

Perceptrons

Let \mathbf{x} be the inputs and \mathbf{w} be the weights. For mathematical convenience, let $x_{n+1} = 1$, so that w_{n+1} becomes the bias weight. The weighted sum u is a dot product:

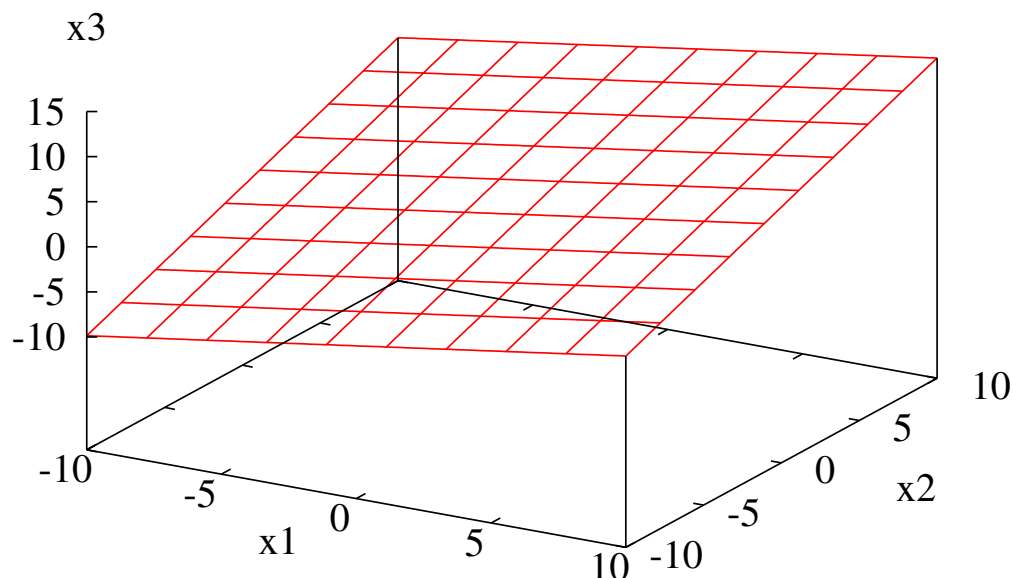
$$u = \mathbf{w} \cdot \mathbf{x} = \sum_i w_i x_i$$

Let sign be the activation function:

$$\text{sign}(u) = \begin{cases} +1 & \text{if } u > 0 \\ -1 & \text{if } u \leq 0 \end{cases}$$

The decision boundary is the hyperplane $u = 0$.

Decision boundary for $-3x_1 - 5x_2 + 7x_3 - 11$



Perceptron Learning Rule

Let η be the learning rate. The learning rule is:

$$\mathbf{if } d \neq 0 \mathbf{ then } \mathbf{w} \leftarrow \mathbf{w} + d\eta\mathbf{x}$$

Applying the learning rule to each example in a dataset is called an *epoch*. It is typical to run hundreds or thousands of epochs.

The perceptron converges to zero training error if possible.

With a slightly different activation function, the perceptron minimizes a modified L_1 error.

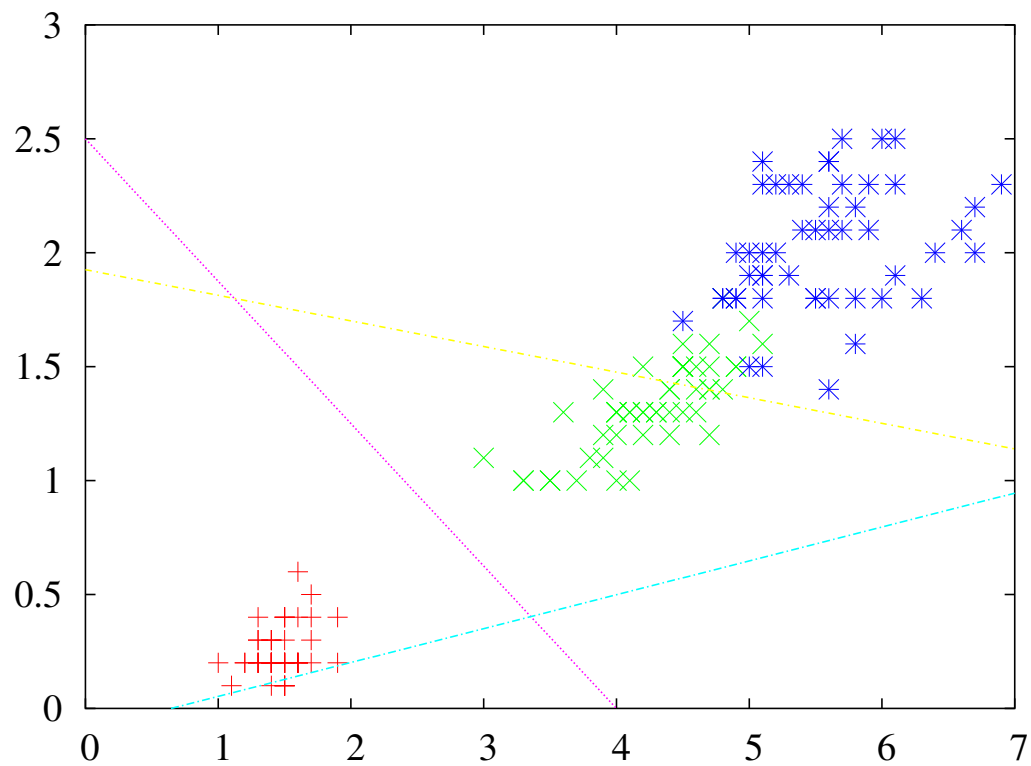
Perceptron Example ($\eta = 1$)

