ITCS 6156/8156 Fall 2023
Machine Learning

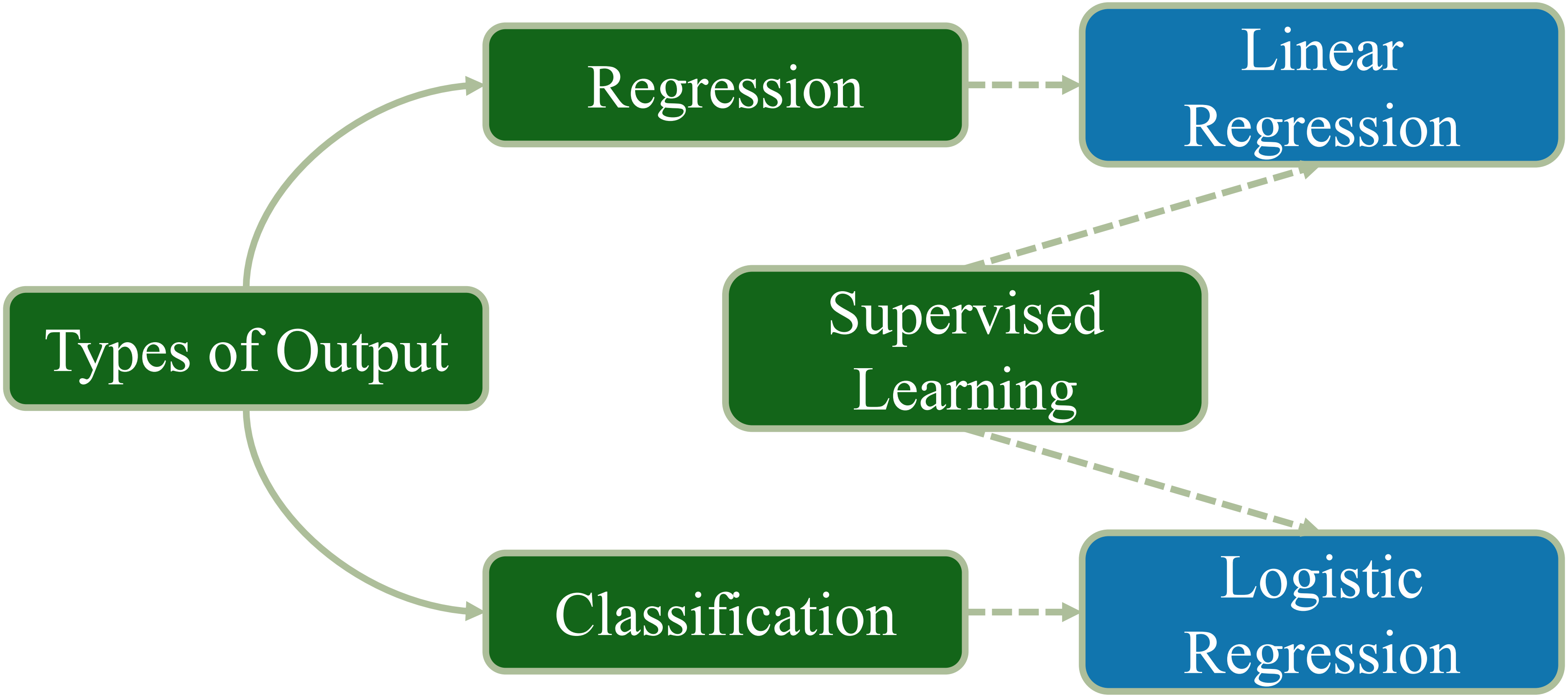
Logistic Regression

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Some content in the slides is based on Dr. Razvan's lecture

