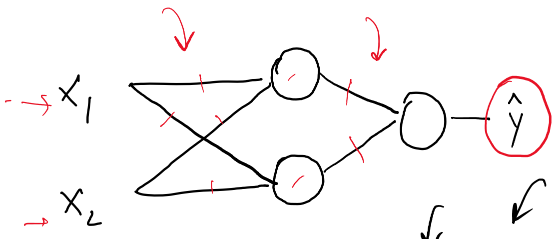
Jasin

Why Weight Initialization is Important?

22 June 2022 12:48



- → Vanishing GP
- → Exploding GP
- → Slow converge

কিসের Prob এর জন্য

2000's Deep learning Research stop হয়ে গিয়েছিলো।

How to train a NN:

1. Initialize the parameters
2. Choose an *optimization algorithm*
3. Repeat these steps:
    1. Forward propagate an input
    2. Compute the cost function
    3. Compute the gradients of the cost with respect to parameters using backpropagation
    4. Update each parameter using the gradients, according to the optimization algorithm

Prob এর কারণ

- Sigmoid Activ
- Wrong weight init

Wrong WI techniques

↓

What to do?

নিচে আমরা wrong weight initialization technique গুলো দেখবো।

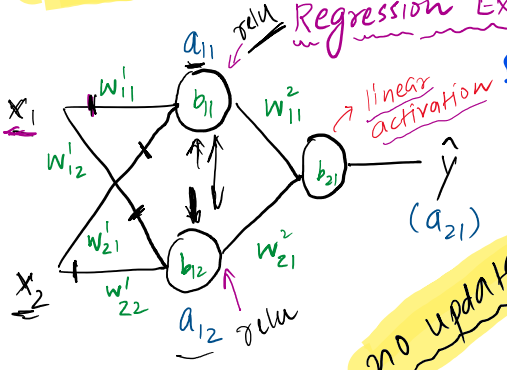
What not to do?

22 June 2022 12:49

Case1 → Zero Initialization

relu
tanh
sigmoid

Regression Example



Relu

$a_{11} = \max(0, z_{11}) = 0$

$z_{11} = w'_{11}x_1 + w'_{21}x_2 + b_{11} = 0$

$a_{12} = \max(0, z_{12}) = 0$

$z_{12} = w'_{12}x_1 + w'_{22}x_2 + b_{12} = 0$

$a_{11} = a_{12}$ → equal and 0

$w = 0, b = 0$

tanh

$a_{11} = \frac{e^{z_{11}} - e^{-z_{11}}}{e^{z_{11}} + e^{-z_{11}}} = 0$

$= \frac{e^0 - e^{-0}}{e^0 + e^{-0}}$

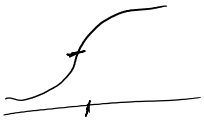
$= \frac{1-1}{1+1} = 0$

no update

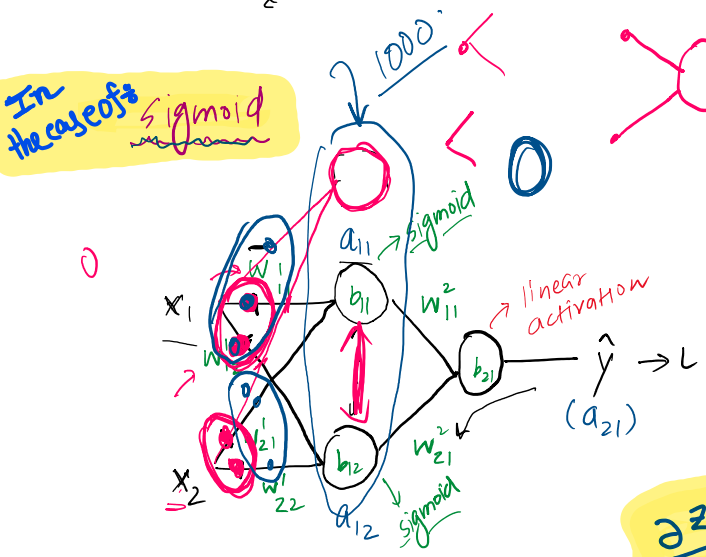
$w'_{11} = w'_{11} - \eta \frac{\partial L}{\partial w'_{11}}$

