

# DEEP LEARNING

→ Tensorflow

Asmita  
b

- Use ML or DL
- Length of data
- Type of data

- High level overview
- What is ML & How it works
- Why we do a ML and we can't predict me.
- What lack in ML so come DL
  - Most of Structure
  - Data
  - Feature Engg

ML

- High human intervention
  - feature Engg
- ML Model get trained quickly
- ML Model can be trained on CPU
- ML Model don't perform as good as DL Models on Big data

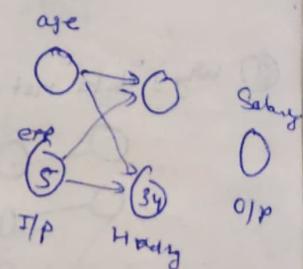
Data like →  
 Image & Text  
 Not so much  
 well in ML  
 So introduce DL

Use neural networks

To do the feature Extracting  
 eg: Human voice

DL

- Subset of ML which basically focus of working like a human brain.
- Deep learning uses Neural Networks (in NLP, CV, Application of DL)
- Automates feature extraction
- Trained on GPU, TPV
- NN have layers
  - data travels from one layer to other
- Processing Unstructured data.



Eg :- Noise dimension

- Lips
- eyebrows
- Cheeks
- RGB

So it is Multidimensional  
 ML Model is hard to extract feature from higher dimension

ML →  
 area - flats  
 location - corner / Rm  
 furnished O/P  
 floor int  
 room int

Structure, data → in data frame  
 and easily apply ML Model

BUT

In hell, i am Shahid → what gone we do → Here DL Model come  
 what DL Do → take input Vocab process in NN  
 then in output like in Hindi

→ In ML Algorithms but DL is a Neural Networks.

→ Artificial Neuron (1<sup>st</sup> create - 1943) (Warren & Weisler)

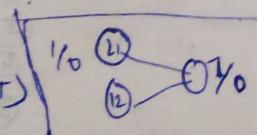
→ It is Mathematical Representation of Biological Neuron

→ Operation on following principles

- ① Binary Inputs & Output

- ② Threshold logic

→ logic (And gate, OR, NOT)

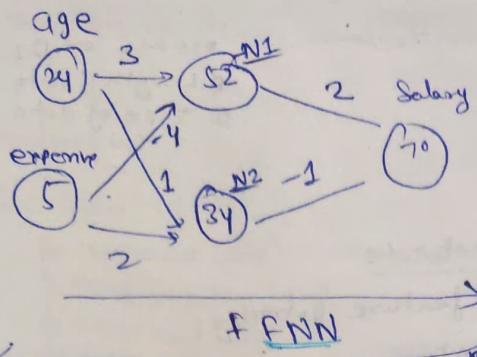


AND  
 l1 = 1  
 l2 = 0  
 O = 0

eg: only for  
 2 binary  
 & 2nd orgy  
 & fruit & veg

- In case of big data  
 NN/DL Perform better than ML Model
- They are good at extracting features from data.

eg:



Backward NN  
Back propagation  
w/o Optimizers

1 Epoch (where 1 FNN)  
1 O/P  
1 Hidd  
1 BNN

$$y_A = 50 \\ y_P = 70$$

① Dot product

$$24 \cdot 3 + 5 \cdot (-4) = N1$$

again for next layer

$$52 \cdot 2 + 34 \cdot (-1) = \text{Output}$$

The weights are automatically decided  
1st Random initialize

e.g. in face check  
happy, sad, neutral  
only lips moment  
Cheese go lips weigh  
high then nose or eye.

↳ This is FEED FORWARD NN

FFNN or MLP

to get input

calculate the hidden

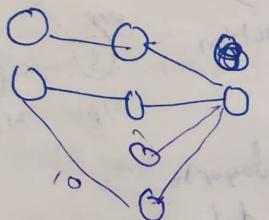
layer value/process

threshold = 5 (minimum) get the output

minimum error we accept

→ feeding the data in forward direction

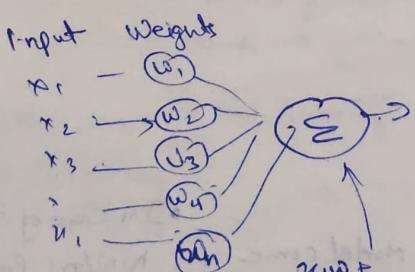
- ⑧ Why 1 data not connect to 2 layers? ⑨ How much layers we define?



↳ Which

# PERCEPTRON: → consists of 1 I Layer  
1 out  
no hidden weight, bias, summation funct, Activation fun, TLV (Threshold)

Simple/Basic ANN Architecture



Activation func

$$\sum \rightarrow [o]$$

threshold

It takes in consideration

the value of threshold & weight sum

• Assign the value to the node

eg: (ReLU, sigmoid)

$$\max(0, 2) = \max(0, 4)$$

$$\text{Out} = 4$$

$\max(0, -2) = \text{output} = 0$ , threshold = 2  
 $\text{TH} > \text{Out} \Rightarrow \text{do Backpropagation}$

$$\max(0, 4) = \text{output} = 4$$

$\text{TH} < \text{Output} \Rightarrow \checkmark$   
return the output

• Whole process called

Perceptron Learning Rule

① Initialize weights

② Compute the weighted sum

③  $\rightarrow z = \text{total (sum)} + b$

④ Apply Activation function

⑤ if Model fails, Update weights

⑥ Otherwise simply return the output

Perception

- ↳ Input layer & 1 output node
- ↳ being Perception, very simple, it can't be used in Real life.

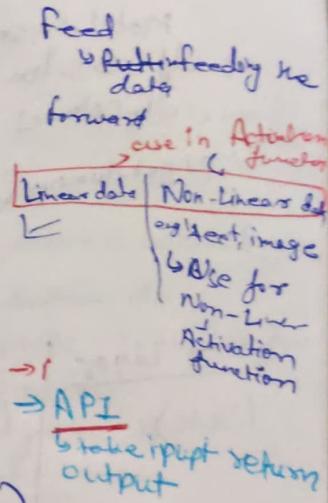
These fine called feed forward net

If Hidden layer the called MLP

In Hidden layer - 2 type

Shallow NN (only 1 HL)

Dense NN (more than 1 HL)



### MLP (Multilayer Perceptron)

- NN build upon perception
- Uses the functionality of Perception, but they hidden layers.
- hidden help us understand complex data.

Eg: High level API

Python - 1, 2, ..., n. (use loop)

or often  $\rightarrow X = \text{Sum}(1, 2)$  (Low level Scratch)

→ keras & Tf integrated  
→ IF → only Python lib

→ KERAS (inbuilt all) to High level API  
↳ can be Integrated

↳ ex. Tf use keras  
↳ from scratch into 14 years

- 2 type of NN

① Shallow NN

- 1 Hidden layer

② Dense NN

→ More than 1 HL

### TensorFlow: Use Build layers in optimize calculation

like

import tensorflow as tf  
from tensorflow import keras

for i in range(1, 10):  
 keras import Sequential

model = Sequential()

↳ Pass the layer from 1 to another  
↳ like Construct

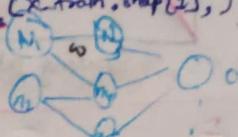
\*-train.shape[0]

Create Shallow NN

Input → Model.add(tf.keras.layers.Dense(1))  
Output → Model.add(tf.keras.layers.Dense(1))

(x\_train.shape[0], ) activation = 'relu')

Shape



So Define in function to define  
fully Connected layers

here these are not  
fully Connected layer  
Why (every node connect  
to all node)

Dense →

- ↳ no of node
- ↳ input shape
- ↳ activation function
- ↳ decides what values

it is hit & trial thing to do)

