Advanced Deep Learning

Trainer: Prashant Nair

Python → Data Science → Machine Learning → Deep Learning → Advanced DL → NLP SR → Reinforcement Learning → Advanced Gen AI (LLMs, RAG (Langchain))

Focus of the Course

1. Computer Vision (OpenCV)
2. Revisiting ConvNets
3. CNN architectures (Style Transfer, Single Shot Detectors, R-CNN, Object Detection)
4. Image Segmentation

→ Computer Vision & CNN

5. Revise Autoencoders
6. Variational Autoencoders
7. Neural Style
8. GANs

→ Regenerative models

9. OCR

→ Regenerative + CV

⑩ Parallel Computation
⑪ Deploying models (MLOps)

} Other aspects

PROJECT

Revision on Deep Learning

@ what do the NN model calculate during the training phase / data convergence / fitting ?

→ Identify and discover OPTIMAL weights.

→ Identify and discover OPTIMAL bias.

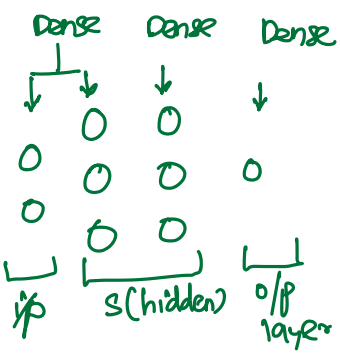
Loss ↓ Metric ↑

④ Layers [Input Layer, Hidden Layers, Output Layer]

tf. keras

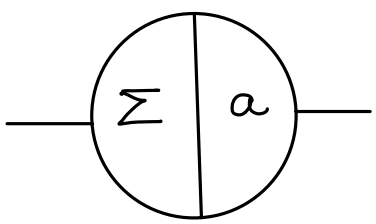
① To create an individual layer ⟶ Dense

② To Link these Dense layers to form a NN ⟶ Sequential -or- Model



③ Neurons

- Which operation does a typical neuron in a perceptron or ANN setup does ?



Summation $\left(\sum_{i=1}^{n} w_i x_i\right) + b$

- List the activation fn for the OUTPUT LAYER in Regression usecase?

tf. keras

↳ linear (pass through)

- List the activation fn for the OUTPUT LAYER in multi-class classification?

tf.keras

↳ Softmax

n_units: n_unique-class label.

multi-class classification
↓
label column will have more than 2 unique values/ class labels.

0s 0.97
0 Ve 0.01
0 Vi 0.02

- List the activation fn for the OUTPUT LAYER in Binary-classification?

tf keras

↳ sigmoid ⟶ n_units = 1

— or —

softmax ⟶ n_units = 2

binary-classification
↓
label column will have 2 unique values.

- List the activation fn recommended for Hidden layers in ANN? (Artificial NN)

tf.keras

- relu
- tanh
- sigmoid
- Leaky Relu
- P Relu
- ⋮

- List the activation fn that may introduce Vanishing Gradient Problem?

Sigmoid
tanh

⟶ Deep NN
↓
number of hidden is increased

Vanishing Gradient Problem

$\Delta w \to \sim 0$

$\Delta b \to$

- How to overcome Vanishing Gradient Problem ?

ⓐ Narrow ANN architecture → Decrease the hidden layer.

ⓑ Use Relu, leaky relu.

DEAD NEURON ?

↓

Leaky Relu

-or-

Parametric Relu

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

x is -ve → 0

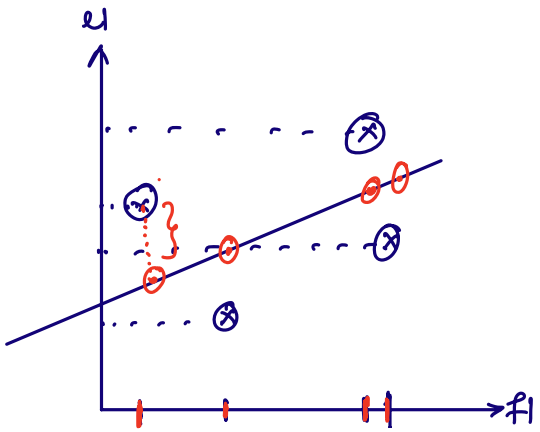
x is +ve → x

ⓐ - Cost Function

✓ Loss Function → single record

✓ Error Function → entire training set -or- either testing -or- entire ds.

