Powered by:

NPTEL Week 5 Live Sessions

on Deep Learning (noc24_ee04)

A course offered by: Prof. Prabir Kumar Biswas, IIT Kharagpur

- Python coding: SVM, KNN
- Week 4 quiz solution (Artificial Neural Nets)
- Week 5 practice questions (Back propagation)

By

Arka Roy
NPTEL PMRF TA

Prime Minister’s Research Fellow
Department of Electrical Engineering, IIT Patna
Web: https://sites.google.com/view/arka-roy/home

FREE ONLINE EDUCATION swayam
शिक्षित भारत, उन्नत भारत

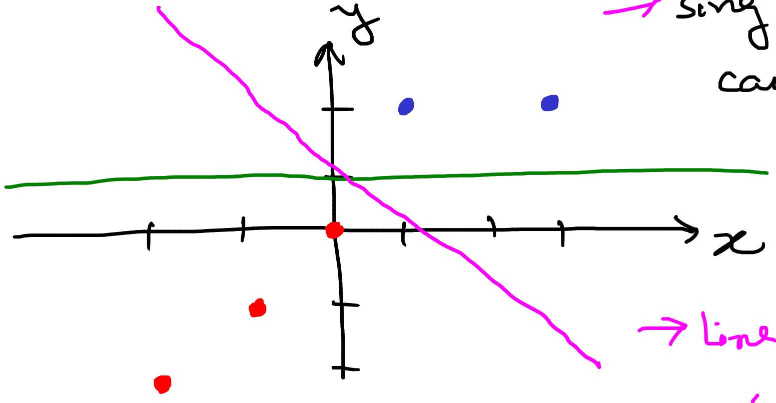
PMRF
Prime Minister’s Research Fellows
Ministry of Education
Government of India
सत्यमेव जयते

Content of the live session

1. Quiz solution of week 4
2. Solving numerical problems from week 5
3. Python coding for week-3 content (SVM, KNN)

Let **X** and **Y** be two features to discriminate between two classes. The values and class labels of the features are given here under. The minimum number of neuron-layers required to design the neural network classifier

i) If the problem is linearly separable → → single neuron we can get the classification



X	Y	#Class
1	2	Class-II
0	0	Class-I
-2	-2	Class-I
3	2	Class-II
-1	-1	Class-I

→ linearly separable classification problem

a. 2
b. 1
c. 5
d. 4

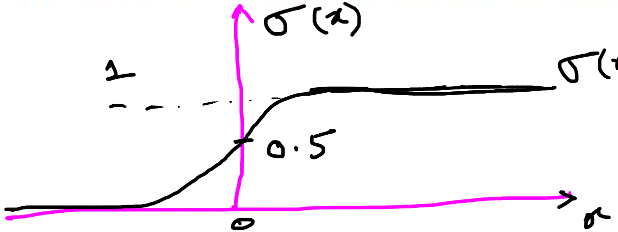
Which among the following options give the range for a logistic function?

a. -1 to 1

b. -1 to 0

c. 0 to 1

d. 0 to infinity



$\sigma(x) \in (0,1)$

Input to SoftMax activation function is [5,3,4]. What will be the output?

$$\frac{e^{v_i}}{\sum_{v_i} e^{v_i}}$$

a. [0.58,0.11, 0.31]

b. [0.43,0.24, 0.33]

c. [0.60,0.10,0.30]

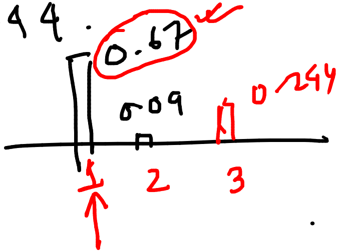
d. [0.67, 0.09,0.24]

$$v = [5,3,4] \xrightarrow{\text{Softmax}(\cdot)} O = [O_1, O_2, O_3]$$

$$O_1 = \frac{e^5}{e^5+e^3+e^4}; \quad O_2 = \frac{e^3}{e^5+e^3+e^4}; \quad O_3 = \frac{e^4}{e^5+e^3+e^4}$$

$$O_1 = 0.665, \quad O_2 = 0.090; \quad O_3 = 0.244.$$

$\simeq 0.67$



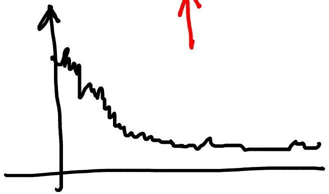
Which of the following options is true?

a. In Full Batch Gradient Descent, a small batch of sample is selected randomly instead of the whole data set for each iteration.

b. In Batch Gradient Descent, the whole data set is processed together for update in each iteration.

c. Batch Gradient Descent considers only one sample for updates and has noisier updates.

d. Batch Gradient Descent produces noisier updates than Stochastic Gradient Descent



Choose the correct option:

i) Inability of a model to obtain sufficiently low training error is termed as ~~Overfitting~~

ii) Inability of a model to reduce large margin between training and testing error is termed as Overfitting ✓

iii) Inability of a model to obtain sufficiently low training error is termed as Underfitting ✓

iv) Inability of a model to reduce large margin between training and testing error is termed as ~~Underfitting~~

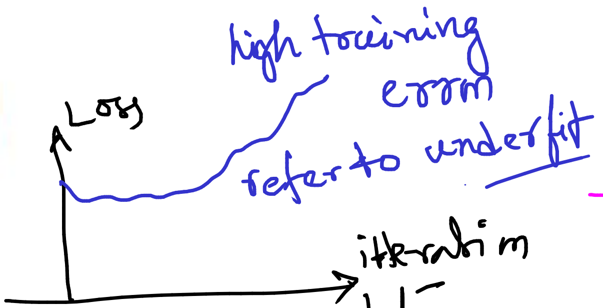
— Training
— Testing.

