17 December 2023 19:59

Deep learning Day-5

(3) ADAM (Adaptive moment estimation)

Best optimization function for weight and bias
→ MBSGDA with momentum

Best optimization function for learning rate
→ Ada Delta (RMS prop)

ADAM = MBSGDA with momentum + Ada Delta.

→ ADAM updates weight, bias and learning rate simultaneously.

* MBSGDA with momentum

$$W_{new} = W_{old} - \alpha \, V_{dw}$$

$$b_{new} = b_{old} - \alpha \, V_{db}$$

* Ada Delta

$$W_{new} = W_{old} - \alpha_{new} \frac{\partial C}{\partial w}$$

$$b_{new} = b_{old} - \alpha_{new} \frac{\partial C}{\partial b}$$

$$\alpha_{new} = \frac{\alpha_{old}}{\sqrt{S_{dwt} + \epsilon}}$$

* ADAM

$$W_{new} = W_{old} - \frac{\alpha_{old}}{\sqrt{S_{dwt} + \epsilon}} \times V_{dw}$$

$$b_{new} = b_{old} - \frac{d_{old}}{\sqrt{Sd_{bt}} + \epsilon} \times \eta d_b$$

* Best optimization function to update weight bias and learning rate → ADAM

(1) MBSGDA with momentum
(2) Ada Delta (RMS prop) } 10%
(3) ADAM — 90%

* Activation function

It controls the output of any neuron in neural network

→ Type of Activation function

(1) Sigmoid → Binary classification
(2) Tanh →
(3) Softmax → multiclass classification (modified version of Sigmoid)
(4) Relu (Rectified linear unit) (Base activation function for regression)
(5) leaky relu
(6) P-relu
(7) Elu
(8) Swish

} variant of relu

7. Elu
8. Swish

1. Sigmoid

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

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

→ Sigmoid always gives probability value of any one class

let's say classes < 1, 0

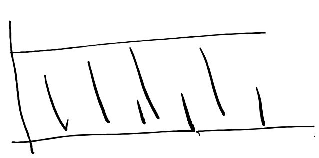
$P(1) = 0.7$

$P(0) = 1 - P(1)$
$= 1 - 0.7$
$= 0.3$

→ Range — 0 to 1

* Advantages of Sigmoid activation funⁿ if we use it in output layer :-

1. Sigmoid activation funⁿ gives uniform result. (Range — 0 to 1)



2. On the basis of probability values, we can take decision confidently.

Classes < 0, 1

→ $P(1) = 0.7$ ✓   threshold — 0.5

$1 \rightarrow P(1) = 0.7$ ✓

threshold — 0.5

$P(0) = 1 - P(1)$

$0 \rightarrow P(0) = 0.3$ ✓

* Why we avoid to use sigmoid in our hidden layer?.

- First reason

  → Sigmoid is a non zero centric function.

* Zero centricity

$\text{mean}(f(x)) \simeq 0 \rightarrow f(x) \rightarrow$ zero centric function

$\text{mean}(f(x)) \neq 0 \rightarrow f(x) \rightarrow$ Non zero centric function

→ Zero centric function conversion faster than non zero centric function.

Sigmoid → Range 0 to 1

→ mean

→ 0.5

Sigmoid → Non zero centric function

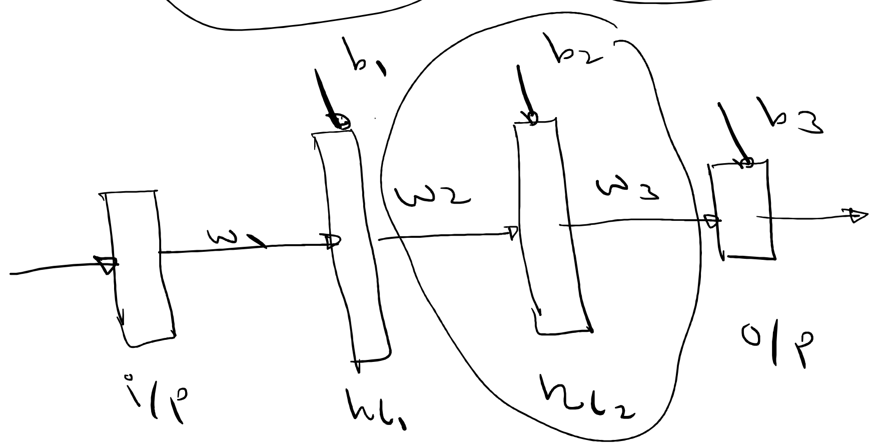
Second Reason

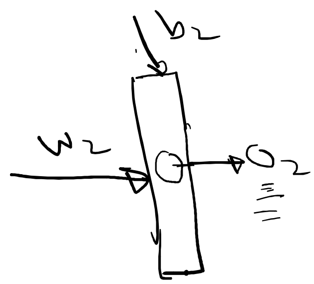
Second Reason

derivative of sigmoid = small Number

$$0.25 \times 25 \rightarrow 0.25$$

$$1 \times 25 \rightarrow 25$$





O_2 (output of Hidden layer 2)