Roadmap

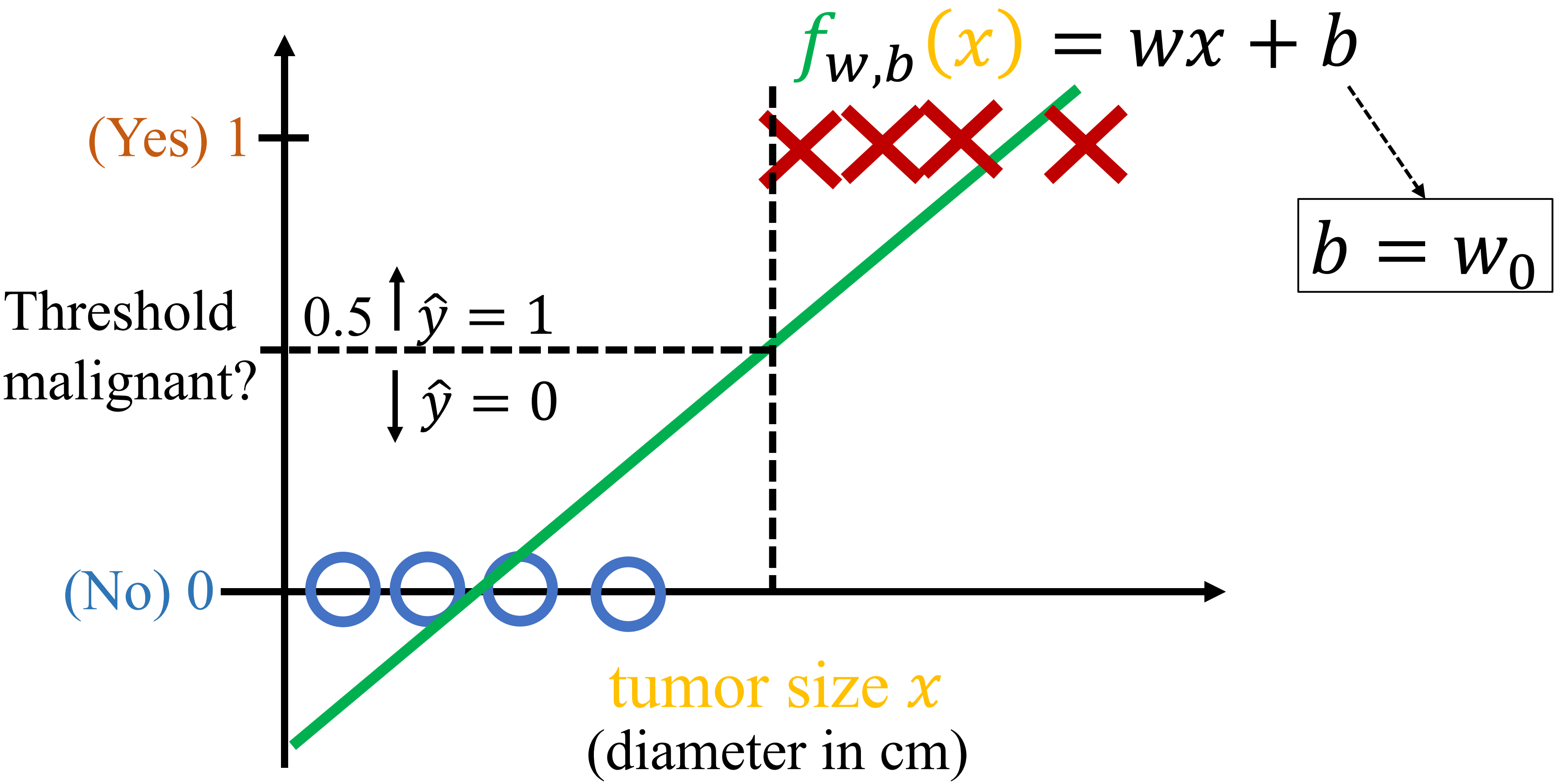

Regression
Linear Regression
Types of Output
Supervised Learning
Classification
Logistic Regression

Classification

Question	Answer "y"
Is this email spam?	no yes
Is the transaction fraudulent?	no yes
Is the tumor malignant?	no yes

- **binary classification:**
  - "y" can only be one of two values:
    - false: 0: "negative class" = "absence"
    - true: 1: "positive class" = "presence"