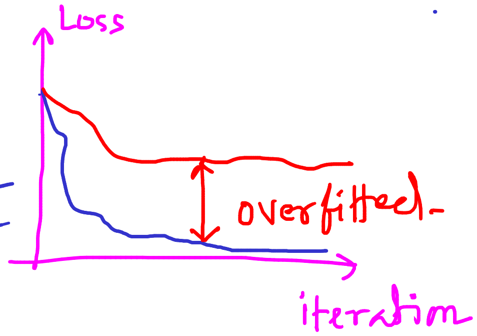
a. Only option (i) is correct

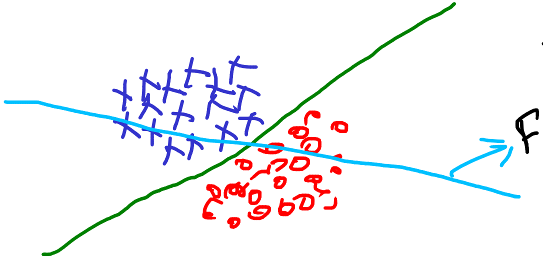
b. Both Options (ii) and (iii) are correct

c. Both Options (ii) and (iv) are correct

d. Only option (iv) is correct







Training dataset.

From the training data model is not able to capture any significant information to classify the data

or

The model is not able to classify well in training phase itself.

Choose the correct options about the assumptions are generally made during optimization in machine learning.

i) Data samples in each data set are dependent on each other ✗

ii) Each data samples present in the training and test set are independent of each other

iii) Training set and test set are overlapping to each other ✗

iv) The distributions of training set and test set are assumed to be identical

a. Only option (i) is correct

b. Both Options (ii) and (iii) are correct

c. Both Options (ii) and (iv) are correct

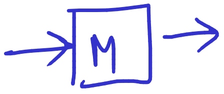
d. Only option (iv) is correct

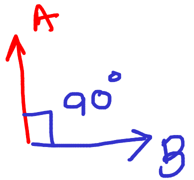
$\vec{A} = \lambda \vec{B} = 2\vec{B}$

A.

B.

(Colinear, dependent).



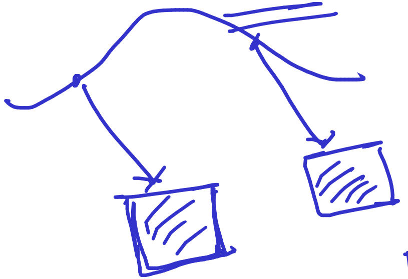


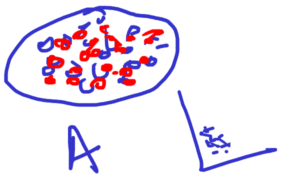
$\vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}| \cos 90° = 0$

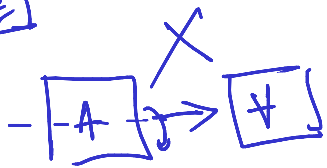
$\vec{A} \cdot \vec{B} = 0 \rightarrow$ orthogonal

$\rightarrow$ independent of each other

Truck.







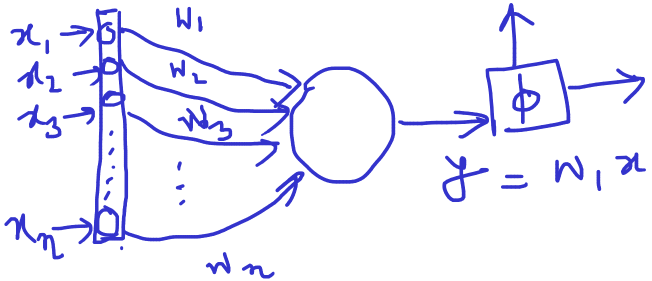
o → Testing Data

o → Training Data

An artificial neuron receives n inputs $x_1, x_2, x_3, \ldots\ldots x_n$ with weights $w_1, w_2, w_3, \ldots\ldots w_n$ attached to the input links. The weighted sum ____ $\sum_i w_i x_i$ ____ is computed to be passed on to a non-linear filter Φ called activation function to release the output. Fill in the blanks by choosing one option from the following.

a. $\sum_i w_i$
b. $\sum_i x_i$
c. $\sum_i w_i + \sum_i x_i$
d. $\sum_i w_i x_i$ ✓

Relu, Sigmoid



$$\phi(y) = \underbrace{\phi\left(\sum_i w_i x_i\right)}$$

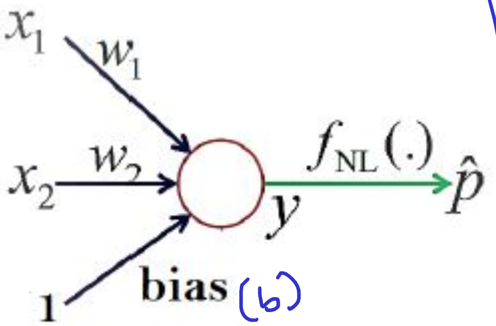
$$y = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

$$y = \sum_{\forall i} w_i x_i$$

$$= \sum_{i=1}^{n} w_i x_i$$

Let us assume that we implement an AND function using a single neuron as shown below. The activation function $f_{NL}(\cdot)$, of our neuron is denoted as: f(y)=0, for y<30, f(y)=1 for y >=30. What would be a possible combination of the weights and bias?

x_1	x_2	p
0	0	0
0	1	0
1	0	0
1	1	1



$y = w_1 x_1 + w_2 x_2 + b$

$\hat{p} = f_{NL}(y)$.

a. Bias = 5, w1 = 5, w2 = 25
b. Bias = 10, w1 = 5, w2 = 5
c. Bias = 10, w1 = 15, w2 = 15
d. Bias = 5, w1 = 10, w2 = 10

$$f_{NL}(y) = \begin{cases} 0 ; & y < 30 \\ 1 ; & y \geq 30. \end{cases}$$

option A :→ $w_1 = 5$, $w_2 = 25$, $b = 5$

$y = 0 + 0 + 5 = 5 \xrightarrow{f_{NL}(\cdot)} 0$

$y = 0 + (1 \times 25) + 5 = 30 \xrightarrow{f_{NL}(\cdot)} 1$.

option B :- $w_1 = 5$, $w_2 = 5$, $b = 10$

$y = 0 + 0 + 10 \xrightarrow{f(\cdot)} 0$

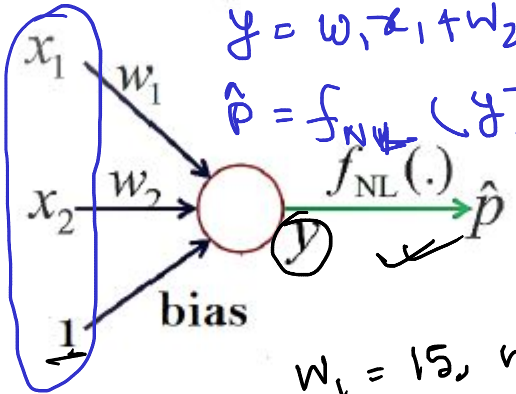
$y = 0 + 5 + 10 = 15 \xrightarrow{f(\cdot)} 0$

$y = 5 + 0 + 10 = 15 \xrightarrow{f(\cdot)} 0$.