y	$\bar{y}$	$y - \bar{y}$
3	2	1
10	8	2
6	1	5
5	4	1
		9



$$MSE = \frac{\sum_{i=1}^{n} (y - \bar{y})^2}{n}$$

Loss fn

error fn

n = no of data pt

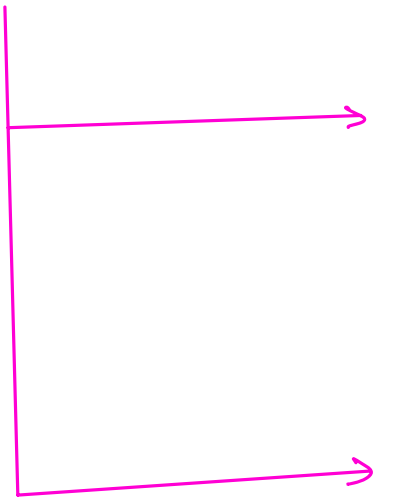
$$MAE = \frac{\sum_{i=1}^{n} |y - \bar{y}|}{n}$$

Regression:

① Mean Squared Error

② Mean Abslute Error

Classification:



→ Binary Classification

binary_crossentropy

→ Multi-class Classifn

el
se
Ve
Ve
se
Vi

encode Label

el
0
1
1
0
2

0	1	2
1	0	0
0	1	0
0	1	0
1	0	0
0	0	1

sparse_categorical_crossentropy

categorical_crossentropy

e. Optimizers in ANN

↳ Optimizers are Back propagation algo implementations that will calc. gradients (Δw & Δb) and update weights and bias. ($w = w + \Delta w$, $b = b + \Delta b$)

1 → a. Adam

b. sgd (stochastic gradient descent)

3 → c. RMSProp

2 → d. Nadam

e. AdaDelta

f. AdaMax

f. Epoch : one full cycle of training.

(Forward Pass + Error calc + Back Prop)

g. Learning Rate: step size to add PENALTY in the gradients

($\Delta w * lr$)

Goal: Find the sweet spot to ensue data is converged successfully.

(achieving optimal minima)

h. How to deal with Overfitting in ANN?

a. change the weights and bias initializers.

a. Glorot Normal

b. Glorot Uniform

c. He Normal

d. He Uniform

(b) Use Regularization.

- (a) L1 regularization
- (b) L2 regularization
- (c) L1L2 regularization
- (d) Dropout.

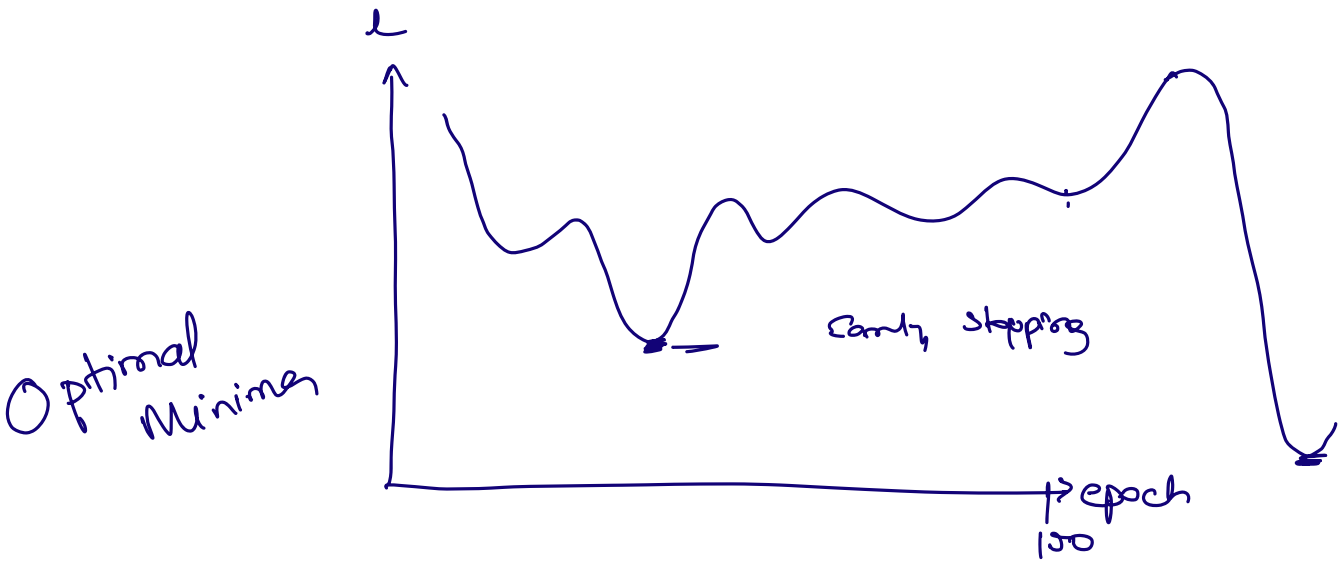
(c) Add momentum in optimizers.

