Deep Learning for Computer Vision

Dr. Konda Reddy Mopuri
Mehta Family School of Data Science and Artificial Intelligence
IIT Guwahati
Aug-Dec 2022

So far in the class..

- Brief introduction to ML

So far in the class..

- Brief introduction to ML
- Artificial neuron models, Perceptron

So far in the class..

- Brief introduction to ML
- Artificial neuron models, Perceptron
- MLP, CNNs and different families of architecture

So far in the class..

- Brief introduction to ML
- Artificial neuron models, Perceptron
- MLP, CNNs and different families of architecture
- (today) Some of the important training aspects of CNNs

Data preprocessing for Computer vision

- Mean subtraction (e.g. AlexNet: $32 \times 32 \times 3$, VGG: $1 \times 1 \times 3$)

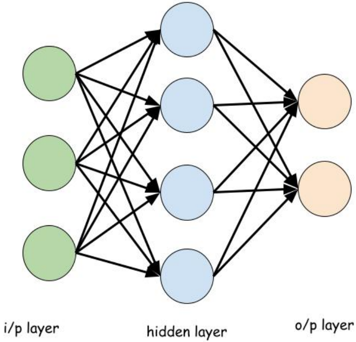
Data preprocessing for Computer vision

- Mean subtraction (e.g. AlexNet: $32 \times 32 \times 3$, VGG: $1 \times 1 \times 3$)
- Mean subtraction and division by standard deviation per channel (e.g. ResNet)

Data preprocessing for Computer vision

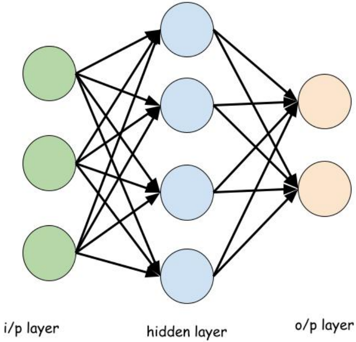
- Mean subtraction (e.g. AlexNet: $32 \times 32 \times 3$, VGG: $1 \times 1 \times 3$)
- Mean subtraction and division by standard deviation per channel (e.g. ResNet)
- PCA or whitening are not common

Weight Initialization



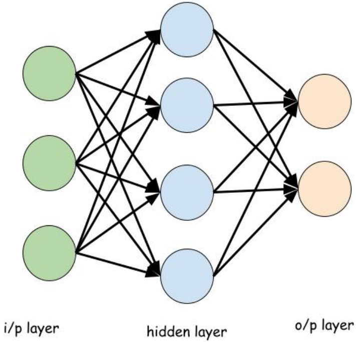
- What if all the parameters are initialized to zero?

Weight Initialization



- What if all the parameters are initialized to zero?
- Or, a different constant?

Weight Initialization



- What if all the parameters are initialized to zero?
- Or, a different constant?
- Leads to a failure mode (often known as the 'symmetry' problem)

Weight Initialization

- How about randomly initializing?
  `W = 0.001 * np.random.randn(`d_l, d_{l-1}`)`

Figure credits: Dr Justin Johnson, U Michigan

Weight Initialization

- How about randomly initializing?
  `W = 0.001 * np.random.randn(`d_l, d_{l-1}`)`
- Okay for the shallow nets

Figure credits: Dr Justin Johnson, U Michigan

Weight Initialization

- How about randomly initializing?
  `W = 0.001 * np.random.randn(`d_l, d_{l-1}`)`
- Okay for the shallow nets
- However, the dynamic range of the activations at later layers goes on shrinking $\rightarrow$ activations tend to zero at deeper layers (e.g. 6 layer MLP with a tanh nonlinearity)

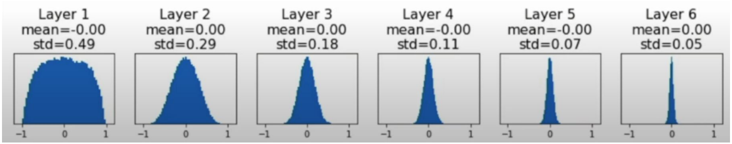
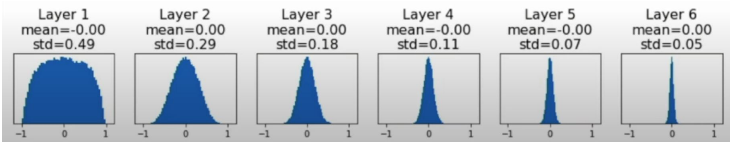


Figure credits: Dr Justin Johnson, U Michigan

Weight Initialization

- How about randomly initializing?
  `W = 0.001 * np.random.randn(`d_l, d_{l-1}`)`
- Okay for the shallow nets
- However, the dynamic range of the activations at later layers goes on shrinking $\rightarrow$ activations tend to zero at deeper layers (e.g. 6 layer MLP with a tanh nonlinearity)



- All zero gradients, no learning!