$y = 5 + 5 + 10 = 20 \xrightarrow{f(\cdot)} 0$

Let us assume that we implement an AND function using a single neuron as shown below. The activation function $f_{NL}(\cdot)$, of our neuron is denoted as: f(y)=0, for y<30, f(y)=1 for y >=30. What would be a possible combination of the weights and bias?

x_1	x_2	p
0	0	0
0	1	0
1	0	0
1	1	1



$y = w_1 x_1 + w_2 x_2 + b$

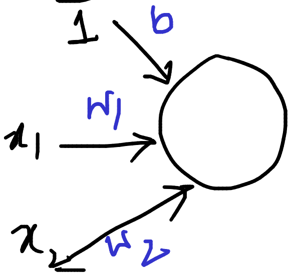
$\hat{p} = f_{NL}(y)$.

$w^t x = [w_1\ w_2\ b] \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$

$\begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix}$

$y = w_1 x_1 + w_2 x_2 + b$.

$w_1 = 15,\ w_2 = 15,\ b = 10$.



$\rightarrow \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$

a. Bias = 5, w1 = 5, w2 = 25
b. Bias = 10, w1 = 5, w2 = 5
c. Bias = 10, w1 = 15, w2 = 15 ✓
d. Bias = 5, w1 = 10, w2 = 10

$y = 0 + 0 + 10 \xrightarrow{f(\cdot)} 0$.

$y = 0 + 15 + 10 = 25 \xrightarrow{f(\cdot)} 0$.

$y = 15 + 0 + 10 = 25 \xrightarrow{f(\cdot)} 0$.

$y = 15 + 15 + 10 = 40 \xrightarrow{f(\cdot)} 1$

$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \quad w = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$

$w^t x = [b\ w_1\ w_2] \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$

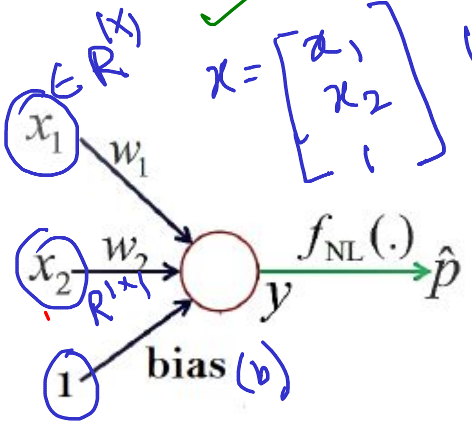
$= b + w_1 x_1 + w_2 x_2$

$$f_{NL}(y) = \begin{cases} 0 \ ; & y < 30 \\ 1 \ ; & \underbrace{y \geq 30}. \end{cases}$$

Consider the below neural network. $\hat{p}$ is the output after applying the non-linearity function $f_{NL}(\cdot)$ on y. The non-linearity $f_{NL}(\cdot)$ is given as a step function i.e.,

$$f(v) = \begin{cases} 0, & if\ v < 0 \\ 1, & if\ v \geq 0 \end{cases}$$

Choose the correct set of weights w_1, w_2 and bias for which the network behaves as an OR function.



$x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$ $W = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix}$

$y = W^t x = [W_1 \ W_2 \ b] \cdot \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$

$y = W_1 x_1 + W_2 x_2 + b.$

class

sample

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

2

$y = W_1 x_1 + W_2 x_2 + b.$

a. $w_1 = 1, w_2 = 1.5,\ bias = 1$
b. $w_1 = 1, w_2 = 0.5,\ bias = -1$
c. $w_1 = 1, w_2 = 1.5,\ bias = -1$
d. $w_1 = 1, w_2 = -0.5,\ bias = 1$

option (A) :→ $W_1 = 1, W_2 = 1.5, b = 1$

$y = [1 \ 1.5 \ 1] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 + 0 + 1 = 1 \xrightarrow{f(\cdot)} 1$ ✗

option (B) :→

$y = 0 + 0 - 1 = -1 \xrightarrow{f(\cdot)} 0.$

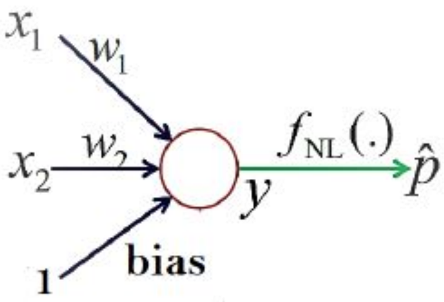
$y = 0 + 0.5 - 1 = -0.5 \xrightarrow{f(\cdot)} 0.$ ✗.

option (C) :→ $y = 0 + 0 + (-1) \xrightarrow{f(\cdot)} 0.$ ✓

Consider the below neural network. $\hat{p}$ is the output after applying the non-linearity function $f_{NL}(\cdot)$ on y. The non-linearity $f_{NL}(\cdot)$ is given as a step function i.e.,

$$f(v) = \begin{cases} 0, & if\ v < 0 \\ 1, & if\ v \geq 0 \end{cases}$$

Choose the correct set of weights w_1, w_2 and bias for which the network behaves as an OR function.