Dense(32,

M

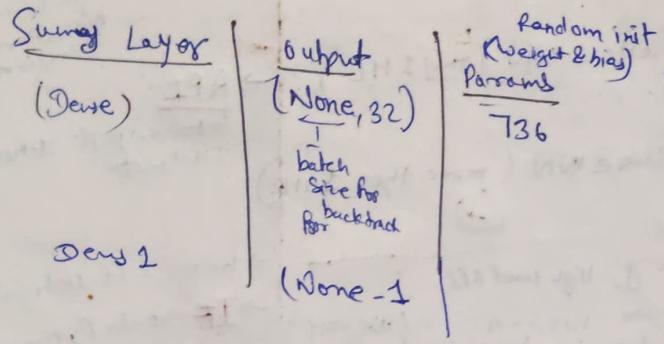
datatype in TF  
is tensor

Skele~~is~~)

If the value is b/w 1 to 1, we don't have any outliers

np. Whales

apply condition tensor  
 $\cdot (\text{df}[\text{'app'}] = \text{np.where}(\text{df}[\text{'not'}] > 200, 200, \text{where}(\text{condition}, \text{Tz}, \text{fx}))$



feature = 22 22  
 node = 32 ] 22x32  
 704 (Weights)  
 + also  
 32 bias add  
 $\Rightarrow$  36

If 2 feature  $N=3$   $\rightarrow$  bias + non lin

```

graph TD
    N1(( )) --> N2(( ))
    N1(( )) --> N3(( ))
    N2(( )) --> N3(( ))
    
```

A diagram illustrating the relationship between three concepts:

- Class Reg** is positioned on the left.
- ML** is positioned at the top center.
- Perimetric Non-param** is positioned on the right.
- An arrow points from **Class Reg** to **ML**.
- A bracket connects **ML** and **Perimetric Non-param**, indicating they are related.
- The word **eq** is written above the bracket, suggesting an equivalence or relationship between the two terms.

What is DL  $\rightarrow$  Parametric model

↳ outcome  $y \sim f(x)$

equation we use in DL

What requirement of DL?

① Where we place DL

~~It is necessary to learn ML before DL~~

If we do in ML then why do DL?

1

but ML ① architecture  $\rightarrow$  Video, Audio, text  
comes in structured data  $\rightarrow$  CSV, JSON

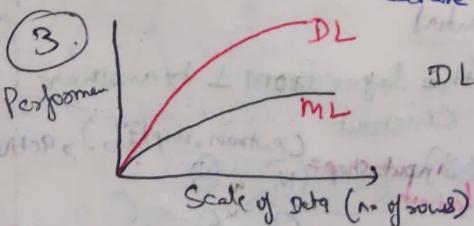
② DL is making

e.g.: doctor's check

for checkup +

→ J → Link audio + Audio  
for sellers  
↳ Multi type data page  
feature extraction

DL → automated feature engineering



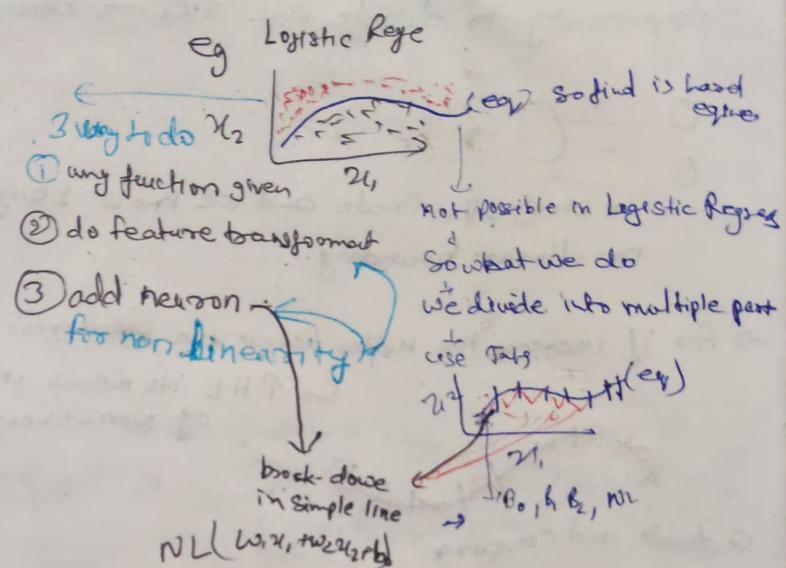
(8) What

3 Aug 82

playground.tensorflow.org

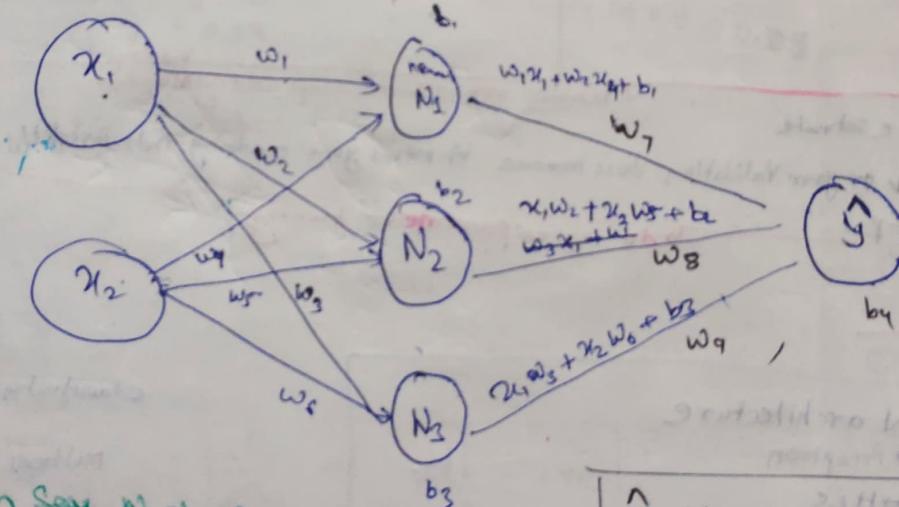
### ③ What is a neuron?

Dr. 40  
Visual  
of Neuron



more local decision boundary → and combine to create  
Complex decision boundary create

1 HL features



So we can say  $N_1, N_2, N_3$  are new features  
Create from  $x_1, x_2 \rightarrow$  which represent data space b/w  
which are closer to the boundary shape

$$N_1 = w_1 x_1 + w_2 x_2 + b_1, \quad N_2 = w_3 x_1 + w_4 x_2 + b_2, \quad N_3 = w_5 x_1 + w_6 x_2 + b_3$$

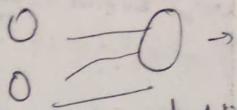
$$\hat{y} = (w_7 (w_1 x_1 + w_2 x_2 + b_1) + w_8 (w_3 x_1 + w_4 x_2 + b_2) + w_9 (w_5 x_1 + w_6 x_2 + b_3))$$

$$\hat{y} = Ax_1 + Bx_2 + C$$

Note: If we not add any Non-linearity in this it always be in linear simple linear fit.

If you not any hidden layer then we cannot divide the boundary into multiple boundary part.

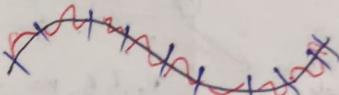
If you don't have hidden layer then you have no opportunity to divide the decision boundary into subsection.



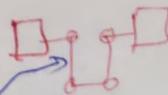
so only 1 line create and we know 1 single line not define non-linear boundary.

Strength

→ So if increase the Hidden layer the Subincrease the Sub Parting  
So ↑ H.L it's means increase the degree of Non-linearity



So divide and conquer



If I give you 10000 straight line but you can't create this path with Mold go what we do need mold

(↓ Activation function for Mold)

Q&A

Byes  
gradient descent

① loss not improve saturate

② Training loss is ↓ or your validation loss increase It means you entering into Overfitting

So you stop

I know where to stop adding layers

hidden  
to do hyperparameter tuning

Classification Sigmoid (Range 0-1)  
multiclasses (Softmax?)