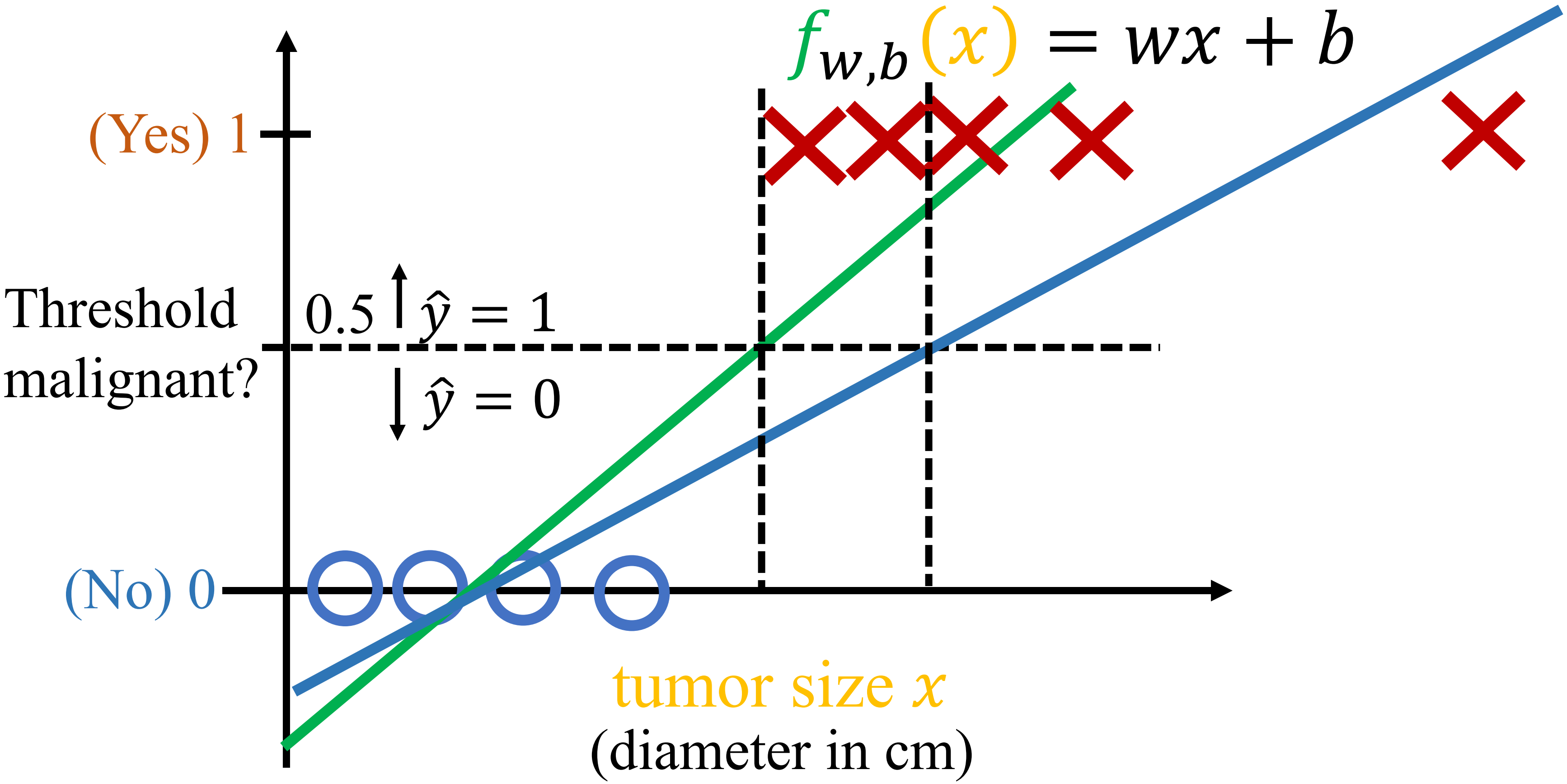
Linear Regression Approach



if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$

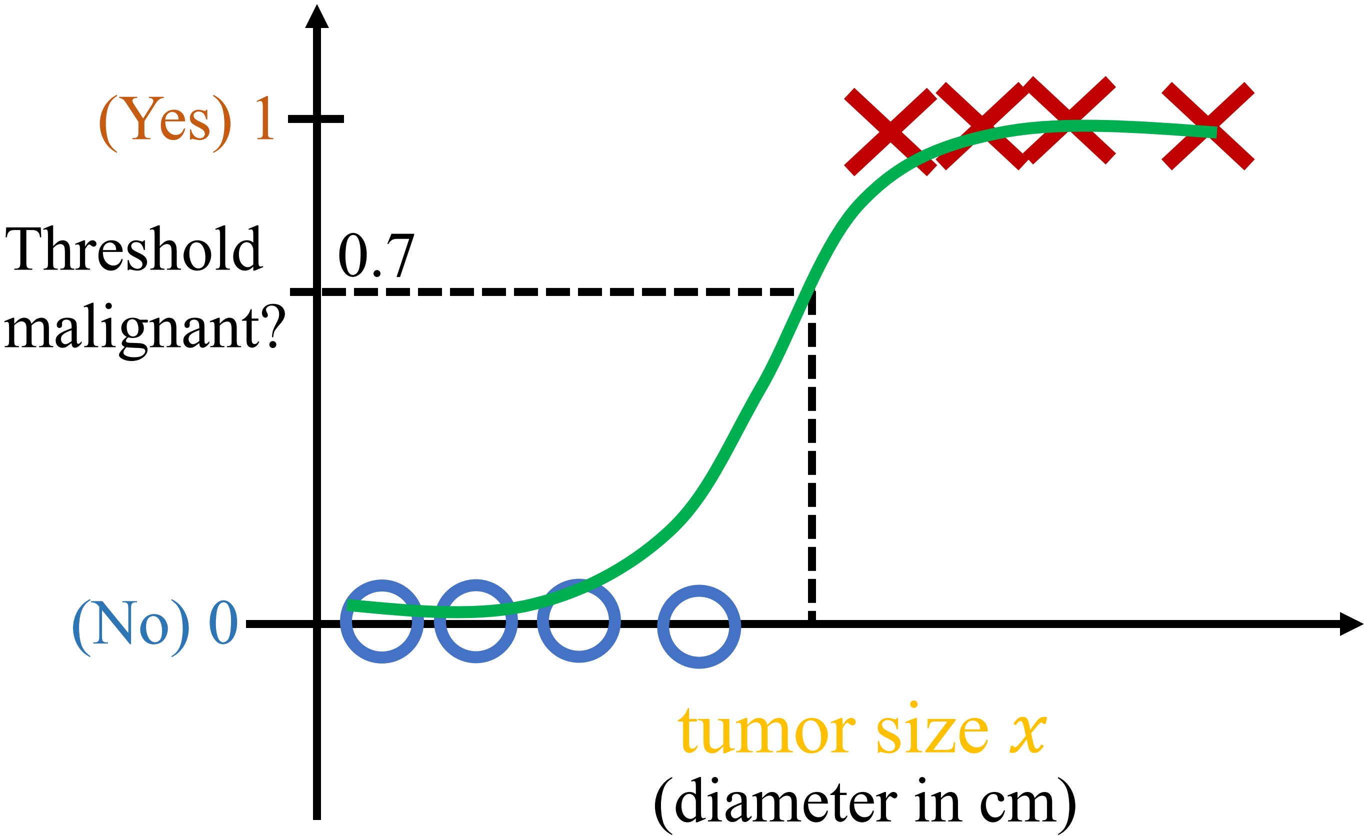
if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

Linear Regression Approach



$$\text{if } f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$$
$$\text{if } f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$$