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

option c : $\rightarrow y = 0 + 0 + (-1) \rightarrow 0.$

$y = 0 + 1.5 - 1 = 0.5 \rightarrow 1.$

$y = 1 + 0 - 1 = 0 \xrightarrow{f(\cdot)} 1.$

$y = 1 + 1.5 - 1 = 1.5 \xrightarrow{f(\cdot)} 1.$

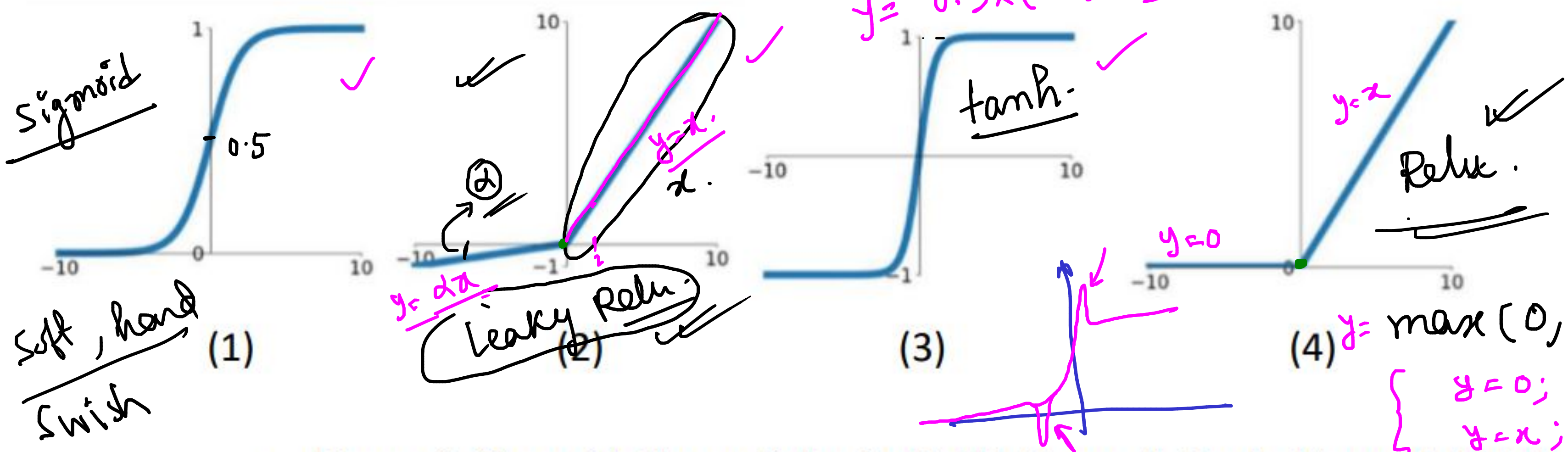
a. $w_1 = 1, w_2 = 1.5,\ bias = 1$
b. $w_1 = 1, w_2 = 0.5,\ bias = -1$
c. $w_1 = 1, w_2 = 1.5,\ bias = -1$
d. $w_1 = 1, w_2 = -0.5,\ bias = 1$

$y = \alpha x$ $\quad$ ~~$y=0$~~ $\quad$ $\alpha \in [0,1]$ $\quad$ $\alpha = 0.3$

Look at the following figures. Can you identify which of the following options correctly identify the activation functions ?

$y = \alpha \cdot x$

$y = 0.3 \times (-10) = -3$

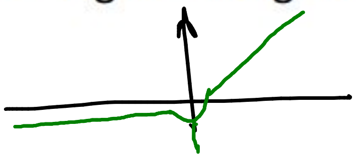


$y = \max(0, x)$.

$\begin{cases} y = 0; & x < 0 \\ y = x; & x > 0 \end{cases}$

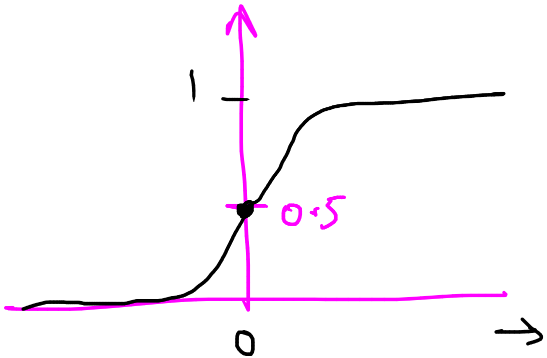
Gelu
Selu
elu

a. Figure 1: Sigmoid, Figure 2: Leaky ReLU, Figure 3: Tanh, Figure 4: ReLU
b. Figure 4: Sigmoid, Figure 3: Leaky ReLU, Figure 2: Tanh, Figure 1: ReLU
c. Figure 2: Sigmoid, Figure 3: Leaky ReLU, Figure 4: Tanh, Figure 1: ReLU
d. Figure 3: Sigmoid, Figure 2: Leaky ReLU, Figure 1: Tanh, Figure 4: ReLU



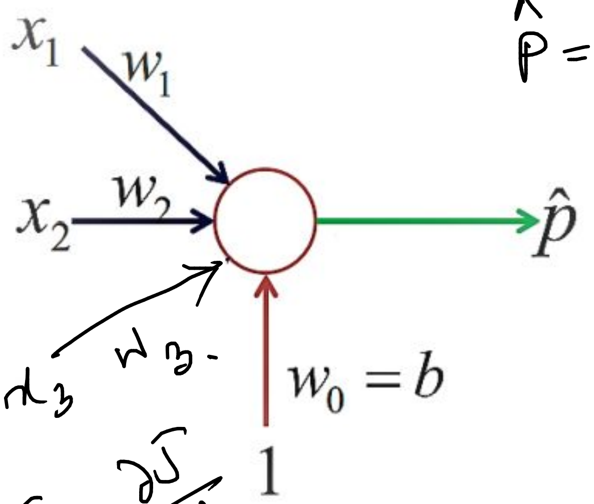
What is the output of sigmoid function for an input with dynamic range $[0, \infty]$?

a. $[0, 1]$
b. $[-1, 1]$
c. $[0.5, 1]$
d. $[0.25, 1]$



$[0.5, 1]$.

