Session's Agenda:

1. DL Landscape
2. ML and DL
3. Why DL is Popular
4. Application
5. DL/ANN History
6. Perceptron | MLP
7. First use case DL

- Deep Learning is the heart of AI

- Deep Learning is subset of Machine Learning which in turn is the subset of Artificial Intelligence. Therefore, we can say that DL is the core of AI.

- Network of Artificial Neurons is called Neural Network

- Neuron is the building block of the neural network

MLP stands for Multi Layer Perceptron. Perceptron is the building block of MLP

Perceptron is the simplest neural network that is formed with the help of only 1 artificial neuron. Perceptron acts as an building block for the artificial neural network(ANN).

...LANDSCPAE OF DL...

1. DL is Subfield of AI and ML
2. inspired from human brain ⇒ Nervous System ⇒ Unit: Neuron → Network
3. Artfical Neurons ⇒ Neural Network

Network formed by the combination of multiple artificial neurons is called Neural Network

Human Neuron Image = Artfical Neuron Image

Artificial Neuron ⇒ Perceptron → MLP → Input
→ Hidden layer
→ O/P layer

Deep Neural layer

There are 3 parts/layers of the MLP or ANN: Input layer, Hidden layer(s) and Output layer.

Due to presence of Hidden Layer it acts as an Deep Neural layer

ML v/s DL

1. I/D ⇒ Model → Mathematical ⇒ Pattern
   Statical techniques

2. AN ⇒ Artificial Neuron ⇒ Neural Network

ANN, CNN, RNN, GAN and RL are the different parts of the Deep Learning where ANN is considered as the basic unit of DL that contributes in all the parts of DL (ANN, CNN, RNN, GAN and RL)

ANN (Artificial Neural Network)
Basic Unit of DL

CNN (Convolution Neural Network)
Convolution + ANN

GAN
ANN

RL
ANN
Q-learning
Deep Q NN

Q-learning and Deep QNN are some famous algorithms used in RL (Reinforcement Learning)

RNN (Recurrent Neural Network)
ANN + Recurrency + BTT
LSTM
GRU

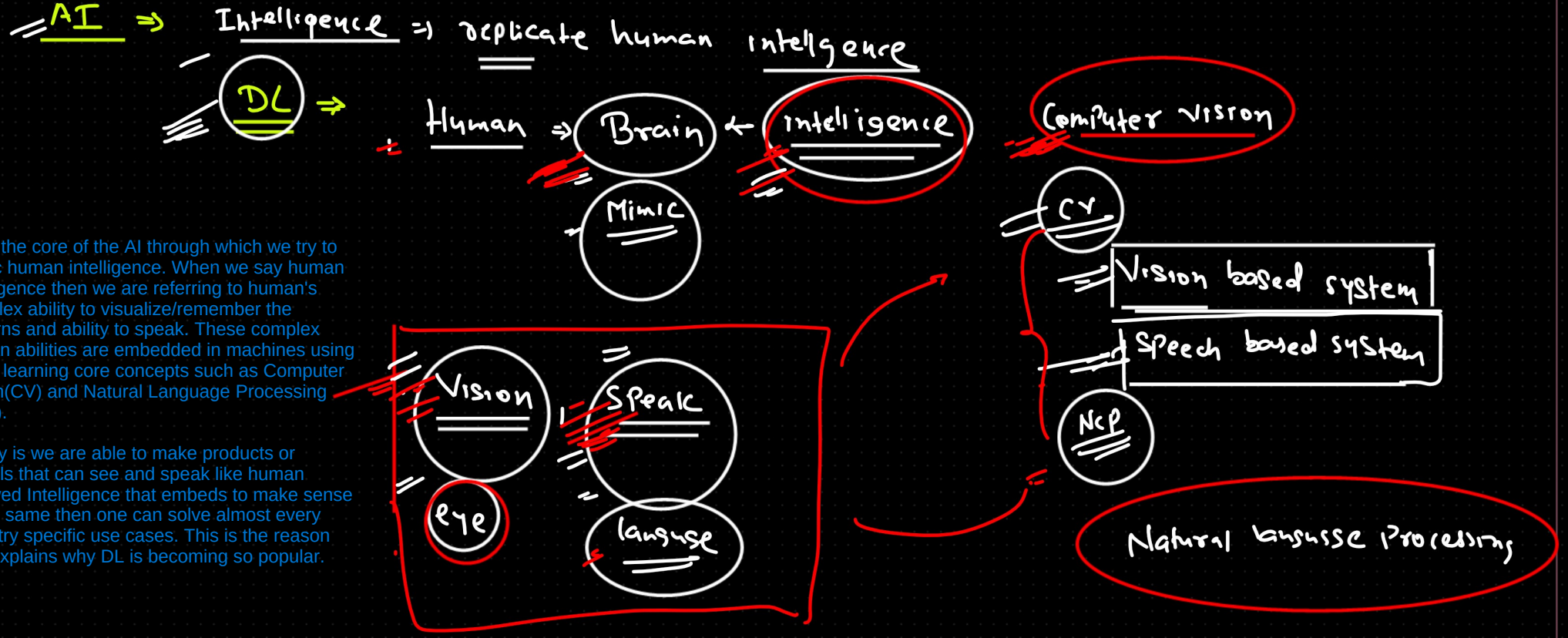
LSTM and GRU are some of the good implementation of RNN

BTT stands for Back Propagation through time

Why this DL is too much famous

Applicable in each and every Domain

Vast amt of Problem , Wide Domain of Problem



DL is the core of the AI through which we try to mimic human intelligence. When we say human intelligence then we are referring to human's complex ability to visualize/remember the patterns and ability to speak. These complex human abilities are embedded in machines using Deep learning core concepts such as Computer Vision(CV) and Natural Language Processing (NLP).

