Session-4

Neural Network-4 N layer NN

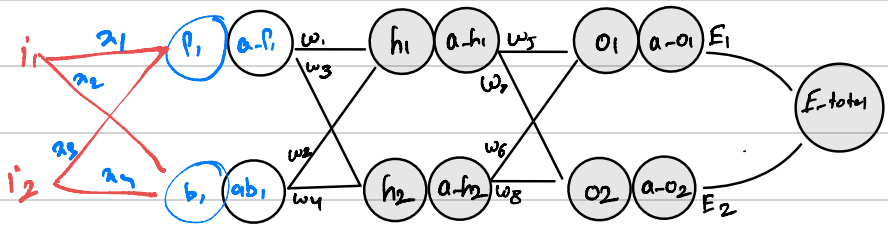
Feb 12, 2024

Agenda

1. Recap → Back prop
   → CCE
   → Linear activation function.
2. Activation function → Sigmoid
   → Tanh
   → Any problems
   → Relu
3. Notation & Tensorflow

Can We Generalize

$\frac{\partial L}{\partial x_1}$



$\frac{\partial E\_total}{\partial w_1}$

$$\frac{\partial E\_total}{x_1} \Rightarrow \frac{\partial E}{w_1} \times \frac{\partial w_1}{\partial a\text{-}p_1} \times \frac{\partial a\text{-}p_1}{\partial p_1} \times \frac{\partial p_1}{\partial x_1}$$

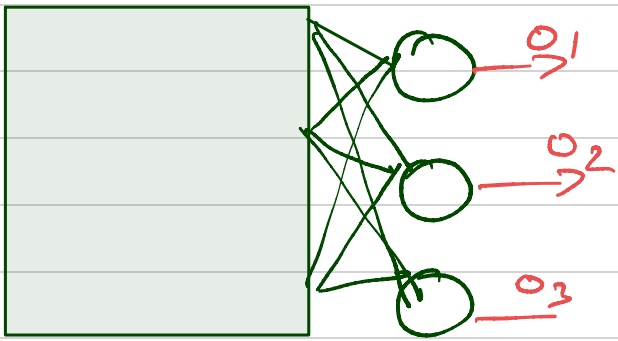
Incomplete

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

Sigmoid
tanh
ReLU
Leaky ReLU

Elelh Dog Cat

[1 0 0]



	Actual	Predicted
1st row	[0 1 0]	[0.9, 0.05, 0.05]
2nd row	[1 0 0]	[0.1 0.8 0.1]
3rd row	[0 0 1]	[0.2 0.2 0.6]

$$CCE = -\sum y_i \times \log(\hat{y}_i)$$

y_i: actual, $\hat{y}_i$: Predicted.

	Actual	Predicted
1st row	$[0 \quad 1 \quad 0]$	$[0.9, \ 0.05, \ 0.05]$

$$\Rightarrow -\left[0 \times \log(0.9) + 1 \times \log(0.05) + 0 \times \log(0.05) \right]$$