Find the gradient component $\frac{\partial J}{\partial w_1}$ for the network shown below if $J(\cdot) = 0.5(\hat{p} - p)^2$ is the loss function, p is the target?



$$\hat{p} = x_1 w_1 + w_2 x_2 + 1 \times b = x_1 w_1 + x_2 w_2 + b.$$

$$J(\cdot) = \frac{1}{2}(\hat{p} - p)^2 \qquad \hat{p} = f(w_1, w_2, b)$$

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial \hat{p}} \cdot \frac{\partial \hat{p}}{\partial w_1}$$

$$= \frac{1}{2} \times 2 \times (\hat{p} - p) \cdot [x_1 + 0]$$

$$= \underline{x_1 (\hat{p} - p)}.$$

$\frac{\partial J}{\partial w_2}$, $\frac{\partial J}{\partial w_3}$

a. $2\hat{p} \times x_1$
b. $2(\hat{p} - p) \times x_1$
c. $(\hat{p} - p) \times x_1$ ✓
d. $2(1 - p) \times x_1$

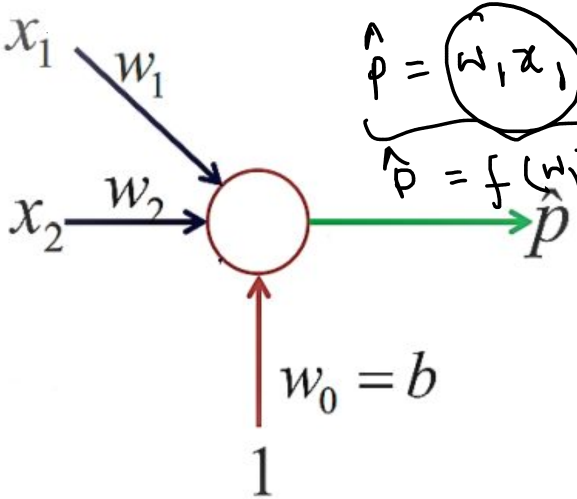
Find the gradient component $\frac{\partial J}{\partial w_1}$ for the network shown below if $J(\cdot) = 0.5(\hat{p} - p)^2$ is the loss function, p is the target?

$J(\hat{p}) = \frac{1}{2}\underbrace{(\hat{p} - p)^2}$. $J(\hat{p})$

$\hat{p} = \underbrace{w_1 x_1 + w_2 x_2 + b}$.

$\hat{p} = f(w_1)$

$\frac{\partial J}{\partial w_1} = \underbrace{\frac{\partial J}{\partial \hat{p}} \cdot \frac{\partial \hat{p}}{\partial w_1}} = \frac{\partial J}{\partial w_1}$



a. $2\hat{p} \times x_1$
b. $2(\hat{p} - p) \times x_1$
c. $(\hat{p} - p) \times x_1$
d. $2(1 - p) \times x_1$

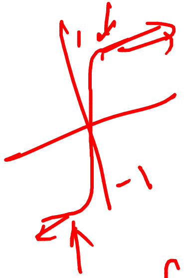
Which of the following are potential benefits of using ReLU activation over sigmoid activation?

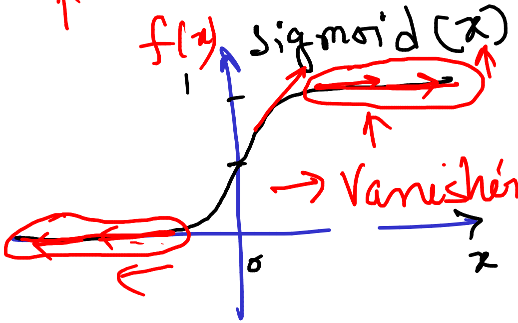
a. ReLu helps in creating ~~dense~~ (most of the neurons are active) representations

b. ReLu helps in creating sparse (most of the neurons are non-active) representations

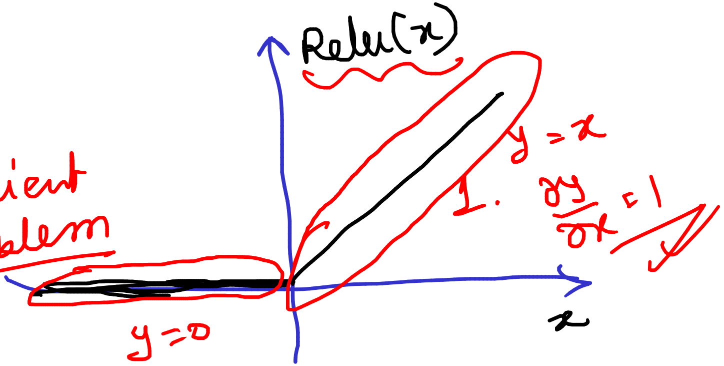
c. ReLu helps in mitigating vanishing gradient effect

d. Both (b) and (c)





→ Vanishing Gradient problem



$\frac{\partial y}{\partial x} = 1$

$\frac{\partial y}{\partial x} = 0$

$f = \sigma(w_1 x_1)$

$\frac{\partial f}{\partial w_1} = f(1-f) \cdot x_1$

Activation function should always be differential

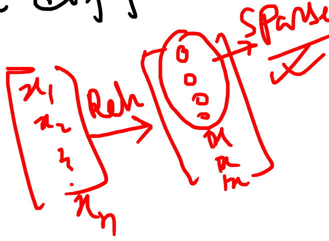
$W(n+1) \leftarrow W(n) - \eta \frac{\partial L}{\partial W}$

$w(n+1) \leftarrow W(n) - \eta f(1-f) x_1$

$W(n+1) = W(n)$

Sig(x) → Real output.

Relu(x) → 0 / x.



