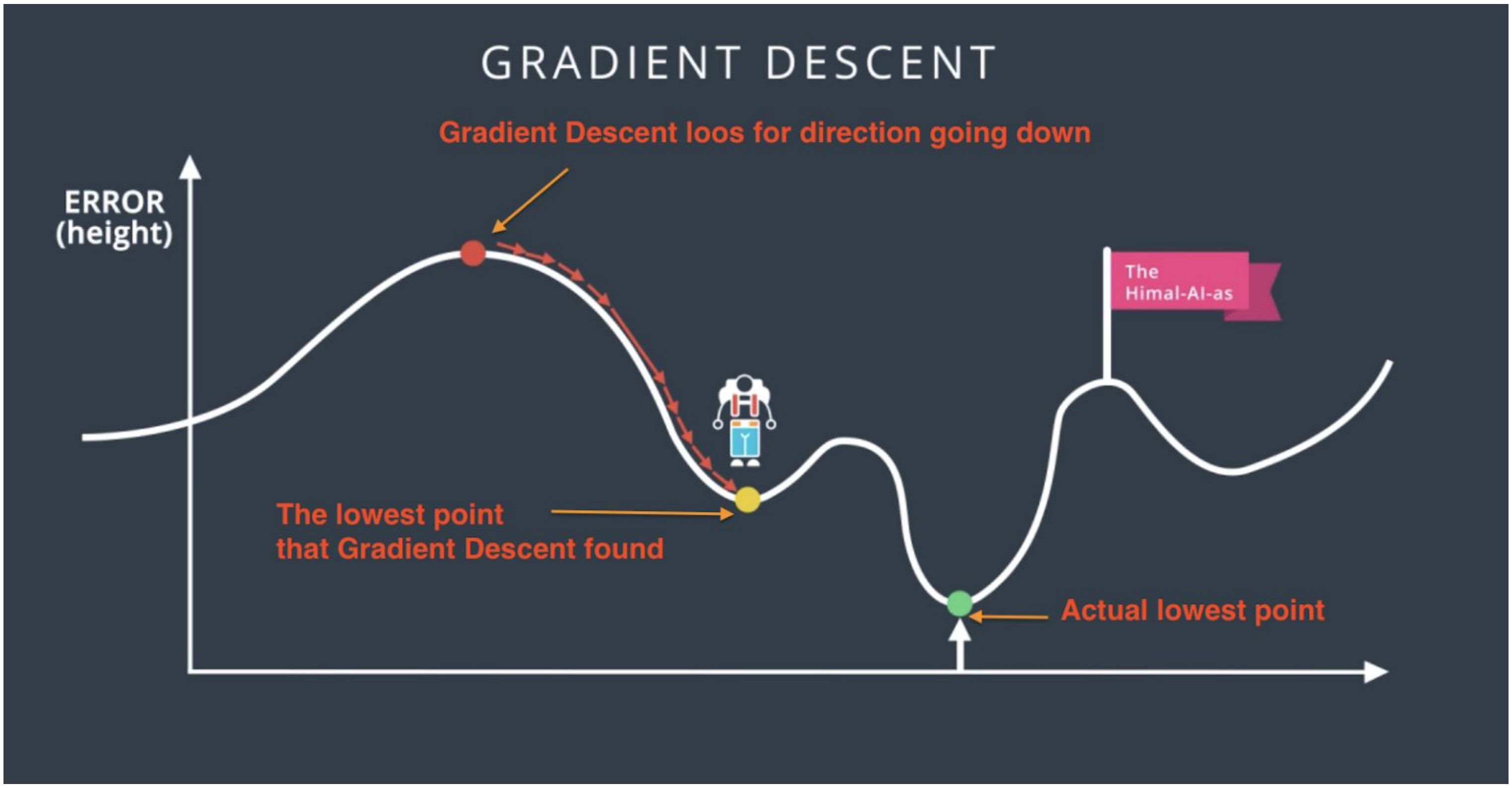
Note: This is the summary note from Udacity Introduction to Deep Learning with PyTorch

