Foundations of Machine Learning (ECE 5984)

- Logistic Regression -

Eunbyung Park

Assistant Professor

School of Electronic and Electrical Engineering

Eunbyung Park (silverbottlep.github.io)

Classification

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

y can only be one of two values

false true

0 1

useful for classification

"binary classification"

class = category

"negative class"

≠ "bad"

absence

"positive class"

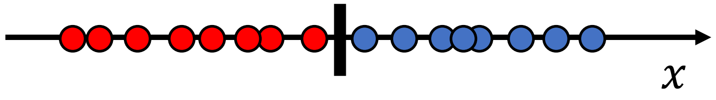
≠ "good"

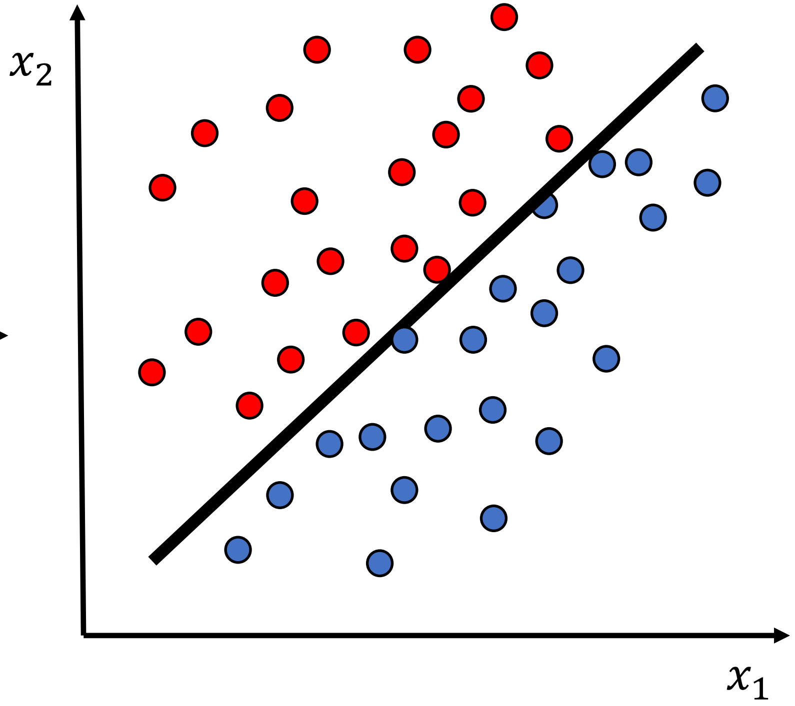
presence

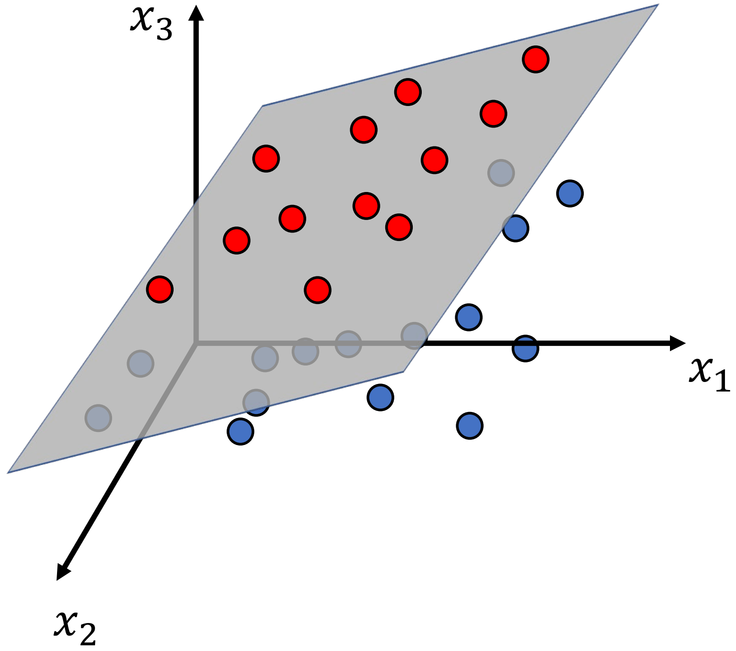
The Perceptron

Linear Classification

- Linear decision boundary







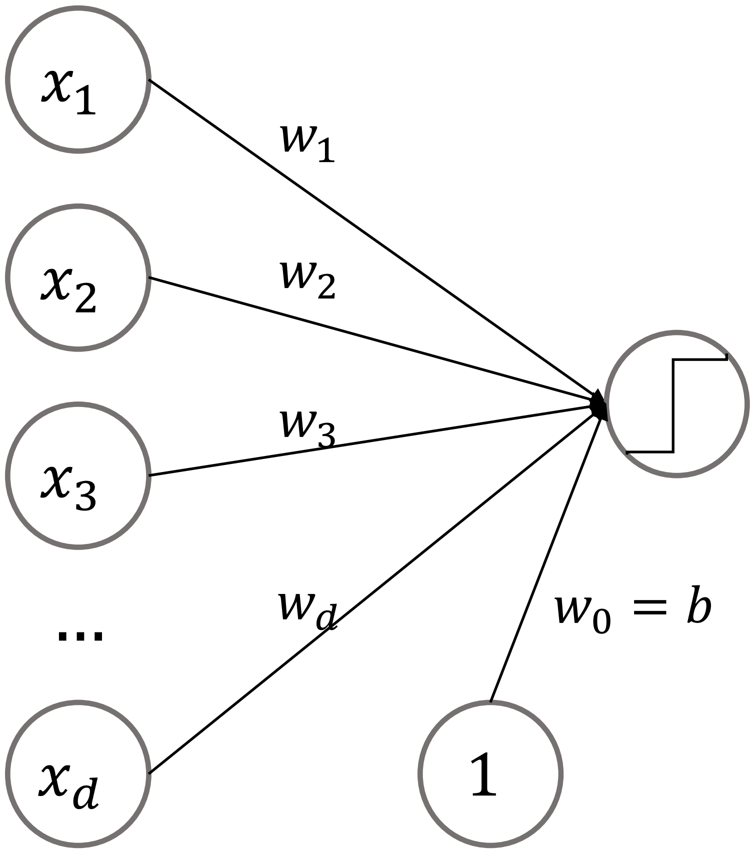
$x + b = 0$

$w_1 x_1 + w_2 x_2 + b = 0$

$w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$

Rosenblatt's Perceptron

- A single perceptron as a linear decision boundary (hyperplane)

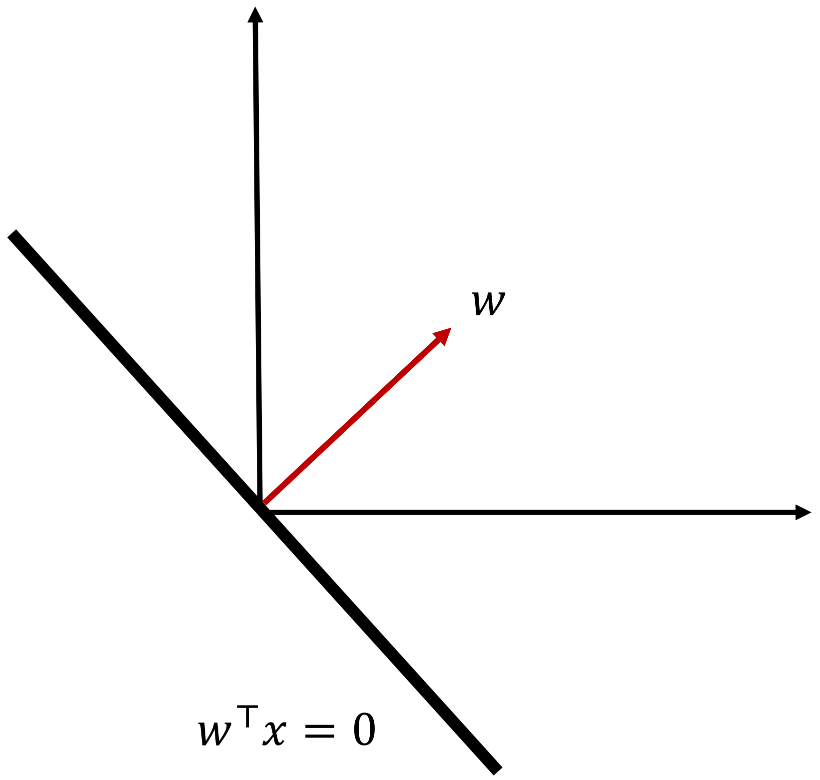


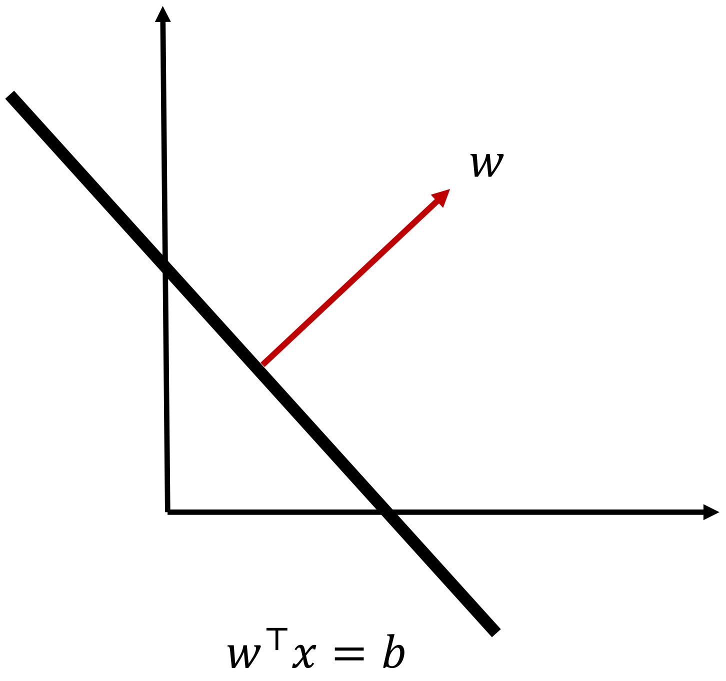
$$f(x) = \begin{cases} 1, & w^{\top}x \geq 0 \\ 0, & w^{\top}x < 0 \end{cases}$$