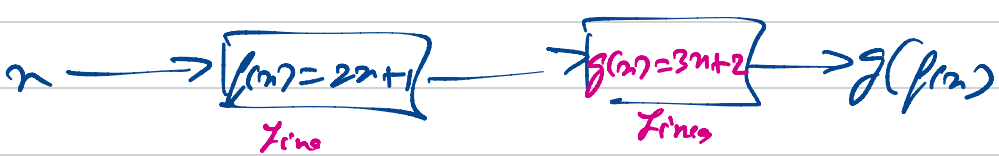
$$\Rightarrow -1.3$$

Row 2 & Row 3

$$-\log(0.1) - \log(0.6) - \log(0.05)$$

$$\Rightarrow -2.52$$

→ Linear activation

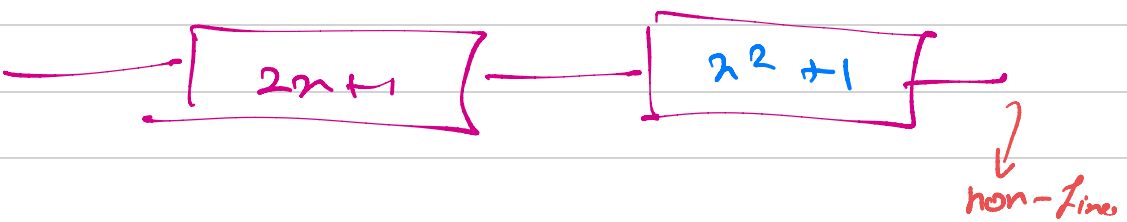


$$g(f(x)) = 3(2x+1) + 2$$

$\Rightarrow 6x + 3 + 2$

$- 6x + 5$ — Line

$n(x) = 6x + 5$



ACTIVATION FUNCTION

1. Sigmoid function

→ Domain $(-\infty, \infty)$

→ Range $(0 - 1)$

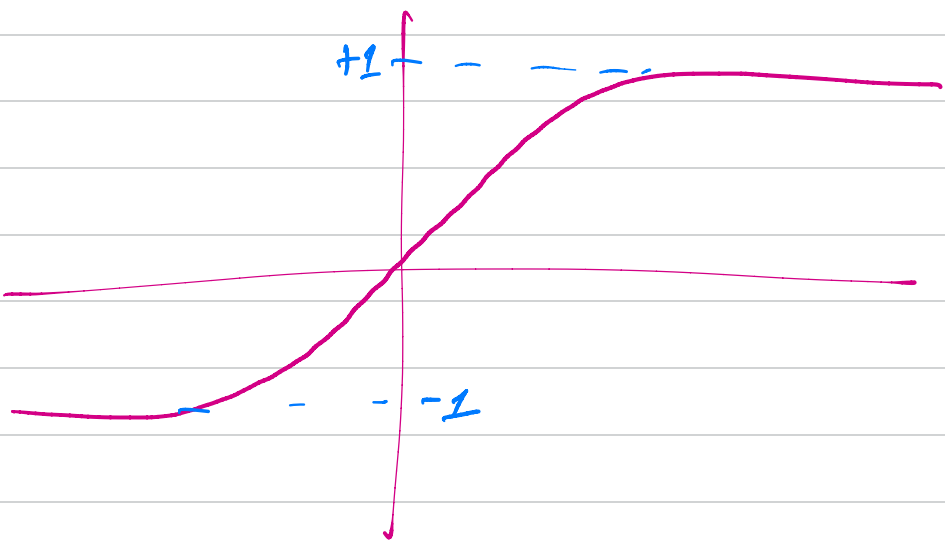
→ $f'(x) = \sigma(x)(1 - \sigma(x)) \rightarrow (0 - 1)$

(2) Tan - h activation func

$$\tanh(x) = \frac{e^{2} - e^{-2}}{e^{2} + e^{-2}}$$

Domain = $(-\infty, \infty)$

Range = $(-1, 1)$



Always works better than sigmoid

→ Derivative of tanh → $1 - \tanh^2(x)$

Range → $(0, 1)$

→ Possible problem with tanh / Sigmoid

1. Comp. expensive
2. Range ??

Range → (-1, 1) → tanh

Sigmoid → (0, 1)

* After 2-3 layers of NN training

gradients → 0

S. no.

U.U.U. small

$W_{new} = W_{old} - \alpha \times \frac{\partial L}{\partial W_{old}}$ → 0

10^{-5} 10^{-6}

$W_{new} \approx W_{old}$

Vanishing Gradients

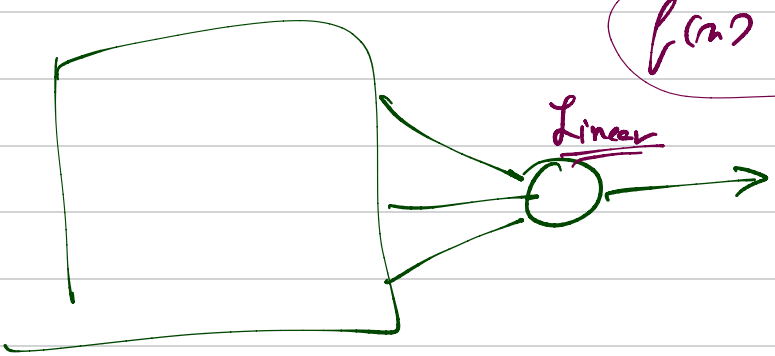
I'm not able to optimize

Q. Why weights of Neuron randomly initialized, why not zero?

$\frac{\partial L}{\partial w}$ also zero

→ For almost 2 decades we couldn't train network for more than 4 Layers

$f(x) = x$ 1. 2.



How do we deal with Vanishing gradient

→ Relu

$$f(x) = \max(0, x)$$

$$= \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$$w_{old} - \alpha \frac{\partial L}{\partial w_0} \rightarrow 0$$

α: 10^{-5} 10^{-4}

gradient → $\frac{\partial L}{\partial w}$ / Parameter

$$f(x) = \max(0, z)$$

$$= \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$f'(x) = \{1, 0\}$ ✗ Not differentiable

