# Drop Out & regularization

* Very huge deep artificial neural network have huge weight, huge bias then happens ann to do overfit the ann data
* ~~kkkk~~ * under fit will happen only single layer neural network
* under fit never happen in deep neural network
* overfit never happen in single layer neural network

overfit → high varies

① regularization
↳ $l_1, l_2$

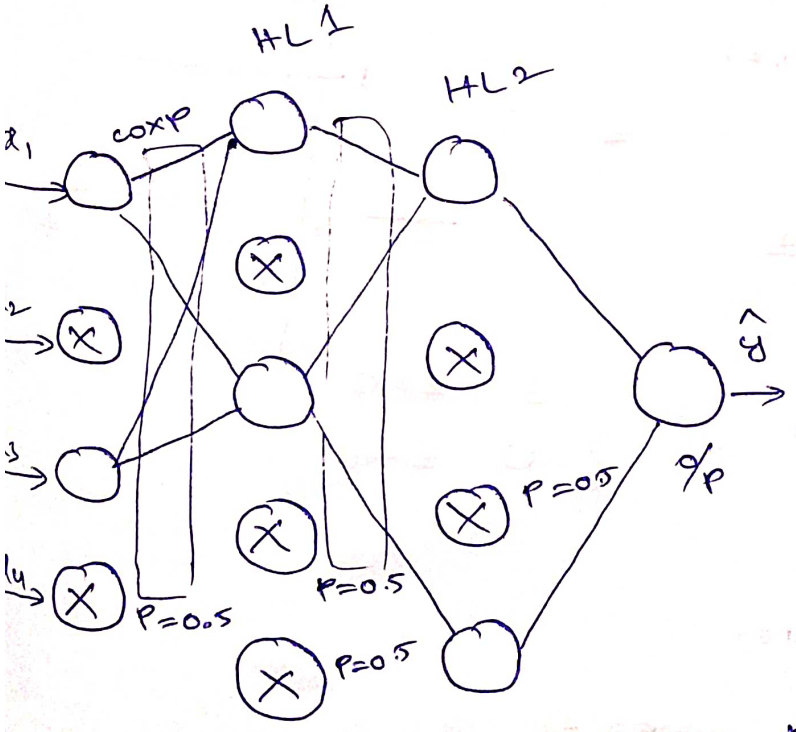
② Drop out

written
① Nitish Shrivasha
② Geoffry Hinton

Random forest

DT → overfitting

Subset of features




Drop out ratio

$0 \leq p \leq 1$

* select subset of features

* randomly select neuron, activation function then deactivate it in forward and backward propagation

* everytime select randomly

train data
test data ~~wxp~~

p → select by using hyper parameter tuning

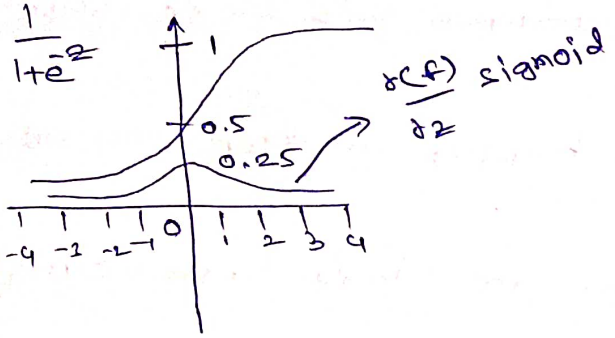
* Error is decreases

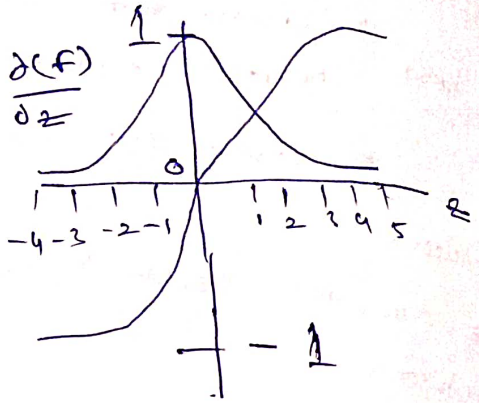
# Rectified linear Unit

$$0 \leq \frac{\partial(f)}{\partial(z)} \leq 0.25 \qquad \tanh(x) = \frac{1-e^{-2z}}{1+e^{-2z}}$$