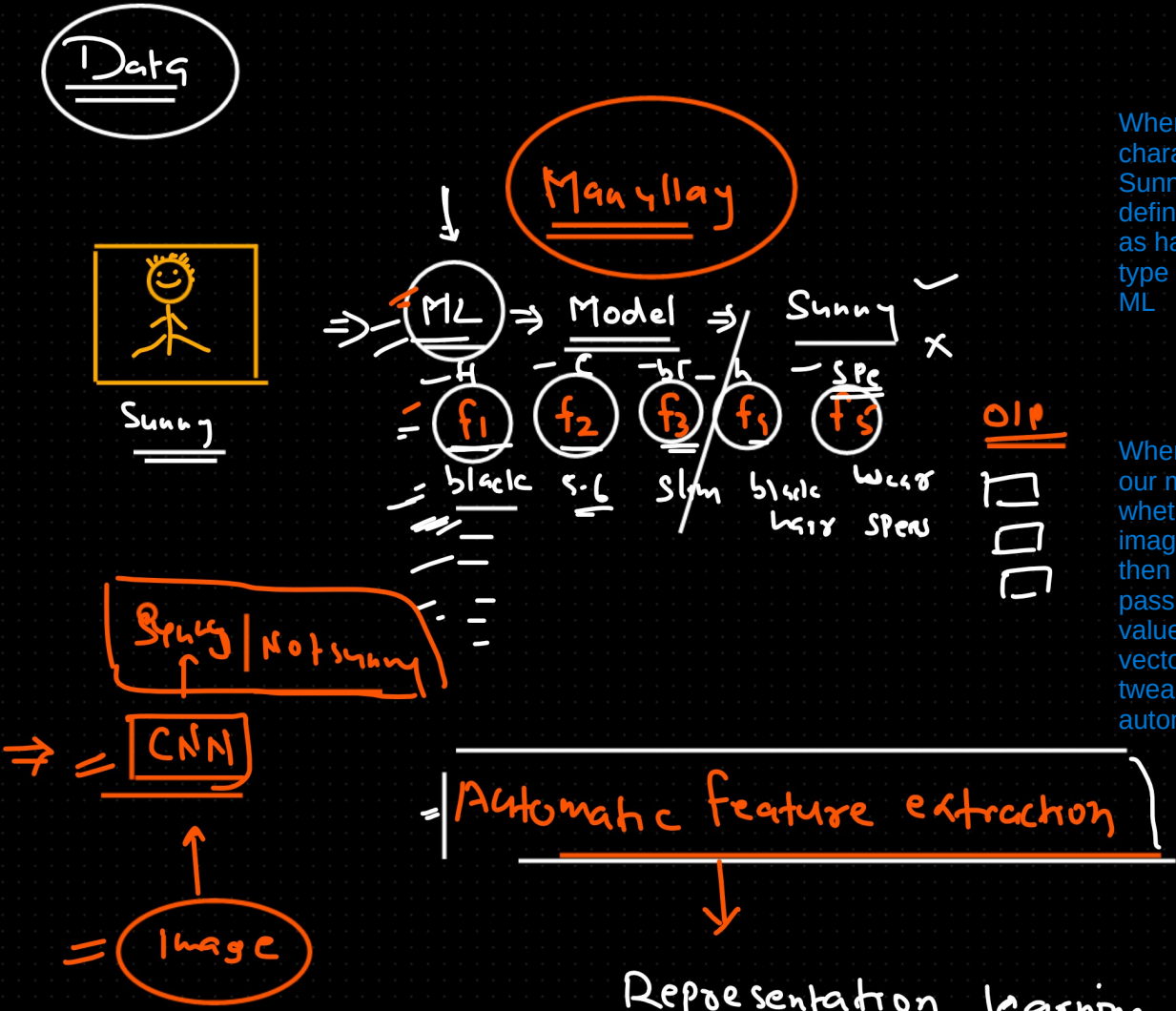
Ideally is we are able to make products or models that can see and speak like human followed Intelligence that embeds to make sense of the same then one can solve almost every industry specific use cases. This is the reason that explains why DL is becoming so popular.

When to use ML or DL :

In ML we need to pass the features manually whereas in DL we don't need to do this. In DL there is automatic feature extraction process through which model itself creates optimal set of features which is also called Representation learning

ML , DL

Feature extraction



When we pass the characteristics of Sunny manually by defining feature such as hair, height, body type etc then we use ML