Problem that can occur with complex deep neural network

1. Problem : Local Minima

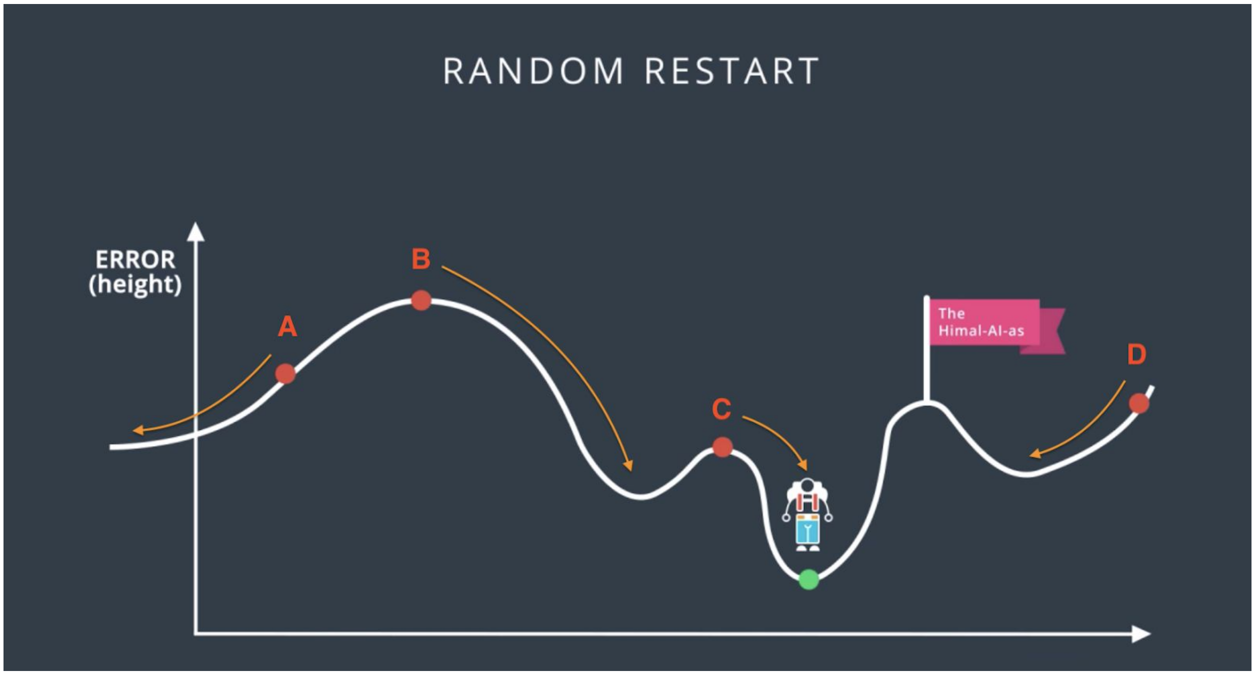
- Problem: cannot find the lowest point; Gradient descent in this case cannot find the actual lowest point.


GRADIENT DESCENT
Gradient Descent loos for direction going down
ERROR (height)
The Himal-AI-as
The lowest point that Gradient Descent found
Actual lowest point

- What could be the solution ?

