Binary Classification

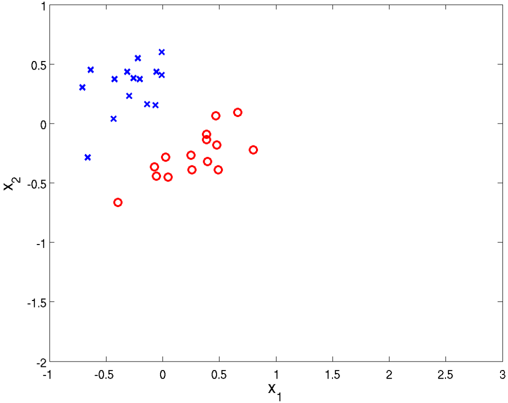
Let $y_i \in \{-1, +1\}$

Binary Classification

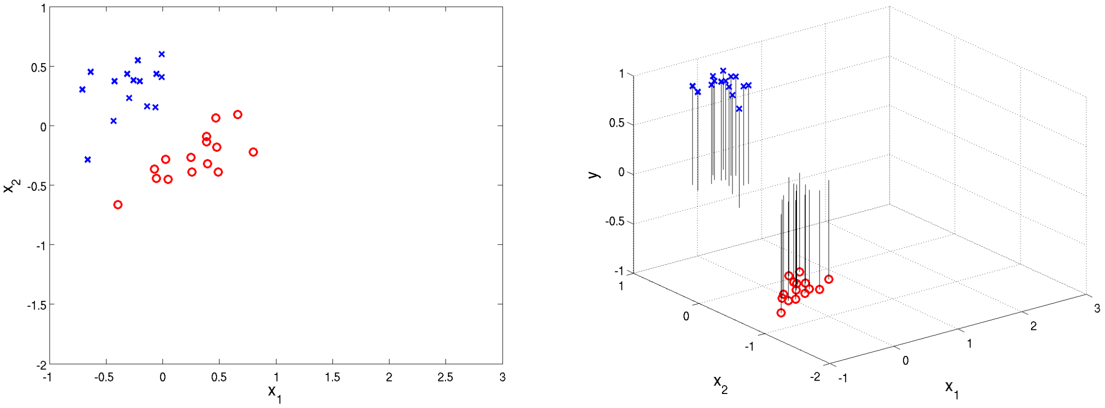
Let $y_i \in \{-1, +1\}$

Why not just use LLS?
If $f(x) > 0$, report class “+1” Otherwise, report class “-1”

