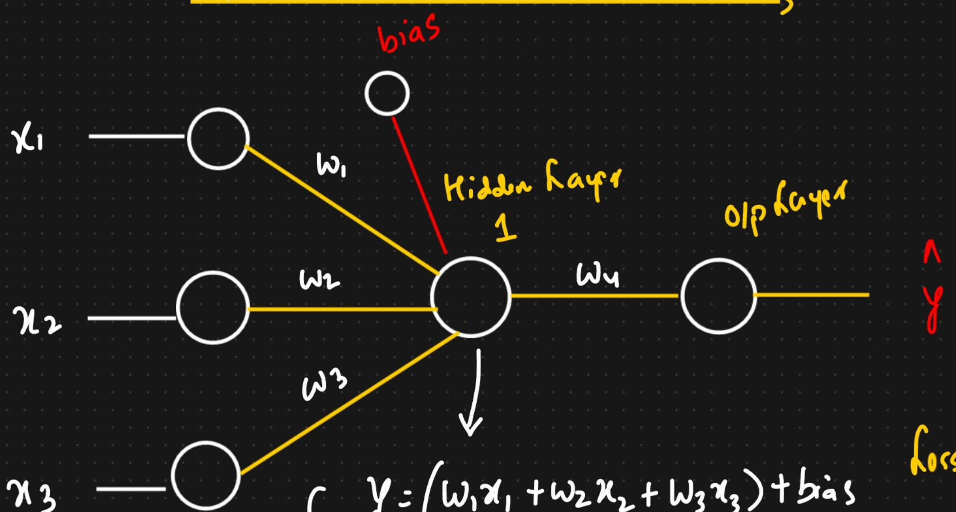
Day 2 - Deep Learning.

Agenda

1. Forward Propogation ✓
2. Chain Rule of Derivative ✓
3. Vanishing Gradient Problem ✓
4. Loss functione ✓

Activation functions



$$\boxed{y = w^T x + b}$$

$$\begin{cases} y = (w_1x_1 + w_2x_2 + w_3x_3) + bias \\ z = \sigma(y) \end{cases}$$

$z = \sigma(y) \rightarrow$ Sigmoid Activations

$\Downarrow$ Non linear problems

$\Downarrow$ 0 or 1

$\geq 0.5 \rightarrow 1$

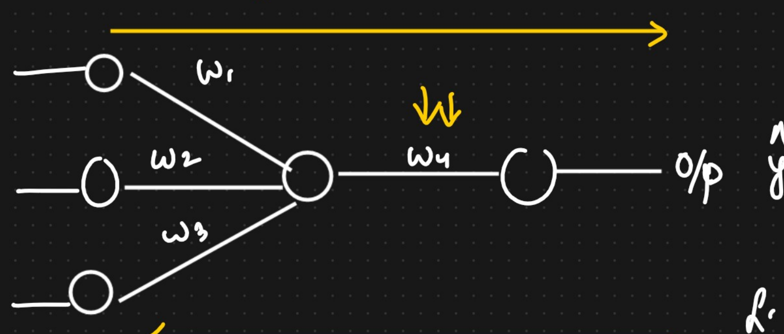
$< 0.5 \rightarrow 0$

Loss $(y - \hat{y}) \downarrow\downarrow$ $\downarrow\downarrow$

$\Rightarrow$ Forward propogation

2. Backpropogation

1. weight updation formula
2. Chain Rule of Diffeentiation ✓



Barckward Propogation

L. $(y - \hat{y}) \downarrow\downarrow$

$\boxed{\eta = 0.01}$

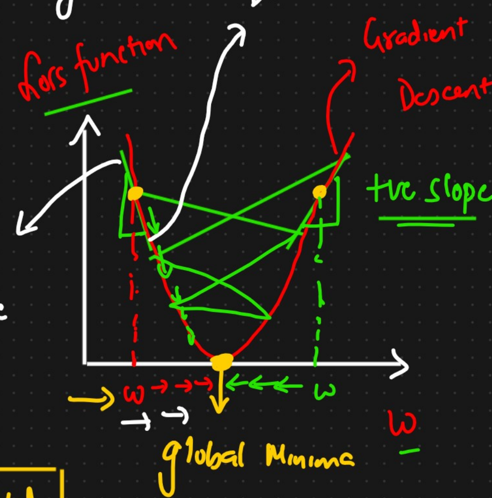
$\eta = 0.001$

① Weight updation formula

✓ $W_{new} = W_{old} - \eta \boxed{\frac{\partial L}{\partial W_{old}}}$ → Learning Rate