Rosenblatt, Psychological Review 1958

Perceptron

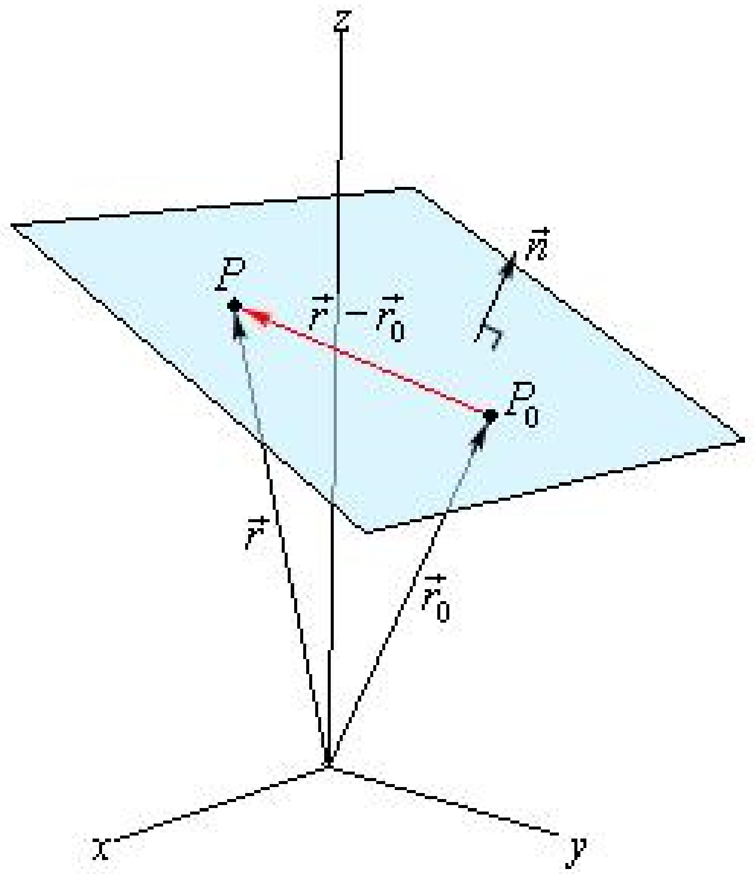
- Weight vector is orthogonal to the hyperplane





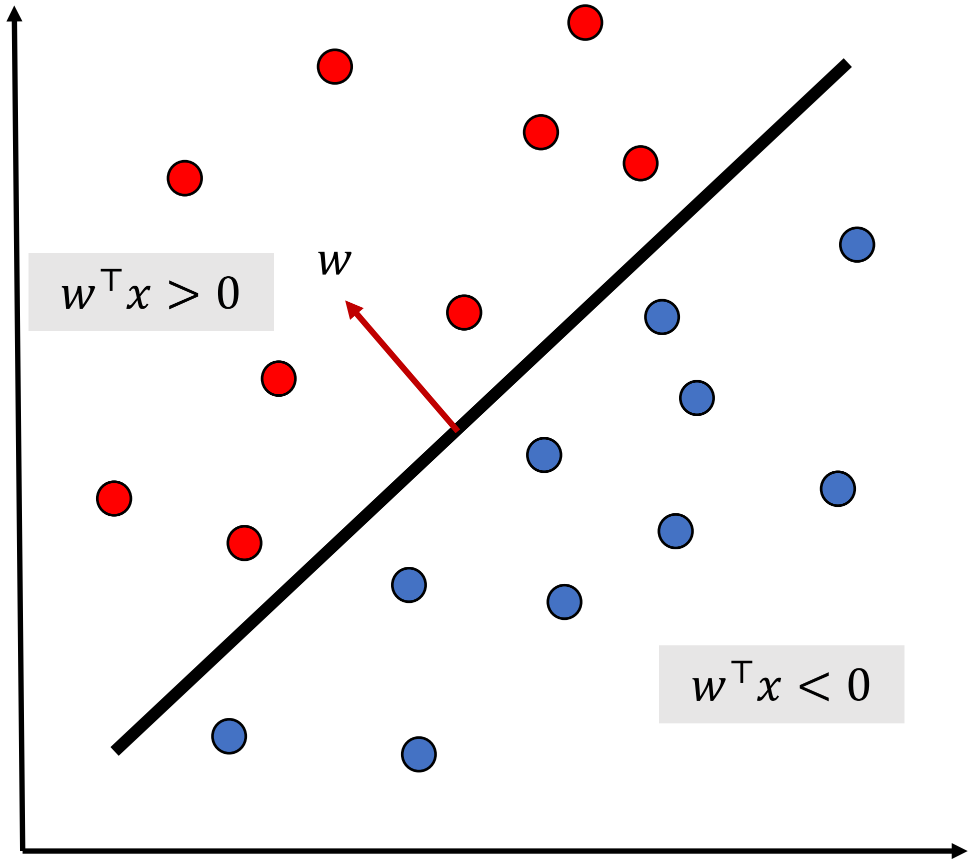
Perceptron

- Weight vector is orthogonal to the hyperplane



Perceptron

- Find a separating hyperplane

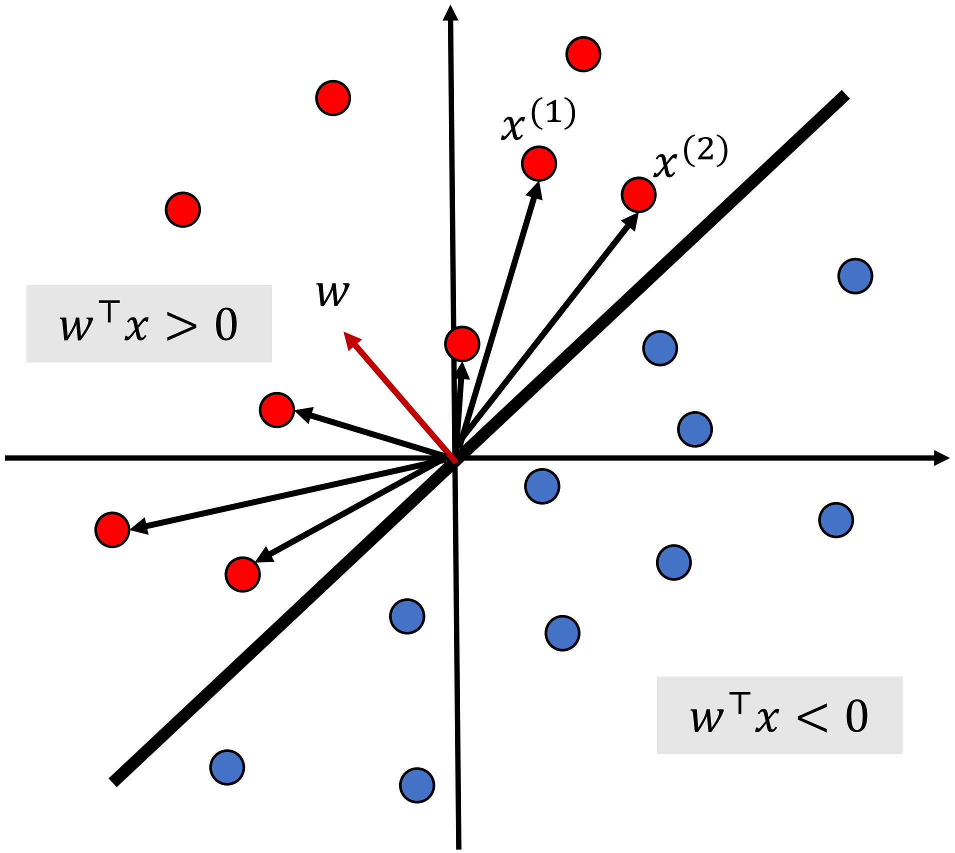


Perceptron

- Find a separating hyperplane

Angles between all positive examples $x^{(i)}$ and w should be less then ?? degree

Angles between all negative examples $x^{(i)}$ and w should be greater then ?? degree



Perceptron Learning Algorithm

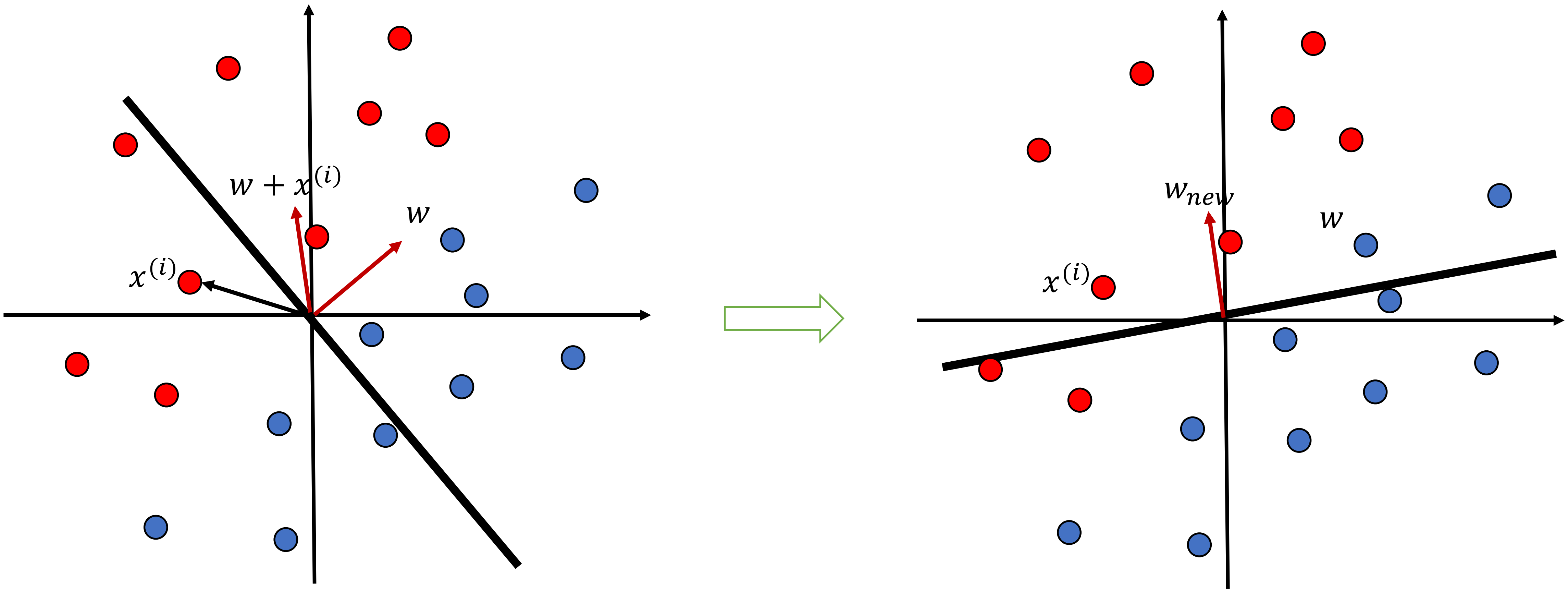
- Find the w vector that perfectly classify training examples

Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while !*convergence* **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\boxed{\mathbf{w} = \mathbf{w} + \mathbf{x}\ ;}$
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\boxed{\mathbf{w} = \mathbf{w} - \mathbf{x}\ ;}$
  end
end
//the algorithm converges when all the
inputs are classified correctly

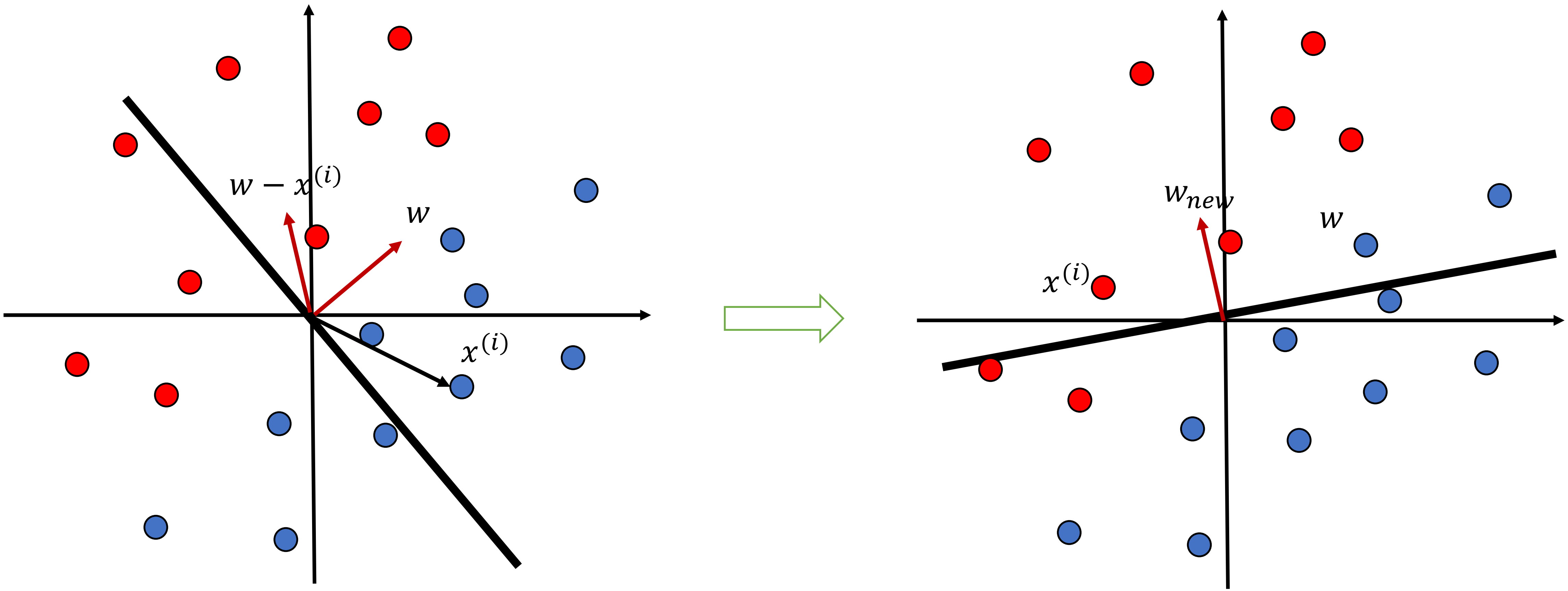
Perceptron Learning Algorithm

- Find a separating hyperplane



Perceptron Learning Algorithm

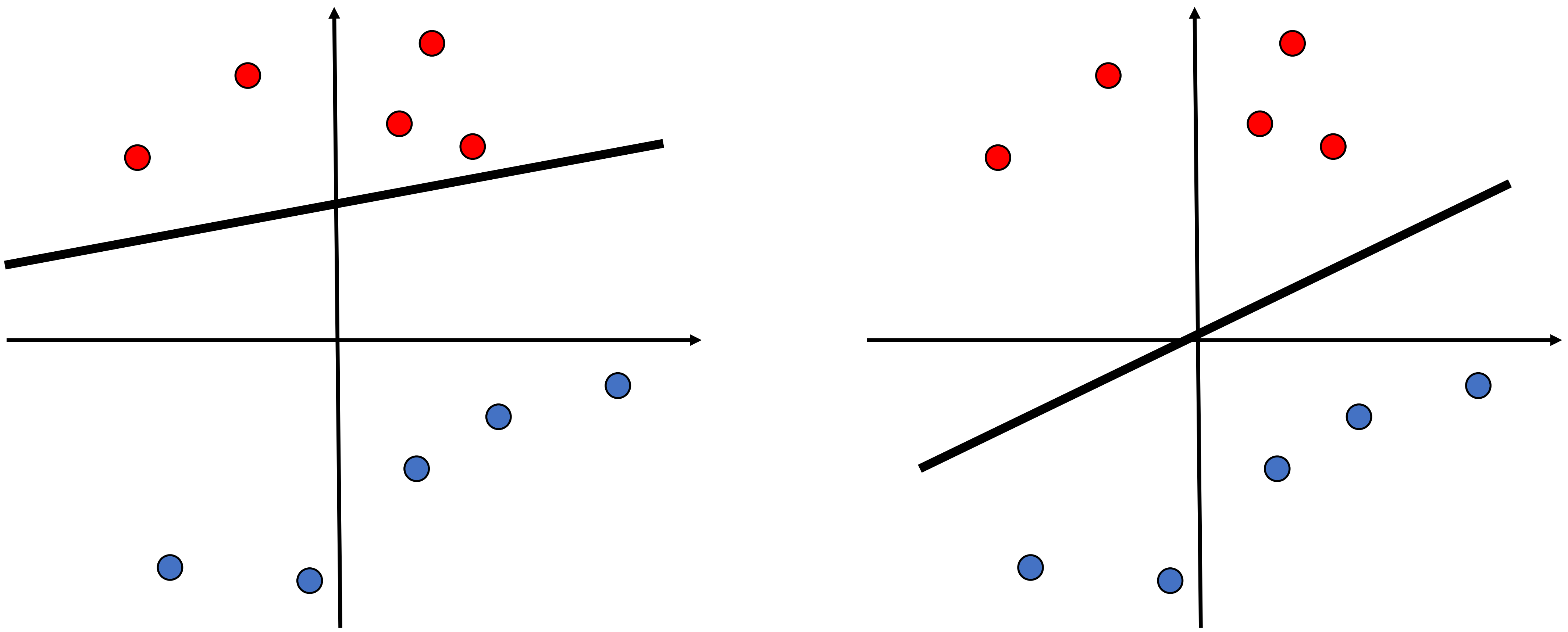
- Find a separating hyperplane



Logistic Regression

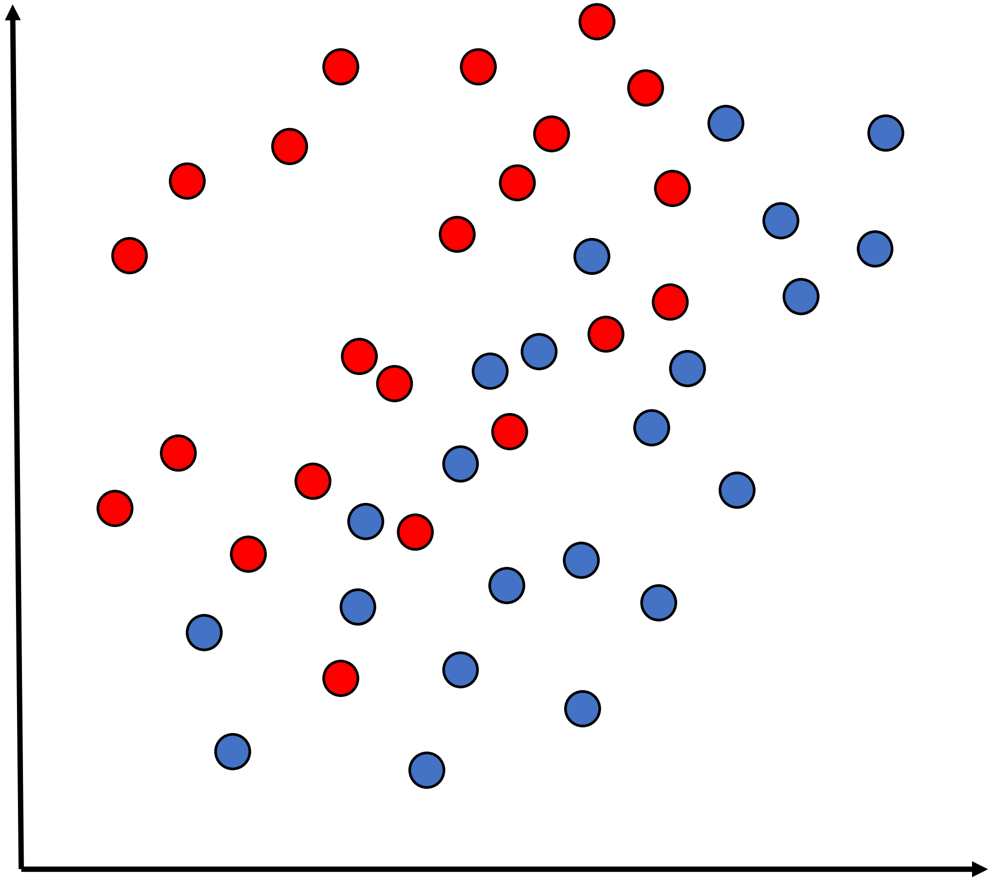
Problems of the Perceptron

- Which one is better?

