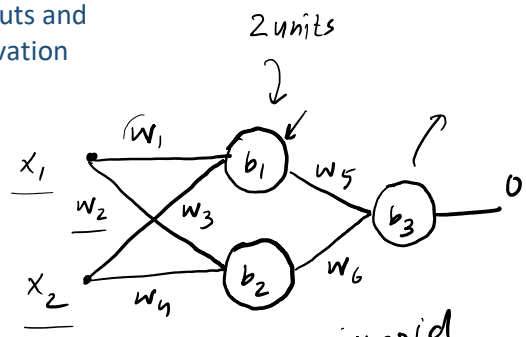
What are Activation Functions?

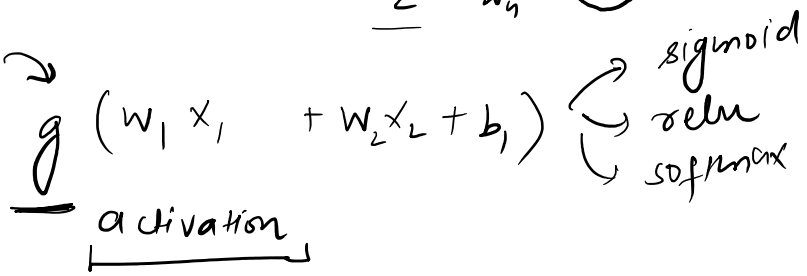
31 May 2022 14:49

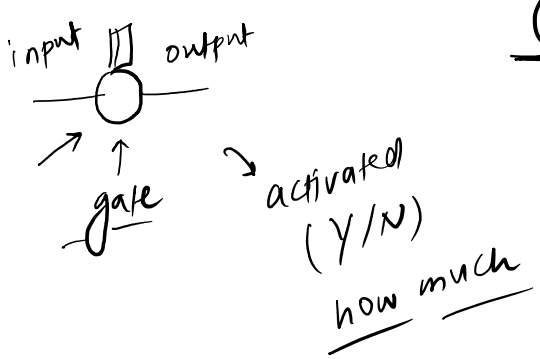
In artificial neural networks, each neuron forms a weighted sum of its inputs and passes the resulting scalar value through a function referred to as an activation function or transfer function. If a neuron has n inputs then the output or activation of a neuron is

$$a = g(w_1x_1 + w_2x_2 + w_3x_3 + \ldots w_nx_n + b)$$

This function g is referred to as the activation function.

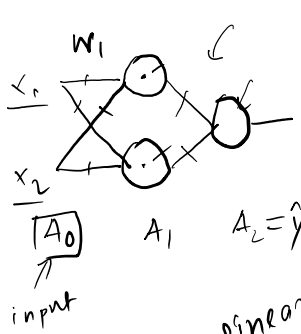






Why Activation Functions are needed?

31 May 2022 14:50



$$Z_1 = W_1 A_0 + b_1$$

$$A_1 = g(Z_1) = Z_1$$

linear

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

ANN

↓

Non-linear

Non-linear

output = input

$$A_2 = g(W_2 A_1 + b_2)$$

$$= W_2 A_1 + b_2 = W_2 (W_1 A_0 + b_1) + b_2$$

$$= W_2 W_1 A_0 + W_2 b_1 + b_2$$

$$= W' A_0 + b'$$

output

input

$$A_2 = \hat{y} = W' A_0 + b'$$

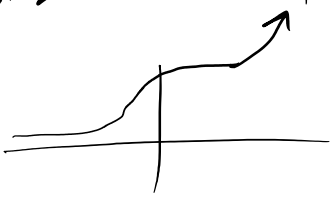
linear

Ideal Activation Function

31 May 2022 14:50

$y = f(x)$ ✓ linear x non-linear

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



1) Non-linear ✓ → Universal Approx Theorem

2) Differentiable ✓ → gradient des → derivatives → relu (0)

3) Computationally inexpensive ✓ → derivative → simple, easy, fast → training slow

4) Zero centered ← Zero cent normalized → mean = 0 → normalized data input → later layers → tanh

