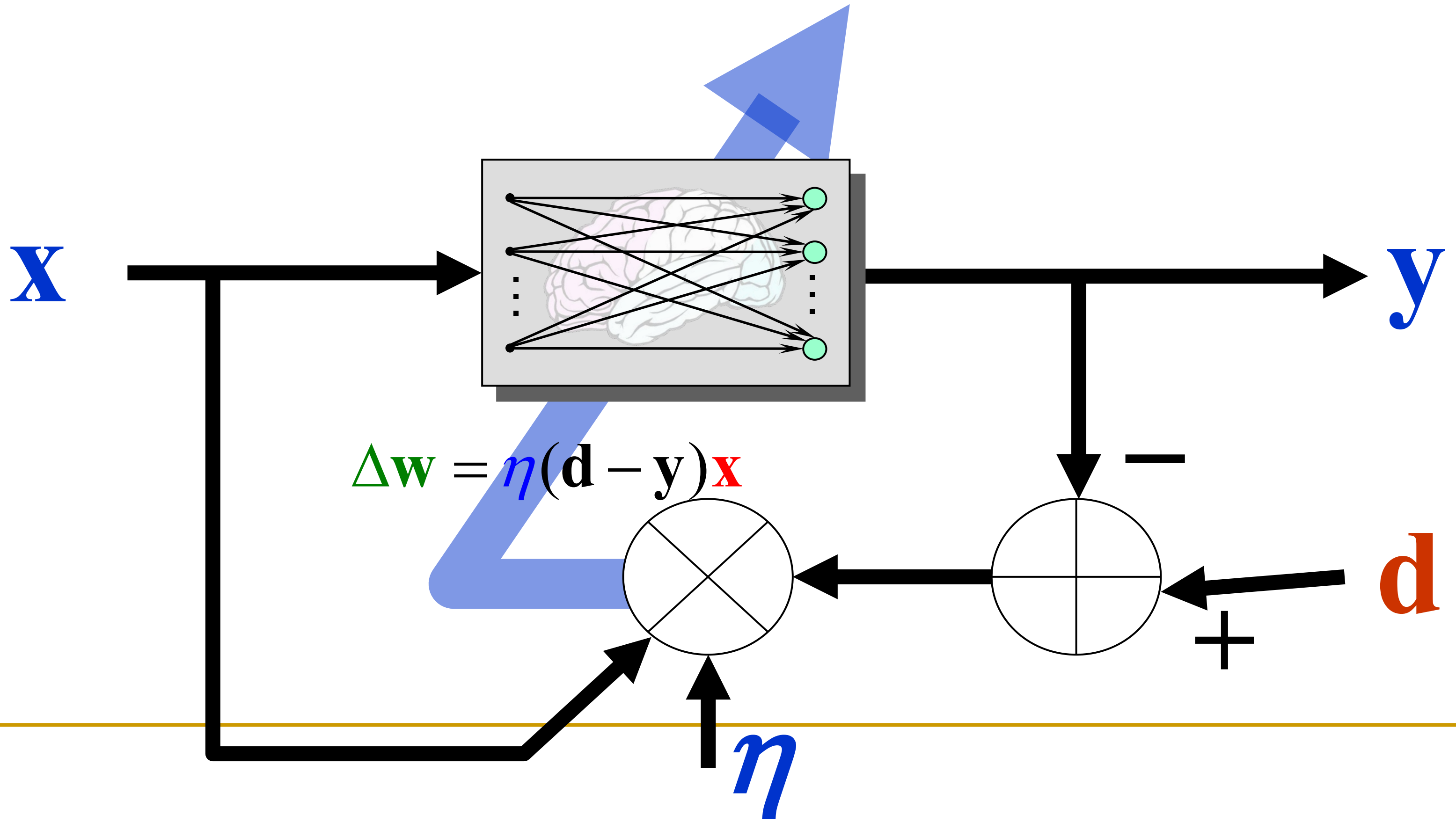
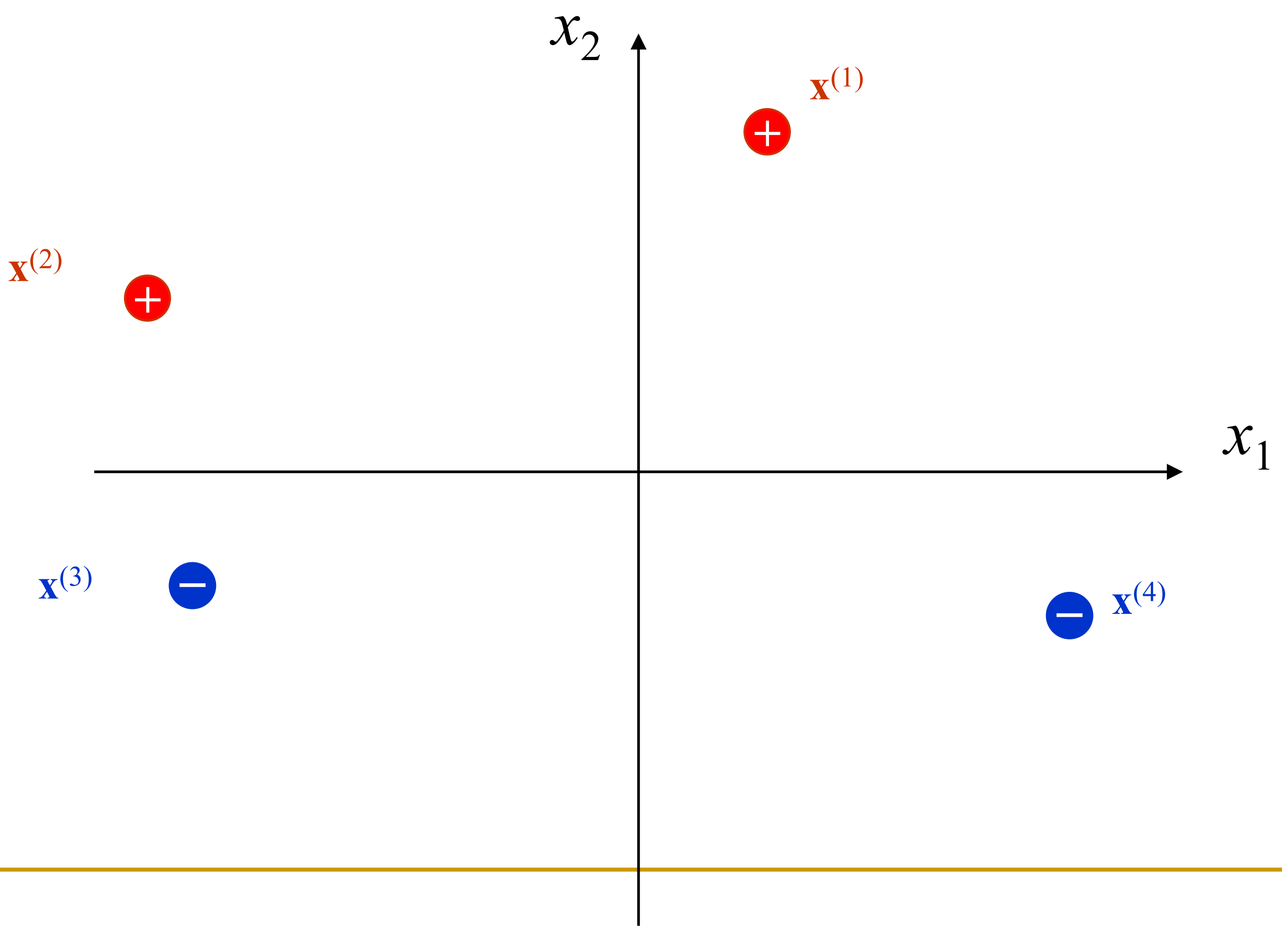
Perceptron Convergence Theorem

If the given training set is linearly separable, the learning process will converge in a finite number of steps.

