ENGR 3321: Introduction to Deep Learning for Robotics

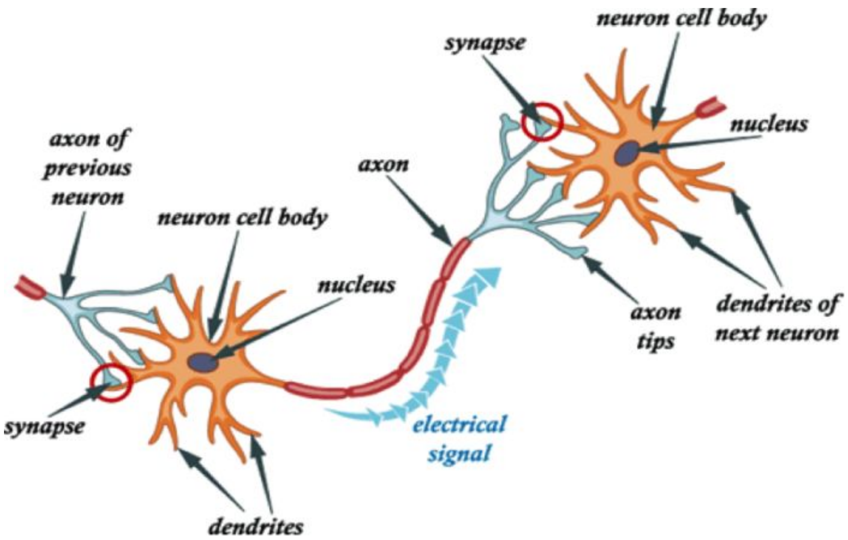
Neural Network 101: SISO Linear Function

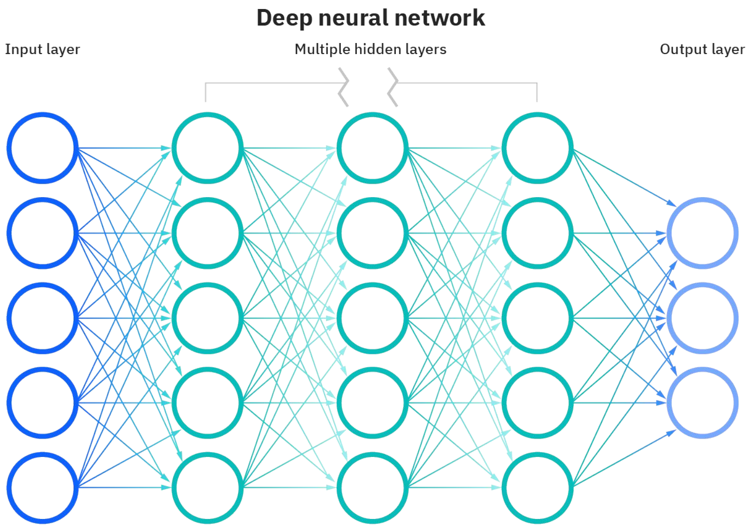
09/03/2025

Outline

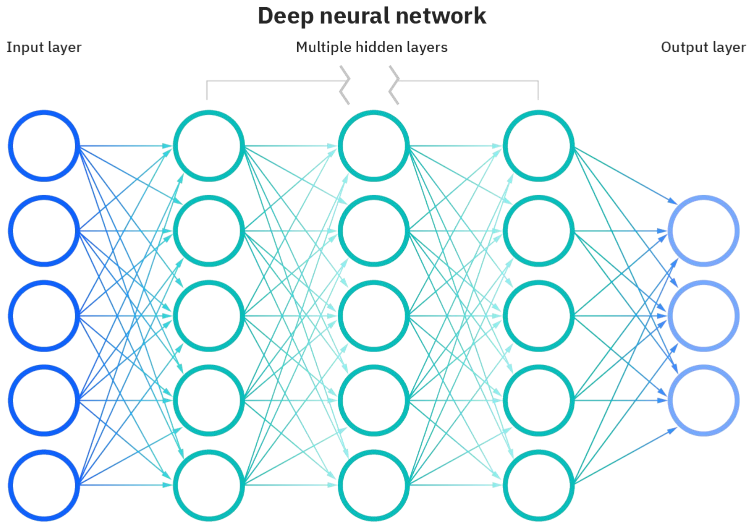
- Simplest neural network model: NN101
- Model learning/training