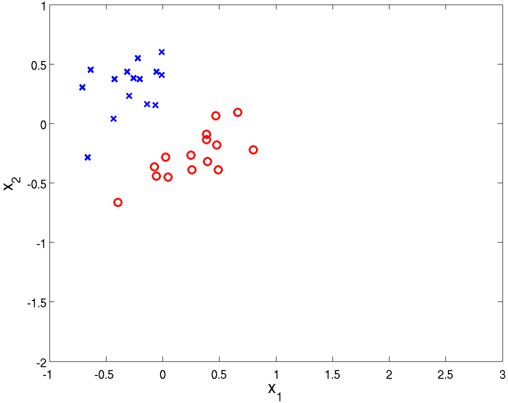
LLS for Classification

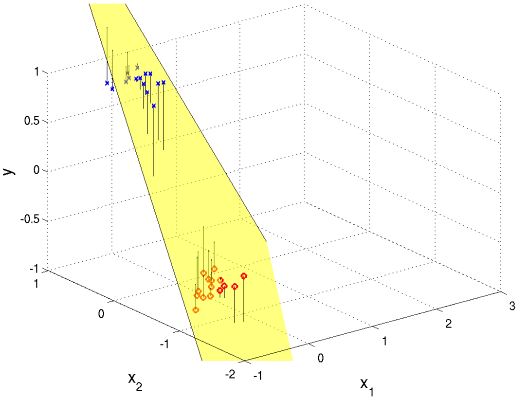


LLS for Classification

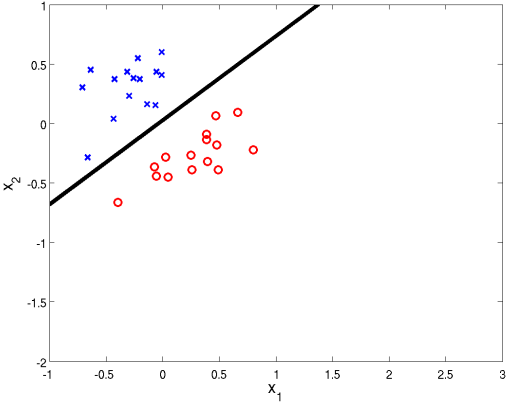


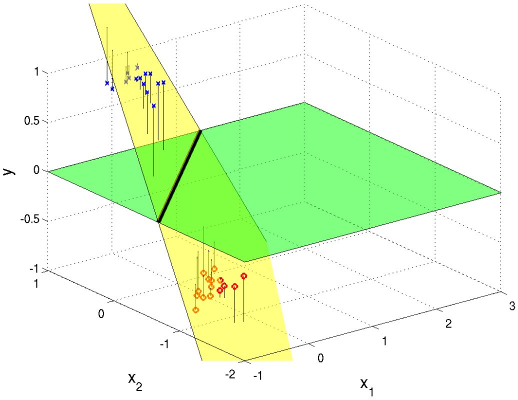
LLS for Classification



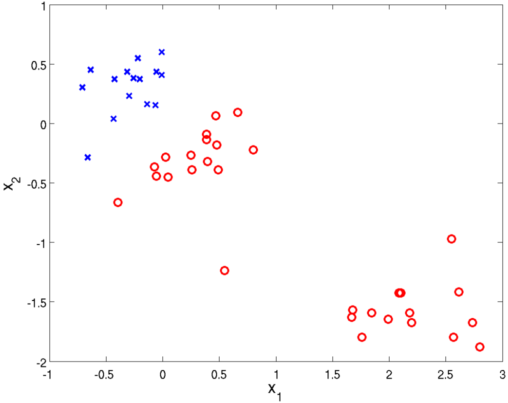


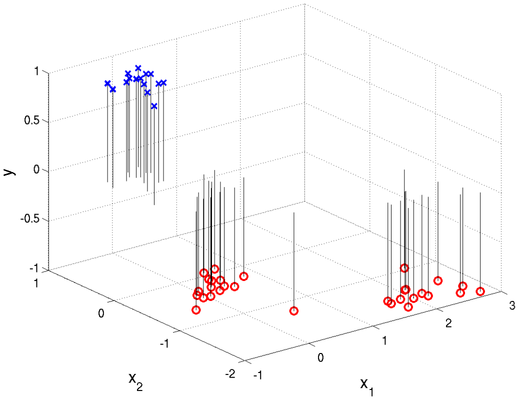
LLS for Classification



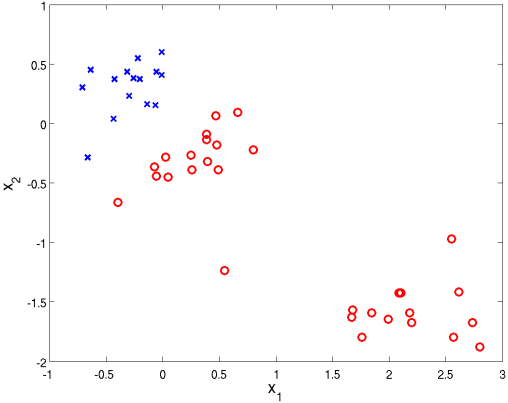


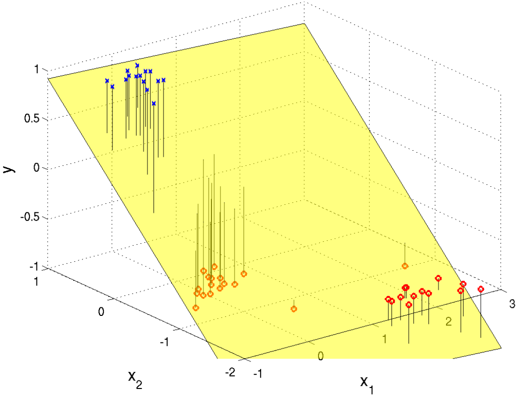
LLS for Classification



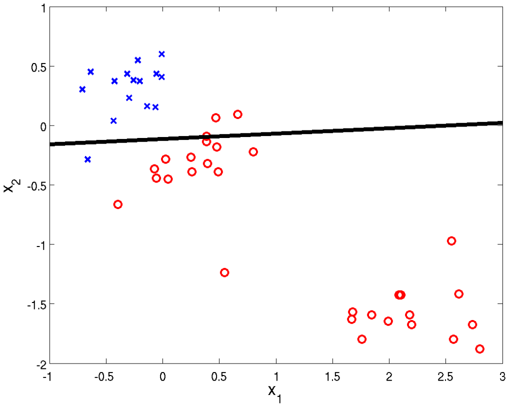


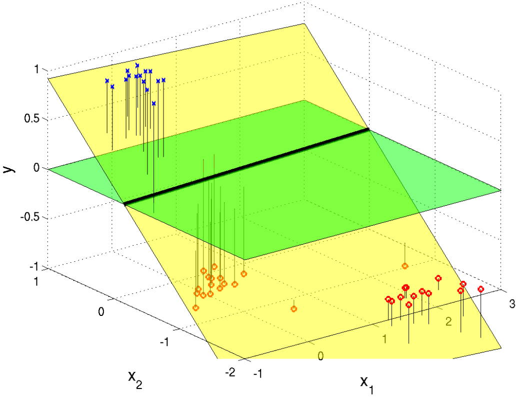
LLS for Classification



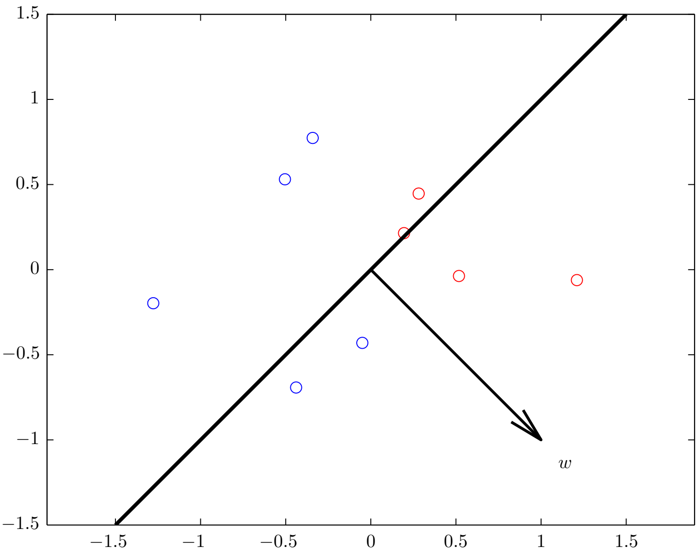


LLS for Classification



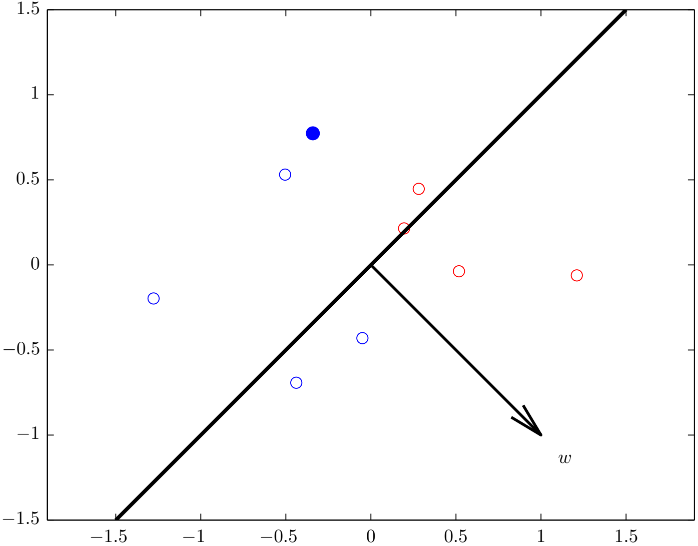


Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

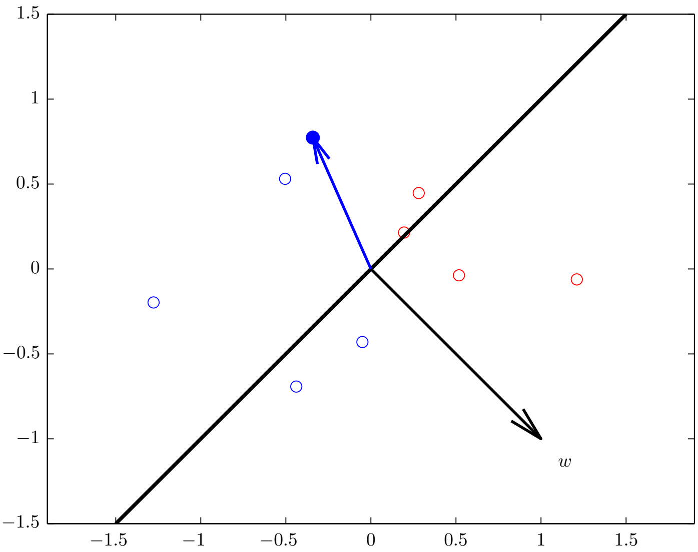


$$w = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.4 \\ 0.75 \end{bmatrix}$$

$$y = +1$$

Perceptron Learning Algorithm, Example

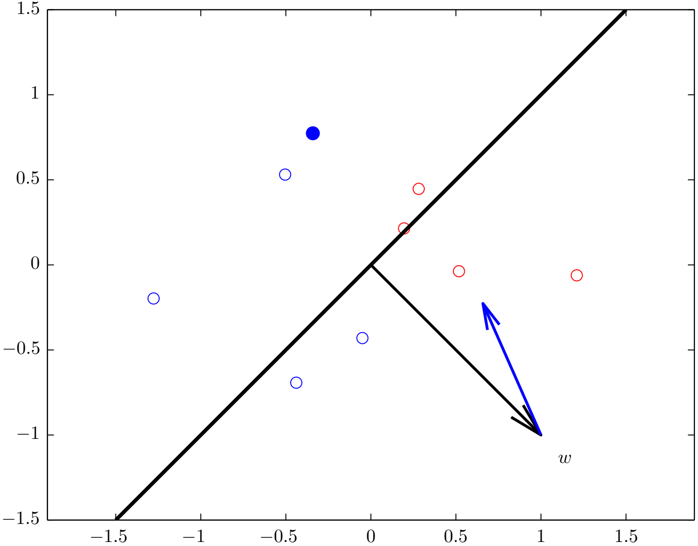


$$w = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.4 \\ 0.75 \end{bmatrix}$$

$$y = +1$$

Perceptron Learning Algorithm, Example

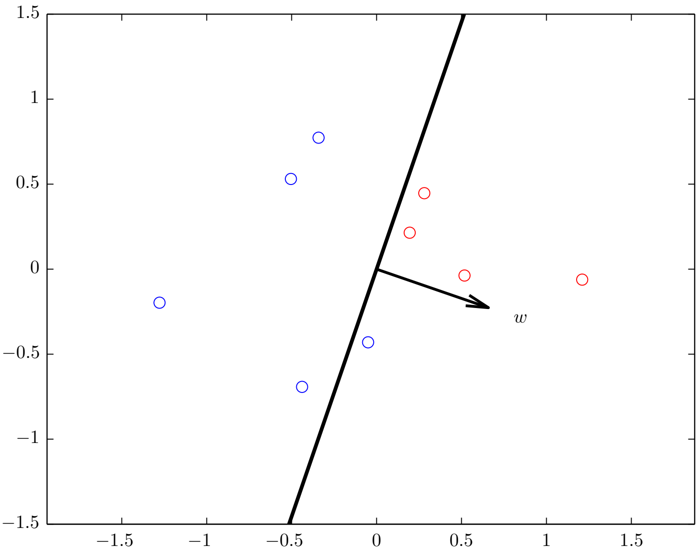


$$w = \begin{bmatrix} 1 - 0.4 \\ -1 + 0.75 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.4 \\ 0.75 \end{bmatrix}$$

