→ Tensorflow

Asmita

DEEP LEARNING

use ML or DL
① Length of data
② Type of data

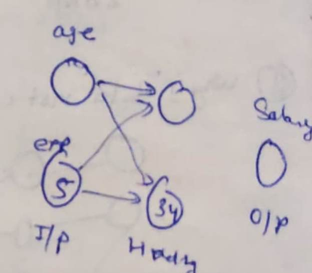
- High level overview
- What is ML & How to it works
- Why we do a ML and use case
  → predict the
- What Lack in ML so Come DL → Use neural networks → To do the feature Extracting
  eg: Human noise

ML → Most of Structure → Data → Feature Engg

DL → Data Like Image & Text not so much well in ML so introduce DL

ML
- High human intervention → feature Engineering
- ML Model get trained Quicky
- ML Model can be trained on CPU
- ML Model don't perform as good as DL Models on big data

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



eg :-
- Noise dimension
- lips
- eyebrows
- Cheeks
- RGB

so it is Multidimention

ML Model hard to extract feature from higher dimention

ML
- area - float
- location - common / Rem
- furnished 0/1
- floor int
- room int

→ Structure data → in dataframe and easily Apply ML Model

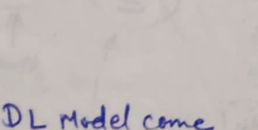
BUT

in hell, i am Shahil → what gone we do → Here DL Model come
What DL Do → take input vocab process in NN then in output like in hindi

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

→ In ML Algorithms but DL is a Neural Networks.

→ Artifical Neuron (1st create - 1943) (Warren & Walter)
  → It is Mathematical Representation of Biological Neuron
  → operation on following principles
  ① Binary Inputs & Output
  ② Threshold logic → logic (And gate, OR, NOT)

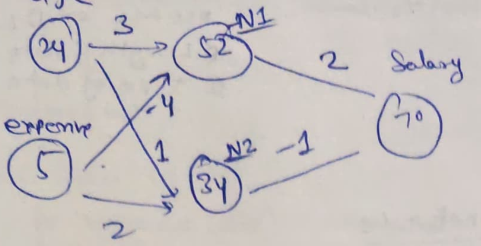


AND
L1 = 1
L2 = 0
O = 0

eg :- only for binary
- girl or boy
- fruits & veg

eg:

age



→

← FNN

Backwarked NN

Back propagation

we (optimizers)

$y_A = 50$

$y_P = 70$

Do - Dot Product

$24*3 + 5*(-4) = N1$

again for next layer

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

↳ This is FEED FORWARD NN

FNN or MLP

↳ get input

↳ Calculate the hidden layer Value/process

→ get the output

Error = 70-50 = 20

threshold = 5 (min error)

Minimum error we accept

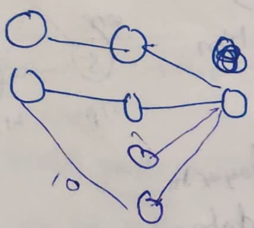
→ feeding the data in forward direction

→ The weights are automatically decide 1st Random initiate

↳ eg in face check happy, sad, netural only lips moment change so lips weigh heigh then Nose or eye.

1 Epoch (Where 1 FNN, 1 OUT, 1 Hidd, 1 BNN)

Q Why 1 data not connect to 1 layers?



Q How much Layer we define?

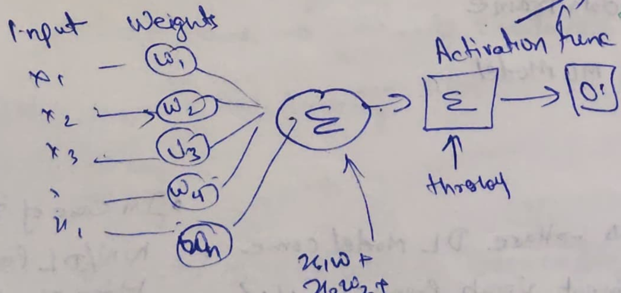
↳ Hit & Trial (

eg: Resnet (try & tested)

Q Which

PERCEPTRON → Consist of 1 I Layer, 1 out, No hidden, weight, bias, Summation funct, Activation fun, TLU (Threshold)

Simple/Basic ANN Architecture



→ It takes in consideration the value of threshold & weight sum

- Assign the value to the node

eg (Relu, Sigmoid

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

$O/l = 4$

↳ $\max(0,-1) = \text{output} = 0$, threshold = 2

TH > Out = do Backpropagation

$\max(0,4) = \text{output} = 4$, TH = 2

→ TH < output ⟹ ✓ return the output

- Whole process called

Perceptron Learning Rule

1. Initialize weights
2. Compute the weighted sum
3. → $Z = \text{total}(w_n x_n + b)$
4. Apply Activation functions
5. if Model fails, Update Weights
6. Otherwise simple return the output

use for create layer of FFN  Simplest ↳ No hidden layers

Perceptron → 1 input layer & 1 output node
↳ beg perceptron, very simple, it can't be used in Real life.