Which of the following are potential benefits of using ReLU activation over sigmoid activation?

a. ReLu helps in creating dense (most of the neurons are active) representations

b. ReLu helps in creating sparse (most of the neurons are non-active) representations

c. ReLu helps in mitigating vanishing gradient effect

d. Both (b) and (c)

$$f(w) = \underbrace{\sigma(wx)}$$

$$\frac{\partial f(w)}{\partial w} = \underbrace{\sigma(\cdot)(1-\sigma(\cdot))}x$$

$$\frac{\partial f(w)}{\partial w} = \underbrace{f(1-f)}_{\text{gradient activation}}x.$$

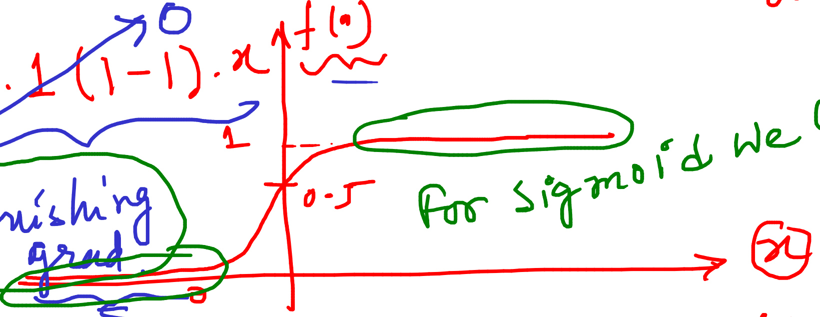
$$w(n+1) \leftarrow w(n) - \eta \frac{\partial f(w)}{\partial w}.$$

$$w(n+1) \leftarrow w(n) - \eta f(1-f)x.$$

CASE 1: → $w(n+1) \leftarrow w(n) - \eta . \underbrace{\cancel{1(1-1)}}_{\to 0} . x$

$w(n+1) = w(n)$ → Vanishing grad

CASE 2: → $x \downarrow\downarrow$ $f(x) \to 0$



For sigmoid we can observe vanishing gradient

$x \uparrow\uparrow$.

$w(n+1) \leftarrow w(n) - \underbrace{\cancel{\eta . 0 (1-0)}}_{\to 0} . x.$

$w(n+1) = w(n)$

Which of the following are potential benefits of using ReLU activation over sigmoid activation?

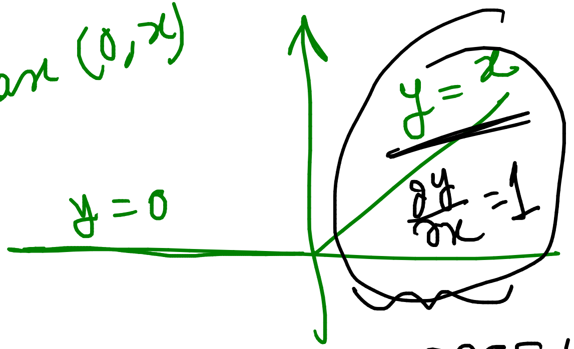
a. ReLu helps in creating dense (most of the neurons are active) representations

b. ReLu helps in creating sparse (most of the neurons are non-active) representations

c. ReLu helps in mitigating vanishing gradient effect

d. Both (b) and (c)

$y = \max(0, x)$



$f = \text{Relu}(wx)$

$\frac{\partial f}{\partial w} = \text{gradient of Relu} \cdot \frac{\partial (wx)}{\partial w}$

$= \text{gradient of Relu} \times x.$

$\frac{\partial f}{\partial w} = \text{gradient}$

CASE 1: $\rightarrow x \uparrow\uparrow$

$\frac{\partial f}{\partial w} = 1 \cdot x.$

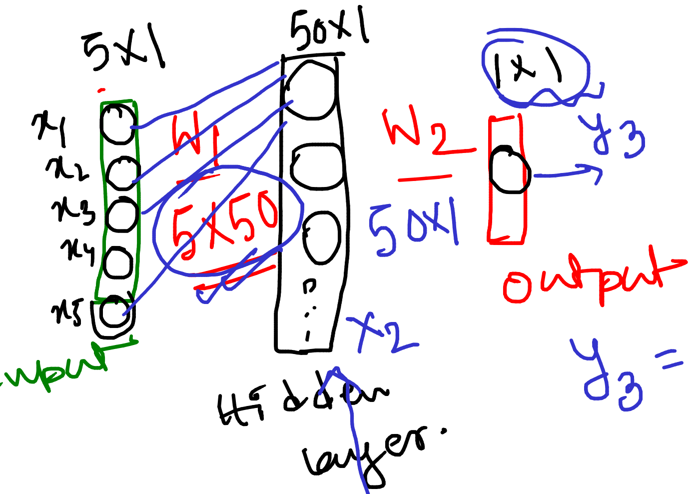
$w(n+1) \leftarrow w(n) - \eta \cdot \frac{\partial f}{\partial w}$

$\leftarrow w(n) - \underbrace{\eta \cdot 1 \cdot x}_{\text{Grad does not vanish}}.$

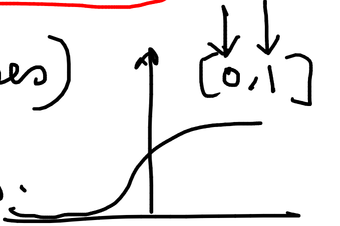
Suppose a fully-connected neural network has a single hidden layer with 50 nodes. The input is represented by a 5D feature vector and we have a binary classification problem. Calculate the total number of parameters of the network. Consider there are NO bias nodes in the network.

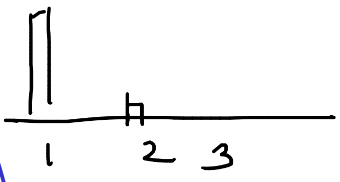
$X \in R^{5\times1}$

$y \in R^{50\times1}$



(Weights and ~~biases~~) Parameters.