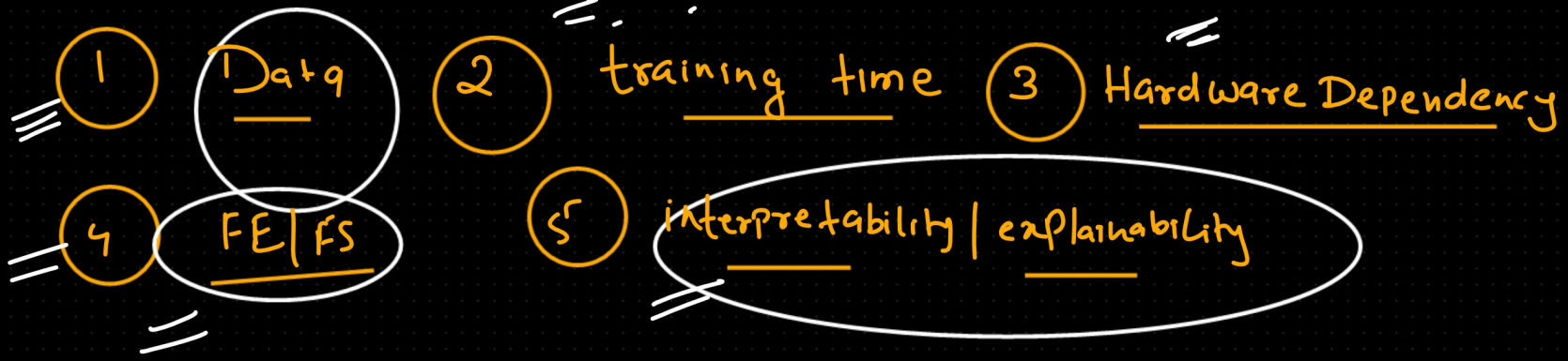
Whereas, if we train our model to identify whether person in image is sunny or not then we are basically passing Pixel RGB values in the for or vector that DL algo is tweaking in an automated fashion

Automatic feature extraction

↓

Representation learning

Now we will understand that when to use ML or DL based on the following parameters:



1. Data:

DL ⇒ huge amr of Data

ML ⇒ Not required (low amr of Data)

Problem

Resume ⇒ Select / Not Select ⇒ ? = ML = Manually

f1 f2 f3 .... f4 ⇒ Y

DL ⇒ CNN ⇒ Image classification / Object Detection / Segmentation / OCR

Automate ⇒ CNN is the basic unit or building block of CV

Computer vision ⇒ CV ⇒ CNN → OD, OS, OCR, OT

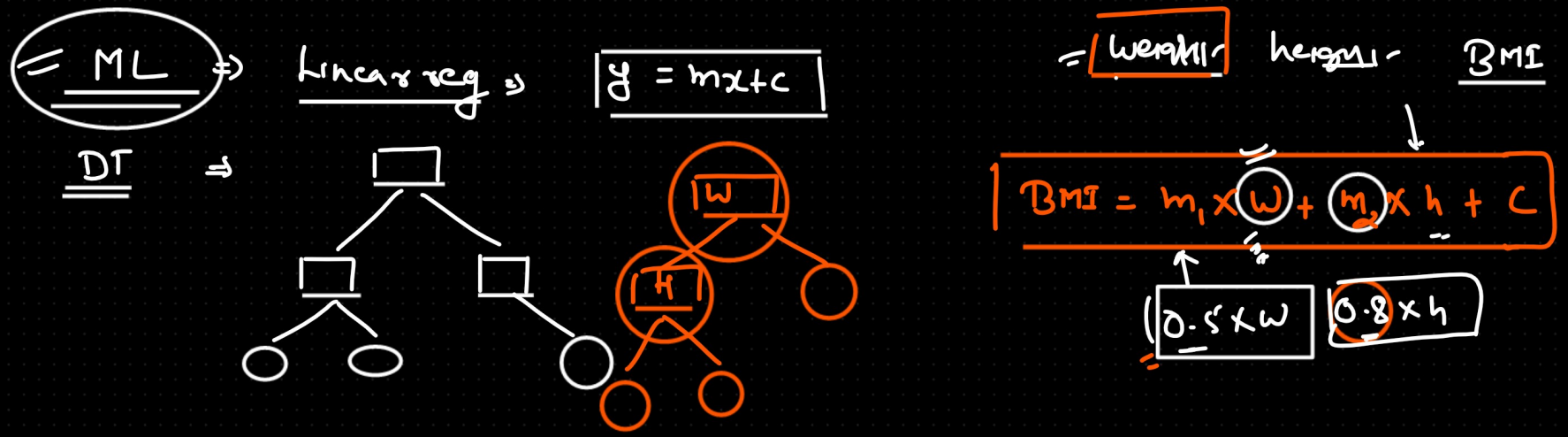
NLP ⇒ RNN ⇒ !!! ⇒ Text-extraction

2. Training Time: Since ML models are trained over less data when compared with DL they require then less time. Whereas, DL model can take a lot of time(days, weeks ,months, year). For Eg; ChatGPT is the DL based model that took several months to get trained

3. Hardware Dependency: ML models don't have any specific hardware prerequisite and they can simply run over CPU. Whereas. DL models on the other hand have special hardware dependency such as GPU(graphical processing unit), edge devices and TPU.

4. Feature Selection or Feature Extraction: This is manual in case of ML whereas in case of DL this is done in an automated fashion that is called Transfer Learning.