Random Restart

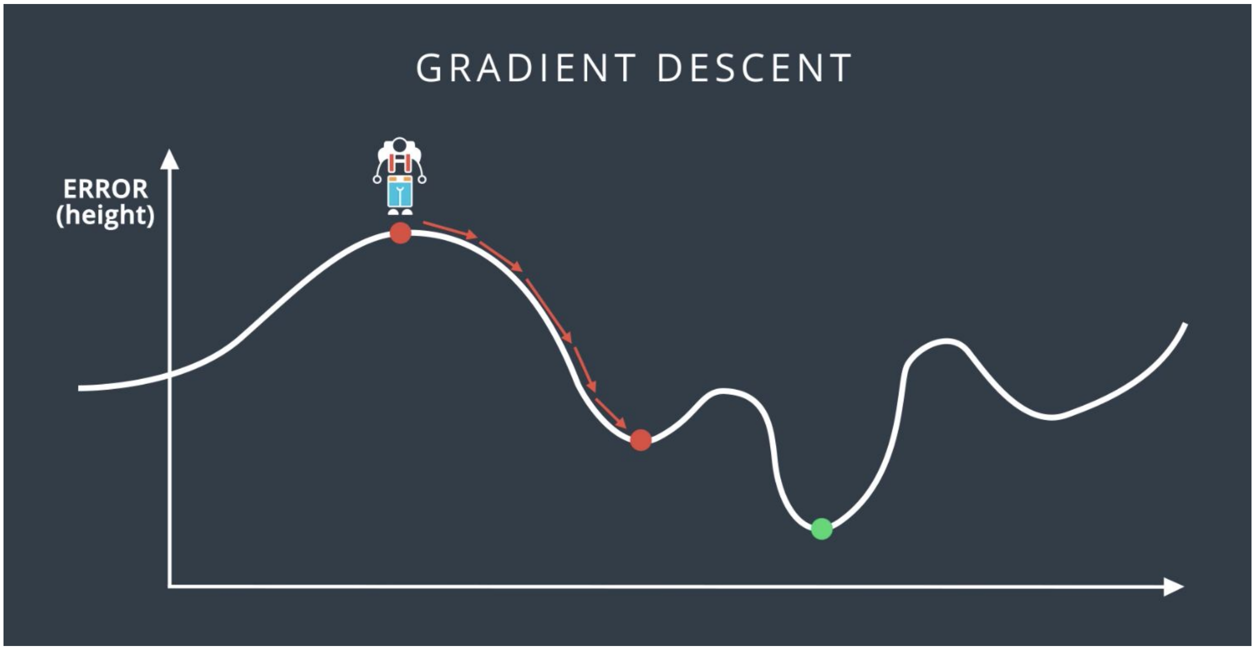
- One of the solutions to the local minima issue
- Choose a few different random places and do gradient descend from all of them, it helps increase probability of getting local minimum OR at least a pretty good local minimum



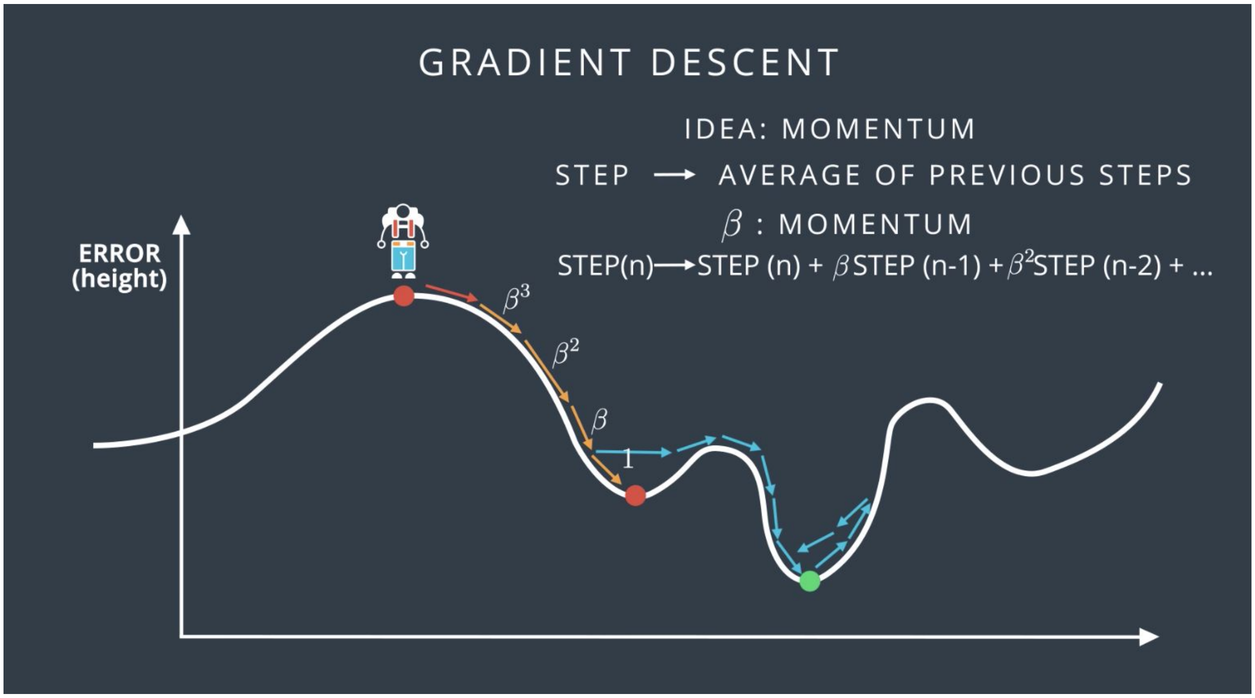
- A, B, C, D are the few points we chose to be random starts