$W_1 \in R^{5\times50}$

$W_2 \in R^{50\times1}$

Parameter $W_1 = 250$

$W_2 = 50$

Total = 300

$y_3 = W_2^t x_2$

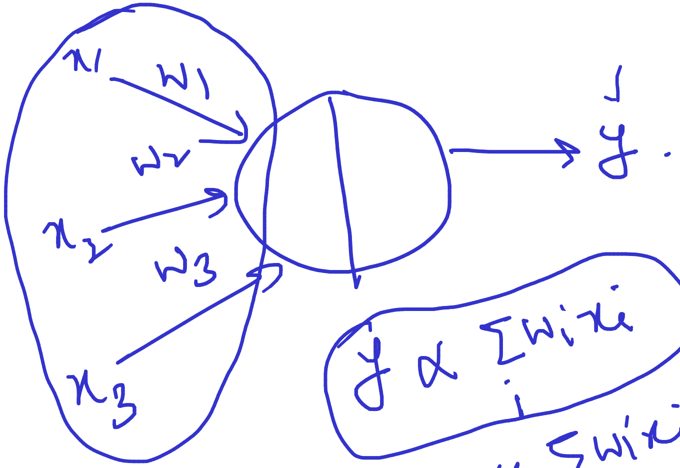
$= [\;]_{1\times50} \times [\;]_{50\times1} = 1\times1$

$y = [W]_{50\times5} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_5 \end{bmatrix}_{5\times1}$

50×1

A 3-input neuron has weights 1.5, 0.5, 0.5. The transfer function is linear, with the constant of proportionality being equal to 2. The inputs are 6, 20, 4 respectively. The output will be:

a. 40
b. 42 ✓
c. 32
d. 12



$$\frac{y}{\sum_i w_i x_i} = 2$$

$$y \propto \sum_i w_i x_i$$

$$y = K \sum_i w_i x_i$$

$$y = 2 \sum w_i x_i$$

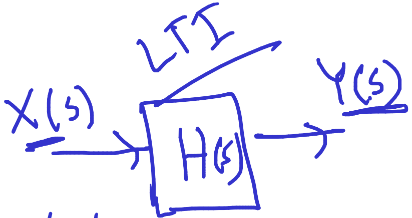
$$= 2(w_1x_1 + w_2x_2 + w_3x_3)$$

$$= 2(1.5\times 6 + 20\times 0.5 + 4\times 0.5)$$

$$= 2(9 + 10 + 2)$$

$$= 2\times 21$$

$$= 42$$



Transfer function $H(s) = \frac{Y(s)}{X(s)}$

Considering all initial condition = 0

You want to build a 5-class neural network classifier, given a leaf image, you want to classify which of the 5 leaf breeds it belongs to. Which among the 4 options would be an appropriate loss function to use for this task?

a. Cross Entropy Loss
b. MSE Loss
c. SSIM Loss
d. None of the above

Loss function

$(y \log o + (1-y) \log(1-o))$

$\frac{1}{N}\sum_i (\quad)$

Multiclass categorocial cross entropy

Binary classification

→ Binary cross entropy.

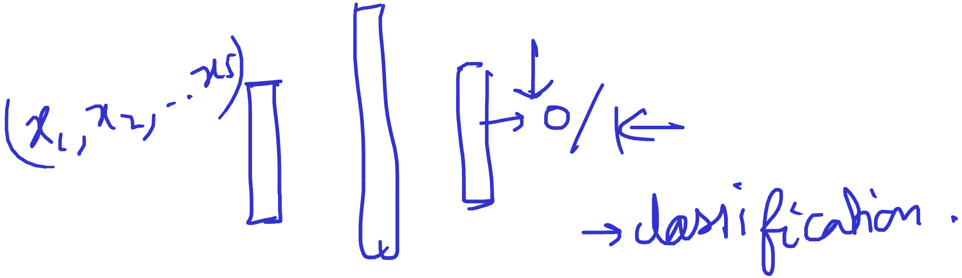
Multi class classification

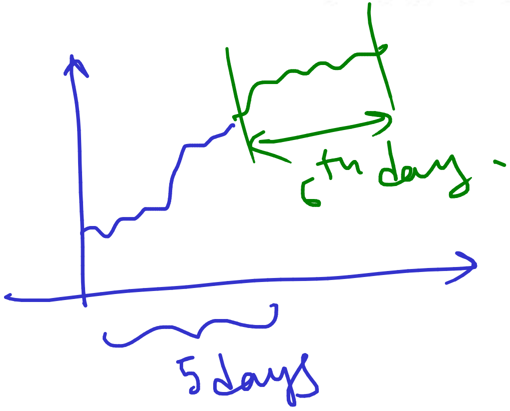
→ Multi class categorical cross loss function.

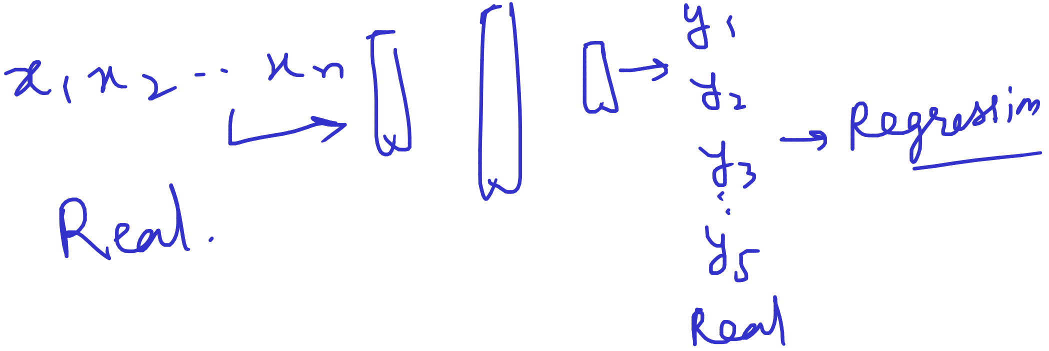
Regression application

You want to build a 5-class neural network classifier, given a leaf image, you want to classify which of the 5 leaf breeds it belongs to. Which among the 4 options would be an appropriate loss function to use for this task?