o →
o → O
o →

These line called feed forward Net

if Hidden layer the called MLP

In Hidden layer – 2 type → Shallow NN (only 1 HL)
→ dense NN (more then 1 HL)

Feed
↳ ~~Pattern~~ feeding the data
forward

use in Activation function

Linear data	Non-Linear data
	eg: text, image ↳ Use for Non-Linear Activation function

→ API
↳ take input return output

eg:- High Level API
Python - 1, 2, ... n. ↳ Low level (use loop) Scratch
or Them → X = Sum(1, 2)

→ keras & Tf integrated
→ TF → only Python lib
→ KERAS (inbuilt all) to call
↳ High level API
↳ can be Integrated
↳ ex. TF use keras
↳ from scratch to Build layers

MLP (Multilayer Preceptron)
- NN build upon perceptron
- Uses the functionality of Perceptron, but they hidden layers.
- hidden help us understand complex data.

– 2 type of NN

① Shallow NN	② Dense NN
• 1 Hidden layer	→ more than 1 HL

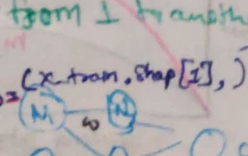
Tensorflow: → Use Build layers in optimize calculation

like
```
import tensorflow as tf
from tensorflow import keras
from "  "  .keras import Sequential

model = Sequential()
```
↳ flow → Pass the layer from 1 to another
↳ like Construct

Create Shallow NN
Input → model.add(tf.keras.layer.Dense(2), input_shape=(x_train.shape[1],), activation = 'relu'))
input net
Output → model.add(tf.keras.layers.Dense(1)

x_train.shape[1]
Sho



So Dense in function to do 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 ,

Model.summary( )

→ df = sns.load_dataset('mpg')
df.columns

→ pd.get_dummies(df, Columns=['origin'], drop_first=True
, Prefix='origin', dtype=int)

datatype in TF is tensor

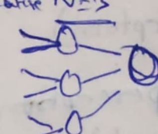
Stats)
if the value is b/w 1 to 1, we don't have any outliers

np.where()
apply condition Ternary Condit
(df['HP'] = np.where(
df['hp'] >200, 200
where(condition, T, F)

Summary Layer	Output	Random init (weight & bias) Params
(Dense)	(None, 32) — batch size for backtrack	736
Dense 1	(None - 1	

feature = 22
node = 32] 22×32
→ 704 (weights)
+ also 32 bias add
→ 736

If 2 feature N=3



→ bias → non lin

3 Aug

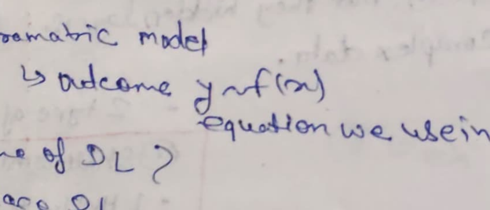
What is DL → Parametric model
↳ outcome y~f(x)
equation we use in DL

. What Requireme of DL ?
① Where we place DL
② It is necessary to Learn ML before DL
③ If we Do in ML then why do DL ?
④
↳ y = f(x)
↳ in DL → video, Audio, text tableause, graph
but ML works in Structe data
① architecture → CNN image
② DL is Multimodality → link audio + Audio
↳ Multi type data Pass
for better feature extraction

eg: doctore check for checkup

ML
Class, Reg | eq Parametric, Non-Param

③



DL → automated feature engineering

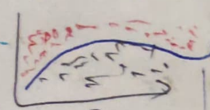
Q What

3 Aug 21

③ What is a neuron?

playground.tensorflow.org

eg Logistic Rege



so find is hard eqne

3 way to do

① any fuction given

② do feature transformat

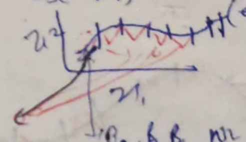
③ add neuron for non linearity

Not possible in Logistic Regres

So what we do

we divide into multiple part

use this



break-down in simple line

→ $B_0, h\ B_2$, NL

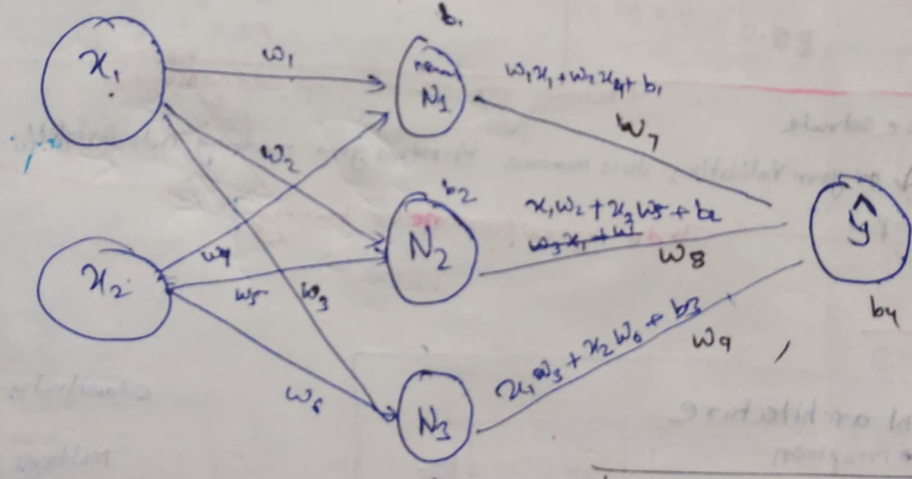
NL ($w_1x_1 + w_2x_2 + b$)

