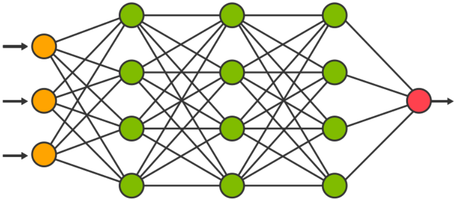
Introduction to Neural Networks

ML Instruction Team, Fall 2022

CE Department
Sharif University of Technology



TO DO

TO DO

TO DO

TO DO

TO DO

TO DO

TO DO

TO DO

TO DO

TO DO

Gradient Clipping

- In case of a large or small gradient, what will happen?

Gradient Clipping

- In case of a large or small gradient, what will happen?
- Gradient descent either won't change our position or will send us far away.

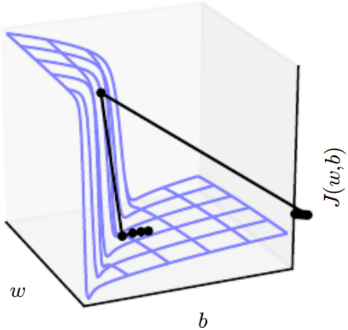


Figure: The problem of large gradient value [1].

Gradient Clipping

- Solve this problem simply by clipping gradient
- Two approaches to do so:
  - Clipping by value
  - Clipping by norm

Gradient Clipping by value

- Set a max (α) and min (β) threshold value
- For each index of gradient $\boldsymbol{g}_i$ if it is lower or greater than your threshold clip it:

$$
\begin{aligned}
&\text{if } \boldsymbol{g}_i > \alpha : \\
&\qquad \boldsymbol{g}_i \leftarrow \alpha \\
&\text{else if } \boldsymbol{g}_i < \beta : \\
&\qquad \boldsymbol{g}_i \leftarrow \beta
\end{aligned}
$$

- Clipping by value will not save gradient direction but still works well in practice.
- To preserve direction use clipping by norm.

Gradient Clipping by norm

- Clip the norm $\|\boldsymbol{g}\|$ of the gradient $\boldsymbol{g}$ before updating parameters:

$$\text{if } \|\boldsymbol{g}\| > v :$$
$$\boldsymbol{g} \leftarrow \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} v$$

  v is the threshold for clipping which is a hyperparameter.
- Gradient clipping saves the direction of gradient and controls its norm.

Gradient Clipping

- The effect of gradient clipping:

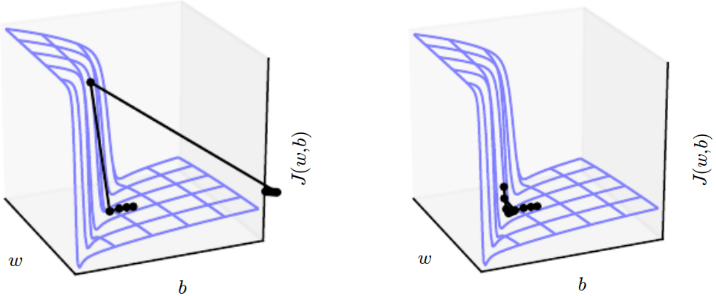


Figure: The "cliffs" landscape (left) without gradient clipping and (right) with gradient clipping [1].

Weight Initialization

- Is initialization really necessary?
- What are the impacts of initialization?

Weight Initialization

- Is initialization really necessary?
- What are the impacts of initialization?
- A bad initialization may increase convergence time or even make optimization diverge.
- How to initialize?
  - Zero initialization
  - Random initialization

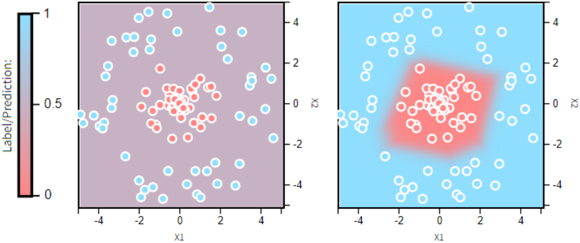
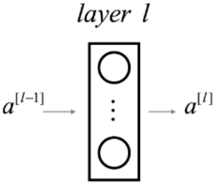


Figure: The output of a three layer network after about 600 epoch. (left) using a bad initialization method and (right) using an appropriate initialization [2].

Weight Initialization

Let's review some notations before we continue:

$$\begin{cases} n^{[l]} := \text{layer } l \text{ neurons number}, \\ W^{[l]} := \text{layer } l \text{ weights}, \\ b^{[l]} := \text{layer } l \text{ biases}, \\ a^{[l]} := \text{layer } l \text{ outputs} \end{cases}$$



Weight Initialization: Zero Initialization

Zero Initialization method:

$$\begin{cases} W^{[l]} = 0, \\ b^{[l]} = 0 \end{cases}$$

Weight Initialization: Zero Initialization

Zero Initialization method:

$$\begin{cases} W^{[l]} = 0, \\ b^{[l]} = 0 \end{cases}$$

- Simple but perform very poorly. (why?)

Weight Initialization: Zero Initialization

Zero Initialization method:

$$\begin{cases} W^{[l]} = 0, \\ b^{[l]} = 0 \end{cases}$$

- Simple but perform very poorly. (why?)
- Zero initialization will lead each neuron to learn the same feature
- This problem is known as network failing to break symmetry
- In fact any constant initialization suffers from this problem.

