CS 316: Introduction to Deep Learning

Linear Regression
Week 4

Dr Abdul Samad

Lecture Outline

- Linear Regression Model
- Single Layer Neural Network
- Evaluating Linear Regression Model
- Training Linear Regression Model
- Optimization
  - Gradient Descent
  - Learning Rate
  - Mini Batch Stochastic Gradient Descent

House Price Prediction

Predict price of the house based on its features.

The features in consideration are:

- Total Area
- Number of Bedrooms
- Number of Bathrooms

Figure Source: [House Price Prediction]

Simple Linear Model

The simple linear model makes the following assumptions:

- Area, number of bedrooms, and number of bathrooms are key factors influencing house price and are denoted by x_1, x_2, x_3.
- The sale price $\hat{y}$ is the weighted sum of the key factors and the bias.

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

where w_1, w_2 and w_3 are the weights and b is the bias term.

Linear Model

Given n dimensional input $\mathbf{x} = [x_1, x_2, \cdots, x_n]^\top$, the Linear Model has n weights $\mathbf{w} = [w_1, w_2, \cdots, w_n]^\top$ and bias b. The output $\hat{y}$ is the weighted sum of the inputs and the bias.

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots w_n x_n + b$$

In vectorized notation, the output $\hat{y}$ can be expressed as the following:

$$\hat{y} = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

Single Layer Neural Network

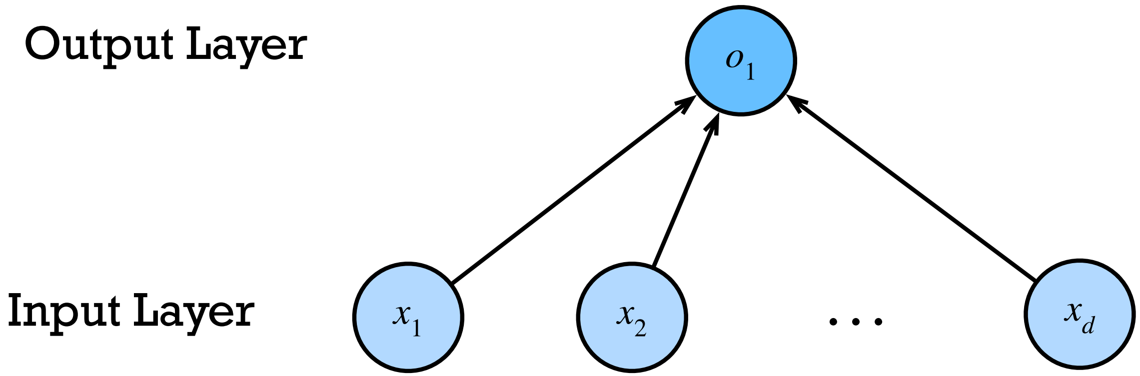
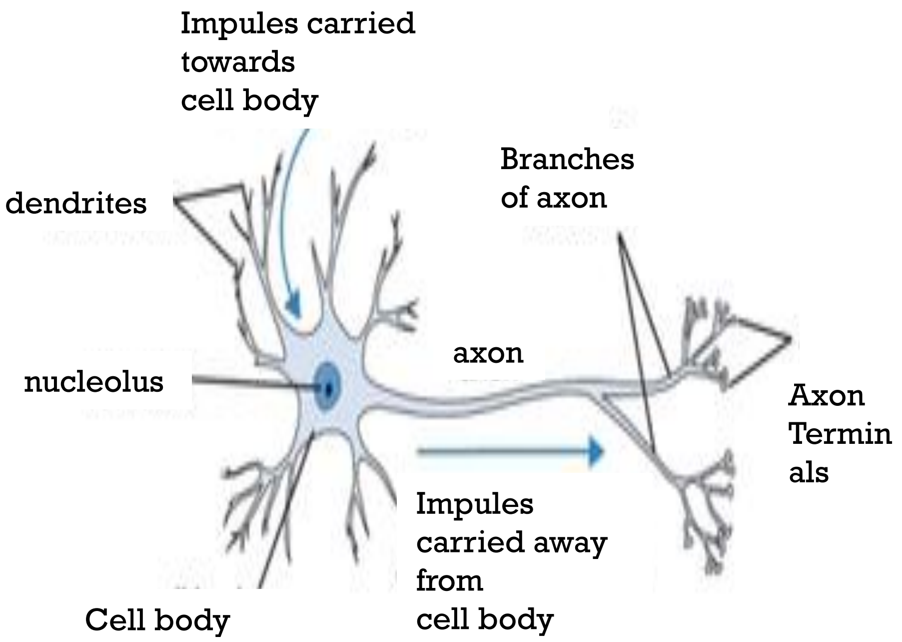


Figure Source: [Single Neuron]

Biological Neuron versus Artificial Neural Network



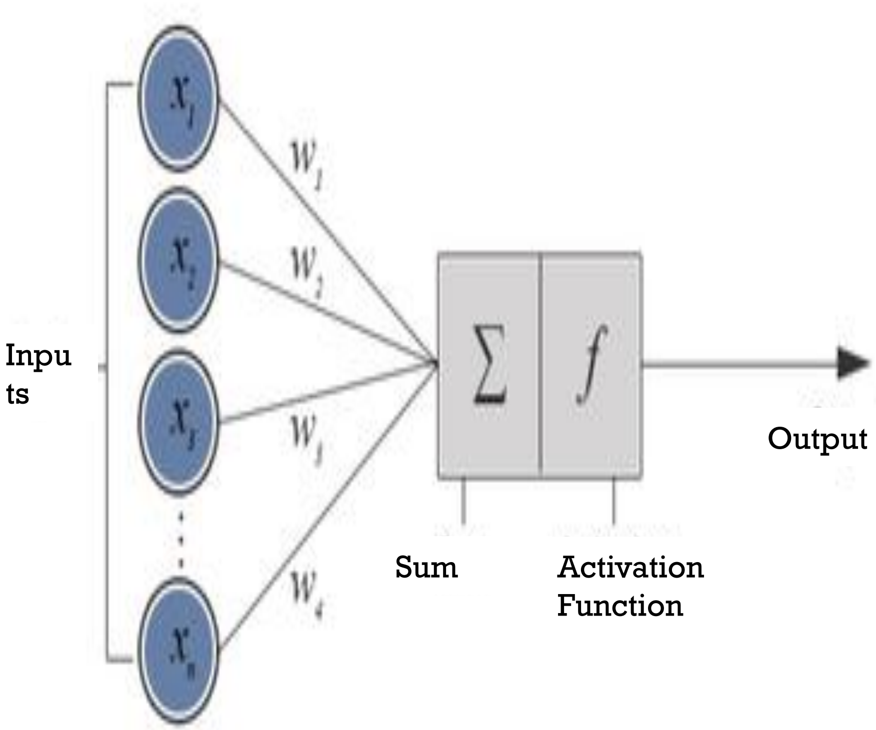


Figure Source: [Biological-Neuron-versus-Artificial-Neural-Network]

Evaluating Accuracy of the Model

- Compare the actual and predicted values. For example, compare the actual price of the house to the expected price of a house.
- Let y be the truth value, and $\hat{y}$ be the predicted value, we can compare the loss.

$$l(y, \hat{y}) = (\hat{y} - y)^2$$

- $l(y, \hat{y})$ is called the square loss.