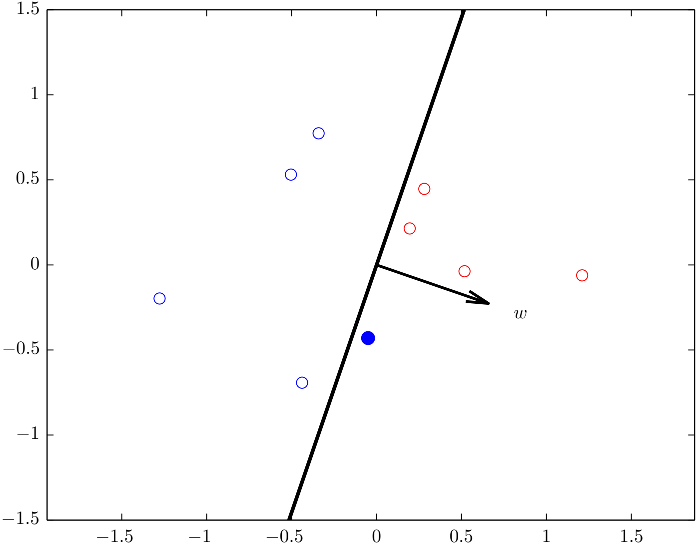
$$y = +1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} 0.6 \\ -0.25 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

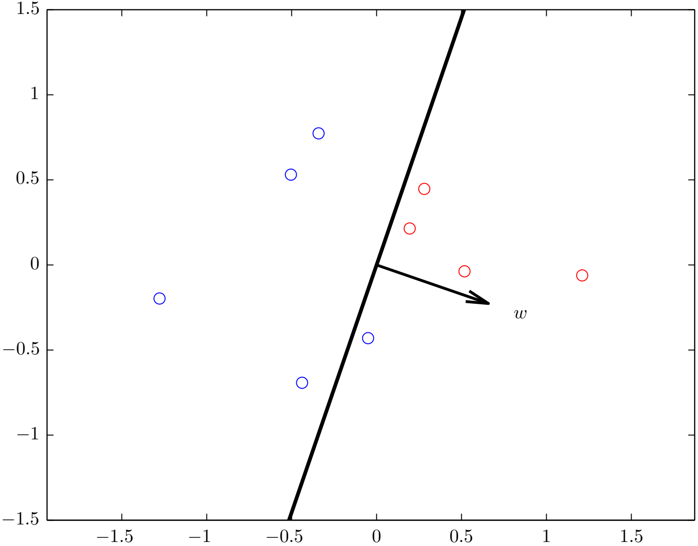


$$w = \begin{bmatrix} 0.6 \\ -0.25 \end{bmatrix}$$

$$x = \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$$

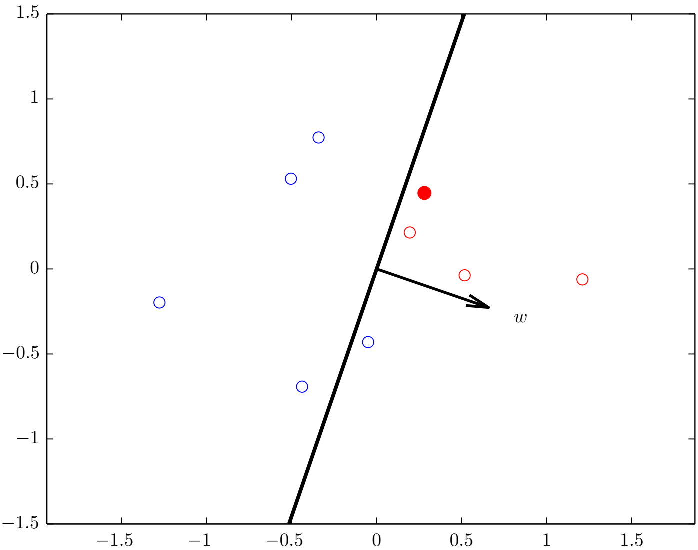
$$y = +1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} 0.6 \\ -0.25 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

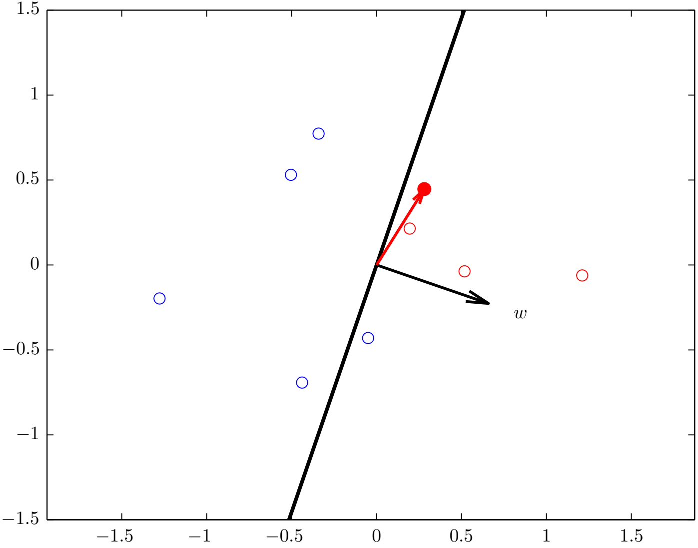


$$w = \begin{bmatrix} 0.6 \\ -0.25 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.2 \\ 0.45 \end{bmatrix}$$

$$y = -1$$

Perceptron Learning Algorithm, Example

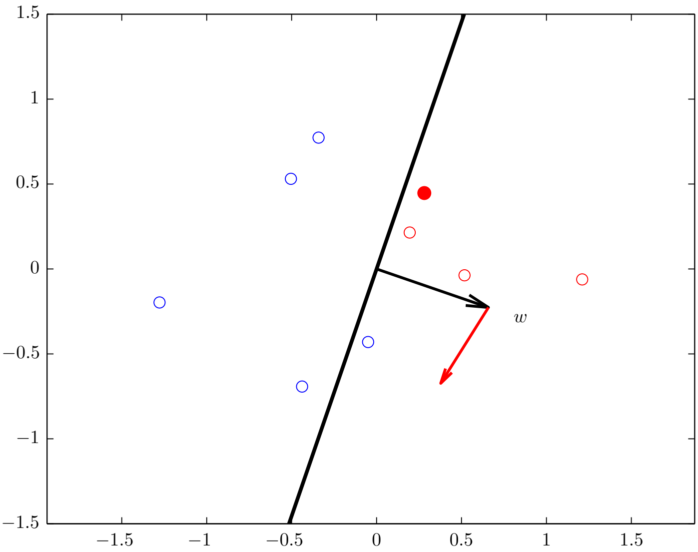


$$w = \begin{bmatrix} 0.6 \\ -0.25 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.2 \\ 0.45 \end{bmatrix}$$

$$y = -1$$

Perceptron Learning Algorithm, Example

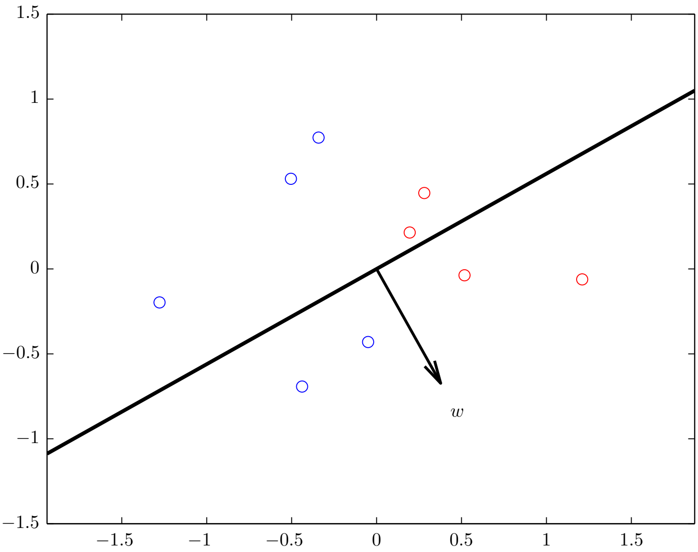


$$w = \begin{bmatrix} 0.6 - 0.2 \\ -0.25 - 0.45 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.2 \\ 0.45 \end{bmatrix}$$