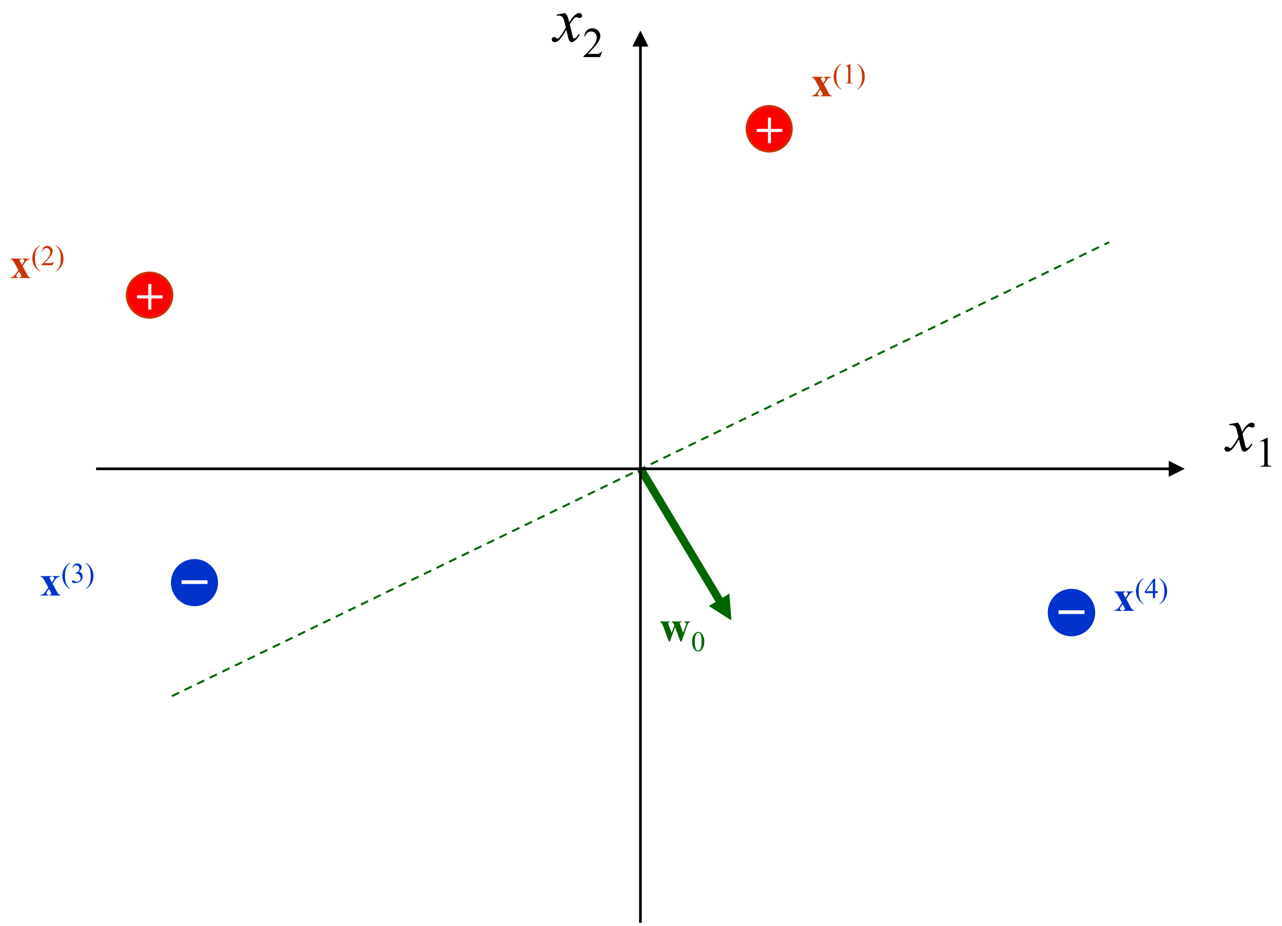


The Learning Scenario