Momentum

- Another solution to the local minima issue
- Analogy:
    - Walk faster with momentum and determination so that when we get stuck at low minimum we can go over the hump to find lower minium
- Neural Network
    - Problem: Normal gradient descent will have small step and will found a low minimum but cannot go find others low environment (it stuck)
    - For instance,
        - red arrow below represent small step
        - Red dot represent low minimum
        - Green dot represent lower minimum



- Solutions:
    - Look at average of last few steps when we reach the red dot (low minimum)
    - Multiple with momentum β



GRADIENT DESCENT

IDEA: MOMENTUM

STEP → AVERAGE OF PREVIOUS STEPS

β : MOMENTUM

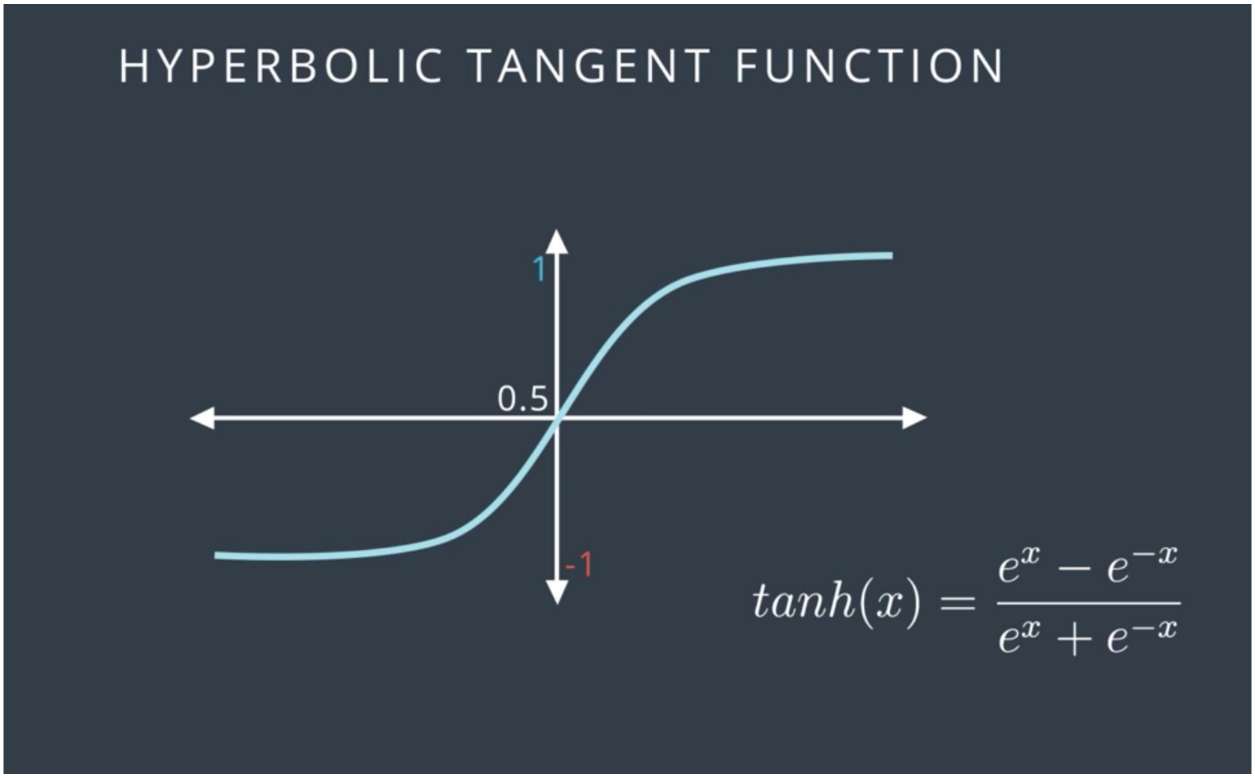
$\text{STEP}(n) \longrightarrow \text{STEP}(n) + \beta\,\text{STEP}(n-1) + \beta^2\,\text{STEP}(n-2) + \ldots$