Training Data

$$\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}]^\top$$

where $\mathbf{x_i}$ is the i^{th}input vector i.e. i^{th} house features.

$\mathbf{x_i^j}$ is the j^{th}feature of the i^{th} input vector. i.e. Total Area of the House.

$$\mathbf{y} = [y_1, y_2, \cdots, y_n]^\top$$

where y_i is the i^{th}output value i.e. i^{th} house price.

Training Loss of a Linear Regression Model

$$l_i(\mathbf{w}, b) = \frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

$$l_i(\mathbf{w}, b) = \frac{1}{2}(\hat{y}_i - y_i)^2$$

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l_i(\mathbf{w}, b)$$

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$$

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(\hat{y}_i - y_i)^2$$

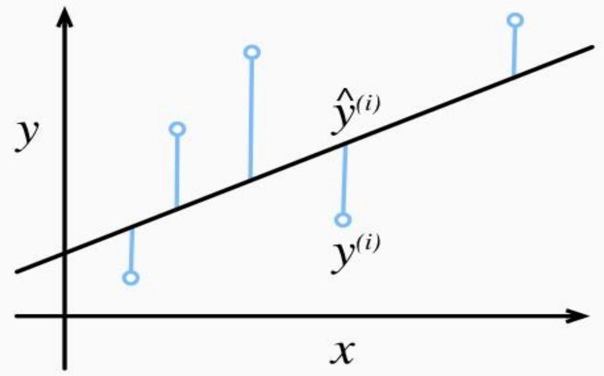


Fig. 3.1.1 Fit data with a linear model.

Figure Source: [Fit Data with linear model]

Vectorized Training Loss

- Concatenate a column of ones to the input and bias into the weights.

$$\mathbf{X} = [1 \quad \mathbf{X}], \mathbf{w} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}$$

- Vectorized Loss function is defined as

$$L(\mathbf{w}) = \frac{1}{n}||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

- Minimize the loss

$$\mathbf{w}^\star = argmin_{\mathbf{w}} L(\mathbf{w})$$

Closed Form Solution

$$\text{Compute } \frac{\delta L(\mathbf{w})}{\delta \mathbf{w}} \text{ where } L(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

Closed Form Solution

$$\text{Compute } \frac{\delta L(\mathbf{w})}{\delta \mathbf{w}} \text{ where } L(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

$$\mathbf{a} = \mathbf{X}\mathbf{w}$$

$$\mathbf{b} = \mathbf{a} - \mathbf{y}$$

$$z = ||\mathbf{b}||^2$$

$$\frac{\delta z}{\delta \mathbf{b}} = 2\mathbf{b}^\top$$

$$\frac{\delta \mathbf{b}}{\delta \mathbf{a}} = \mathbf{I}$$

$$\frac{\delta \mathbf{a}}{\delta \mathbf{w}} = \mathbf{X}$$

$$\frac{\delta L(\mathbf{w})}{\delta \mathbf{w}} = \frac{\delta z}{\delta \mathbf{b}} \times \frac{\delta \mathbf{b}}{\delta \mathbf{a}} \times \frac{\delta \mathbf{a}}{\delta \mathbf{w}}$$

$$\frac{\delta L(\mathbf{w})}{\delta \mathbf{w}} = 2\mathbf{b}^\top \cdot \mathbf{I} \cdot \mathbf{X}$$

$$\frac{\delta L(\mathbf{w})}{\delta \mathbf{w}} = 2(\mathbf{X}\mathbf{w} - \mathbf{y})^\top \cdot \mathbf{I} \cdot \mathbf{X}$$

Closed Form Solution

Set $\dfrac{\delta L(\mathbf{w})}{\delta \mathbf{w}} = 0$ and find $\mathbf{w}^\star$

Closed Form Solution

Set $\dfrac{\delta L(\mathbf{w})}{\delta \mathbf{w}} = 0$ and find $\mathbf{w}^\star$

$$2(\mathbf{X}\mathbf{w} - \mathbf{y})^\top \mathbf{X} = 0$$

Expand and simplify

$$\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} - \mathbf{y}^\top \mathbf{X} = 0$$

$$\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} = \mathbf{y}^\top \mathbf{X}$$