0:40 Visual of Neuron

⇒ N are local decision boundary → and compine to create Complex decision boundry create

1 HL features

01:15 eg: Maths of Neuron



So we can say $N_1\ N_2\ N_3$ are new feature create from x_1, x_2 → which represent data space b/w of sp of closest to dicision boundry which are closer to the boundary shape

$$\boxed{\hat{y} = w_7N_1 + w_8N_2 + w_9N_3 + b_4}$$

$N_1 = w_1x_1 + w_2x_4 + b$, $N_2 = x_2w_2 + x_2w_5 + b_2$, $N_3 = x_1w_3 + x_2w_6 + b_3$

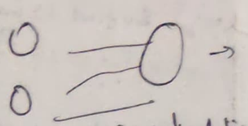
$\hat{y} = (w_7(w_1x_1 + w_2x_2 + b_1) + w_8(w_2x_2 + x_1w_5 + b_2) + w_9(x_1w_3 + x_2w_6 + b_3)$

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

Note: if we not add any Non-linearity in this it alway be in ~~line~~ simple linear line fit.

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

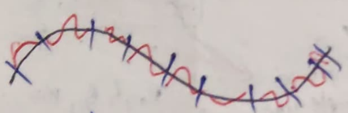
If you dont have hidden layer then you have no any oppurtunity to divide the decision boundary into sub section.



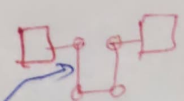
a soonly 1 line create and we know 1 single straigh line not define non-linear boundary.

→ So if increase the Hidden layer the sub increse the sub parting

So ↑ HL its means increase the degree of Non-linearity



So divide and concore



if i give you 10000 staight Line but you cant to create this path without Mold so what we do need mold ↓ Activation funtion for Mold.

⇒ When click the Play button to Playground-tensorflow it adjust the weight by minimizing the loss.
(called ↓ using Gradient Desent, → minimne the loss by updating w.B for all neurons.
↓ 1st time random initiliz

Q&A
Bys Gradient Desent

Know where to stop adding hidden layers
1) Loss not improve saturate
2) Traing Loss is ↓ or your Validation loss increse it mens you entering into Overfitting so you stop
↳ do hyperparameter tuning

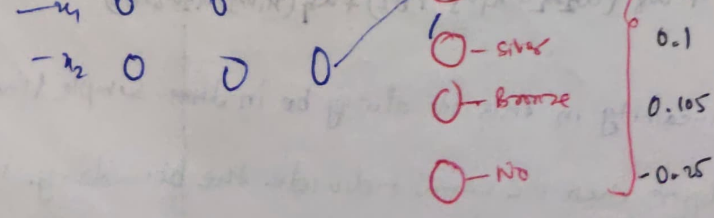
Aug 4 DL
1. Define NN architecture
→ Precptron
→ HLS
→ Nurons
→ activation functions
→ Weights and biases
→ optimizer and loss function

Classification Sigmoid (Binary (0-1)
multicall (Softmax(5)
Reg tanh → -1 to 1

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

work on Probability distribution
↳ Softmax ($\hat{y}$) convert into Probability
4 outp, Gold, Silver, Bronze, No Medal

eg: to how NN work



Softmax → change Scaler/numer into Probability (to 0 to 1) $\frac{e^{a_1}}{e^{a_1}+e^{a_2}+e^{a_3}}$

redcrab-software.com/en/Calculator/softmax

$a_1 = 1$ $\frac{e^1}{e^1+e^3+e^2}$

a_2 3

a_3 2

0 0 0

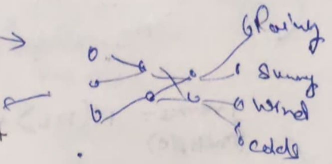
0 0 0

1

(Q) (a) The output layers of neurons in multiclass classification is Softmax (→ Change Value in Probability Value)

agree

1st extract the Non-linear



So yo tell Predict toward 1

why need Probability Value

↳ Use log loss = for all rows of data sum

$-(y_1 \log(\hat{y}_1) + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3 + y_4 \log \hat{y}_4)$