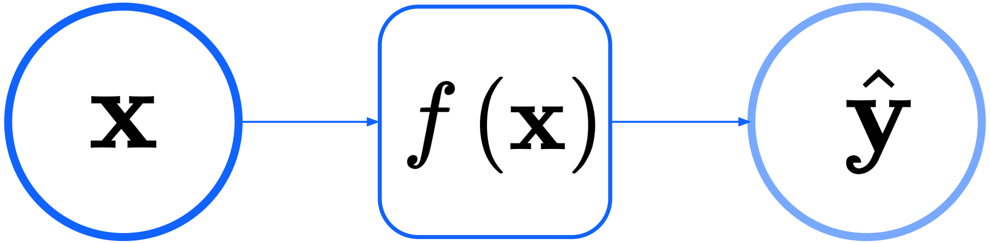
Neural Network



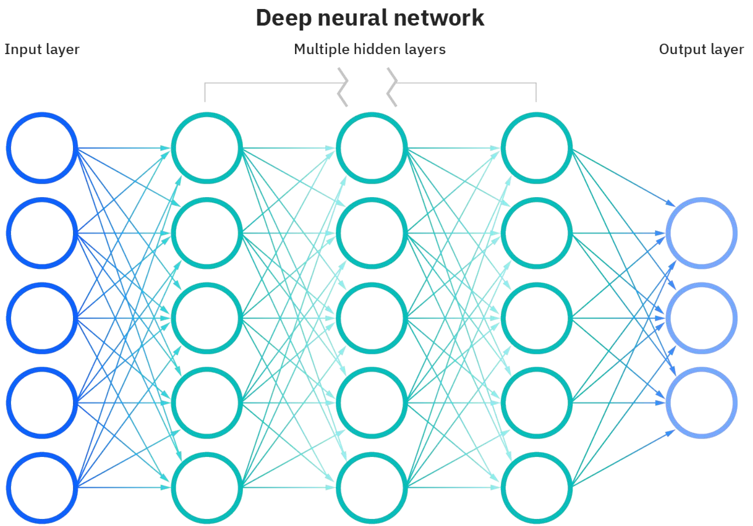


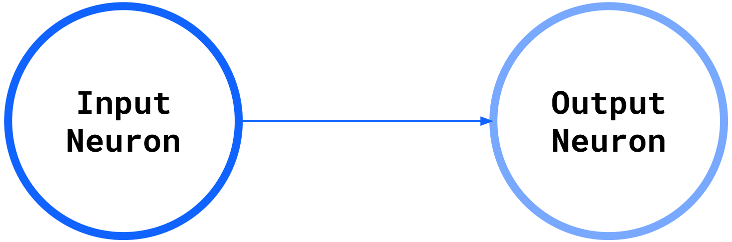
Neural Network == Function



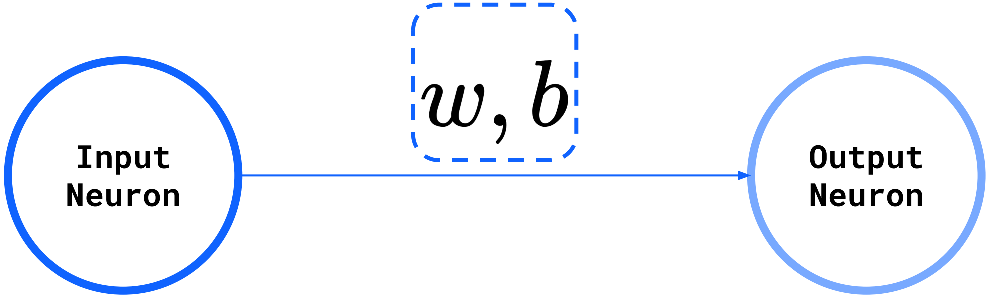


SISO Neural Network (NN101)