① Sigmoid AF

② threshold Activation

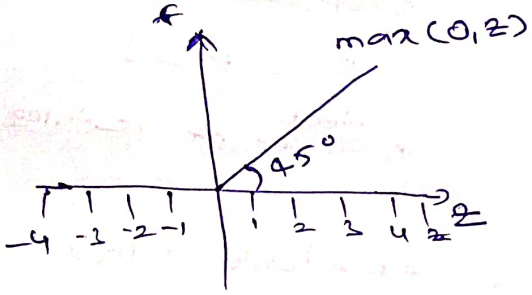


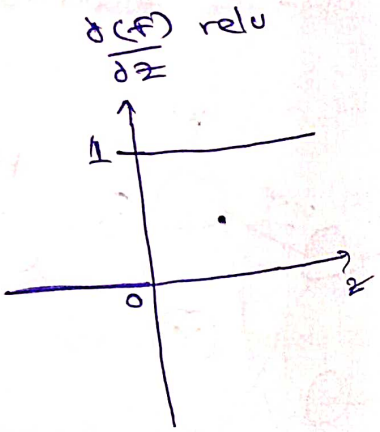

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$$

$$0.25 \times 0.10 \times 0.001$$

relu :-




⇒



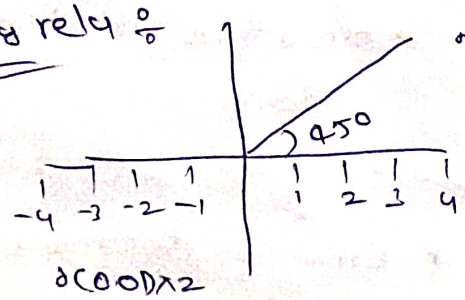

$$w_{old} = w_{new} - \eta \frac{\partial L}{\partial w}$$

$$1 \times 1 \times 1$$
$$0 \times 1 \times 1$$

↳ dead neuron

$$\begin{cases} 1 & z>0 \\ 0 & z<0 \end{cases}$$

leaky relu :-



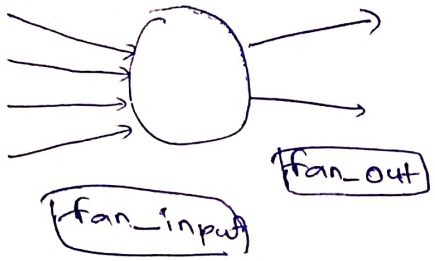

$$\frac{\partial(0.01 \times z}{\partial z} = 0.01$$

$$\begin{cases} z & z>0 \\ 0.01(a) & z<0 \end{cases}$$

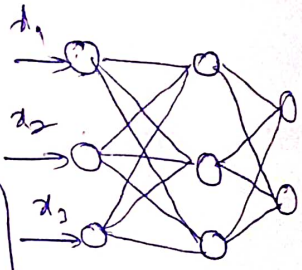
# Weight Initialization

## Key Points

1. weight should be small
2. weights should not be same because each neuron will learn new
3. weights should have good variance




## Weight Initialization Techniques




1. Uniform distribution

$$w_{ij} \sim \text{Uniform}\left[\underbrace{\frac{-1}{\sqrt{fan\_in}}}_{a}, \underbrace{\frac{1}{\sqrt{fan\_out}}}_{b}\right]$$

2. Xavier / Glorot

   1. Xavier Normal

$$w_{ij} \sim N(0, \sigma)$$

$$\sigma = \sqrt{\frac{2}{(fan\_in, fan\_out)}}$$

   2. Xavier Uniform

$$w_{ij} \sim U\left[\underbrace{\frac{-\sqrt{6}}{\sqrt{fan\_in + fan\_out}}}_{a}, \underbrace{\frac{\sqrt{6}}{\sqrt{fan\_in + fan\_out}}}_{b}\right]$$

