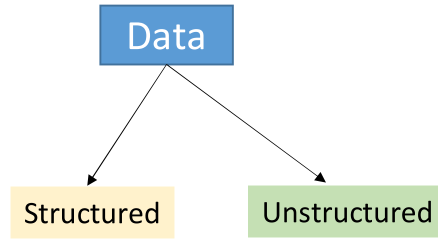
When is Neural Network preffered

1. Unstructured Data
2. Difficult to Feature Engineer
3. High Accuracy Required
4. Large Amount of Data Required



1. Unstructured Data: Deep learning models can learn to recognize complex patterns and structures in data, such as images, speech, and natural language, which can be difficult for traditional machine learning models to capture.

2. Feature engineering is difficult: Deep learning models can automatically learn features from raw data, eliminating the need for manual feature engineering, which can be time-consuming and error-prone.

3. High accuracy is required: Deep learning models have shown to achieve state-of-the-art performance on various tasks, such as image and speech recognition, natural language processing, and recommendation systems.

4. Large amounts of data are available: Deep learning models require a large amount of data to be trained effectively, and they tend to perform better as the size of the training dataset increases.

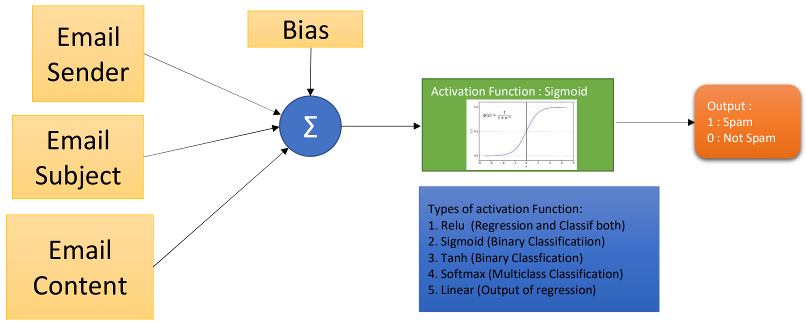
Deep Learning Example - Alexa

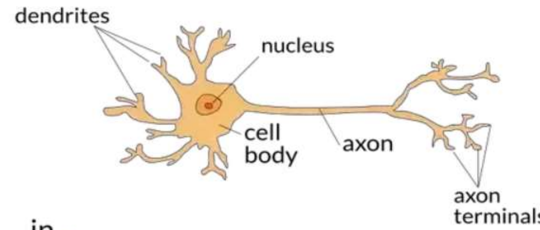
Voice Input to Alexa by User → Conversion of Voice to Text → Identifying the Category of Input → Response → Acknowledge

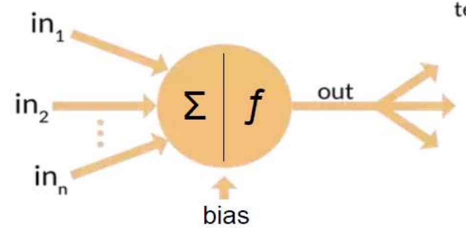
Training Data :
Voice Clip 1 : "Hello"
Voice Clip 2 : "Hello"
Voice Clip 3 : "Hello"
Voice Clip 4 : "Hello"
Voice Clip 5 : "Thank you"
Voice Clip 6 : "Thank You"
Voice Clip 7 : "Thank You"

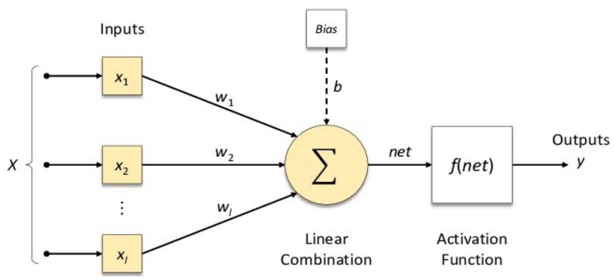
Training Data

Comment	Category
Set a reminder for tomorrow for cancelling my Netflix Subscription 9:00 am	Reminder
Call XYZ person	Call Action
Give me location for Mumbai	Location
Play "Yeh Shyam mastani" song on YouTube	Yotube Music Action
Tell me distance from wakad to Mumbai	Location
Explain to me about Neural Networks	Internet Search
Set Alarm for tomorrow 8:00 am	Alarm