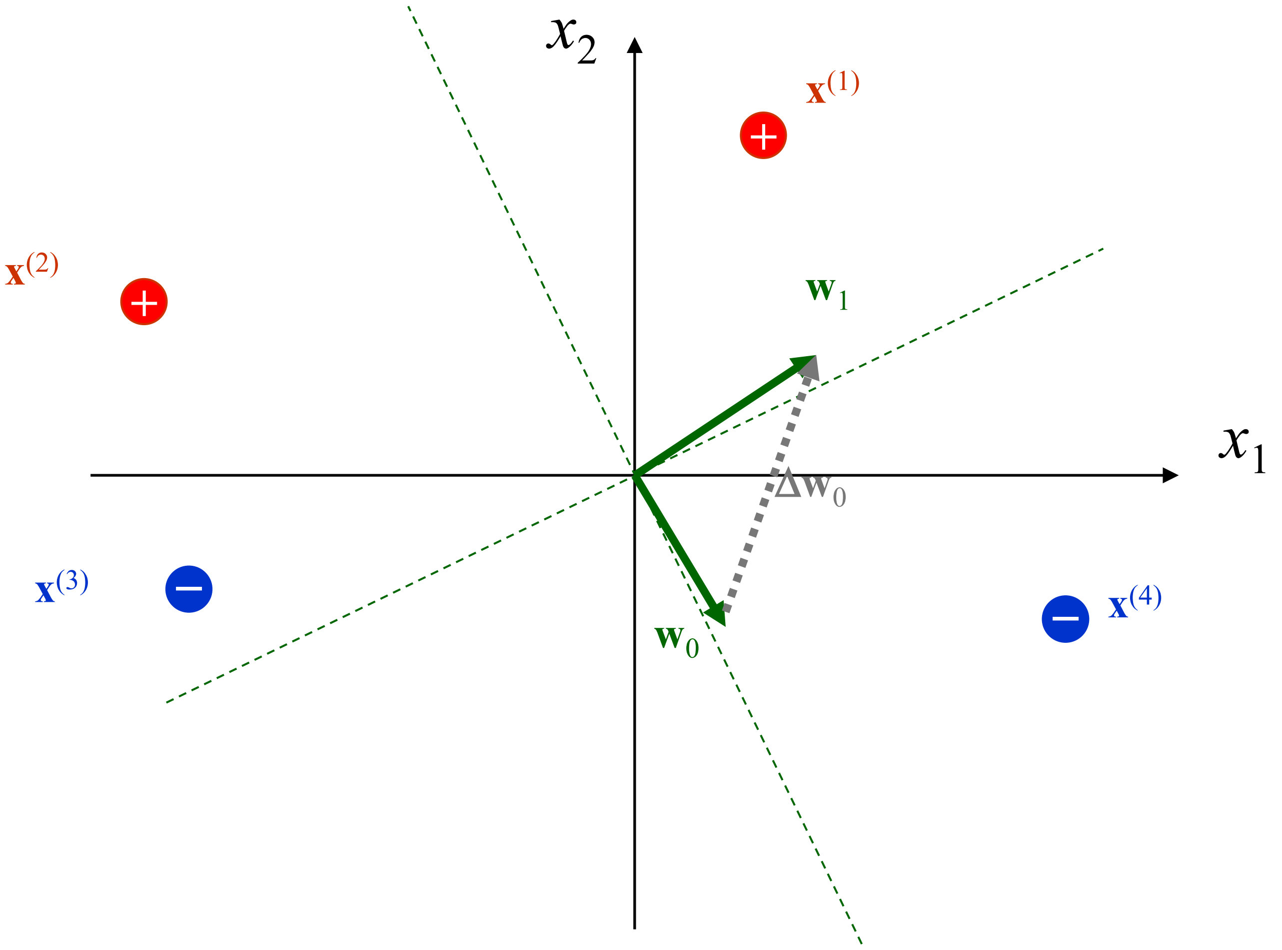
Linearly Separable.