$= -\log 0.2$

$= 0.69$

ex log 0.8 ⇒ 0.09

→ the loss function Continuous & differentiable

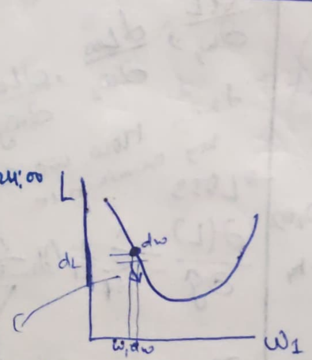
↳ is log loss function do

01:27

10 Aug

24:00

Loss



more to opposite of the gradient So it go to the minima side → d

$\boxed{\frac{dL}{\partial w} = -ve \quad \frac{dL}{\partial w} = +ve}$

$w_1(new) = w_1(old) - \left(\pm \frac{dL}{\partial w}\right)$

± depends of slop

if slop decline −ve

or incline then +ve

To minimize loss x → [W&B] → Loss

When calculate the Loss Called forward Propagation (FP)

Loss = f (y Pred, y actual)

L = f ((w, B, x), y actual)

L = f(w, B)

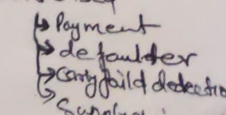
To improve or change LOSS by changing w, B called Backward Propagation (BP)

why Choose DL (Paramete

① x → Any type of data (video, audio, text, ...) Multi modularity

② Text → [ML] → Diagnosis ↑ test Scan

③ performance



① DL → automated feature engineering

eg

→ Merchindised

↳ Payment

↳ defaulter

↳ Card fould detection

↳ Supply chain opting

→ Activation

00:27:

Lecture 3

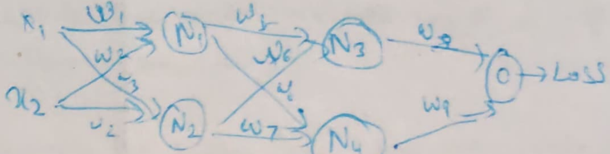
| Chain → if i change w_1 to 1 unit where all the effects are getting changed

(9) Activation function

00:43

1:15

ex, Chain



$$\left(\frac{dL}{d\hat{y}}\times\frac{\partial \hat{y}}{\partial N_3}\times\frac{\partial N_3}{\partial N_1}\times\frac{\partial N_1}{\partial w_1}\right)+\left(\frac{\partial L}{\partial \hat{y}}\times\frac{\partial \hat{y}}{\partial N_4}\times\frac{\partial N_4}{\partial N_1}\times\frac{\partial N_1}{\partial w_1}\right)$$

Terms (multiple) = N(HL)+2 हो जाता है in Back probagation

↳ May be Vanishing Gradient)

ex $(0.99)^{365} \approx$ ...,
$(1.01)^{365}$

if number are small in hidden layer & Multiply then these number more small small & when based on this calculate Loss it is also very very small

→ So what happen → gradient is not happen.
↳ Prob (Vanishing gradient)

→ if multiple is big Number → Loss is high → so gradient is very large → So it over-suit with minimum Loss
↳ (Exploding gradient Problem

$\hat{y} = w_7N_1 + w_2N_2 + w_3N_3 + b_4$

↳ why we see this because which ~~Gradient~~ Activation function we use in these Problem to overcome.

$e^x \sin(y)$

0:30 ex:

$y = x^2$
$m = y^2$
$Z = m^2$

differentiation

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

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

→ not direct connected so we do change y w.r.t x so indirectly solve by Partial derivative

here easy to do P.D because here simple eq

but $\frac{\partial e^x \sin(y)}{\partial x}$ how we do now

here come $\boxed{\text{Chain Rule}}$

How Solve ← $\frac{dz}{dx}$

↳ $\frac{dz}{\partial m} \times \frac{\partial m}{\partial y} \times \frac{\partial y}{\partial x}$

↳ $2m \times 2y \times 2x$

What purpose to teach us?

Yes because of Loss

$\frac{dL}{\partial w_1}, \frac{dLoss}{\partial w_2}, \frac{\partial Loss}{\partial w_9}$ .....

Then How we calculate it by chain rule

→ Loss

$$\frac{\partial (L)}{\partial \hat{y}} = \sum_{i=1}^{n} \frac{1}{n}(y_i - \hat{y}_i)^2 \Big\} mse$$

$$\frac{\partial L}{\partial w_7} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_7}$$

$\Downarrow$

$$\sum_{i=1}^{n} \frac{1}{n} \times 2(y_i - \hat{y}_i) \times -1 \times N_1$$

Class Works

$\frac{dL}{\partial w_9} = ?$

$\frac{\partial L}{\partial w_1} =$ using chain rule

1:10

• AF → should be continous & diffrentiable

Sigmoid
↳ binary
Softmax
↳ multi Class

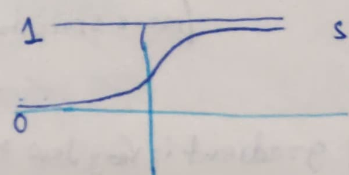
Activation functions are functions which should be Continuous & diffrenciable that can be applied on the output of my hidden layer to create non-linear feature or at the output layer to create bounded output.
(like Probabilty)