For tanh → -1 to 1

In Regression Problem  
The range is -∞, ∞  
So we not need any activation functions

Aug 4 DL

- Define NN architecture
  - Perceptron
  - HLS
  - Neurons
  - activation functions
  - Weights and biases
  - optimizer and loss functions

e.g.: to how NN work

$$\begin{pmatrix} \text{Input} \\ \text{---} \\ -1 & 0 & 0 & 0 \end{pmatrix}$$

Work on probability distribution

Softmax ( $\gamma$ ) convert into probability

4 categories: Gold, Silver, Bronze, Normal

O - Gold

0.7

O - Silver

0.1

O - Bronze

0.105

O - No

-0.25

Softmax  $\xrightarrow{\text{to 0 to 1}}$  change Scalars/Numbers into Probability  $\frac{e^{q_1}}{e^{q_1} + e^{q_2} + e^{q_3}}$   
[redcarb-software.com/en/calculus/softmax](http://redcarb-software.com/en/calculus/softmax)

$$\begin{matrix} q_1 & = 1 \\ q_2 & = 3 \\ q_3 & = 2 \end{matrix}$$

$$\frac{e^1}{e^1 + e^3 + e^2}$$

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & & \end{matrix}$$

- Q) ② The output layers of neurons in multiclass classification is  
 Softmax (Change Value in Probability Value)  
 agree  
 to extract the Non-linear

Probability  
 0.1 Suny  
 0.2 Windy  
 0.7 Cloudy  
 Why need Probability Value

Use log loss = for all rows of data sum

$$-(y_1 \log(\hat{y}_1) + y_2 \log(\hat{y}_2) + y_3 \log(\hat{y}_3) + \dots + y_n \log(\hat{y}_n))$$

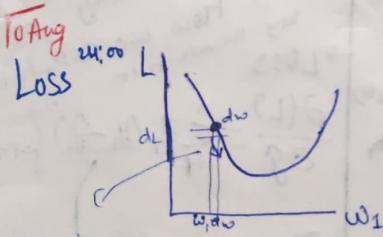
$$= -y \log 0.2$$

$$= 0.69$$

So you tell Predict toward 1  
 $y$   
 $\hat{y}$   
 $y_i$   
 $\hat{y}_i$   
 $\ln$   
 $\exp \log 0.8 \Rightarrow 0.89$

the loss function  
 Continuous & differentiable  
 is log loss function

1:27



move to opposite of the gradient  
 So it go to the minima side  $\rightarrow d$

$$\frac{dL}{dw} = -ve, \frac{dL}{dw} = +ve$$

$$w_1(\text{new}) = w_1(\text{old}) - \left( \frac{dL}{dw} \right)$$

$\frac{dL}{dw}$  depends of slope  
 if slope decline  
 -ve or incline then +ve

To minimize loss  $\boxed{\text{Forward Propagation}}$   $\geq \text{Loss}$

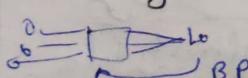
When calculate the loss called forward propagation (FP)

$$\text{Loss} = f(y_{\text{Pred}}, y_{\text{actual}})$$

$$L = f((w, b, x), y_{\text{actual}})$$

$$L = f(w, b)$$

To improve or change LOSS by changing w, b called Backward Propagation (BP)



why choose DL (Parametric)  
 ① X → Any type of data  
 (Video, audio, text, etc.)  
 Multimodality

② Take  $\frac{dL}{dw}$  → Diagnosis  
 test scan

③ Performance  
 Acc  
 ML

④ DL → automated  
 feature engineering

e.g.  
 → Merchandised  
 Payment  
 defaulter  
 癌 cell detection  
 Supply chain optimisation

→ Activation

00:27

Lecture 3

(chain → if i change  $v_1 \leftrightarrow 1$  unit where all the effects are getting changed)

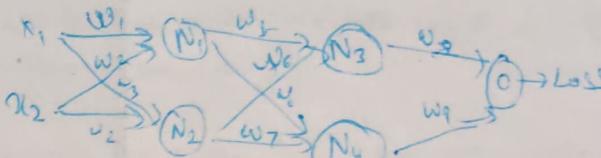
Q

Activation function

Q

1:15

Ex. Chain



$$\left( \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial N_3} \times \frac{\partial N_3}{\partial N_1} \times \frac{\partial N_1}{\partial w_3} \right) + \left( \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial N_4} \times \frac{\partial N_4}{\partial N_1} \times \frac{\partial N_1}{\partial w_3} \right)$$

Terms =  $N(HL) + 2$  terms in Backpropagation  
(multiple)

↳ May be Vanishing gradient)

ex  $(0.00)^{100} \approx 1.00$   
 $(1.00)^{100}$

If numbers are small in hidden layer & multiply then these numbers more & more small & when based on this calculate loss it is also very very small  
→ So what happen → gradient is not happen.

↳ (Vanishing gradient)  
if multiple is big Number → loss is high → so gradient is very large → so it over-suit with min loss

↳ Exploding gradient  
Problem  $y_i = w_1N_1 + w_2N_2 + w_3N_3 + b_i$

Why we see this because which Activation function we use in these Problem to overcome.

Q

Ex:

$$y = x^2$$

$$m = y^2$$

$$z = m^2$$

differentiation

$$\frac{\partial z}{\partial m} = 2m$$

$$\frac{\partial z}{\partial x} = ?$$

hit direct connected  
so we do change y v.r. x  
so indirectly solve by  
Partial derivative  
here easy to do P.D.  
because here simple eq  
but  $e^x \sin(x)$

$\frac{\partial z}{\partial x}$  how we do now

here come Chain Rule  
How solve

$$\frac{\partial z}{\partial x}$$

$$\frac{\partial z}{\partial m} \times \frac{\partial m}{\partial y} \times \frac{\partial y}{\partial x}$$

what purpose to teach,  
yes because of loss

$$\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial w_3}$$

then How we calculate loss  
by chain rule

$$\frac{\partial L}{\partial g} = \sum_{i=1}^n \frac{1}{n} (y_i - g_i)^2$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial w_1}$$

$$\sum_{i=1}^n \frac{1}{n} \times 2(y_i - g_i) \times 1 \times w_1$$

Loss Works

$$\frac{dL}{dg} = ?$$

$$\frac{\partial L}{\partial w_1} = ?$$

Using chain rule  
1.10

AF → should be continuous & differentiable

Sigmoid  
is binary  
Sigmoid  
is multi  
Class

Activation functions are functions which should be continuous & differentiable that can be applied on the output of my hidden layer to create non-linear feature or at the output layer to create bounded output.  
(like probability)

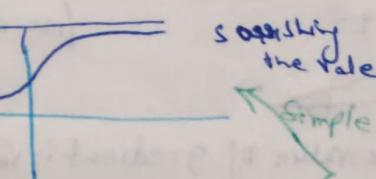
①

→ Sigmoid:

$$i/p = -2 \leq x \leq 2$$

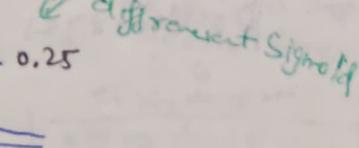
$$o/p = 0 \leq a \leq 1$$

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



Sigmoid  
the rate

Simple Sigmoid



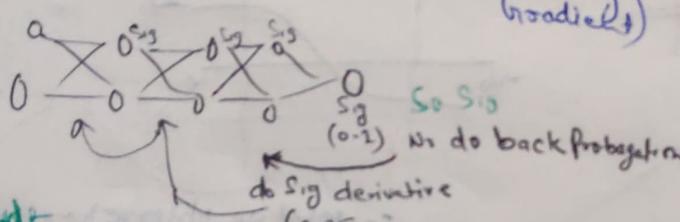
0.25

different Sigmoid

Use: So first derivative of Sigmoid is  $(0-0.25)e^{-x}$   
So its gradient is low

What is problem

(Problem is Vanishing  
gradient)



problematic

RNN

Sigmoid  
(Good for Last  
Output  
but not good for  
hidden layers)

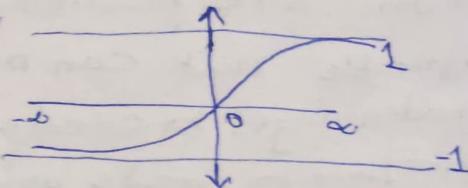
Tanh

## ② Tanh:

I/P = -∞ to +∞

O/P = -1 to 1

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



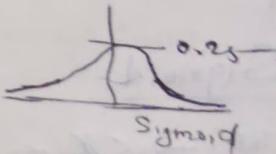
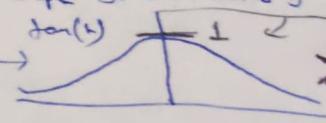
→ More Variance from Sigmoid

→ tanh → more use in Hidden layers

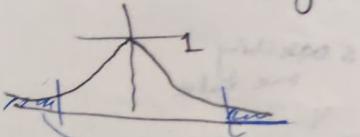
→ It Solve Vanishing Problem or not?