x_1	x_2	x_3	d	u	w_1	w_2	w_3	$w_4 = b$
					0	0	0	0
1	1	1	1	0	1	1	1	1
1	1	-1	-1	2	0	0	2	0
1	-1	1	1	2	0	0	2	0
1	-1	-1	-1	-2	0	0	2	0
-1	1	1	1	2	0	0	2	0
-1	1	-1	-1	-2	0	0	2	0
-1	-1	1	1	2	0	0	2	0
-1	-1	-1	1	-2	-1	-1	1	1

3 additional epochs are needed for convergence.

Epoch	w_1	w_2	w_3	$w_4 = b$	mistakes
1	-1	-1	1	1	1
2	-1	-1	1	3	2
3	-2	-2	2	2	1
4	-1	-1	3	3	0

1000 epochs of perceptron on iris data



Perceptron Convergence Theorem

Theorem: If some weight vector \mathbf{w}^* classifies all examples (\mathbf{x}, d) correctly, then the perceptron learning rule converges to zero training error.

Let $m = \min\{d\mathbf{x} \cdot \mathbf{w}^*\}$ over all examples.

Let $M = \max\{\mathbf{x} \cdot \mathbf{x}\}$ over all examples.

Let \mathbf{w}_0 be the zero weight vector.

Let (\mathbf{x}_i, d_i) be an example missed by \mathbf{w}_i .

Let $\mathbf{w}_{i+1} = \mathbf{w}_i + d_i\mathbf{x}_i$, i.e., learning rule.

This implies $\mathbf{w}_k = d_0\mathbf{x}_0 + \dots + d_{k-1}\mathbf{x}_{k-1}$

$\mathbf{w}_i \cdot d\mathbf{x}_i \leq 0$ so

$$\begin{aligned} & \mathbf{w}_{i+1} \cdot \mathbf{w}_{i+1} \\ &= (\mathbf{w}_i + d_i\mathbf{x}_i) \cdot (\mathbf{w}_i + d_i\mathbf{x}_i) \\ &= \mathbf{w}_i \cdot \mathbf{w}_i + 2\mathbf{w}_i \cdot d_i\mathbf{x}_i + d_i^2\mathbf{x}_i \cdot \mathbf{x}_i \\ &\leq \mathbf{w}_i \cdot \mathbf{w}_i + M \end{aligned}$$

Math. induction implies $\mathbf{w}_k \cdot \mathbf{w}_k \leq kM$

$d\mathbf{x}_i \cdot \mathbf{w}^* \geq m$ so

$$\begin{aligned} \mathbf{w}_k \cdot \mathbf{w}^* &= (d_0\mathbf{x}_0 + \dots + d_{k-1}\mathbf{x}_{k-1}) \cdot \mathbf{w}^* \\ &\geq km \end{aligned}$$

Because $\mathbf{w}_k \cdot \mathbf{w}^* \leq \|\mathbf{w}_k\| \cdot \|\mathbf{w}^*\|$, we can now derive

$$\begin{aligned}(km)^2 &\leq (\mathbf{w}_k \cdot \mathbf{w}^*)^2 \\ &\leq (\|\mathbf{w}_k\| \cdot \|\mathbf{w}^*\|)^2 \\ &= \|\mathbf{w}_k\|^2 \cdot \|\mathbf{w}^*\|^2 \\ &= (\mathbf{w}_k \cdot \mathbf{w}_k)(\mathbf{w}^* \cdot \mathbf{w}^*) \\ &\leq kM(\mathbf{w}^* \cdot \mathbf{w}^*)\end{aligned}$$

This implies an upper bound on k , the number of mistakes:

$$k \leq \frac{M(\mathbf{w}^* \cdot \mathbf{w}^*)}{m^2}$$

Linear Separability

The decision boundary $\mathbf{w} \cdot \mathbf{x} = 0$ is a line ($n = 2$), a plane ($n = 3$), or, in general, a hyperplane in n -dimensional space.

Using 0 for false and 1 for true, then $x_1 \vee x_2$ has the decision boundary $2x_1 + 2x_2 - 1 = 0$

$x_1 \wedge x_2$ has $2x_1 + 2x_2 - 3 = 0$

What about $x_1 \wedge (x_2 \vee x_3)$?

What about $x_1 \vee (x_2 \wedge (x_3 \vee x_4))$?

Most functions do not have linear boundaries:

$$x_1 \text{ XOR } x_2$$

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

This implies that there are a limited set of functions where perceptron can achieve zero error.

Different basis functions can make a function learnable for the perceptron. E.g., for XOR:

$$2x_1 + 2x_2 - 4x_1x_2 - 0.5 = 0$$

SVMs can take advantage of this possibility.

Graph of $2x_1 + 2x_2 - 4x_1x_2 - 0.5 = 0$