$w = 0$ No training will place

Code example



In the case of sigmoid



0

$a_{11} = \sigma(z_{11}) = 0.5$

$z_{11} =$

$a_{12} = 0.5$

$a_{11} = a_{12}$

Non-linear proof

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

$a_{11} = a_{12}$

$z_{11} = z_{12}$

0.5 হওয়ায় $a_{11} = a_{12}$

$\frac{\partial z_{11}}{\partial w_{11}} = x_1$

$\frac{\partial L}{\partial w'_{11}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{11}} \frac{\partial a_{11}}{\partial z_{11}} x_1$

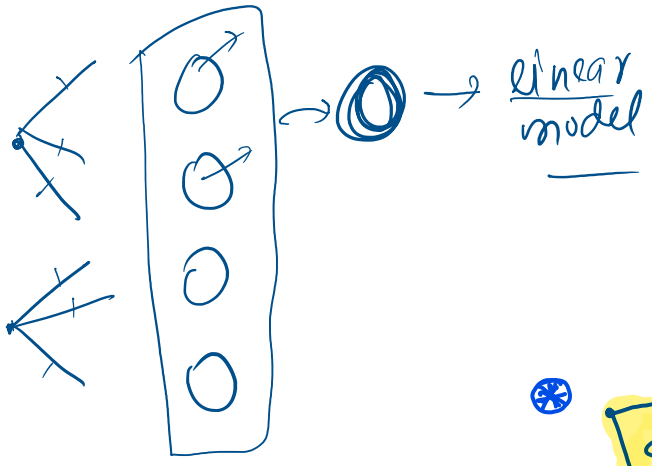
same

$\frac{\partial L}{\partial w'_{12}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{12}} \frac{\partial a_{12}}{\partial z_{12}} x_1$

same

$\frac{\partial L}{\partial w'_{21}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{11}} \frac{\partial a_{11}}{\partial z_{11}} x_2$

$\frac{\partial L}{\partial w'_{22}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{12}} \frac{\partial a_{12}}{\partial z_{12}} x_2$



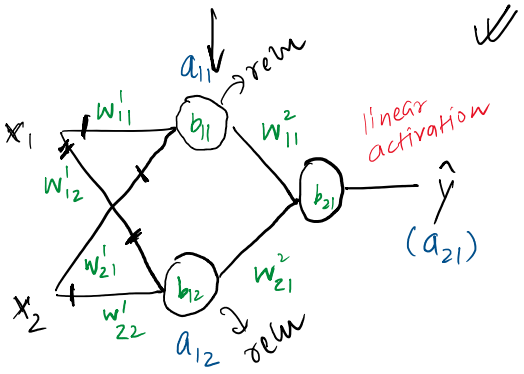
আগের, hidden layer এর neuron গুলো একটা neuron এর মতো আচরণ করবে। যদি w,b=0 and activation fn is sigmoid.

✱ see code example

Case two

Case2 → Non-0 constant value

$w = 0.5 \quad b = 0.5$ sigmoid → 0



$a_{11} = \max(0, z_{11})$ → non-zero

$z_{11} = w^1_{11} x_1 + w^1_{21} x_2 + b_{11} \neq 0$

$a_{12} = \max(0, z_{12}) \neq 0$

$z_{12} = w^1_{12} x_1 + w^1_{22} x_2 + b_{12} \neq 0$

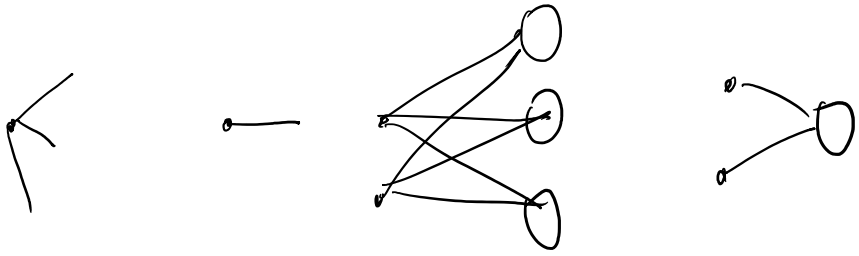
আগের কারণের মতো $z_{12} = z_{11}$ $\boxed{a_{11} = a_{12}}$ হওয়ায় derivate এ 1st 3টা constant হবে।

$$\frac{\partial L}{\partial w^1_{11}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{11}} \frac{\partial a_{11}}{\partial z_{11}} x_1$$

$$\frac{\partial L}{\partial w^1_{21}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{11}} \frac{\partial a_{11}}{\partial z_{11}} x_2$$

$$\frac{\partial L}{\partial w^1_{12}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{12}} \frac{\partial a_{12}}{\partial z_{12}} x_1$$

$$\frac{\partial L}{\partial w^1_{22}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_{12}} \frac{\partial a_{12}}{\partial z_{12}} x_2$$



hidden layer এর সবগুলো node একটা node এর মতো behave করবে।

If positive integer arguments are given (e.g., np.random.randn(2, 4)), it generates an array of shape (2, 4) filled with random floats sampled from the standard normal distribution.