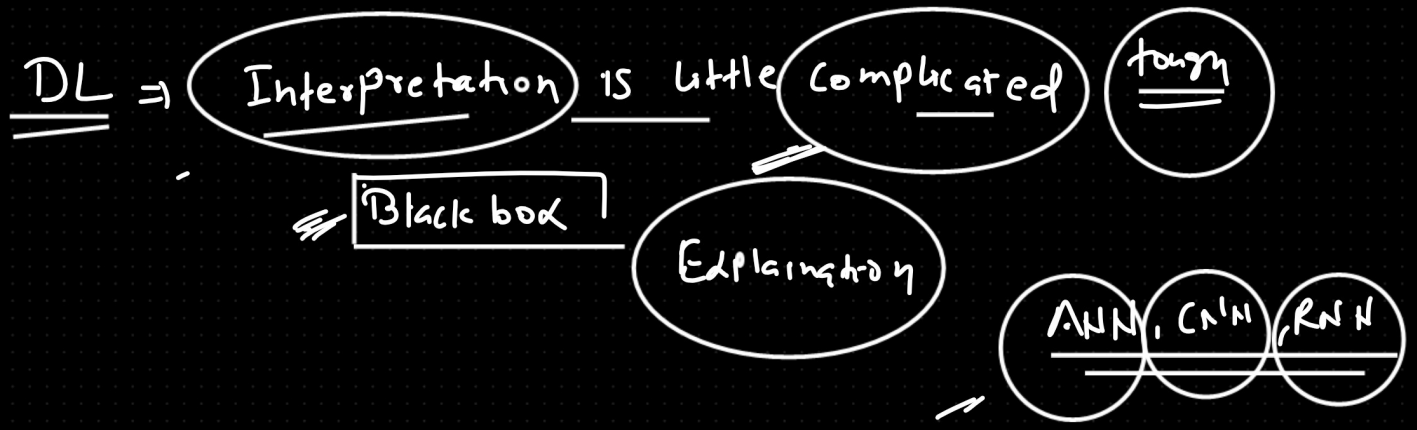
5.
Interpretability/
Explain-ability:



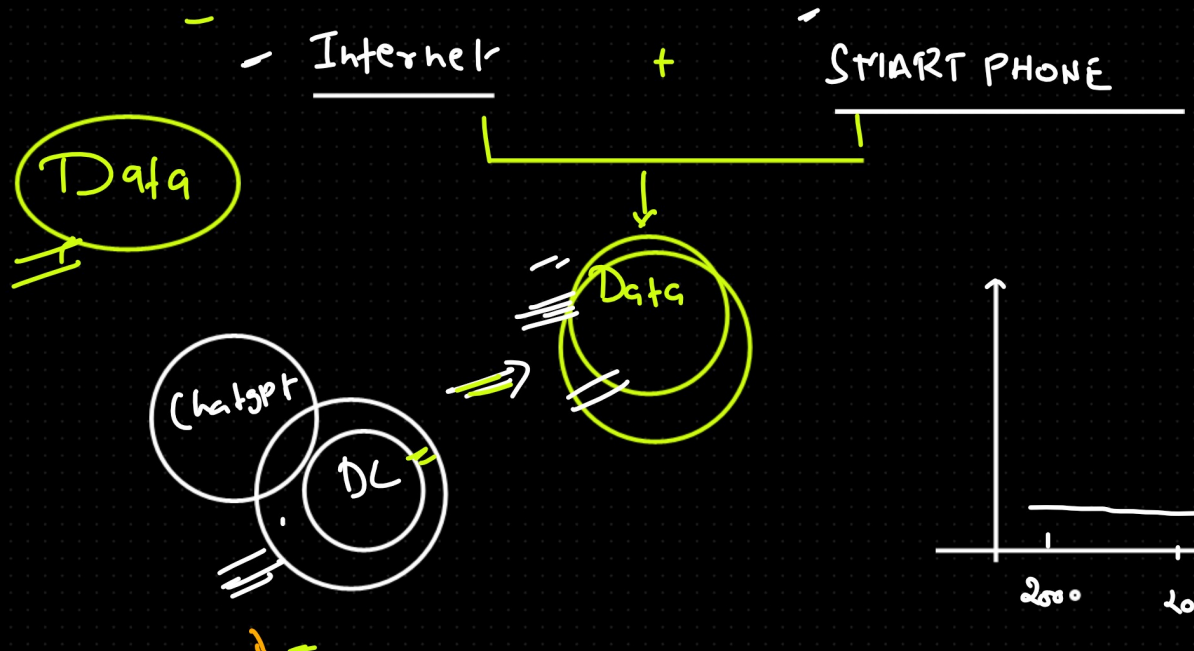
In Machine Learning models say for example we are provided with a linear regression based hypothesis where we can easily tell the importance of the features included by comparing the respective weights(Learning Parameter). The feature that is having high importance will be assigned with bigger weight and hence if we talk in terms of Decision tree then that particular feature will be treated as the root node.
To, conclude we can say that ML models are easy to Interpret and Explain.

Whereas, DL model on the other are little complicated and a bit difficult to Interpret and Explain due to presence of complex network of hidden layers. This is the reason why DL models are also often called as a BLACK BOX.

