Making Predictions With The Speech Recognition System By Using Deep Learning

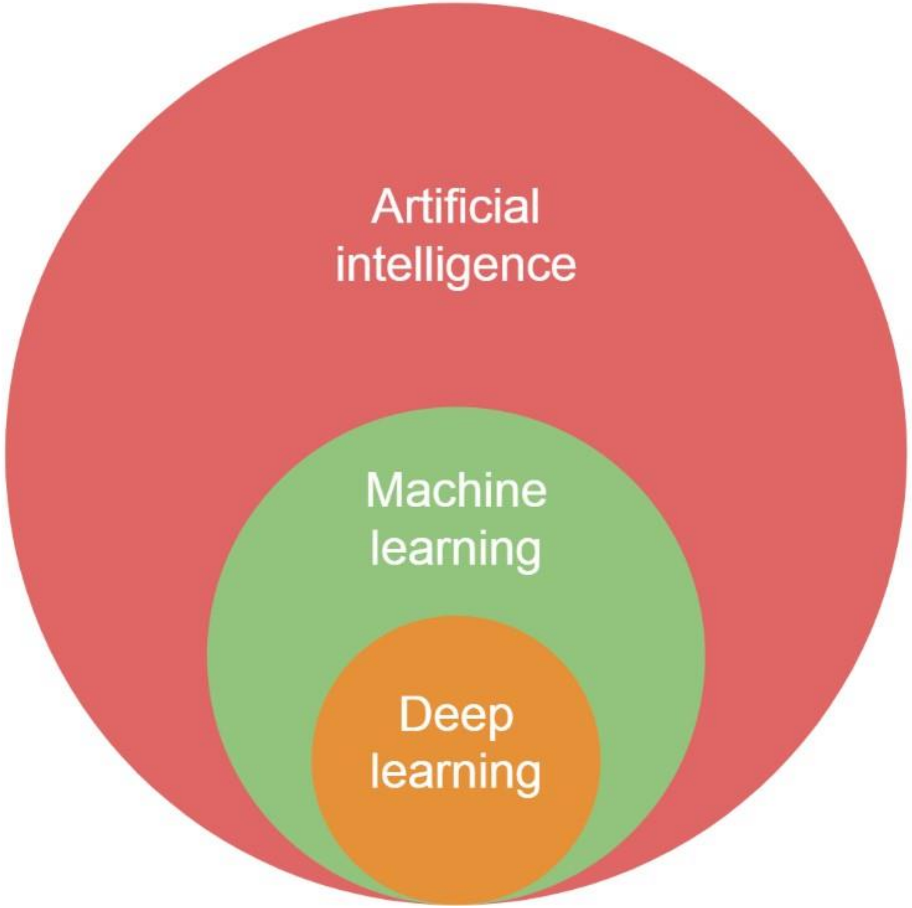
Members:

Muhammed Talha Yaşar

Recep Berkay Karagöz

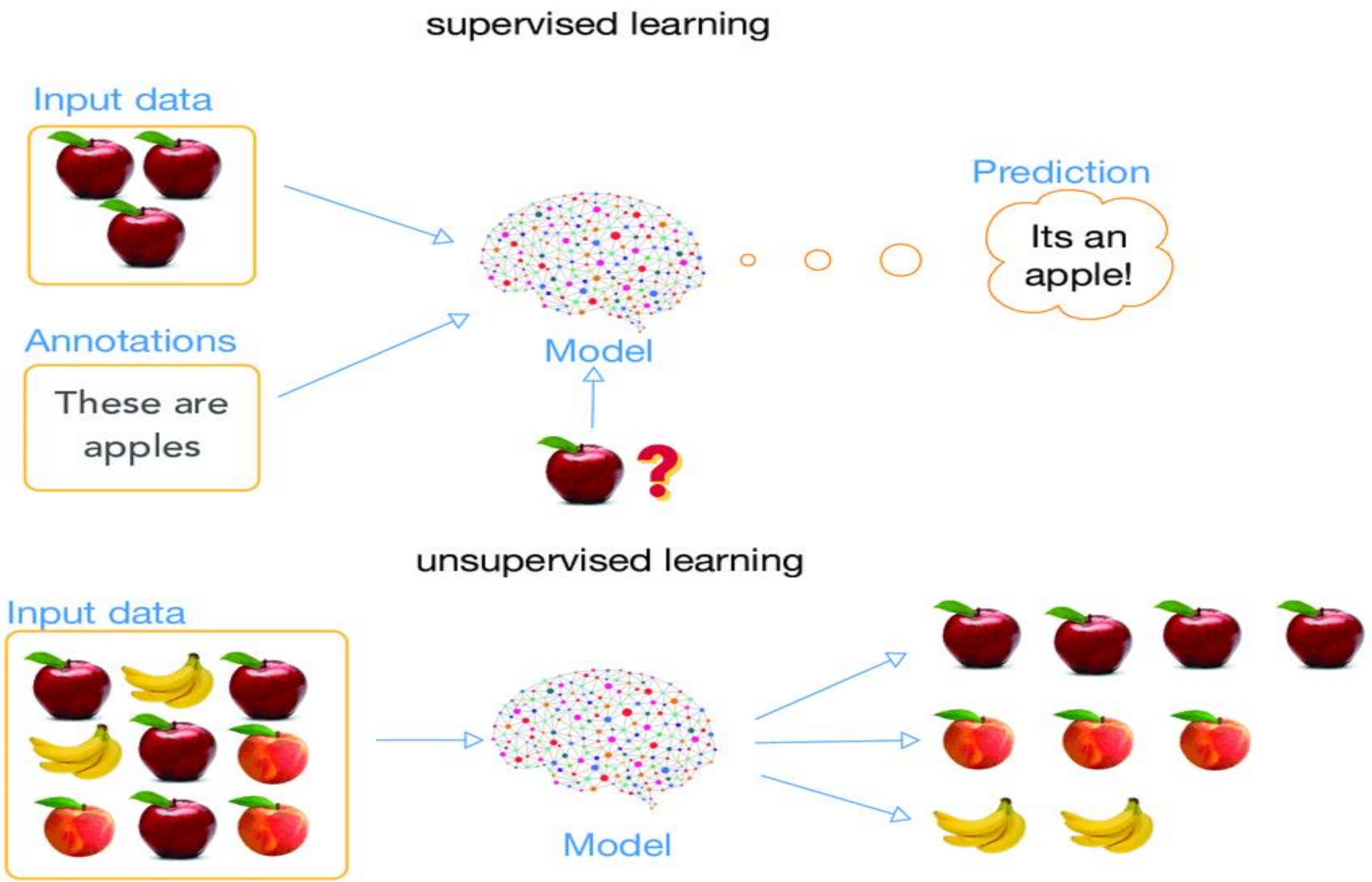
Batuhan Bayram Altıparmak

Veysel Karani Yanpala

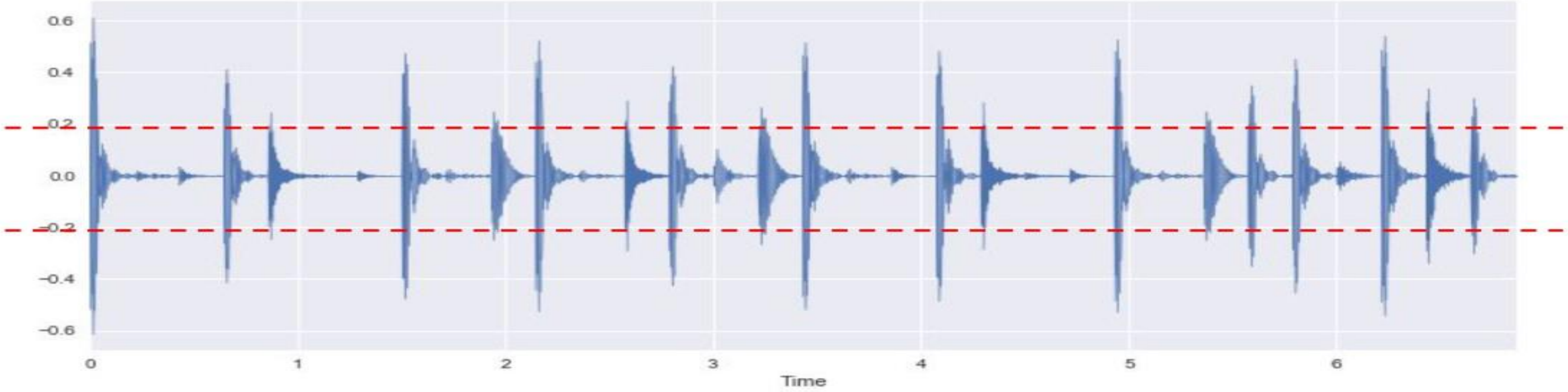

Artificial
intelligence
Machine
learning
Deep
learning

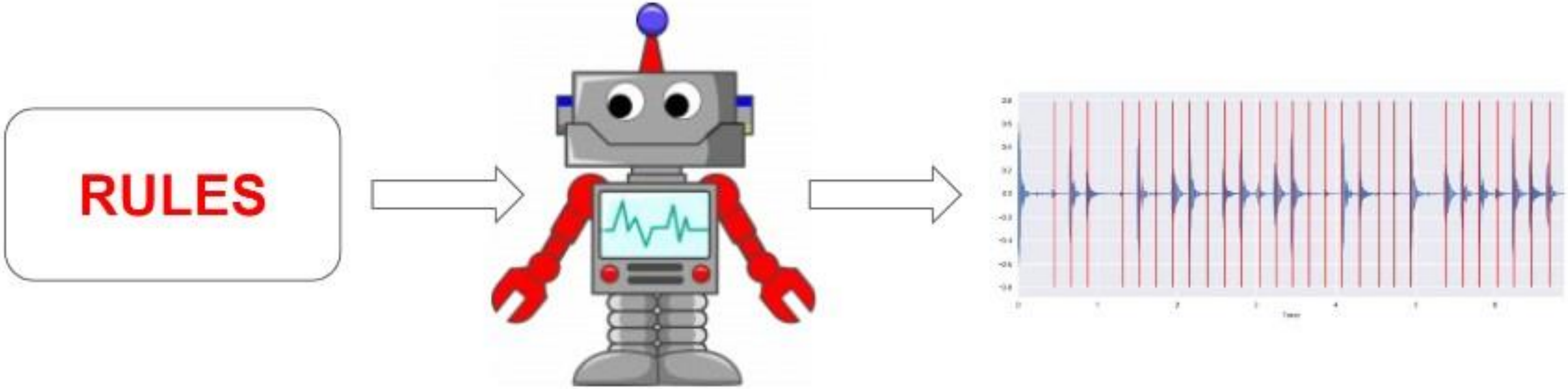
Types Of Machine Learning

- Supervised Learning : Tell The Algorithm
- Unsupervised Learning: Don’t Tell The Algorithm
- Reinforcement Learning : Learns through rewards or mistakes.