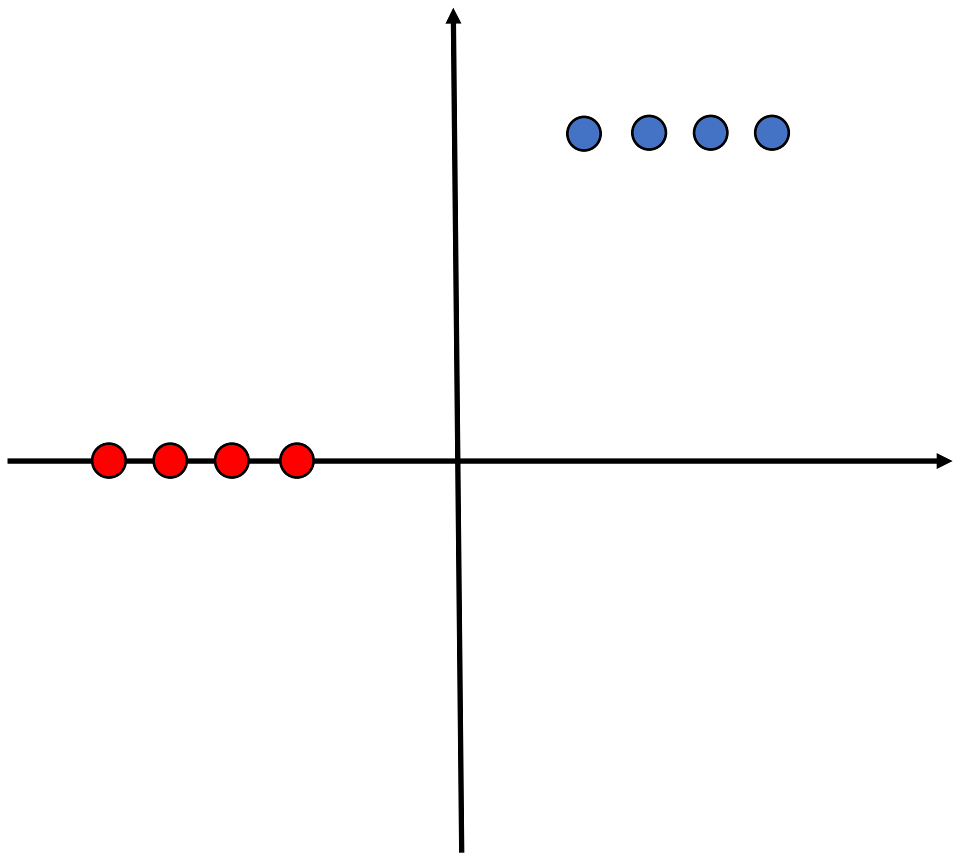


Problems of the Perceptron

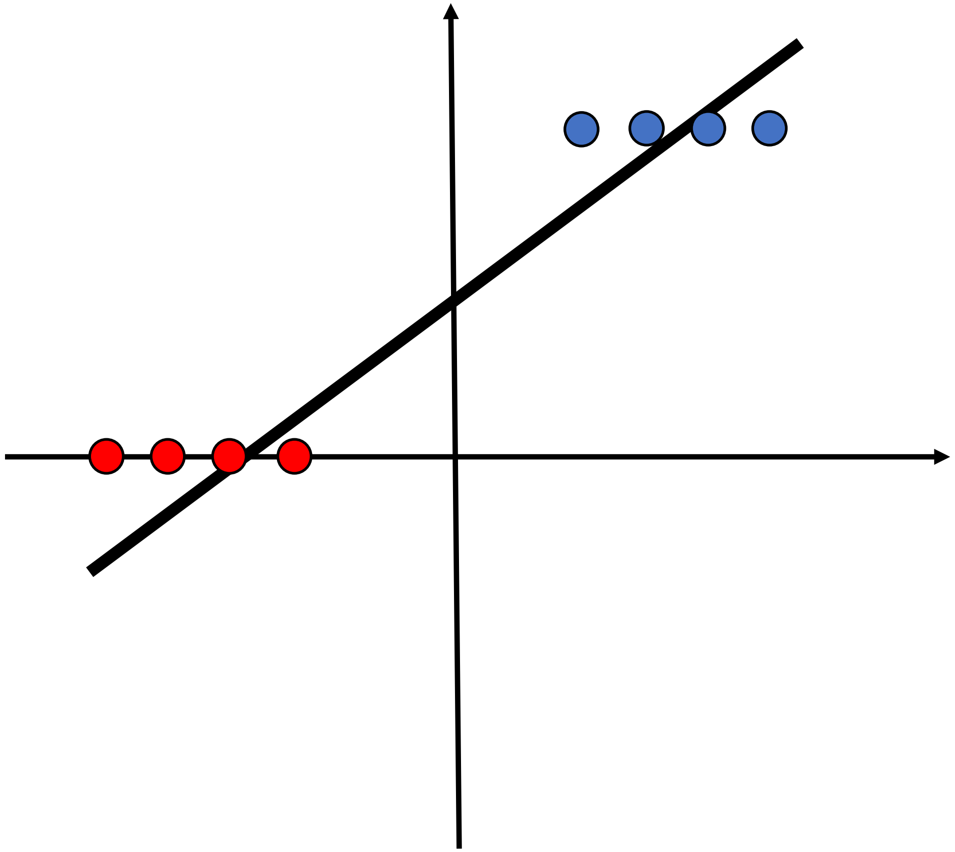
- What about not linearly separable cases?