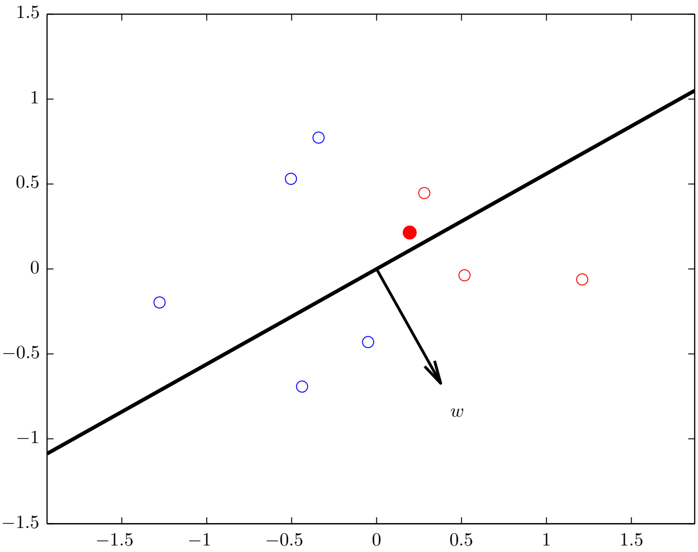
$$y = -1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

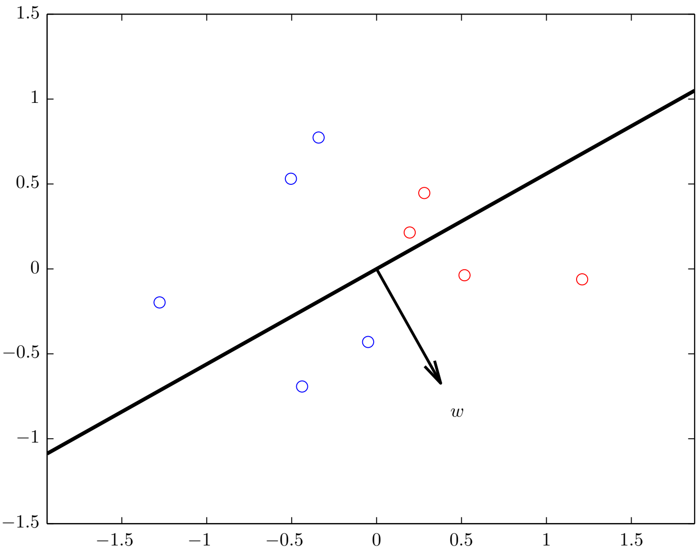


$$w = \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

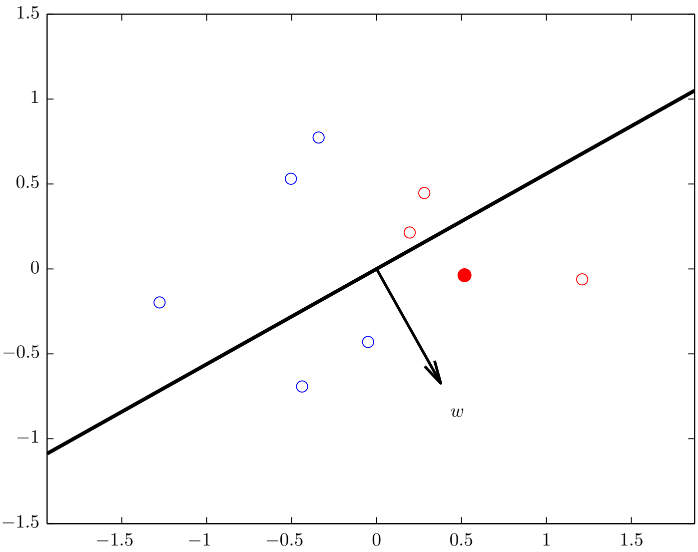
$$y = -1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

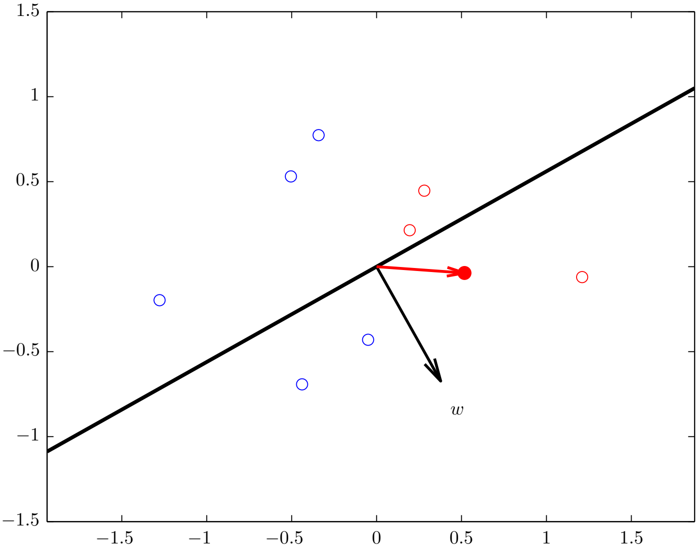


$$w = \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.5 \\ -0.1 \end{bmatrix}$$

$$y = -1$$

Perceptron Learning Algorithm, Example

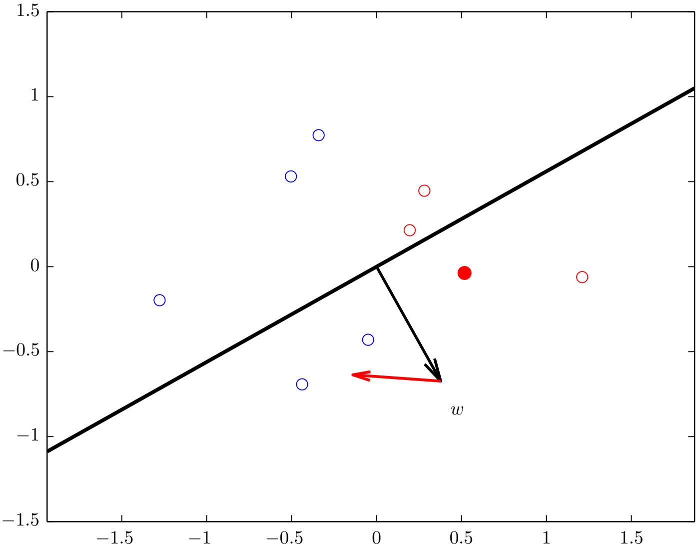


$$w = \begin{bmatrix} 0.4 \\ -0.7 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.5 \\ -0.1 \end{bmatrix}$$