5) Non-saturating → Sigmoid → [0, 1] → faster → $f(x) = \max(0, x)$

tanh → (−1, 1) Relu

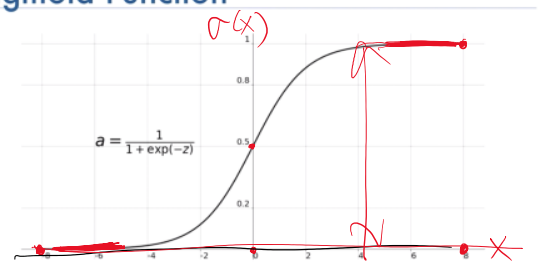
update x [0, 1] → backprop → saturating AF → Vanishing gradient Problem

0.01 0.0001 0.0001

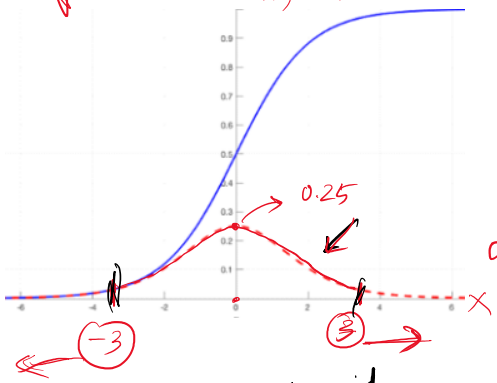
Sigmoid Activation Function

31 May 2022 14:50

Sigmoid Function



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



$w_1x_1 + w_2x_2 + b$

input ↑ → Norm

$x \simeq 0$

$\sigma'(x) \simeq 0$

w_1, w_2, b

weight init

Yes / No

sigmoid → 0-1 → yes / No

Advantages

1) [0,1] → probability → output layer → Binary classification

2) Non-linear → Non-linear data → good option

3) Differentiable → Backprop → $\frac{\partial L}{\partial w}$

Disadvantages

1) Saturating function → $[-\infty, \infty] \rightarrow [0,1]$

↳ Vanishing gradient problem

Backprop → update

$$w_n = w_0 - \eta \frac{\partial L}{\partial w}$$

$w_n = w_0$ → training ✗

No update will take place