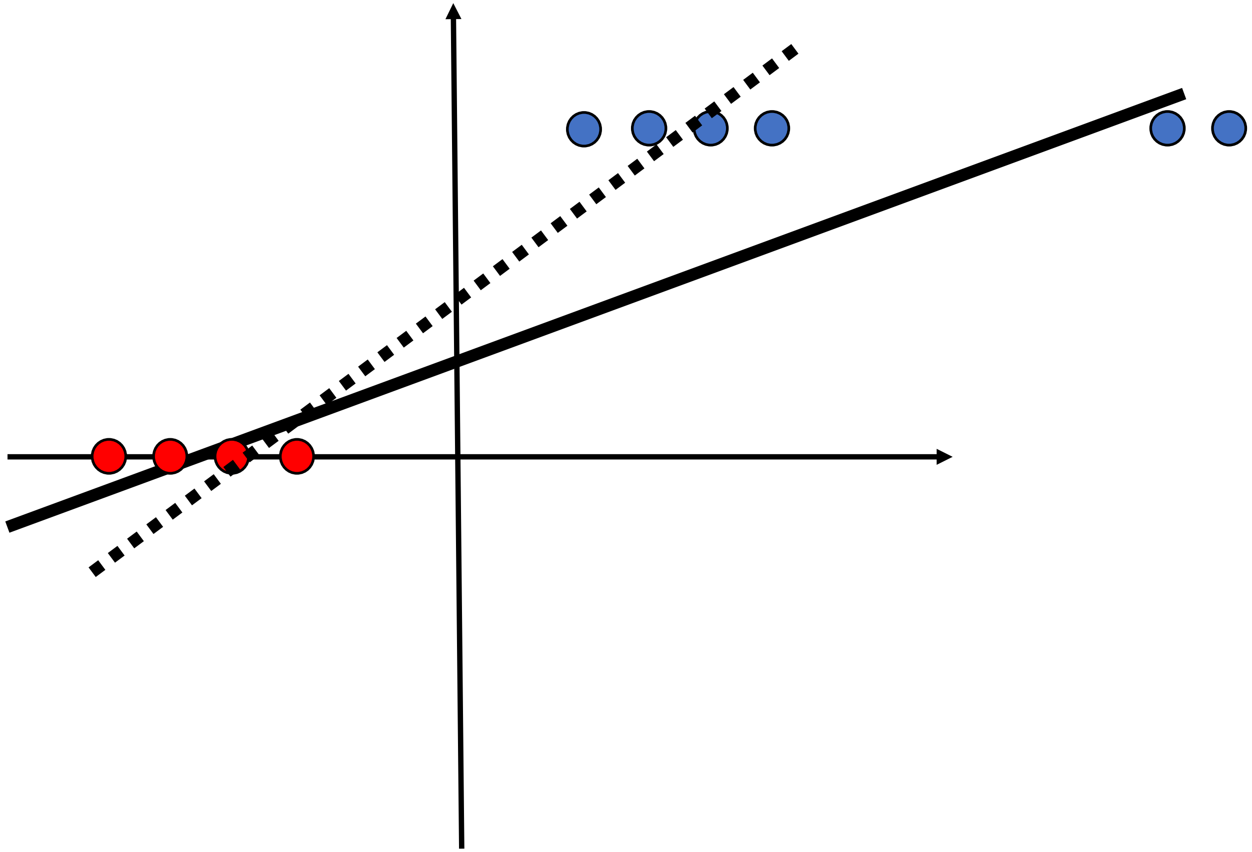
Classification w/ Linear Regression



Classification w/ Linear Regression

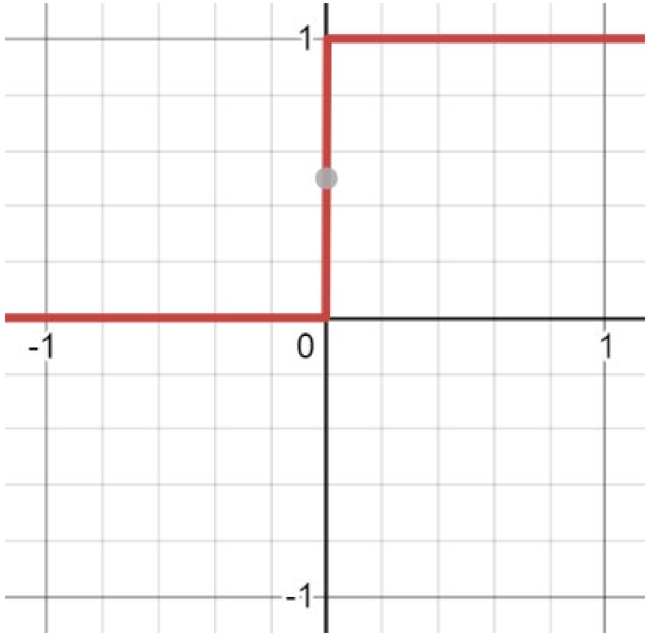


Classification w/ Linear Regression

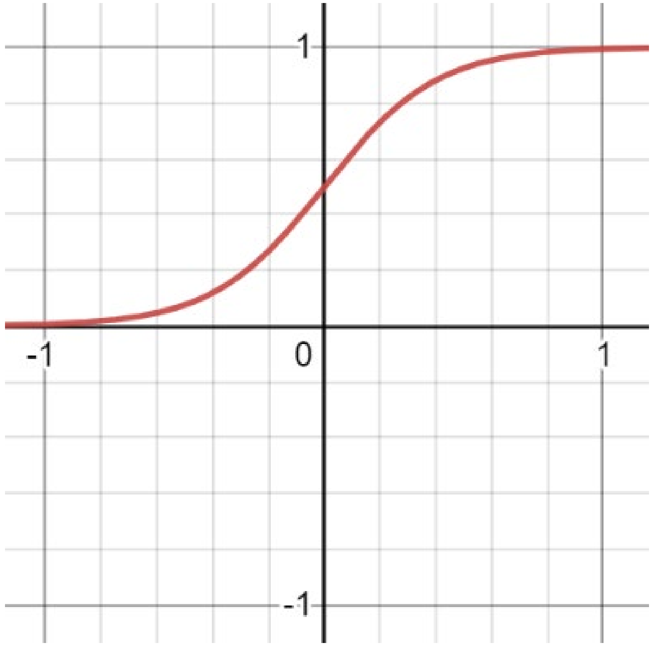


Logistic Function (aka Sigmoid)

- Squeezing the output of a 'linear equation' between 0 and 1



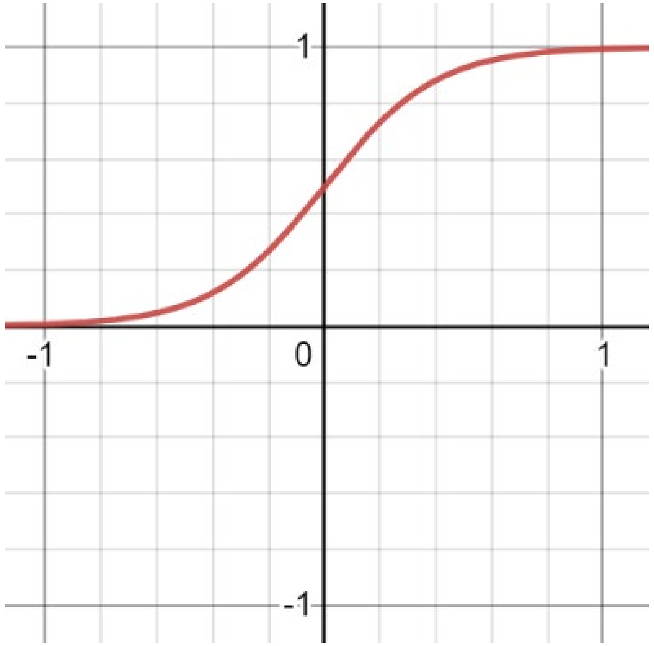
$$\text{step}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Logistic Function (aka Sigmoid)

- Squeezing the output of a ‘linear equation’ between 0 and 1



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$z = w^{\top} x$$

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

Logistic Regression

- $y = \sigma(w^\top x) = \frac{1}{1+e^{-w^\top x}}$

- Using the ‘logistic function’ to squeeze the output of a ‘linear equation’
  - $\sigma(w^\top x) \in [0,1]$ (w/ sigmoid)
  - $\mathrm{step}(w^\top x) \in \{0,1\}$ (thresholding)

- So, now it’s more like probability
  - $p(y = 1|x; w) = \sigma(w^\top x)$
  - $p(y = 0|x; w) = 1 - \sigma(w^\top x)$
  - $p(y = 0|x; w) = p(y = 1|x; w)$

MSE Loss for Logistic Regression

- Training set

	tumor size (cm) x_1	...	patient's age x_n	malignant? y
$i=1$	10		52	1
$\vdots$	2		73	0
	5		55	0
	12		49	1
$i=m$	...		...	...

MSE Loss for Logistic Regression

- Can we apply MSE loss function to logistic regression?

$$D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$$
$$x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{0,1\}, w \in \mathbb{R}^d$$
$$X \in \mathbb{R}^{N \times d}, Y \in \{0,1\}^N$$

$$\mathrm{MSE}(w) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \sigma(w^\top x^{(i)})\right)^2$$

Is it convex?

MSE Loss for Logistic Regression

- Convexity Check

Derivative of Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

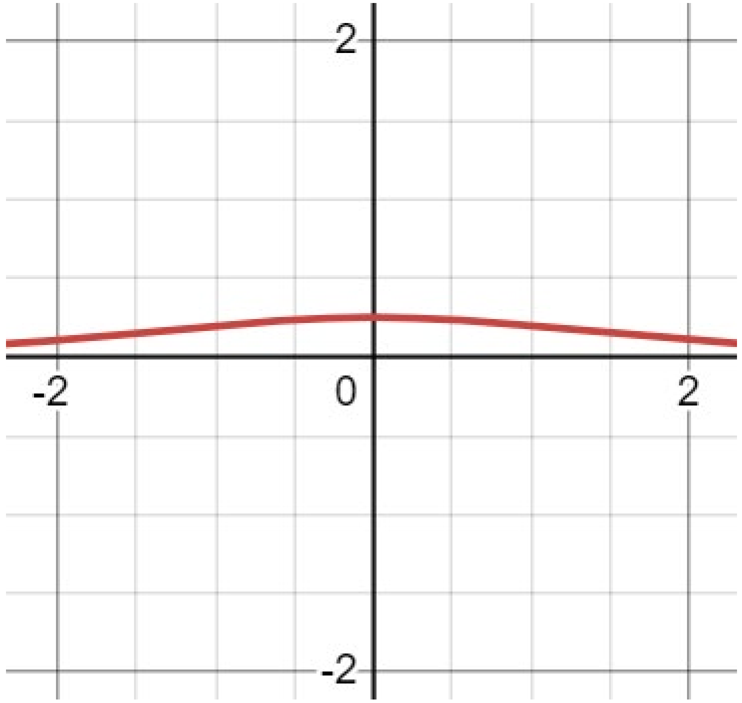
$$\frac{d\sigma(x)}{dx} =$$

Derivative of Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{1}{(1 + e^{-x})} \frac{e^{-x}}{(1 + e^{-x})}$$

$$= \sigma(x)(1 - \sigma(x))$$



MSE Loss for Logistic Regression

- Convexity Check in 1D

$$\frac{\partial^2 L(w)}{\partial w^2} \geq 0$$

$$L(w) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$\hat{y}^{(i)} = \sigma(wx^{(i)})$$

$$\frac{\partial L(w)}{\partial w} = \sum_{i=1}^{N} -\left(y^{(i)} - \hat{y}^{(i)}\right)\hat{y}^{(i)}\left(1 - \hat{y}^{(i)}\right)x^{(i)} = \sum_{i=1}^{N} -\left(y^{(i)}\hat{y}^{(i)} - y^{(i)}{\hat{y}^{(i)}}^2 - {\hat{y}^{(i)}}^2 + {\hat{y}^{(i)}}^3\right)x^{(i)}$$

$$\frac{\partial^2 L(w)}{\partial w^2} = \sum_{i=1}^{N} -\left(y^{(i)} - 2y^{(i)}\hat{y}^{(i)} - 2\hat{y}^{(i)} + 3{\hat{y}^{(i)}}^2\right)\underbrace{\hat{y}^{(i)}\left(1 - \hat{y}^{(i)}\right){x^{(i)}}^2}_{> 0}$$

MSE Loss for Logistic Regression

- Convexity Check in 1D

$$-3\hat{y}^{(i)^2} + 2\left(y^{(i)} + 1\right)\hat{y}^{(i)} - y^{(i)} \; ?$$

$$y^{(i)} \in \{0,1\}$$

if $y^{(i)} = 0$

$$-3\hat{y}^{(i)^2} + 2\hat{y}^{(i)} = -3\left(\hat{y}^{(i)} - \frac{2}{3}\right)\hat{y}^{(i)}$$

$$\hat{y}^{(i)} \in \left[0, \frac{2}{3}\right] \qquad \hat{y}^{(i)} \in \left[\frac{2}{3}, 1\right]$$

$$> 0 \qquad < 0$$

MSE Loss for Logistic Regression

- Convexity Check in 1D

$$-3{\hat{y}^{(i)}}^2 + 2\left(y^{(i)} + 1\right)\hat{y}^{(i)} - y^{(i)} \text{ ?}$$

$$y^{(i)} \in \{0,1\}$$

if $y^{(i)} = 1$

$$-3{\hat{y}^{(i)}}^2 + 4\hat{y}^{(i)} - 1 = -3\left(\hat{y}^{(i)} - \frac{1}{3}\right)\left(\hat{y}^{(i)} - 1\right)$$

$$\hat{y}^{(i)} \in \left[\frac{1}{3}, 1\right] \qquad \hat{y}^{(i)} \in \left[0, \frac{1}{3}\right]$$

$$> 0 \qquad < 0$$

MSE Loss for Logistic Regression

- Convexity Check in 2D

Log Loss

Log Loss (a.k.a Logistic Loss, Binary Cross Entropy)

