Deep Learning

① Vanishing Gradient Problem ✓
② Activations Functions ✓
③ Loss functions or Cost function ✓

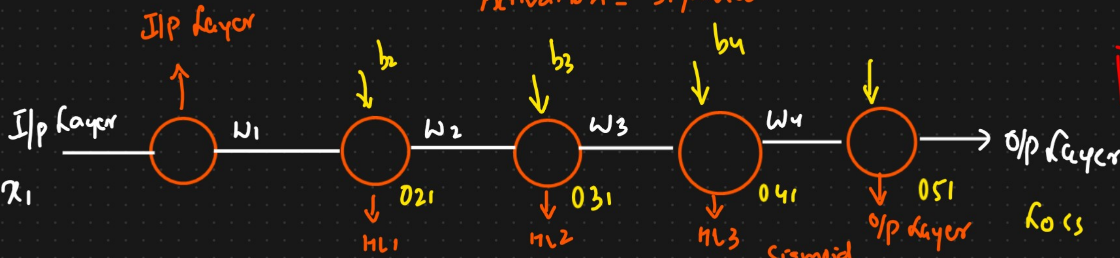
Optimizers → ANN.

inblog.in

Medium

④ Vanishing Gradient Problem

Activation = Sigmoid



ineuron, KRISH

⇓

{ Connection Request }

$$W_{1new} = W_{1old} - \eta \frac{\partial L}{\partial W_{1old}}$$

$$O_{51} = \sigma([O_{41} * w_4] + b_5)$$

Sigmoid ↑ → Sigmoid

$$\text{Sigmoid} = \frac{1}{1+e^{-z}}$$

⇓ Chain Rule of Derivatives.

$$z = (w^T x + b)$$

$$\frac{\partial L}{\partial W_{1old}} = \frac{\partial L}{\partial O_{51}} * \frac{\partial O_{51}}{\partial O_{41}} * \frac{\partial O_{41}}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial W_{1old}}$$

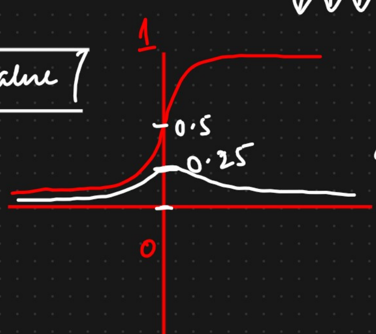
val * 0.25 * 0.10 * 0.05 * 0.001 * 0.005

⇓ ⇓ ⇓ ⇓

⇓

Smaller value

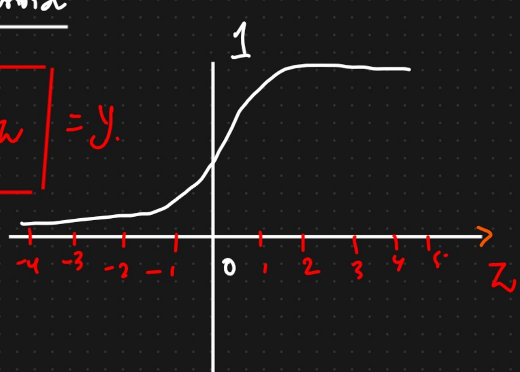
$$\text{Sigmoid} = \frac{1}{1+e^{-z}}$$



$$\frac{d\,\sigma(z)}{dz} = 0 \text{ to } 0.25$$

Sigmoid

$$\frac{1}{1+e^{-z}} = y.$$

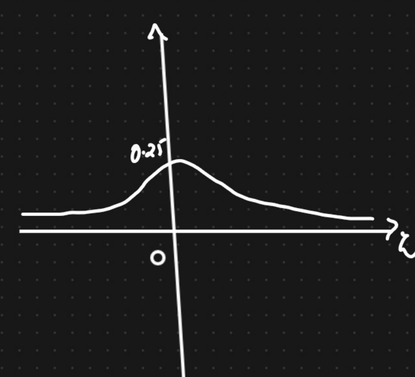


$$z = (w^T x + b)$$

$$y = \sigma(z)$$

$$y = \sigma(w^T x + b)$$

Derivative of Sigmoid



① Weights will not get updated

$\eta := 0.001$

$$W_{1new} = W_{1old} - \eta\left(\frac{\partial L}{\partial W_{1old}}\right) \Rightarrow \text{Small}$$

