOS
    - Installing on Linux
    - Installing on Linux POWER
    - Installing in silent mode
    - Verifying your installation
    - Anaconda installer file hashes
    - Updating from older versions
    - Uninstalling Anaconda
  - User guide
  - Reference
  - End User License Agreement
- Anaconda Cloud
- Archive

Installation

Review the system requirements listed below before installing Anaconda Distribution. If you don't want the hundreds of packages included with Anaconda, you can Miniconda, a mini version of Anaconda that includes just conda, its dependencies, and Python.

ⓘ **Tip**

Looking for Python 3.5 or 3.6? See our FAQ.

System requirements

- License: Free use and redistribution under the terms of the End User License Agreement.
- Operating system: Windows 7 or newer, 64-bit macOS 10.10+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.
- If your operating system is older than what is currently supported, you can find older versions of the Anaconda installers in our archive that might work for you. Cl FAQ for version recommendations.
- System architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit Power8/Power9.
- Minimum 5 GB disk space to download and install.

On Windows, macOS, and Linux, it is best to install Anaconda for the local user, which does not require administrator permissions and is the most robust type of inst However, if you need to, you can install Anaconda system wide, which does require administrator permissions.

- Installing on Windows
- Installing on macOS
- Installing on Linux
- Installing on Linux POWER
- Installing in silent mode
- Verifying your installation
- Anaconda installer file hashes
- Updating from older versions
- Uninstalling Anaconda

Silent mode install

https://docs.anaconda.com/anaconda/install/

Installing Python2 or 3 and creating environments

- Create a **Python2 environment** named py2, install Python 2.7: conda create --name py2 python=2.7
- Create a **Python3** environment named py3, install Python 3.6: conda create --name py3 python=3.6
- Now you have two environments to work with. You can install packages and run programs as desired in either one.

- **Activate** and use the **Python2** environment; WINDOWS: activate py2 ; LINUX, macOS: source activate py2
- Use your py2 environment to install packages and run programs as desired. When finished, **deactivate** the environment
- WINDOWS: deactivate py2; macOS, LINUX: source deactivate py2

- **Activate** and use the **Python3** environment; WINDOWS: activate py3; LINUX, macOS: source activate py3
- Use the py3 environment to install and run programs as desired. When finished, **deactivate** the environment
- WINDOWS:deactivate; macOS, LINUX:source deactivate

- Install **anaconda** as the next step (guide on moodle)

Installing Python Libraries Windows

- Pip: (py2) D:\>pip install --upgrade pip
- NLP: (py2) D:\> pip install nltk
- Numpy: (py2) D:\>pip install numpy -> express images as multi-dimensional arrays which can be manipulated
- Jupyter Notebook: (py2) D:\>pip install Jupyter
- Matplotlib: (py2) D:\>conda install matplotlib
- OpenCV: (py2) D:\>conda install -c conda-forge opencv
- Verifying the installation (no errors should pop-up)

```
(py2) D:\>python
Python 2.7.16 |Anaconda, Inc.| (default, Mar 14 2019, 15:42:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import nltk
>>> import cv2
```

Launching Jupyter Notebook (similar to colab):

- (py2) D:\HRI Coding>jupyter notebook
- Ctrl^c to quit running the server instance in command prompt

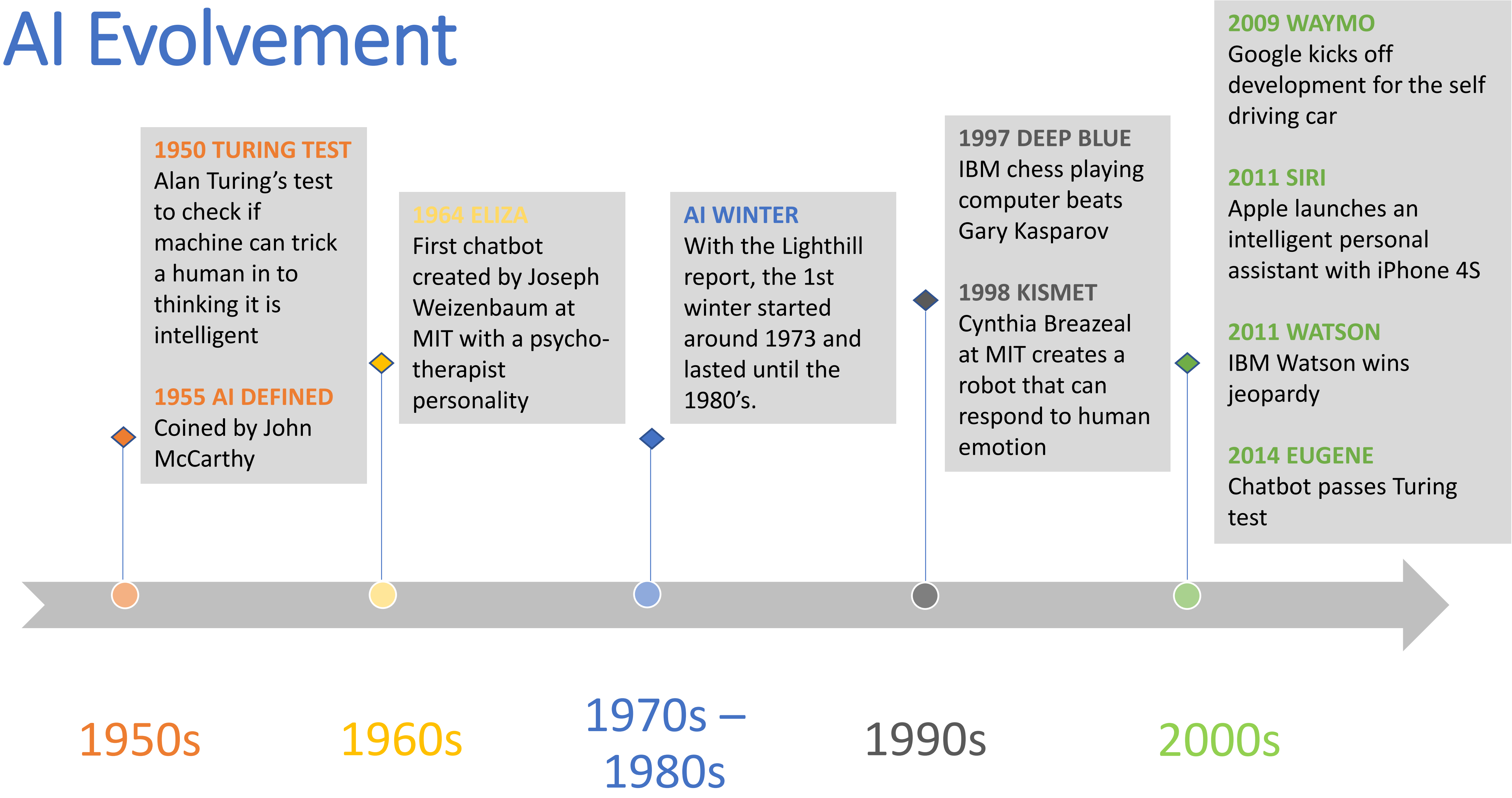
Artificial Intelligence (AI) Evolvement

- AI is the intelligence displayed by an artificial being/ machine.
- It works at an intersection of computer science, mathematics, mechatronics, psychology, linguistics, philosophy and other fields.
- Interestingly, as machines become more capable, they are not considered intelligent enough, a phenomenon known as the AI effect.
- Historically, researchers in AI would try to instill intelligence in systems by reasoning. For these systems to work in the real world a large number of rules had to be defined and applied.
- Overall goal for AI is to achieve general intelligence also referred to as Artificial General Intelligence (AGI)

AI Evolvement

- Other viewpoint is to use biology as a reference point and understand how human brain could be used as a model to mimic neural networks. This was particularly interesting as the goal was to identify how the networks could learn from experience to avoid specifying all the knowledge or rules by hand.
- Alan Turing, “Humans use available information, as well as reason, to solve problems and make decisions — then why can’t machines do the same thing?”
- Having billions of adaptive connections without any prior knowledge can be used now to translate French to English, although the rules of each language are different.

AI Evolvement



1950s

1950 TURING TEST
Alan Turing's test to check if machine can trick a human in to thinking it is intelligent

1955 AI DEFINED
Coined by John McCarthy

1960s

1964 ELIZA
First chatbot created by Joseph Weizenbaum at MIT with a psycho-therapist personality

1970s – 1980s

AI WINTER
With the Lighthill report, the 1st winter started around 1973 and lasted until the 1980's.