$$y = -1$$

Perceptron Learning Algorithm, Example

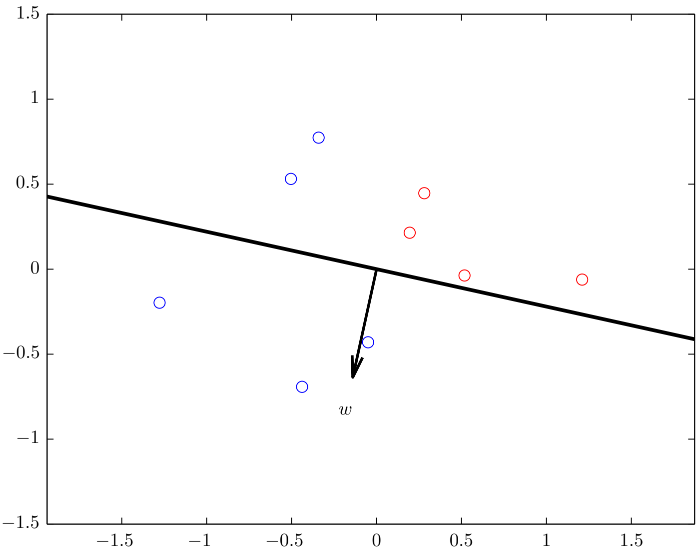


$$w = \begin{bmatrix} 0.4 - 0.5 \\ -0.7 + 0.1 \end{bmatrix}$$

$$x = \begin{bmatrix} 0.5 \\ -0.1 \end{bmatrix}$$

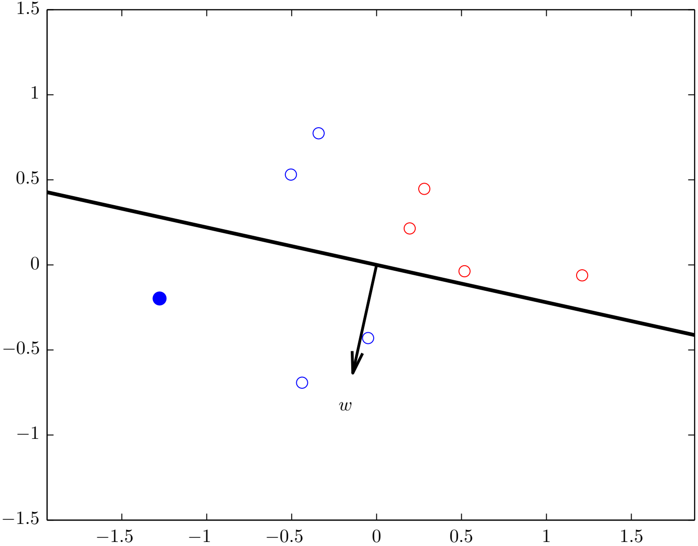
$$y = -1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

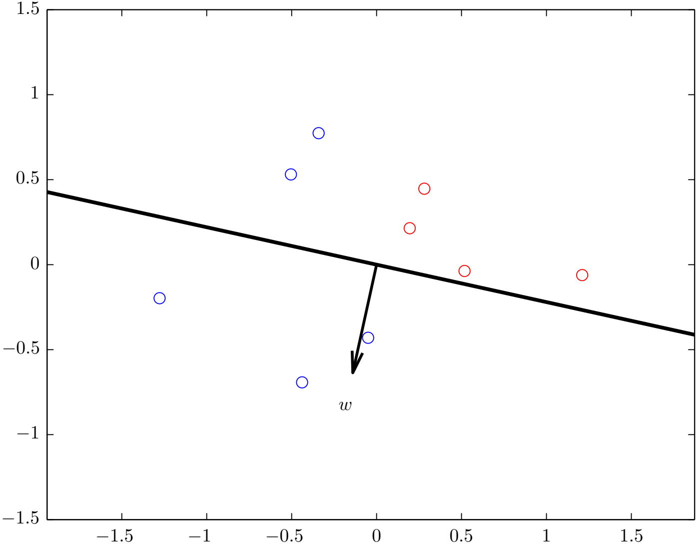


$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

$$x = \begin{bmatrix} -1.25 \\ -0.25 \end{bmatrix}$$

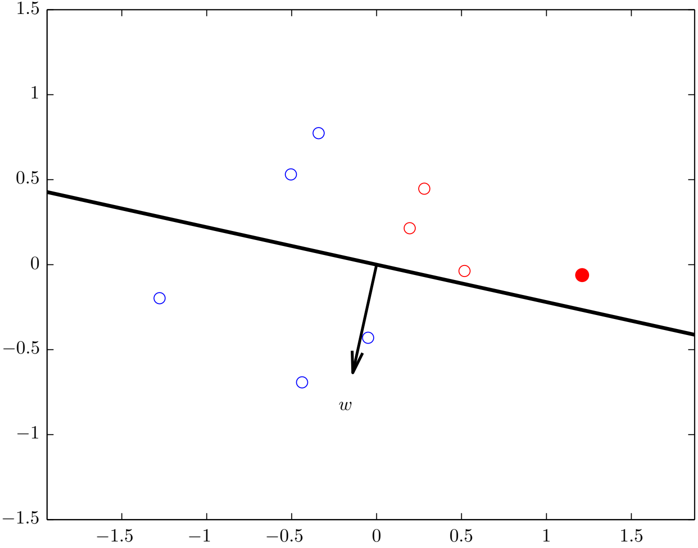
$$y = +1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

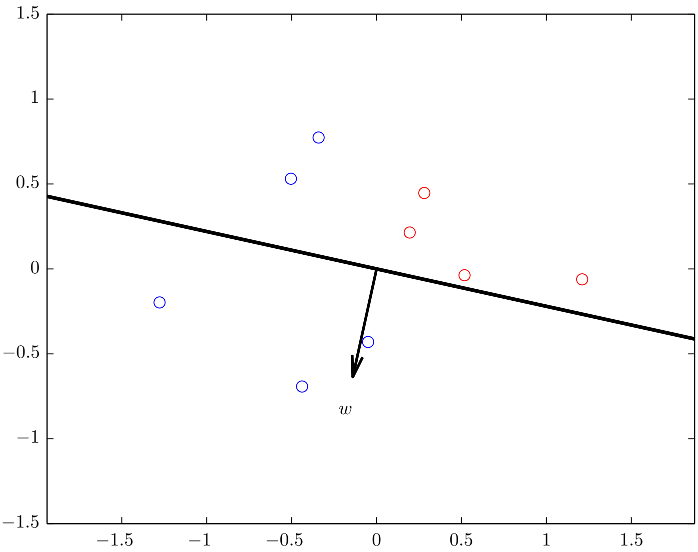


$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

$$x = \begin{bmatrix} 1.25 \\ 0 \end{bmatrix}$$

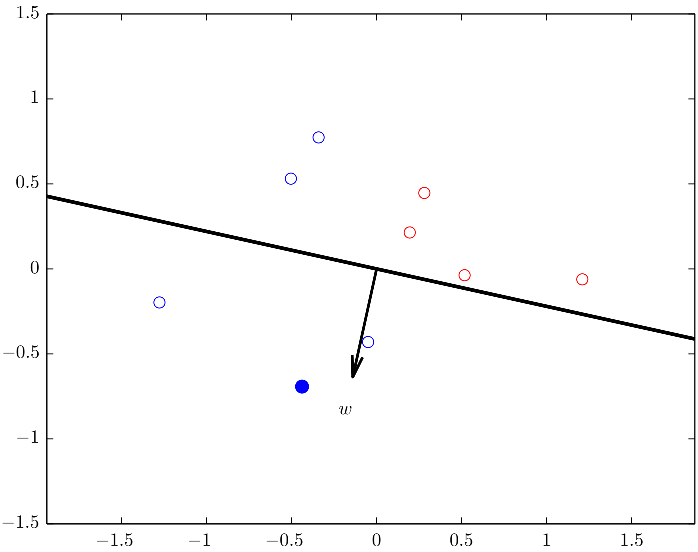
$$y = -1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