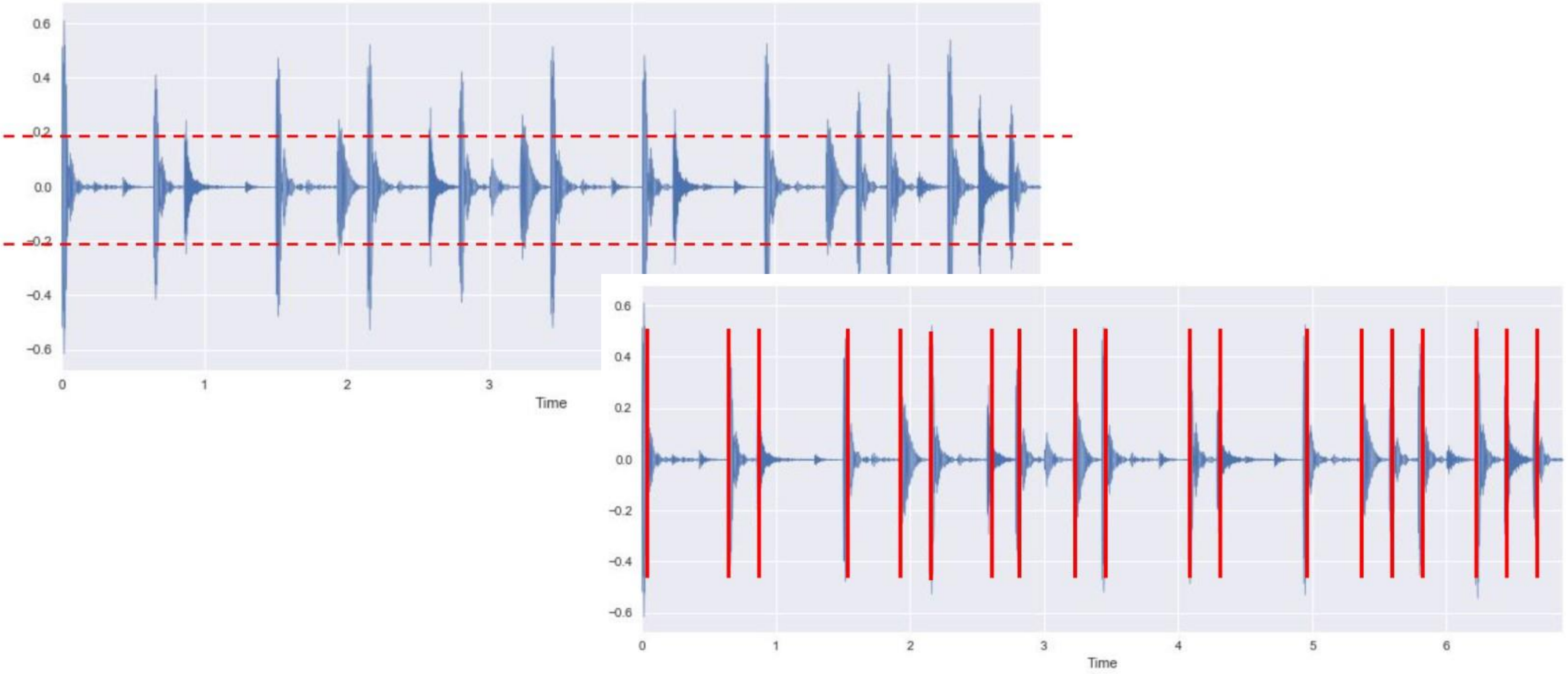

supervised learning
Input data
Annotations
These are apples
Model
Prediction
Its an apple!
unsupervised learning
Input data
Model

Supervised Learning
Rule: Don’t Pass Larger Than 0.2





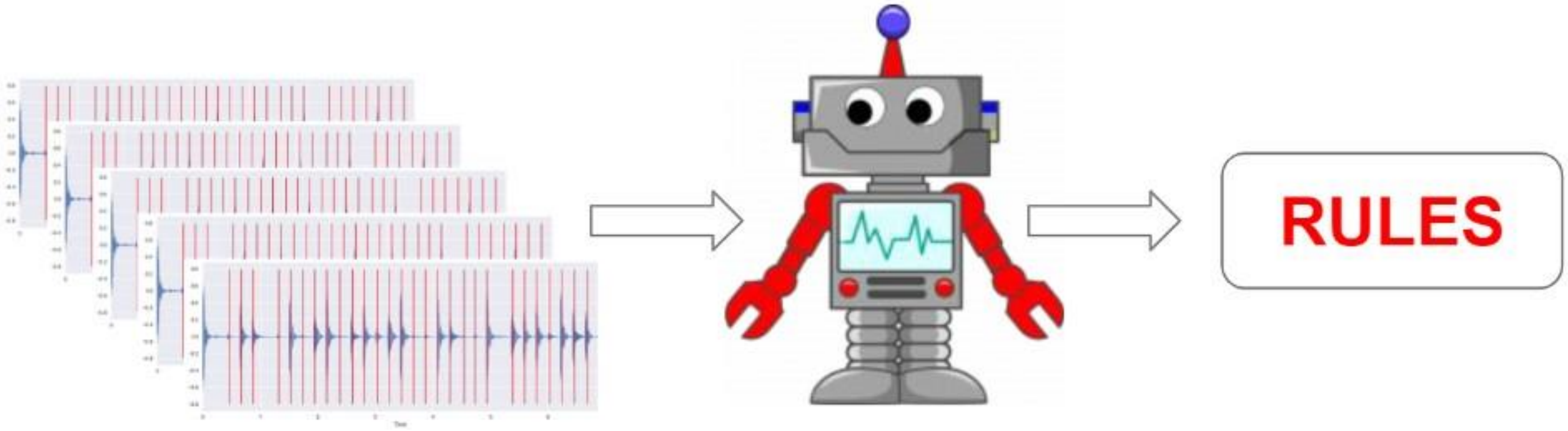
Rule Applied

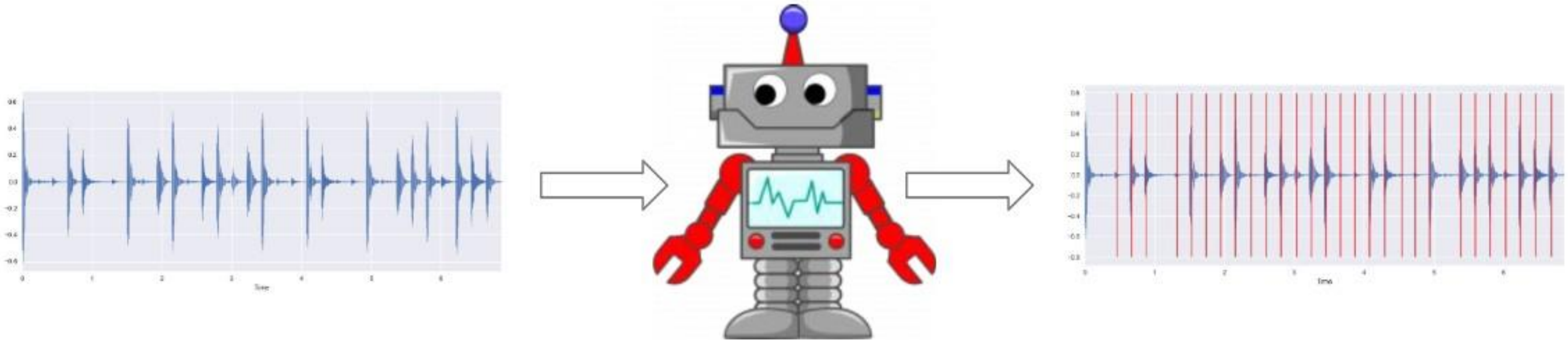


Supervised Learning

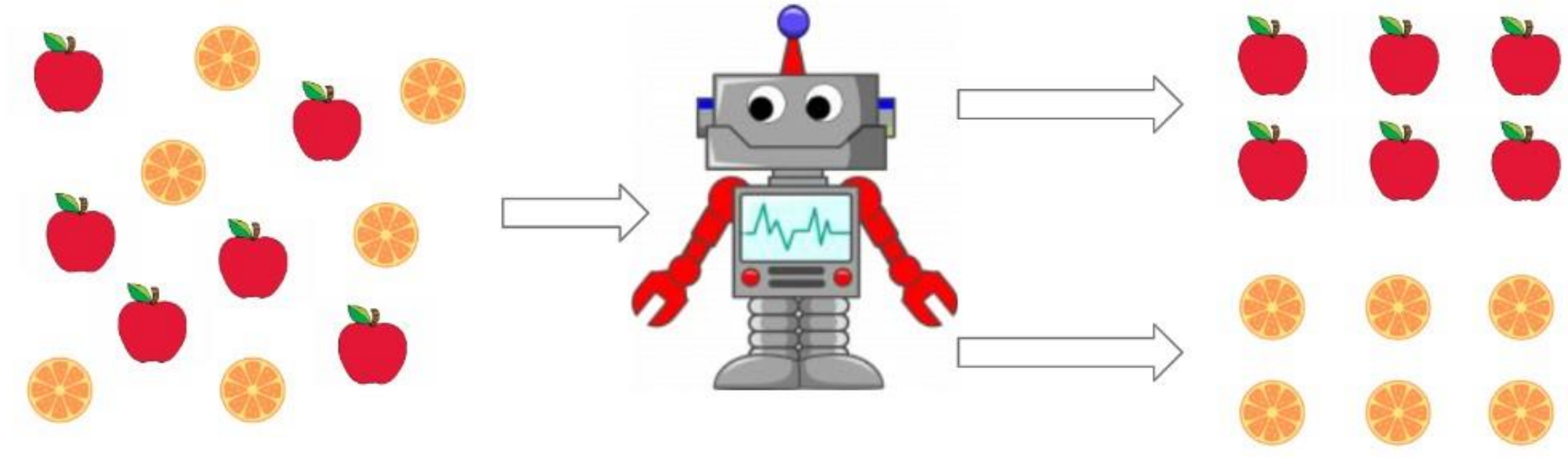
Taking data+Algorithm.

Extract Rule.