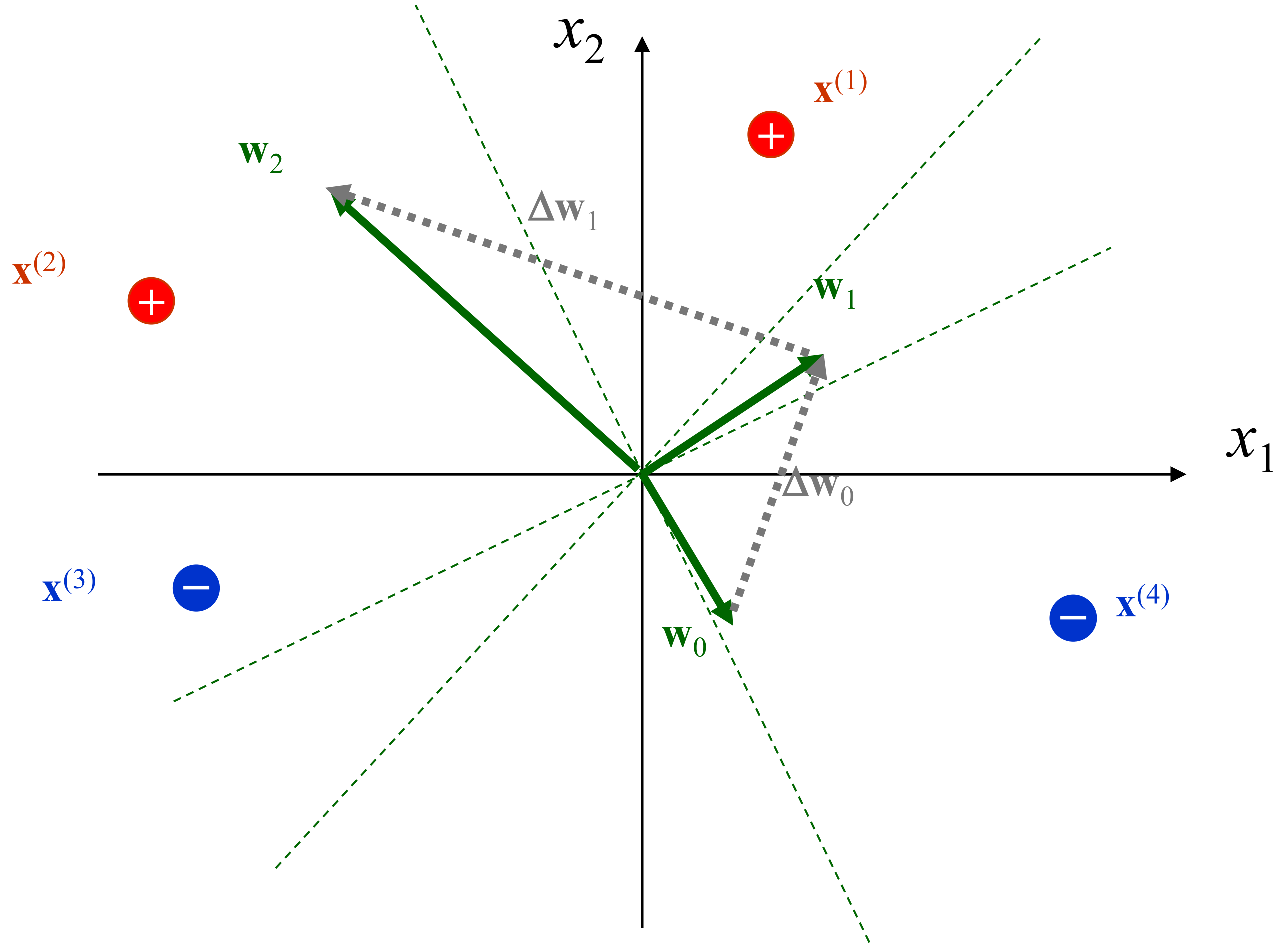
The Learning Scenario