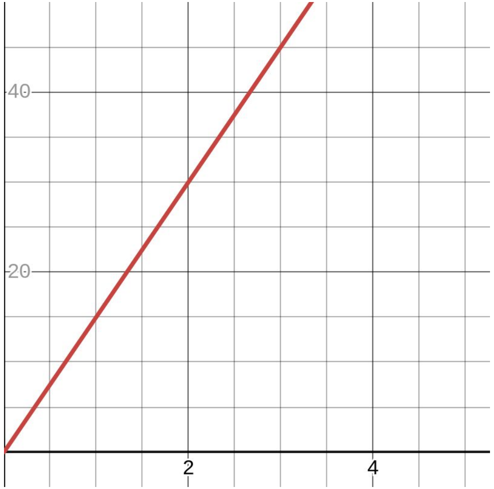


NN101 == Linear Model



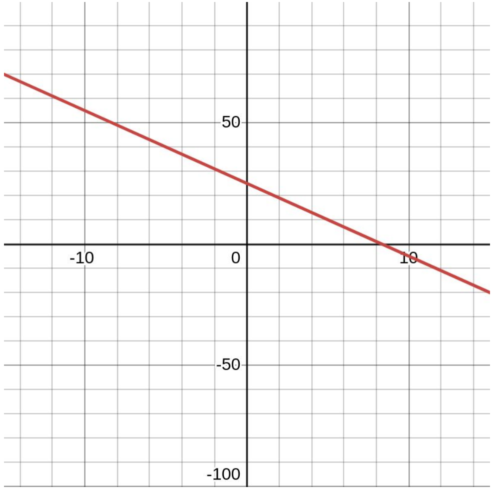
$$\hat{y} = f(x) = wx + b$$

Linear Model Examples



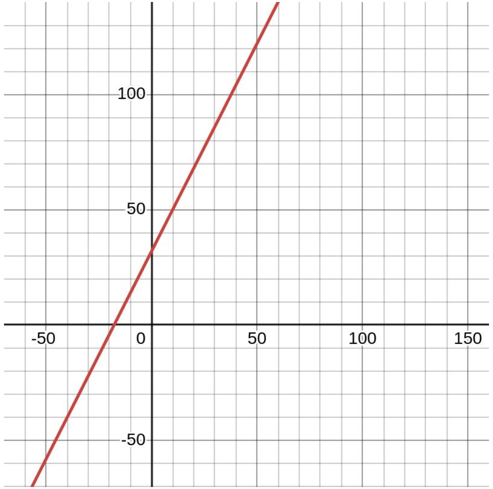
$$y = 15x$$

Hourly wage



$$y = -3x + 25$$

Car decelerate



$$y = \frac{9}{5}x + 32$$

Temperature conversion

Learning Objective

$$\hat{y} = f(x) = wx + b$$

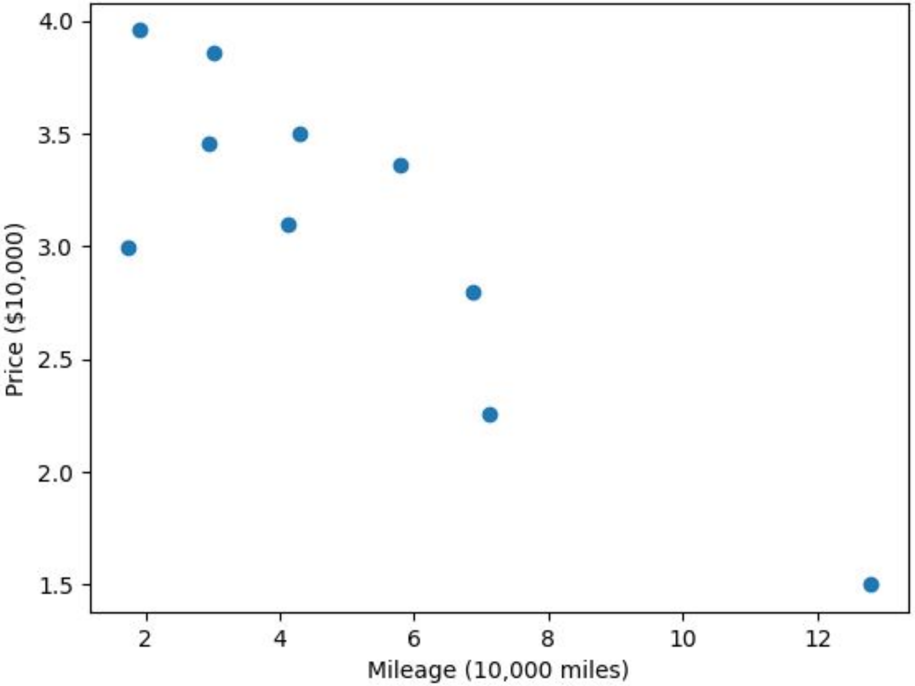