2

→ Signoid:

i/p = $-\infty < x < \infty$

o/p = $0 < \sigma(x) < 1$ $\quad \sigma(x) = \dfrac{1}{1+e^{-x}}$

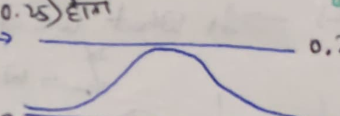


squashing the tale

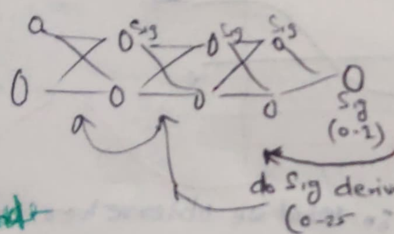
Simple Sigmoid

diffrencient Sigmoid

Use: So first derivative of Sigmoid is (0-0.25) होता
So its gradient is low



What is problem
(Problem is Vanishing Gradient)



So Sig
we do back probagation

do Sig derivative
(0-25

Biggradient
RNN

Sigmoid
(good for Last Output
↓
but not good for hidden layer)

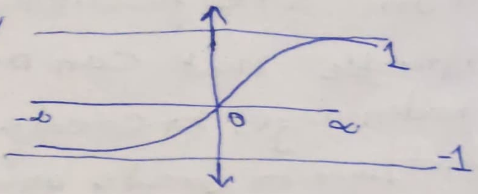
Tanh

② Tanh :

I/p = $-\infty$ to $+\infty$

O/p = -1 to 1

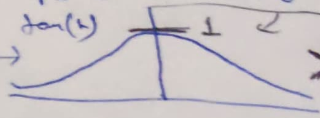
$$\boxed{\tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}}$$



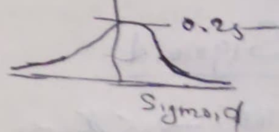
→ more variance ~~from~~ Sigmoid

→ tanh → more use in Hidden layer

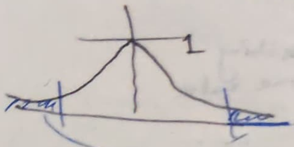
→ & Both has S-shape so derivative of tanh is also gradient / →



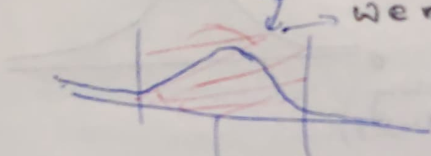
\>



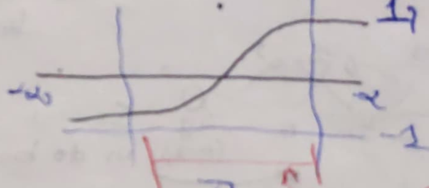
→ It Solve Vanishing Problem or not / more space to fit value



if our Value of gradient is very low then it also show Vanishing Problem? yes → So what we do



→ we need only this area ok then How we get it



So what we observe here?
if input of (x) lies in this region

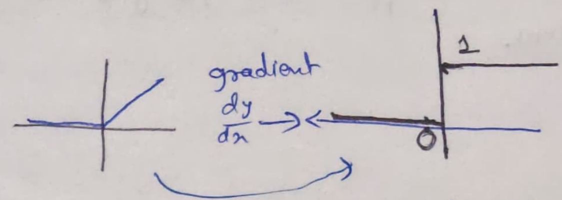
How fix range → ① Do Standard Scaler
② also w_1, w_n Value range to be -1 to 1 at initialize time

③ ReLu :-

Relu(x) = $x \le 0$ → 0

$x > 0$ → x

gradient $\frac{dy}{dx}$



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

• dropout is good but in ReLu dropout is ~~and~~ not intentionally

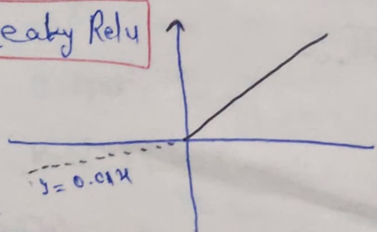
So It means we can't control it. ~~gest~~

⇓

then How Remove it (or control it)

Using Leaky ReLu

④ Leaky Relu



LR(x) ⇒ $x \le 0$	$0.01x$
$x > 1$	1

⑤ → Parametric Relu:

$y = x \quad x > 0$

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

$0 < \alpha < 1$

if $\alpha = 0.01$ → called Leaky Relu

How is this a non linear function

↓

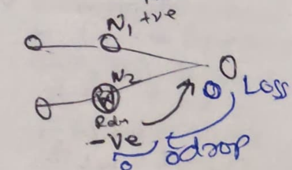
at least 2 slop at any point in the ~~x~~ space

and

↳ not vanishing gradient here beside +ve is 1

disAdv

↳ -ve (drop)



So when do backprop the ~~lay~~ NN is drop because Value is 0

So N_2 drop here

• So in deep NN the drop is good

↳ but not for all

↳ (Shallow NN)

↳ Changes the batch

↳ example

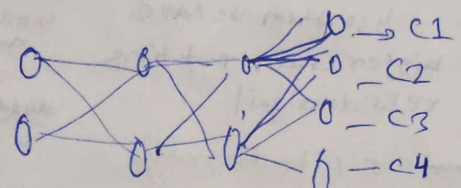
↳ Student

Chap1 Chap2

ताकी हम दोनों Exam में मार करे

(V) **Softmax** :- need to be applied the O/L of a multiclass Problem.