⑤ He init

⑥ He uniform

$$w_{ij} \approx U\left[-\sqrt{\frac{6}{fan_{in}}}, \sqrt{\frac{6}{fan_{out}}}\right]$$

$a, b$

⑦ He Normal

$$w_{ij} \approx N(0, \sigma)$$

$$\sigma = \sqrt{\frac{2}{fan_{in}}}$$

## Stochastic Gradient Descent

1000 data points

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

$\frac{\partial L}{\partial w_{old}}$ → n data point → Gradient Descent

↳ 1 data → SGD

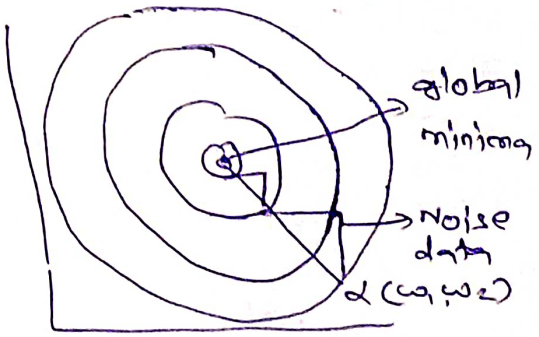
↳ k data point → miniBatch SGD

100 (sample)

$$\text{miniBatch SGD} = \text{loss} = \sum_{i=1}^{k} (y - \hat{y})^2$$

$$\text{GD} = \text{loss} = \sum_{i=1}^{n} (y - \hat{y})^2$$

$$\text{SGD} = \text{loss} = (y - \hat{y})^2$$




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

$$\left[\frac{\partial L}{\partial w_{old}}\right] \approx \left[\frac{\partial L}{\partial w_{old}}\right]$$

miniBatch SGD → Sample

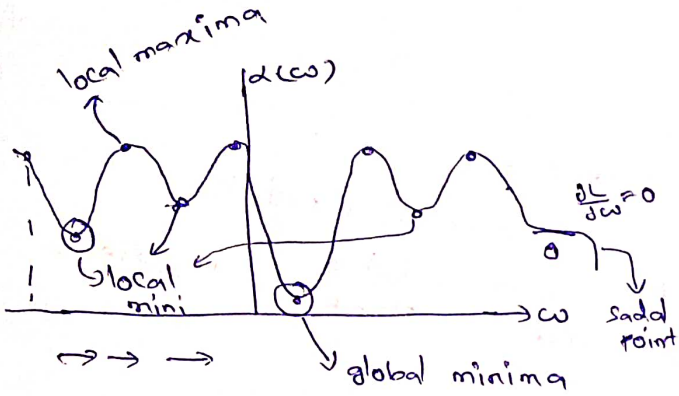
GD → population

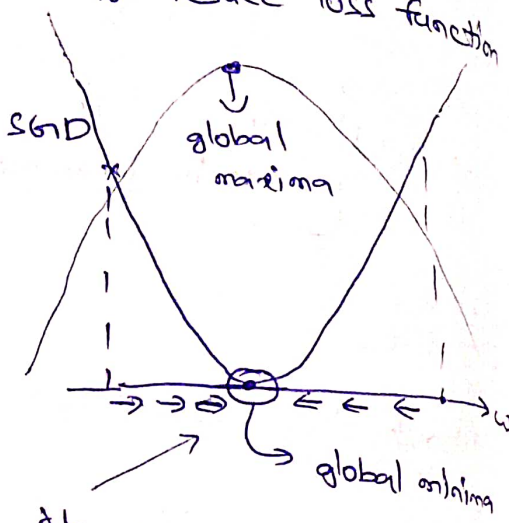
* remove noise stocastic GD with momentum