η ≈ small number

η = larger number



→ slope

$\frac{\partial L}{\partial W_{old}}$ $\boxed{\text{-ve slope}}$

$W_{new} = W_{old} - \eta(-ve)$

$= W_{old} + \eta(+ve)$

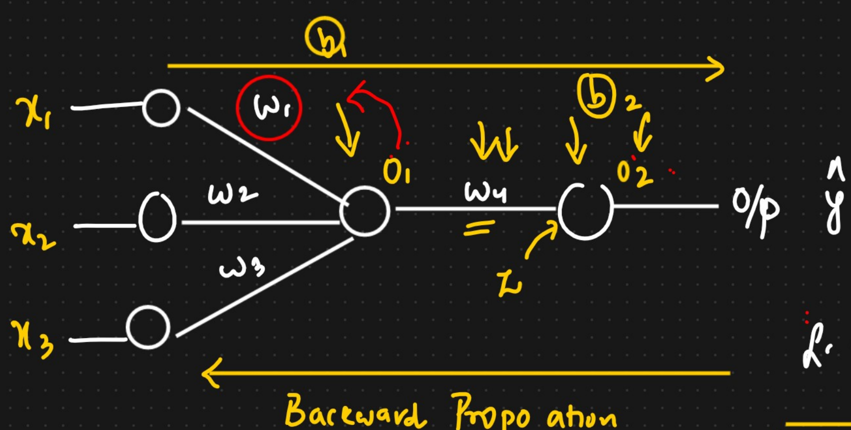
⇓

$\boxed{W_{new} >> W_{old}}$

$W_{new} = W_{old} - \eta(+ve)$

$\boxed{W_{new} << W_{old}}$

② Chain Rule of Differentiation



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

$z = \sigma(O_1 w_4 + b)$

$L \cdot (y - \hat{y})$ ⇓

$\boxed{W_{4new} = W_{4old} - \eta \frac{\partial L}{\partial W_{4old}}}$

$\boxed{b_{2new} = b_{2old} - \eta \frac{\partial L}{\partial b_{2old}}}$

{Chain Rule of Derivative}

$\boxed{\frac{\partial L}{\partial W_{4old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial W_{4old}}}$

$$w_{1_{new}} = w_{1_{old}} - \eta \boxed{\frac{\partial L}{\partial w_{1_{old}}}}$$

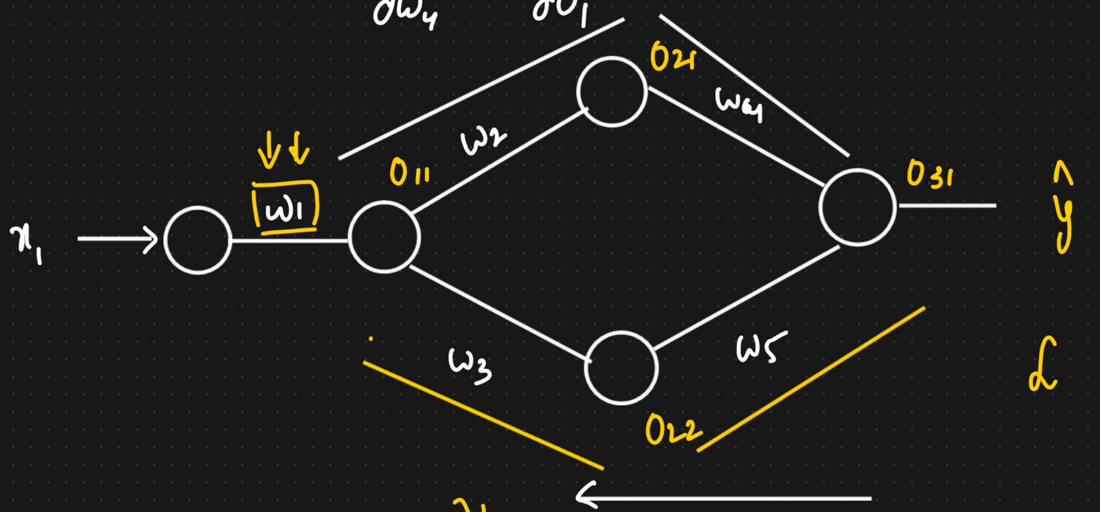
$$\boxed{w_{2_{new}} = w_{2_{old}} - \eta \frac{\partial L}{\partial w_{2_{old}}}}$$