1990s

1997 DEEP BLUE
IBM chess playing computer beats Gary Kasparov

1998 KISMET
Cynthia Breazeal at MIT creates a robot that can respond to human emotion

2000s

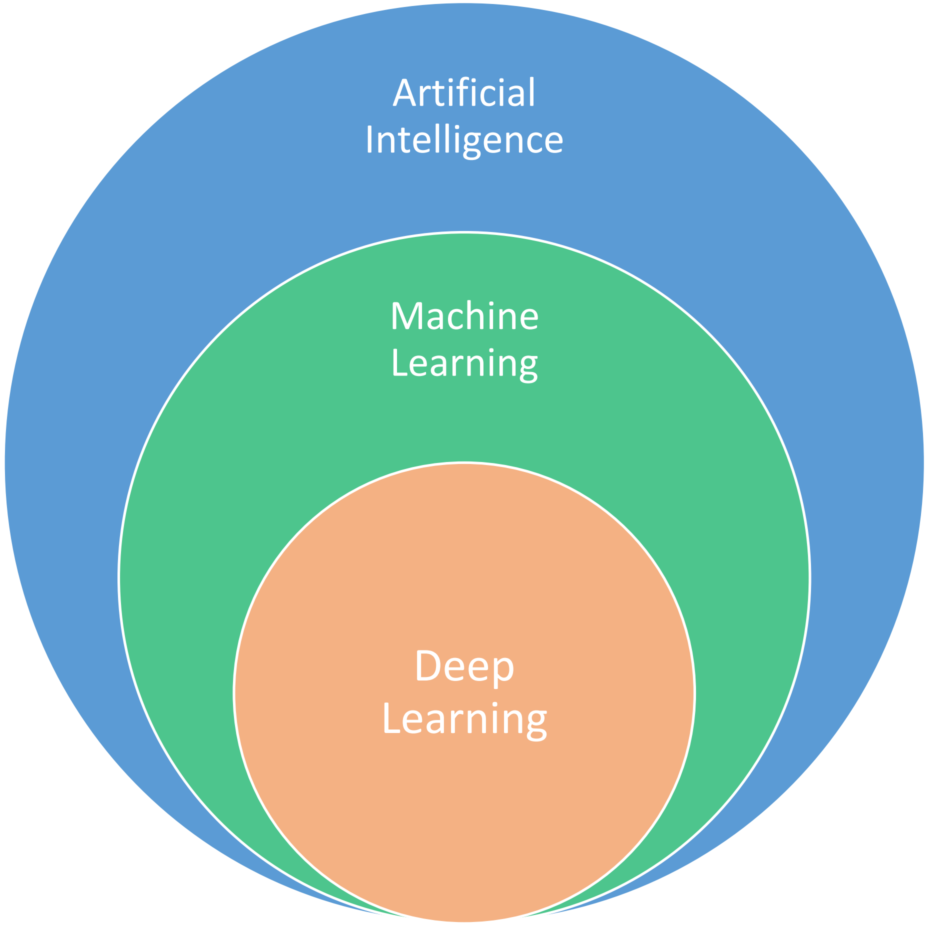
2009 WAYMO
Google kicks off development for the self driving car

2011 SIRI
Apple launches an intelligent personal assistant with iPhone 4S

2011 WATSON
IBM Watson wins jeopardy

2014 EUGENE
Chatbot passes Turing test

What is Deep Learning?



Deep Learning Evolvement

- 1960s: Shallow neural networks
- 1960-70s: Backpropagation emerges
- 1974-80: First AI Winter
- 1980s: Convolution emerges
- 1987-93: Second AI Winter
- 1990s: Unsupervised deep learning
- 1990s-2000s: Supervised deep learning
- 2006s-present: Modern deep learning

Reference: https://www.import.io/post/history-of-deep-learning/

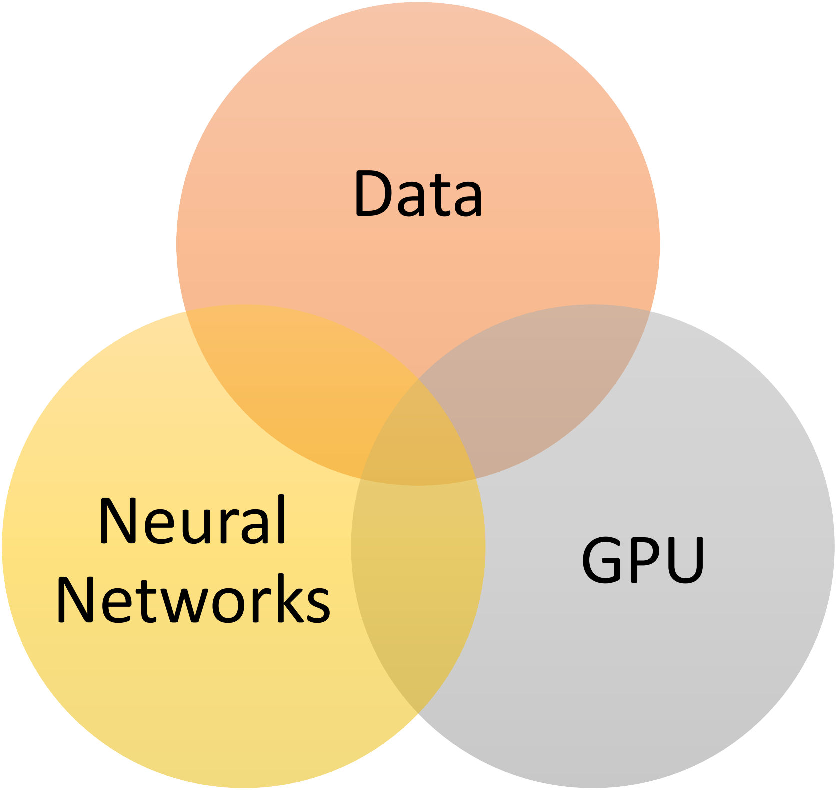
What is Deep Learning?

- Deep Learning is a subfield of machine learning and AI concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.
- It uses supervised learning when labeled data is used to train the neural network and unsupervised learning when unlabeled data is involved.

- Computer vision advancements have been mainly due to deep learning
- Alexnet won the 2012 Imagenet LVSRC competition by a tremendous margin using CNN architecture. Network achieved a top 5 error of 15.3% which was around 10% better than the runner up.

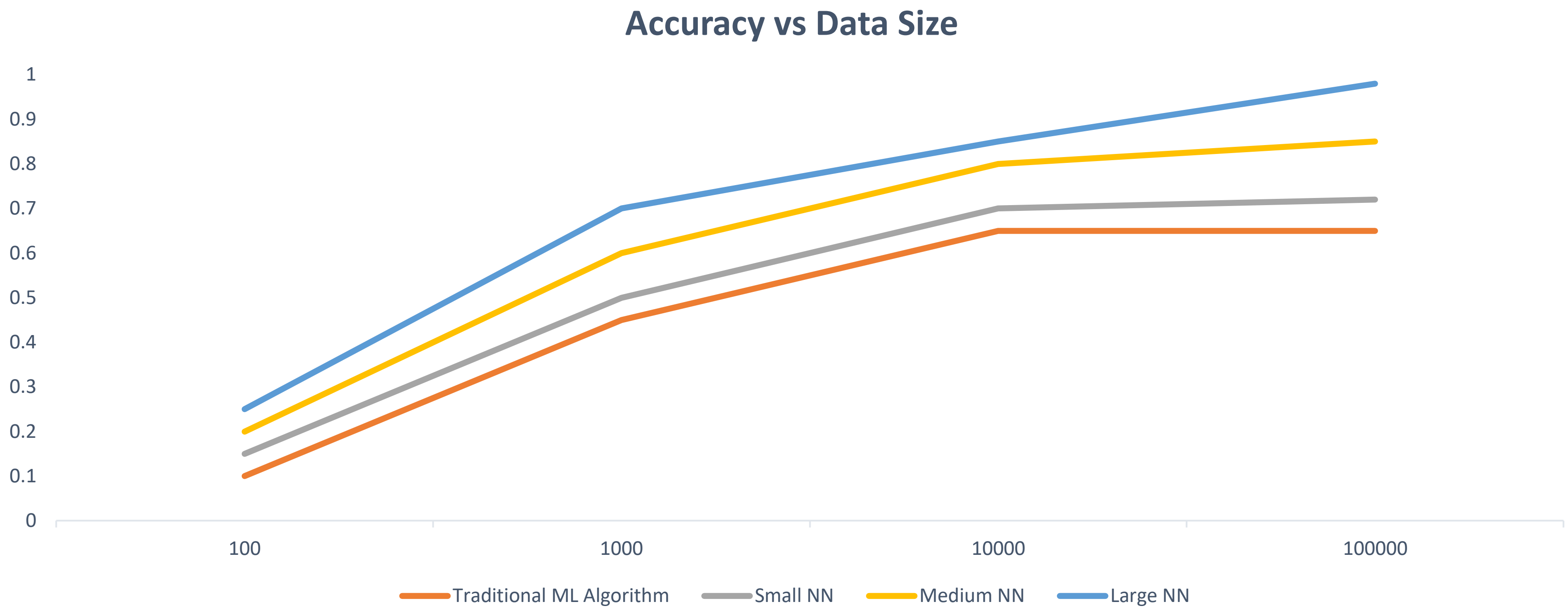
Reference: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Why DL now?