$$D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$$

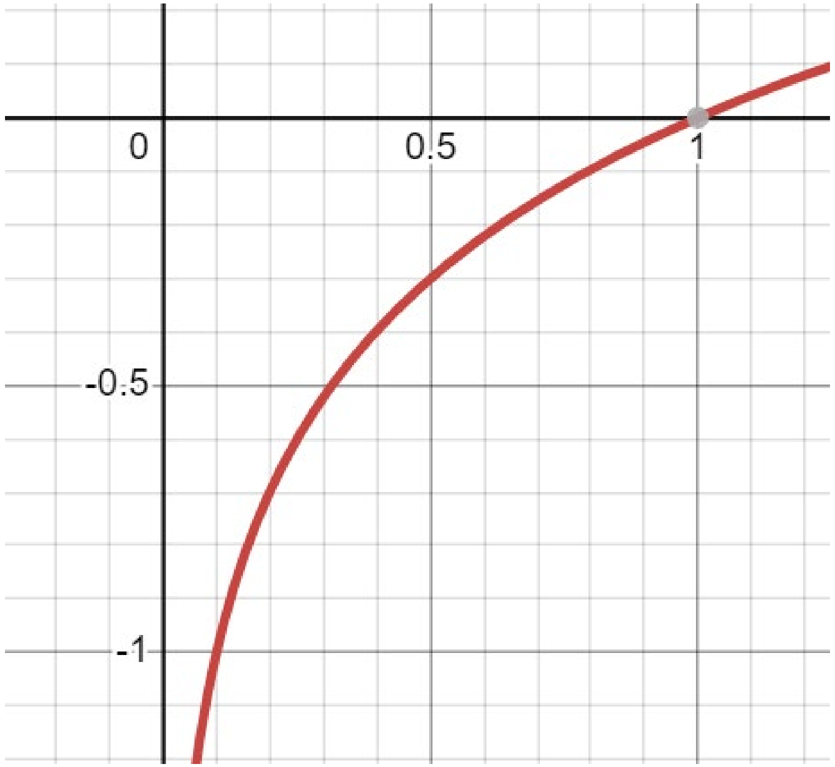
$$x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{0,1\}, w \in \mathbb{R}^d$$

$$\hat{y}^{(i)} = \sigma\left(w^\top x^{(i)}\right)$$

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$-\log(1-\hat{y}), \qquad y^{(i)} = 0$$

$$-\log(\hat{y}), \qquad y^{(i)} = 1$$

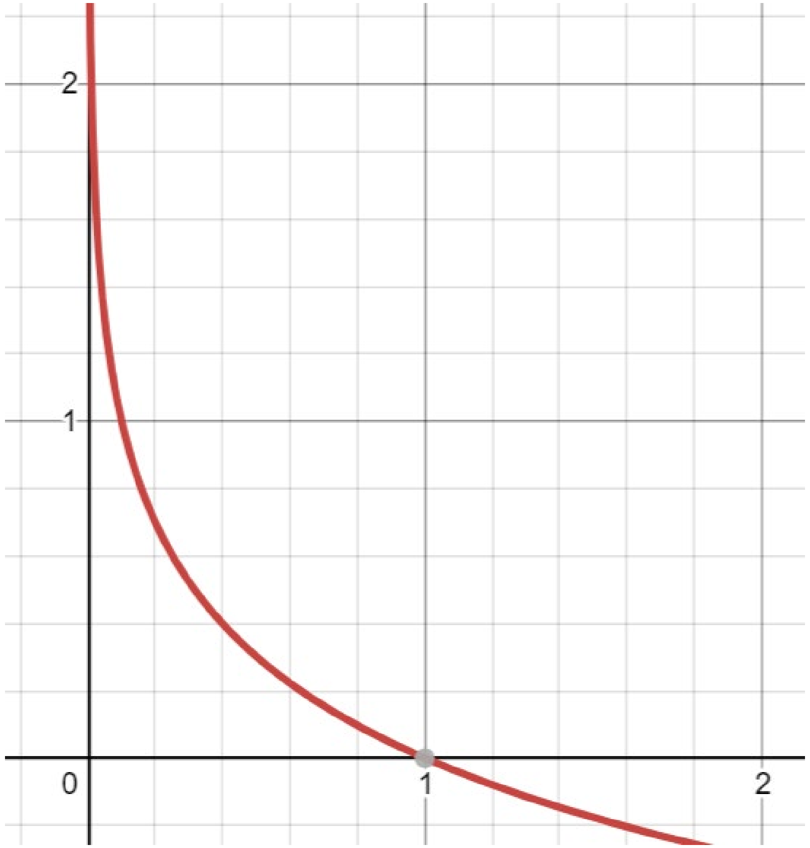


$\log(x)$

Log Loss (a.k.a Logistic Loss, Binary Cross Entropy)

if $y^{(i)} = 1$,

$-\log(\hat{y})$

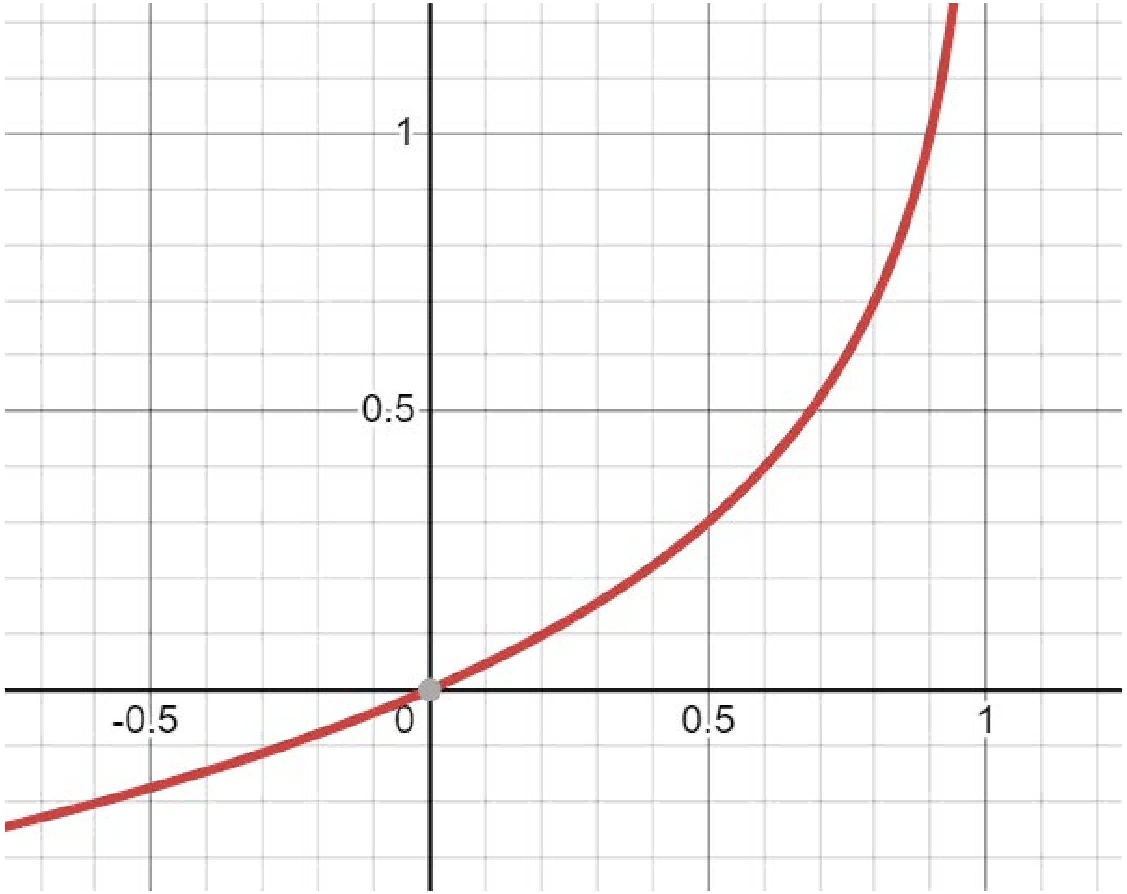


$-\log(\hat{y})$

Log Loss (a.k.a Logistic Loss, Binary Cross Entropy)

if $y^{(i)} = 0$,

$-\log(1 - \hat{y})$



$-\log(1 - \hat{y})$

Log Loss (a.k.a Logistic Loss, Binary Cross Entropy)

- Convexity Check in 1D

if $y^{(i)} = 1,$ $$L(w) = -\sum_{i=1}^{N} \log\left(\hat{y}^{(i)}\right)$$ $$\hat{y}^{(i)} = \sigma\left(wx^{(i)}\right)$$

$$\frac{\partial L(w)}{\partial w} =$$

$$\frac{\partial^2 L(w)}{\partial w^2} =$$

Log Loss (a.k.a Logistic Loss, Binary Cross Entropy)

- Convexity Check in 1D

if $y^{(i)} = 0,$

$$L(w) = -\sum_{i=1}^{N} \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y}^{(i)} = \sigma(wx^{(i)})$$

$$\frac{\partial L(w)}{\partial w} =$$

$$\frac{\partial^2 L(w)}{\partial w^2} =$$

Solving Logistic Regression

- Is it convex?
- Does it have a closed form solution?