→ Both has S-shape So derivative of tanh is also gradient

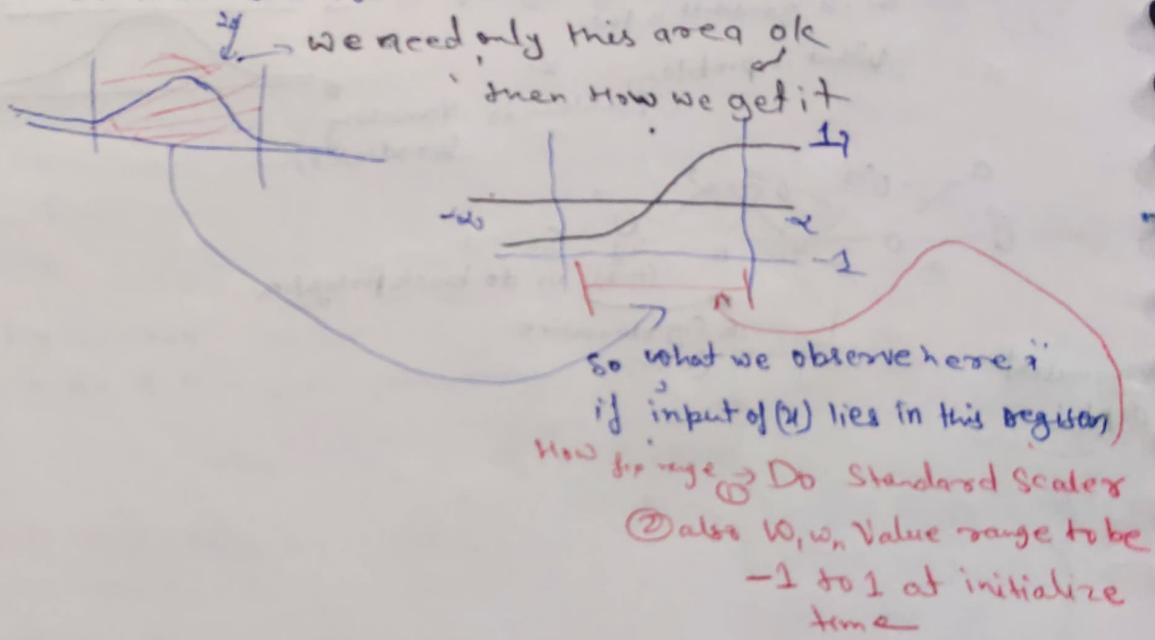
/ more space to fit value



Sigmoid

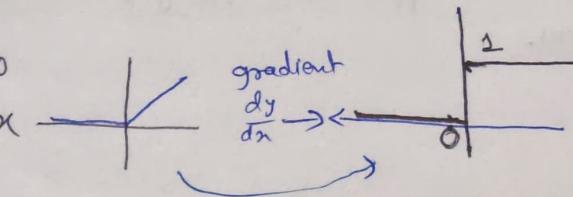


If our value of gradient is very low then it also shows Vanishing Problem? Yes → So What we do



### ③ Relu:

$$\text{Relu}(x) = \begin{cases} x & x \leq 0 \\ 0 & x > 0 \end{cases}$$

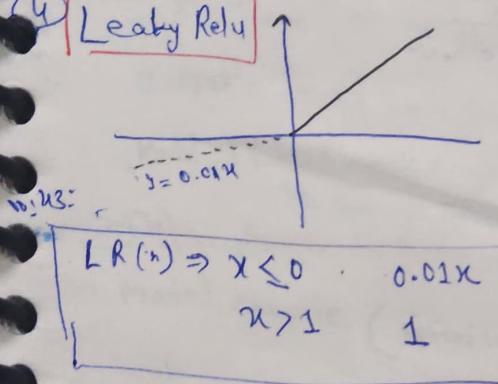


↳ so when we have  
bigger chain that time  
relu doesn't fail

dropout is good but in ReLU Dropout is ~~not~~ intentionally.  
So It means we cannot control it.

↓  
then How Remove it (or control it)  
Using Leaky Relu

### ④ Leaky Relu

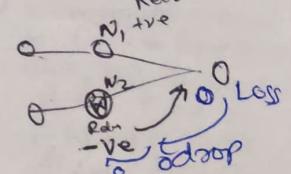


$$\text{LR}(x) = \begin{cases} 0.01x & x \leq 0 \\ 1 & x > 1 \end{cases}$$

↳ How is this a non linear function  
↓  
at least 2 slope  
at any point in the x space

and

↳ not Vanishing  
gradient here  
because +ve is  
disAdv 1  
↳ -ve (drop)



↳ so when do backprop  
the leaky NN is drop  
because value  
is 0

↳ So in deep NN  
the drop is good

(but not for all)  
↳ (Shallow NN)

↳ changes the batch

example

↳ Student  
Chap 1 Chap 2

↳ Exam  
# Exam Exam  
# Exam Exam

### ⑤ Parametric Relu:

$$y = x \quad x > 0$$

$$y = \alpha x \quad x \leq 0$$

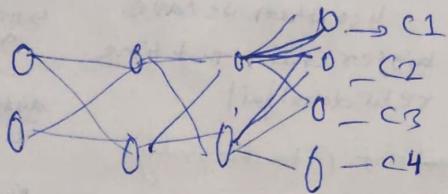
$$0 < \alpha < 1$$

if  $\alpha = 0.01 \rightarrow$  called Leaky Relu

(3) Softmax:

- Need to be applied the O/L of a multiclass Problem.

→ Softmax is the extension of the Sigmoid for multiple Class.



$$z_1 + z_2 + z_3 = 1$$

$$\text{Flow } P_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$P_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$P_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

Softmax

: (1) -  $P_1$  only  
(2) -  $P$

$$P_1 = f(P)$$
$$P_2 = \frac{f(P)}{1 - P_1}$$

## Code Apply

compose → column  
transforming

Code

→ to\_categorical

OHE.factor(y.reshape(-1, 2))

→ Input layers

→ functional API

↳ inputs

but in Segmentation not need  
to define in Input layer  
because वही ही Segmentation  
पर होता है लेकिन लेकिन  
F API की ओर  
Input layer define करना है

Define functional API

input(Shape=(Xtrain.shape[1],))  
X = Dense(64, activation='relu')(input)  
Applied in input layer  
here this layer look like a function  
जो इसे top में (input) को Apply करते  
होते हैं तो we are free to apply any input layers,  
X = Dense(32, activation='relu')(X)

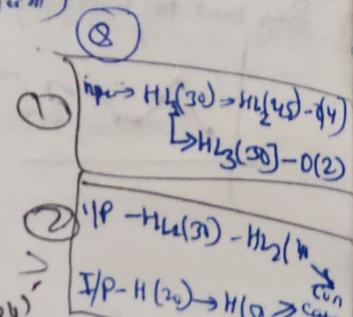
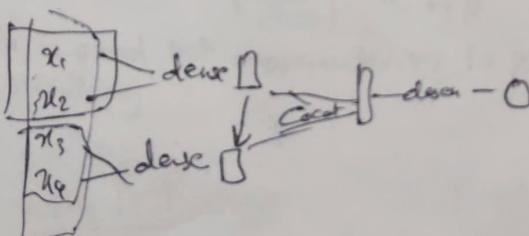
Output = Dense(Xtrain.shape[1], activation='softmax')  
Apply to output