a. Cross Entropy Loss
b. MSE Loss
c. SSIM Loss
d. None of the above

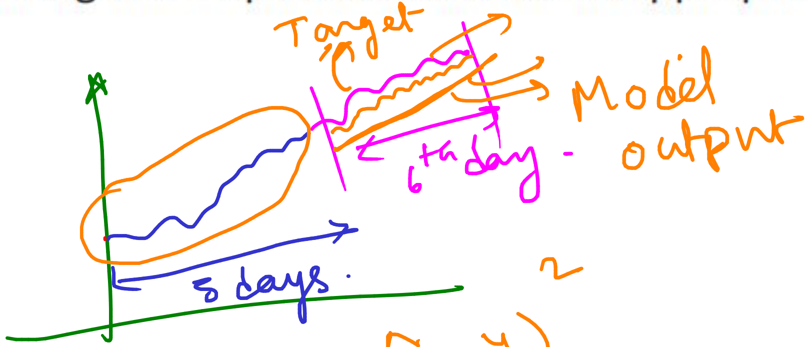




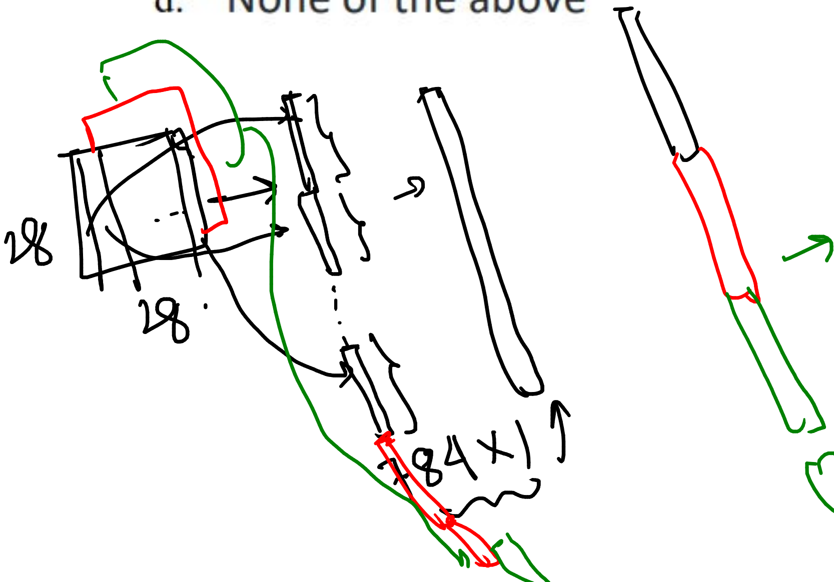


You want to build a 5-class neural network classifier, given a leaf image, you want to classify which of the 5 leaf breeds it belongs to. Which among the 4 options would be an appropriate loss function to use for this task?

a. Cross Entropy Loss
b. MSE Loss
c. SSIM Loss
d. None of the above



$$\frac{1}{N}\sum_i (\hat{y}_i - y_i)^2$$



$\rightarrow R^{3072 \times 1}$

$(32 \times 32 \times 3) = 3072$

5000

$\underbrace{5000 \times 3072 \times 1}$

You want to build a 5-class neural network classifier, given a leaf image, you want to classify which of the 5 leaf breeds it belongs to. Which among the 4 options would be an appropriate loss function to use for this task?

a. Cross Entropy Loss
b. MSE Loss
c. SSIM Loss
d. None of the above

KNN implementation:-

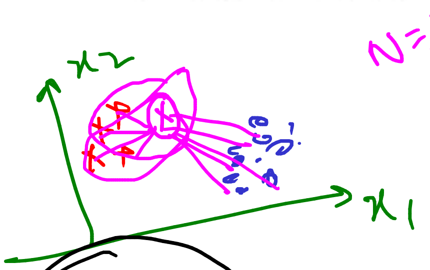
(1):→

$[d_1, d_2, \ldots, d_{100}] \xrightarrow{\text{Sort}} [d_{50}, d_{40}, \ldots, d_{10}]$

$d_{50} \to$ Blue

$d_{40} \to$ Red

N=3



$M = \frac{2}{\|W\|}$ learning the weight.

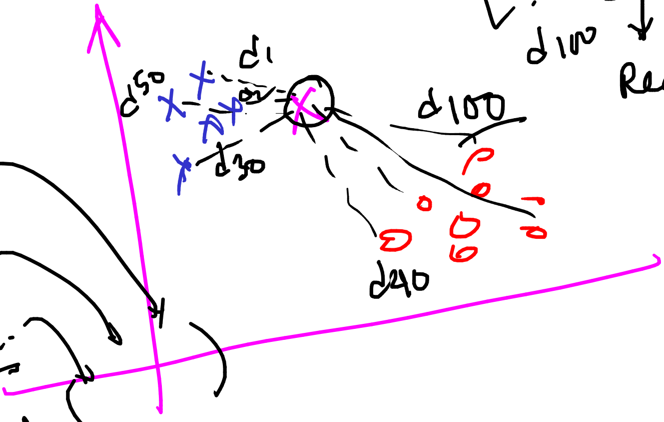
K=3

$d_{50} \to$ Blue

$d_{40} \to$ Red

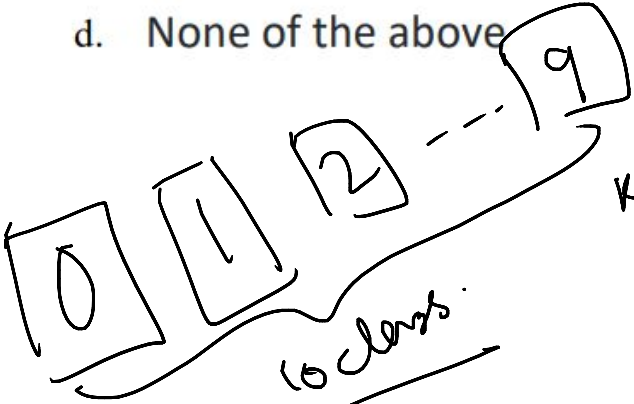
$d_{30} \to$ Blue.

mode( ) → Blue.



You want to build a 5-class neural network classifier, given a leaf image, you want to classify which of the 5 leaf breeds it belongs to. Which among the 4 options would be an appropriate loss function to use for this task?

a. Cross Entropy Loss
b. MSE Loss
c. SSIM Loss
d. None of the above



10 class.

K no of classes = 10

D feature = 784

W →

50×784

X

Y

[0, 1, ... 9]