Gradient Descent

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y}^{(i)} = \sigma\left(w^\top x^{(i)}\right)$$

$$\frac{\partial \mathrm{BCE}(w)}{\partial w_j} =$$

Gradient Descent

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y}^{(i)} = \sigma\left(w^{\top} x^{(i)}\right)$$

$$\frac{\partial \mathrm{BCE}(w)}{\partial w_j} = \sum_{i=1}^{N} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

$$w_j \coloneqq w_j - \alpha\left(\sum_{i=1}^{N} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}\right)$$

(Gradient Descent)

Algorithm: Perceptron Learning Algorithm

$P \leftarrow$ *inputs with label* 1;
$N \leftarrow$ *inputs with label* 0;
Initialize $\mathbf{w}$ randomly;
while !*convergence* **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P$ *and* $\mathbf{w.x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N$ *and* $\mathbf{w.x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

Gradient Descent

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y}^{(i)} = \sigma\left(w^{\top} x^{(i)}\right)$$

$$w \in \mathbb{R}^{d}$$

$$Y \in \mathbb{R}^{N}$$

$$X \in \mathbb{R}^{N \times d}$$

$$\frac{\partial \mathrm{BCE}(w)}{\partial w} = ?$$

Gradient Descent

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

$$\hat{y}^{(i)} = \sigma\left(w^{\top} x^{(i)}\right)$$

$$w \in \mathbb{R}^{d}$$

$$\frac{\partial \mathrm{BCE}(w)}{\partial w} = X^{\top}(\sigma(Xw) - Y)$$

$$Y \in \mathbb{R}^{N}$$

$$X \in \mathbb{R}^{N \times d}$$

$$w \coloneqq w - \alpha(X^{\top}(\sigma(Xw) - Y))$$

(Gradient Descent)

MLE

MLE for Logistic Regression

- Bernoulli distribution

parameter

$$p(x;p) = p^x(1-p)^{1-x}, \qquad x \in \{0,1\}$$

$$\begin{cases} x = 0, & 1-p \\ x = 1, & p \end{cases}$$

$$E[x] = p$$

$$\sum_{x\in\{0,1\}} xp(x) = 1 \cdot p + 0 \cdot (1-p) = p$$

MLE for Logistic Regression

- Finding the parameters that maximize 'conditional likelihood'

Assumption1: $p(y|x)$ is a Bernoulli distribution

Assumption2: I.I.D

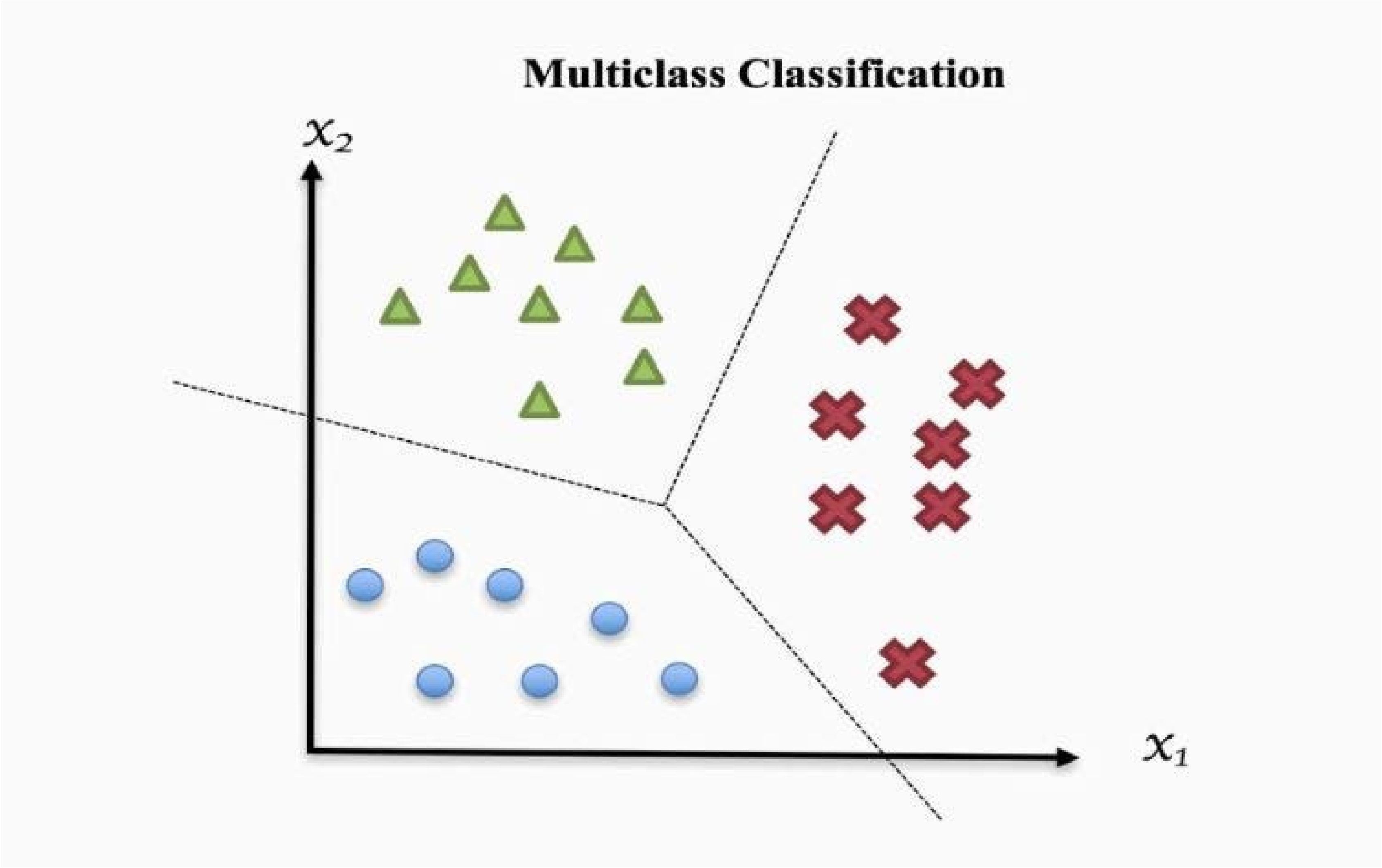
$$\log L(w) = \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}; w) = \sum_{i=1}^{N} \log \sigma\left(w^\top x^{(i)}\right)^{y^{(i)}} \left(1 - \sigma\left(w^\top x^{(i)}\right)\right)^{1-y^{(i)}}$$

$$= \sum_{i=1}^{N} y^{(i)} \log \sigma\left(w^\top x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \sigma\left(w^\top x^{(i)}\right)\right)$$

a.k.a Binary Cross Entropy (BCE) Loss

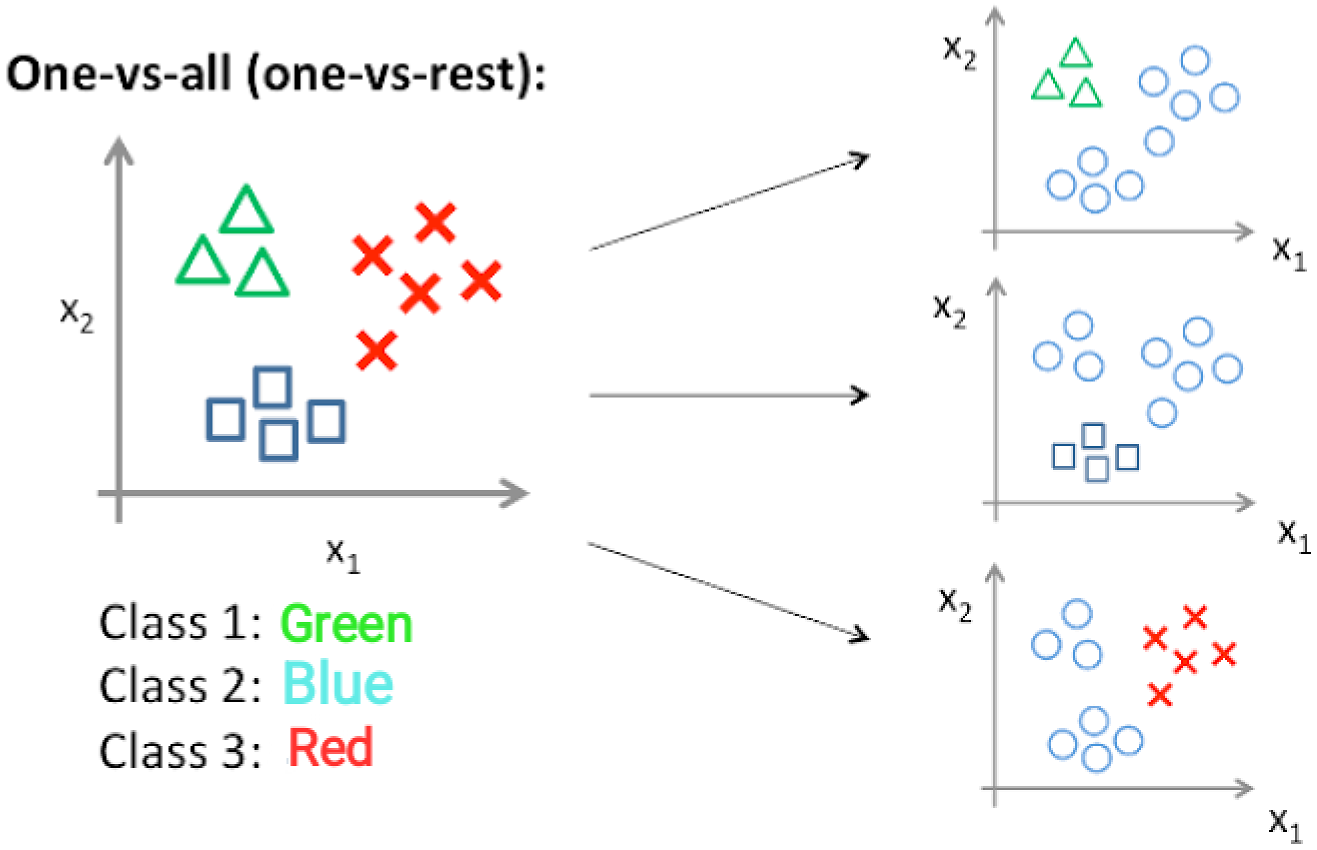
Multiclass Classification

Multiclass Classification



One vs. All for Multiclass Classification

- Sigmoid function and binary logistic regression



Softmax Function

- Sigmoid function and binary logistic regression

$$\sigma(x) = \frac{1}{1 + e^{-x}} \qquad y = \sigma(w^\top x)$$

Softmax Function

- 'Soft' 'Max' function
  - $[1,2,3,2,1] \rightarrow [0.0674, 0.183\ , 0.498, 0.183\ , 0.0674]$

$$\text{softmax: } \mathbb{R}^C \rightarrow [0,1]^C$$

$$||\text{softmax}||_1 = 1$$

$$\text{softmax}(z)_j = \frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}}$$

$$\text{softmax}(z) = \begin{bmatrix} \frac{e^{z_1}}{\sum_{i=1}^{C} e^{z_i}} \\ \frac{e^{z_2}}{\sum_{i=1}^{C} e^{z_i}} \\ \vdots \\ \frac{e^{z_C}}{\sum_{i=1}^{C} e^{z_i}} \end{bmatrix} \in [0,1]^C$$

Multiclass Classification w/ Softmax Function

- Weight vectors for each class!