Hidden layer → sigmoid → training ✗

output → binary classification

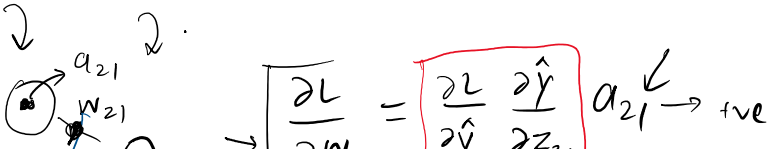
2) Non Zero centered

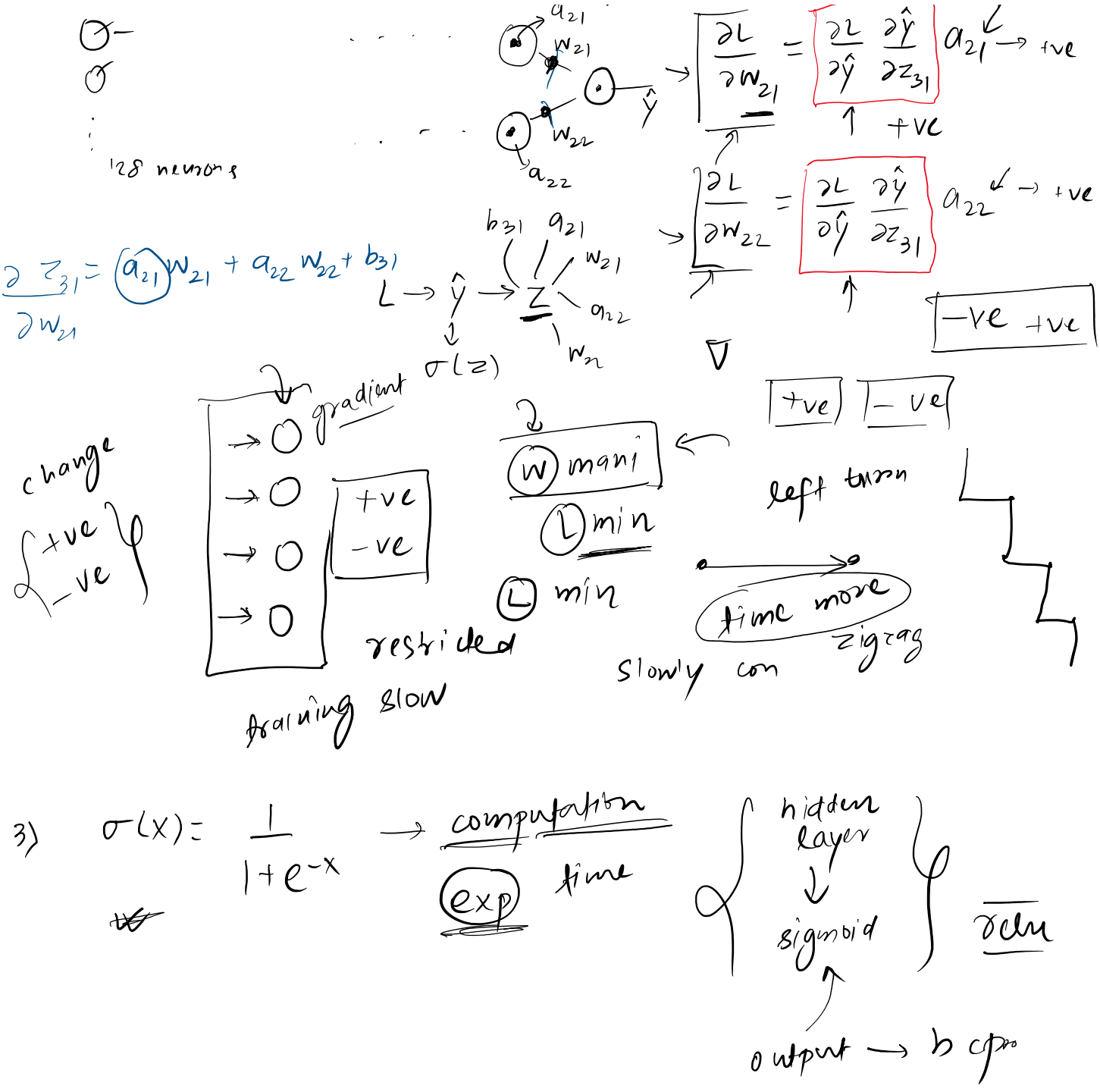
Normalized → mean = 0

+ve / -ve → Training slow

zero → Convergence problem

+ve -ve → $0 \rightarrow 1$

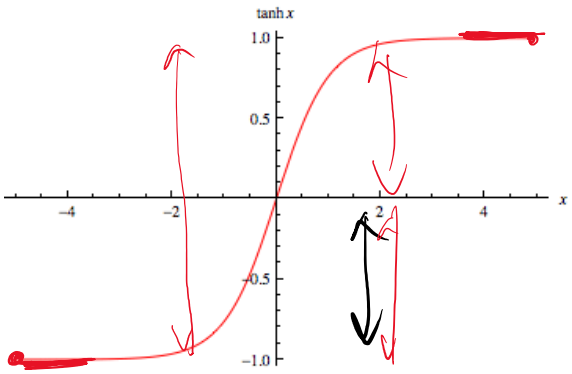



128 neurons
a_{21}
w_{21}
w_{22}
a_{22}
$\hat{y}$
$\frac{\partial L}{\partial w_{21}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_{31}} a_{21}$ → +ve
↑ +ve
$\frac{\partial L}{\partial w_{22}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_{31}} a_{22}$ → +ve
$\frac{\partial z_{31}}{\partial w_{21}} = a_{21} w_{21} + a_{22} w_{22} + b_{31}$
b_{31}
$L \to \hat{y} \to z$
$\sigma(z)$
-ve +ve
gradient
change
+ve
-ve
+ve
-ve
restricted
training slow
(W) many
(L) min
(L) min
+ve
-ve
left turn
time more
slowly con
zigzag
3) $\sigma(x) = \frac{1}{1+e^{-x}}$
computation
exp
time
hidden layer
sigmoid
relu
output → b cpu

