Introduction to Deep Learning for Computer Vision

Adhyayan '23 - ACA Summer School
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Lecture 5

Training NNs: Tricks of the Trade!

Weights Initialization

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\qquad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}
$$

Weights Initialization

Gradient descent algorithm

$$\text{repeat until convergence } \{$$

What values of thetas shall we start with?

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(\text{for } j = 1 \text{ and } j = 0)$$

$$\}$$

Random Initialization

Small Variance

$$W \sim \mathcal{N}(0, \sigma^2)$$

Zero Mean

Xavier/Glorot Initialization

$$W^{(l)} \sim \mathcal{N}(\mu = 0, \sigma^2 = \sqrt{\frac{1}{n^{(l-1)}}})$$

Weights in the
l-th layer.

Number of
nodes in the
(l-1)-th layer.

$$W^{(l)} \sim \mathcal{N}(\mu = 0, \sigma^2 = \sqrt{\frac{1}{n^{(l-1)} + n^{l}}})$$

Fan in

Fan out

Uniform Xavier/Glorot Initialization

$$W^{(l)} \sim \mathcal{U}\left(-\sqrt{\frac{6}{n^{(l-1)}+n^{(l)}}}, \sqrt{\frac{6}{n^{(l-1)}+n^{(l)}}}\right)$$

Read More: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

Kaiming Initialization

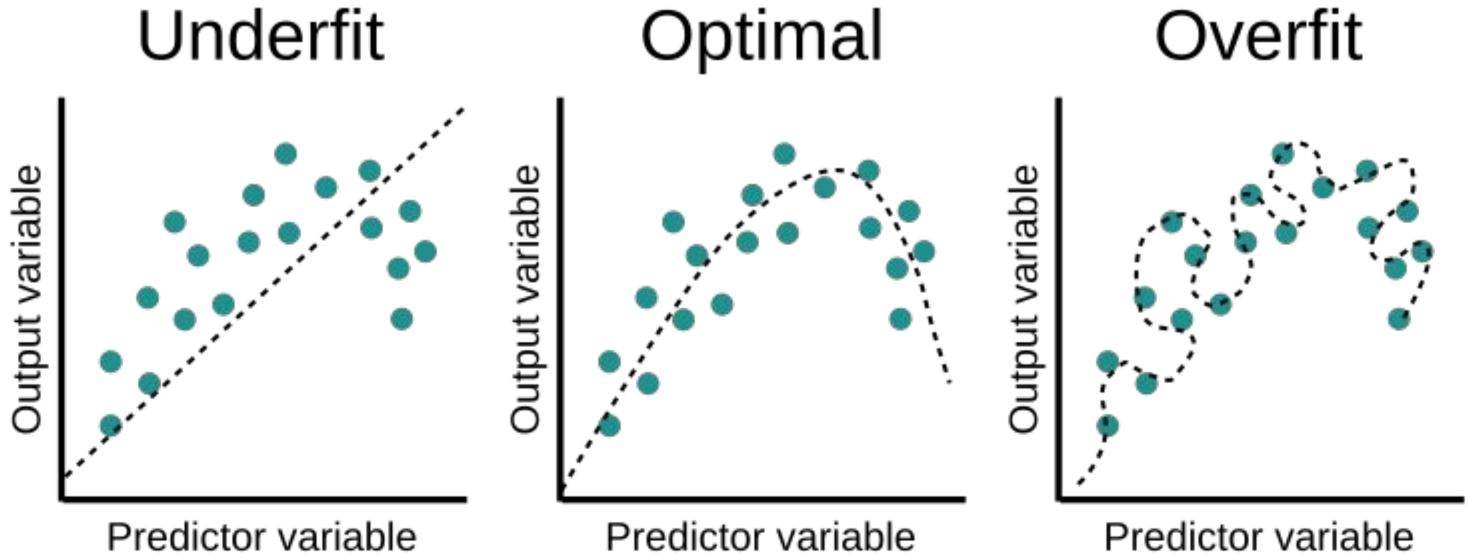
$$W^{(l)} \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{1}{n^{(l)}})$$

Read More: https://arxiv.org/pdf/1502.01852.pdf

Choosing the Right Initialization

- Depends on network architecture and activations.
- Trial and Error.