Model = Model(inputs=input, outputs=Output)

→ Define Architecture

model.compile(optimizer='adam', loss='categorical\_crossentropy',  
metrics=[accuracy])

Also do  
like



Concat(H1, H2, H3)

• [ ] - [ ] - [ ]  
10 5 4 → page (Temp)  
Input(shape=(10,))  
X = Dense(5, activation='relu')(input)

X = Dense(4, activation='relu')(X)

O1 = Dense(3, activation='softmax')(X)

O2 = Dense(2, activation='softmax')(X)

model = Model(inputs=input, outputs=[O1, O2])

Model.compile(loss=[

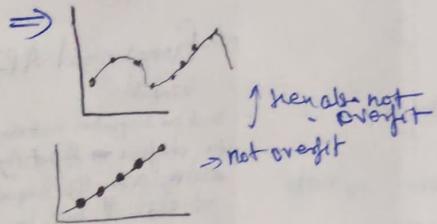
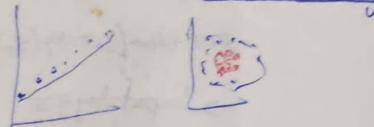
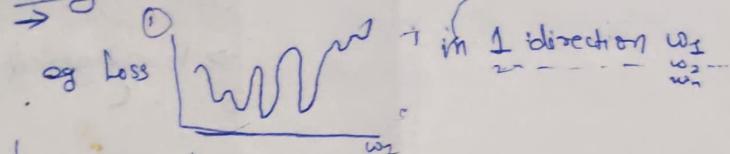
• O1: 'mse', O2: 'categorical\_crossentropy'

optimizers='adam')

metrics=[O1: 'acc', O2: 'accuracy'])

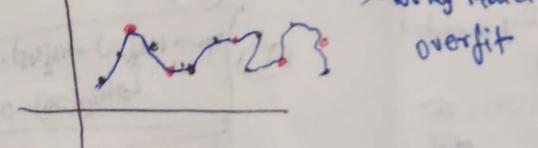
④ Any fan Model Accuracy impact by Segemental &  
functional api use?  
Any NO

18 Aug 2024



$\sin(\omega)$   
AFT

If data is Noise then we can  
Say that the Overfitting



- Not fit
- Proper
- Local Min
- Batch size
- Regularizer
- Batch Norm
- Weight init
-

# Optimizers

- ① Regularization (ridge/l2)
- ② Dropout (Drops of Nth will be 0) for certain iteration
- ③ What is Optimizer
- ... " gradient descent

SGD

Momentum

④ Types

Callbacks

⑤ Batch Normalization

⑥ Weight Initialization

2

"It is enough Momentum"

b don't HP

Hyperparameter

→ It takes time but get global minima

but <

but if shape of Error

W

here the Minima stuck because shape is same

historical gradient

① Momentum Solve the local minima

② Adagrad

= Why create Adagrad? What problem is solve?

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{v_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t}$$

$$v_t = v_t - \frac{\alpha}{\sqrt{v_t + \epsilon}} \left( \frac{\partial L}{\partial w_t} \right)^2$$

→ India is good but implementation is good.  
no decay

so introduce RMS Prop

add decay

HP →  $\alpha$

$v_t$  works

a Hyperparameter

→ Here

momentum is missing

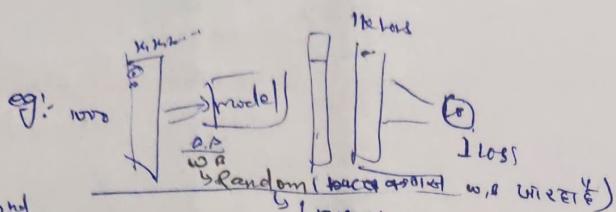
④ ADAM → Some thing come with ADAM  
add momentum

(Adaptive Moments)

add moment

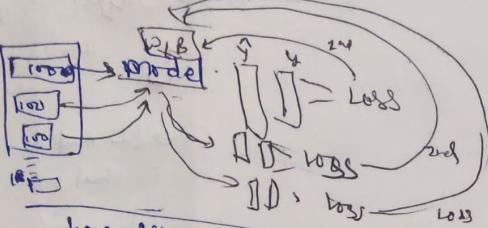
Cell back is  
not function  
↓ it is tracking  
function

→ In time series it  
work  
by yes worse but  
different way to  
use there.



2nd example

case



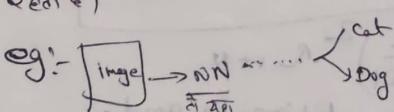
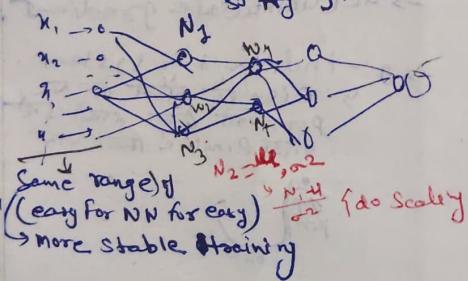
→ will be combine  $w, b$ ? → No just improve here the same  $w, b$

3rd Aug

## ① Batch Normalization :

so Ag guarantee of  $N_1$  Normalize होगा  $\rightarrow N_2, N_3$  तो अंदर  $N_4, N_5$  की नहीं होगा तो यह

इस जू नormalize करने में हुआ वे इसी जू ही HL से नहीं  
विद्युत है।



BC  
BD  
CG  
CD

to make sure this happens decision  
Boundary हो जाएगा।

called Covariate shift

→ it is waste of time

remove means  
to get in  
Centers of all data

$\rightarrow$  to  $N_1 \rightarrow N_2 \rightarrow \text{Stabilize}$

faster  $N_1$

|        | Original           | Scalr |
|--------|--------------------|-------|
| Senior | 55, 45, 25, 15, 10 | 2     |
| Adult  | 30, 25, 20, 15, 10 | 2     |
| Child  | 5, 4, 3, 2, 1      | 2     |