# Global minima Vs local minima.

① GD, SGD, Batch SGD ⇒ optimizers used for reduce loss function

$$L(w) = \sum_{i=1}^{k} (y - \hat{y})^2$$




$$\frac{\partial L}{\partial w} = 0$$

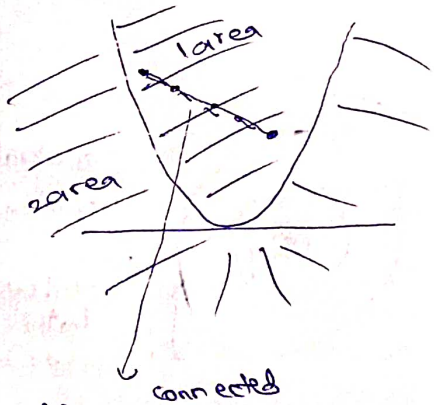
$$\Rightarrow w_{new} = w_{old} - \eta \left[\frac{\partial L}{\partial w_{old}}\right]$$

slope

$\eta \to 0$

① Convex function :

linear reg
logistic

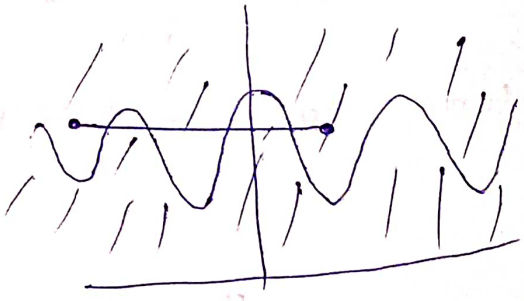



if point present in same curve is called Convex fun

* occure in ml problem as teneant

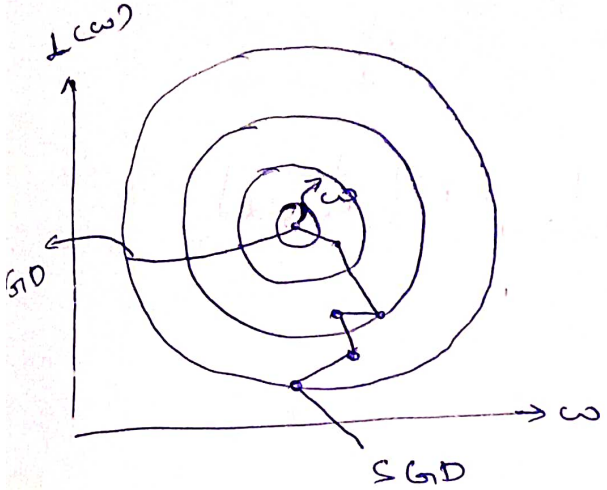
② Non Convex function :

↳ deeplearning



* mostly occure in deeplearning
* we choose two points and connect both. If both are present in same curve is Conver other wise non Convex

# Stocastic Gradient Descent with Momentum :-

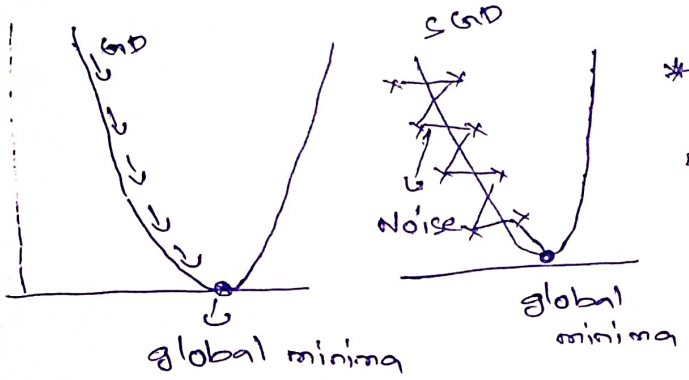



GD optimizer :-

* Iam take all the datapoints and it will converge fastly
* if data point is more it will take more time and computational power overcame we use SGD

SGD :-

* select batch (like, 100, 200 data points select)
* Noise data
* remove Noise using exponential moving average




Exponat moving average

time interval

| $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ ... $t_n$ |
|---|---|---|---|---|
| $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ ... $b_n$ |

$0 \leq \gamma \leq 1$

$0.5$

$V_{t_1} = b_1$

$V_{t_2} = \gamma \times V_1 + b_2$

$= 0.5 \times b_1 + b_2$

$V_{t_3} = \gamma \times V_2 + b_3$

$= \gamma * (\gamma \times V_1 + b_2) + b_3$

$= \gamma^2 + b_1 + \gamma b_2 + b_3$

$= 0.25 \times b_1 + 0.5\, b_2 + 1 \times b_3$