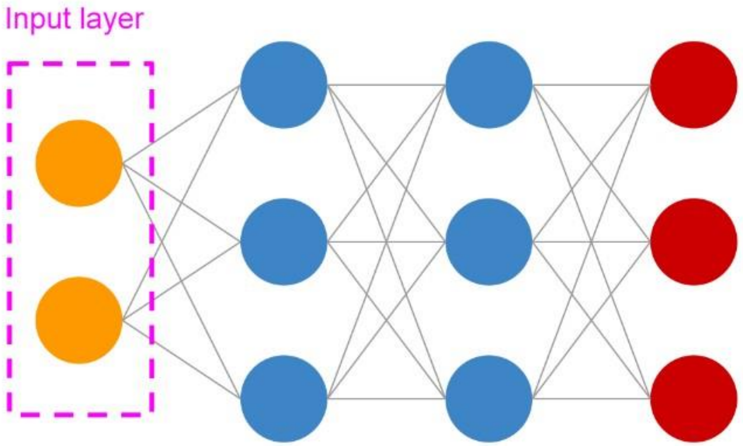
Unsupervised Learning

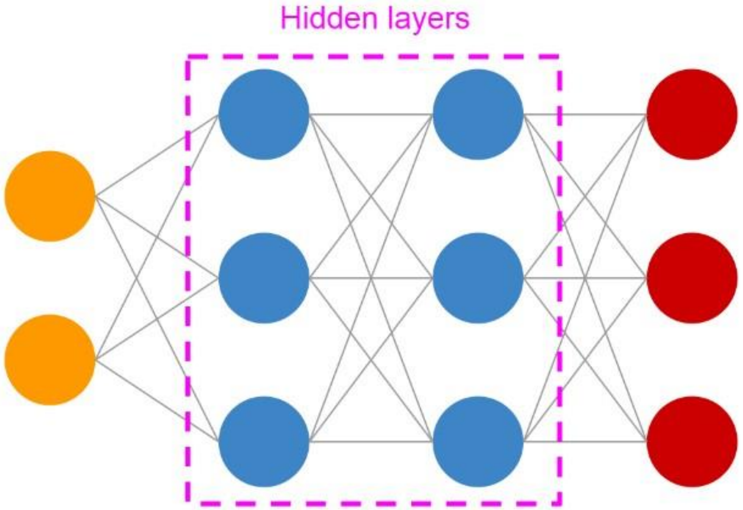


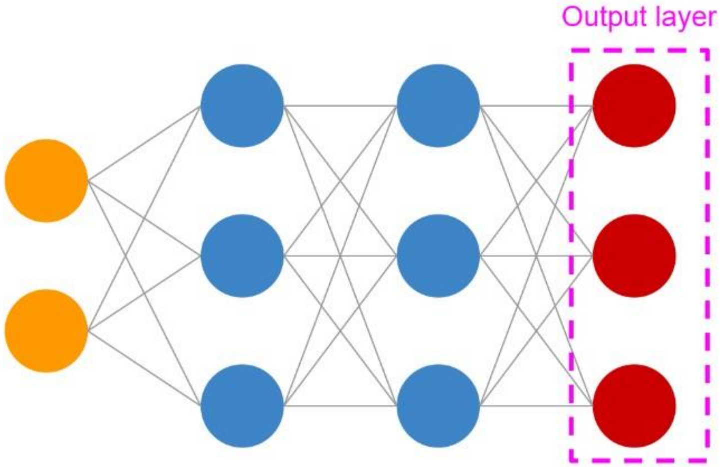
Reinforcement Learning

Learns through rewards.

Neural Network- One Algorithm of Machine Learning





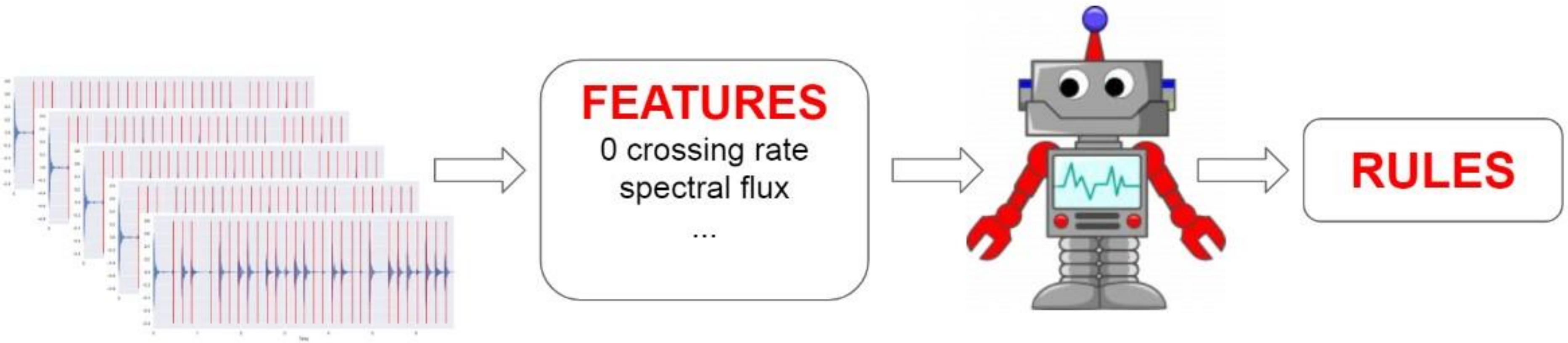


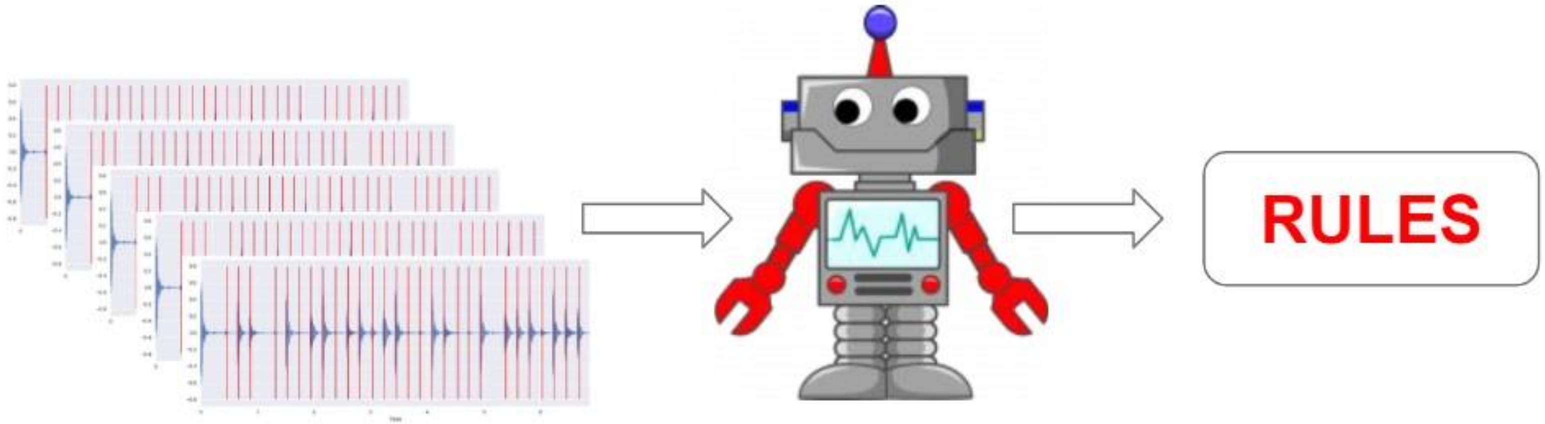
Deep Learning

- Subset of ML.
- Deep neural network.
- >1 hidden layer.

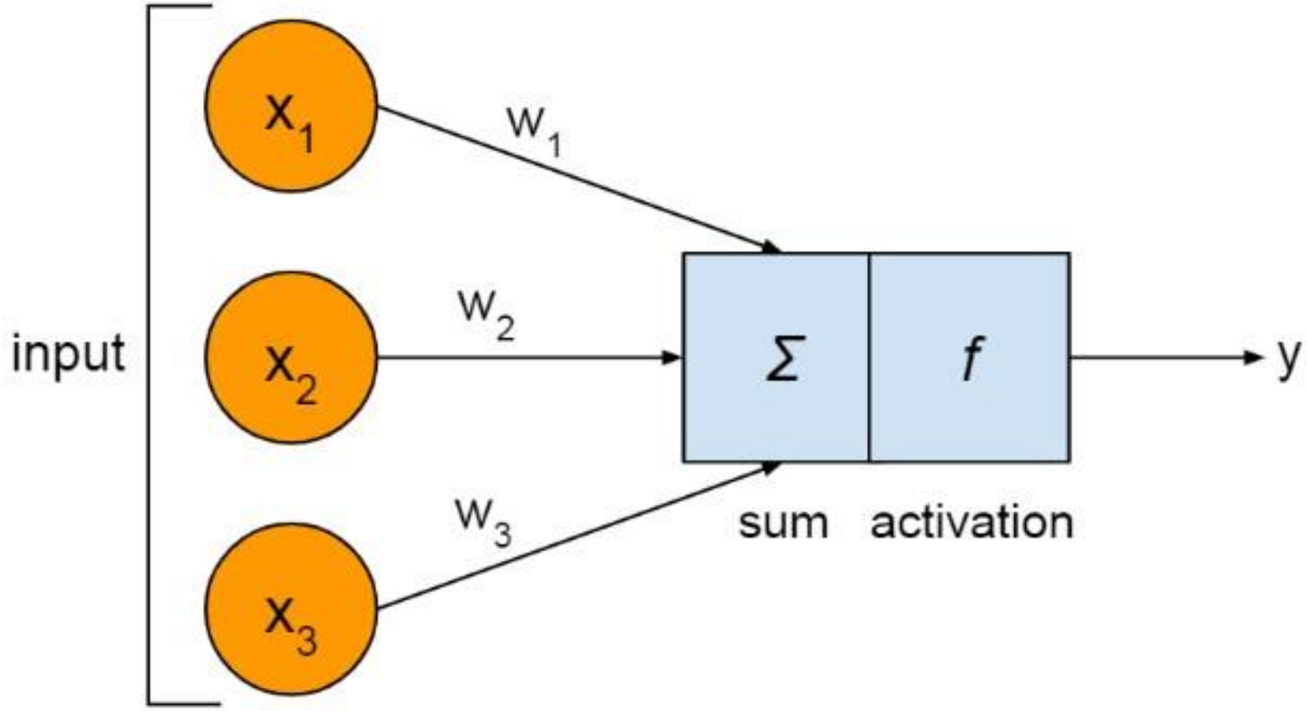
A little bit more ML vs DL

Determine feature for ML.
Row data input for DL.