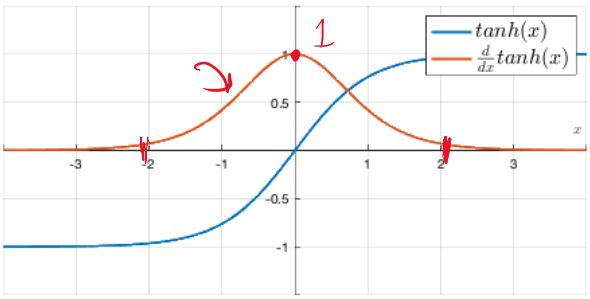
Tanh Activation Function

31 May 2022 14:50

(0,1) (-1,1)



0.25



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

$$f'(x) = (1 - \tanh^2(x))$$

Advantages

1) Non-linear
2) Differentiable
3) zero centered (+ve, -ve) → training faster sigmoid

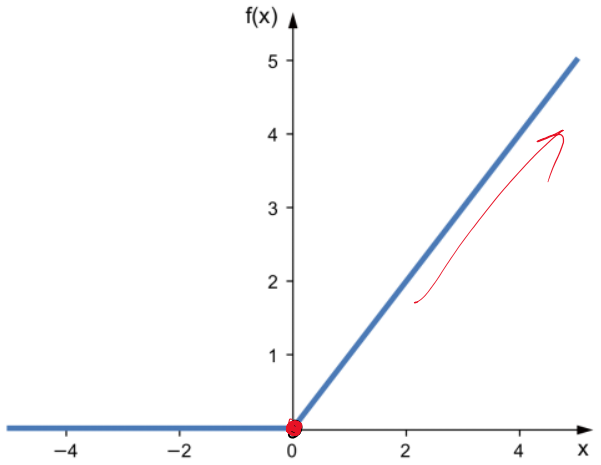
Disadvantage

→ saturating function → vanishing gradient prob

→ Computationally exp [exp] slow

Relu Activation Function

01 June 2022 16:43



Relu activation

$f(x) = \max(0, x)$ ← $f(x) = x$

Advantage

$f(x) = \max(0, x+1) - \max(0, x-1)$

1) Non-linear
2) Not saturated in the +ve region
3) Computationally inexpensive
4) Converge → faster → sigmoid
   ↳ tanh