Parameters Learning Procedure

1. Guess a model (w and b).
2. Evaluate the model .
3. Find out hints to improve the model.
4. Perform improvements.
5. Repeat 2 to 4 until converge.

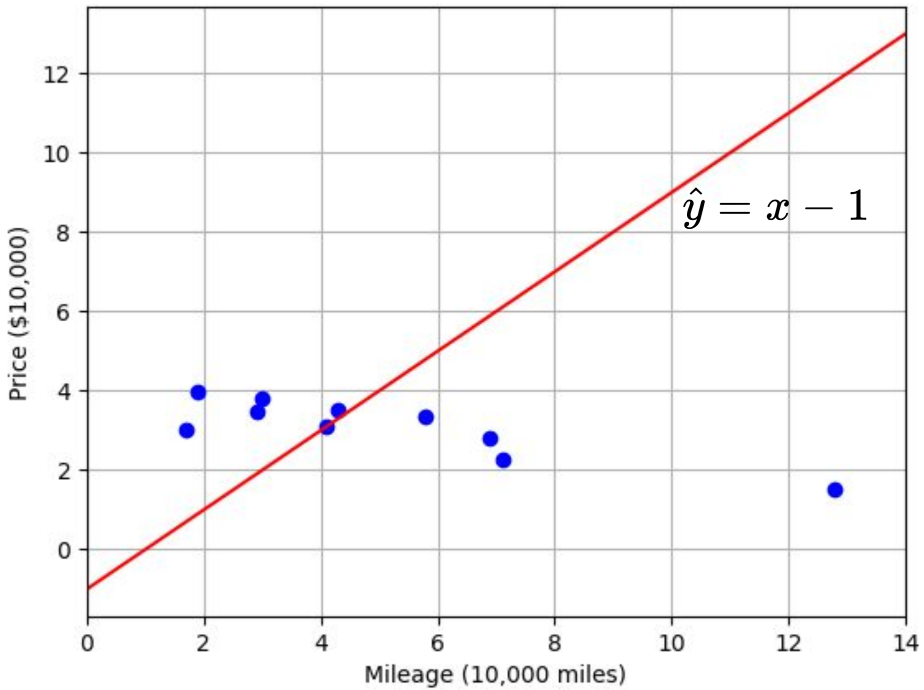
Example: Predict Used Car Price

Mileage (10,000 miles)	Selling Price ($10,000)
5.7923	3.359
7.1229	2.259
1.9160	3.959
4.1124	3.099
12.8000	1.5
6.8696	2.799
2.9499	3.459
4.3000	3.5
1.7302	2.999
3.0237	3.859