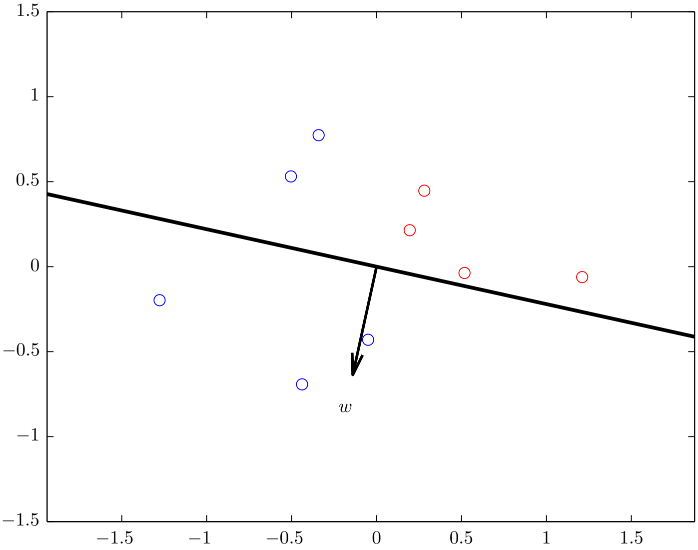


$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.2 \\ -0.6 \end{bmatrix}$$

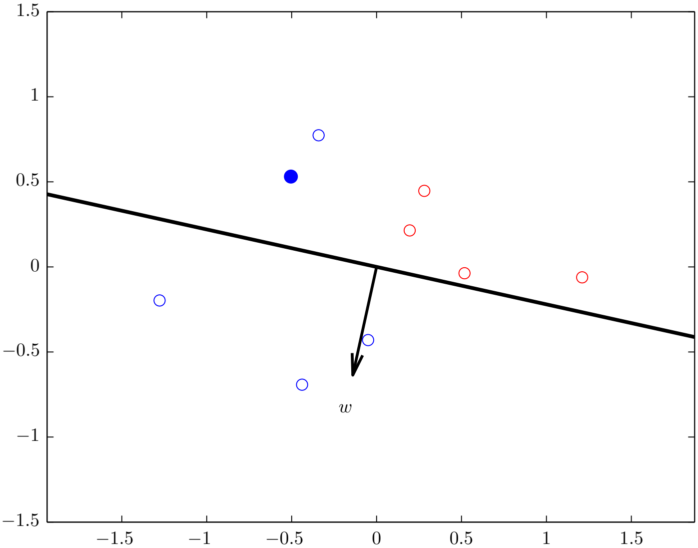
$$y = +1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

Perceptron Learning Algorithm, Example

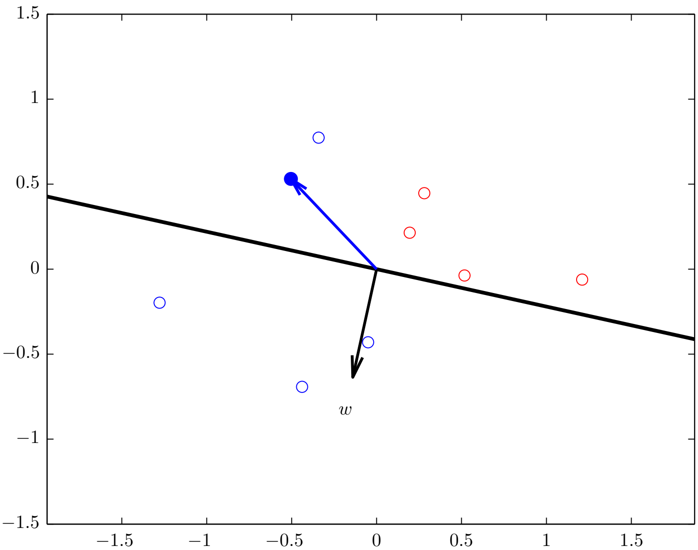


$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}$$

$$y = +1$$

Perceptron Learning Algorithm, Example

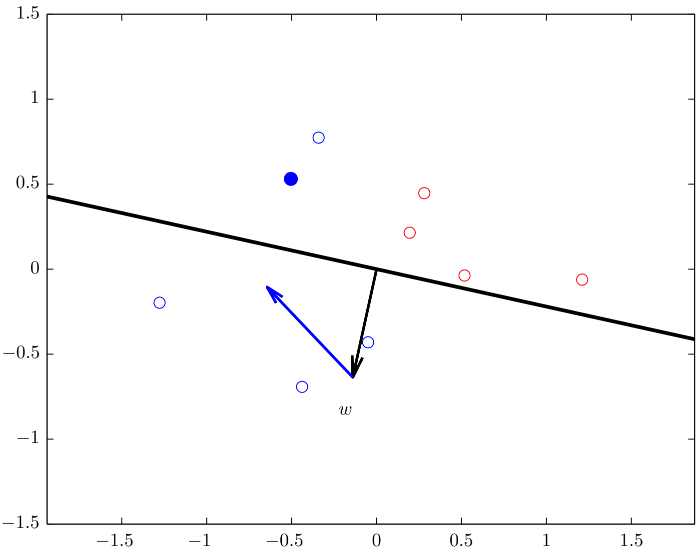


$$w = \begin{bmatrix} -0.1 \\ -0.6 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}$$

$$y = +1$$

Perceptron Learning Algorithm, Example

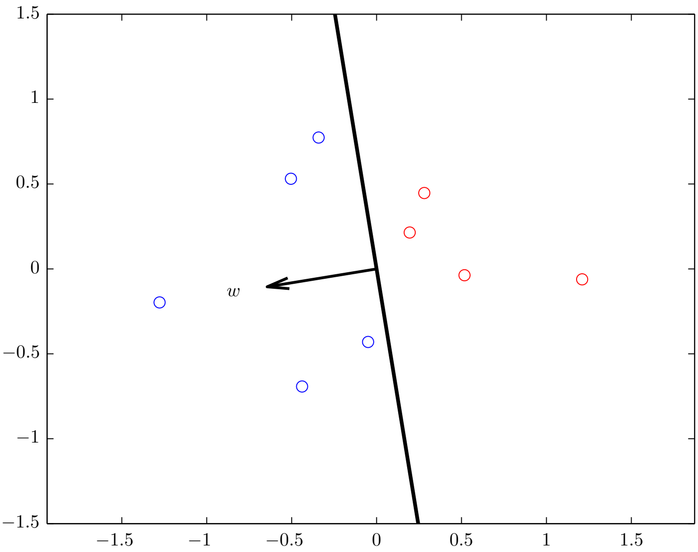


$$w = \begin{bmatrix} -0.1 - 0.5 \\ -0.6 + 0.5 \end{bmatrix}$$

$$x = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix}$$

$$y = +1$$

Perceptron Learning Algorithm, Example



$$w = \begin{bmatrix} -0.6 \\ -0.1 \end{bmatrix}$$

Perceptron Learning Algorithm

1. Let w be a random vector
2. Let η be a positive scalar
3. While not done
    1. For $i = 1, \ldots, m$
        1. $h_i \leftarrow w^\top x_i$
        2. If $h_i y_i < 0$
           Then $w \leftarrow w + \eta y_i x_i$

Perceptron Convergence

Perceptron Convergence Theorem:
If the data are linearly separable,
  the perceptron learning algorithm will find a separating
  hyperplane in a finite number of steps.
If the data are not linearly separable,
  it will run forever.

Perceptron Convergence

Perceptron Convergence Theorem:
If the data are linearly separable,
the perceptron learning algorithm will find a separating hyperplane in a finite number of steps.
If the data are not linearly separable,
it will run forever.

No way to tell the difference until it stops.