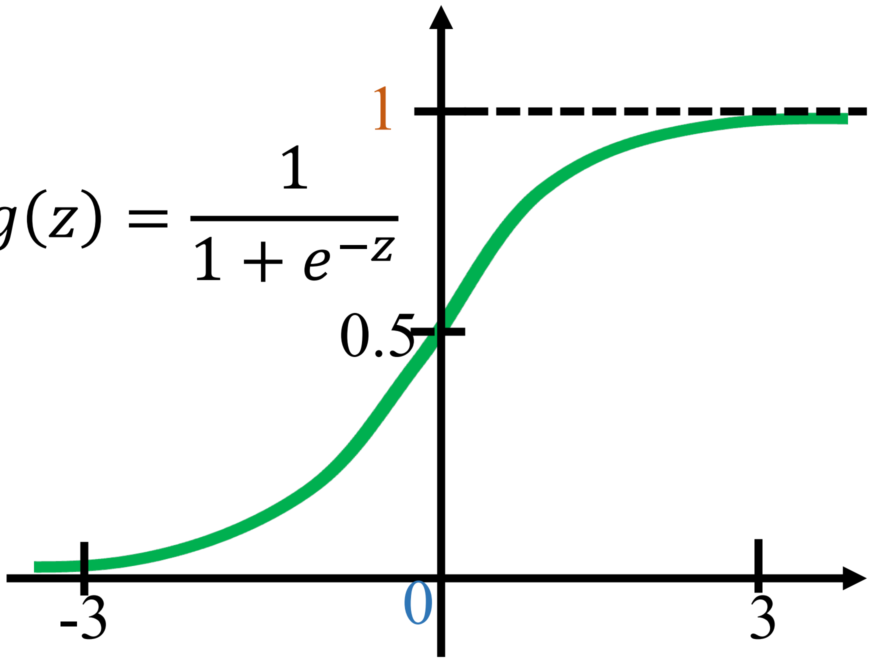
Logistic Function



Probabilistic Discriminative Models: directly model the posterior class probabilities $p(C|\mathbf{x}; \boldsymbol{w}, b)$

Logistic Function

- Want outputs between 0 and 1



$$g(z) = \frac{1}{1+e^{-z}}$$

$$w \cdot x + b$$

$$\downarrow$$

$$z$$

$$\downarrow$$

$$g(z) = \frac{1}{1+e^{-z}}$$

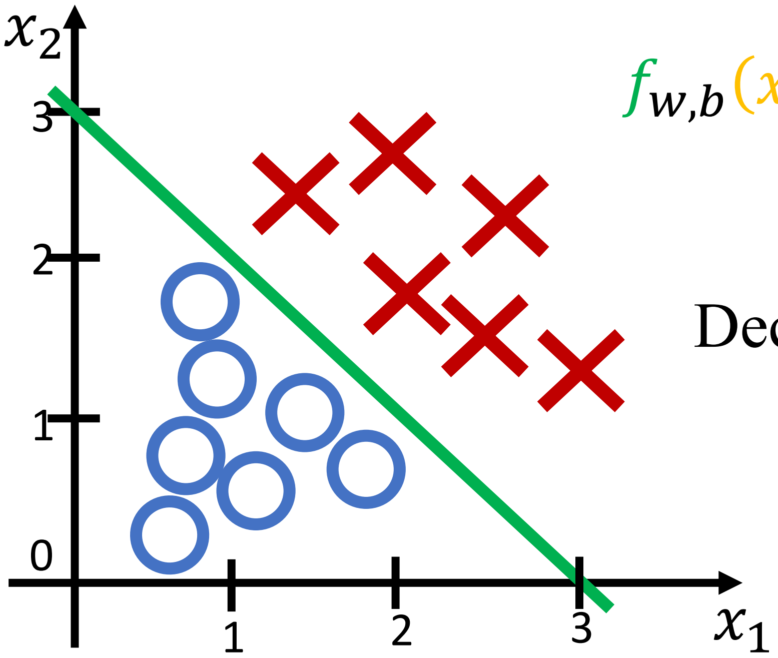
$$f_{w,b}(x) = g(w \cdot x + b)$$

$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$

logistic regression

- sigmoid function
- logistic function
- outputs between 0 and 1 $g(z) = \frac{1}{1+e^{-z}}$, $0 < g(z) < 1$