$$\hat{y}^{(i)} = \text{softmax}\left(Wx^{(i)}\right) \in \mathbb{R}^C \qquad W \in \mathbb{R}^{c\times d} \qquad w_k \in \mathbb{R}^d$$

$$p(y=0|x) =$$

Multiclass Classification w/ Softmax Function

- Weight vectors for each class!

$$\hat{y}^{(i)} = \mathrm{softmax}\left(W x^{(i)}\right) \in \mathbb{R}^{C} \qquad W \in \mathbb{R}^{c \times d} \qquad w_k \in \mathbb{R}^{d}$$

$$p(y = 0|x) = \frac{e^{w_0^\top x}}{e^{w_0^\top x} + e^{w_1^\top x} + e^{w_2^\top x}}$$

$$p(y = 1|x) = \frac{e^{w_1^\top x}}{e^{w_0^\top x} + e^{w_1^\top x} + e^{w_2^\top x}}$$

$$p(y = 2|x) = \frac{e^{w_2^\top x}}{e^{w_0^\top x} + e^{w_1^\top x} + e^{w_2^\top x}}$$

Multiclass Classification w/ Softmax Function

- When they 2 classes

$$p(y = 0|x) =$$

$$p(y = 1|x) =$$

Multiclass Classification w/ Softmax Function

- When they 2 classes

$$w^* = -(w_0 - w_1)$$

$$p(y=0|x) = \frac{e^{w_0^\top x}}{e^{w_0^\top x} + e^{w_1^\top x}} = \frac{e^{w_0^\top x}}{e^{w_0^\top x} + e^{w_1^\top x}} \frac{e^{-w_1^\top x}}{e^{-w_1^\top x}} = \frac{e^{(w_0 - w_1)^\top x}}{1 + e^{(w_0 - w_1)^\top x}} = \frac{e^{-w^{*\top} x}}{1 + e^{-w^{*\top} x}}$$

$$p(y=1|x) = \frac{e^{w_1^\top x}}{e^{w_0^\top x} + e^{w_1^\top x}} = \frac{e^{w_1^\top x}}{e^{w_0^\top x} + e^{w_1^\top x}} \frac{e^{-w_1^\top x}}{e^{-w_1^\top x}} = \frac{1}{1 + e^{-w^{*\top} x}}$$

$$1 - \frac{1}{1 + e^{-w^{*\top} x}} = \frac{e^{-w^{*\top} x}}{1 + e^{-w^{*\top} x}}$$

Categorical Distribution

- Categorical distribution can be used to model a random variable X that takes values in {1, …, C}

$$p(x;\phi) = \phi^x(1-\phi)^{1-x}$$

Bernoulli distribution

$$p(x = i) = \phi_i \qquad \phi_{1,\dots,}\phi_{C-1}$$

$$\sum_{i=1}^{C} \phi_i = 1 \qquad 1 - \sum_{i=1}^{C-1} \phi_i = \phi_C$$

$$p(x) = \prod_{i=1}^{C} \phi_i^{\mathbb{I}_i(x)} = \phi_1^{\mathbb{I}_1(x)} \phi_2^{\mathbb{I}_2(x)} \dots \phi_C^{\mathbb{I}_C(x)}$$

$$\mathbb{I}_i(x) = \begin{cases} 1 & \text{if } x == i \\ 0 & \text{otherwise} \end{cases}$$

MLE w/ categorical distribution

$$p(y|x) = \prod_{i=1}^{N} \prod_{j=1}^{C} \phi_j^{\mathbb{I}_j(y^{(i)})} \qquad y^{(i)} \in \{1, \ldots, C\}$$

$$\log p(y|x) =$$

MLE w/ categorical distribution

$$p(y|x) = \prod_{i=1}^{N} \prod_{j=1}^{C} \phi_j^{\mathbb{I}_j(y^{(i)})}$$

$$\log p(y|x) = \sum_{i=1}^{N} \log \prod_{j=1}^{C} \phi_j^{\mathbb{I}_j(y^{(i)})} = \sum_{i=1}^{N} \sum_{j=1}^{C} \log \phi_j^{\mathbb{I}_j(y^{(i)})} = \sum_{i=1}^{N} \sum_{j=1}^{C} \mathbb{I}_j(y^{(i)}) \log \phi_j$$

Cross Entropy Loss

- Cross Entropy Loss
  - BCE is a special case of CE (two classes)

$$\mathrm{CE}(w) = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_c^{(i)} \log\left(\hat{y}_c{}^{(i)}\right)$$

$$\hat{y}^{(i)} = \mathrm{softmax}\left(Wx^{(i)}\right) \in \mathbb{R}^C \qquad W \in \mathbb{R}^{c\times d}$$

$$y^{(i)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$ (one-hot vector) $\mathbb{I}_j(y^{(i)})$

$$\mathrm{BCE}(w) = -\sum_{i=1}^{N} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

Derivative of the Softmax Function

$$y_j = \frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}}$$

$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{g(x)f'(x) - f(x)g'(x)}{\left(g(x)\right)^2}$$

1) $i \neq j$

$$\frac{\partial y_i}{\partial z_j} = \frac{\left(\sum_{k=1}^{C} e^{z_k}\right) \cdot 0 - e^{z_i} e^{z_j}}{\left(\sum_{k=1}^{C} e^{z_k}\right)^2} = -y_i y_j$$

2) $i = j$

$$\frac{\partial y_i}{\partial z_j} = \frac{\left(\sum_{k=1}^{C} e^{z_k}\right) \cdot e^{z_i} - e^{z_i} e^{z_i}}{\left(\sum_{k=1}^{C} e^{z_k}\right)^2} = y_i - y_i^2 = y_i(1 - y_i)$$

Derivative of the Softmax Function

$$y_j = \frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}}$$

$$\frac{\partial y_i}{\partial z_j} = \begin{cases} y_i(1 - y_i), & i = j \\ -y_i y_j, & i \neq j \end{cases} = y_i\left(1\{i = j\} - y_j\right)$$

$$\frac{dy}{dz} = \begin{bmatrix} y_1(1 - y_1) & \cdots & -y_1 y_C \\ \vdots & \ddots & \vdots \\ -y_C y_1 & \cdots & y_C(1 - y_C) \end{bmatrix}$$

Cross-Entropy + Softmax

$$\log \mathrm{CE}(W) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \qquad \hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \qquad z_c = W_c^{\top} x \qquad W_c \in \mathbb{R}^d \quad W \in \mathbb{R}^{c\times d}$$

$$\frac{\partial \hat{y}_i}{\partial z_j} = \begin{cases} \hat{y}_i(1-\hat{y}_i), & i = j \\ -\hat{y}_i\hat{y}_j, & i \neq j \end{cases}$$

$$\frac{\partial L}{\partial z_j} = -\frac{\partial}{\partial z_j}\sum_{i=1}^{C} y_i \log(\hat{y}_i) = -\sum_{i=1}^{C} y_i \frac{\partial \log(\hat{y}_i)}{\partial z_j} = -\sum_{i=1}^{C} \frac{y_i}{\hat{y}_i}\frac{\partial \hat{y}_i}{\partial z_j}$$

Cross-Entropy + Softmax

$$\log \mathrm{CE}(W) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \qquad \hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \qquad z_c = W_c^{\top} x \qquad W_c \in \mathbb{R}^d \quad W \in \mathbb{R}^{c\times d}$$

$$\frac{\partial \hat{y}_i}{\partial z_j} = \begin{cases} \hat{y}_i(1-\hat{y}_i), & i = j \\ -\hat{y}_i\hat{y}_j, & i \neq j \end{cases}$$

$$\frac{\partial \mathrm{CE}(W)}{\partial z_j} = -\frac{\partial}{\partial z_j}\sum_{i=1}^{C} y_i \log(\hat{y}_i) = -\sum_{i=1}^{C} y_i \frac{\partial \log(\hat{y}_i)}{\partial z_j} = -\sum_{i=1}^{C} \frac{y_i}{\hat{y}_i}\frac{\partial \hat{y}_i}{\partial z_j}$$

$$= -\frac{y_j}{\hat{y}_j}\frac{\partial \hat{y}_j}{\partial z_j} - \sum_{i\neq j}^{C} \frac{y_i}{\hat{y}_i}\frac{\partial \hat{y}_i}{\partial z_j} = -\frac{y_j}{\hat{y}_j}\hat{y}_j\left(1-\hat{y}_j\right) + \sum_{i\neq j}^{C} \frac{y_i}{\hat{y}_i}\hat{y}_i\hat{y}_j$$

$$= -y_j + y_j\hat{y}_j + \sum_{i\neq j}^{C} y_i\hat{y}_j = -y_j + \hat{y}_j\sum_{i=1}^{C} y_i = \boldsymbol{\hat{y}_j - y_j}$$

$$\boxed{\frac{dL}{dz} = \hat{y} - y}$$

Gradient Descent

$$\frac{\partial}{\partial W_{c,j}} \mathrm{CE}(W) = \frac{\partial \mathrm{CE}(W)}{\partial z} \frac{\partial z}{\partial W_{c,j}} = \sum_{i=1}^{N} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

$$W_{c,j} \coloneqq W_{c,j} - \alpha \left(\sum_{i=1}^{N} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)} \right)$$

(Gradient Descent)

Gradient Descent

$$\frac{\partial}{\partial W_{c,j}}\mathrm{CE}(W) = \frac{\partial \mathrm{CE}(W)}{\partial z}\frac{\partial z}{\partial W_{c,j}} = \sum_{i=1}^{N}\left(\hat{y}^{(i)} - y^{(i)}\right)x_j^{(i)}$$

$$\frac{\partial}{\partial W}\mathrm{CE}(W) = \frac{\partial \mathrm{CE}(W)}{\partial z}\frac{\partial z}{\partial W_{c,j}} = (\mathrm{softmax}(WX) - Y)X^\top$$

$$W \in \mathbb{R}^{c\times d}$$

$$Y \in \mathbb{R}^{c\times N}$$

$$X \in \mathbb{R}^{d\times N}$$

$$\boxed{W \coloneqq W - \alpha(\mathrm{softmax}(WX) - Y)X^\top}$$

(Gradient Descent)