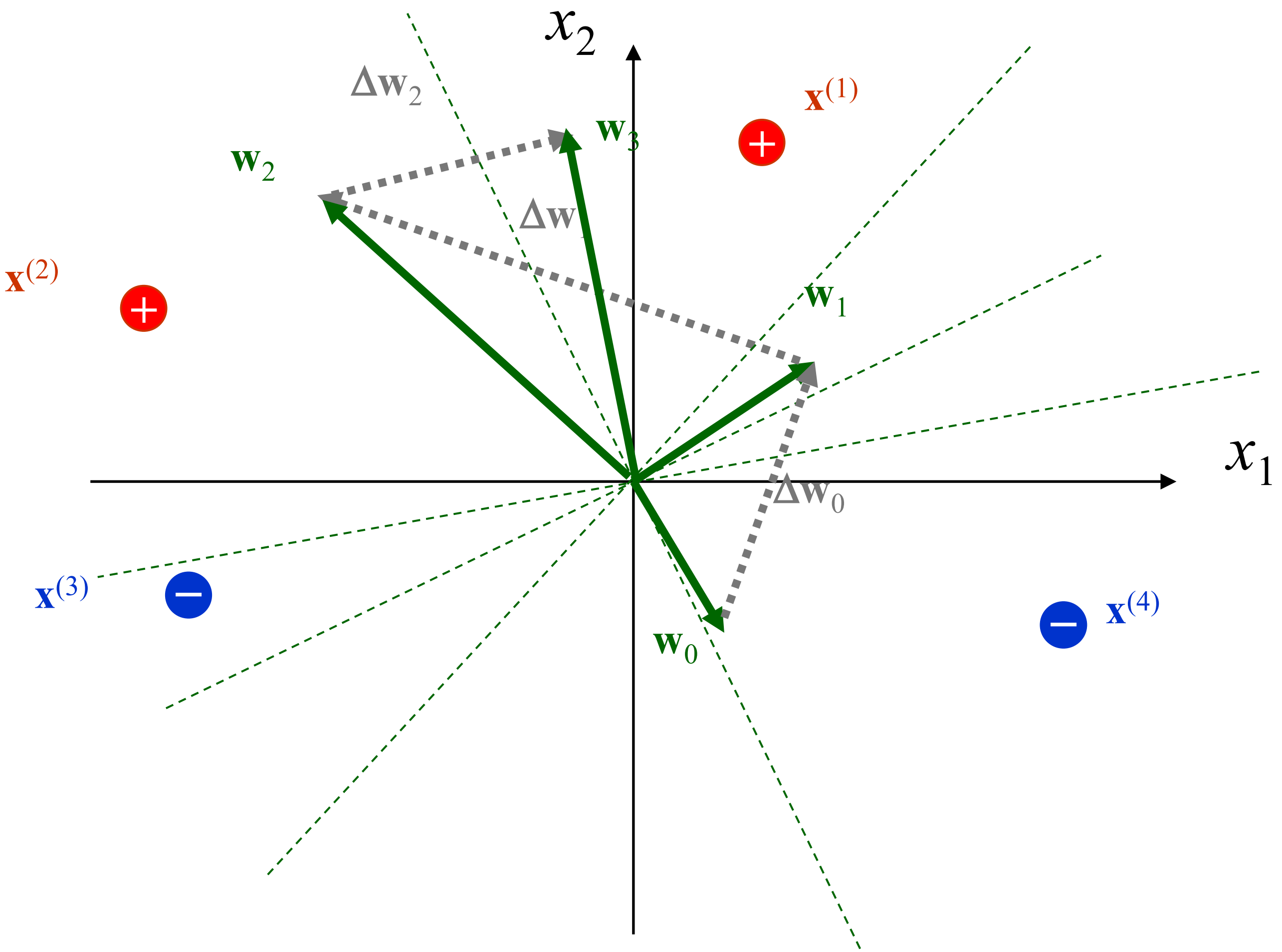
The Learning Scenario