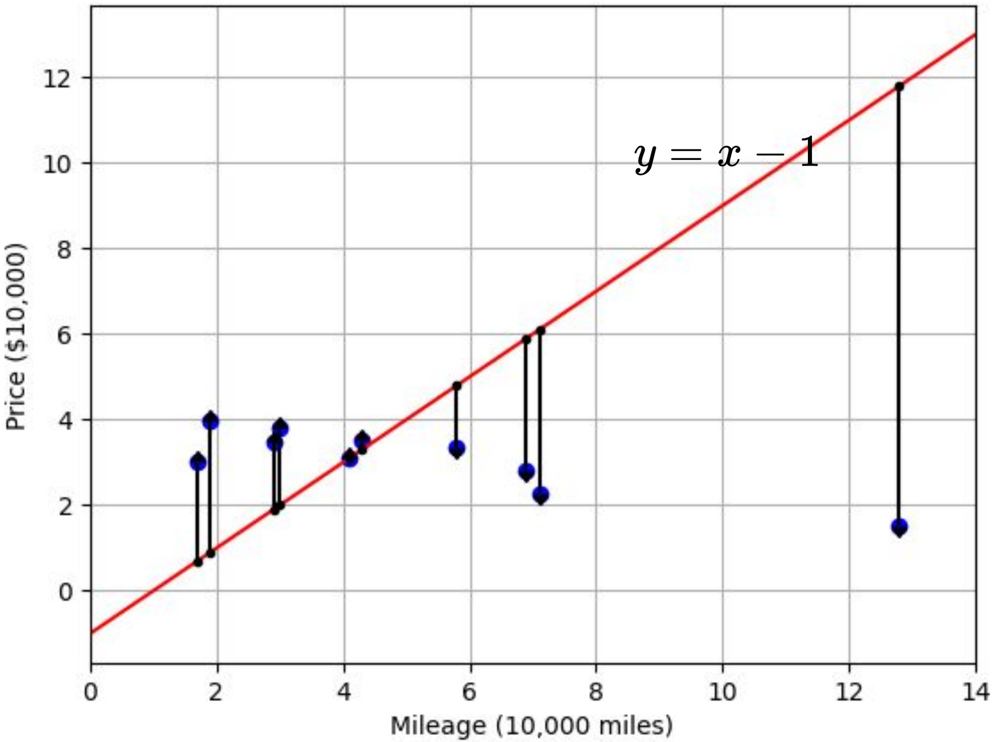
Dataset: $\mathcal{D} = \{({}^{(1)}x, {}^{(1)}y), ({}^{(2)}x, {}^{(2)}y), \ldots, ({}^{(M)}x, {}^{(M)}y)\}$



Initial Guess



Evaluate Model Performance



Mean Squared Error (MSE) function

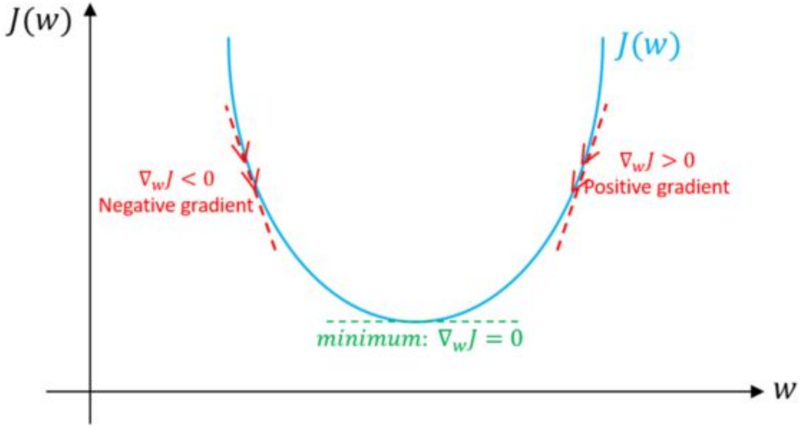
$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2} ({}^{(i)}\hat{y} -{}^{(i)} y)^2$$