Take transpose on both sides

$$\mathbf{X}^\top \mathbf{X}\mathbf{w} = \mathbf{X}^\top \mathbf{y}$$

Multiply both sides with the inverse of $\mathbf{X}^\top X$

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

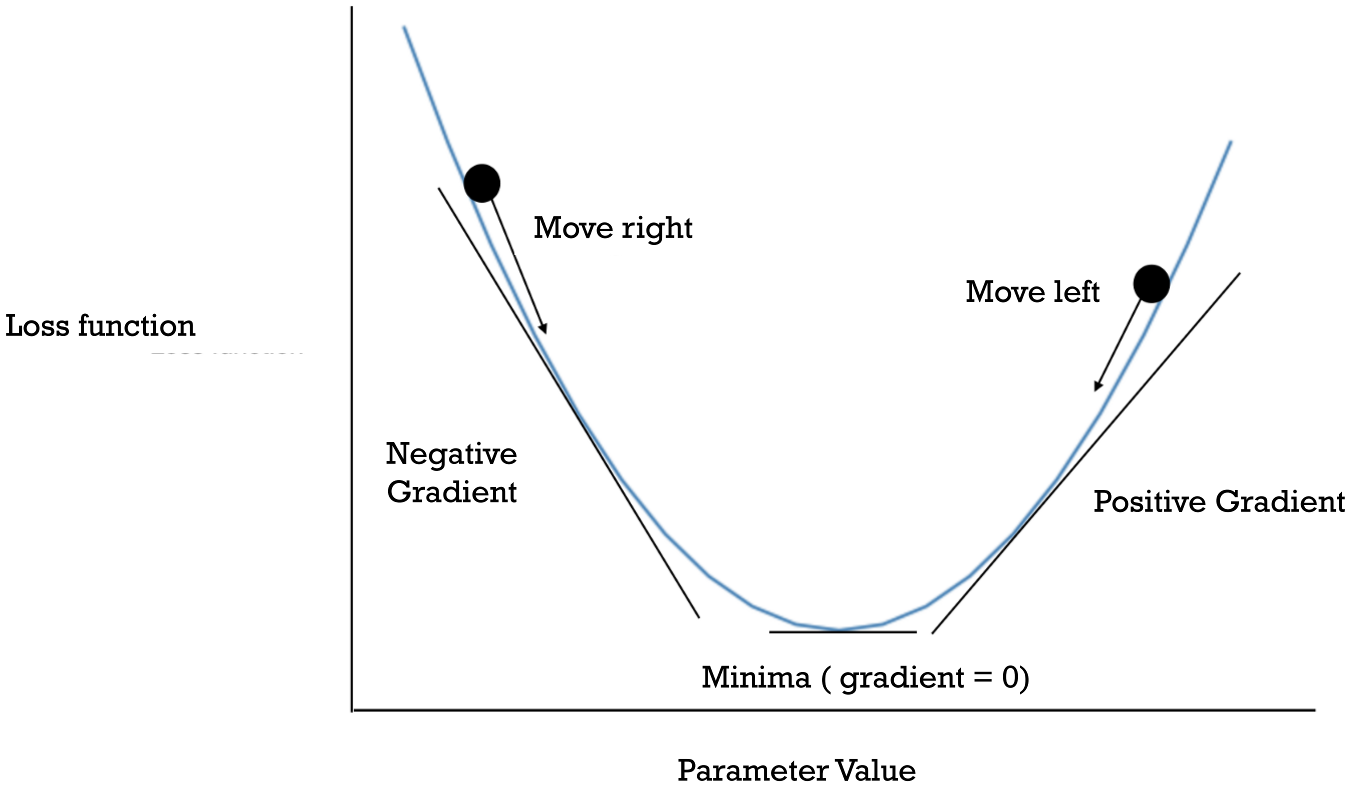
Gradient Descent

- Choose starting point $\mathbf{w}_0$
- Update weights $t = 1, 2, 3, \cdots$

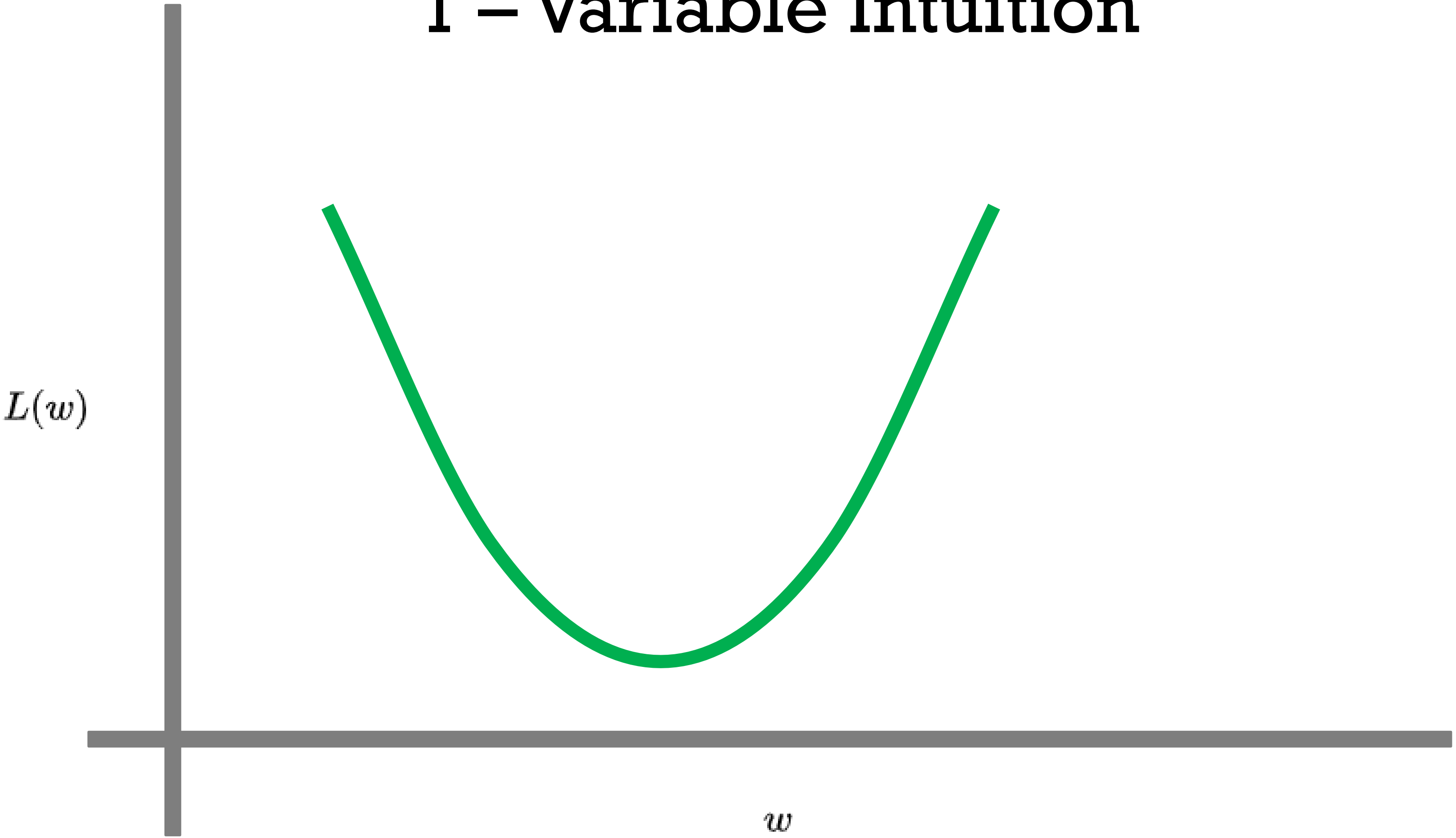
$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \frac{\delta L(\mathbf{w})}{\delta \mathbf{w}_{t-1}}$$

where α is a hyperparameter called learning rate that specifies step length.