Disadvantage

→ Differentiable

$x < 0 \rightarrow 0$

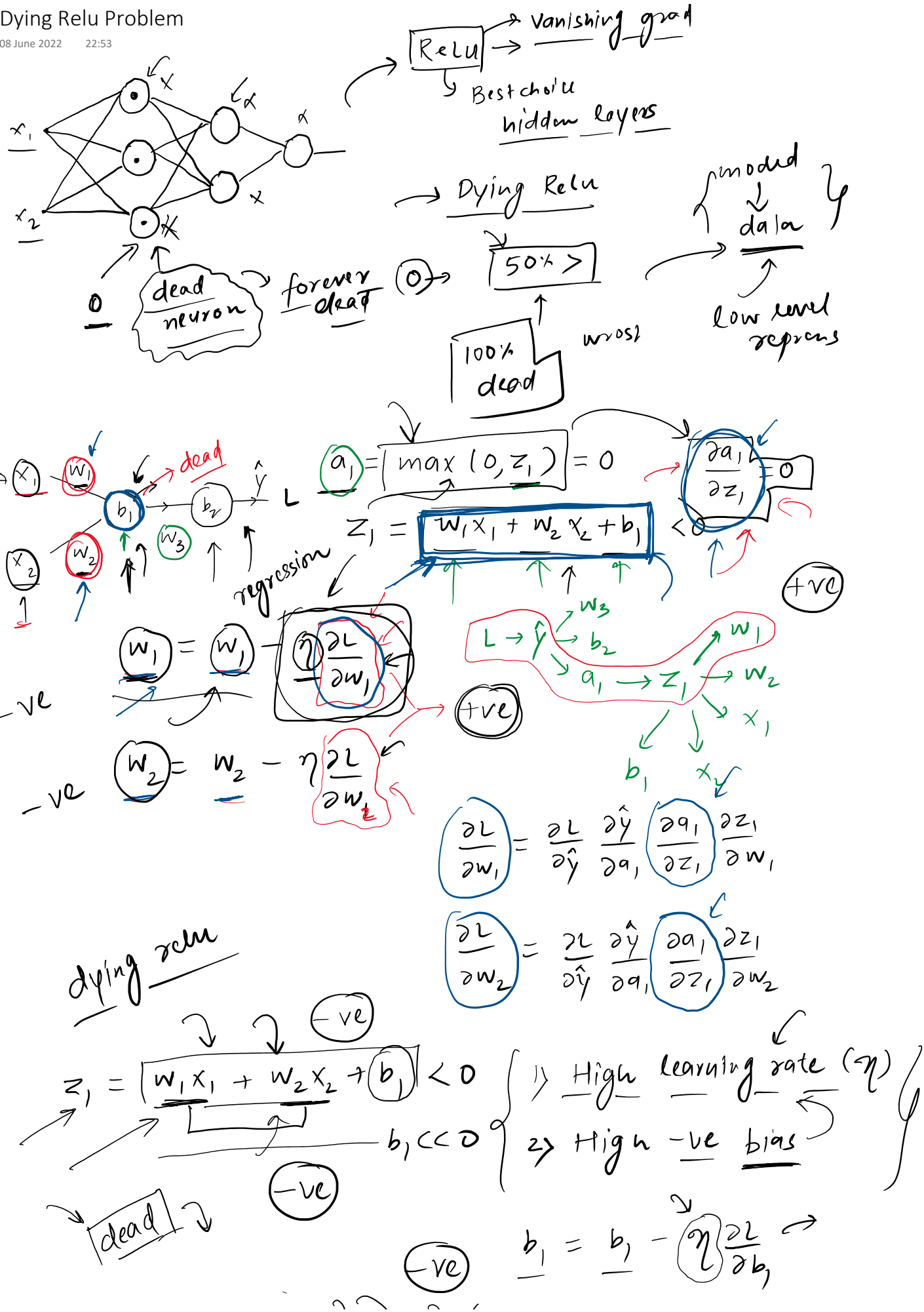
$x \geq 0 \rightarrow 1$

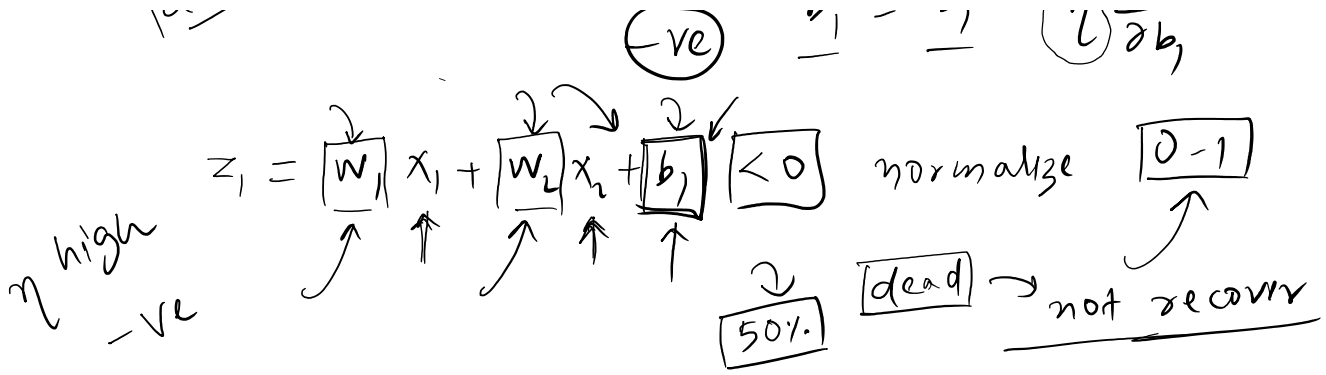
→ Non Zero centered → sigmoid → Batch Normalization
normalize →