Mean & Varience  
identities  
etc 1 log  
etc 2 log  
etc 3 log  
etc 4 log  
etc 5 log  
etc 6 log  
etc 7 log  
etc 8 log  
etc 9 log  
etc 10 log  
etc 11 log  
etc 12 log  
etc 13 log  
etc 14 log  
etc 15 log  
etc 16 log  
etc 17 log  
etc 18 log  
etc 19 log  
etc 20 log  
etc 21 log  
etc 22 log  
etc 23 log  
etc 24 log  
etc 25 log  
etc 26 log  
etc 27 log  
etc 28 log  
etc 29 log  
etc 30 log  
etc 31 log  
etc 32 log  
etc 33 log  
etc 34 log  
etc 35 log  
etc 36 log  
etc 37 log  
etc 38 log  
etc 39 log  
etc 40 log  
etc 41 log  
etc 42 log  
etc 43 log  
etc 44 log  
etc 45 log  
etc 46 log  
etc 47 log  
etc 48 log  
etc 49 log  
etc 50 log  
etc 51 log  
etc 52 log  
etc 53 log  
etc 54 log  
etc 55 log  
etc 56 log  
etc 57 log  
etc 58 log  
etc 59 log  
etc 60 log  
etc 61 log  
etc 62 log  
etc 63 log  
etc 64 log  
etc 65 log  
etc 66 log  
etc 67 log  
etc 68 log  
etc 69 log  
etc 70 log  
etc 71 log  
etc 72 log  
etc 73 log  
etc 74 log  
etc 75 log  
etc 76 log  
etc 77 log  
etc 78 log  
etc 79 log  
etc 80 log  
etc 81 log  
etc 82 log  
etc 83 log  
etc 84 log  
etc 85 log  
etc 86 log  
etc 87 log  
etc 88 log  
etc 89 log  
etc 90 log  
etc 91 log  
etc 92 log  
etc 93 log  
etc 94 log  
etc 95 log  
etc 96 log  
etc 97 log  
etc 98 log  
etc 99 log  
etc 100 log  
etc 101 log  
etc 102 log  
etc 103 log  
etc 104 log  
etc 105 log  
etc 106 log  
etc 107 log  
etc 108 log  
etc 109 log  
etc 110 log  
etc 111 log  
etc 112 log  
etc 113 log  
etc 114 log  
etc 115 log  
etc 116 log  
etc 117 log  
etc 118 log  
etc 119 log  
etc 120 log  
etc 121 log  
etc 122 log  
etc 123 log  
etc 124 log  
etc 125 log  
etc 126 log  
etc 127 log  
etc 128 log  
etc 129 log  
etc 130 log  
etc 131 log  
etc 132 log  
etc 133 log  
etc 134 log  
etc 135 log  
etc 136 log  
etc 137 log  
etc 138 log  
etc 139 log  
etc 140 log  
etc 141 log  
etc 142 log  
etc 143 log  
etc 144 log  
etc 145 log  
etc 146 log  
etc 147 log  
etc 148 log  
etc 149 log  
etc 150 log  
etc 151 log  
etc 152 log  
etc 153 log  
etc 154 log  
etc 155 log  
etc 156 log  
etc 157 log  
etc 158 log  
etc 159 log  
etc 160 log  
etc 161 log  
etc 162 log  
etc 163 log  
etc 164 log  
etc 165 log  
etc 166 log  
etc 167 log  
etc 168 log  
etc 169 log  
etc 170 log  
etc 171 log  
etc 172 log  
etc 173 log  
etc 174 log  
etc 175 log  
etc 176 log  
etc 177 log  
etc 178 log  
etc 179 log  
etc 180 log  
etc 181 log  
etc 182 log  
etc 183 log  
etc 184 log  
etc 185 log  
etc 186 log  
etc 187 log  
etc 188 log  
etc 189 log  
etc 190 log  
etc 191 log  
etc 192 log  
etc 193 log  
etc 194 log  
etc 195 log  
etc 196 log  
etc 197 log  
etc 198 log  
etc 199 log  
etc 200 log  
etc 201 log  
etc 202 log  
etc 203 log  
etc 204 log  
etc 205 log  
etc 206 log  
etc 207 log  
etc 208 log  
etc 209 log  
etc 210 log  
etc 211 log  
etc 212 log  
etc 213 log  
etc 214 log  
etc 215 log  
etc 216 log  
etc 217 log  
etc 218 log  
etc 219 log  
etc 220 log  
etc 221 log  
etc 222 log  
etc 223 log  
etc 224 log  
etc 225 log  
etc 226 log  
etc 227 log  
etc 228 log  
etc 229 log  
etc 230 log  
etc 231 log  
etc 232 log  
etc 233 log  
etc 234 log  
etc 235 log  
etc 236 log  
etc 237 log  
etc 238 log  
etc 239 log  
etc 240 log  
etc 241 log  
etc 242 log  
etc 243 log  
etc 244 log  
etc 245 log  
etc 246 log  
etc 247 log  
etc 248 log  
etc 249 log  
etc 250 log  
etc 251 log  
etc 252 log  
etc 253 log  
etc 254 log  
etc 255 log  
etc 256 log  
etc 257 log  
etc 258 log  
etc 259 log  
etc 260 log  
etc 261 log  
etc 262 log  
etc 263 log  
etc 264 log  
etc 265 log  
etc 266 log  
etc 267 log  
etc 268 log  
etc 269 log  
etc 270 log  
etc 271 log  
etc 272 log  
etc 273 log  
etc 274 log  
etc 275 log  
etc 276 log  
etc 277 log  
etc 278 log  
etc 279 log  
etc 280 log  
etc 281 log  
etc 282 log  
etc 283 log  
etc 284 log  
etc 285 log  
etc 286 log  
etc 287 log  
etc 288 log  
etc 289 log  
etc 290 log  
etc 291 log  
etc 292 log  
etc 293 log  
etc 294 log  
etc 295 log  
etc 296 log  
etc 297 log  
etc 298 log  
etc 299 log  
etc 300 log  
etc 301 log  
etc 302 log  
etc 303 log  
etc 304 log  
etc 305 log  
etc 306 log  
etc 307 log  
etc 308 log  
etc 309 log  
etc 310 log  
etc 311 log  
etc 312 log  
etc 313 log  
etc 314 log  
etc 315 log  
etc 316 log  
etc 317 log  
etc 318 log  
etc 319 log  
etc 320 log  
etc 321 log  
etc 322 log  
etc 323 log  
etc 324 log  
etc 325 log  
etc 326 log  
etc 327 log  
etc 328 log  
etc 329 log  
etc 330 log  
etc 331 log  
etc 332 log  
etc 333 log  
etc 334 log  
etc 335 log  
etc 336 log  
etc 337 log  
etc 338 log  
etc 339 log  
etc 340 log  
etc 341 log  
etc 342 log  
etc 343 log  
etc 344 log  
etc 345 log  
etc 346 log  
etc 347 log  
etc 348 log  
etc 349 log  
etc 350 log  
etc 351 log  
etc 352 log  
etc 353 log  
etc 354 log  
etc 355 log  
etc 356 log  
etc 357 log  
etc 358 log  
etc 359 log  
etc 360 log  
etc 361 log  
etc 362 log  
etc 363 log  
etc 364 log  
etc 365 log  
etc 366 log  
etc 367 log  
etc 368 log  
etc 369 log  
etc 370 log  
etc 371 log  
etc 372 log  
etc 373 log  
etc 374 log  
etc 375 log  
etc 376 log  
etc 377 log  
etc 378 log  
etc 379 log  
etc 380 log  
etc 381 log  
etc 382 log  
etc 383 log  
etc 384 log  
etc 385 log  
etc 386 log  
etc 387 log  
etc 388 log  
etc 389 log  
etc 390 log  
etc 391 log  
etc 392 log  
etc 393 log  
etc 394 log  
etc 395 log  
etc 396 log  
etc 397 log  
etc 398 log  
etc 399 log  
etc 400 log  
etc 401 log  
etc 402 log  
etc 403 log  
etc 404 log  
etc 405 log  
etc 406 log  
etc 407 log  
etc 408 log  
etc 409 log  
etc 410 log  
etc 411 log  
etc 412 log  
etc 413 log  
etc 414 log  
etc 415 log  
etc 416 log  
etc 417 log  
etc 418 log  
etc 419 log  
etc 420 log  
etc 421 log  
etc 422 log  
etc 423 log  
etc 424 log  
etc 425 log  
etc 426 log  
etc 427 log  
etc 428 log  
etc 429 log  
etc 430 log  
etc 431 log  
etc 432 log  
etc 433 log  
etc 434 log  
etc 435 log  
etc 436 log  
etc 437 log  
etc 438 log  
etc 439 log  
etc 440 log  
etc 441 log  
etc 442 log  
etc 443 log  
etc 444 log  
etc 445 log  
etc 446 log  
etc 447 log  
etc 448 log  
etc 449 log  
etc 450 log  
etc 451 log  
etc 452 log  
etc 453 log  
etc 454 log  
etc 455 log  
etc 456 log  
etc 457 log  
etc 458 log  
etc 459 log  
etc 460 log  
etc 461 log  
etc 462 log  
etc 463 log  
etc 464 log  
etc 465 log  
etc 466 log  
etc 467 log  
etc 468 log  
etc 469 log  
etc 470 log  
etc 471 log  
etc 472 log  
etc 473 log  
etc 474 log  
etc 475 log  
etc 476 log  
etc 477 log  
etc 478 log  
etc 479 log  
etc 480 log  
etc 481 log  
etc 482 log  
etc 483 log  
etc 484 log  
etc 485 log  
etc 486 log  
etc 487 log  
etc 488 log  
etc 489 log  
etc 490 log  
etc 491 log  
etc 492 log  
etc 493 log  
etc 494 log  
etc 495 log  
etc 496 log  
etc 497 log  
etc 498 log  
etc 499 log  
etc 500 log  
etc 501 log  
etc 502 log  
etc 503 log  