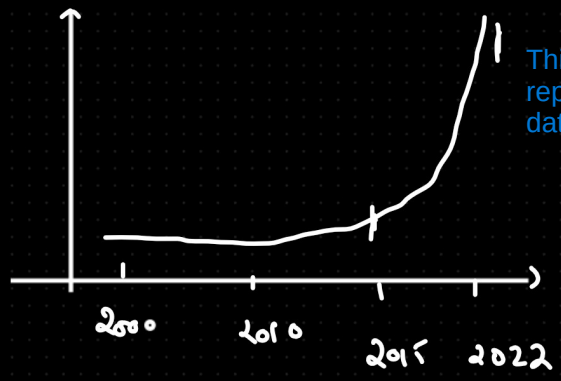
Interpretation | Explain



Understanding the reasons that actually gave the breakthrough in the field of data resulting in the generation of huge amount of data thus expanding the scope of AI, ML and DL:



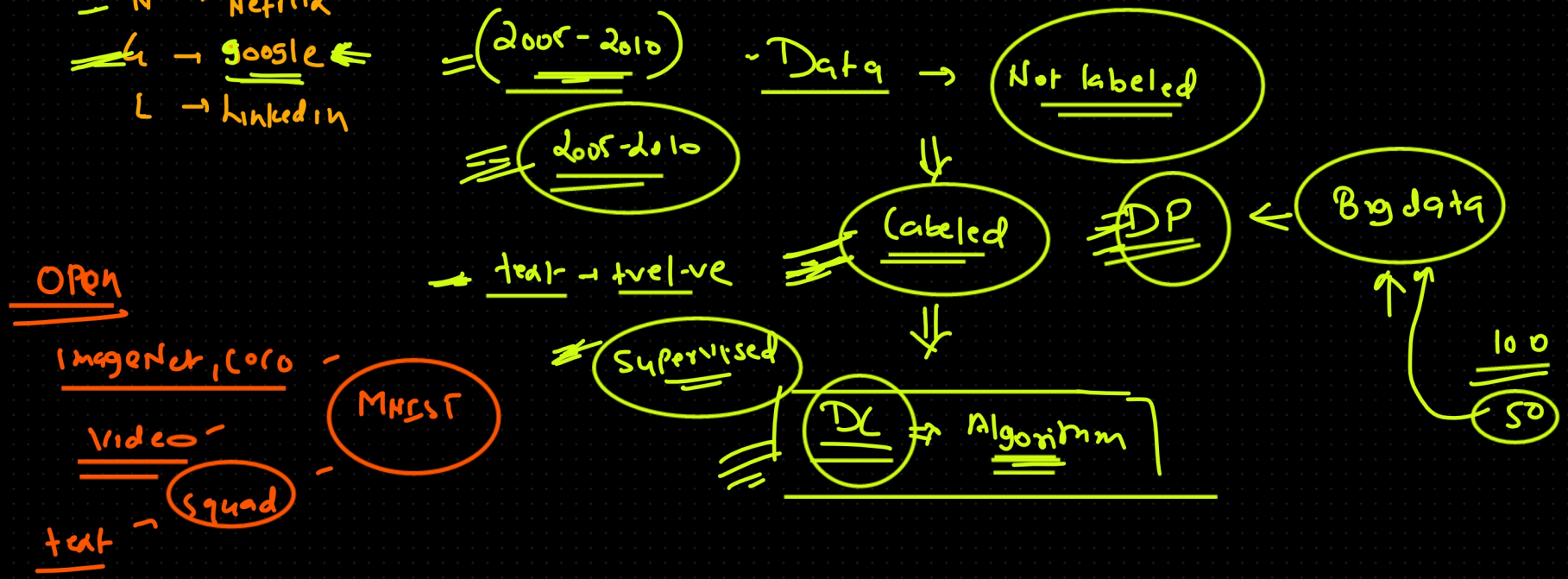
Internet and Smartphones are the ones that has given a breakthrough in the field of data thus generating huge volume of data