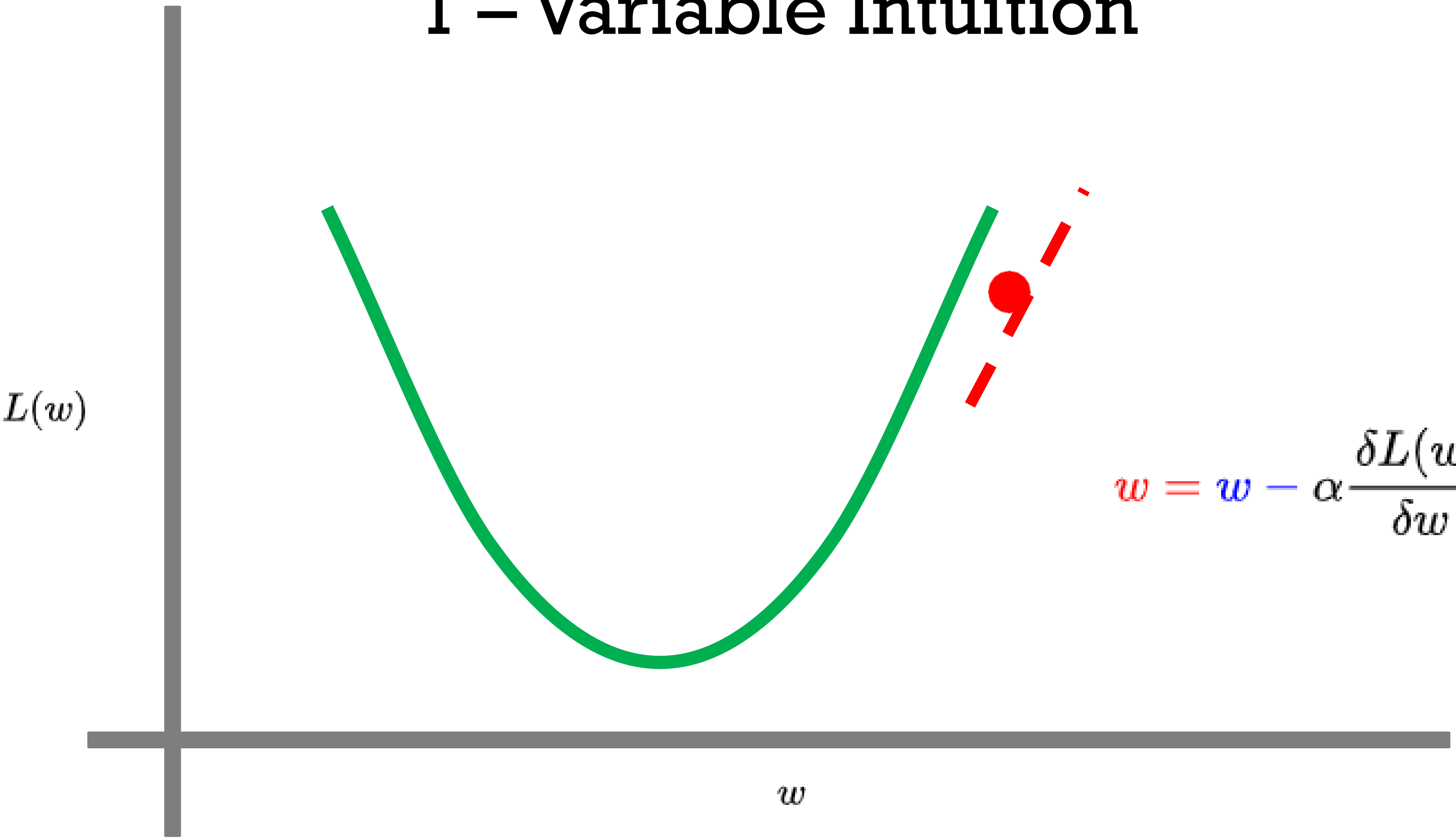
Visualization of Gradient Descent



1 – Variable Intuition

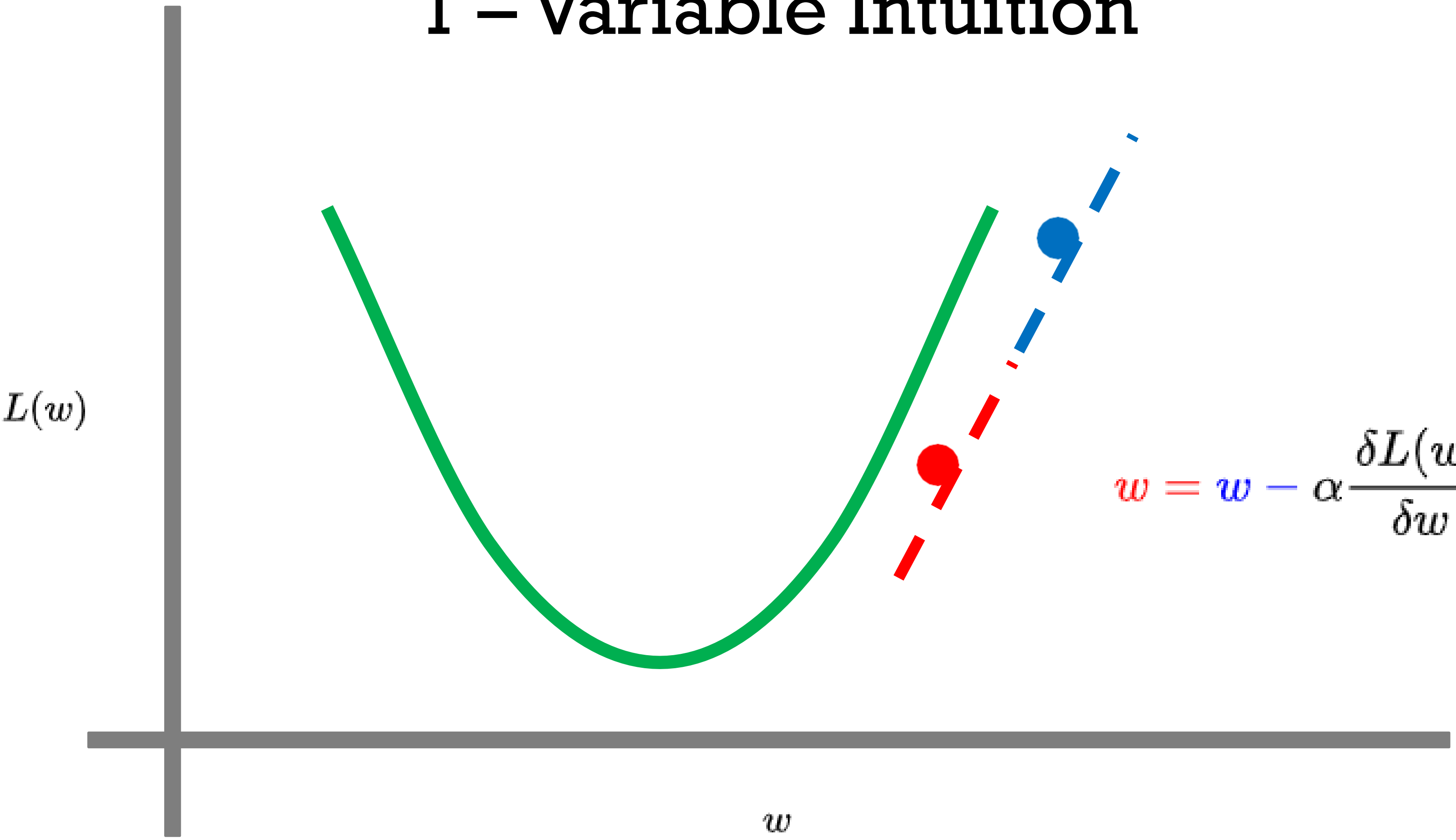