Figure credits: Dr Justin Johnson, U Michigan

Xavier Initialization

- `W = 0.001 * np.random.randn(`d_l, d_{l-1}`)/np.sqrt(`d_{l-1}`)`

Figure credits: Dr Justin Johnson, U Michigan

Xavier Initialization

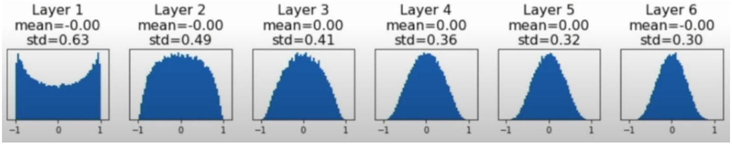
- W = 0.001 * np.random.randn(d_l, d_{l-1})/np.sqrt(d_{l-1})
- 

Figure credits: Dr Justin Johnson, U Michigan

Xavier Initialization

- We prefer the o/p to have similar variance as the input

Xavier Initialization

- We prefer the o/p to have similar variance as the input
- Consider a single layer, $y = Wx$, i.e. $y_i = \sum_{j=1}^{d_{l-1}} x_j \cdot w_j$

Xavier Initialization

- We prefer the o/p to have similar variance as the input
- Consider a single layer, $y = Wx$, i.e. $y_i = \sum_{j=1}^{d_{l-1}} x_j \cdot w_j$
- var$(y_i) = d_{l-1} \cdot var(x_i \cdot w_i)$ (Assuming w_i and x_i are i.i.d)

Xavier Initialization

- We prefer the o/p to have similar variance as the input
- Consider a single layer, $y = Wx$, i.e. $y_i = \sum_{j=1}^{d_{l-1}} x_j \cdot w_j$
- var$(y_i) = d_{l-1} \cdot var(x_i \cdot w_i)$ (Assuming w_i and x_i are i.i.d)
- var$(y_i) = d_{l-1} \cdot \left(E({x_i}^2) \cdot E({w_i}^2) - E(x_i)^2 \cdot E(w_i)^2 \right)$ (Assuming x and w are independent)

Xavier Initialization

- We prefer the o/p to have similar variance as the input
- Consider a single layer, $y = Wx$, i.e. $y_i = \sum_{j=1}^{d_{l-1}} x_j \cdot w_j$
- var$(y_i) = d_{l-1} \cdot var(x_i \cdot w_i)$ (Assuming w_i and x_i are i.i.d)
- var$(y_i) = d_{l-1} \cdot \left(E({x_i}^2) \cdot E({w_i}^2) - E(x_i)^2 \cdot E(w_i)^2 \right)$ (Assuming x and w are independent)
- var$(y_i) = d_{l-1} \cdot$ var$(x_i) \cdot$ var(w_i) Assuming (x_i and w_i are zero-mean)

Xavier Initialization

- We prefer the o/p to have similar variance as the input
- Consider a single layer, $y = Wx$, i.e. $y_i = \sum_{j=1}^{d_{l-1}} x_j \cdot w_j$
- var$(y_i) = d_{l-1} \cdot var(x_i \cdot w_i)$ (Assuming w_i and x_i are i.i.d)
- var$(y_i) = d_{l-1} \cdot \left(E({x_i}^2) \cdot E({w_i}^2) - E(x_i)^2 \cdot E(w_i)^2\right)$ (Assuming x and w are independent)
- var$(y_i) = d_{l-1} \cdot$ var$(x_i) \cdot$ var(w_i) Assuming (x_i and w_i are zero-mean)
- $\rightarrow$ var$(w_i) = \frac{1}{d_{l-1}}$

Weight Initialization with ReLU activations

- Kaiming or MSRA initialization

Figure credits: Dr Justin Johnson

Weight Initialization with ReLU activations

- Kaiming or MSRA initialization
- std=sqrt$(2/d_{l-1})$

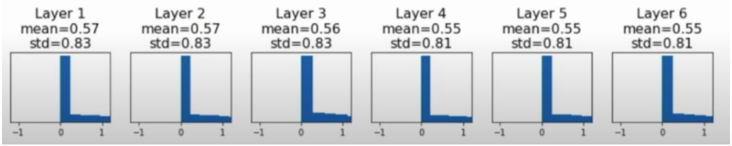


Figure credits: Dr Justin Johnson