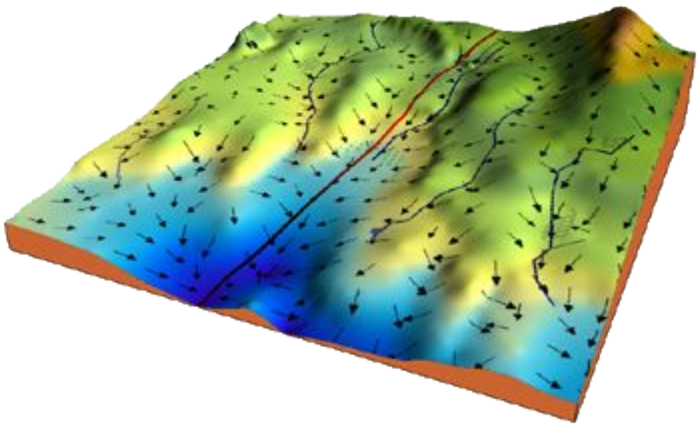
Gradient/Derivatives

$$\nabla \mathcal{L}(w,b) = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w} \\ \frac{\partial \mathcal{L}}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial w} \\ \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{1}{M} \sum_{i=1}^{M} (^{(i)}\hat{y} -^{(i)} y)^{(i)} x \\ \frac{1}{M} \sum_{i=1}^{M} (^{(i)}\hat{y} -^{(i)} y) \end{bmatrix}$$

Chain Rule

Gradient Descent Concept





Find w and b that minimize $\mathcal{L}(w, b)$

Gradient Descent Algorithm

Initialize $w \, and \, b$

Repeat until converge {

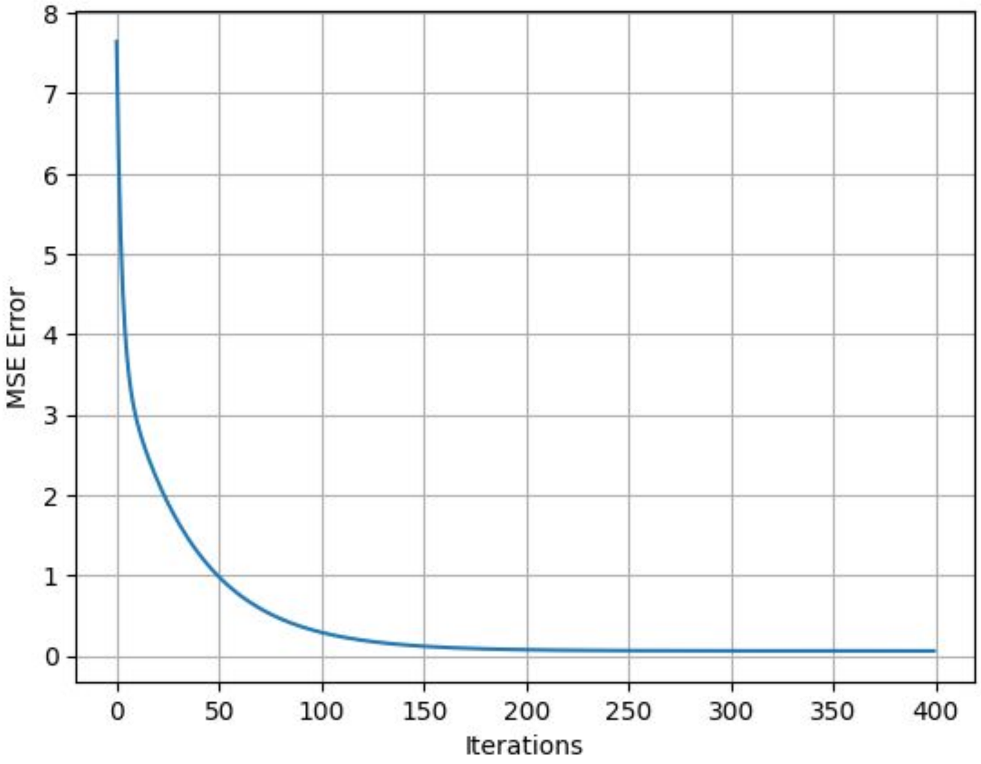
$$w := w - \alpha \frac{\partial \mathcal{L}}{\partial w}$$