1 – Variable Intuition



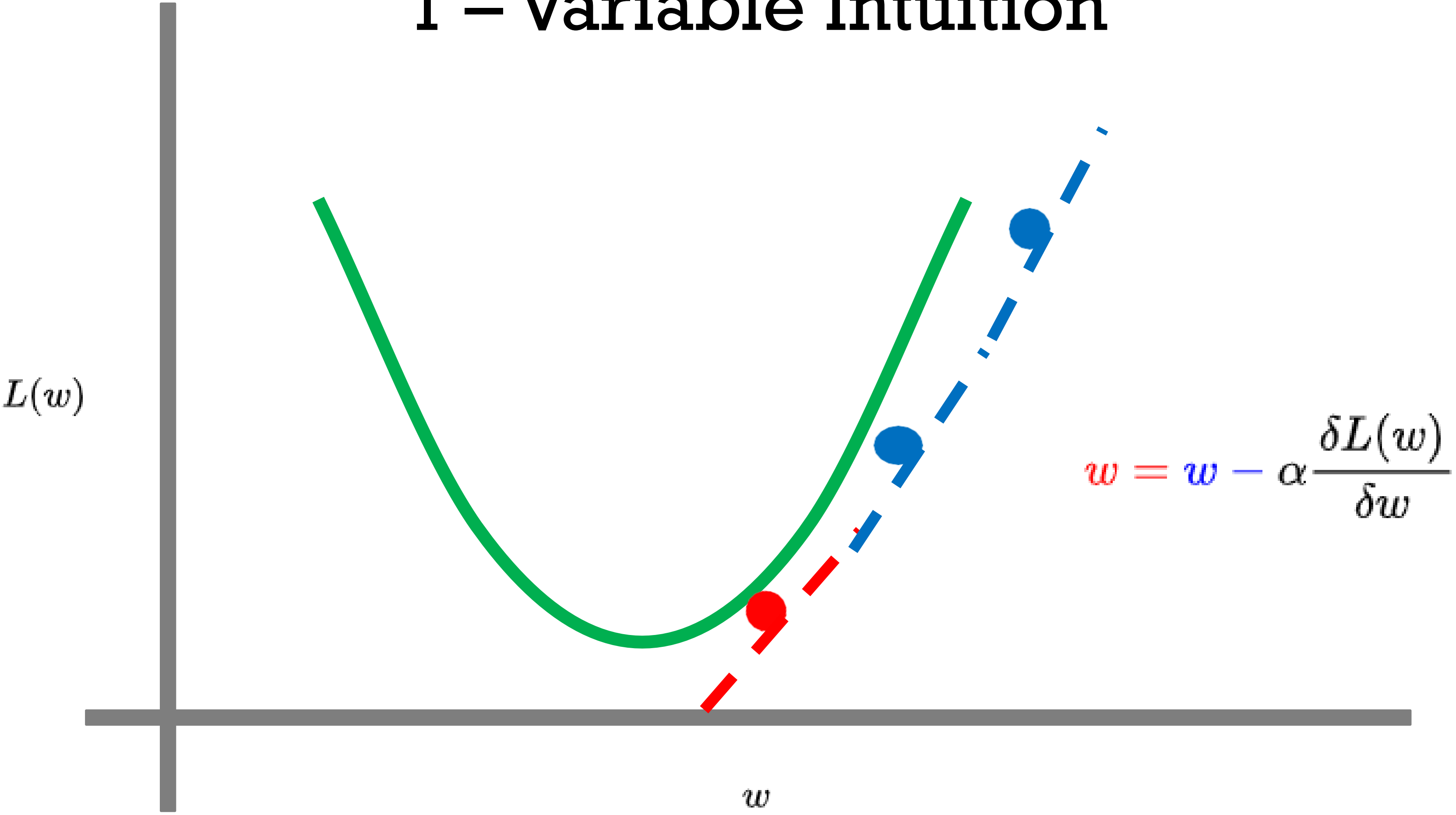
$$w = w - \alpha \frac{\delta L(w)}{\delta w}$$