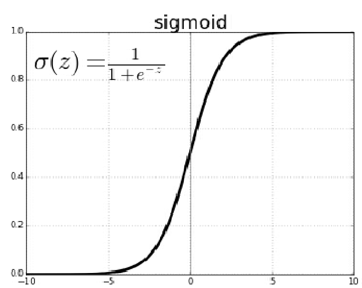


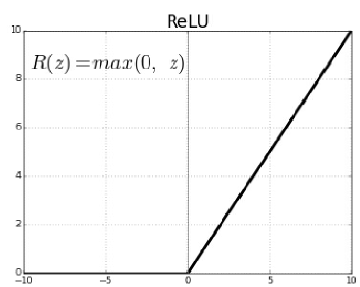


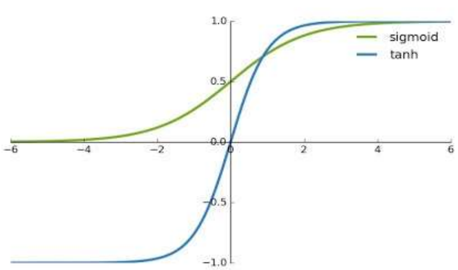


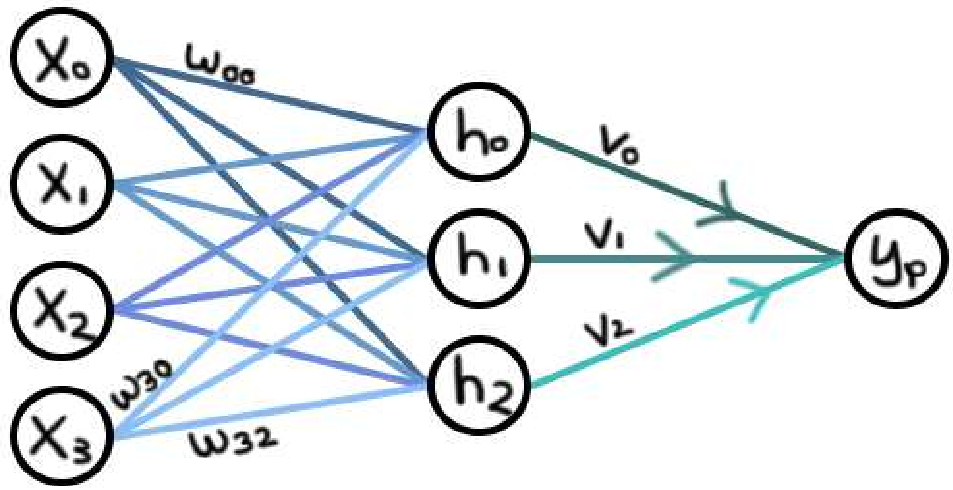


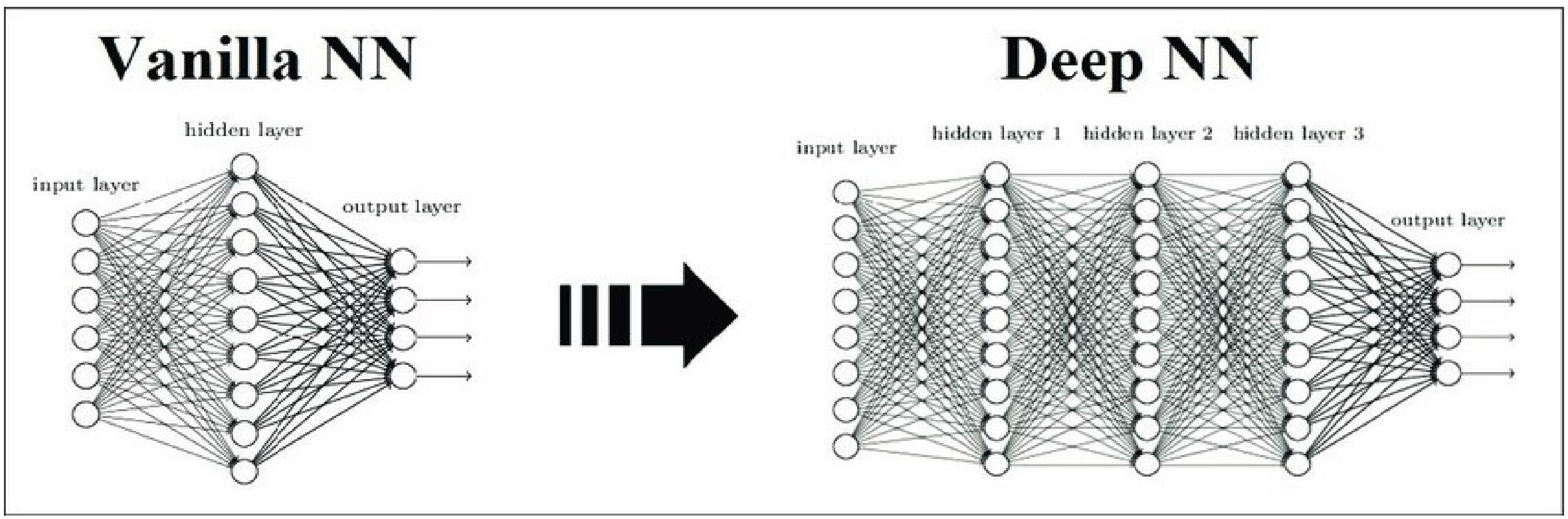
Sum = b + x1*w1 + x2*w2 + x3*w3








X_0
X_1
X_2
X_3
w_{00}
w_{30}
w_{32}
h_0
h_1
h_2
v_0
v_1
v_2
y_p


Vanilla NN
input layer
hidden layer
output layer
Deep NN
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

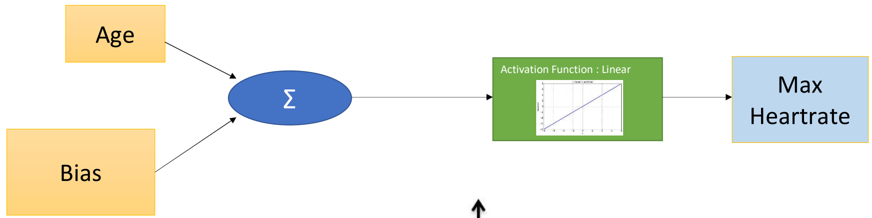
Tensorflow, Keras - Building ANN

Opencv - Image Processing

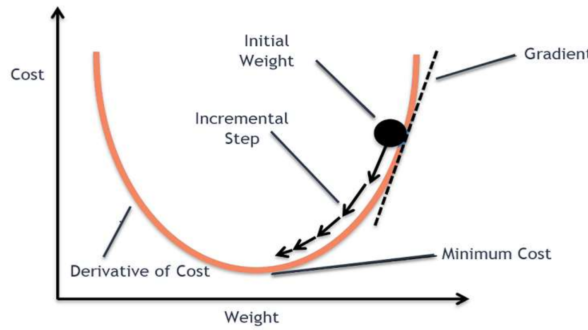
NLTK - Text Processing

1. pip install --upgrade pip
2. pip install --upgrade tensorflow --user
3. pip install keras --user
4. pip install opencv-python --user
5. pip install nltk -- user

Age	max Heartrate
50	184
40	215
26	187
28	197
41	194
25	177
46	162



Max Heartrate = b + Age*w1



Initialize Random Weights and Bias

b	150
w1	1.5

Forward Propogation

Age	max Heartrate	Predicted	Error	Squared Error
50	184	225	-41.00	1681.00
40	215	210	5.00	25.00