etc 504 log  
etc 505 log  
etc 506 log  
etc 507 log  
etc 508 log  
etc 509 log  
etc 510 log  
etc 511 log  
etc 512 log  
etc 513 log  
etc 514 log  
etc 515 log  
etc 516 log  
etc 517 log  
etc 518 log  
etc 519 log  
etc 520 log  
etc 521 log  
etc 522 log  
etc 523 log  
etc 524 log  
etc 525 log  
etc 526 log  
etc 527 log  
etc 528 log  
etc 529 log  
etc 530 log  
etc 531 log  
etc 532 log  
etc 533 log  
etc 534 log  
etc 535 log  
etc 536 log  
etc 537 log  
etc 538 log  
etc 539 log  
etc 540 log  
etc 541 log  
etc 542 log  
etc 543 log  
etc 544 log  
etc 545 log  
etc 546 log  
etc 547 log  
etc 548 log  
etc 549 log  
etc 550 log  
etc 551 log  
etc 552 log  
etc 553 log  
etc 554 log  
etc 555 log  
etc 556 log  
etc 557 log  
etc 558 log  
etc 559 log  
etc 560 log  
etc 561 log  
etc 562 log  
etc 563 log  
etc 564 log  
etc 565 log  
etc 566 log  
etc 567 log  
etc 568 log  
etc 569 log  
etc 570 log  
etc 571 log  
etc 572 log  
etc 573 log  
etc 574 log  
etc 575 log  
etc 576 log  
etc 577 log  
etc 578 log  
etc 579 log  
etc 580 log  
etc 581 log  
etc 582 log  
etc 583 log  
etc 584 log  
etc 585 log  
etc 586 log  
etc 587 log  
etc 588 log  
etc 589 log  
etc 590 log  
etc 591 log  
etc 592 log  
etc 593 log  
etc 594 log  
etc 595 log  
etc 596 log  
etc 597 log  
etc 598 log  
etc 599 log  
etc 600 log  
etc 601 log  
etc 602 log  
etc 603 log  
etc 604 log  
etc 605 log  
etc 606 log  
etc 607 log  
etc 608 log  
etc 609 log  
etc 610 log  
etc 611 log  
etc 612 log  
etc 613 log  
etc 614 log  
etc 615 log  
etc 616 log  
etc 617 log  
etc 618 log  
etc 619 log  
etc 620 log  
etc 621 log  
etc 622 log  
etc 623 log  
etc 624 log  
etc 625 log  
etc 626 log  
etc 627 log  
etc 628 log  
etc 629 log  
etc 630 log  
etc 631 log  
etc 632 log  
etc 633 log  
etc 634 log  
etc 635 log  
etc 636 log  
etc 637 log  
etc 638 log  
etc 639 log  
etc 640 log  
etc 641 log  
etc 642 log  
etc 643 log  
etc 644 log  
etc 645 log  
etc 646 log  
etc 647 log  
etc 648 log  
etc 649 log  
etc 650 log  
etc 651 log  
etc 652 log  
etc 653 log  
etc 654 log  
etc 655 log  
etc 656 log  
etc 657 log  
etc 658 log  
etc 659 log  
etc 660 log  
etc 661 log  
etc 662 log  
etc 663 log  
etc 664 log  
etc 665 log  
etc 666 log  
etc 667 log  
etc 668 log  
etc 669 log  
etc 670 log  
etc 671 log  
etc 672 log  
etc 673 log  
etc 674 log  
etc 675 log  
etc 676 log  
etc 677 log  
etc 678 log  
etc 679 log  
etc 680 log  
etc 681 log  
etc 682 log  
etc 683 log  
etc 684 log  
etc 685 log  
etc 686 log  
etc 687 log  
etc 688 log  
etc 689 log  
etc 690 log  
etc 691 log  
etc 692 log  
etc 693 log  
etc 694 log  
etc 695 log  
etc 696 log  
etc 697 log  
etc 698 log  
etc 699 log  
etc 700 log  
etc 701 log  
etc 702 log  
etc 703 log  
etc 704 log  
etc 705 log  
etc 706 log  
etc 707 log  
etc 708 log  
etc 709 log  
etc 710 log  
etc 711 log  
etc 712 log  
etc 713 log  
etc 714 log  
etc 715 log  
etc 716 log  
etc 717 log  
etc 718 log  
etc 719 log  
etc 720 log  
etc 721 log  
etc 722 log  
etc 723 log  
etc 724 log  
etc 725 log  
etc 726 log  
etc 727 log  
etc 728 log  
etc 729 log  
etc 730 log  
etc 731 log  
etc 732 log  
etc 733 log  
etc 734 log  
etc 735 log  
etc 736 log  
etc 737 log  
etc 738 log  
etc 739 log  
etc 740 log  
etc 741 log  
etc 742 log  
etc 743 log  
etc 744 log  
etc 745 log  
etc 746 log  
etc 747 log  
etc 748 log  
etc 749 log  
etc 750 log  
etc 751 log  
etc 752 log  
etc 753 log  
etc 754 log  
etc 755 log  
etc 756 log  
etc 757 log  
etc 758 log  
etc 759 log  
etc 760 log  
etc 761 log  
etc 762 log  
etc 763 log  
etc 764 log  
etc 765 log  
etc 766 log  
etc 767 log  
etc 768 log  
etc 769 log  
etc 770 log  
etc 771 log  
etc 772 log  
etc 773 log  
etc 774 log  
etc 775 log  
etc 776 log  
etc 777 log  
etc 778 log  
etc 779 log  
etc 780 log  
etc 781 log  
etc 782 log  
etc 783 log  
etc 784 log  
etc 785 log  
etc 786 log  
etc 787 log  
etc 788 log  
etc 789 log  
etc 790 log  
etc 791 log  
etc 792 log  
etc 793 log  
etc 794 log  
etc 795 log  
etc 796 log  
etc 797 log  
etc 798 log  
etc 799 log  
etc 800 log  
etc 801 log  
etc 802 log  
etc 803 log  
etc 804 log  
etc 805 log  
etc 806 log  
etc 807 log  
etc 808 log  
etc 809 log  
etc 810 log  
etc 811 log  
etc 812 log  
etc 813 log  
etc 814 log  
etc 815 log  
etc 816 log  
etc 817 log  
etc 818 log  
etc 819 log  
etc 820 log  
etc 821 log  
etc 822 log  
etc 823 log  
etc 824 log  
etc 825 log  
etc 826 log  
etc 827 log  
etc 828 log  
etc 829 log  
etc 830 log  
etc 831 log  
etc 832 log  
etc 833 log  
etc 834 log  
etc 835 log  
etc 836 log  
etc 837 log  
etc 838 log  
etc 839 log  
etc 840 log  
etc 841 log  
etc 842 log  
etc 843 log  
etc 844 log  
etc 845 log  
etc 846 log  
etc 847 log  
etc 848 log  
etc 849 log  
etc 850 log  
etc 851 log  
etc 852 log  
etc 853 log  
etc 854 log  
etc 855 log  
etc 856 log  
etc 857 log  
etc 858 log  
etc 859 log  
etc 860 log  
etc 861 log  
etc 862 log  
etc 863 log  
etc 864 log  
etc 865 log  
etc 866 log  
etc 867 log  
etc 868 log  
etc 869 log  
etc 870 log  
etc 871 log  
etc 872 log  
etc 873 log  
etc 874 log  
etc 875 log  
etc 876 log  
etc 877 log  
etc 878 log  
etc 879 log  
etc 880 log  
etc 881 log  
etc 882 log  
etc 883 log  
etc 884 log  
etc 885 log  
etc 886 log  
etc 887 log  
etc 888 log  
etc 889 log  
etc 890 log  
etc 891 log  
etc 892 log  
etc 893 log  
etc 894 log  
etc 895 log  
etc 896 log  
etc 897 log  
etc 898 log  
etc 899 log  
etc 900 log  
etc 901 log  
etc 902 log  
etc 903 log  
etc 904 log  
etc 905 log  
etc 906 log  
etc 907 log  
etc 908 log  
etc 909 log  
etc 910 log  
etc 911 log  
etc 912 log  
etc 913 log  
etc 914 log  
etc 915 log  
etc 916 log  
etc 917 log  
etc 918 log  
etc 919 log  
etc 920 log  
etc 921 log  
etc 922 log  
etc 923 log  
etc 924 log  
etc 925 log  
etc 926 log  
etc 927 log  
etc 928 log  
etc 929 log  
etc 930 log  
etc 931 log  
etc 932 log  
etc 933 log  
etc 934 log  
etc 935 log  
etc 936 log  
etc 937 log  
etc 938 log  
etc 939 log  
etc 940 log  
etc 941 log  
etc 942 log  
etc 943 log  
etc 944 log  
etc 945 log  
etc 946 log  
etc 947 log  
etc 948 log  
etc 949 log  
etc 950 log  
etc 951 log  
etc 952 log  
etc 953 log  
etc 954 log  
etc 955 log  
etc 956 log  
etc 957 log  
etc 958 log  
etc 959 log  
etc 960 log  
etc 961 log  
etc 962 log  
etc 963 log  
etc 964 log  
etc 965 log  
etc 966 log  
etc 967 log  
etc 968 log  
etc 969 log  
etc 970 log  
etc 971 log  
etc 972 log  
etc 973 log  
etc 974 log  
etc 975 log  
etc 976 log  
etc 977 log  
etc 978 log  
etc 979 log  
etc 980 log  
etc 981 log  
etc 982 log  
etc 983 log  
etc 984 log  
etc 985 log  
etc 986 log  
etc 987 log  
etc 988 log  
etc 989 log  
etc 990 log  
etc 991 log  
etc 992 log  
etc 993 log  
etc 994 log  
etc 995 log  
etc 996 log  
etc 997 log  
etc 998 log  
etc 999 log  
etc 1000 log