1 – Variable Intuition

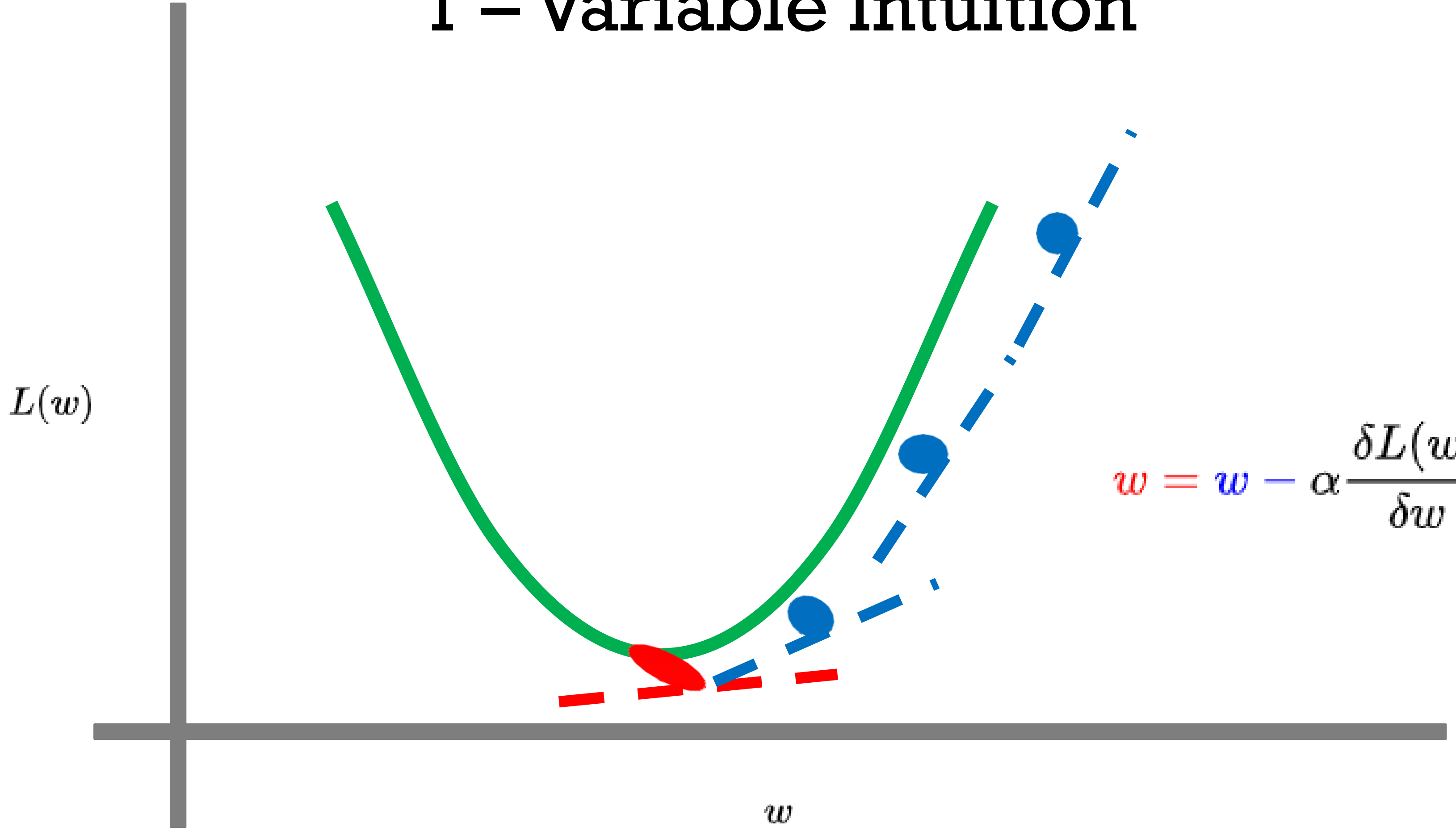


$$w = w - \alpha \frac{\delta L(w)}{\delta w}$$

1 – Variable Intuition

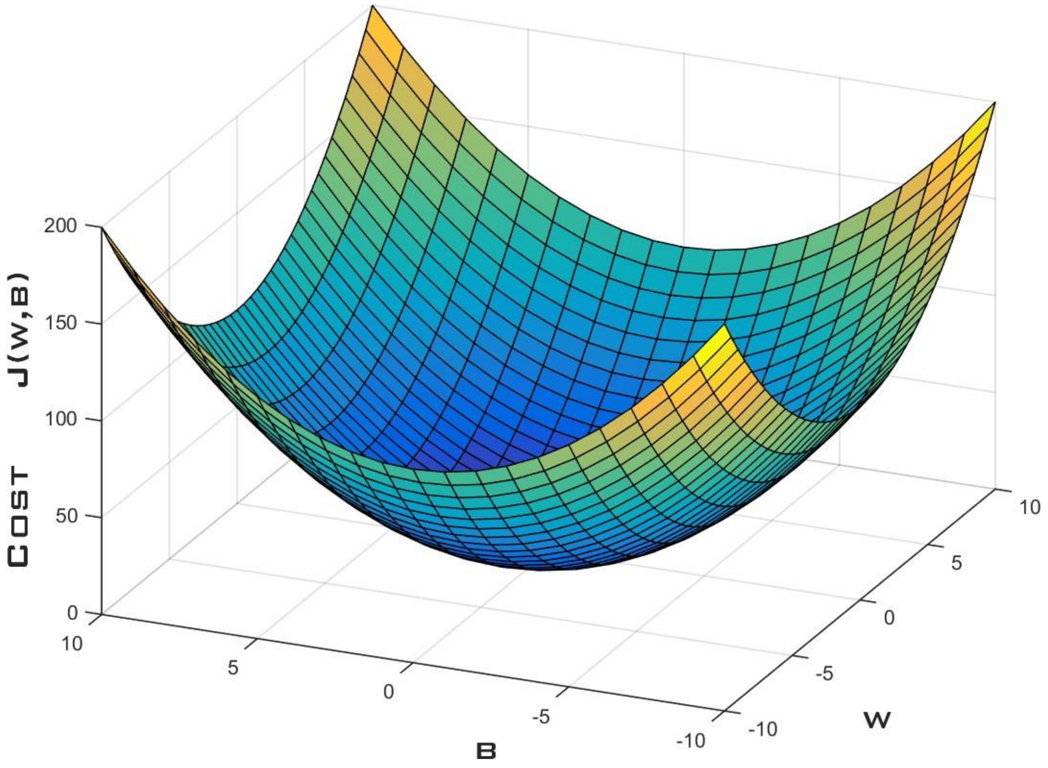


1 – Variable Intuition



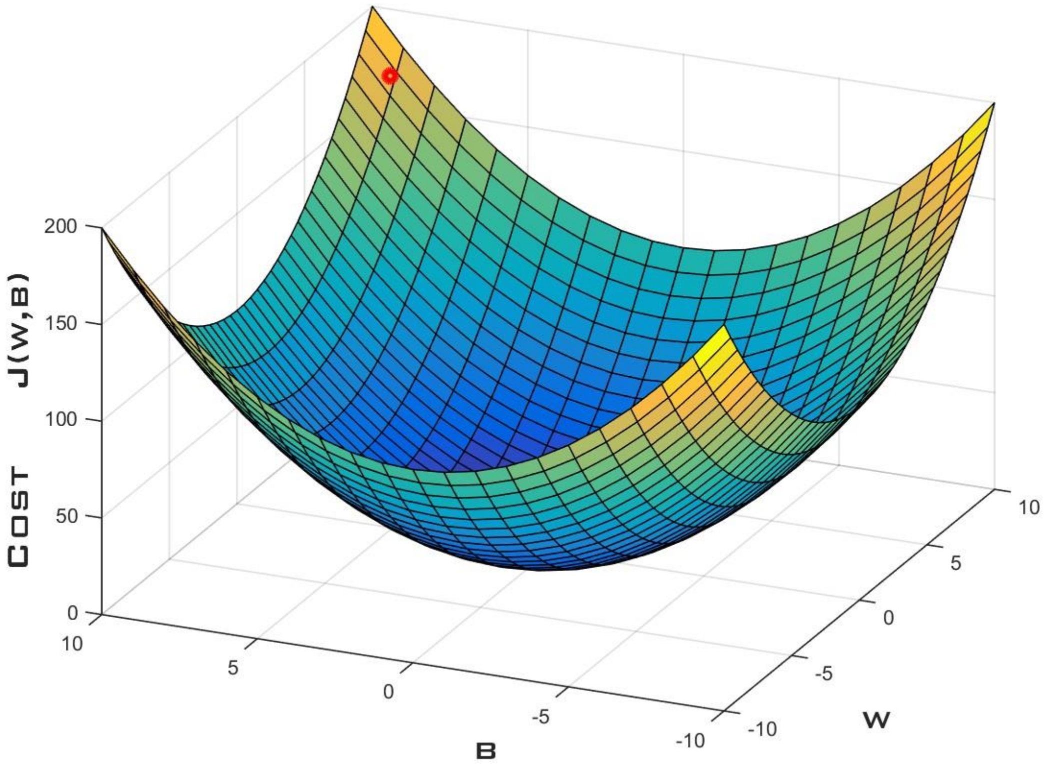
$$w = w - \alpha \frac{\delta L(w)}{\delta w}$$

