ENGR 4350: Applied Deep Learning

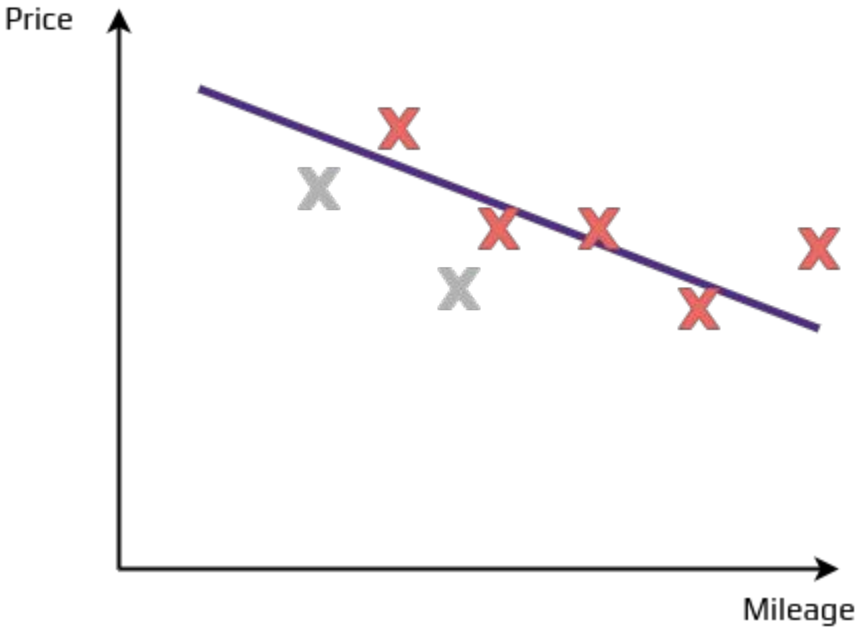
Logistic Regression: Part 1

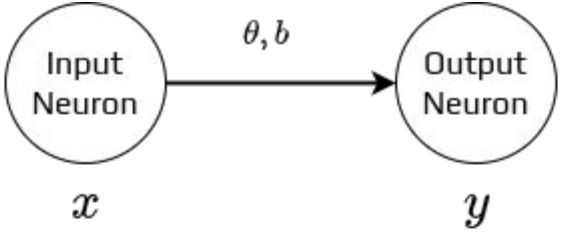
08/31/2022

Outline

- An example of neural network
- Logistic Regression
  - Forward Pass
  - Loss Function
  - Gradient Descent