Reference: https://pdfs.semanticscholar.org/58d3/c288cdeeeba045514f32f8a5c5f75aa19fd7.pdf?_ga=2.97644252.2103249884.1561826614-1560707148.1560379449

Then why DL now? Higher Accuracy



- Better algorithms need larger data size as shown in the graph.
- Advent of GPU Performance

Then why DL now? ->ILSVRC or ImageNet

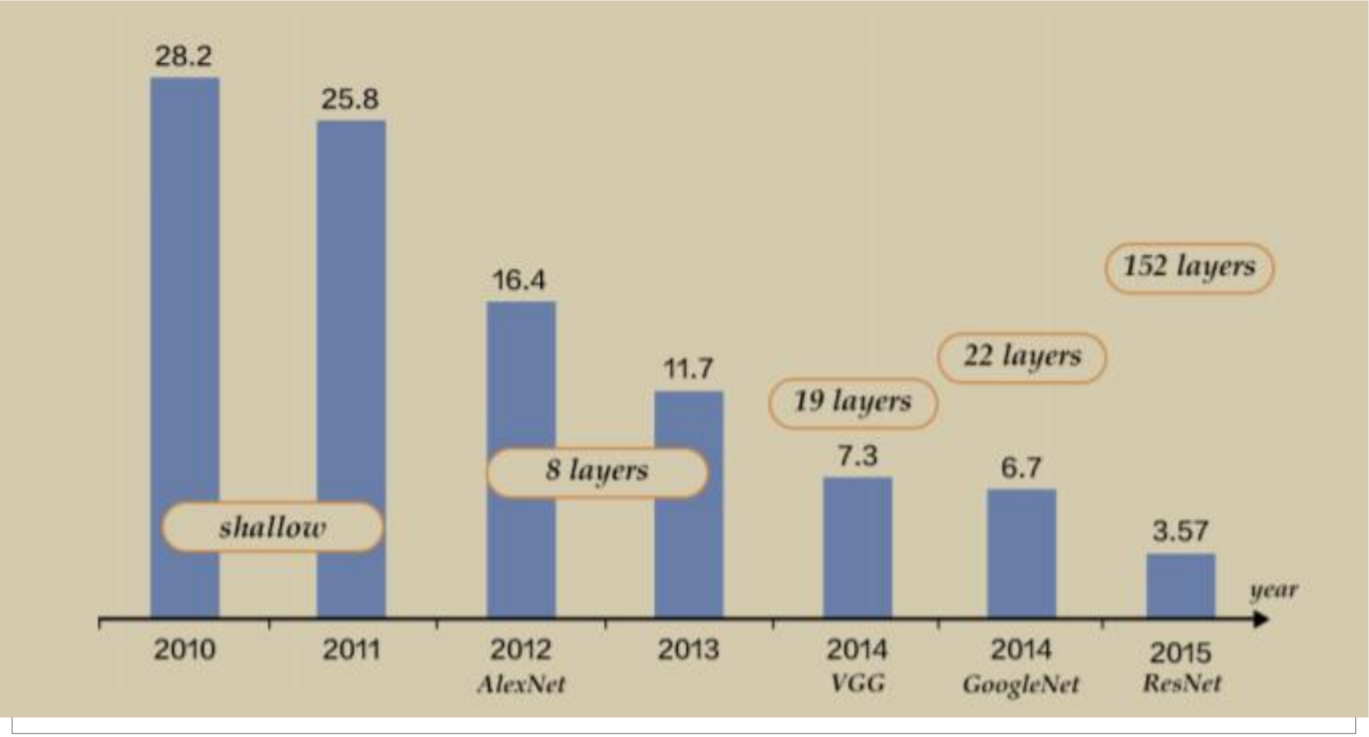
Background

- Created by Dr. Fei Fei Li in 2009
- > 14 million hand labelled images
- Labeled using Amazon's Mechanical Turk Crowd Sourcing tool
- Resolution is rescaled from the original image

Competition

- Classification: make 5 guesses about the image label; 1K categories; 1.2M training images and 100k test images
- Tasks: Image classification, object localization and detection.

Then why DL NOW? More layers



Reference: https://pdfs.semanticscholar.org/58d3/c288cdeeeba045514f32f8a5c5f75aa19fd7.pdf?_ga=2.97644252.2103249884.1561826614-1560707148.1560379449

But.. how deep is deep?

Until early 2000’s we could not easily train networks with more than 2 hidden layers.

There is no consensus amongst experts as to what makes a neural network “deep”; but we can say that:

- DL algorithms learn in a hierarchical fashion: stack multiple layers to learn increasingly more abstract concepts.
- Network should be > 2 layers to be considered “deep”
- Network with > 10 layers is considered very deep (architectures such as ResNet have been successfully trained with over 100 layers)

Reference: PyImage Book

Summary: Why DL didn’t work before?

Reasons DL did not take off during the 1990s or before:

- Labeled datasets were 1000 times smaller.
- Machines didn’t have enough computational power.
- Lack of deep neural network architectures
- Limited understanding of weight initialization and activation functions for non-linearity

What is a neural network?

- Linear Regression – Housing Prices
- Logistic Regression
- Gradient Descent
- Derivatives
- Activation Function
- Binary Classification
- Forward and Backward Propagation
- Parameters and Hyperparameters
- Supervised vs Unsupervised Learning
- RL
- Advice on using ML

What are artificial neural networks (ANN’s)?

- Neural networks or ANN’s are modeled based on the human brain for recognizing patterns in data as done by humans using sensory information. Before data can be analyzed by the computer it has to be labelled.

- Neural network are a group of neurons organized in layers. Each neuron is a mathematical operation that takes it’s input, multiplies it by it’s weights and then passes the sum through the activation function to the other neurons. Neural network learns how to classify an input through adjusting it’s weights based on previous examples.

What are artificial neural networks (ANN’s)?

- There are various artificial intelligence (AI) algorithms such as linear regression, logistic regression, k-nearest neighbors, support vector machines (SVM) and deep learning/ deep neural networks (DNN).

- DNN are most popularly used as they have spurred advances in text and speech recognition, computer vision and OCR, as well as empowering reinforced learning and robotic movement, along with other applications.

Applications of ANN’s

Supervised Learning: It depends upon labeled datasets; where humans annotate the dataset in order for a neural network to establish a relationship with the data.

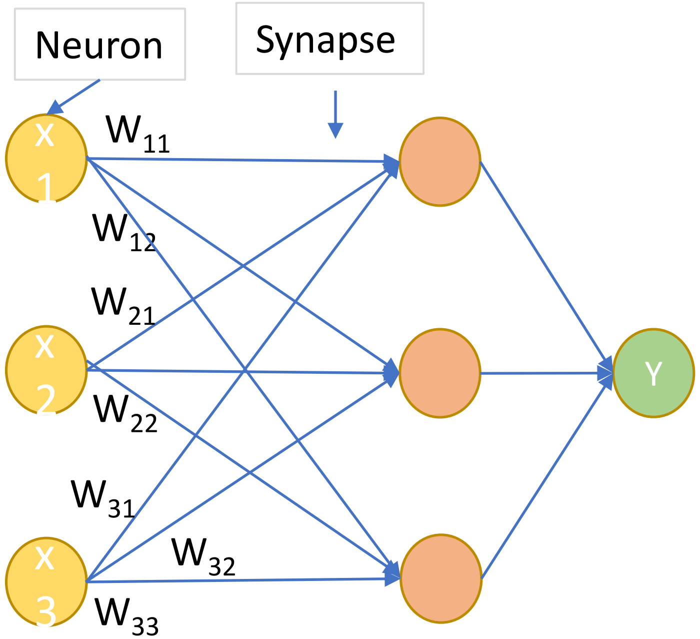
- Face recognition, emotion detection
- Object detection (cars, stop signs, pedestrians, lane lines etc.)
- Voice recognition, speech to text, identifying songs
- Classify email as spam or fraudulent
- Sentiment classification based on text

Applications of ANN’s

Unsupervised Learning: When labeled data is not present, which is the case for most of the data.