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



$\boxed{P_1 + P_2 + P_3 = 1}$

How $P_1 = \dfrac{e^{z_1}}{e^{z_1} + e^{z_2} + e}$

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

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

Softmax

: ① — P_1 only

⓪ — P

$P_1 = (P_1)$

$P_2 = \cancel{1(P_1)}$ $1 - P_1$

compose → column transforming

Code Apply

Code

→ to_categorical
(

OHE.fit_transform(y.reshape(-1,1))

→ Input layer

→ functional API
↳ Inputs
but in sequential not need to define Input layer क्योंकि वहाँ पर Sequential में होता है but in F API कही भी कोई भी layer define कर सकते है।

Define functional API

(4,2)
[1]
→ 4 cols

input (Shape = (X_train.shape[1],))

X = Dense (64, Activation = 'relu') (inputs) ← Applied in input layer

↳ here define the explicitly the hidden layer

here this layer look like a function
और इसके top में (inputs) को Apply करो
↳ तो we are free to apply any input layer,

X = Dense (32, activation = 'relu') (X)

Output = Dense (y_train.Shape[1], activation = 'softmax') (X) → यहाँ पर हम (inputs layer भी) Apply कर सकते है But if needed

Model = Model (inputs = inputs, outputs = outputs)
(यहाँ 2 things clear की Input कहा से आएगा और output क्या होगा) बस

→ Define Architecture

model.compile (optimizer = 'adam', loss = 'categorical_crossentropy',
metrics = ['accuracy'])

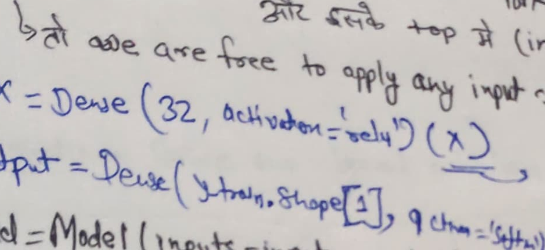
Q

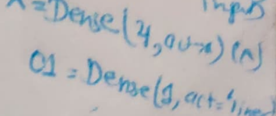
① inputs → $HL_1(30)$ → $HL_2(45)$ → (4)
↳ $HL_3(30)$ → O(2)

② I/P → $HL_1(30)$ → $HL_2(4)$
I/P → H(20) → H(50) → Concat

Concat()([HL3, HL4])

Also do like





Input (Shape = (10,))
X = Dense(5, activation = 'relu') (inputs)
X = Dense(4, activation) (X)
O1 = Dense(1, act = 'linear') (X)
O2 = Dense(3, act = softmax) (X)

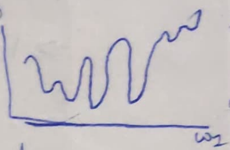
ये जो है वो Sequential में नहीं हो पाता है

model = Model (input = inputs, outputs = [O1, O2])
Model.compile (Loss = {O1 : 'MSE', O2 : 'categorical_crossentropy'},
optimizer = 'adam')
metrics = {O1 : 'mae', O2 : 'accuracy'}

Q Any fair Model Accuracy impact by Sequential & function api use?
Ans NO

18 Aug 2024

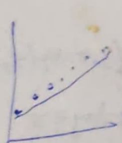
→ ① eg Loss



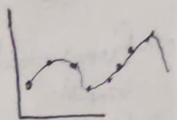
in 1 direction w_1

2 - - - - - w_2 ...

w_n



⇒

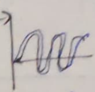


↑ then also not overfit

→ not overfit

sin(x)

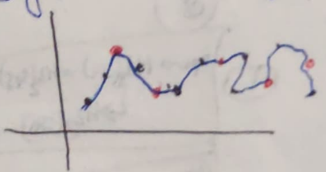
↳



If data is Noise then we can Say that ~~No~~ Overfitting.

• outliner

↳ No my Model is overfit



→ Not fit Propert

→ Local Minima

→ Batch size

→ Regularization

→ Batch Normal

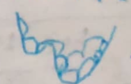
→ Weight Initial

→

Optimizers

(1) → Regularization (Ridge | Lasso)
(2) → ① Dropout (Weights of NN will be 0) for certain iteration
② What is Optimizers
" " Gradient descent
SGD
· Momentum
③ types
Callbacks
④ Batch Normalization
⑤ Weight Initialization

What is enough Momentum
↳ Pols HP
Hyperparameter
→ It takes time but get Global minim



↳ but if shape of Error
W ← here the Moment Stuck becs become same

• historical gradien

① → Momentum Solve the local minima

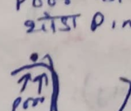
② Ada Grand
= Why create Adagrad? what problem is solve?