$$= W_{1old} - (\text{Small})\ (\text{Small})\ \ (0.001) * (0.000005)$$

$W_{new} \approx W_{old} \Rightarrow$ **Vanishing GRADIENT PROBLEM**

↓

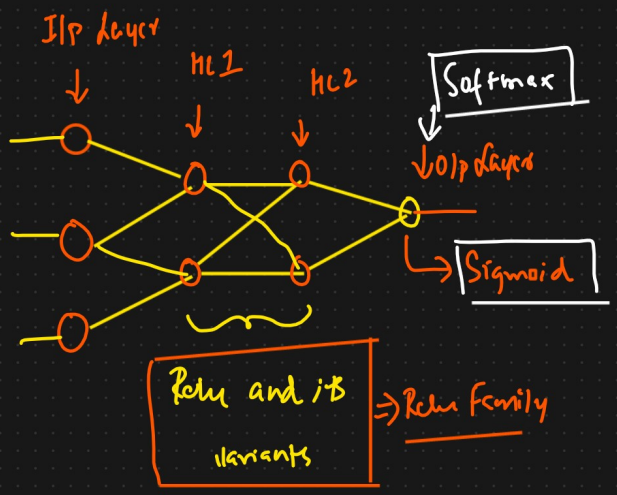
[Weights are not getting updated]

② To fix this problem we change the Activation function

[Which Activation function To use when]

[Multiclass classification]

Binary classification



Relu and its variants ⇒ Relu Family

$\max(0, x)$ X

① 0 or 1 ⇓ Sigmoid

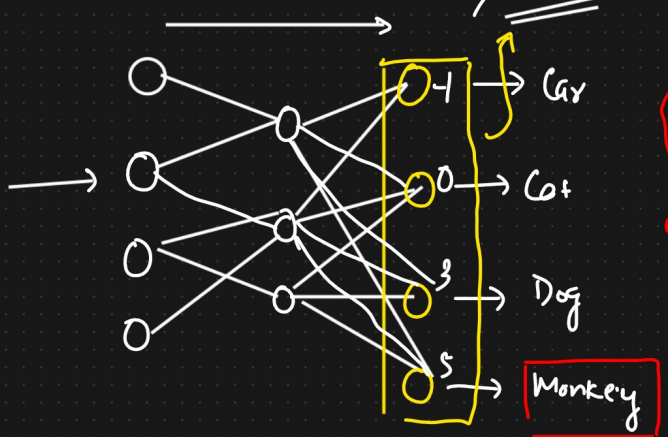
Vanishing Gradient problem

↑ ↑

X Sigmoid or Tanh

Softmax

$= \frac{1}{1+e^{-y}} = 0 \text{ or } 1$



+1

⇒ 0.002

⇒ 1 ⇒ 0.005

⇒ 0.118

⇒ 0.874

$$\text{Softmax} = \left\{ \frac{e^{y_i}}{\sum_{k=0}^{n} e^{y_k}} \right\}$$

$$= \frac{e^{-1}}{\left[e^{-1} + e^{+0} + e^{+3} + e^{5}\right]} = \frac{0.36}{169.87} =$$

$$\text{Car} = \frac{e^{-1}}{\left[e^{-1} + e^{0} + e^{3} + e^{5}\right]} = \frac{0.36}{169.87} = \boxed{0.002} \Rightarrow \text{probability}$$

$$\text{Cat} = \frac{e^{0}}{\left[e^{-1} + e^{0} + e^{3} + e^{5}\right]} = 0.006$$

$$\text{Dog} = \frac{e^{3}}{169.87} = 0.118$$

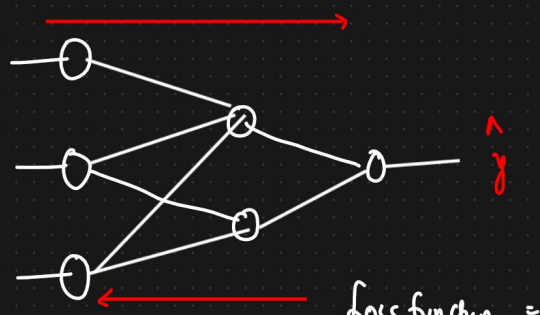
$$\text{Monkey} = \frac{e^{5}}{169.87} = 0.874$$

Linear Regression ✓

Sigmoid, Softmax } [output layer]

④ Loss functions And Cost functions

① Regression
② Classification



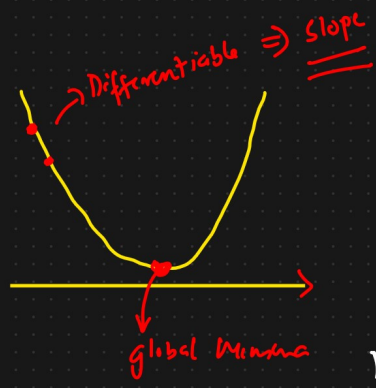
Loss function $= \frac{1}{n}(y - \hat{y})^2$