Perceptron as Loss Minimization

The Perceptron learning algorithm is trying to minimize

$$\begin{aligned} L &= \sum_{i=1}^{m} l\left(y_i w^\top x_i\right) \\ &= \sum_{i=1}^{m} l\left(y_i \sum_{j=1}^{n} w_j x_{i,j}\right) \end{aligned}$$

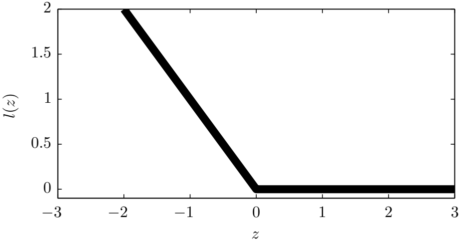
Perceptron as Loss Minimization

The Perceptron learning algorithm is trying to minimize

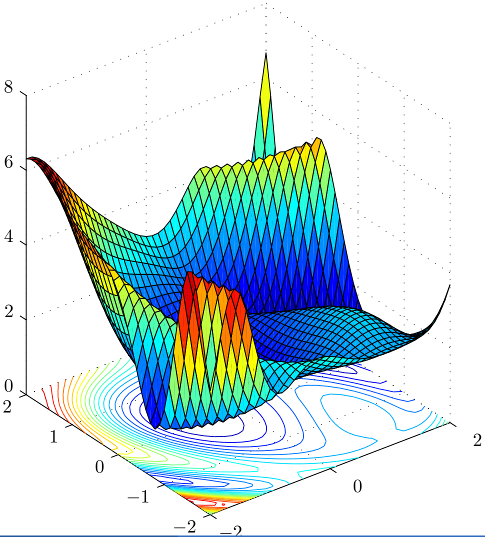
$$
\begin{aligned}
L &= \sum_{i=1}^{m} l\left(y_i w^\top x_i\right) \\
&= \sum_{i=1}^{m} l\left(y_i \sum_{j=1}^{n} w_j x_{i,j}\right)
\end{aligned}
$$

where

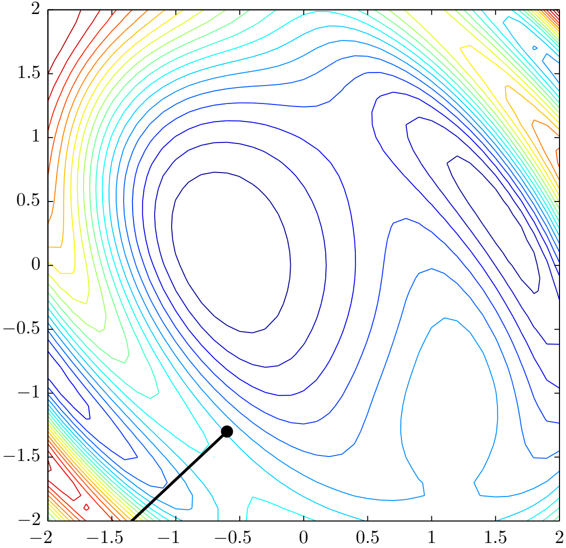
$$l(z) = \max(0, -z)$$



Gradient Descent



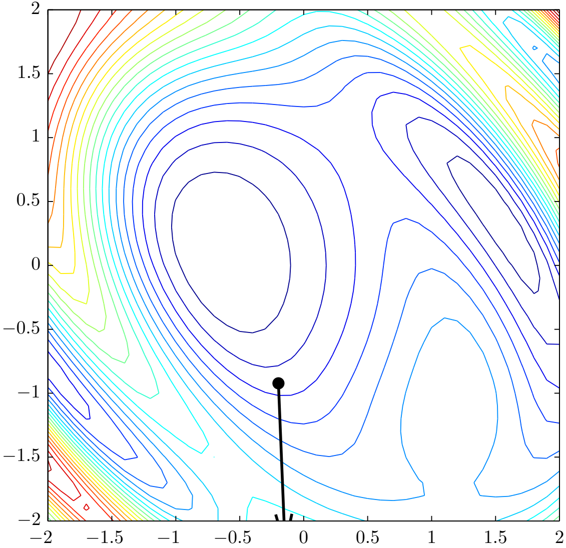
Gradient Descent



$$x = \begin{bmatrix} -0.6 \\ -1.3 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2.01 \\ -1.89 \end{bmatrix}$$

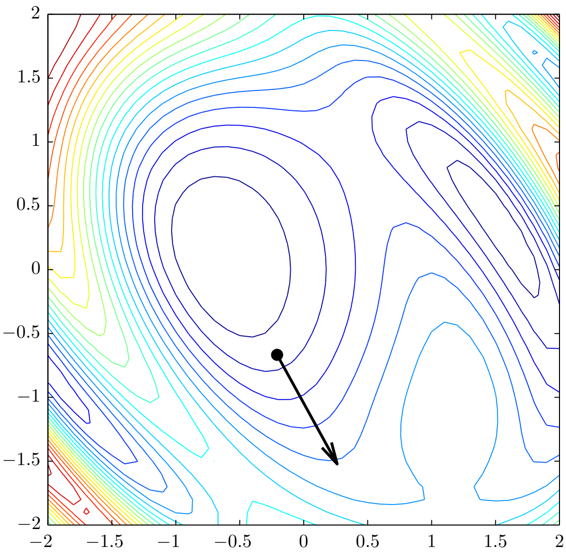
Gradient Descent



$$x = \begin{bmatrix} -0.20 \\ -0.92 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -0.05 \\ -1.28 \end{bmatrix}$$

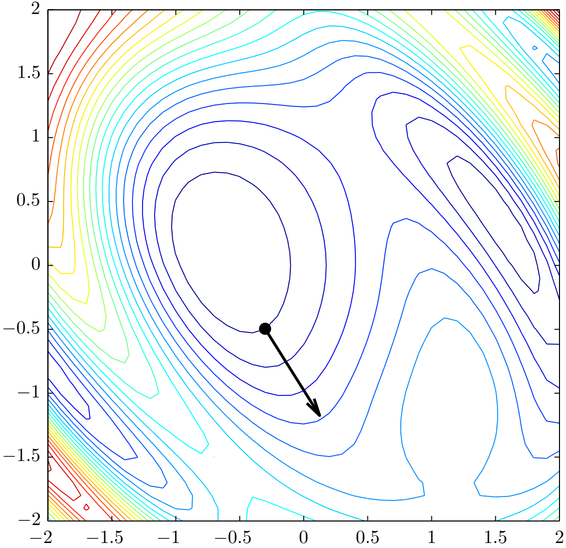
Gradient Descent



$$x = \begin{bmatrix} -0.21 \\ -0.67 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -0.21 \\ -0.67 \end{bmatrix}$$

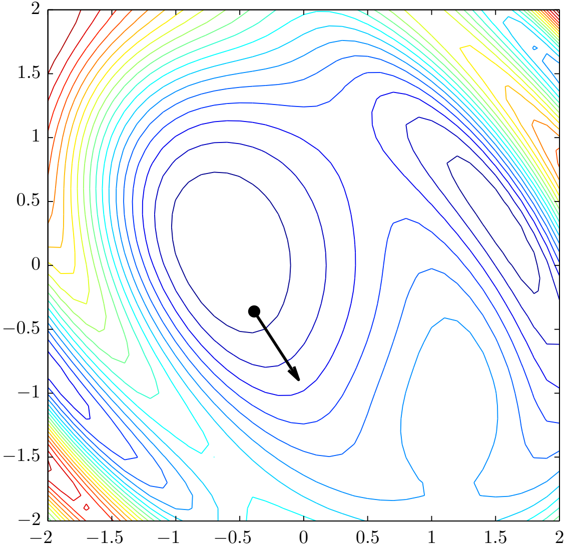
Gradient Descent



$$x = \begin{bmatrix} -0.30 \\ -0.50 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0.43 \\ -0.68 \end{bmatrix}$$