↳ Dying RELU problem

Dying Relu Problem

08 June 2022 22:53



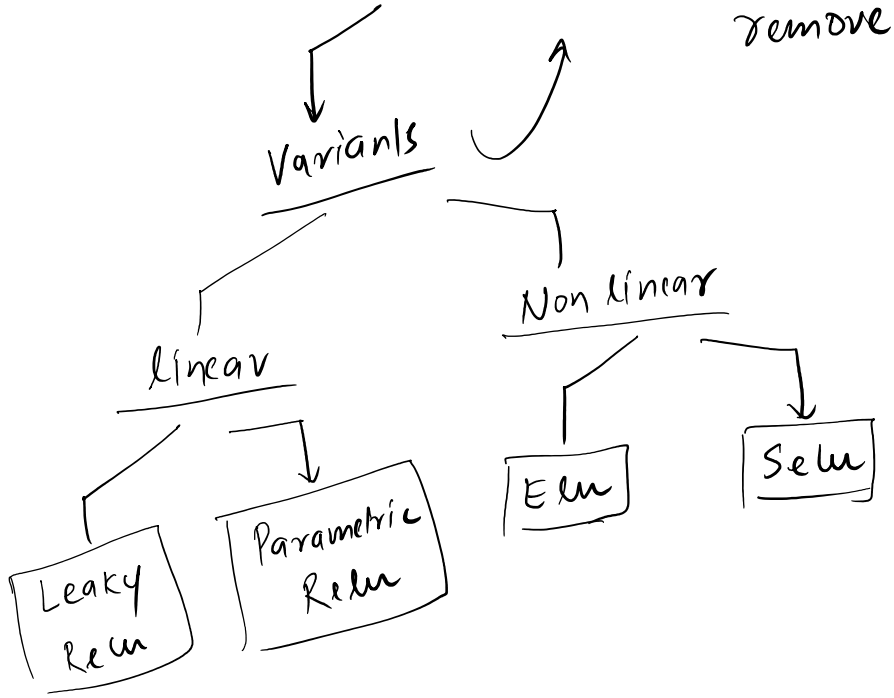


Solutions

→ Set low learning rate

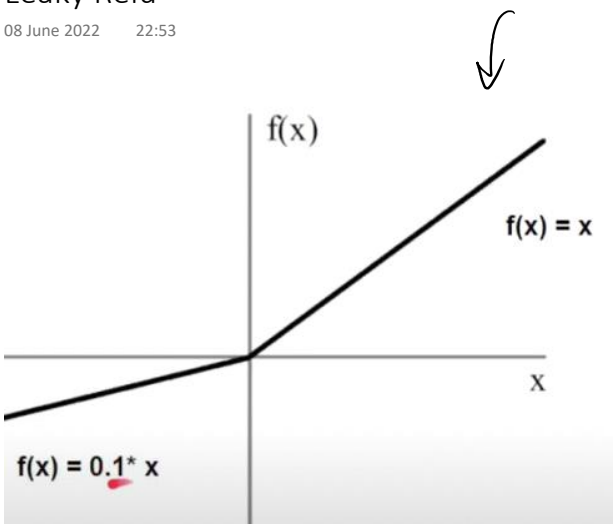
→ bias → +ve value → 0.01

→ Don't use relu → variants (+ve keep) / remove



Leaky Relu

08 June 2022 22:53



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

$z \geq 0 \rightarrow z$

$z < 0 \rightarrow \frac{1}{100} z$ (fraction of z)