Cost function $= \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$

① Mean Squared Error (MSE)

Loss function = $\frac{1}{n}(y-\hat{y})^2$ ⇓ Quadratic Equation

Cost function = $\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$



Advantages

① Differentiable ✓

② It has 1 local or global Minima

③ It converges faster

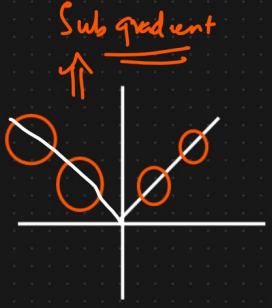
Disadvantages

① Not Robust to outliers

② Mean Absolute Error (MAE).

loss = $\frac{1}{n}|y-\hat{y}|$

Cost fn = $\frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|$



Adv

① Robust to outliers

RMSE ←

③ Huber Loss ←

① MSE ② MAE ⇒ Time an ni outlier

Hyperparameter ⇓

$$loss = \begin{cases} \frac{1}{n}(y-\hat{y})^2 & \text{if } |y-\hat{y}| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}$$

④ Classification

CROSS ENTROPY
- → Binary CROSS ENTROPY → Binary Classification
- → Categorical CROSS ENTROPY → Multiclass classification

① Binary CROSS Entropy [Log loss] → Logistic Regression

$$loss = -y * \log(\hat{y}) - (1-y) * \log(1-\hat{y}) \Rightarrow \text{Logistic Regression}$$

$$loss = \begin{cases} -\log(1-\hat{y}) & \text{if } y=0 \\ -\log(\hat{y}) & \text{if } y=1 \end{cases}$$

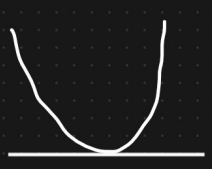
$$\hat{y} = \frac{1}{1+e^{-x}} \Rightarrow \text{Sigmoid} \Rightarrow \hat{y}$$

② Categorical CROSS Entropy {Multi Classification}

↑ Softmax ⇒ $\hat{y}$

				y_{11}		y_{13}	
				j=1	j=2	j=3	
f_1	f_2	f_3	O/P	Good	Bad	Neutral	i = Row
→ 2	3	4	Good	[1]	0	[0]	j = Column
5	6	7	Bad	0	[1]	0	
8	9	10	Neutral	0	0	[1]	

$$L(x_i, y_i) = -\sum_{j=1}^{c} y_{ij} * \ln(\hat{y}_{ij}) \Leftarrow$$



$$y_{ij} = [y_{11}, y_{12}, y_{13} \cdots y_{1j}]$$
$$[y_{21}, y_{22}, y_{24} \cdots y_{2j}]$$

$$y_{ij} = \begin{cases} 1 & \text{if the element is in the class} \\ 0 & \text{otherwise} \end{cases}$$

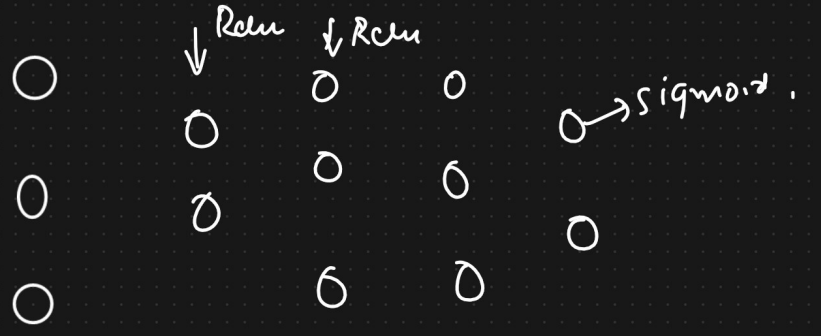
Predicted $\Leftarrow \hat{y}_{ij}$ = Softmax Activation $\Rightarrow$ O/p layer

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \Rightarrow \text{Softmax}$$

Conclusion

- [Relu, Softmax] $\Rightarrow$ Categorical Cross Entropy
- [Relu, Sigmoid] $\Rightarrow$ Binary Cross Entropy.
- [Relu, Linear] $\Rightarrow$ MSE, MAE, Huber Loss, RMSE

Linear $\Rightarrow$ Regression


Relu
Relu
sigmoid.