$$b := b - \alpha \frac{\partial \mathcal{L}}{\partial w}$$

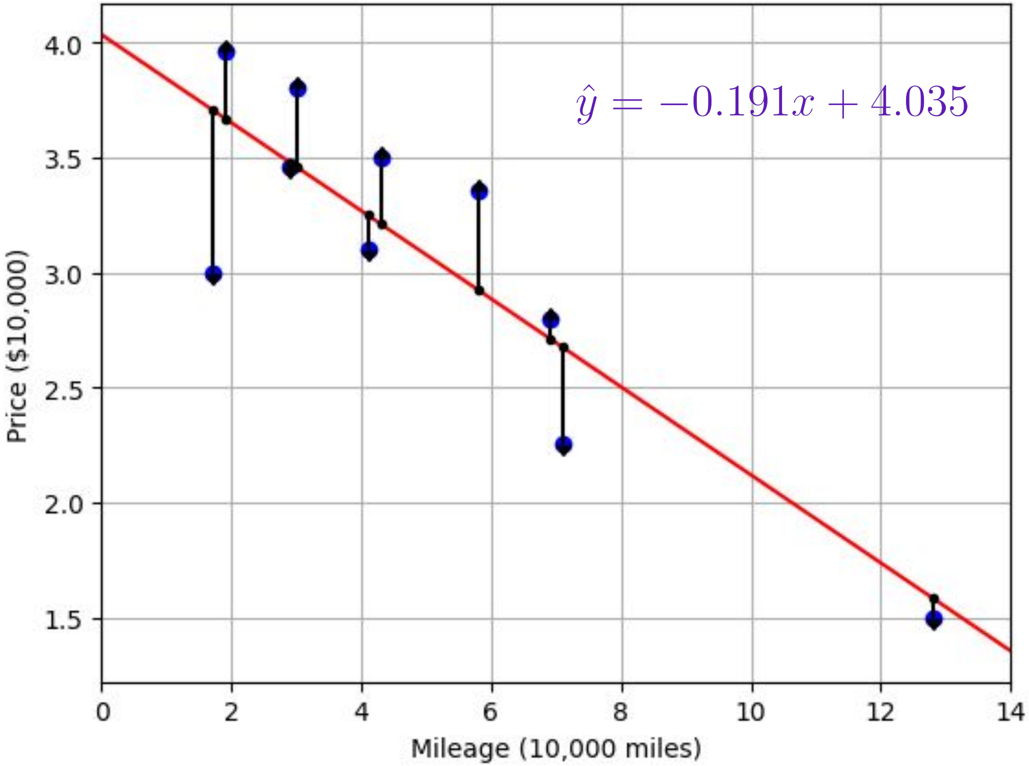
}

where α is learning rate

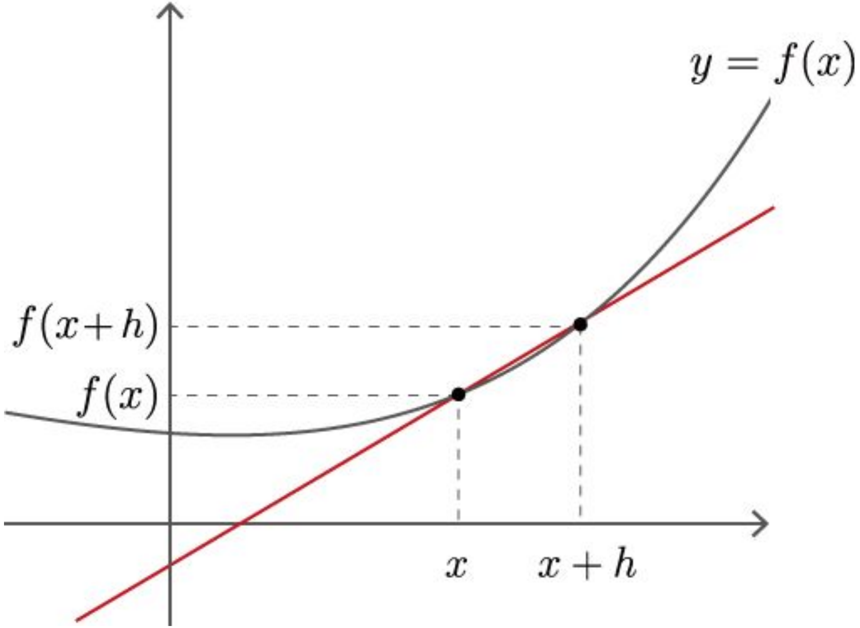
Loss Decrease



Trained Model



Derivative Review



the derivative of a function at a given point gives us the rate of change or slope of the tangent line to the function at that point.