2 – Variable Intuition



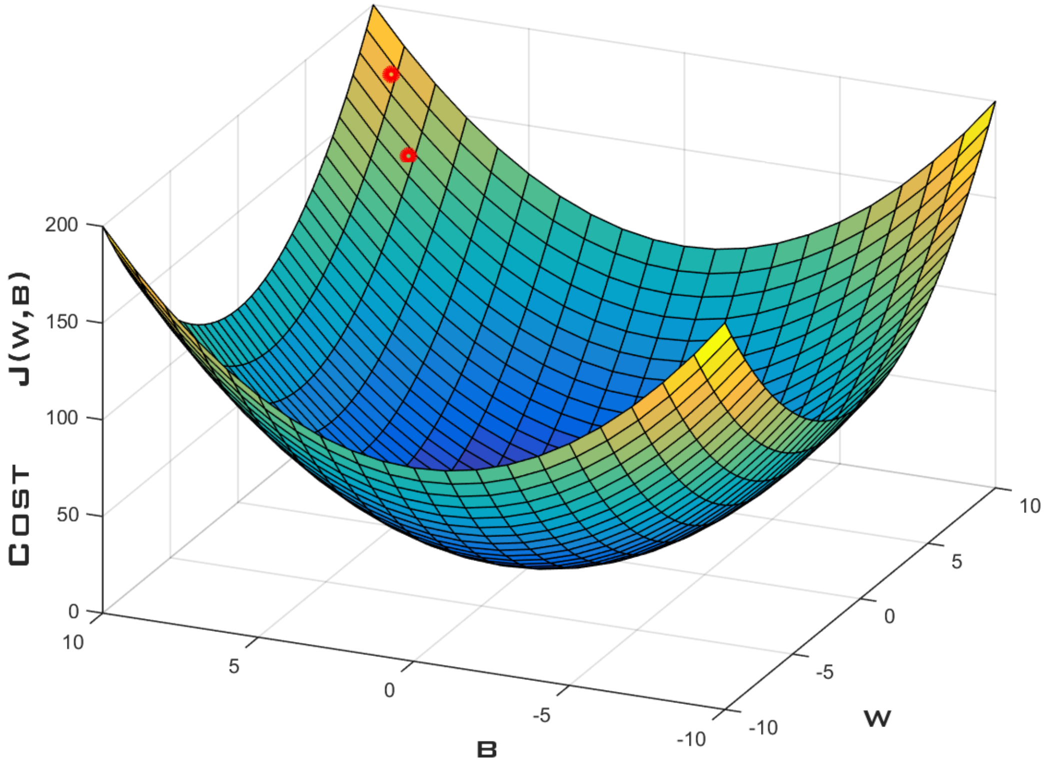
In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

2 – Variable Intuition



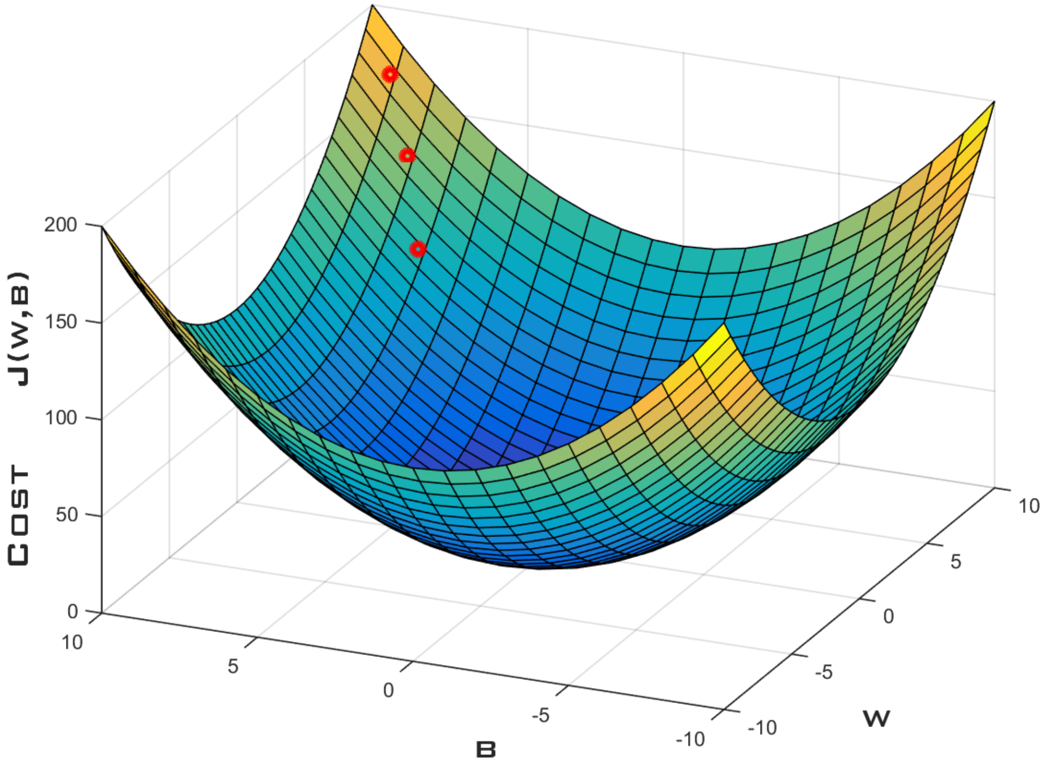
In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

2 – Variable Intuition



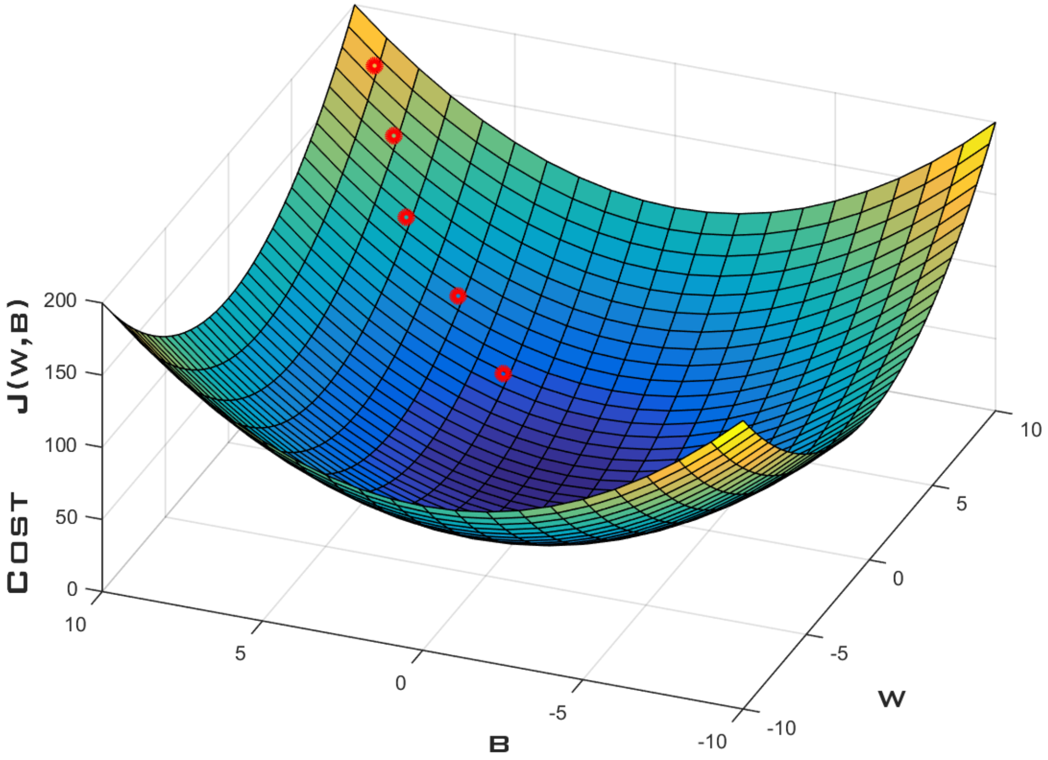
In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