A Neural Network Example

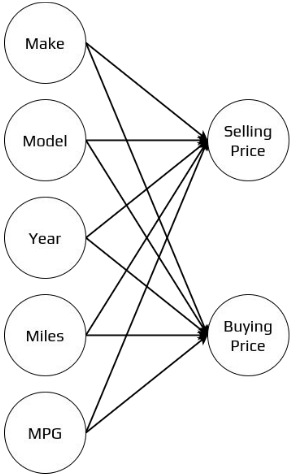




$$y = \sigma(\theta x + b)$$

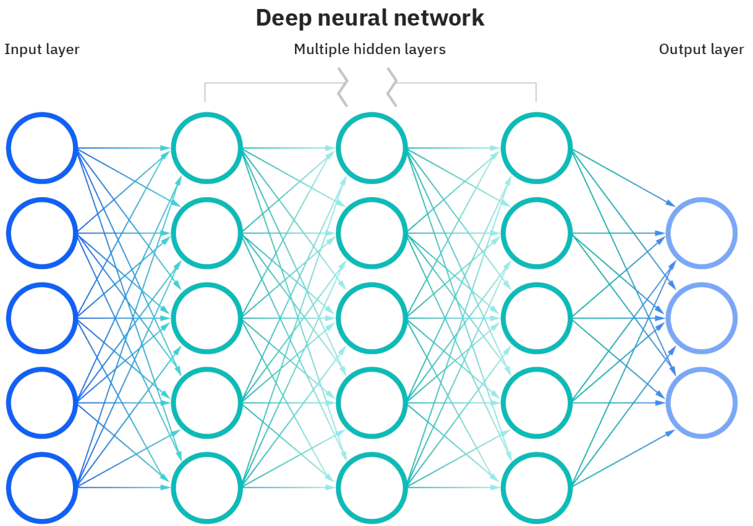
A Neural Network Example

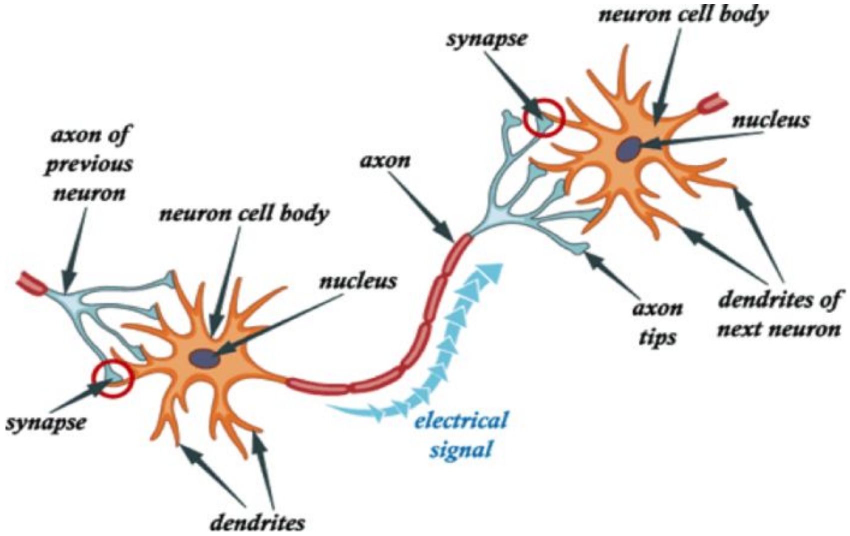
Make	Model	Year	Mileage	MPG	**Buying Price**	**Selling Price**
Ford	Edge	2018	50,000	23	**$19,000**	**$10,000**
Toyota	Land Cruiser	2020	10,000	15	**$80,000**	**$50,000**
VW	Golf	2010	150,000	36	**$7,000**	**$2,000**



$$\vec{y} = \sigma(\vec{\theta}\vec{x} + \vec{b})$$

A Neural Network Example





Binary Classification

- Complex decision makings can be simplified to classification problems.
  - Vehicle control
  - Robotic arm control
  - Gaming
  - ...
- Binary classification works most of the time.
  - Visitor identification
  - Animal protection
  - Farming
  - ...

Logistic Regression

Logistic regression estimates the probability of an event occurring, $P(\mathbf{y} = \mathbf{1} \mid \mathbf{x})$ such as voted or didn’t vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

- Classification
- Prediction
- Rating

…

E.g. Given the “make, model, year, mileage, MPG” of vehicles, estimate probabilities of prices of these vehicles under $20,000.

Problem Settings

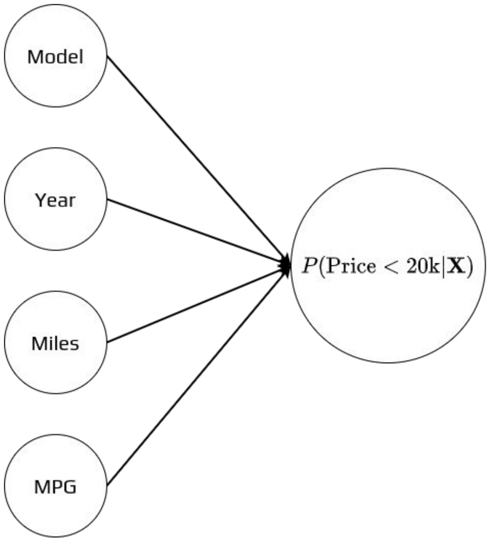
Dataset: $\left\{\left(\mathbf{x}^{(1)}, y^{(1)}\right), \left(\mathbf{x}^{(2)}, y^{(2)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}$

$$\text{Features: } \mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & . & . & . & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_1^{(m)} & x_2^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix} \qquad \text{Labels: } \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ . \\ . \\ . \\ y^{(m)} \end{bmatrix}$$

m examples, n independent variables

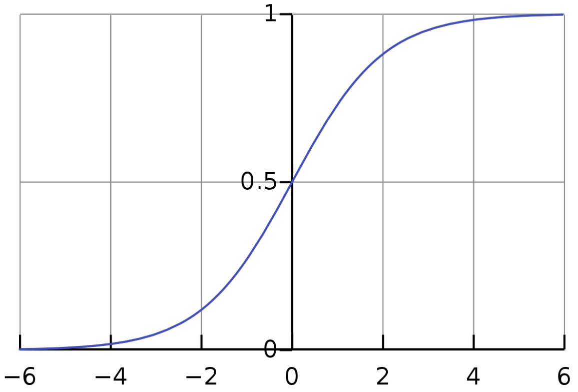
Forward Pass / Prediction

Id (Model)	Year	Mileage	MPG	**Buying Price**
5 (Ford Edge)	2018	50,000	23	**1 ($19,000)**
105 (Toyota Landcruiser)	2020	10,000	15	**0 ($80,000)**
233 (VW Golf)	2010	150,000	36	**1 ($7,000)**