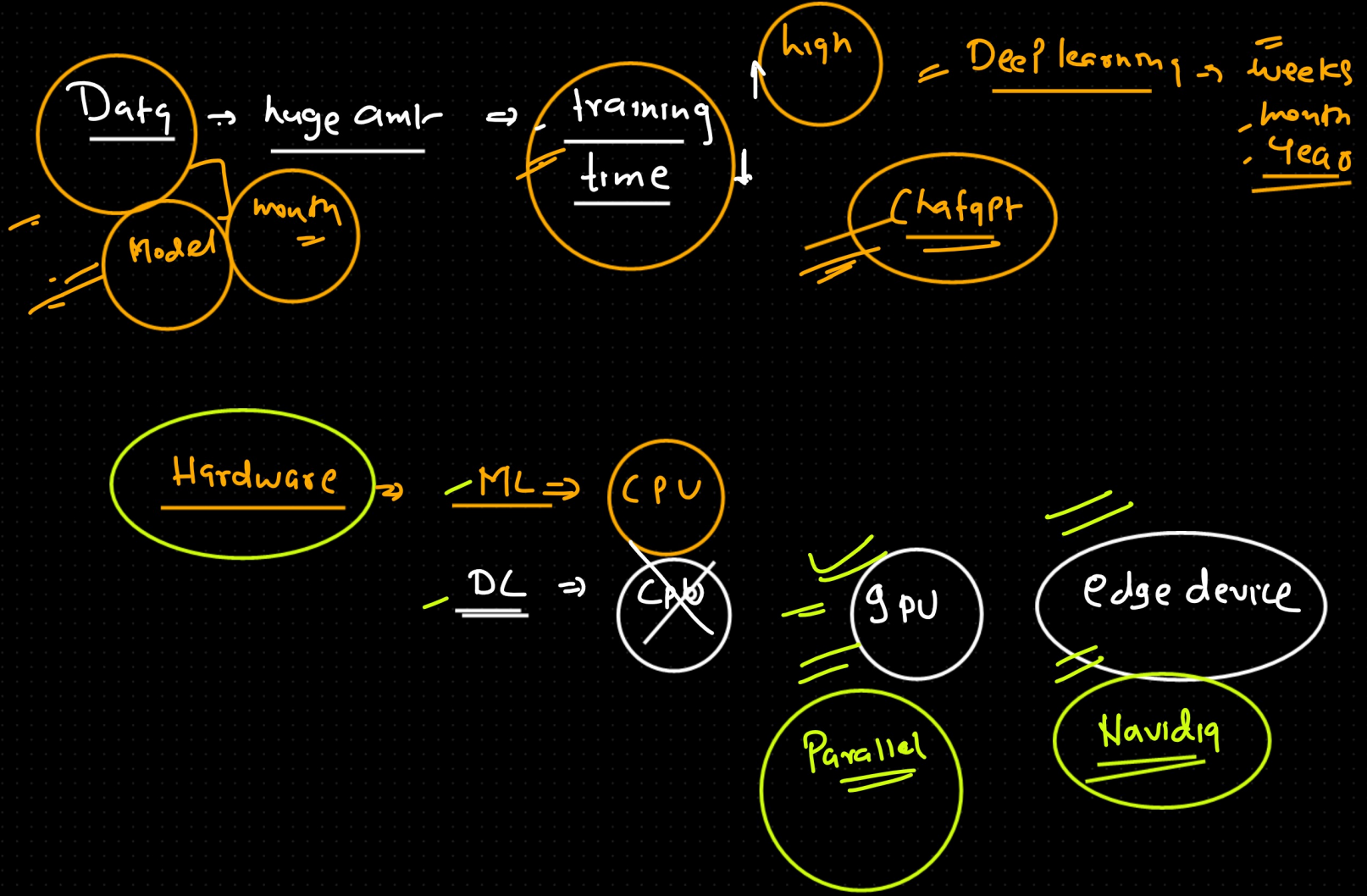


This is the graph that gives the pictorial representation of the increase in the volume of data with the increase in time

- M → Meta, Microsoft
- A → Apple, Amazon
- N → Netflix
- G → google
- L → Linkedin

Data coming to tech giants in timeframe 2005-2010 where there was a lot of data which were not labled. Hence labeling was done and then feeded to the alogirithms





In ML we used sklearn or sckit learn pythonic Library extensively for implementing the ML based concepts.
In the same way For implementing DL we are provided with 2 pythonic lib:
TensorFlow by Google and Pytorch by Facebook

- Tensorflow ⇒ 2015 ⇐ Dbrbeilf

Interal Proj. of google

In 2015 tensorflow was made open source with strong community support. But it was was difficult to implement for developers. So community made Keras on top of Tensor flow which simplified the process.
Tensorflow can be seen as Backend whereas, Kereas as frontend working on top of Tensorflow

2015 → Open Source ⇒ TF ⇐ Keras

google Likedin

- Community support

Fm

Backend

FE

2.0 ⇒ 2018

TF-2.x ⇒ TF+keras

COMMUNITY SUPPORT

Power ⇒ DL

TRANSFER LEARNING

Below are some of the examples of DL based models or frameworks which one can download and use are per their need. Also, one can perform fine tuning of these existing models as per the use case

- If someone already trained model on the data you want to use it for your purpose you can do it → fine tune Model → Download

Image classification ⇒ { Rest Net, Google Net }

Rest Net or Google Net are Image Classification framework/architecture/model that one can download and use for simpler implementation

Object Detection ⇒ { YOLO, SSD, Fast, faster RCNN }

Similarly we have framework for Object Segmentation, Image translation, Speech generation, Text classification etc

Object Segmentation ⇒ { Mask RCNN, U-Net, V-Net }

Image translation ⇒ Pix to Pix

Speech genration ⇒ Webnet

text classification ⇒ BERT

HISTORY ⇒ Old History + New History

Modern

In 1956 Frank Rosenblatt is the person who coined the ideal of artificial neurons

McCulloch, Pitts, Frank Rosenblatt

1950 - 1960 ⇒ NN ⇒ AN ← Perceptron

Geoffery Hinton is the father of modern of AI — Hinton

1970 - 1980 ⇒ Deep NN = MLP = BP (Back Propagation)

Yann LeCun is the father of CNN/Compter vision who created model to classify the MNIST dataset