Regularization: Weight Decay



Weights of the Neural Net should be **regulated** so that we can control overfitting.

Regularization: Weight Decay

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)};W) - y^{(i)})^2$$

Regularization: Weight Decay

Regularization Constant

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)};W) - y^{(i)})^2 + \frac{\lambda}{2}||W||_2^2$$

Regularization Term

L_2 Regularization

Regularization Constant

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)}; W) - y^{(i)})^2 + \frac{\lambda}{2}W^TW$$

Regularization Term

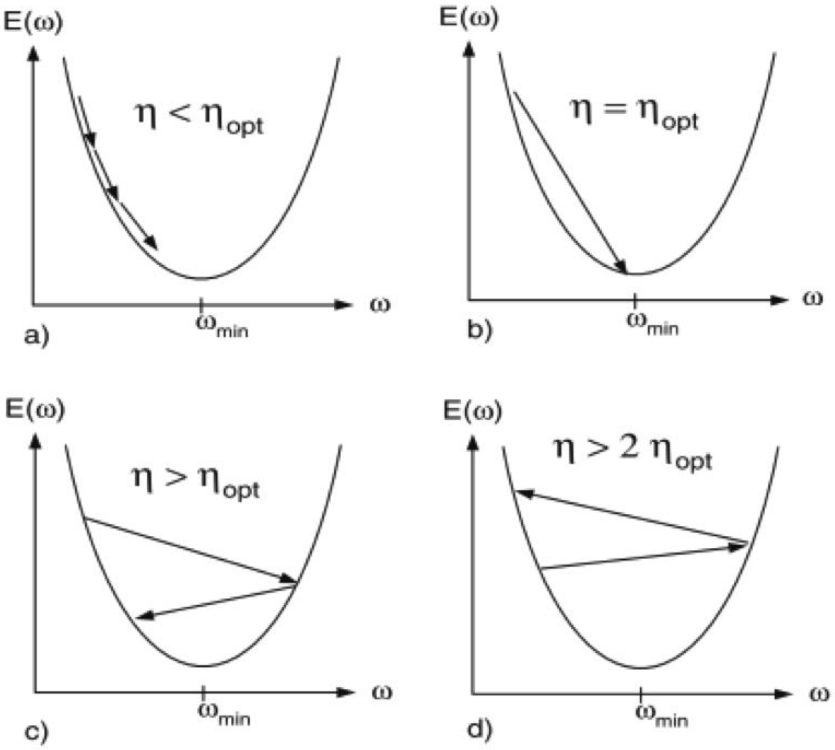
L_1 Regularization

Regularization Constant

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}\left(f(x^{(i)}; W) - y^{(i)}\right)^2 + \lambda||W||_1$$

Regularization Term

Learning Rate Scheduling



Common LR Scheduling Techniques

1. **Step Decay**: reduces the learning rate by a factor at specific epochs or after a certain number of iterations.