- Anomaly detection: Identifying unusual patterns to prevent fraud
- Clustering automatically by splitting the data into groups base on their similarities
- Data associations by identifying features/ item that reoccur

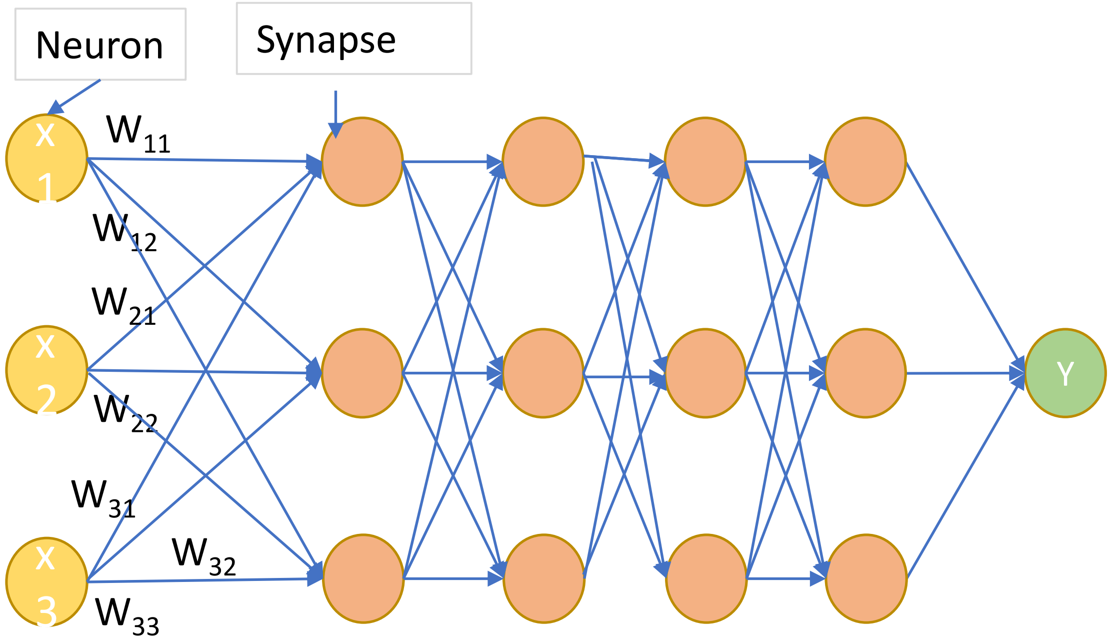
How does a neural network look like?



Input Layer
(Layer 0)

Hidden Layer
(Layer 1)

Output
Layer



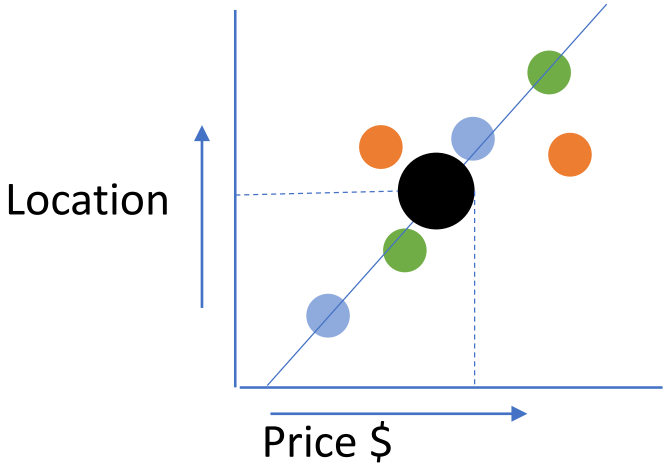
Input Layer
(Layer 0)

Multiple Hidden Layers
(Layers 1-4)

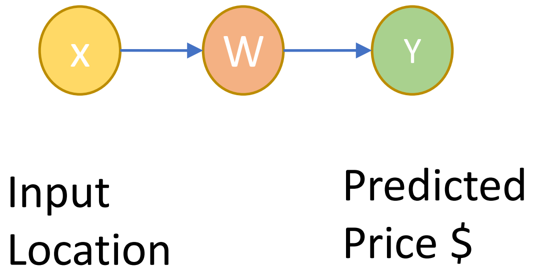
Output
Layer

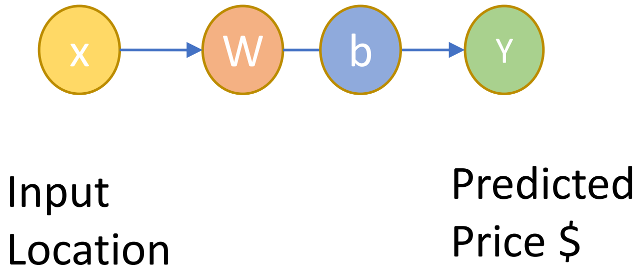
Neural network-> Single Feature

Housing Problem



$$Y = W * X + b$$

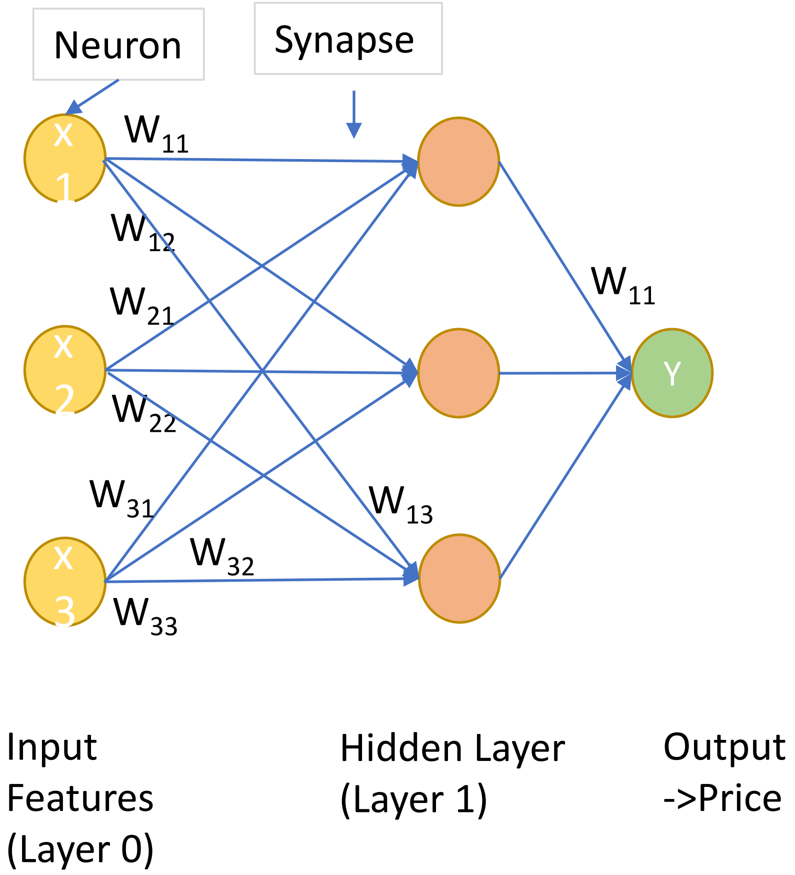




How does a neural network look like?

Features
Location
Area (Sq. Footage)
School District (Ranking)

$$\begin{pmatrix} a1 \\ a2 \\ a3 \end{pmatrix} = \sigma\left(\begin{pmatrix} .8 & .5 & .6 \\ .1 & .2 & .3 \\ .4 & .5 & .7 \end{pmatrix} \begin{pmatrix} .7 \\ .4 \\ .5 \end{pmatrix} + \begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix} \right)$$



Neuron

Synapse

Input Features (Layer 0)

Hidden Layer (Layer 1)