random init

then, see code example

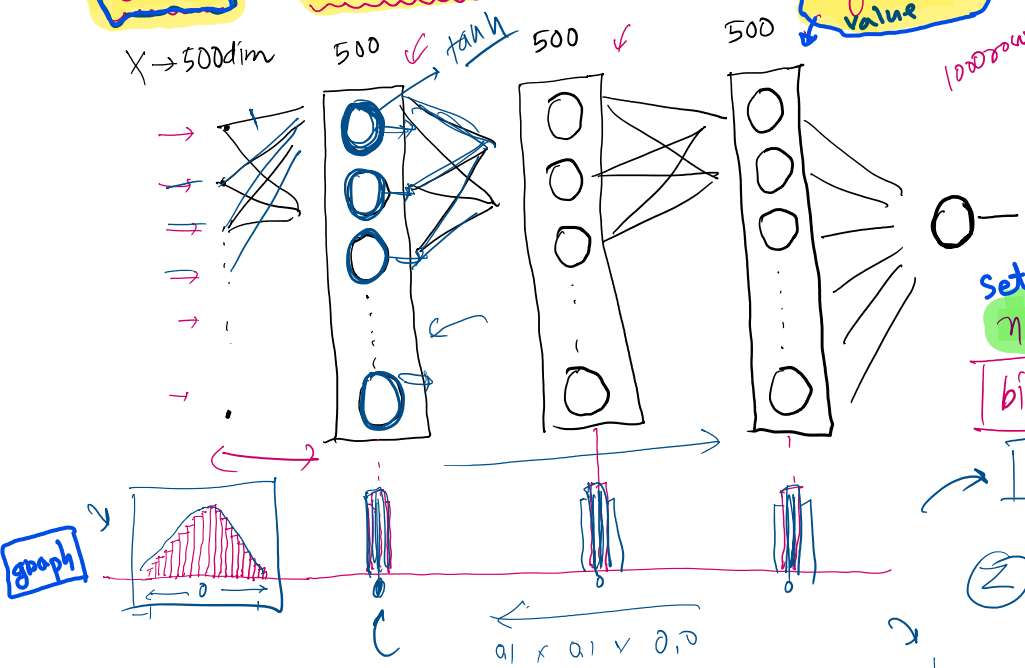
VGP sigmoid

Case-03

Case3 — Random Initialization

small / large random value → tanh / sigmoid

(1000, 500)