⑥ Scaling just like it scale the batch. But this batch is scaling set  
→ to standard individual scaling of individual batch to fit overfitting reason of standard deviation  
→ so that we do  
DO → add global value like  $\mu, \sigma$ , then it will be fit same as fit  
etc 222 में वर्क करोगा।

- Single origin के लिए

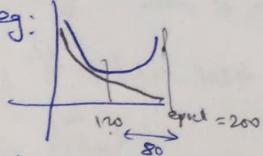
## # Inference

• Why momentum including in Batch Normalization?

→  $\hat{x}_1, \hat{x}_2, \hat{x}_3, \dots$  Normalized  
→  $\hat{x}_1, \hat{x}_2, \hat{x}_3, \dots$  overfitting

## # Callbacks

e.g.:



but test result in 120 epochs  
then how to do early stop

so we need utility function  
to monitor the inference &  
if not improve same the  
best then save  $W, B$ :

① Model checkpoint: → more all epoch save  
↳ class use for ~~any~~ check

② Early Stopping: class

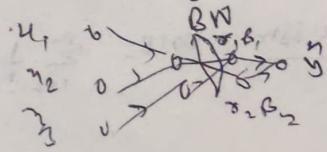
↳ Not do unnecessary epoch

min delta → improve point or criteria  
Patience = How many epoch you need

⇒ How to implement

↳ in model.fit(..., callbacks=[..., ...])

Inference; time & time  
(testing time)



• during testing time ~~not~~ EBT  
 $\approx B$

③ fit 12000 data to predict  
in test time

↳ not in  $N_1, N_2 \rightarrow$  it

$\bar{D} \text{ on } N_1^t \rightarrow W_2^t \Rightarrow$

$$N_1^t = \frac{N_1 + \bar{D}}{2}$$

not during in testing time

↳ moving mean

↳ moving variance

↳ USE

Next

## # Hyperparameter tuning

→ pip install keras-tuner -q

→ import KerasTuner

→ tuner = KerasTuner(Hyperparameters())

→ def function(hp):

↳  $\text{layer} = \text{Dense}(\text{units} = \text{hp.Int('units', min_value=32, max_value=512, step=32), activation='relu')$

↳ n of unit start  
Variable name min=32  
define

④ How we define n of layer

• bestmodel

① Grid Search

② Random

③ Bayesian

→ custom Class → create

Loss

Regression MSE  $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

How to solve by  $\frac{\partial L}{\partial w}$   
Chain Rule

In Classification:

| $y$ | $\hat{y}$ |
|-----|-----------|
| 0   | 0.2       |
| 1   | 0.9       |
| 0   | 0.8       |
| 1   | 0.2       |
| 0   | 0.9       |

then How to find diff error  
because  $y$  in  $0, 1$  &  $\hat{y}$  is range of  $0$  to  $1$

Use Log Loss (mutually exclusive)

$$\sum_{i=1}^n \frac{1}{n} (\hat{y}_i \log \hat{y}_i + (1-\hat{y}_i) \log (1-\hat{y}_i))$$

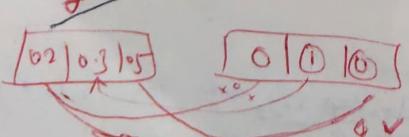
दोनों तरफ दोनों तरफ असे

Categorical crossentropy:

$$\sum_{i=1}^n \sum_{j=1}^K y_{ij} \log \hat{y}_{ij}$$

If  $i=j$  then it is binary cross entropy.

If any goes closer to 1  
another automatically  
goes 0.



Sparse categorical cross if we represent in matrix  
if all data called sparse matrix

Imbalanced Class

use SMOTE | under/over sample  
not adding any  
Value import

→ In Decision Tree class  
(Class-Weight?)

• Then How to assign weight

$$\frac{(S \times y \log(\hat{y})) + 2(1-y)\log(1-\hat{y})}{T}$$

weight to give

DT  
calculate split  
(Gini, ...)

① In keras fit → which parameter  
solve this problem

• Class-Weight

• Sample-Weight (what is use of it)

② then weight use 0

③ then weight use 1

4 Steps of Analytics  
 ① Descriptive, ② Diagnostic, ③ Predictive,  
~~④ Model~~ ④ Prescriptive

## # Model Explainability (NN)

very explainability)

↳ So I can overcome on that point to improve my bad side.

- Why is loan is rejected or Accepted so we can

1. Explain why

↓  
How

Using some Model

①

LIME

Local Interpretable

model agnostic

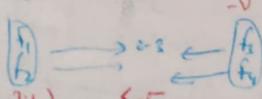
Explainability

for all model type data

also explain for single data

all type of model

True



where equilibrium is 0.3  
so reject because by criteria is 0.5.

Eg Credit risk prediction

create simple linear Eq of these local part

so fit it  
 $f_1, f_2, \dots, f_n$

↳ to

(the greatest -ve side is that why reject here.)

→ lime package output

Step → model.predict( $x_{test}$ ) →  $\hat{y}$  → Accept

→ Reject ✓

then why rejected

sum → add some synthetic data  
 sum → near proveen. and create model

Model  $\boxed{x} = \boxed{y}$  via LIME

- to explain the case why rejected.

② Shapely

is used to explain the model how it works.

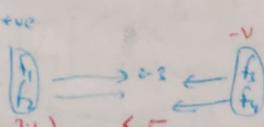
In single row of data also explain

→ explain how  $y$  sperred is coming.

eg:-  
 Model → Prob  
 ↓  
 0 → 0.3

Rejected → how right no need but need why rejected.

- which feature is more putting the model to +ve side so which feature is putting to the -ve side



where equilibrium is 0.3  
so reject because by criteria is 0.5.

qui est plus nombreux dans le sud

les deux seuls qui sont très rares

Griffon

à l'abordage des navires

qui  $\rightarrow$  à la fin  
bonne boussole pour faire une  
cette chose difficile

peut être utile

qui  $\rightarrow$  à la fin

qui est un peu bête (peut)  $\rightarrow$  à la fin

qui est assez malin

qui a de bonnes idées

qui a de bonnes idées

qui est assez malin

qui a de bonnes idées

qui

bien sûr

qui

qui est assez malin

qui est assez malin