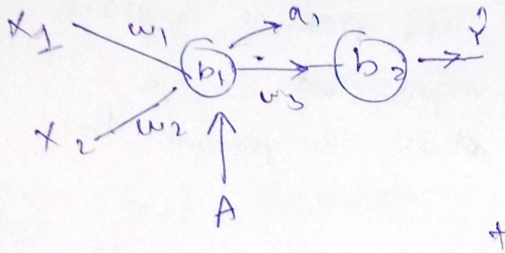
03-Oct 2024

Dying ReLU Problem:

for any input value in the neural network. is some of the. neuron of hidden layer. is dead (there is no contribution) we can also say that it is forever dead due to this (50% neuron are dead) the data patterns are not capture effectively. (not ~~easy~~ easy to find the pattern in the data).

why dying ReLU problem Occur:



$a_1 = max(0, z_1)$

$z_1 = w_1 x_1 + w_2 x_2 + b_1$

if $z_1 < 0$

then $a_1 = 0$

then $\frac{\partial a_1}{\partial z_1} = 0$ —— (iii)

due to this w_1 and w_2 not able to update

because in back propagation

$$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1} \quad \text{—— (i)}$$

$$w_2 = w_2 - \eta \frac{\partial L}{\partial w_2} \quad \text{—— (ii)}$$

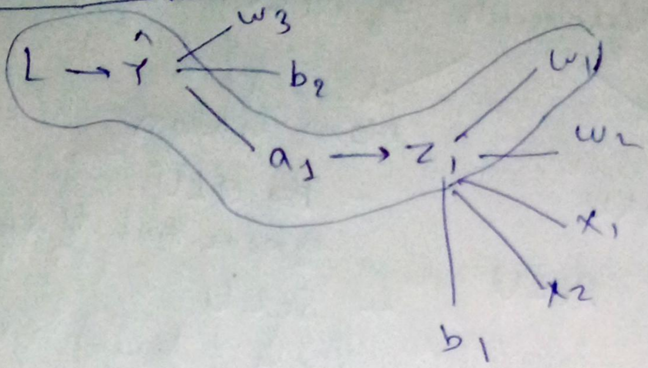
for calculating $\frac{\partial L}{\partial w_1}$ and $\frac{\partial L}{\partial w_2}$ the (iii) is used. due to this the equation

$w_1 = w_1$ and $w_2 = w_2$

there is no update hence.

node A is consider as a dead node.

detail explanation:



hence $\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}$

similarly,

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_2}$$

it is the term (iii)

when $z_1 = w_1 x_1 + w_2 x_2 + b$, become -ve?

there is two reasons

(I) learning rate is very high.

(II) High ~~posive b~~ negative bias

$b_1 << 0$

* dead neuron is not ~~reco~~ recoverable.

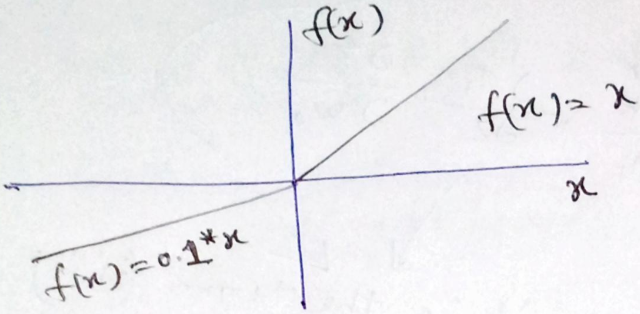
Solutions to Resolve dying ReLU:

- set law learning rate
- bias set +ve value example 0.01
- don't use ~~ReLU~~ ReLU, use it variants.

Variants of ReLU

- Linear
  - Leaky ReLU
  - Parametric ReLU
- Non-linear
  - eLU
    exponential linear unit
  - SeLU
    Scale linear unit.

(i) Leaky ReLU



$f(z) = \max(0.01z, z)$

if $z \geq 0 \rightarrow z$

if $z < 0 \rightarrow 0.01z$

due to this

value of $f'(z)$ for $z \geq 0 \rightarrow 1$

for $z < 0 \rightarrow 0.01$

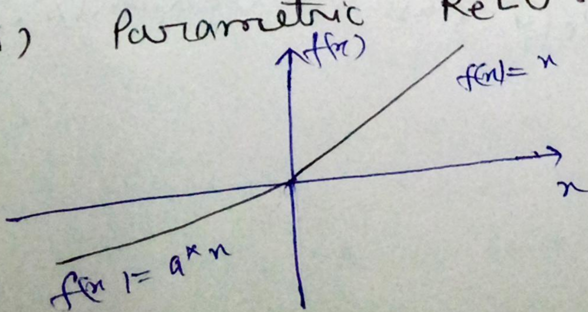
advantages:

- Non-saturated. (unbounded in both direction)
- easy to compute
- No dying ReLU
- close to zero-centered.

Disadvantages:

- why we use 0.01 value only.

(ii) Parametric ReLU:



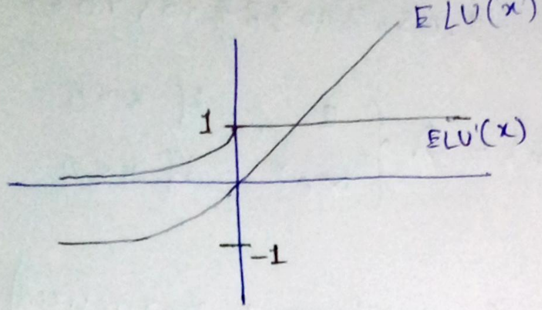
$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{otherwise} \end{cases}$$

Here 'a' is trainable parameter

advantages:

- all advantage are same as leaky ReLU
- it is flexible and performance better than the leaky ReLU

(iii) ELU - Exponential Linear Unit:



$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

$$ELU'(x) = \begin{cases} 1 & \text{if } x > 0 \\ ELU(x) + \alpha & \text{if } x \leq 0 \end{cases}$$

Here α is constant

α range is 0.1 to 0.3

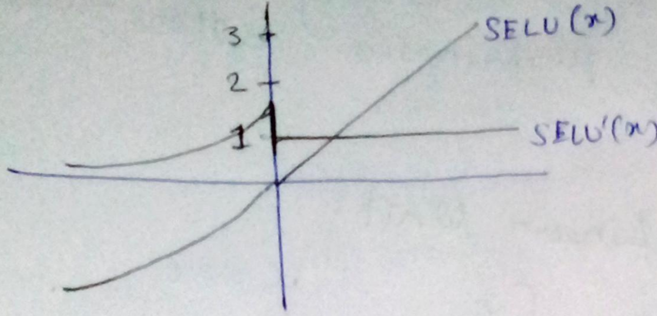
advantages:

- performance better than ReLU
- continuous at every point
- always differentiable.
- values are close to zero centered hence convergence is faster.
- generalized results are better (in test data)
- there no dying ReLU problem.

disadvantages:

- computationally expensive due to e^x

(iv) SeLU - Scaled Exponential Linear Unit



$$SELU(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^{x} - \alpha & \text{if } x \le 0 \end{cases}$$

$\alpha \approx 1.6732632423543$

$\lambda \approx 1.0507009873554 8 0$

$$SELU'(x) = \lambda \begin{cases} 1 & \text{if } x > 0 \\ \alpha e^{x} & \text{if } x \le 0 \end{cases}$$

advantages:

- it is self normalizing. (activation is normalized)
  means. mean of activations = 0
  standard deviation = 1
  hence NN converges faster

Disadvantages:

- New in market
- there is less research work on it.