Decision Boundary



$$f_{w,b}(x) = g(z) = g(w_1x_1 + w_2x_2 + b)$$

Decision Boundary: $z = w \cdot x + b = 0$

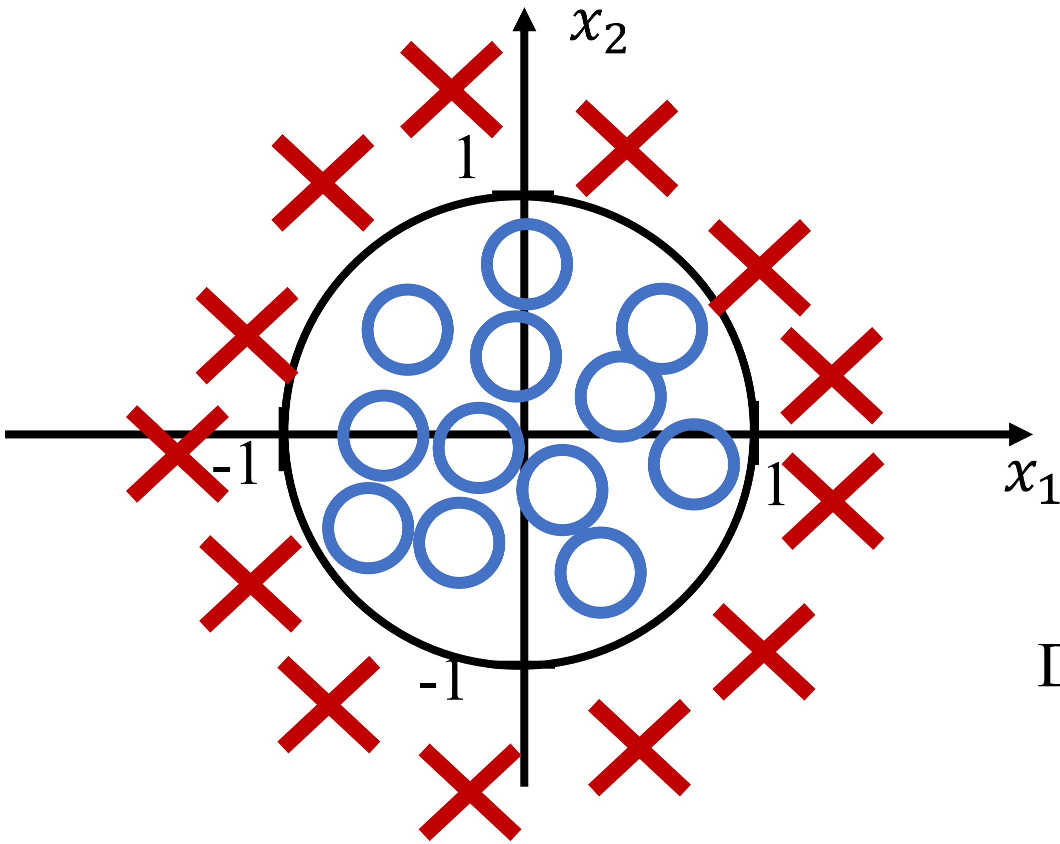
(set $w_1 = 1, w_2 = 1$)

$$z = x_1 + x_2 - 3 = 0$$

$$x_1 + x_2 = 3$$

Decision boundary is hyperplane $f(x) = 0.5 \ \rightarrow z = 0$

Non-linear Decision Boundary



$$z = w_1 x_1^2 + w_2 x_2^2 + b$$

Decision Boundary:

(set $w_1 = 1, w_2 = 1$)

$z = {x_1}^2 + {x_2}^2 - 1 = 0$

${x_1}^2 + {x_2}^2 = 1$

Loss Function

Training Set

tumor size(cm)	…	patient's age	malignant?
x_1		x_n	y
10		52	1
2		73	0
5		55	0
12		49	1
…		…	…

$i = 1,2,\cdots m$: number of training samples

$j = 1,2,\cdots n$: number of features

target y is 0 or 1

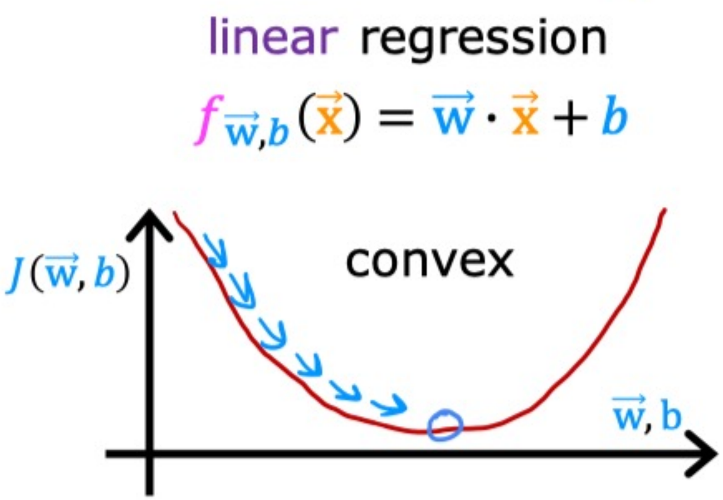
$$f_{w,b}(x) = \frac{1}{1 + e^{-(w\cdot x+b)}}$$

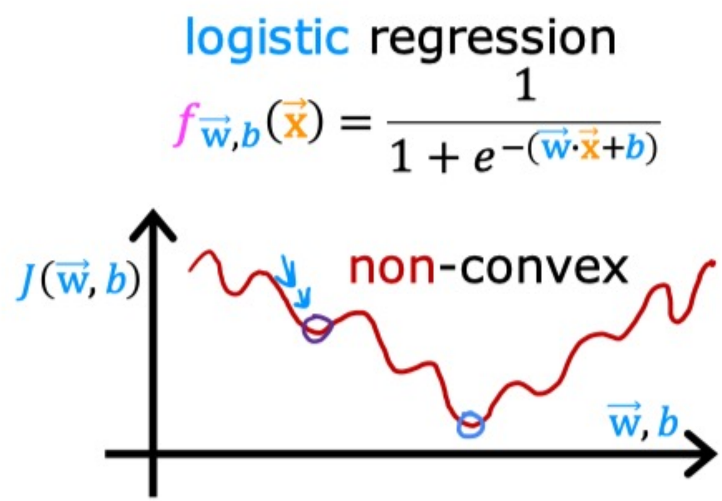
How to choose $w = [w_1, w_2, w_3, \cdots w_n]$ and b?