Loss

$$\boxed{\frac{\partial L}{\partial w_{1_{old}}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1} * \frac{\partial O_1}{\partial w_{1_{old}}}}$$

$\downarrow$

$$\frac{\partial O_2}{\partial w_4} * \frac{\partial w_4}{\partial O_1}$$



Optimizers

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

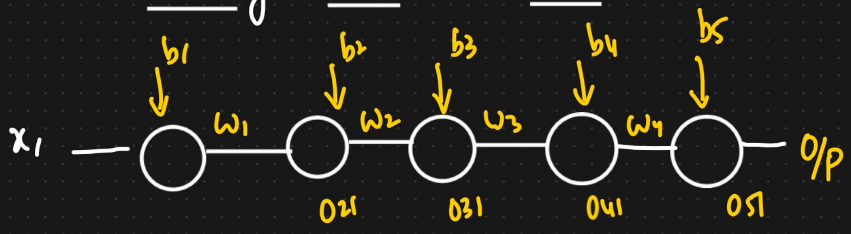
Chain Rule of Derivatives

$$\frac{\partial L}{\partial w_{1_{old}}} = \left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial w_{1_{old}}}\right]$$

$$+$$

$$\left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial w_{1_{old}}}\right]$$

③ Vanishing Gradient Problem



$\hat{y}$

Sigmoid Activation

$\boxed{\text{MSE}}$

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

$\boxed{0 \text{ to } 0.25}$

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

$$O_{51} = \sigma[(O_{41} * W_4) + b]$$

σ → Sigmoid Activation

$$\frac{\partial L}{\partial W_{1_{new}}} = \frac{\partial L}{\partial O_{51}} * \frac{\partial O_{51}}{\partial O_{41}} * \frac{\partial O_{41}}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial W_1}$$

$$= 0.25 * 0.15 * 0.10 * 0.05 * 0.02$$

$$= \boxed{\text{Small value}}$$

Derivative.

$\boxed{0 \le \sigma(y) \le 0.25}$

$y = w^T x + b$

$$\boxed{z = \frac{1}{1+e^{-x}}} \rightarrow 0 \text{ or } 1$$

≥ 0.5 1

< 0.5 0

$$W_{new} = W_{old} - \eta \,(\text{small number})$$

$$\boxed{W_{new} \approx W_{old}} \Rightarrow \text{Vanishing gradient Problem}$$

⇓

No change in weights

Another Activation function.

⇓

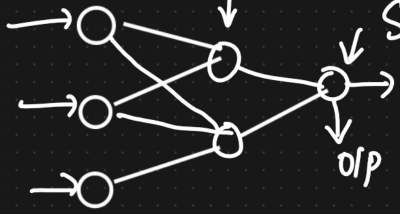
1. Sigmoid
2. Tanh
3. Relu
4. Leaky Relu
5. Prelu

Technique which Activation fn we should Used

Binary classification

(Relu) (prelu) (Elu)

Sigmoid Activation

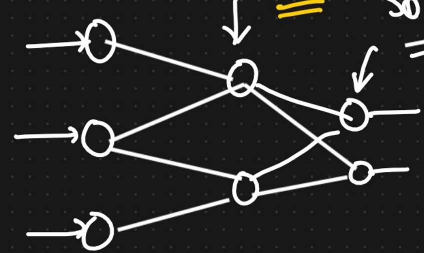


O/P

Multiclass

Relu

Softmax Activation



Regression

Relu

O/P { Linear Activation }

loss function

④ Loss functions

Deep Learning (ANN)