Output ->Price

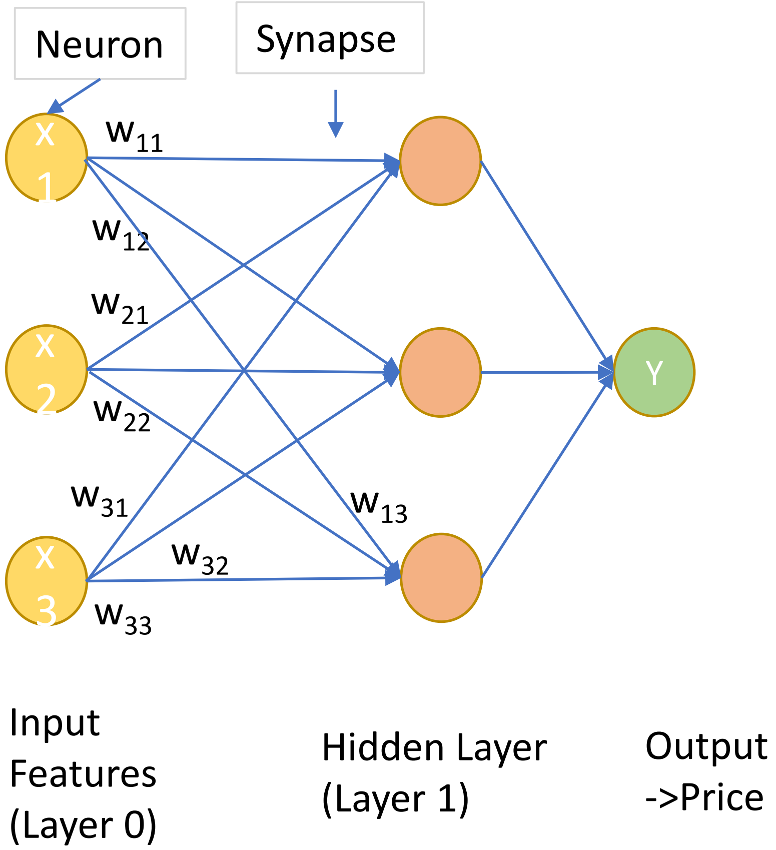
How does a neural network look like?

Features
Location
Area (Sq. Footage)
School District (Ranking)

$$\begin{pmatrix} a1 \\ a2 \\ a3 \end{pmatrix} = \sigma\left(\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix} \right)$$

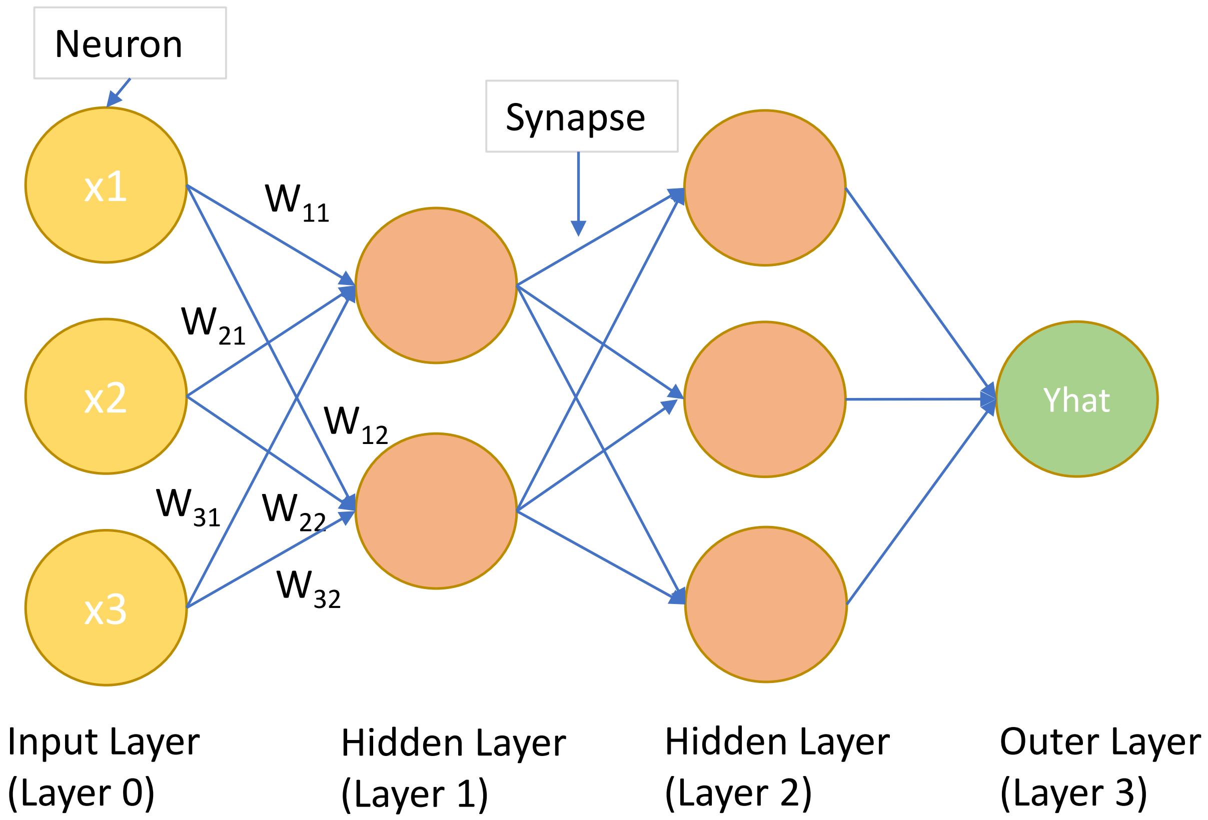
$$\left.\begin{aligned} &Z^{[1]}=W^{[1]}X+B^{[1]} \\ &A^{[1]} = \sigma(Z^{[1]}) \\ &A^{[1]} = \sigma(W^{[1]}X+B^{[1]}) \end{aligned}\right\} \quad \underline{\text{Layer1}}$$

$$\left.\begin{aligned} &Z^{[2]}=W^{[2]}A^{[1]}+B^{[2]} \\ &A^{[2]} = \sigma(Z^{[2]}) \end{aligned}\right\} \quad \underline{\text{Layer2}}$$



Note: Generally final activation used is a softmax to squeeze the output in to a probability distribution between 0 and 1.

Forward Propagation: General Equation



Layer1:

$Z^{[1]}=W^{[1]}X+B^{[1]}$

$A^{[1]}= \sigma(Z^{[1]})$

Layer2:

$Z^{[2]}=W^{[2]}A^{[1]}+B^{[2]}$

$A^{[2]}= \sigma(Z^{[2]})$

Output Layer:

$Z^{[3]}=W^{[3]}A^{[2]}+B^{[3]}$

$A^{[3]}= \sigma(Z^{[3]})$

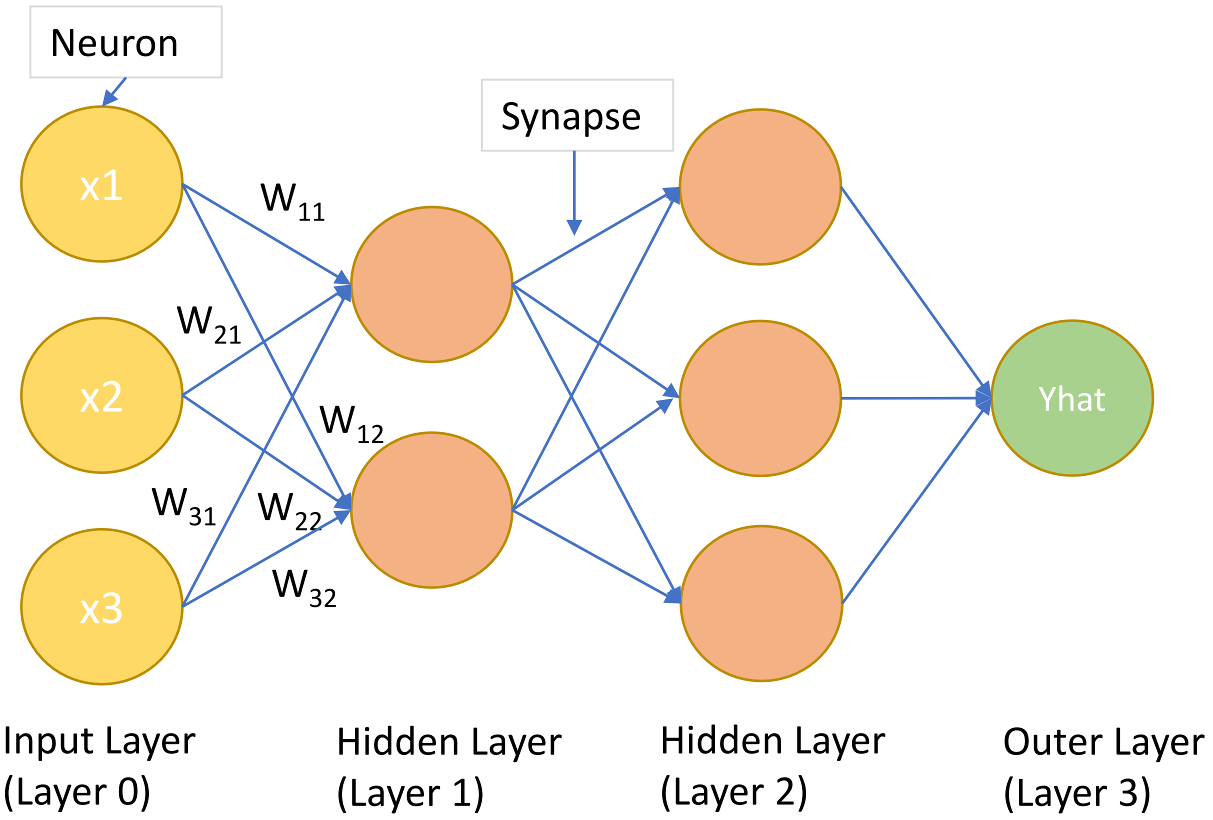
General Equation:

$Z^{[i]}=W^{[i]}A^{[i-1]}+B^{[i]}$

$A^{[i]}=F^{[i]}(Z^{[i]})$

$W^{[i]}=(n^{[i]},n^{[i-1]})$; $B^{[i]}=(n^{[i]},1)$

Forward Propagation



Layer1:

$Z^{[1]}=W^{[1]}X+B^{[1]}$; [2,1]=[2,3].[3,1]+[2,1]

$A^{[1]}= \sigma(Z^{[1]})$

Layer2:

$Z^{[2]}=W^{[2]}A^{[1]}+B^{[2]}$; [3,1]=[3,2].[2,1]+[3,1]

$A^{[2]}= \sigma(Z^{[2]})$

Output Layer:

$Z^{[3]}=W^{[3]}A^{[2]}+b^{[3]}$; [1,1]=[1,3].[3,1]+[1,1]

$A^{[3]}= \sigma(Z^{[3]})$

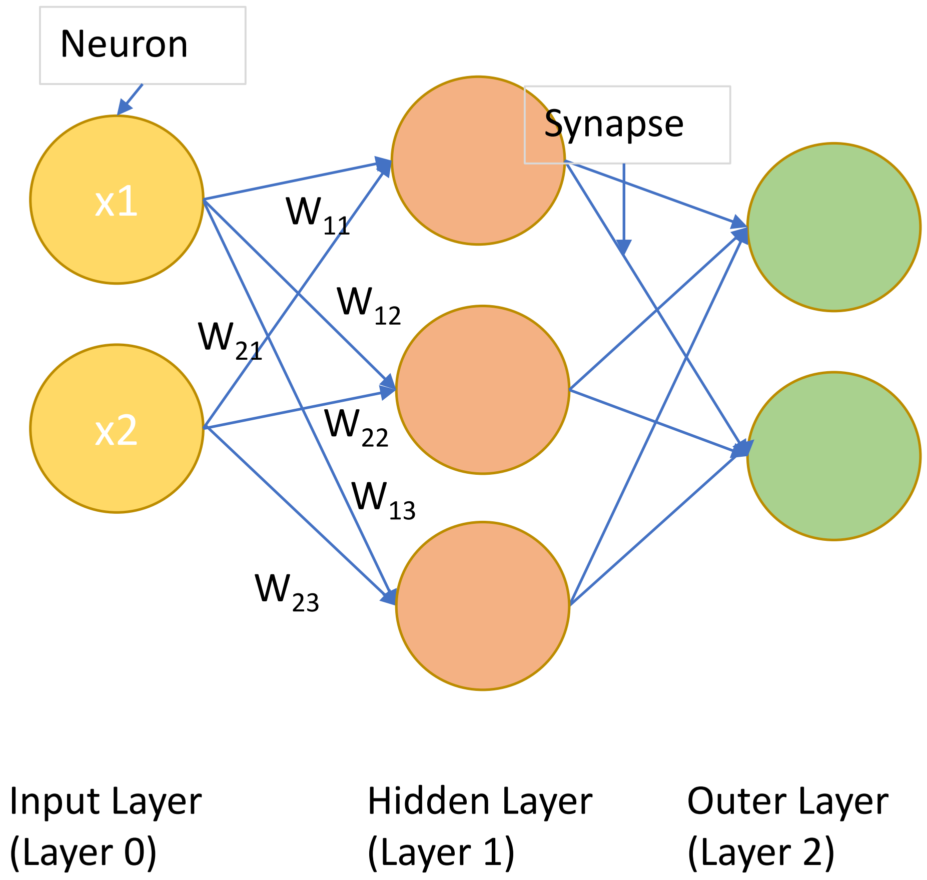
General Equation:

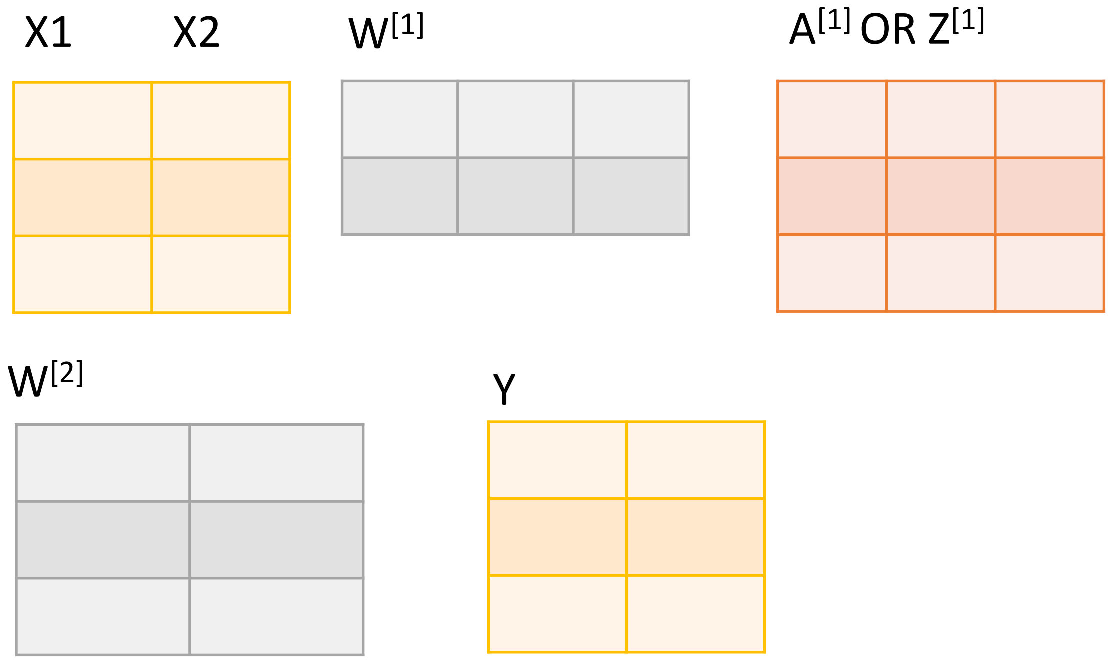
$Z^{[i]}=W^{[i]}A^{[i-1]}+B^{[i]}$;

$W^{[i]}=(n^{[i]},n^{[i-1]})$; $B^{[i]}=(n^{[i]},1)$

$A^{[i]}=F^{[i]}(Z^{[i]})$

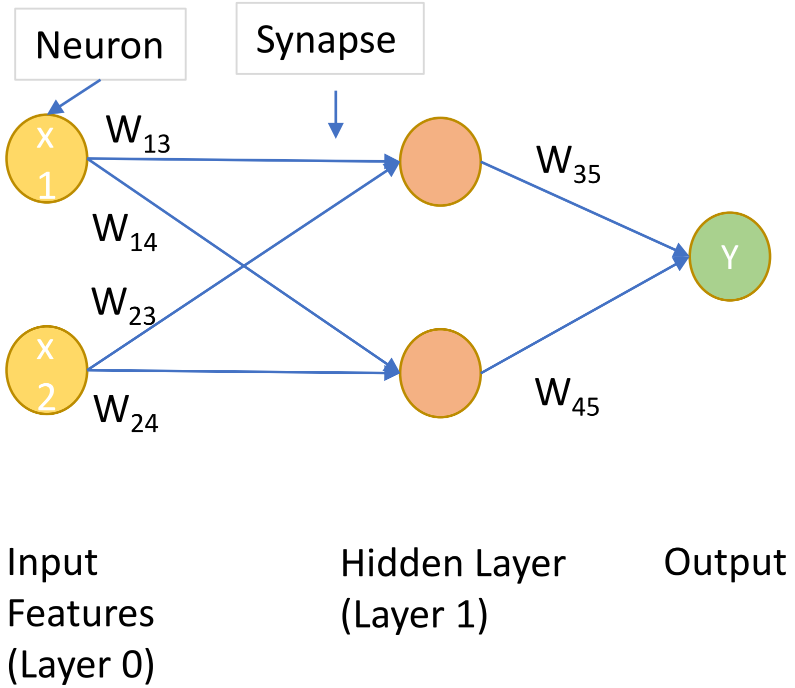
Forward Propagation – Matrix Dimensions





Input	[Features, Input Neurons]
Weights	[Neurons in previous layer, Neurons in present layer]
Bias	[1, Neurons in present layer]
Output Weights	[Hidden layer neurons, Neurons in output layer]

Practice Problem: Solve the neural network



Input, x1	**1**
Input, x2	0
w13	2
w14	1
w23	-3
w24	4
w35	2
w45	-1

$f(x) = 1$ if x>=0
else $f(x) = 0$

A1=f(w13.x1+w23x2)
=f(2*1+0*-3)=1

A2=f(w14.x1+w24x2)
=f(1+0)=1
Y=f(2-1)=1

Note: Real world problems are often more complicated as there are more unknowns (no weight information) than known values(only inputs and outputs are known).
This is where the term training comes in to play to identify the appropriate weights to achieve the desired output.