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

$$v_t = v_{t-1} + \left(\frac{\partial L}{\partial w_t}\right)^2 \quad v_{t+1}$$

↳ next → India is good but implementation is good.
no decay

→ accummulate gradient
eg:- indian tax system
↳ Rich से पैसे ले कर Poor को देना ताकी उन्हें थोड़ा Pinalize किया जाए



↓ so ③
introduce Rms Prop
add decay

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

HP → β
It works as a Hyperparameter

$$v_t = \underset{0.99}{\beta}\, v_{t-1} + (1-\underset{0.01}{\beta}) \times \left(\frac{\partial L}{\partial w_t}\right)^2$$

why → depend on shape of case
→ Adaptive → it depends on the shape of curve
So called adaptive
not depending on user input or Learning Rate

→ Here momentum is missing
↓ so Some ring Come with ADAM

④ ADAM (Adaptive Moments)
add moment

momentum
$$\boxed{m_t = \beta_1 (m_{t-1}) + (1-\beta_1) \frac{\partial L}{\partial w_t}}$$

$$\boxed{v_t = \beta_2 (v_{t-1}) + (1-\beta_2) \times \left(\frac{\partial L}{\partial w_t}\right)^2}$$

$$\boxed{w_{t+1} = w_t - \frac{\alpha}{\sqrt{v_t+\epsilon}} \times m_t}$$

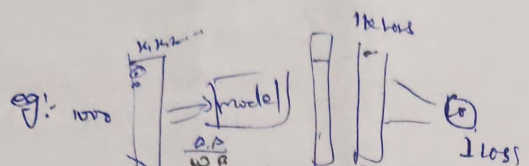
→ why moment
↳ degree
→ 1st → momentum
2nd deg → Momentum

Call back is not function
↓ it is tracking function

↪ In time sense it work
↪ yes work but different way to use there.

⑤ MiniBatch Gradient descent

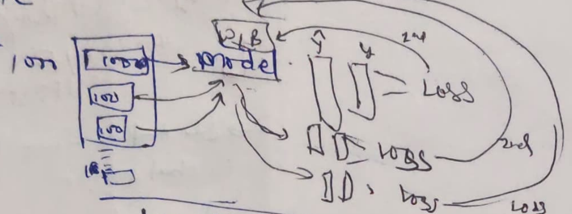
→



eg:- 1000

↪ Random (सबके बराबर w, B आ रहा है)
↪ 1 epoch

2nd example Case

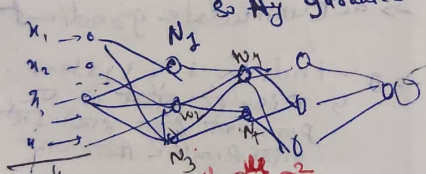


here 1st Random → 2nd time Some learning
तो हर time जो w, B improve होगा इसलिये
यहाँ थोड़ा अच्छा है

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

31th Aug

① Batch Normalization :

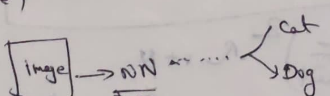


Same range (easy for NN for easy)
→ more stable training

$\frac{N_1 - \mu}{\sigma^2}$ {do Scaling

So Any garantee की N_1 Normalize होगा → No w_9
तो आगे w_4, N_5 भी नहीं होगा So on

तो जो Normalize शुरू में हुआ वो किसी भी HL में नहीं रहता है।



eg:- image → NN → Cat / Dog

तो यहा Model Shallow होगा
black dog अलग
white dog अलग

तो यहा सबके लिए अलग Decision Boundary बनाएगा।

called Covariate Shift

→ it is waste of time

remove means to get in Centers of all data → तो यहाँ NN → Stabalize faster N_9

• We Scaled the feature in HL it help

→ it is do overfitting.

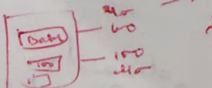
तो data dead हो जाएगा

So what we do then

$\boxed{\gamma_1 N_1 + \beta_1}$

γ, β_1 → सबमे Same रहेगा ताकी

these are the Mean & Variance of all data



Scale

अलग Mean & Variance

अगर होते तो अलग Original करेंगे तो होता identical तो हमें 1 original set इसे होगा

→ So that we do

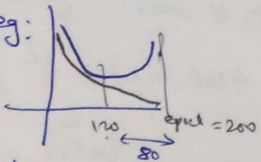
③ यदि Scaling जरूरी है तो Scale कर रहे हैं Batch में But अगर Batch में Scaling करते है तो सबका individual Scaling हो जाएगा Batch को जो overfit का Reason हो सकता है तो

→ Do → Add global Value like γ_1, β_1 ताकी वह हर Batch के लिए Same रहे। तब यह अच्छे से work करेगा। - Single original के लिए

Inference

Callbacks

eg:



but best result in 120 epoch
then How to do early stop
⇓
so we need utilite function
to monitor the improvent &
if not improve same the
best then save W, B.