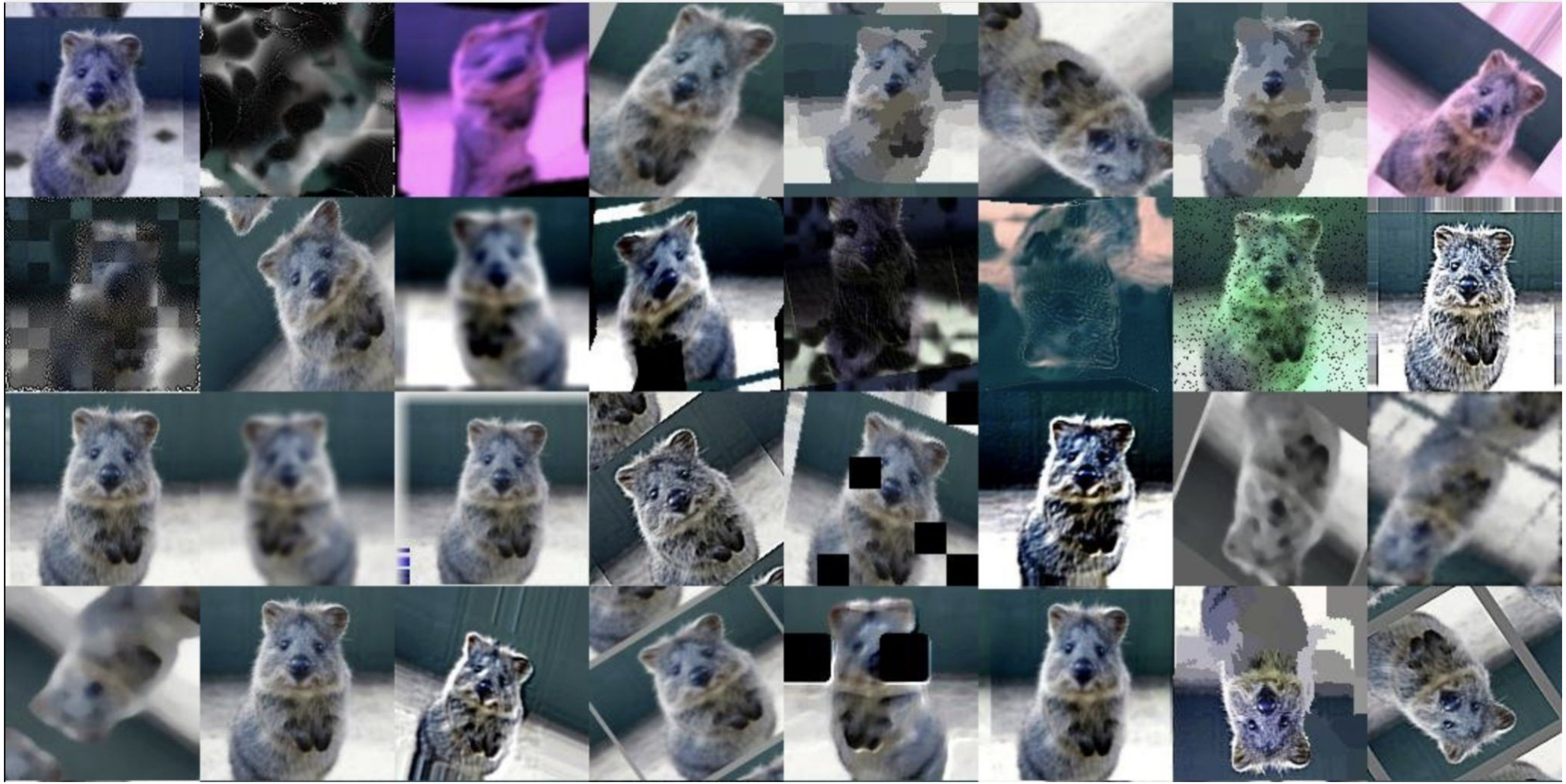
Common LR Scheduling Techniques

1. **Step Decay**: reduces the learning rate by a factor at specific epochs or after a certain number of iterations.
2. **Exponential Decay**: reduces the learning rate exponentially over time.

Common LR Scheduling Techniques

1. **Step Decay**: reduces the learning rate by a factor at specific epochs or after a certain number of iterations.
2. **Exponential Decay**: reduces the learning rate exponentially over time.
3. **Adaptive LR Scheduling**: adjust the LR based on factors such as the validation loss or gradient information. Example:
   a. **ReduceLROnPlateau**: reduces the learning rate when the validation loss plateaus
   b. **Cyclical Learning Rates**: cyclically vary the learning rate within a predefined range.