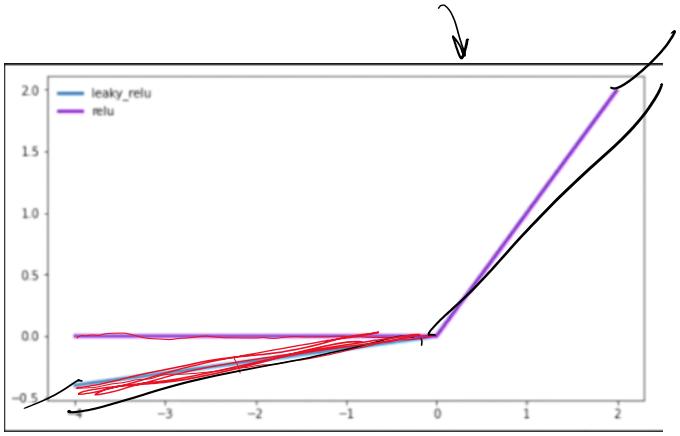
$w_1 = w_1 - \eta \frac{\partial L}{\partial w_1}$

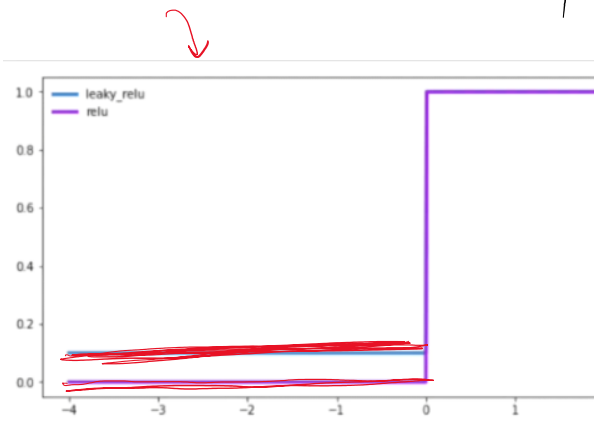
-ve (z) $\rightarrow \neq 0$ [0.01] z

↳ derivative

$f'(z)$ $z \geq 0 \rightarrow$ [1]

$z < 0 \rightarrow$ [0.01]