26	187	189	-2.00	4.00
28	197	192	5.00	25.00
41	194	211.5	-17.50	306.25
25	177	187.5	-10.50	110.25
46	162	219	-57.00	3249.00

SUM	5400.50
MSE	771.50

Back Propogation

learning Rate 0.0001

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Age	max Heartrate	Predicted	Error	Error*Age
50	184	225	-41.00	-2050.00
40	215	210	5.00	200.00
26	187	189	-2.00	-52.00
28	197	192	5.00	140.00
41	194	211.5	-17.50	-717.50
25	177	187.5	-10.50	-262.50
46	162	219	-57.00	-2622.00

Bias_new = Bias_old - learning_rate*Gradient of Bias

Weights_new = Weights_old -learning_rate*Gradient of Weights

SUM	-118.00	-5364.00
	Gradient for Bias	Gradient for Slope

b_old	150
w1_old	1.5

b_new	149.9882
w1_new	0.9636

Forward Propogation with New weights

Age	max Heartrate	Predicted	Error	Squared Error
50	184	198.17	-14.17	200.74
40	215	210.00	5.00	25.00
26	187	189.00	-2.00	4.00
28	197	192.00	5.00	25.00
41	194	211.50	-17.50	306.25
25	177	187.50	-10.50	110.25
46	162	219.00	-57.00	3249.00

SUM	3920.24
MSE	560.03

MSE Before	MSE After
771.50	**560.03**