O_2 → activation funn $\left(\sigma(z) = \frac{1}{1+e^{-z}}\right)$ → summation function $\left(z = \sum_{i=1}^{n} w_i x_i + b\right)$ → w, b → loss → $\frac{\partial c}{\partial w}, \frac{\partial c}{\partial b}$

$$\frac{d}{dz}\sigma(z)$$

$$\boxed{\frac{\partial L}{\partial w} = \frac{\partial O_2}{\partial w}}$$

$$\frac{\partial L}{\partial w} = \frac{\partial O_2}{\partial z} \times \frac{\partial z}{\partial w}$$

0.25×25 $\quad$ 1×25

$$\frac{\partial L}{\partial w} = \frac{\partial O_2}{\partial z} \times \frac{\partial z}{\partial w}$$

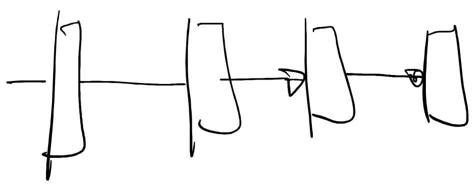
$0.25 \times .25 \rightarrow 6.25$ $\qquad$ $.1 \times .25 \rightarrow .25$

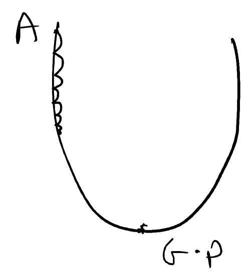
$$O_2 = \sigma(z)$$

$$\frac{\partial L}{\partial w} = \frac{\partial \sigma(z)}{\partial z} \times \frac{\partial z}{\partial w}$$

derivative of sigmoid

→ vanishing gradient





* Drawback of Sigmoid

1. It is a Non Zero centric function. hence it will take longer time to conversion.

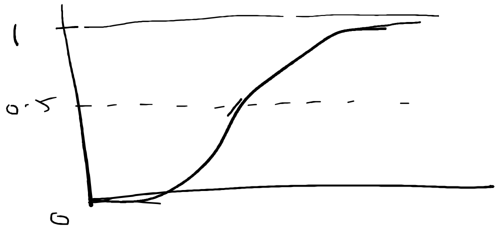
2. Derivative of sigmoid is very small number and because of that we can face issue of vanishing gradient.

② Tanh

Sigmoid	Tanh
Range — 0 to 1	Range → -1 to 1
mean ~ 0.5	mean → 0

mean 0.5

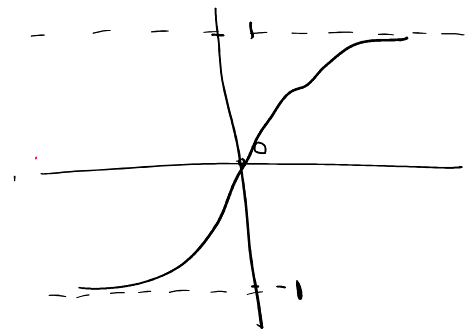
→ it is a non zero centric funn



→ $\sigma(z) = \frac{1}{1+e^{-z}}$

mean → 0

→ it is zero centric funn



→ $\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z}+e^{-z}}$

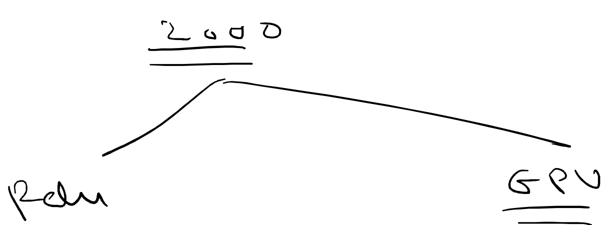
* Advantage of Tanh :-

① It is a zero centric function hence it will conversion faster than sigmoid.

* Disadvantage of Tanh :-

① It is mathematically heavy as compare to sigmoid

② Derivative of tanh is still small number.



③ Relu (Rectified linear unit)

→ Relu is a non zero centric function.