Artificial Neuron



Inputs

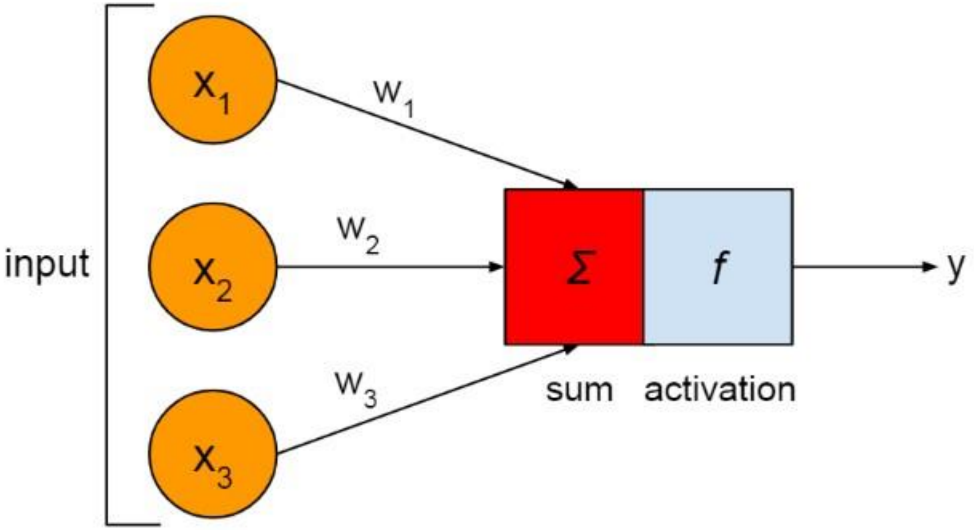
Weights

Functions

Output

Sum Function

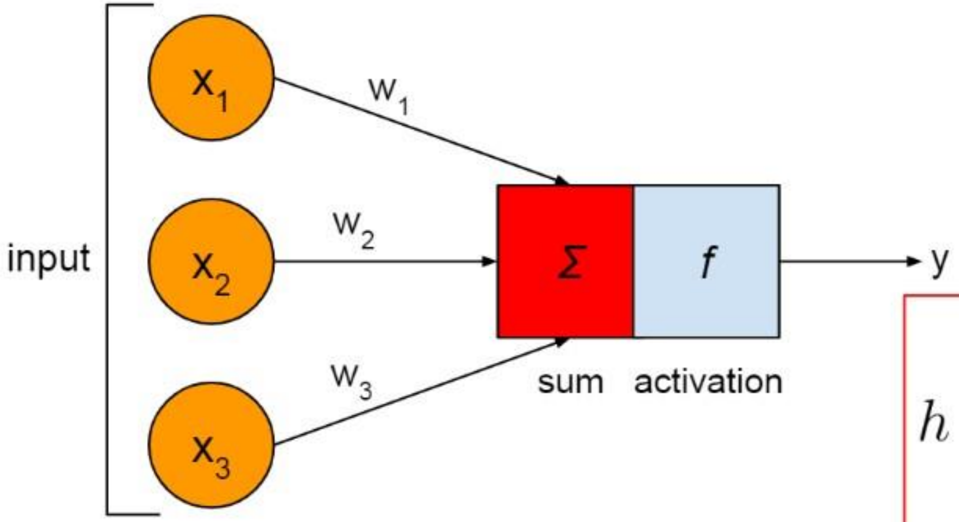
h=All the inputs multiplied by their ways.
h= Net input.



$$h = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3$$

Sum Function

Dot Product

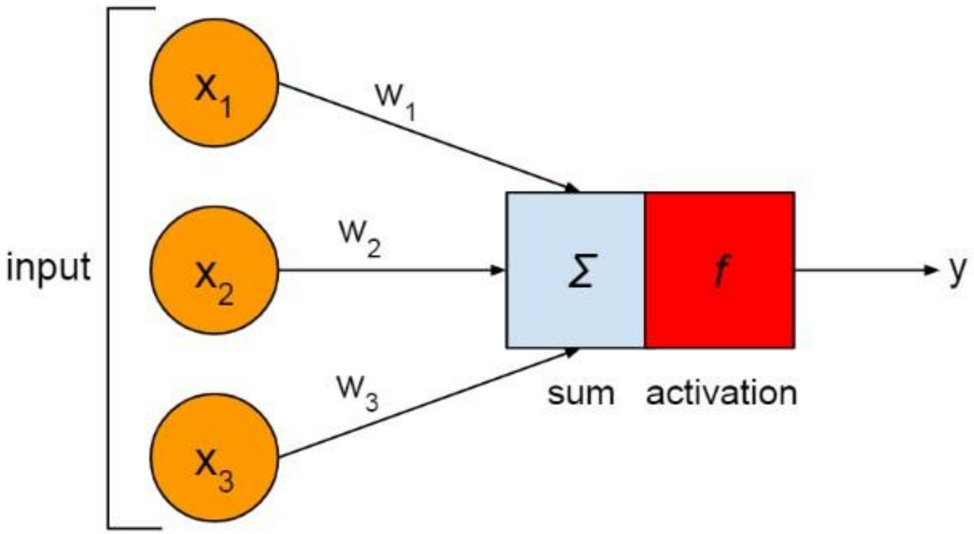


$$h = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$\boxed{h = \mathbf{x} \cdot \mathbf{w} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = x_1 w_1 + x_2 w_2 + x_3 w_3}$$

Activation Function

Output y is function of activation function.

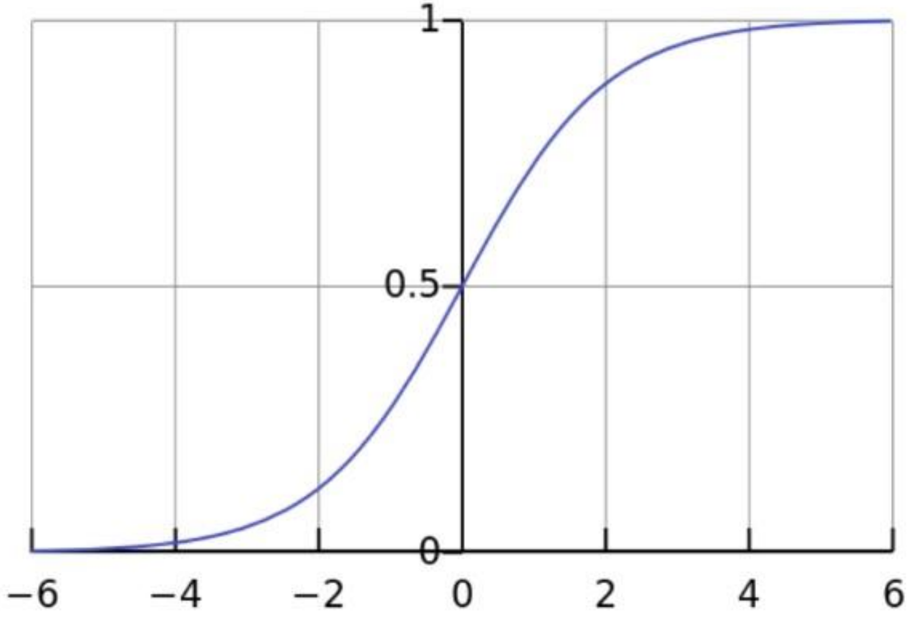


$$h = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$\boxed{y = f(h) = f(x_1 w_1 + x_2 w_2 + x_3 w_3)}$$

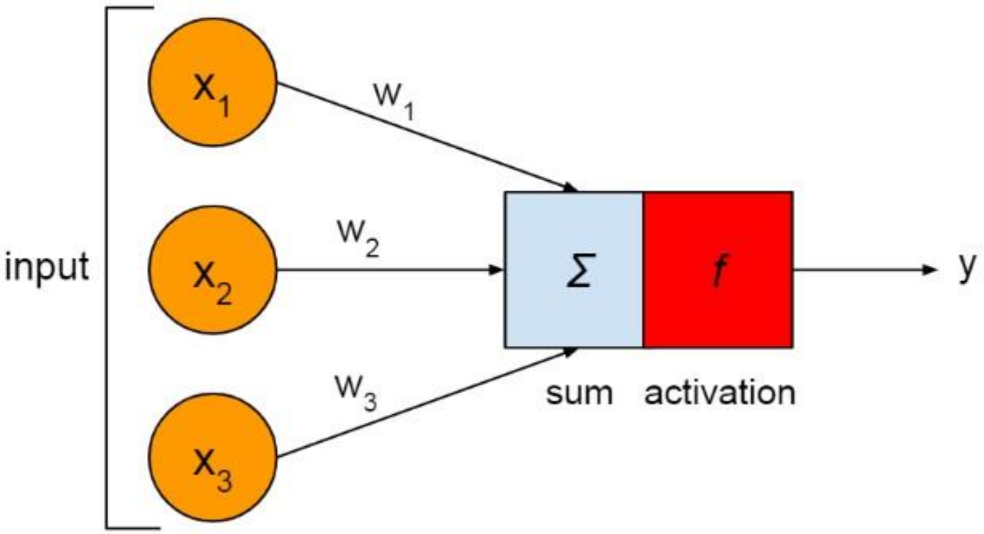
Activation Func. As Sigmoid Function – WHY?

Bounded 0 to 1
Smooth
No discotinuity
Can be differentiated



$$y = \frac{1}{1 + e^{-x}}$$

Output

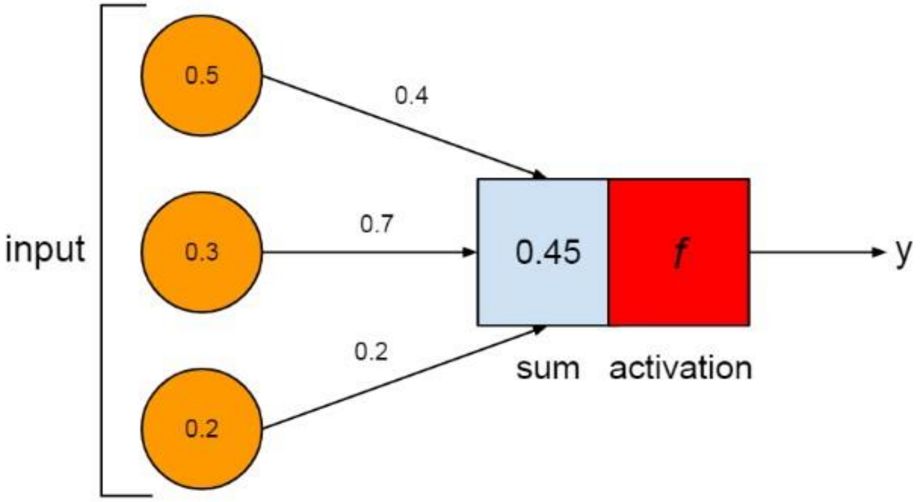


$$h = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$y = f(h) = f(x_1 w_1 + x_2 w_2 + x_3 w_3)$$

$$\boxed{y = \frac{1}{1 + e^{-(x_1 w_1 + x_2 w_2 + x_3 w_3)}}}$$

Example



$$h = x_1 w_1 + x_2 w_2 + x_3 w_3 = 0.5 \cdot 0.4 + 0.3 \cdot 0.7 + 0.2 \cdot 0.2 = 0.45$$