father of CNN → YannLecun

1990 - 1995 ⇒ Lenet — MNIST

Hinton, Deep belief Network → 2006 DNN, Unsupervised

Hardware, GPU → 2012

Imagenet ← 25-30% ; Alexnet — GPU ⇒ 60-70%

2014 ← GAN, RL — Ian Goodfellow is the father of GAN

2015 ← AlphaGo — Google

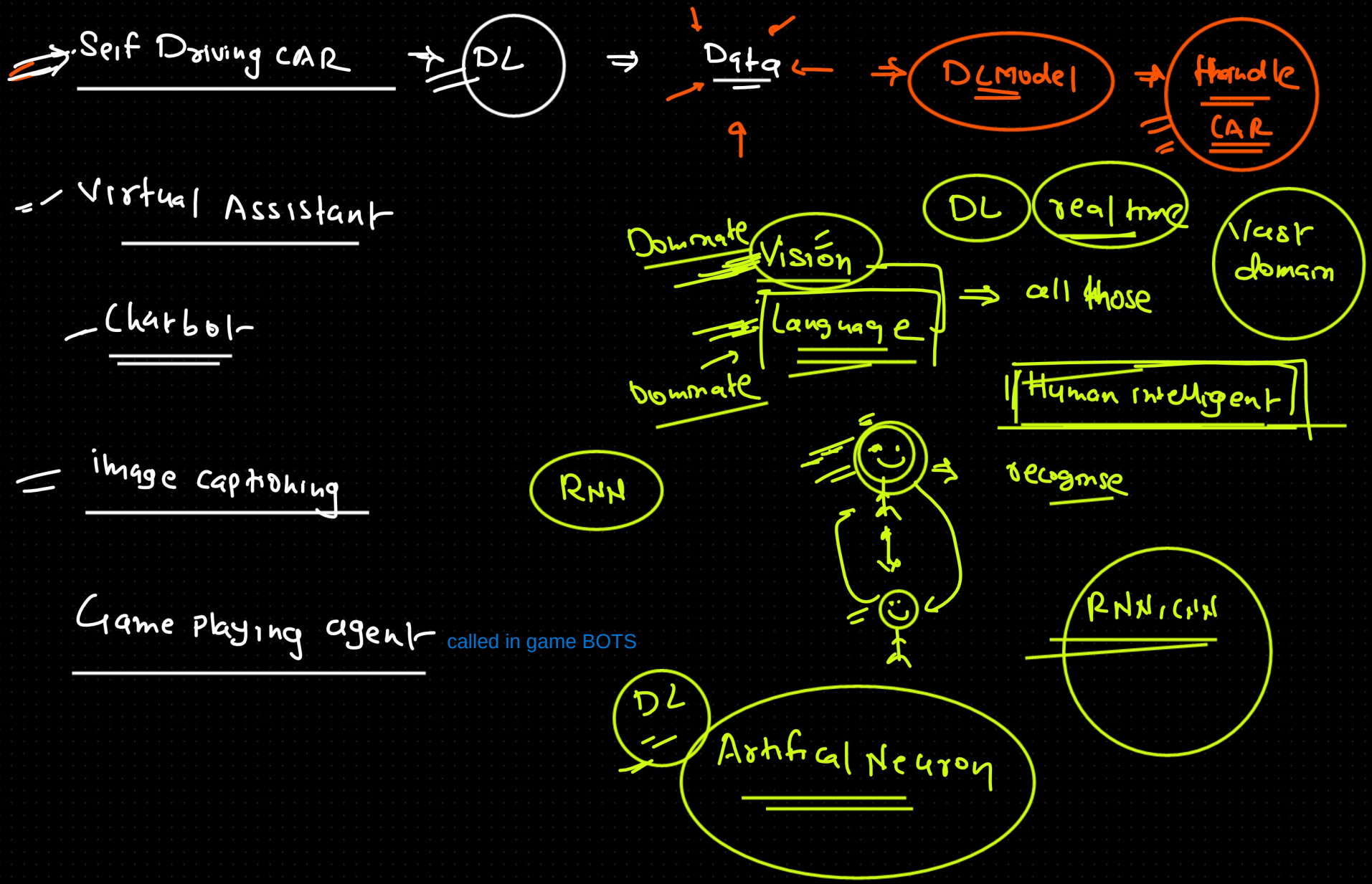
Once there was a competition where computer vision was involved. No one was able to pass the accuracy level of 25-30%. One Person trained the model on top of special hardware like GPU and observed that accuracy improved drastically to 60-70%

This is the point where Deep Learning and use of special hardware started gaining popularity.

= (2017, 2018)

BERT, transform

APPLICATIONS


Self Driving CAR
DL
Data
DL Model
Handle
CAR
Virtual Assistant
Chatbot
Dominate
Vision
Language
Dominate
DL
real time
Vast domain
all those
Human intelligent
image captioning
RNN
recognise
RNN, CNN
Game playing agent
called in game BOTS
DL
Artfical Neuron

DL

Roadmap →

ANN

SE

Perceptron

MULTIlayer perceptron

Python → TF, PT

activation

Loss

optimizer

Vanishing gradient

exploding gradient

Regularization, Normalization (L_1, L_2, BatchN, Dropouts)

- Hyperparameter tuning (keras tuner)
- tensorboard

CNN =

1. Convolution
2. Poolings
3. Kernals
4. LeNet, AlexNet, Vgg, ResNet, GoogleNet (These are some of the CNN frameworks)
5. Practical ⇒ CNN App
6. Object Detection — SSD, RCNN, YOLO (These are some of the frameworks of the Object Detection)
7. Obj Seg
8. tracking, OCR
9. GPU, Edge devices
10. Projects

OpenCV

RNN

1. RNN
2. LSTM, GRU
3. Encoder, Decoder
4. types of RNN
   (seq to seq)
5. attention
6. Self attention, transformer
7. BERT (LLM)
8. GPT
9. OpenAI (Chatgpt)
11. Project