Data Augmentation



Data Augmentation

1. Random **transformations** or modifications to the training data.

Data Augmentation

1. Random **transformations** or modifications to the training data.
2. **Increases** training data size with same labels.

Data Augmentation

1. Random **transformations** or modifications to the training data.
2. **Increases** training data size with same labels.
3. Transformations can be:
    a. **Image Augmentation**: Random flips, rotations, translations, scaling, and cropping of images.
    b. **Color Jittering**: Randomly adjusting brightness, contrast, saturation, or hue of images.
    c. **Gaussian Noise**: Adding random Gaussian noise to the input data.

Data Normalization

- **What is it? Scaling** and **shifting** the input data to ensure it has a *consistent range and distribution*.

Data Normalization

- **What is it? Scaling** and **shifting** the input data to ensure it has a *consistent range and distribution*.
- **Why is it needed?** Stabilizes the learning process, improves convergence, and makes the network less sensitive to differences in input magnitudes.

Common Data Normalization Techniques

- **Mean Subtraction**: Subtracting the mean of the data from each input feature, resulting in zero mean. $\hat{x}_i = x_i - \mu$ $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$

Common Data Normalization Techniques

- **Mean Subtraction**: Subtracting the mean of the data from each input feature, resulting in zero mean. $\hat{x}_i = x_i - \mu$ $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
- **Standardization**: Scaling the data to have zero mean and unit variance by subtracting the mean and dividing by the standard deviation. $\hat{x}_i = \frac{x_i - \mu}{\sigma}$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Common Data Normalization Techniques

- **Mean Subtraction**: Subtracting the mean of the data from each input feature, resulting in zero mean. $\hat{x}_i = x_i - \mu \qquad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
- **Standardization**: Scaling the data to have zero mean and unit variance by subtracting the mean and dividing by the standard deviation. $\hat{x}_i = \frac{x_i - \mu}{\sigma}$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2}$$

- **Min-Max Scaling**: Scaling the data to a specific range (e.g., [0, 1]) by subtracting the minimum value and dividing by the range (maximum - minimum).

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Batch Normalization

- normalize the activations of each layer by **subtracting the batch mean** and **dividing by the batch standard deviation**.

Batch Normalization

- normalize the activations of each layer by **subtracting the batch mean** and **dividing by the batch standard deviation**.
- Benefits: Improved training stability

Batch Normalization

- normalize the activations of each layer by **subtracting the batch mean** and **dividing by the batch standard deviation**.
- Benefits: Improved training stability, Reduced Sensitivity to Initialization

Batch Normalization

- normalize the activations of each layer by **subtracting the batch mean** and **dividing by the batch standard deviation**.
- Benefits: Improved training stability, Reduced Sensitivity to Initialization, Regularization Effect

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Paper: https://arxiv.org/abs/1502.03167

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming
- **Weights Decay**: L2 or L1 Regularization

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming
- **Weights Decay**: L2 or L1 Regularization
- **LR Scheduling**: Step, Exponential, Adaptive

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming
- **Weights Decay**: L2 or L1 Regularization
- **LR Scheduling**: Step, Exponential, Adaptive
- **Data Augmentation**: Image, Color, Noise

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming
- **Weights Decay**: L2 or L1 Regularization
- **LR Scheduling**: Step, Exponential, Adaptive
- **Data Augmentation**: Image, Color, Noise
- **Data Normalization**: Mean, Standardization, Min-Max

Summary

- **Weights Initialization**: Random, Xavier/Glorot, Kaiming
- **Weights Decay**: L2 or L1 Regularization
- **LR Scheduling**: Step, Exponential, Adaptive
- **Data Augmentation**: Image, Color, Noise
- **Data Normalization**: Mean, Standardization, Min-Max
- **Batch Normalization**

Next Week: Advanced Deep Learning