Advantages

→ Non-saturated → Unbounded

→ Easily computed

→ No dying relu problem

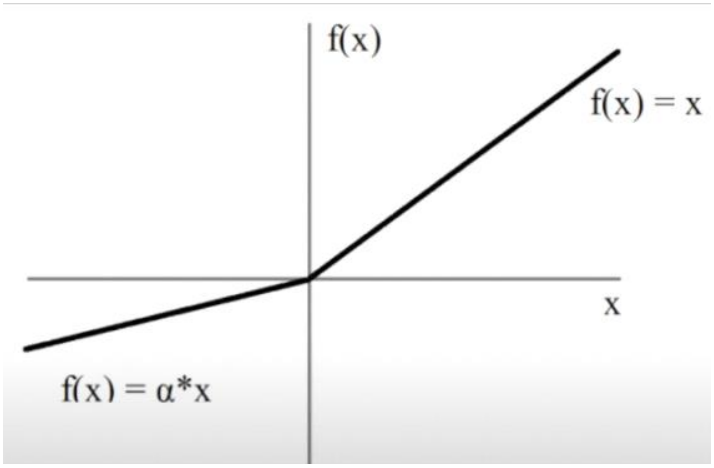
→ Close to 0 centered

Disadv

-ve/+ve

Parametric Relu

08 June 2022 22:53



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

0.01x

a → trainable parameter

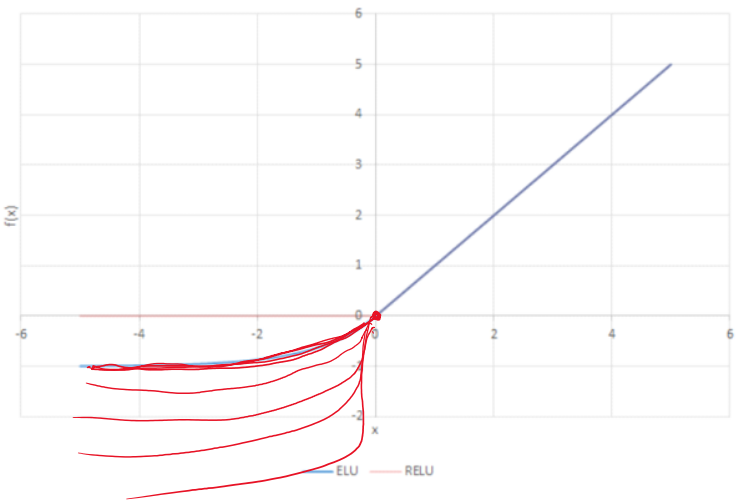
depend on data

flexibility

Elu - Exponential Linear Unit

09 June 2022 00:29

performance better Relu

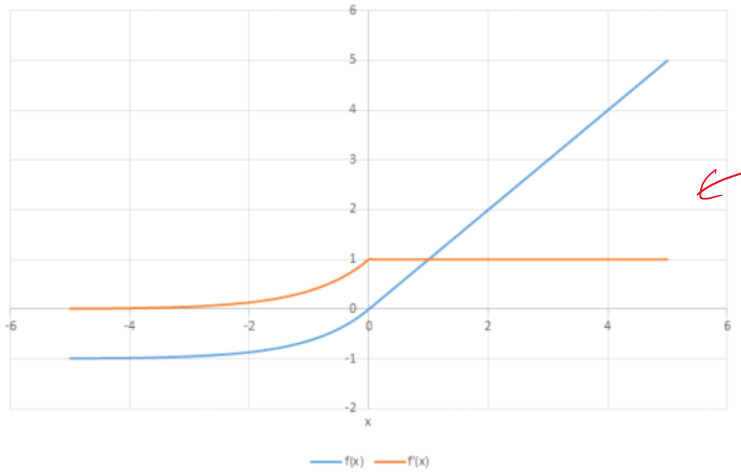


$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

constant

0.1 to 0.3

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



Advan

→ Close zero centered
↳ convergence faster

→ Better generalized

→ Dying relu

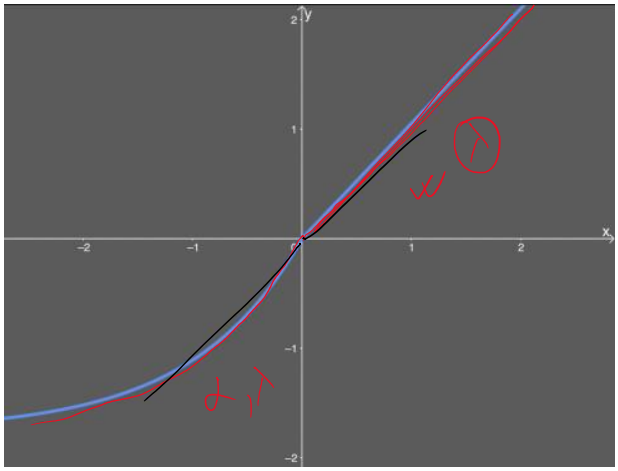
→ Always continous as well diffen

Disadv

→ computation expensive (ex)

Selu - Scaled Exponential Linear Unit

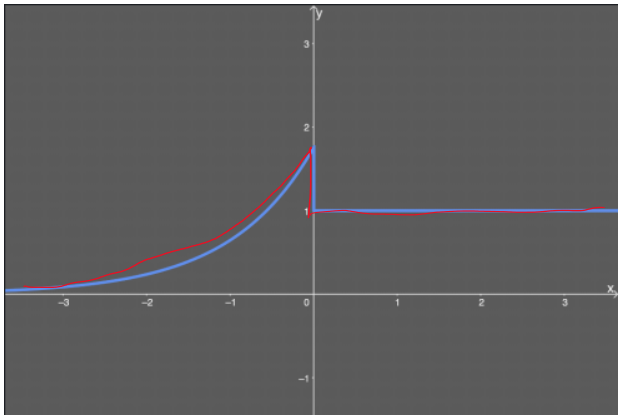
09 June 2022 00:29



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

$$\alpha \approx 1.6732632423543772848170429916717$$

$$\lambda \approx 1.0507009873554804934193349852946$$



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