→ it is work on maximize theory

$$relu(z) = max(0, z)$$

Case I → $z \to +ve$ $(z = 55)$

$$relu(z) = max(0, 55)$$

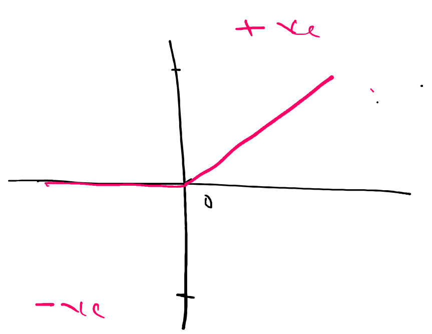
$$relu(z) = 55$$

Case II → $z = -ve$ $(z = -55)$

$$relu(z) = max(0, -55)$$

$$relu(z) = 0$$

Rectified linear unit



* How Relu solved vanishing gradient problem

Case I → $z \to +ve$

$$\frac{\partial}{\partial z} relu(z) = \frac{\partial}{\partial z} max(0, z)$$

$$\frac{\partial}{\partial z} relu(z) = \frac{\partial}{\partial z}(z)$$

0.25×25 $\qquad$ 1×25

$\frac{\partial}{\partial z} relu(z) = \frac{\partial}{\partial z}(z)$

0.25 → 25

$\downarrow$ 0.25 $\downarrow$ 25

$= 1$

$$\boxed{\frac{\partial L}{\partial w} = \underbrace{\frac{\partial}{\partial z} relu(z)}_{1} \times \frac{\partial z}{\partial w}}$$

Case II → z → -ve

$\frac{\partial}{\partial z} relu(z) = \frac{\partial}{\partial z} max(0, z)$

$= \frac{\partial}{\partial z}(0)$ → constant

$\frac{\partial}{\partial z} relu(z) = 0$

$\frac{\partial L}{\partial w} = \frac{\partial}{\partial z} relu(z) \times \frac{\partial z}{\partial w}$

→ 0 → 0

$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w}$ → 0

$w_{new} = w_{old}$

→ Dead Neuron or Dead Relu

z → -ve

→ $z = \sum_{i=1}^{n} w_i x_i + b$ -ve

10 lakh

x_1 x_2 x_3 y

↓ Humidity — Temp ↑

↓ Humidity — Temp ↑

$z \rightarrow -ve$
- 5 %
- 95 % $\rightarrow z \rightarrow +ve$

* Advantage of Relu
(1) mathematically light weight.
(2) with the help of Relu we can avoid vanishing gradient problem.

* Disadvantage of Relu
(1) It is a Non zero centric function
(2) We might face Dead Neuron or Dead Relu issue.

(4) leaky Relu

$$relu(z) = max(0, z)$$

$$lr(z) = max(0.001z, z)$$

Case I $\rightarrow z = +ve$

$$\frac{\partial}{\partial z} lr(z) = \frac{\partial}{\partial z} max(0.001z, z)$$

$$= \frac{\partial}{\partial z}(z)$$

$$\frac{\partial}{\partial z} lr(z) = 1$$

$$\frac{\partial}{\partial z} lr(z) = 1$$

Case II $\rightarrow$ $z = -ve$

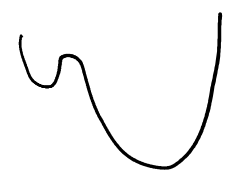
$$\frac{\partial}{\partial z} lr(z) = \frac{\partial}{\partial z} \max(0.001z, z)$$

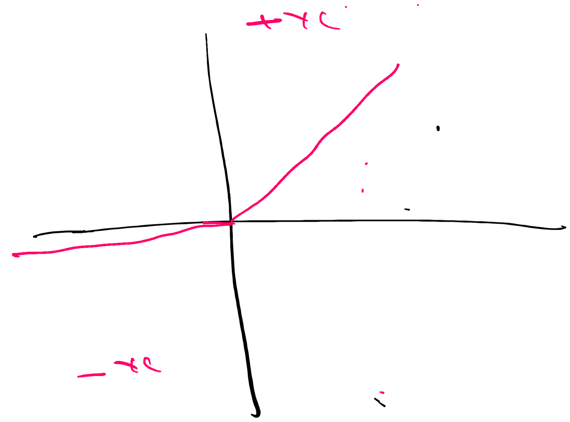
$$= -0.001 \frac{\partial}{\partial z}(z)$$

$$\frac{\partial}{\partial z} lr(z) = \underline{-0.001} \text{ —— small}$$

$$\frac{\partial L}{\partial w} = \frac{\partial}{\partial z} lr(z) \times \frac{\partial z}{\partial w}$$