Discussion so far-> Neural network layers

Input Layer : Provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information(features) to the hidden layer.

Hidden Layer : Nodes of this abstraction layer are not exposed to the outer world. Hidden layer performs computation on the features entered through the input layer and transfer the result to the output layer.

Output Layer : This layer brings up the information learned by the network to the outer world.

Discussion so far-> Activation Function

They are used to determine the output of neural network like yes or no by squishing the output in a range of values. Range can lie in between 0 to 1 or -1 to 1 etc. depending upon the function selected.

Activation Functions can be basically divided into 2 types:

- Linear Activation Function
- Non-linear Activation Functions

So what are neurons? -> Linear Activation Function

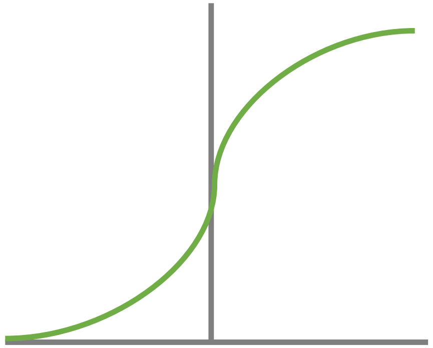
Linear Activation Function: Equation : Linear function has the equation similar to as of a straight line i.e. y = ax

No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear constant function of the input of first layer. Therefore, it is not used for training ANN’s or DNN’s.

Example: For linear regression of calculation of price of a house, linear activation is applied at output layer but non-linear activation function is used for the hidden layers.

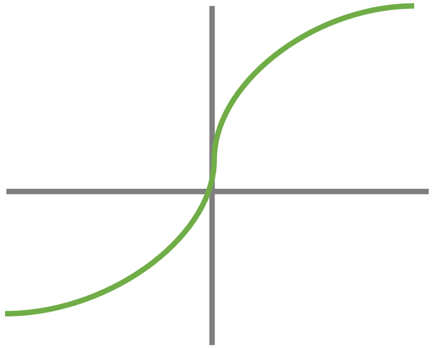
Non-Linear Activation Functions

Sigmoid



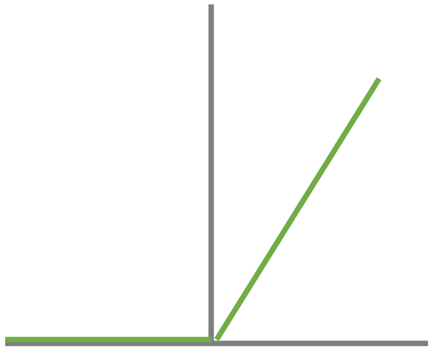
$$\mathrm{f(x)}= \frac{1}{1+e^{-(x)}}$$

Tanh



$$\mathrm{tanh(x)}= \frac{e^{(x)} - e^{-(x)}}{e^{(x)} + e^{-(x)}}$$

ReLU



$$relu(x) = \max(0, x)$$

- A neural network without an activation function is just a linear regression model.
- The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

Sigmoid-> Non-Linear Activation Function

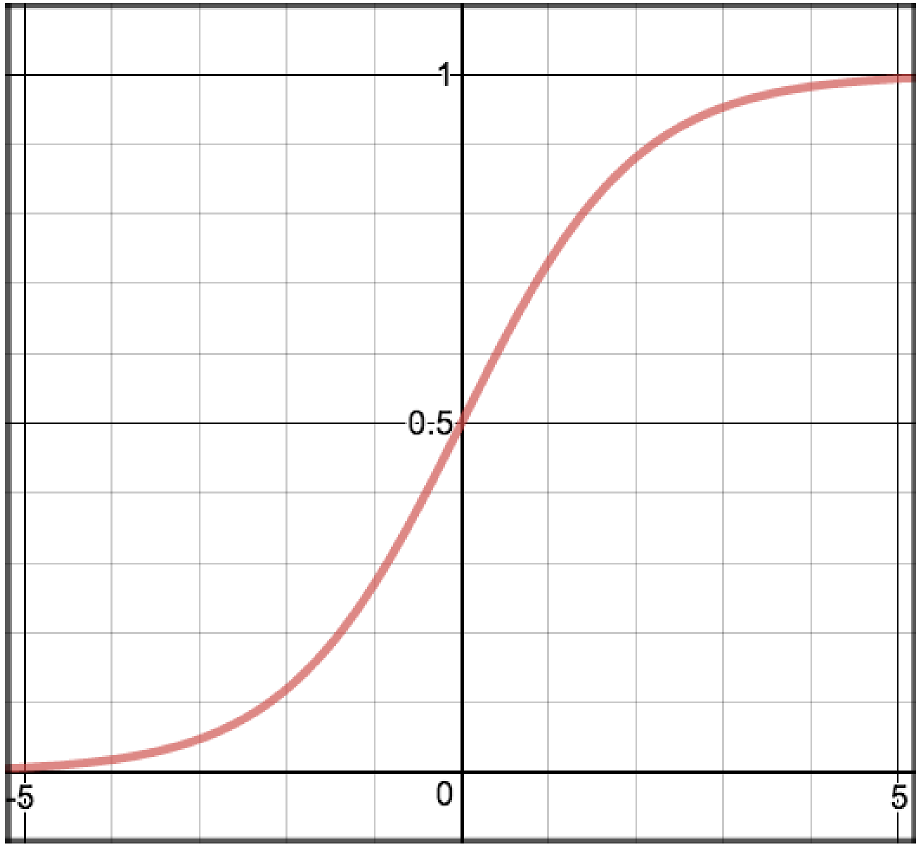
Small changes in x would also bring about large changes in the value of y.

Value Range : 0 to 1

Uses : Usually used in output layer of a binary classification, where result is either 0 or 1, result can be predicted easily to be 1 if value is greater than 0.5 and 0 otherwise.

Disadvantages: Extremely slow learning near the ends; y changes very slowly with changes in x.

It can lead to vanishing/ exploding gradient.

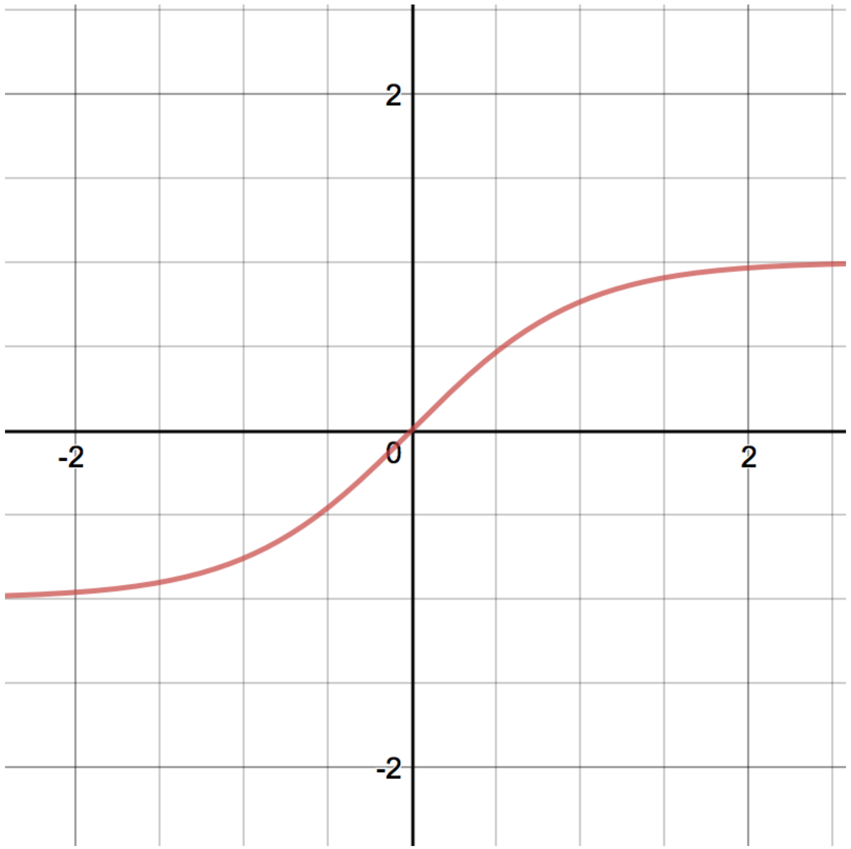


So what are neurons? -> Non-Linear Activation Function

Tanh function: Mathematically shifted version of the sigmoid function and can be derived from each other.