Weight Initialization: Zero Initialization

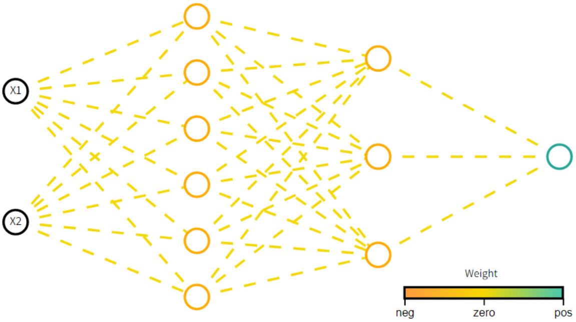


Figure: As we can see network has failed to break symmetry. There has been no improvement in weights after about 600 epochs of training [2].

- We need to break symmetry. How? using randomness.

Weight Initialization: Random Initialization

Simple Random Initialization:

$$\begin{cases} W^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2\right), \\ b^{[l]} = 0 \end{cases}$$

Weight Initialization: Random Initialization

Simple Random Initialization:

$$\begin{cases} W^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2\right), \\ b^{[l]} = 0 \end{cases}$$

- It depends on standard deviation (σ) value
- If it choose carefully, will perform well for small networks
- One can use $\sigma = 0.01$ as a best practice.
- But still has problems with deeper networks.
- Too small/large value for σ will lead to vanishing/exploding gradient problem.

Weight Initialization: Random Initialization

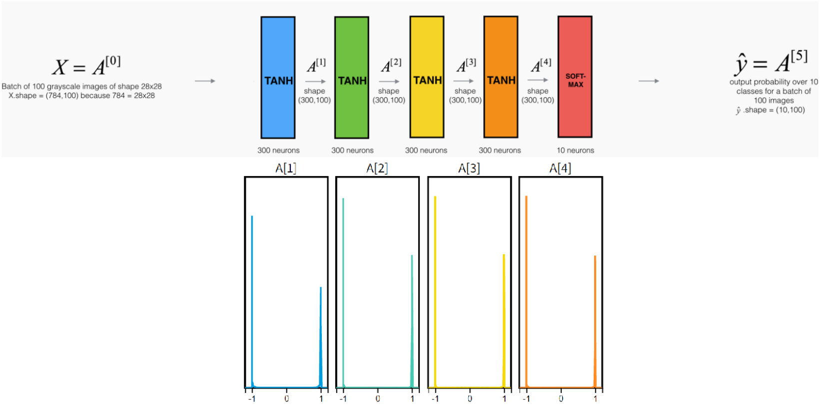


Figure: The problem of normal initialization. On the top, you can see the model architecture, and on the bottom, you can see the density of each layer's output. Model has trained on MNIST dataset for 4 epoch. Weights are initialized randomly from $\mathcal{N}(0, 1)$ [2].

Weight Initialization: Random Initialization

- How to have a better random initialization?
- We need to follow these rules:
  - keep the mean of the activations zero.
  - keep the variance of the activations same across every layer.
- How to do so?

Weight Initialization: Random Initialization

- How to have a better random initialization?
- We need to follow these rules:
  - keep the mean of the activations zero.
  - keep the variance of the activations same across every layer.
- How to do so?

Xavier Random Initialization:

$$\begin{cases} W^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{1}{n^{[l]}}\right), \\ b^{[l]} = 0 \end{cases}$$

(this method works fine for *tanh*, and you can read about why it works at [2].)

Weight Initialization: Random Initialization

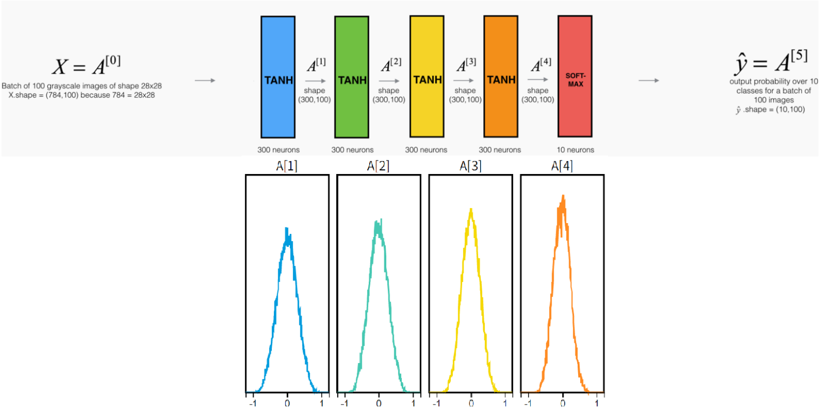


Figure: Vanishing gradient is no longer problem using Xavier initialization. Model has trained on MNIST dataset for 4 epoch. [2].

Weight Initialization

- We discussed weight initialization on previous slides.
- A god initialization will help model on vanishing/exploding gradient problem.
- Xavier method works well with *tanh* activation function.
  - If you use $ReLU$ activation use He initialization:

He Initialization:

$$\begin{cases} W^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{2}{n^{[l]}}\right), \\ b^{[l]} = 0 \end{cases}$$

Various GD types

Final Notes

Thank You!

Any Question?

References

I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*.
MIT Press, 2016.
http://www.deeplearningbook.org.

K. Katanforoosh and D. Kunin, "Initializing neural networks," 2019.
https://www.deeplearning.ai/ai-notes/initialization/.