2 – Variable Intuition



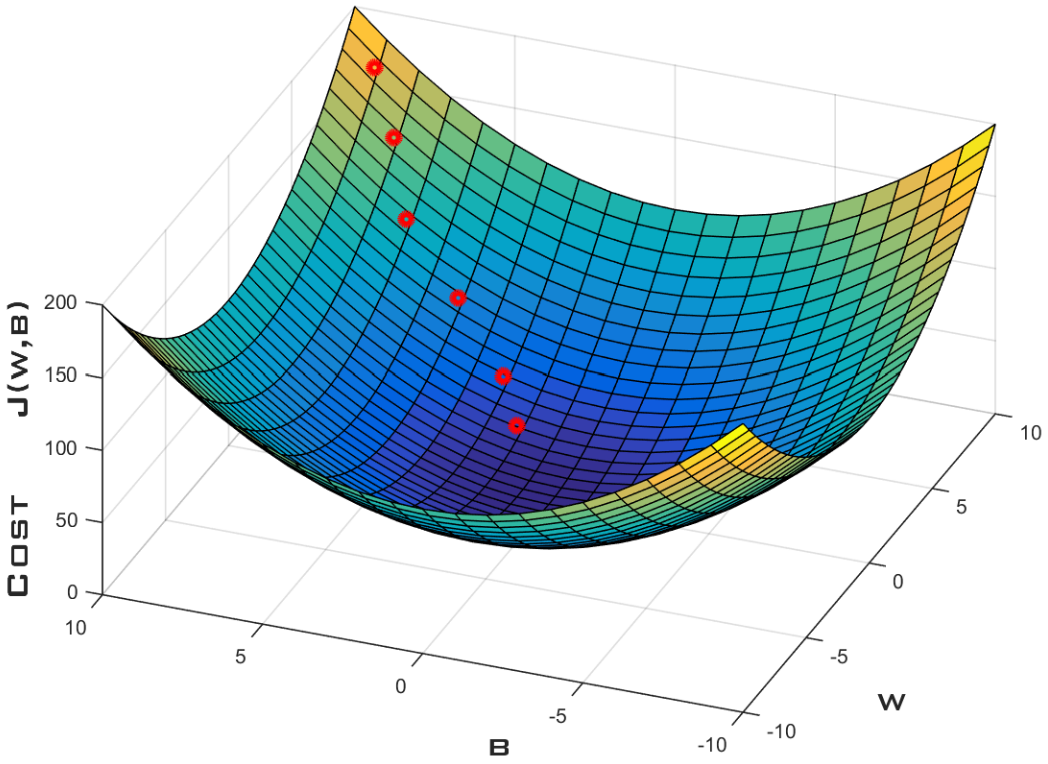
In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

2 – Variable Intuition



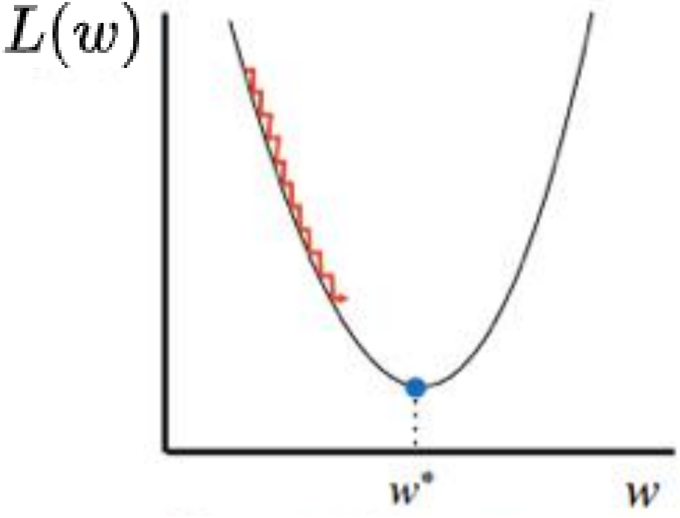
In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

2 – Variable Intuition

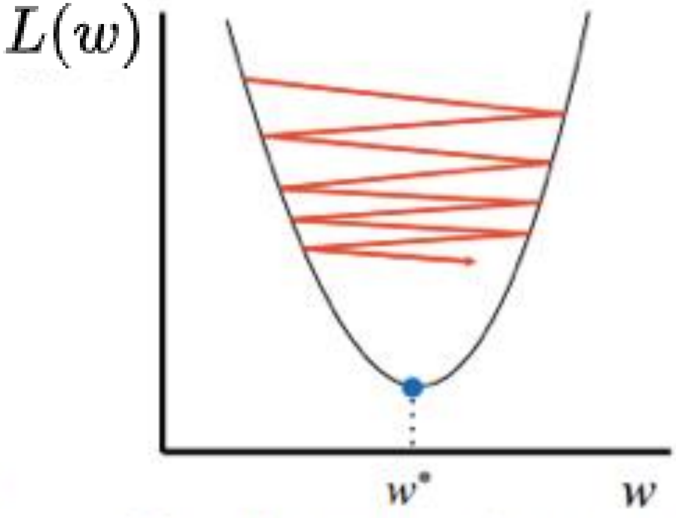


In an alternate notation $J(w, b)$ is equal to $L(\mathbf{w})$

Learning Rate



Too small: converge very slowly



Too big: overshoot and even diverge

Mini-Batch Stochastic Gradient Descent

- If the number of samples is large, a single iteration over the entire data set is not a viable technique.
- We can choose a fixed number of instances from set at random, call it a batch, and perform one step of gradient descent, known as mini-batch stochastic gradient descent.
- An extreme case would be to choose one sample at random and then perform a gradient descent step.

Mini-Batch Stochastic Gradient Descent

- Randomly initialize the w model parameters.
- Iteratively sample random mini-batch β from the data.
- Update the parameters in the direction of the negative gradient.

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}_t} \partial_{\mathbf{w}} l^{(i)}(\mathbf{w}, b) &&= \mathbf{w} - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}_t} \mathbf{x}^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right) \\
b &\leftarrow b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}_t} \partial_b l^{(i)}(\mathbf{w}, b) &&= b - \frac{\eta}{|\mathcal{B}|} \sum_{i \in \mathcal{B}_t} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right).
\end{aligned}$$

Choosing Batch Size

- **Not too small**

Workload is low, making it hard to fully utilize computational power.

- **Not too big**

Memory problem. Wasted computation, i.e. if all x_i are identical.