Value Range :- -1 to +1

Uses : Usually used in hidden layers of a neural network as it’s values lies between -1 to 1 hence the mean for the hidden layer comes out be close to 0 which helps in centering the data . This makes learning for the next layer much easier.

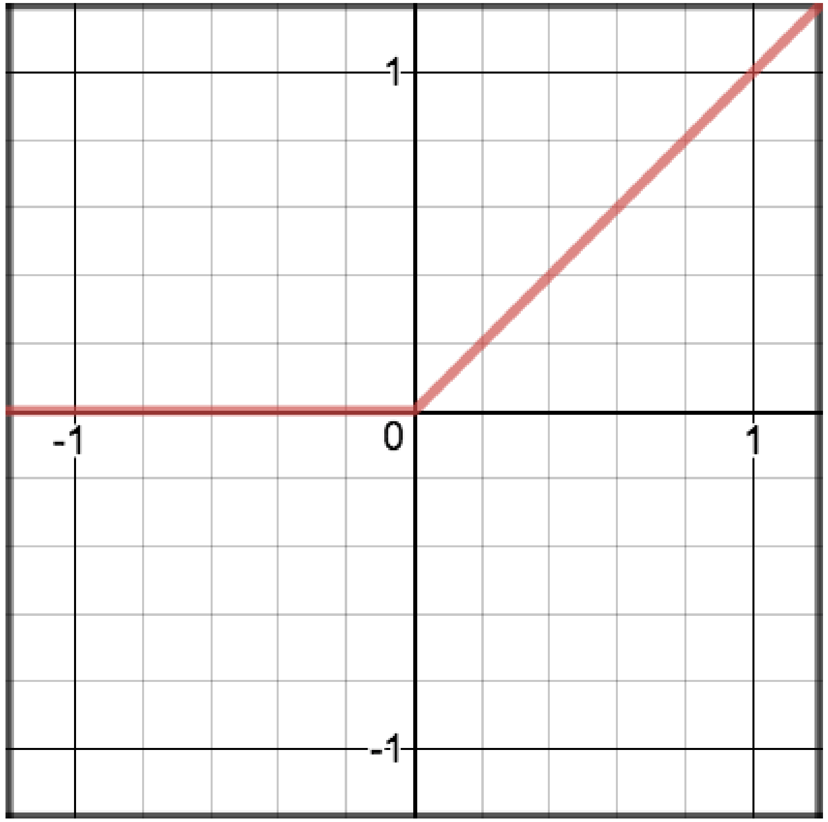


So what are neurons? -> Non-Linear Activation Function

RELU :- Rectified linear unit function is the most widely used activation function. Value Range :- [0, inf)

Uses :- ReLU is cheaper than tanh and sigmoid since it involves simpler mathematical operations as it activates only a few neurons.

Disadvantages: For activations in the region (x<0), gradient will be 0 for which the weights will not get adjusted during descent which leads to neurons not responding to variations in error/ input.

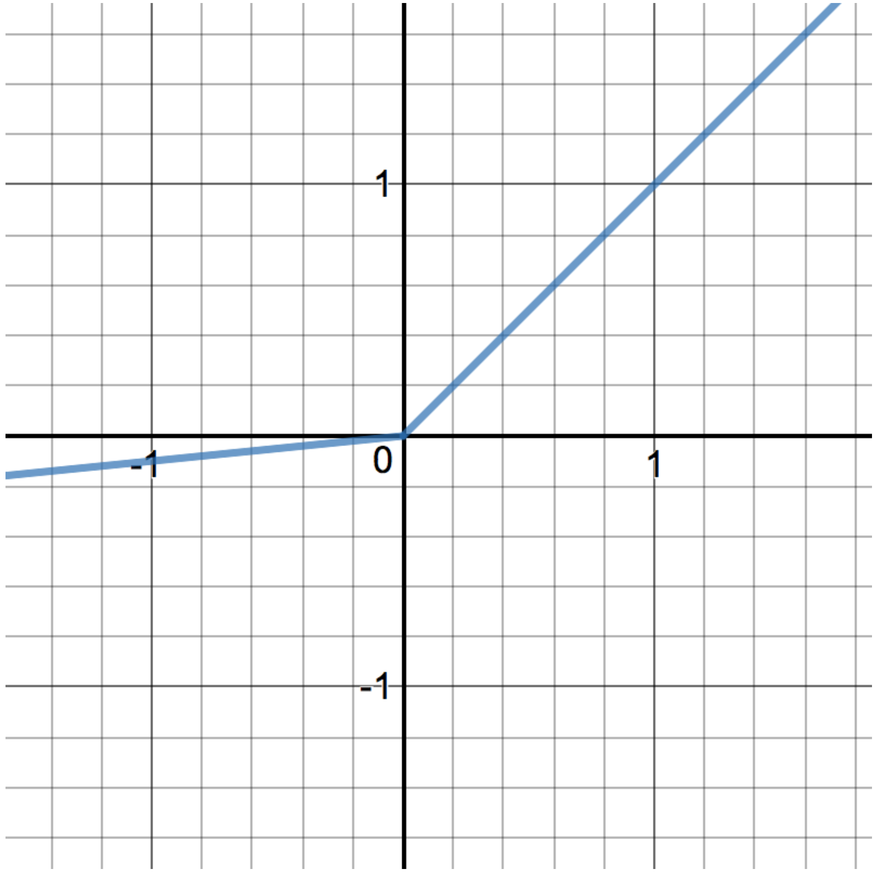


So what are neurons? -> Non-Linear Activation Function

Leaky RELU : Improved version of the ReLU function since in ReLU, the gradient is 0 for x<0, which made the neurons die for activations in that region. Leaky ReLU is defined as a function where value is a small linear component of x for x less than 0.

Value Range :- [az, inf), a=small constant

Uses :- Fixes the dying ReLU problem by having a small negative slope (of 0.01, or so).



Non-Linear Activation Functions->Softmax

- Softmax Function : It is also a type of sigmoid function but useful for classification problems.
- Uses : To handle multiple classes, the softmax function would squeeze the outputs for each class between 0 and 1 by dividing by the sum of the outputs.
- Ouput:- Ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.

Activation Function Selection

- Sigmoid functions and their combinations generally work better in the case of classifiers
- Sigmoids and tanh functions are sometimes avoided due to the vanishing gradient problem
- ReLU function is a general activation function and is used in most cases these days
- Softmax is used for classifying different classes by defining them in terms of probabilities.

So what are neurons? -> Activation Function Selection

- If we encounter a case of dead neurons in our networks the leaky ReLU function is the best choice
- Always keep in mind that ReLU function should only be used in the hidden layers
- As a rule of thumb, you can begin with using ReLU function and then move over to other activation functions in case ReLU doesn’t provide with optimum results

Vanishing/ Exploding Gradient Problem

- DNN’s when run through backpropagation a common problem with using sigmoid activation function is the vanishing gradient
- Training times scale up tremendously and accuracy decreases
- Cost is calculated by difference between the NN’s predicted output and the actual output
- Cost is minimized by adjusting weights and biases by training
- Training requires gradient which is the change in cost with respect to a change in weight or a bias
- Gradients are smaller in the initial layers where simple patterns can be learned y the NN as compared to a later layer
- During back propagation, each gradient is a derivative of previous gradients and hence they become flatter and flatter, and eventually they get vanished.

Summary/ Overview of Forward Propagation

- Takes inputs as a matrix (2D array of numbers)
- Dot product of the input by weights
- Applies an activation function
- Returns an output
- Error is calculated by taking the difference from the desired output from the data and the predicted output. This creates our gradient descent, which we can use to alter the weights
- The weights are then altered slightly according to the error.
- To train, this process is repeated 1,000+ times to achieve better accuracy