Loss Function

- Squared Error Cost:

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2$$





- Differentiable => can use gradient descent ✓
- Non-convex => not guaranteed to find the global optimum ✗

Loss Function

- Logistic Loss Function:

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)), & if\ y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & if\ y^{(i)} = 0 \end{cases}$$

$if\ y^{(i)} = 1,$ As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$, then loss $\to 0$
As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$, then loss $\to \infty$

$if\ y^{(i)} = 0,$ As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$, then loss $\to \infty$
As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$, then loss $\to 0$

Simplified Loss Function

- Logistic Loss Function:

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right), & if\ y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & if\ y^{(i)} = 0 \end{cases}$$

- Simplified Logistic Loss Function (Convex):

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

- Overall:

$$\begin{aligned} J(\vec{w}, b) &= \frac{1}{m}\sum_{i=1}^{m}\left[L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)\right] \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right] \end{aligned}$$

Can be derived from Maximum Likelihood

Gradient Descent

- Overall Loss (Cost):

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right]$$

- Gradient Decent:

Repeat {

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}\left(J(\vec{w}, b)\right),$$

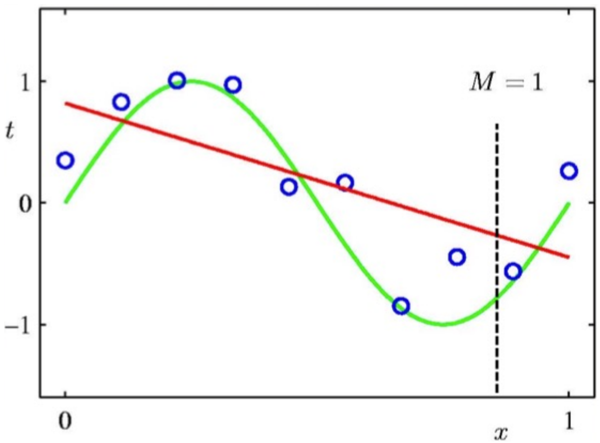
where $\frac{\partial}{\partial w_j}\left(J(\vec{w}, b)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$

$$b = b - \alpha\frac{\partial}{\partial b}\left(J(\vec{w}, b)\right),$$

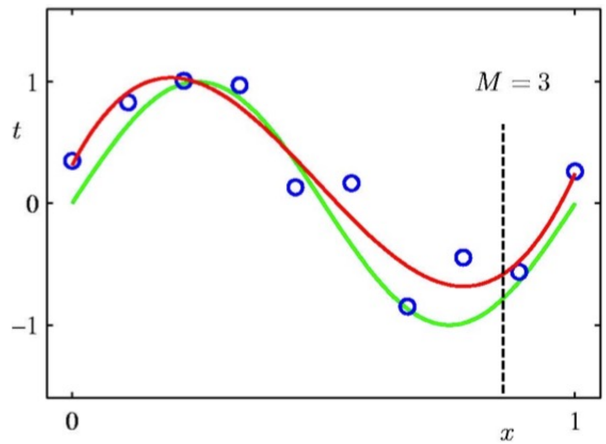
where $\frac{\partial}{\partial b}\left(J(\vec{w}, b)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)$

} simultaneous updates

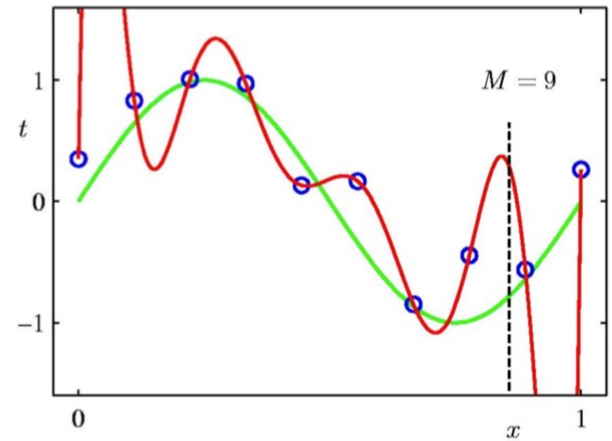
Polynomial Regression Examples



M=1



M=3



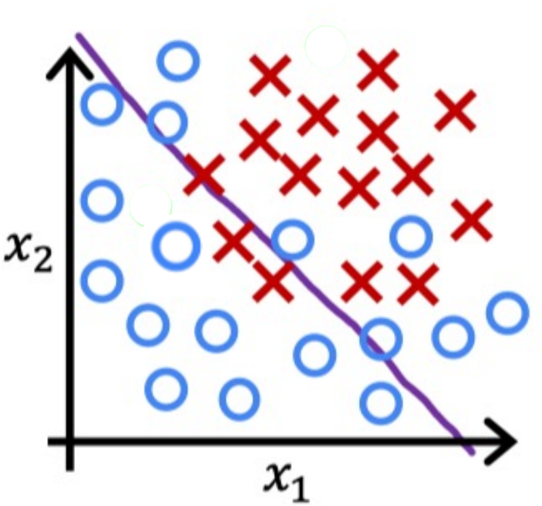
M=9

- **Underfitting**
- Does not fit the training set well
- Cannot fit the test set as well
- High bias

- **Just right**
- Fits training set pretty well
- Fits test set well
- Generalization