- β is the momentum that attach to each step
- Notice β^1, β^2, β^3, β^4 , this mean the step that happens long time ago will matter less than the ones that happened recently
- Once we jump over the hump, and hit green dot, global minimum

2. Problem: Vanishing Gradient

- Sigmoid function has pretty curve flat so if we calculate point at way left or way right, this derivative is almost zero. Why is it an issue ?
- Derivative tell us the direction to move.
- In perceptron, it means the derivative sigmoid function is small; and the product of these small value leads to tiny values.
- This makes the training difficult because grading give us very tiny changes to make on the weights

- **Solution to vanishing gradient**
    - **Choose other activation functions**
        - Hyperbolic Tangent function



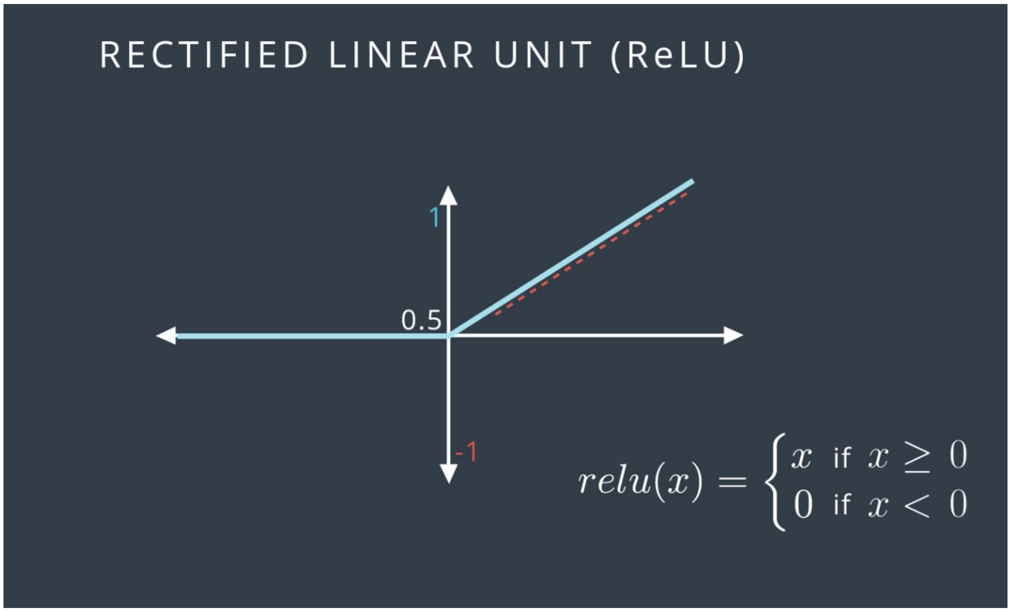
HYPERBOLIC TANGENT FUNCTION

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

The range is from -1 and 1, so the derivatives are larger

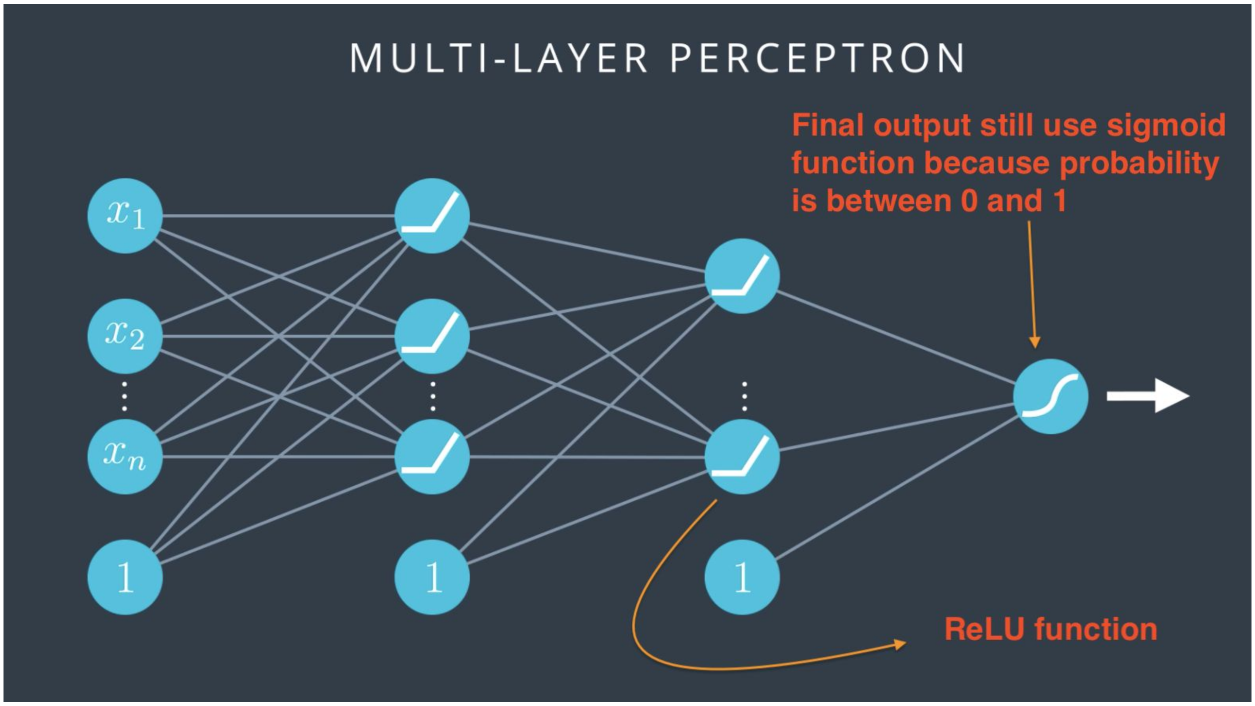
    - Rectified Linear Unit (ReLU)

- Return same value if the value is positive
- Return 0 if the value is negative



RECTIFIED LINEAR UNIT (ReLU)

$$relu(x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$


MULTI-LAYER PERCEPTRON
Final output still use sigmoid function because probability is between 0 and 1
x_1
x_2
x_n
1
1
1
ReLU function

==

Mini Summary

- Problems occur in deep neural network
  - Local minima
  - Vanishing gradient
- Solution for local minima: random start
- Solutions for vanishing gradient
  - Hyperbolic Tangent function
  - ReLU function