weight init → random → small values

set up

np.random.randn(500, 500) * 0.01

bias = 0

$x_i \rightarrow$ 0-1 → -1 to 1

$\sum w_i x_i$

0.0007

small number tanh

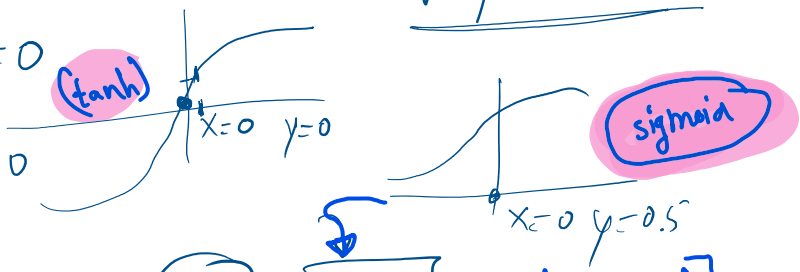
z is a small

very close to 0

small small ≈ 0

$w = w - \eta \frac{\partial L}{\partial w} \approx 0$

vanishing gradient problem



VGP → deep neural network

small weights → 0.0007

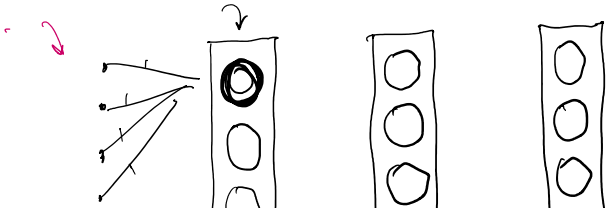
tanh → VGP

sigmoid → VGP

Relu → small values → slow convergence

Code example

*** Random Initialization (Large values)



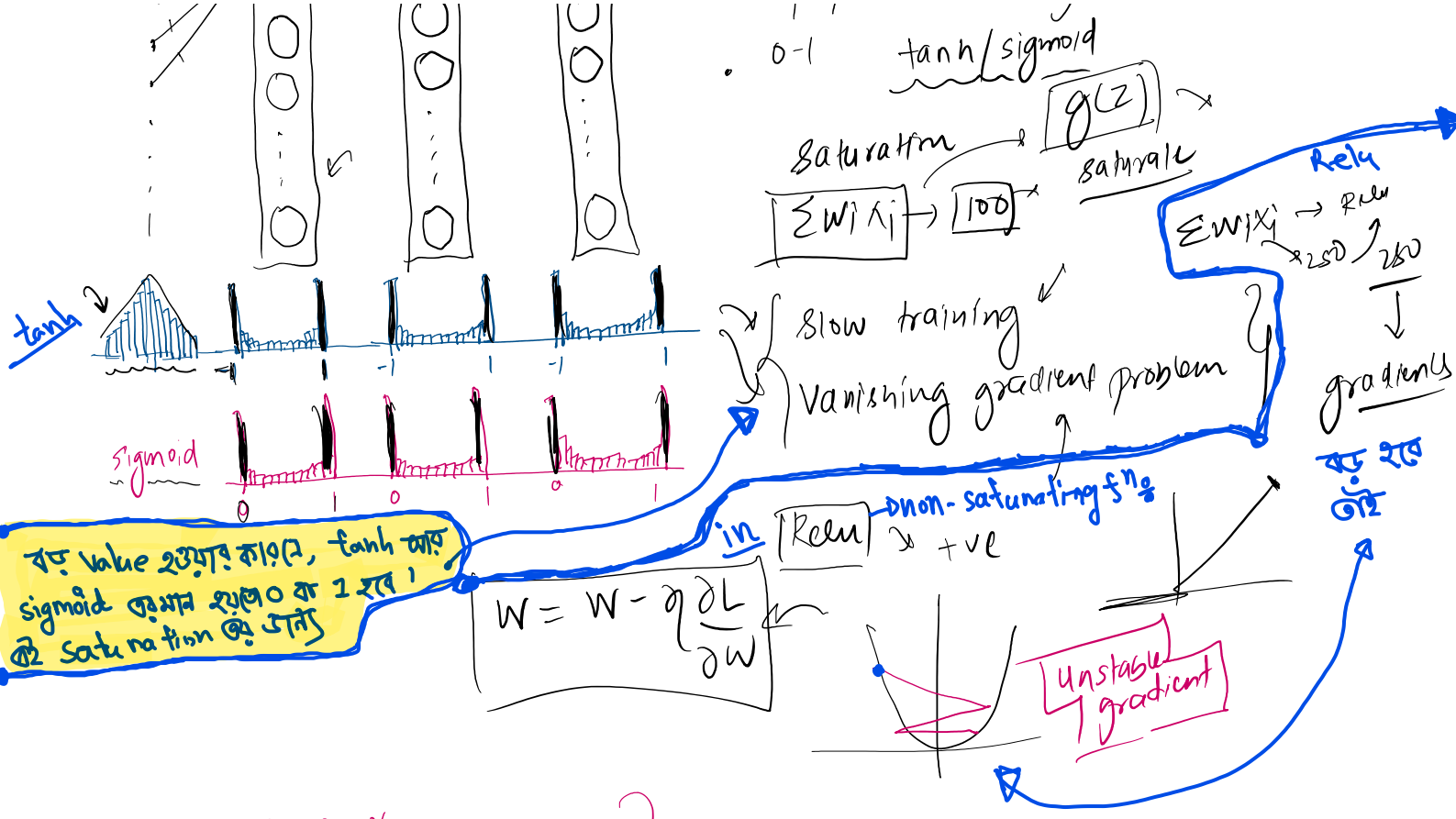
np.random.randn(500, 500)

$\sum w_i x_i$ → 500 → 100 → 500

0-1

tanh / sigmoid

0-1 এই range এই থাকলে আমরা কখনো বড় কোনো দিয়ে initi করতাম।



→ zero ✗

→ non-zero same ✗

→ small random ✓

→ large random ✗