GAN

R2

Wordembedding

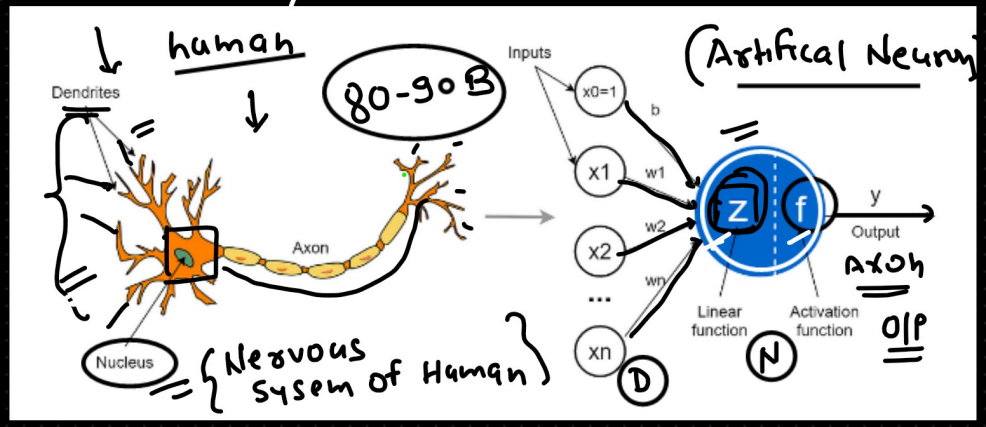
Word2 vec

elmo

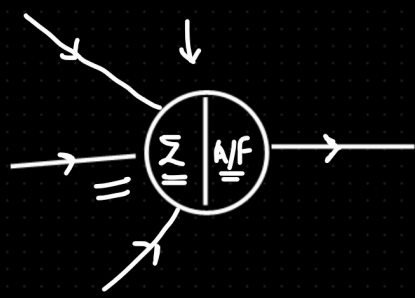
= Hugging face

Network — Inspired

Array and tensor

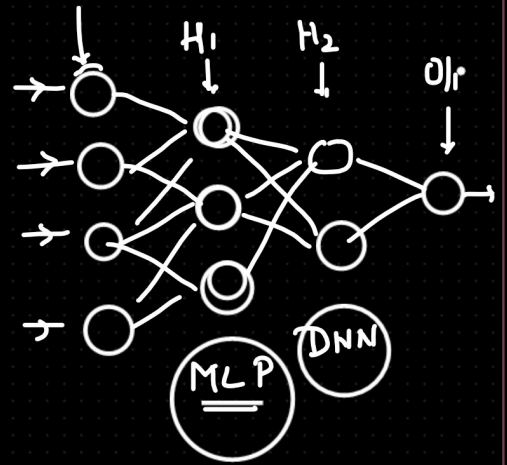


Perceptron



Summation + Activation

I/P

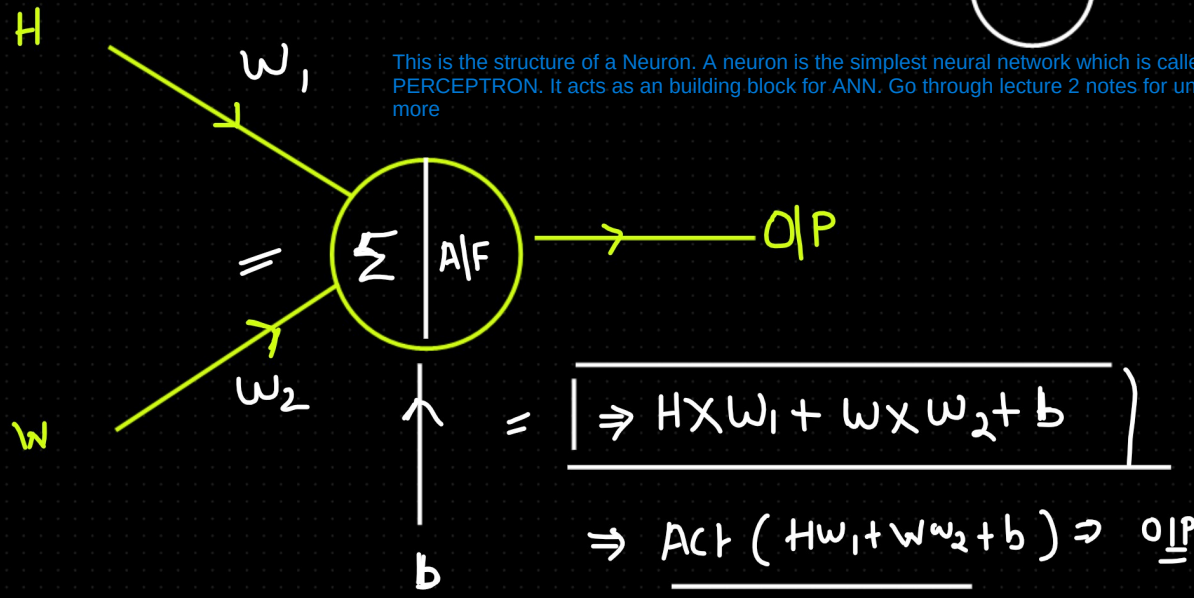


Height | weight | O/No

Training <= DL

Prediction <= DL

{ Feed forward NN
BP }



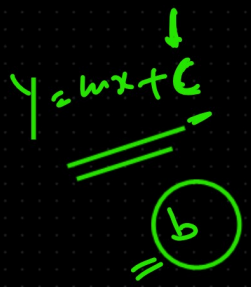
This is the structure of a Neuron. A neuron is the simplest neural network which is called PERCEPTRON. It acts as an building block for ANN. Go through lecture 2 notes for understanding more

Linear reg ⇒ $Ax + By + C = 0$ ⇒ equation of line

Step Fln

$x \geq 0$ Obese

$x < 0$ Not obese



Sigmoid, tanh, Softmax, Step, Linear

Loss

BP ⇒ GD, SGD, Minibatch, Adagrad, RMS prop

SGD wid. Momentum

Adadelta, ADAM

Pytorch, TF