① Model checkpoint ① → here all epoch run 1st
↳ class use for ~~early check~~

② Early Stopping ; class
↳ Not do unnecessary epoch
min delta → improve point/or criteria } experiment
patience = How many epoch you need

⇒ How to implement
↳ in model.fit(....., Callback = [,])

1st
Hyperparameter tuning
→ Pip install keras-tuner -q
→ import keras_tuner
→ function (keras_tuner. Hyperparameters())
→ def function(hp):
layer = Dense(units = hp. Int("units", min_value=32, max_value=512, Step=32), activation='relu')
variable name define; no of unit start min=32; No of NN का grid define here

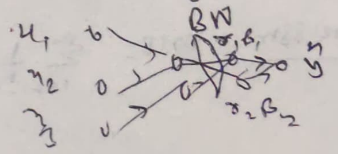
• best_model

① Grid Search
② Random " "
③ Bayesian

• why momentum including in Batch Normalization?

→ जो 2 या 3 HL मे Normalize करते है ताकी ज्यादा overfit न हो

Inference ; time के time
(testing time)



• during testing time क्या होगा γ, β

⊙ और 1 row data to predict in test time
↳ Not in N_1, N_2 → it
On N_i', N_2' = ?
$N_1' = \frac{N_1 - \mu_B}{\sigma}$
not during in testing time
↓
moving mean ✓
moving variance
↑ USE

→ Tuning
pip install keras-tuner -q

⊙ why do flatten?
↳ In ANN → Not pass as 2D so convert into 1D !
but image case use CNN

⊙ How we define no of layers

→ custom class → create

8th Sep

Loss

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

$w \rightarrow$ How to Solve by $\rightarrow \frac{\partial L}{\partial w}$

Chain Rule

In Classification :-

y	$\hat{y}$
0	0.2
1	0.1
0	0.8
1	0.2
0	0.9

than How to find diff error

because y in 0,1 & $\hat{y}$ is range of 0 to 1

Use LogLoss (Mutually exclusive

$$\boxed{\sum_{i=1}^{n} \frac{1}{n}\left(y * \log \hat{y} + (1-y)\log(1-\hat{y}) \right)}$$

दोनो मे कोई एक ही होगा फिर

Categorical crossentropy :

$$\boxed{\sum_{i=1}^{n} \frac{1}{n} \sum_{j=1}^{n} \frac{1}{k}\, y_i \log \hat{y}_{ij}}$$

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

$\hat{y}$

0.2	0.3	0.5

0	1	0

If any goes closer to 1 another automatically goes 0.

if we represent in matrix if all data called sparse matrix

→ Sparse categorical cross

→ Imbalanced Class

use SMOTE / under/over sample

not adding any value importa

→ In Decision Tree class (class_weight)

• Then How to assign weight

$(5\, y \log(\hat{y})) + 2(1-y) \log(1-\hat{y})$

weight to give

DT Calculate Split (Gini,

Q In keras.fit → which parameter solve this problem

• Class_weight

• Sample_weight (what is use of it)

↳ sample Any weight ...

Q

Q then why we 0

4 Steps of Analytics (4) Descriptive (a), Diagnostive (b), Predictive (c), Prescriptive (d)

~~Types of Model~~

Model Explainability (NN):

→ explain how y predicted is coming.

eg:- $\boxed{\text{Model}} \rightarrow 0.3$ Prob

↓ rejected → How reject no need but need why rejected.

why explainability?

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

- Why is Loan is rejected or Accepted so we can

↳ Explain USP

↓

How

Using some Model

- Which feature is more pushing the model to +ve side or which feature is pushing to the -ve side

+ve f_1, f_2 → 0.3 ← f_3, f_4 -V

<.5

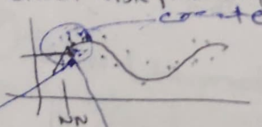
So here equilibrium is 0.3 so reject because by Criteria is 0.5.

1 LIME :- Local interpretable model agnostic Explainability

- LIME → for all model type data
- Local interpretable → also explain for single data
- model agnostic → all type of model

→ lime package Output

eg Credit risk prediction



NN

create simple linear Eq of these Local part so find $B_1, B_2, B_3 \dots f$

↳ to $\hat{y}$

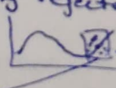
(its goes to -ve side so thats why reject here.

↳ Pravin is rejected. the Pravin ask me why reject my loan Explain

↳ So we here need

Step → model.predict(xtest) → $\hat{y}$ → Accept / Reject ✓

then why rejected



→ add some synthetic data near praveen, and create model

from Pred $\boxed{} = \hat{y}$ via LIME

- to explain the case why rejected.

2 Shaphly

is used to explain the model how it works.

In single row of data also explain