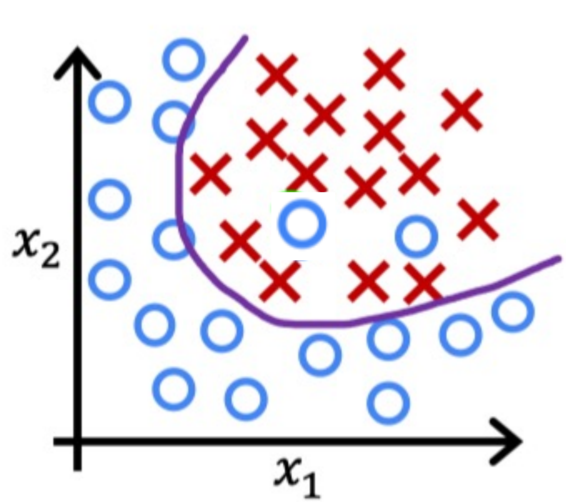
- **Overfitting**
- Fit the training set extremely well
- Cannot fit the test set as well
- High variance

Classification Examples



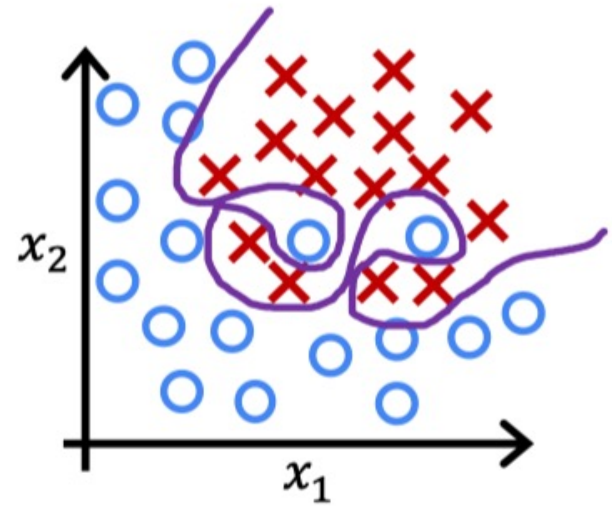
$z = w_1x_1 + w_2x_2 + b$

- **Underfitting**



$z = w_1 x_1 + w_2 x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$

- **Just right**

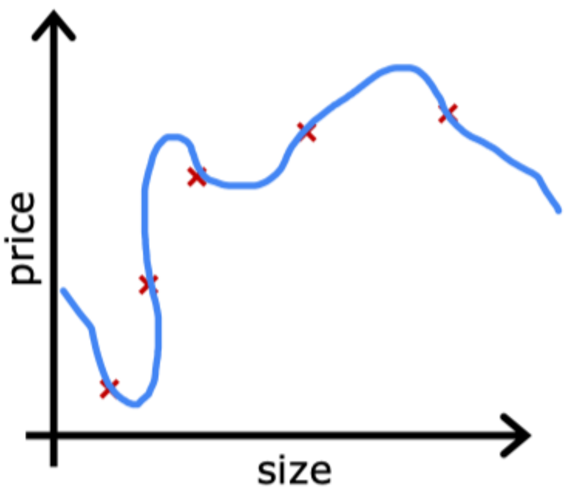


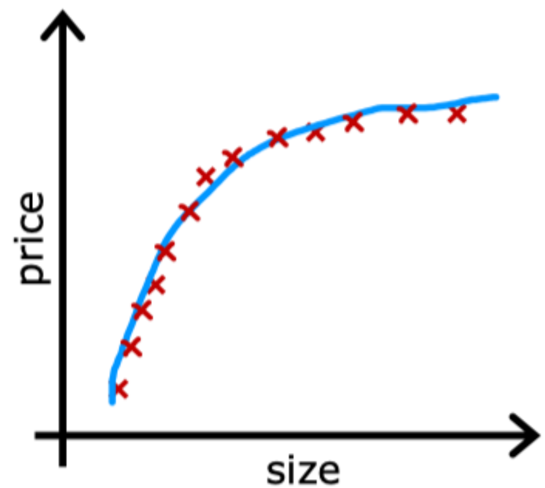
$z = w_1 x_1^3 + w_2 x_2^3 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + \ldots + b$