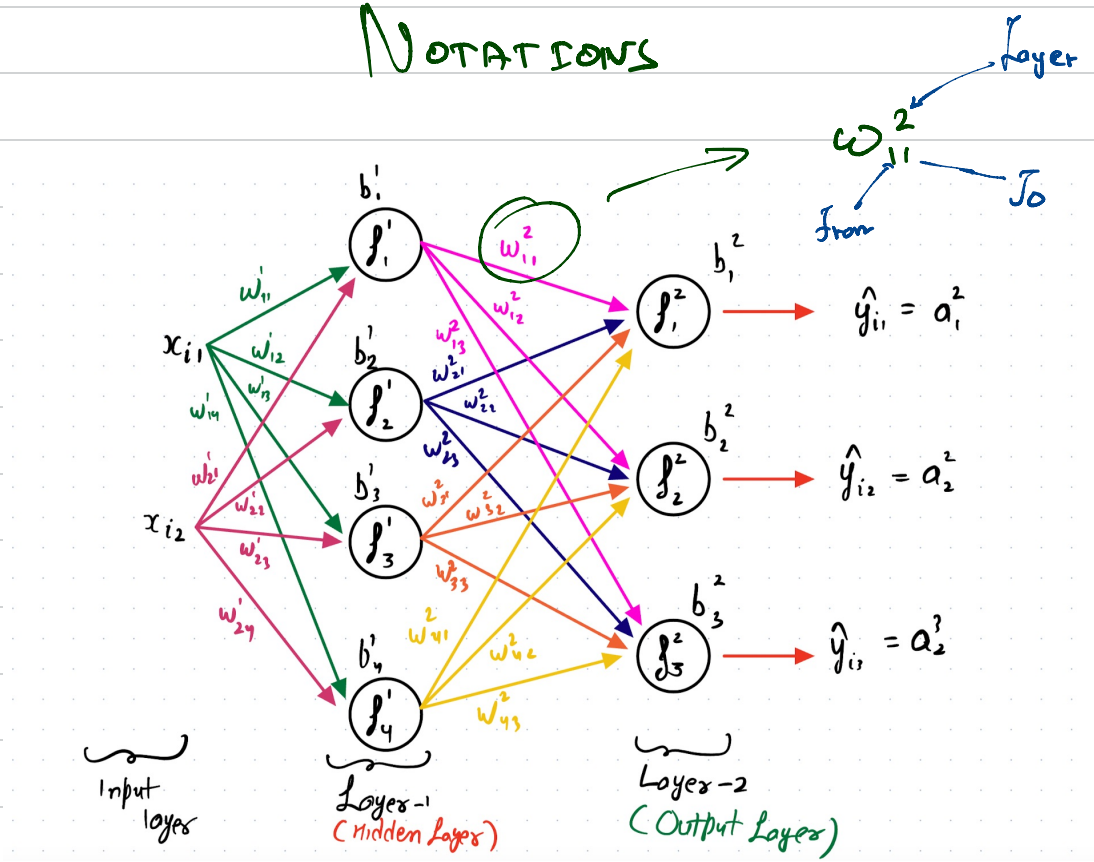
Leaky ReLu →

$$g(x) = \begin{cases} x & \text{if } x > 0 \\ 0.1x & \text{if } x \leq 0 \end{cases}$$

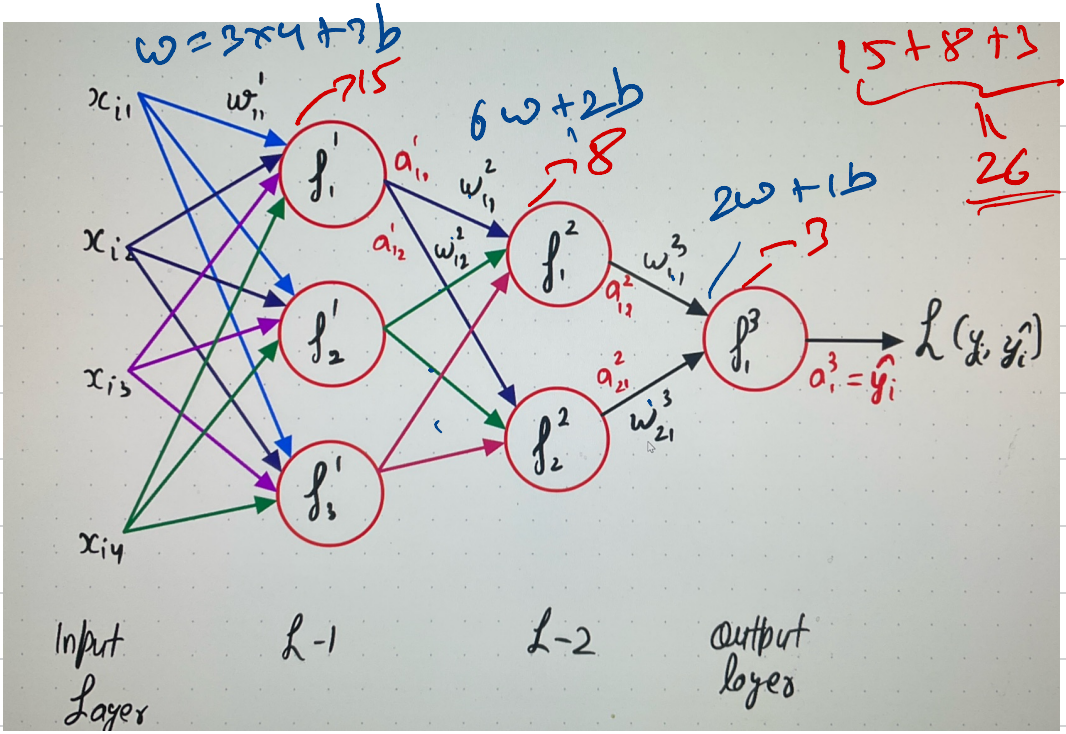
ReLu still performs better

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

NOTATIONS


Layer no
To
From
Input layer
Layer-1 (Hidden Layer)
Layer-2 (Output Layer)


L=2
3
4



multiclass

gorha va

[1 0 0]

SGD → Gradient descent

Adam → Most commonly use optimizer