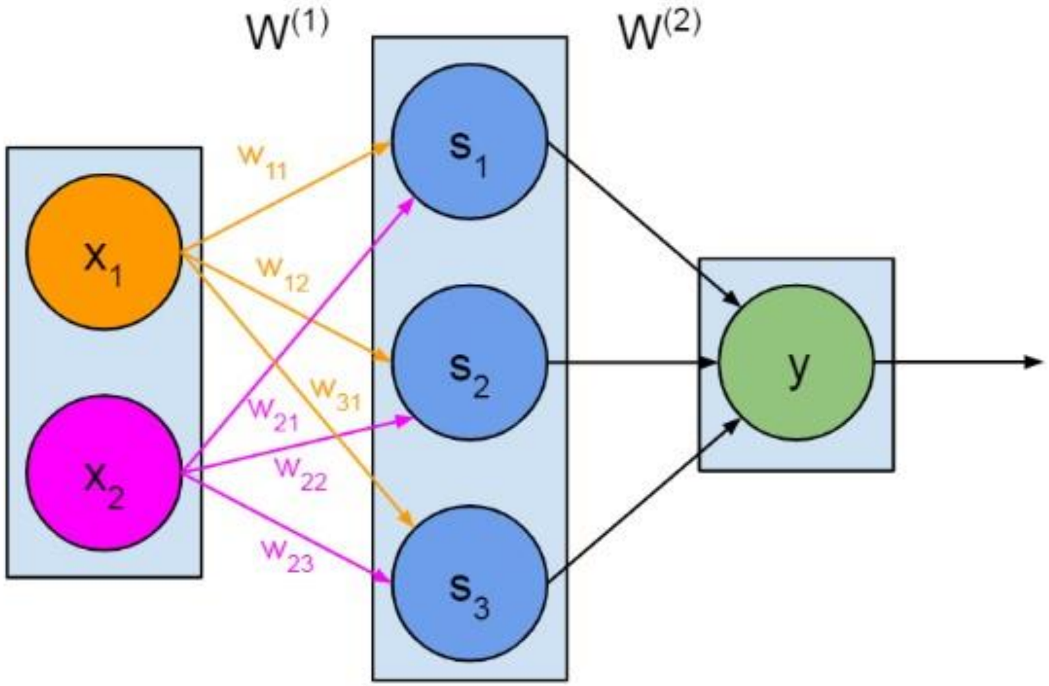
$$\boxed{y = \frac{1}{1 + e^{-0.45}} = 0.61}$$

0.5 0.3 0.2 are inputs

0.4 0.7 0.2 are weights

Output y= activation(inputs,weights)

Weights



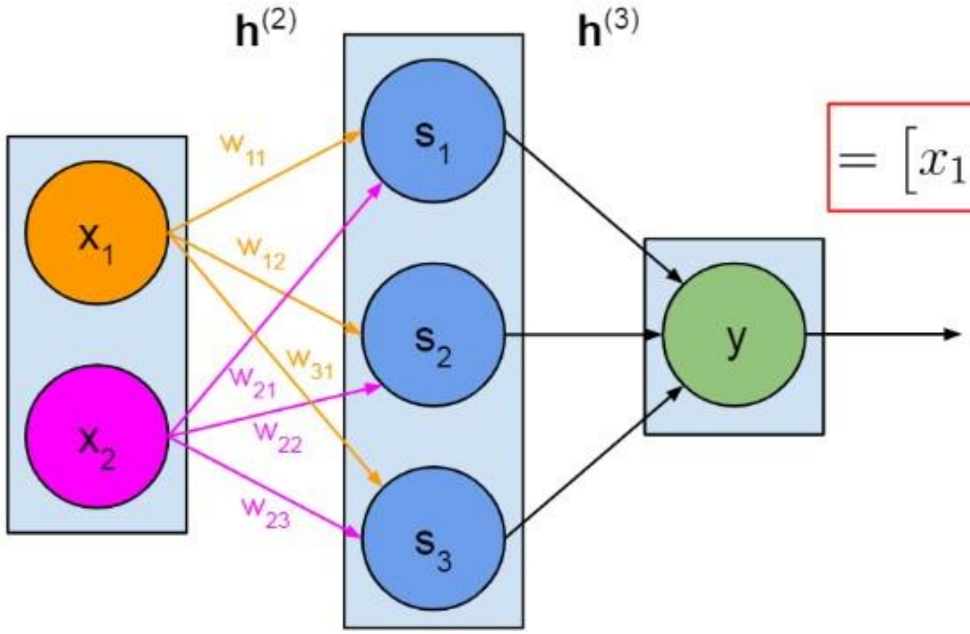
$$W^{(1)} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix}$$

Net Input

Matrice multiplication
Find all inputs for all s and sum.
h1=>s1 h2=>s2 h3=>s3

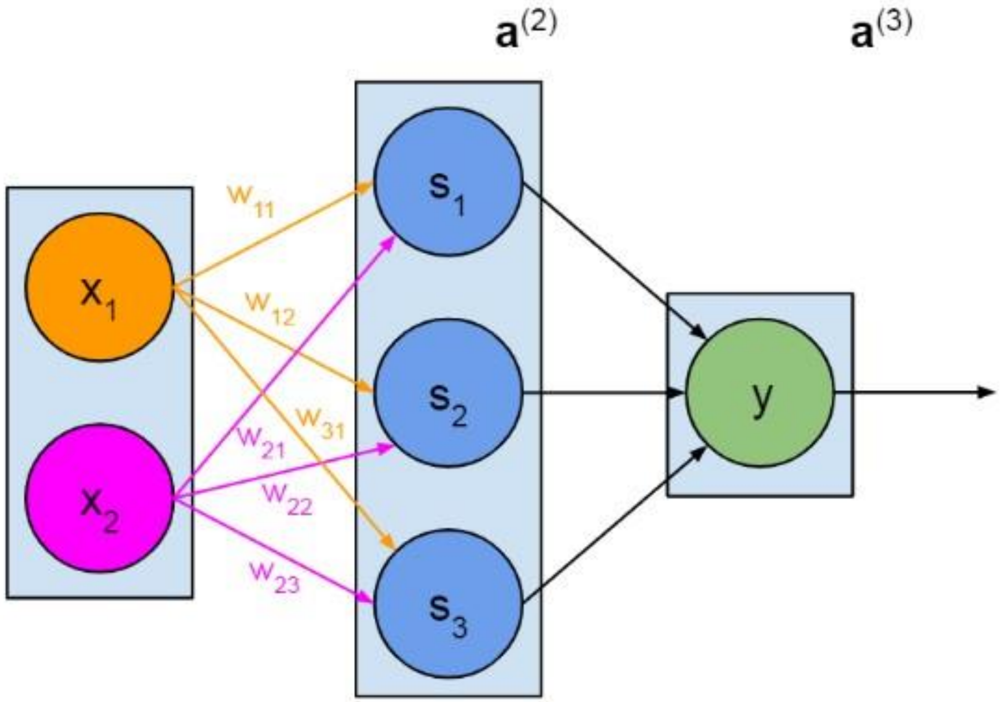
$$\mathbf{h}^{(2)} = \mathbf{x}W^{(1)} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix}$$

$$\boxed{= \begin{bmatrix} x_1 w_{11} + x_2 w_{21} & x_1 w_{12} + x_2 w_{22} & x_1 w_{13} + x_2 w_{23} \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}}$$



Activation

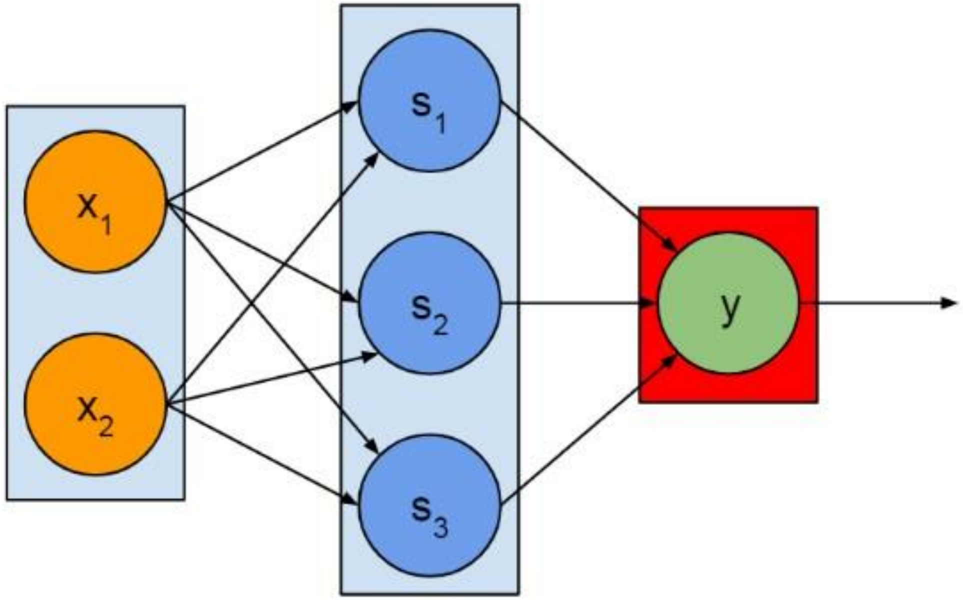
a is matrice = output of second layer
Activation applied for s1,s2,s3



$$\mathbf{a}^{(2)} = f(\mathbf{h}^{(2)})$$

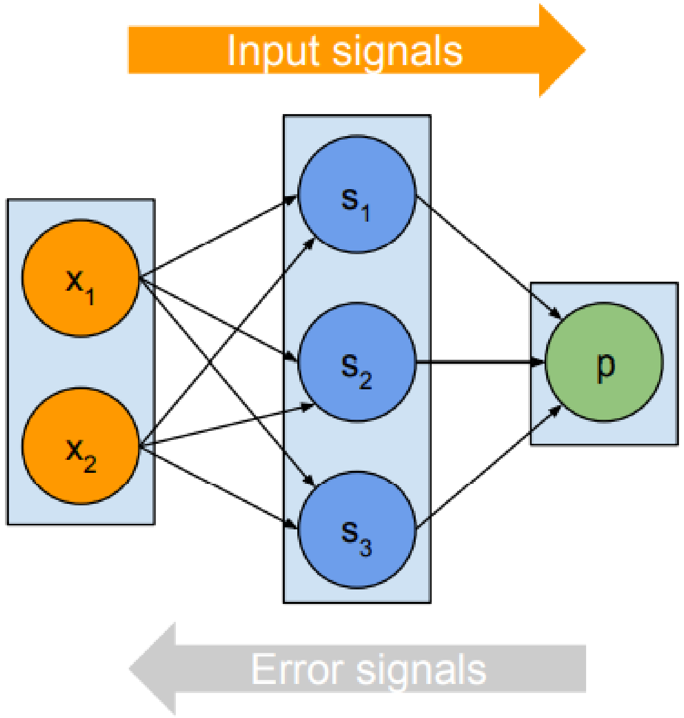
3rd Layer

a(2) acts as an input
h(3): net input of y



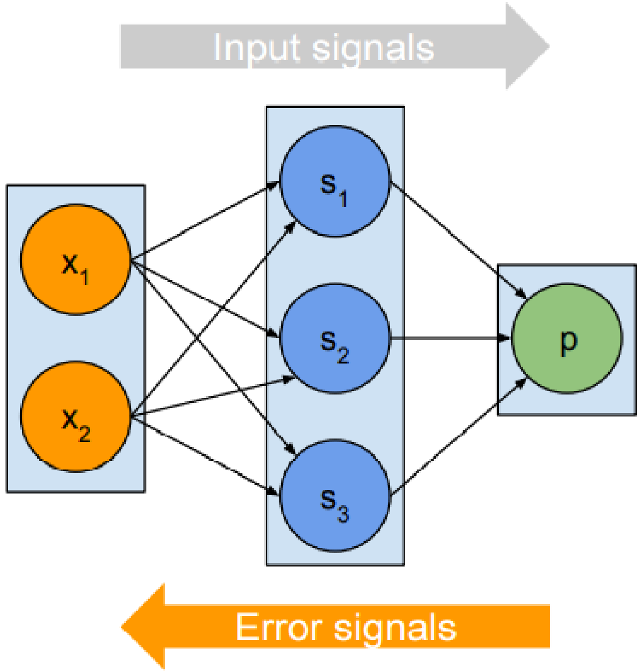
$$\mathbf{h}^{(3)} = \mathbf{a}^{(\mathbf{2})} W^{(2)}$$
$$y = f(\mathbf{h}^{(3)})$$