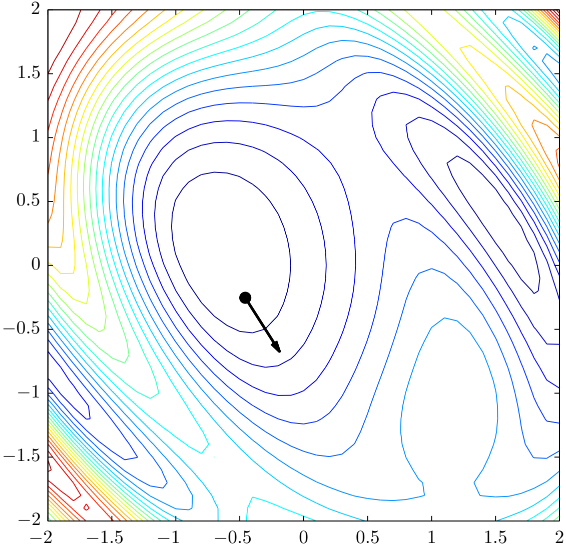
Gradient Descent



$$x = \begin{bmatrix} -0.39 \\ -0.36 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0.35 \\ -0.54 \end{bmatrix}$$

Gradient Descent



$$x = \begin{bmatrix} -0.47 \\ -0.25 \end{bmatrix}$$

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0.27 \\ -0.42 \end{bmatrix}$$

Gradient Descent Algorithm

1. Let x be a random point
2. While x is not at local minimum of f
    1. Let $g \leftarrow \nabla_x f(x)$
    2. Let $x \leftarrow x - \eta g$

Stochastic Gradient Descent

If

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Then

$$\nabla_x f(x) = \sum_{i=1}^{m} \nabla_x f_i(x)$$

Stochastic Gradient Descent

If

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Then

$$\nabla_x f(x) = \sum_{i=1}^{m} \nabla_x f_i(x)$$

1. Let x be a random point
2. While x is not at local minimum of f
   1. Let $g \leftarrow \nabla_x f(x)$
   2. Let $x \leftarrow x - \eta g$

Stochastic Gradient Descent

If

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Then

$$\nabla_x f(x) = \sum_{i=1}^{m} \nabla_x f_i(x)$$

1. Let x be a random point
2. While x is not at local minimum of f
   1. Let $g \leftarrow 0$
   2. For $i = 1, \ldots, m$
      1. Let $g_i \leftarrow \nabla_x f_i(x)$
      2. Let $g \leftarrow g + g_i$
   3. Let $x \leftarrow x - \eta g$

Stochastic Gradient Descent

If

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Then

$$\nabla_x f(x) = \sum_{i=1}^{m} \nabla_x f_i(x)$$

1. Let x be a random point
2. While x is not at local minimum of f
    1. For $i = 1, \ldots, m$
        1. Let $g_i \leftarrow \nabla_x f_i(x)$
        2. Let $x \leftarrow x - \eta g_i$

Stochastic Gradient Descent

If

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Then

$$\nabla_x f(x) = \sum_{i=1}^{m} \nabla_x f_i(x)$$

1. Let x be a random point
2. While x is not at local minimum of f
   1. For $i = 1, \ldots, m$
      1. Let $g_i \leftarrow \nabla_x f_i(x)$
      2. Let $x \leftarrow x - \eta g_i$

- Only applicable when f has this sum form.
- Instead of waiting to update x until the gradient has been completely summed, update after each component is computed
- η must be much smaller
- Tends to bounce around more

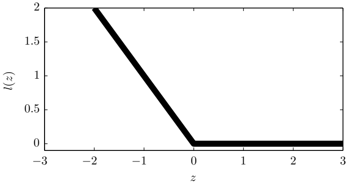
The Perceptron and Stochastic Gradient Descent

For the perceptron, we want to minimize (over w)

$$L = \sum_{i=1}^{m} \underbrace{l\left(y_i \sum_{j=1}^{n} w_j x_{i,j}\right)}_{l_i}$$

where

$$l(z) = \max(0, -z)$$



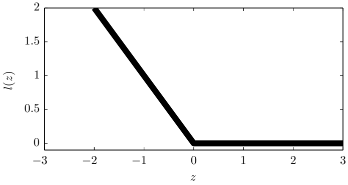
The Perceptron and Stochastic Gradient Descent

For the perceptron, we want to minimize (over w)

$$L = \sum_{i=1}^{m} \underbrace{l\left(y_i \sum_{j=1}^{n} w_j x_{i,j}\right)}_{l_i}$$

where

$$l(z) = \max(0, -z)$$



With stochastic gradient descent, consider each point in turn. Given point i, we calculate

$$-\nabla_w l_i = \begin{bmatrix} \frac{\partial l_i}{\partial w_1} \\ \frac{\partial l_i}{\partial w_2} \\ \vdots \\ \frac{\partial l_i}{\partial w_n} \end{bmatrix}$$

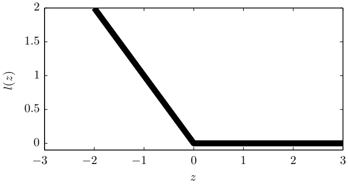
The Perceptron and Stochastic Gradient Descent

For the perceptron, we want to minimize (over w)

$$L = \sum_{i=1}^{m} \underbrace{l\left(y_i \sum_{j=1}^{n} w_j x_{i,j} \right)}_{l_i}$$

where

$$l(z) = \max(0, -z)$$



With stochastic gradient descent, consider each point in turn. Given point i, we calculate

$$-\nabla_w l_i = \begin{bmatrix} \frac{\partial l_i}{\partial w_1} \\ \frac{\partial l_i}{\partial w_2} \\ \vdots \\ \frac{\partial l_i}{\partial w_n} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \text{if } y_i w^\top x_i \geq 0 \\ \begin{bmatrix} y_i x_{i,1} \\ y_i x_{i,2} \\ \vdots \\ y_i x_{i,n} \end{bmatrix} & \text{if } y_i w^\top x_i < 0 \end{cases}$$

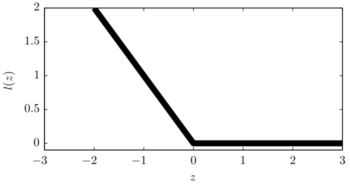
The Perceptron and Stochastic Gradient Descent

For the perceptron, we want to minimize (over w)

$$L = \sum_{i=1}^{m} \underbrace{l\left(y_i \sum_{j=1}^{n} w_j x_{i,j}\right)}_{l_i}$$

where

$$l(z) = \max(0, -z)$$



With stochastic gradient descent, consider each point in turn. Given point i, we calculate

$$-\nabla_w l_i = \begin{bmatrix} \frac{\partial l_i}{\partial w_1} \\ \frac{\partial l_i}{\partial w_2} \\ \vdots \\ \frac{\partial l_i}{\partial w_n} \end{bmatrix} = \begin{cases} 0 & \text{if } y_i w^\top x_i \geq 0 \\ y_i x_i & \text{if } y_i w^\top x_i < 0 \end{cases}$$

The Perceptron and Stochastic Gradient Descent

Stochastic Gradient Descent Algorithm

1. Let x be a random point
2. While x is not at local minimum of f
    1. For $i = 1, \dots, m$
        1. Let $g_i \leftarrow \nabla_x f_i(x)$
        2. Let $x \leftarrow x - \eta g_i$

Perceptron Learning Algorithm

1. Let w be a random vector
2. While not done
    1. For $i = 1, \dots, m$
        1. $h_i \leftarrow w^\top x_i$
        2. If $h_i y_i < 0$
           Then $w \leftarrow w + \eta y_i x_i$

For perceptrons (but not in general), η does not matter, so it is set to 1.

Gradient of just point i:

$$-\nabla_w l_i = \begin{cases} 0 & \text{if } y_i w^\top x_i \geq 0 \\ y_i x_i & \text{if } y_i w^\top x_i < 0 \end{cases}$$