- **Overfitting**

Dealing with Overfitting

- Collect more training examples





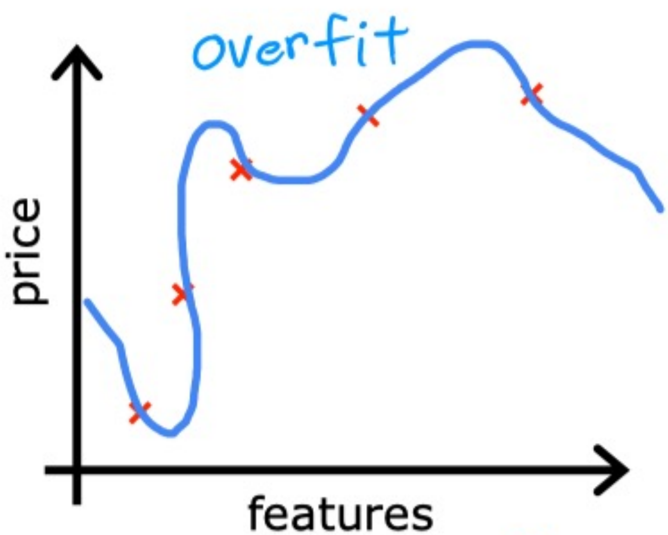
Dealing with Overfitting

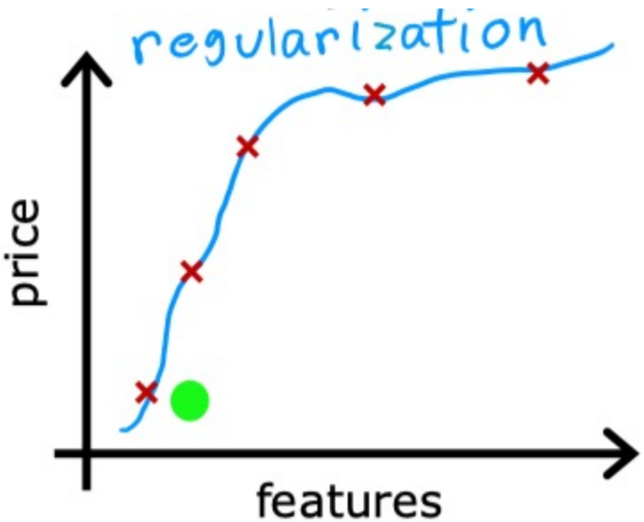
- Select features to include/exclude:
  - 100 features → 10 feature
  - 100 features + insufficient data → Overfitting
  - Just right 10 features + same data → Just right (possible)

- Disadvantage:
  - Useful features could be lost

Regularization

- Reduce the size of parameters **w**





$f(x)$
$= 28x - 385x^2 + 39x^3$
$- 174x^4 + 100$

$f(x)$
$= 13x - 0.23x^2 + 0.000014x^3$
$- 0.0001x^4 + 10$

Regularized Linear Regression

- Overall Loss with Regularizer:

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} {w_j}^2$$

- Gradient Decent:

Repeat {

$w_j = w_j - \alpha \frac{\partial}{\partial w_j}\left(J(\vec{w}, b)\right),$

where $\frac{\partial}{\partial w_j}\left(J(\vec{w}, b)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right){x_j}^{(i)} + \frac{\lambda}{m} w_j$

$b = b - \alpha \frac{\partial}{\partial b}\left(J(\vec{w}, b)\right),$

where $\frac{\partial}{\partial b}\left(J(\vec{w}, b)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)$

} simultaneous updates

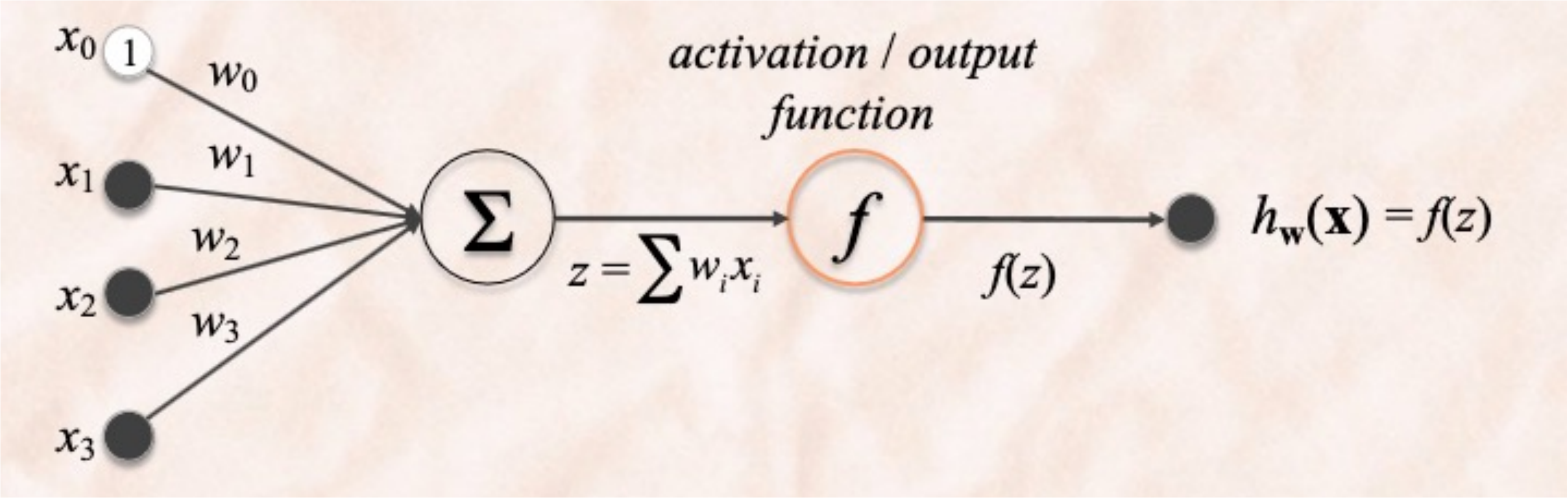
Machine Learning Objective

- Find a model **M**:
  - that ***fits the training data*** + that is ***simple***

$$\hat{\mathbf{M}} = \underset{\mathbf{M}}{\operatorname{argmin}} \; \mathit{Complexity}(\mathbf{M}) + \mathit{Error}(\mathbf{M}, \mathit{Data})$$

- **Inductive hypothesis**: Models that perform well on training examples are expected to do well on test (unseen) examples.
- **Occam's Razor**: Simpler models are expected to do better than complex models on test examples (assuming similar training performance).

Algebraic Interpretation



- The output of the neuron is a linear combination of inputs from other neurons, rescaled by the weights.
- summation corresponds to combination of signals
- It is often transformed through an **activation/output** function.

Questions?

UNIVERSITY OF NORTH CAROLINA CHARLOTTE