TRAINING NEURAL NETWORK



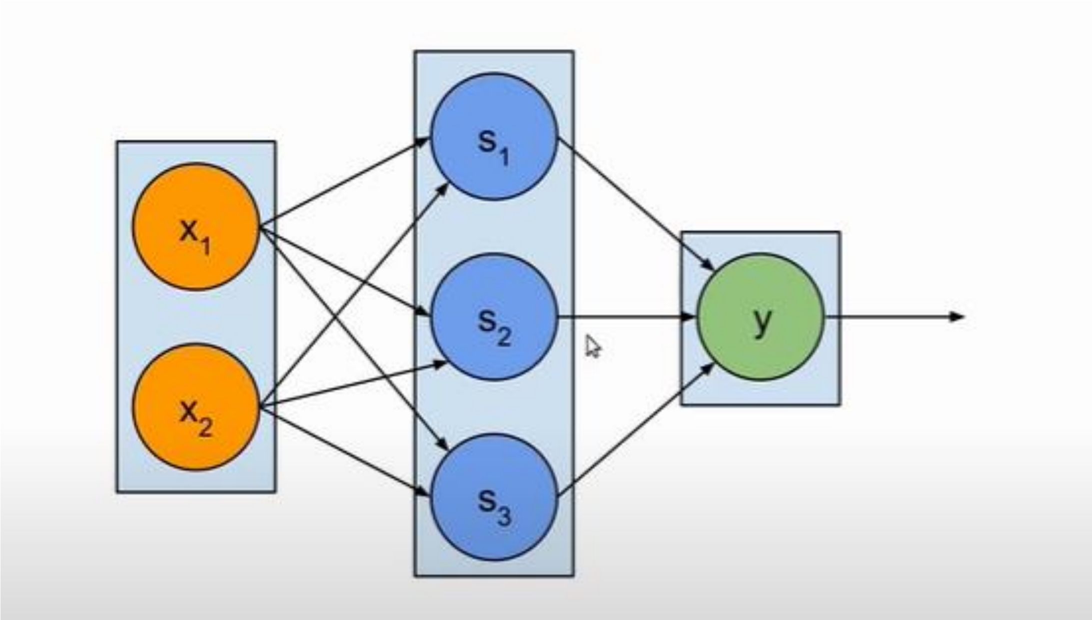
- Get prediction
- Calculate Error

INTRODUCTION TO BACK PROPAGATION



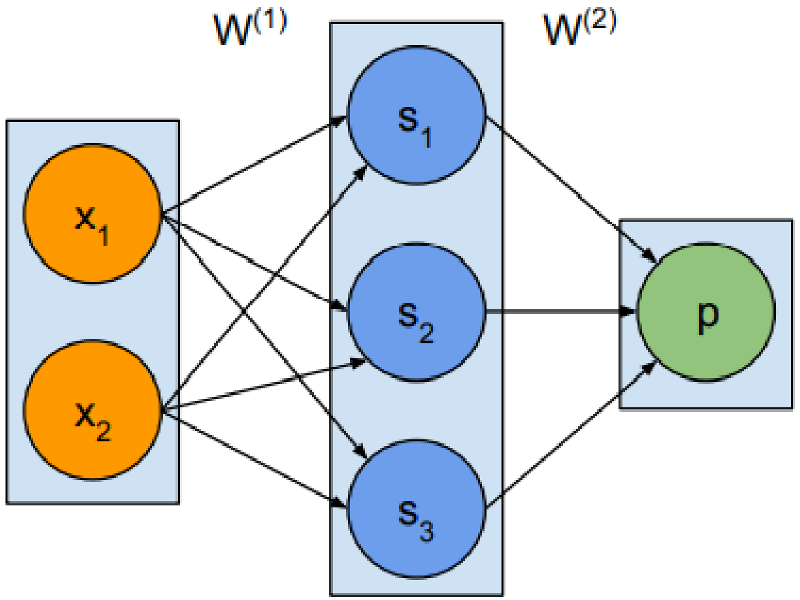
- Calculate error function of error function over the weights
- Update parameters

CALCULATE ERROR FUNCTION



$$E = E(\mathbf{p}, \mathbf{y}) = \boxed{\frac{1}{2}(\mathbf{p} - \mathbf{y})^2}$$

CALCULATE GRADIENT OF ERROR FUNCTION

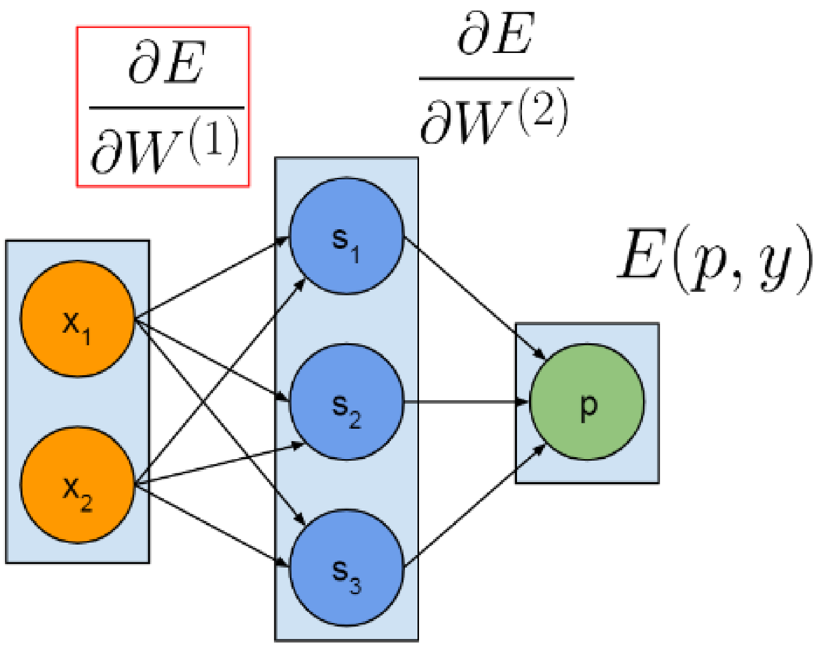


$$\frac{\partial E}{\partial W^{(n)}}$$

$$F = F(\mathbf{x}, W)$$

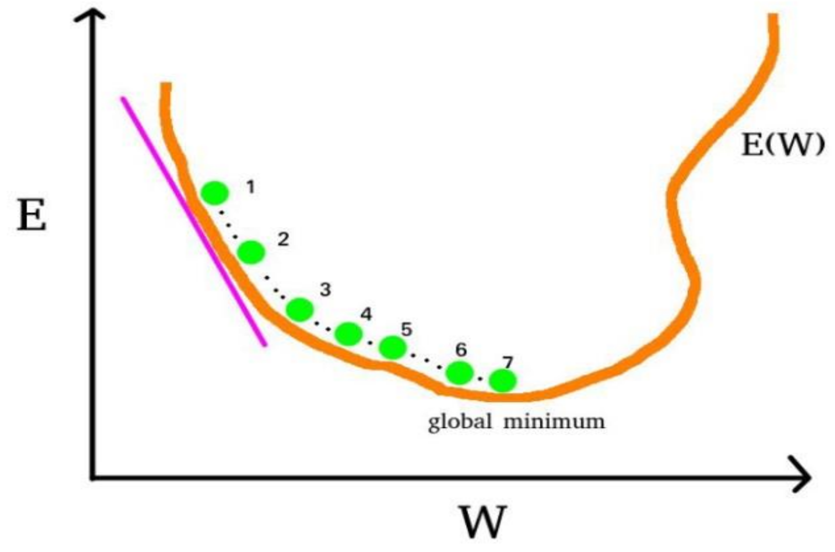
$$E = E(\mathbf{p}, \mathbf{y}) = E(F(\mathbf{x}, W), \mathbf{y})$$

BACKPROPAGATION



- ❖We calculated error.
- ❖We use errors to calculate first derivative with respect to w2 and afterly w1

GRADIENT DESCENT

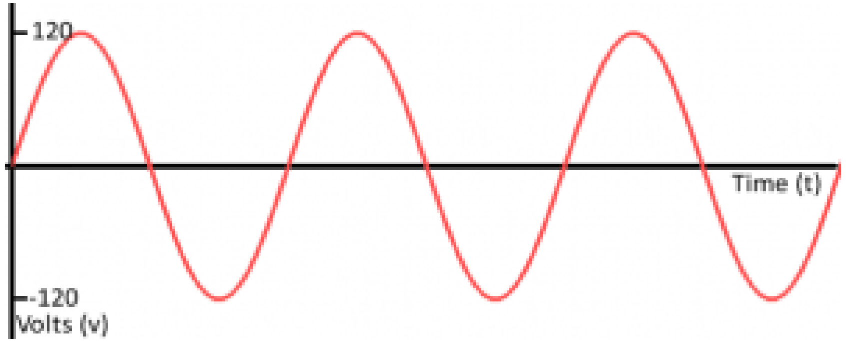


E(W) : Error Function

Purple line : Gradient

Take a step in opposite direction to gradient to get minimum error

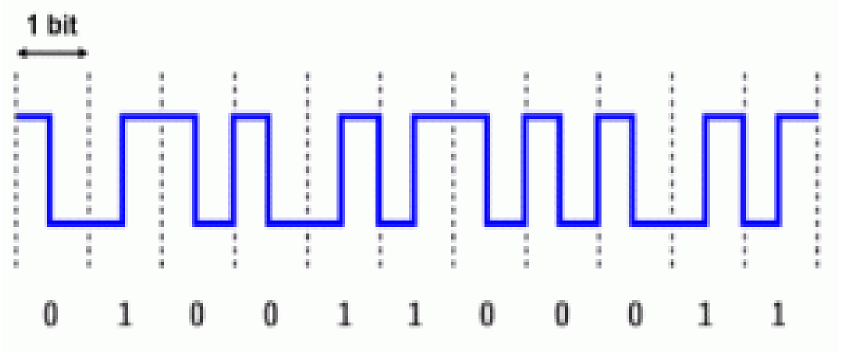
SIGNAL TYPES



Analog Signal:

It is continuous and uniterrupted

It consist of infinite point.