$-ve$





(5) Parametric Relu (p-relu)

$$relu(z) = \max(0, z)$$

$$lr(z) = \max(0.001z, z)$$

$$p\text{-}relu(z) = \max(cz, z)$$

learnable parameter

$[0, 0.001, 0.1, 1]$

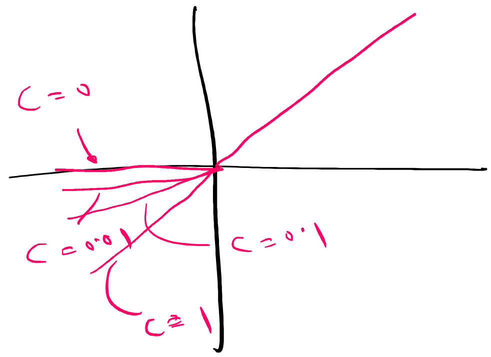
$\rightarrow$ when $c = 0$

$$prelu(z) = \max(0, z) \longrightarrow \text{Relu}$$

prelu(z) = . . .

when c = 0.001

prelu(z) = max(0.001z, z) → lr.



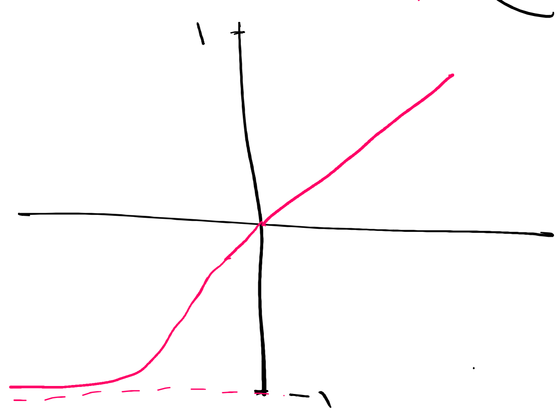
(6) Elu (Exponential linear unit)

Exponential function
→ learning curve
→ smooth

→ Elu is a zero centric function.

→ $\text{Elu}(z) = \max(\alpha(e^z - 1), z)$

α → learnable parameter.



$$\text{elu}(x) = f(x) : \begin{cases} x : \text{if } x > 0 \\ \alpha(e^x - 1) : \text{otherwise} \end{cases}$$

$$elu(x) = f(x) : \begin{cases} \alpha(e^x - 1) : \text{otherwise} \end{cases}$$

$$\boxed{elu(z) = \max(\alpha(e^z - 1), z)}$$

(7) Swish

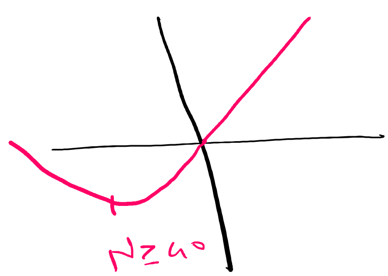
$$Swish(z) = z * \sigma(z)$$

$$Swish(z) = z * \frac{1}{1+e^{-z}}$$

→ we can not use Swish in case of Shallow Neural Network

→ we can use Swish when $N \geq 40$

where N → No of hidden layer.



(8) Softmax

→ it is a modified version of sigmoid and it is specially design for multi class classification.

let's say we are dealing with multiclass classification and we have 5 classes.

output of softmax

Class

A

B

C

D

E

$P(x \in y = A) = P(A)$

$P(x \in y = B) = P(B)$

$P(x \in y = C) = P(C)$

$P(x \in y = D) = P(D)$

$P(x \in y = E) = P(E)$

$$P(A) + P(B) + P(C) + P(D) + P(E) = 1$$

$$\underbrace{\max [P(A), P(B), P(C), P(D), P(E)]}_{\text{Classification}}$$

$$S(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$$

Real Number

$$S(A) = P(A) = \frac{e^{A}}{e^{A} + e^{B} + e^{C} + e^{D} + e^{E}}$$

— 10

— 20

— 40

— 60

— 70

$$\sigma(10) = \frac{1}{1 + e^{-10}} = \text{—}$$

A — 10

B — 20

$$S(A) = P(A) = \frac{e^{A}}{e^{A} + e^{B} + e^{C} + e^{D} + e^{E}}$$

$$\left.\begin{matrix} A - 10 \\ B - 20 \\ C - 40 \\ D - 60 \\ E - 70 \end{matrix}\right]$$

$$S(A) = P(A) = \frac{}{e^A + e^B + e^C + e^D + e^E}$$

$$= \frac{e^{10}}{e^{10} + e^{20} + e^{40} + e^{60} + e^{70}}$$

$$= \underline{}$$

→ Softmax convert a ~~vector~~ vector of ~~k~~ real number into a probability distribution of k possible outcome.