(d) Batch Normalization

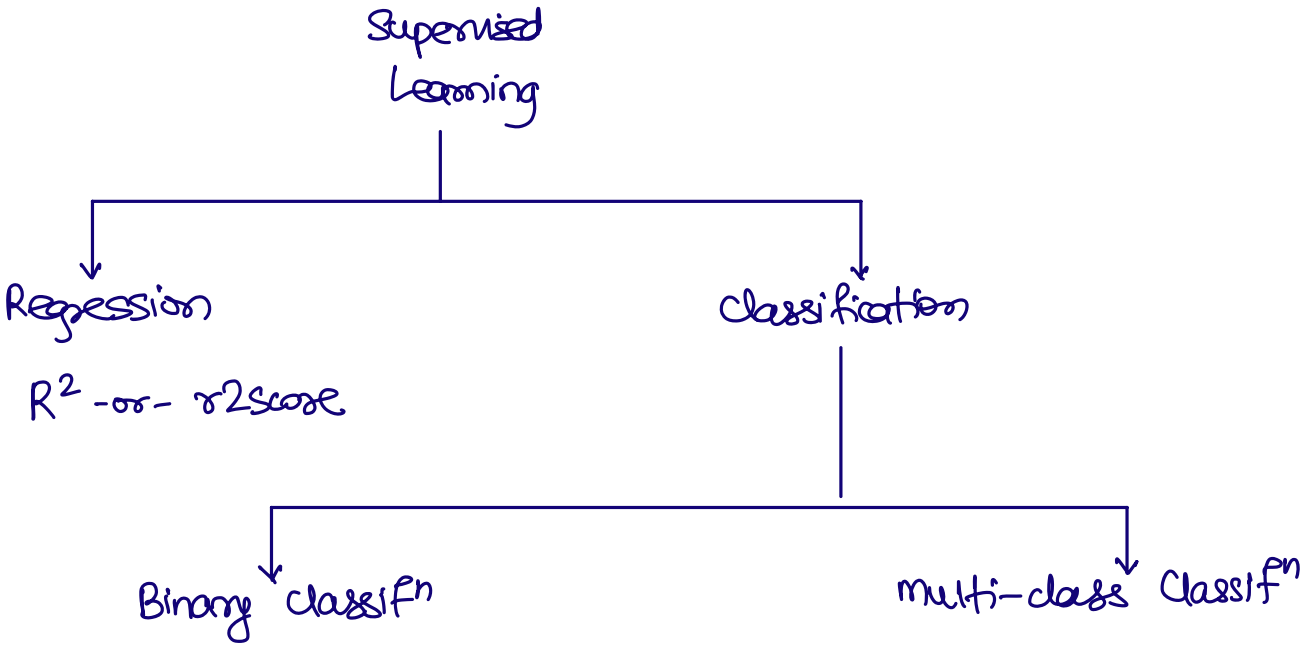
(e) Early Stopping

(f) Try reducing the complexity of Neural Network.

(g) Add / wait for more data ☺



Evaluation Metrics for Supervised Learning



email/spam.csv (100 records)

ham → 50 } Balanced
spam → 50

ham → 40 } Unbalanced / imbalance
spam → 60

- Accuracy → Balanced Dataset
- Precision
- Recall
- F1 Score

} → imbalanced dataset