Digital Signal:

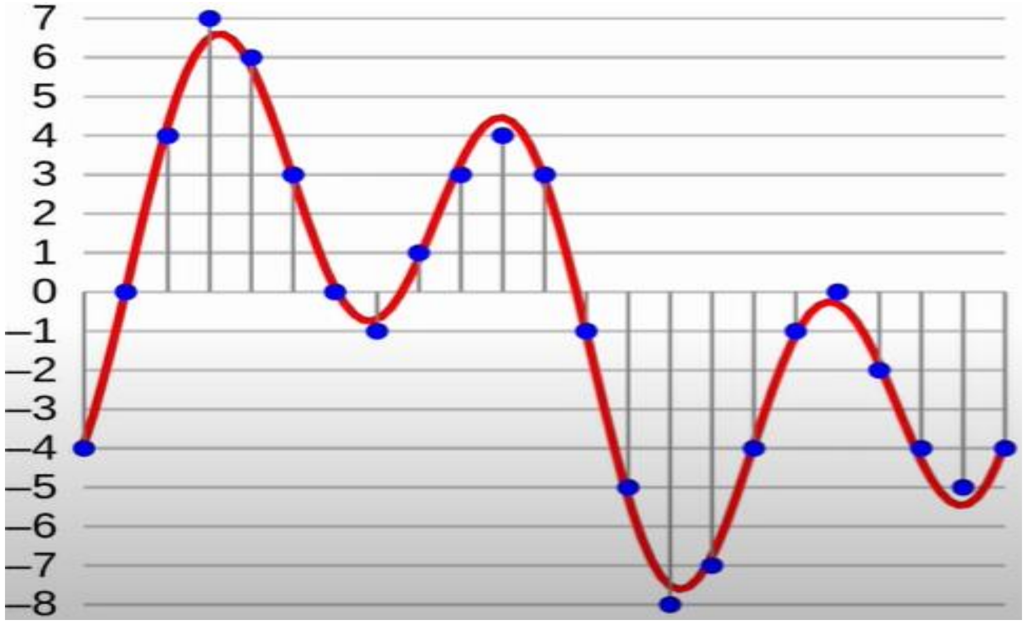
It is a discrete time signal

A set of values is available and it is temporary.

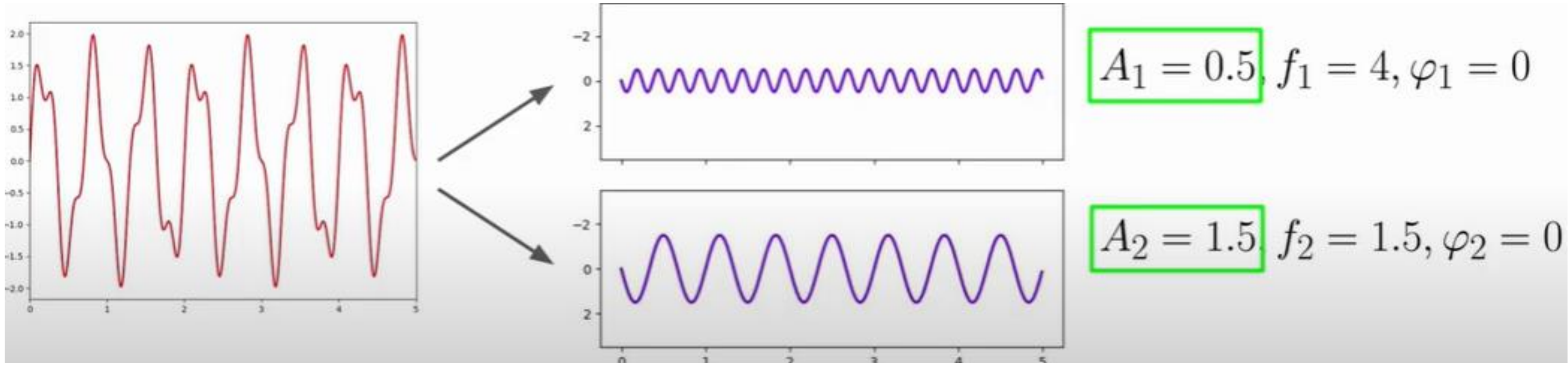
Analog to digital conversion consist of two steps:

1:Sampling

2:Quantization

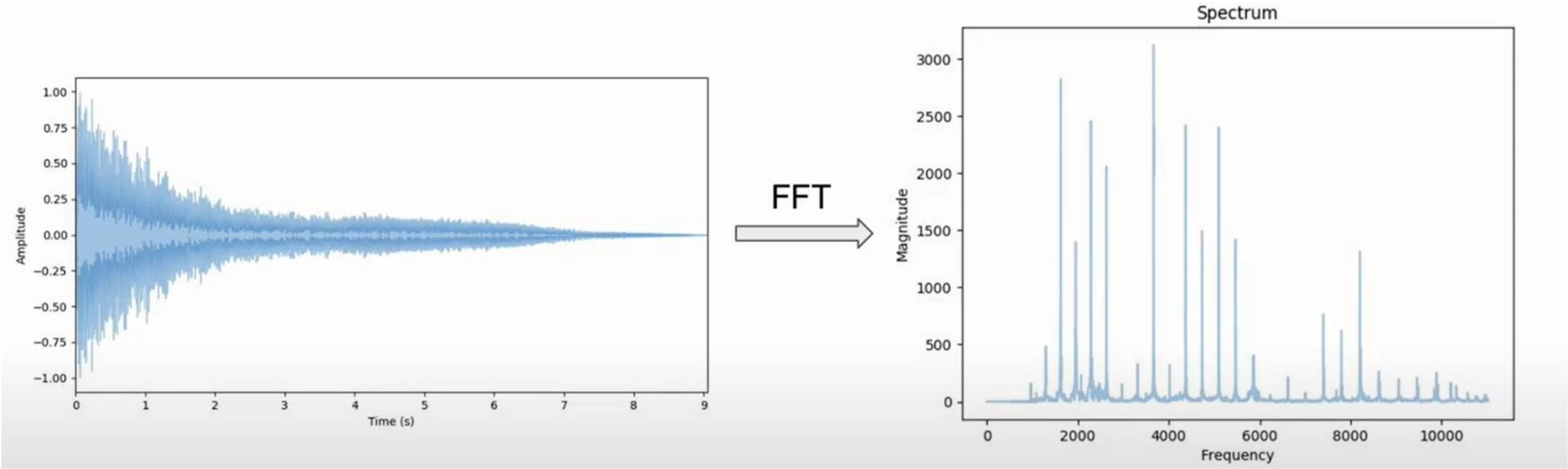


HOW WE CAN ANALYZE COMPOSED SIGNAL?