Forward Pass / Prediction

Sigmoid function: $\sigma(z) = \dfrac{1}{1+e^{-z}}$



Input: $\mathbf{X}$

Weights: $\mathbf{w} \in \mathbb{R}^{n_\mathbf{x}}$, bias: $b \in \mathbb{R}$

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & . & . & . & w_n \end{bmatrix}$$

Output: $\hat{\mathbf{y}} = \sigma\left(\mathbf{X}\mathbf{w}^\mathbf{T} + b\right)$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & . & . & . & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_1^{(m)} & x_2^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ . \\ w_n \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ . \\ . \\ . \\ \hat{y}^{(m)} \end{bmatrix}$$

Loss Function

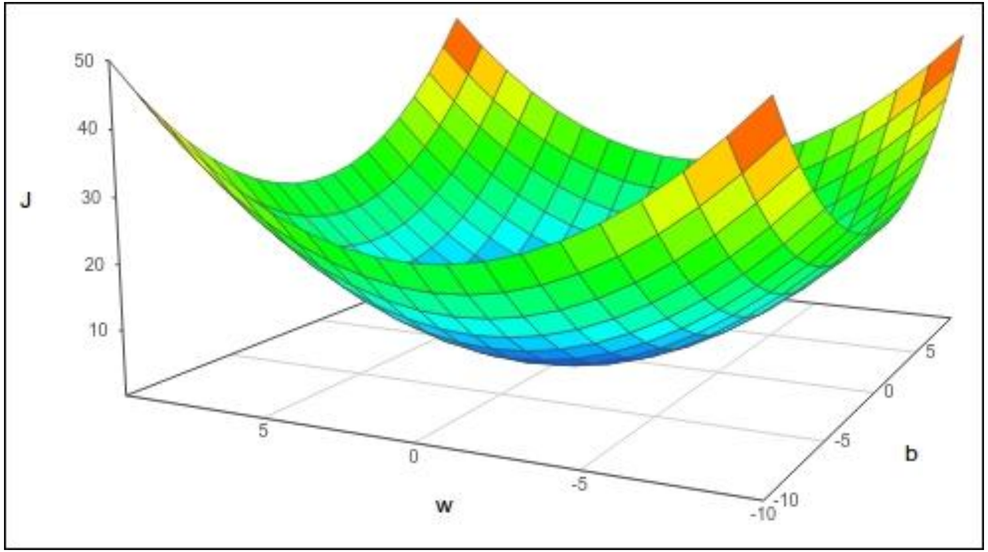
Dataset: $\left\{\left(\mathbf{x}^{(1)}, y^{(1)}\right), \left(\mathbf{x}^{(2)}, y^{(2)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}$

Mean squared error (MSE) loss function: $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \dfrac{1}{2}(\hat{\mathbf{y}} - \mathbf{y})^2$

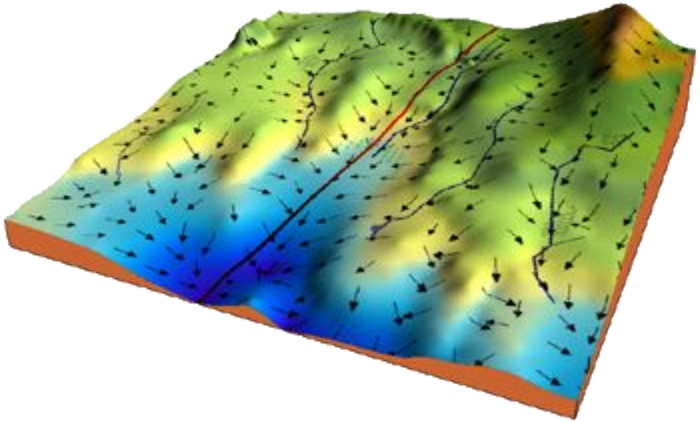
Cross entropy loss function: $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -(\mathbf{y}\log \hat{\mathbf{y}} + (1 - \mathbf{y})\log\ (1 - \hat{\mathbf{y}}))^2$

Cost function: $J(\mathbf{w}, b) = \dfrac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$

Gradient Descent



Find $\mathbf{w}$ and b that minimize $J(\mathbf{w}, b)$



Gradient Descent

repeat until converge {

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial J}{\partial \mathbf{w}}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$

}

α is the learning rate