- Regression
- Classification

Loss functions

Regression

O/P

Exp	Degree	Salary
10	phd	—
—	—	—
—	—	—

Regression Problem

Dataset

Play	Study	Pass/Fail
10	2	Fail
4	3	Fail
5	5	Maybe
2	7	Pass

① Regression

① MSE {Mean Squared Error}
② MAE {Mean Absolute Error}
③ Huber Loss

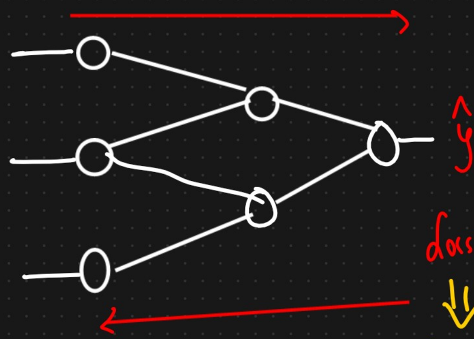
Loss function AND

Cost Function

10 records

⇓

Dataset = 100 records



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

⇓ VS

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

① Mean Squared Error (MSE)

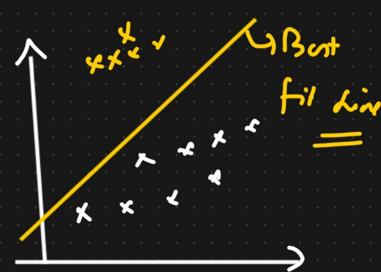
$Loss\ function = \frac{1}{2}(y-\hat{y})^2$ ↳ Error

⇓

Quadratic Equation

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

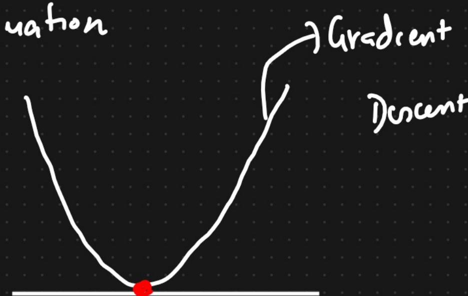
Penalizing Error



$(a-b)^2 = a^2 - 2ab + b^2$

⇓

$\boxed{ax^2 + bx + c}$

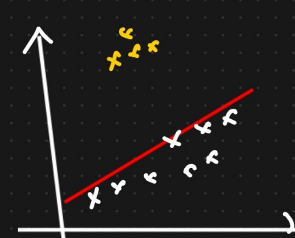


Advantages

① Differentiable
② It has only 1 local or Global Minima.
③ It converges faster

Disadvantage

① Not Robust to outliers.



② Mean Absolute Error

$Loss\ fn = \frac{1}{2}|y-\hat{y}|$

$Cost\ fn = \frac{1}{2}\sum_{i=1}^{n}|y-\hat{y}|$

① Robust to outliers ⇒ Time consuming ↑↑.

⇒ Subgradient

③ Huber Loss

① MSE ② MAE

outlier are not present

Hyperparameter

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

③ Classification

Cross Entropy
- → Binary Cross Entropy → Binary classification
- → Categorical Cross Entropy → Multiclass classification.

① Binary Cross Entropy

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

⇓

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

$$\boxed{\hat{y} = \frac{1}{1+e^{-x}}}$$

② Categorical Cross Entropy {Multiclass Classification Problem}

f1	f2	f3	O/P	Good	Bad	Neutral
2	3	4	Good	[1	0	0]
5	6	7	Bad	0	1	0
8	9	10	Neutral	0	0	1

{ i = Row
j = Column }

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

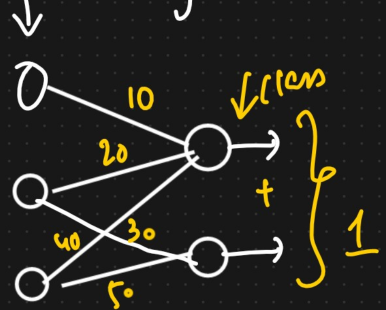
$$y_i = [y_{i1}, y_{i2}, y_{i3} \cdots y_{ic}]$$

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

$\hat{y}_{ij}$ = Softmax Activation ↗ O/P Layer

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

0.4, 0.5, 0.6



$$= \frac{e^{10}}{e^{10+20+30+40+50}} \Rightarrow$$

Conclusions

Relu, Softmax ⇒ Multiclass → Categorical Cross Entropy

Relu, Sigmoid ⇒ Binary → Binary Cross Entropy.

Linear Regression

Relu, Linear Activa → MSE, MAE, Huber Loss