$$s = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$

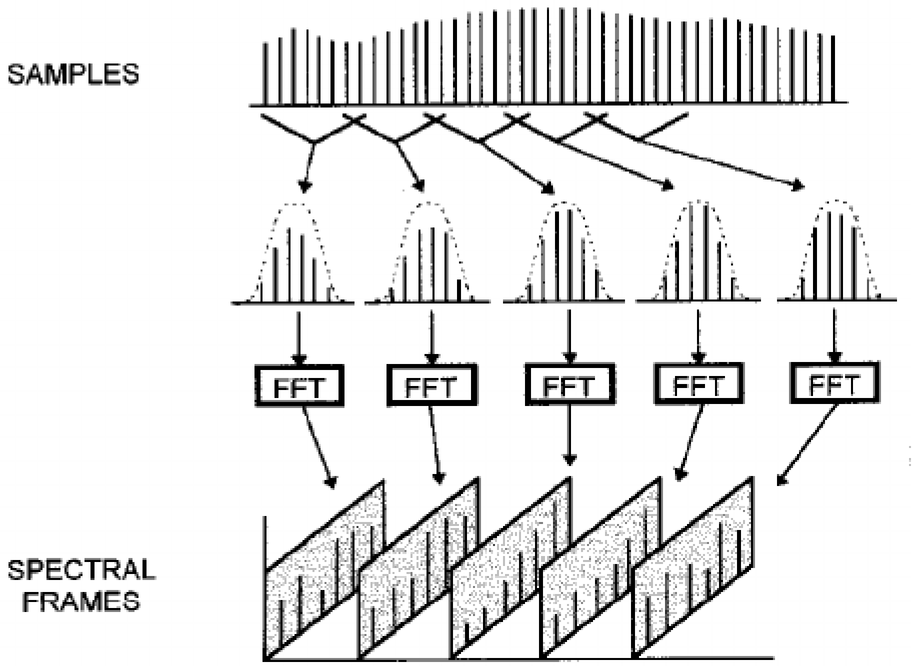
TIME DOMAIN → FREQUENCY DOMAIN

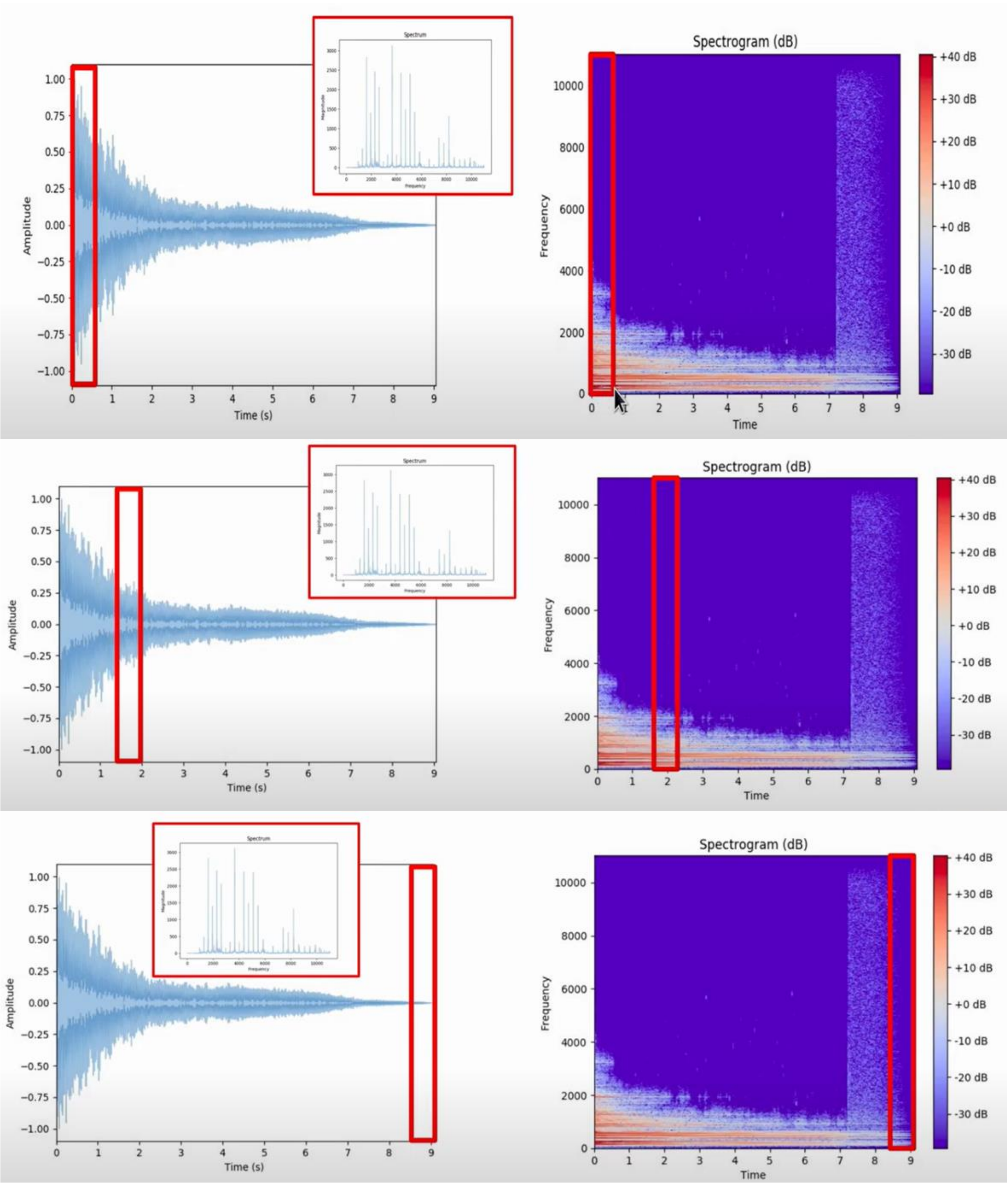


STFT

- Compute several FFT at different intervals

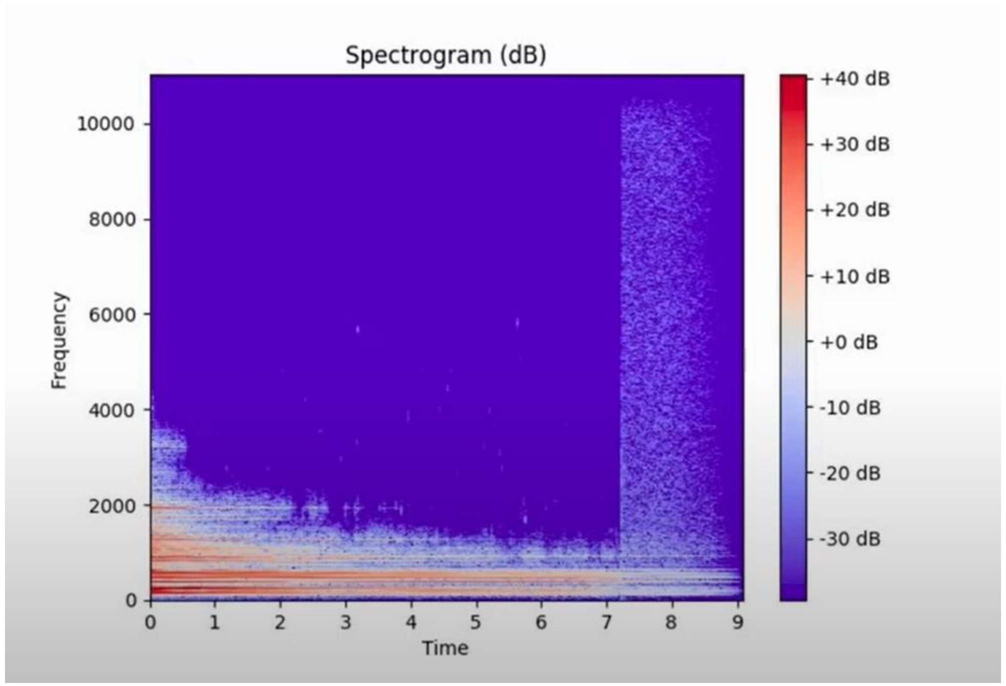
  Preserve time information





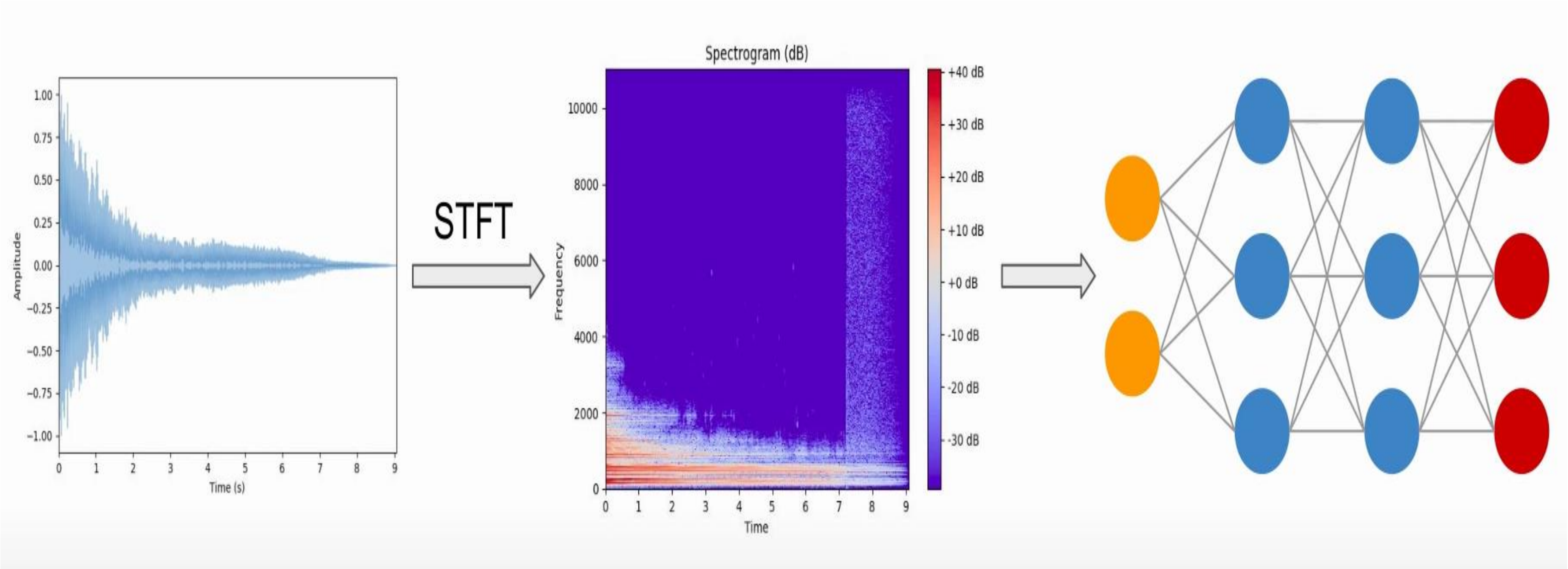
- We shift and move frame left to right
- By applying STFT, we axamine the wave in small frames
- We use spectrogram as a input for deep learning model

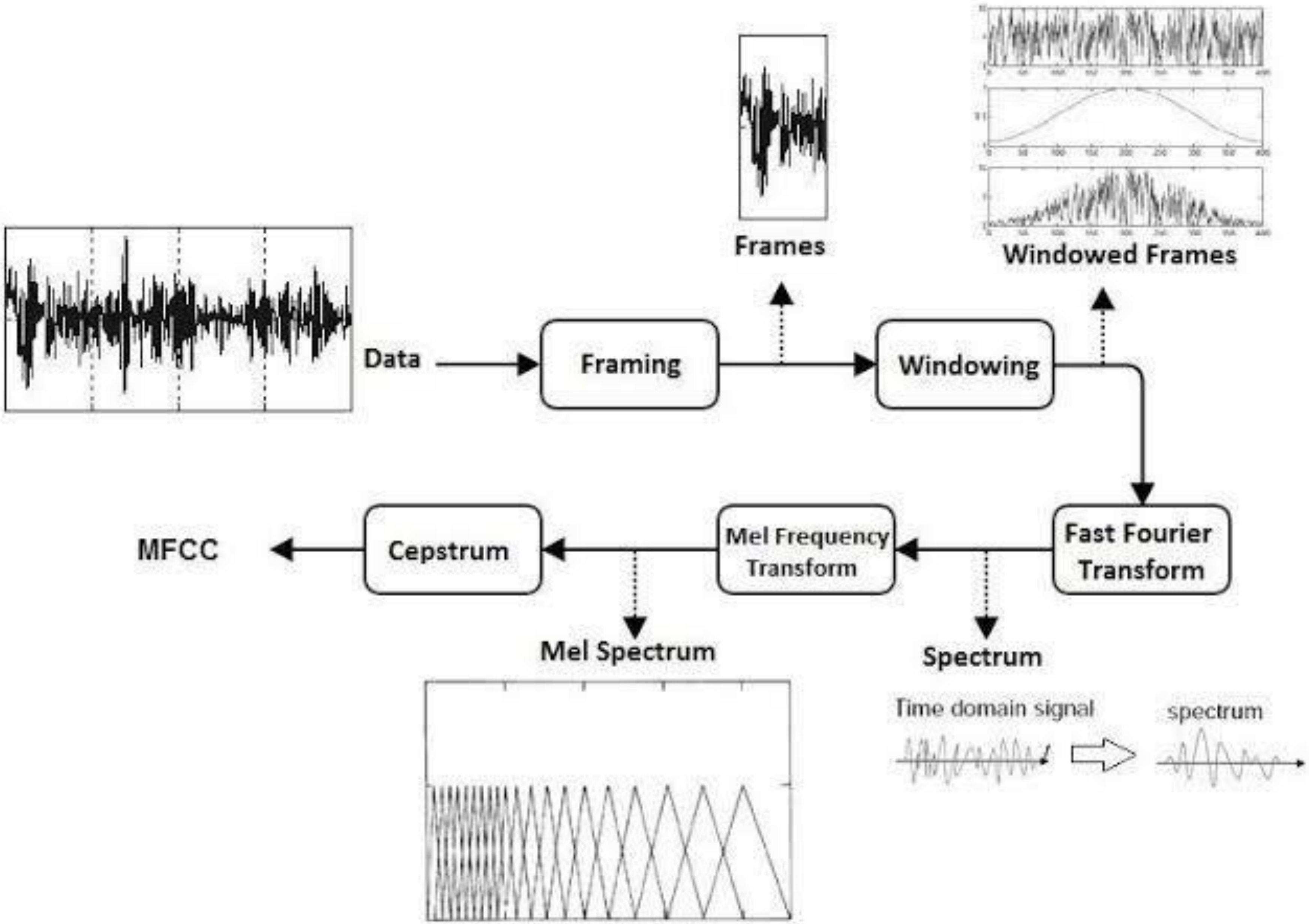
SPECTROGRAM



It includes time, frequency and magnitude(dB) information

INPUT OF DEEP LEARNING NETWORK IS READY !




Frames
Windowed Frames
Data
Framing
Windowing
MFCC
Cepstrum
Mel Frequency Transform
Fast Fourier Transform
Mel Spectrum
Spectrum
Time domain signal
spectrum