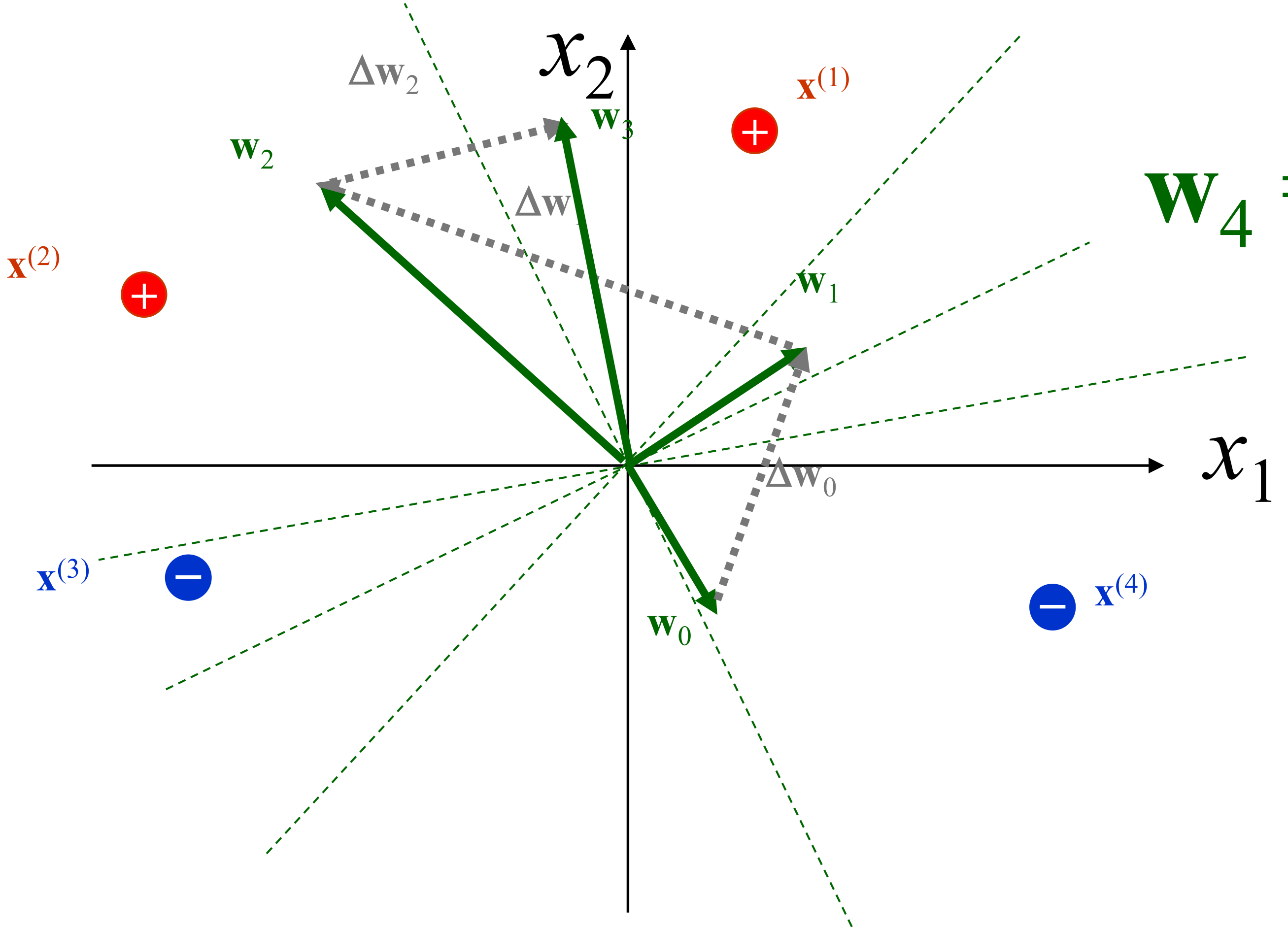
The Learning Scenario



The Learning Scenario

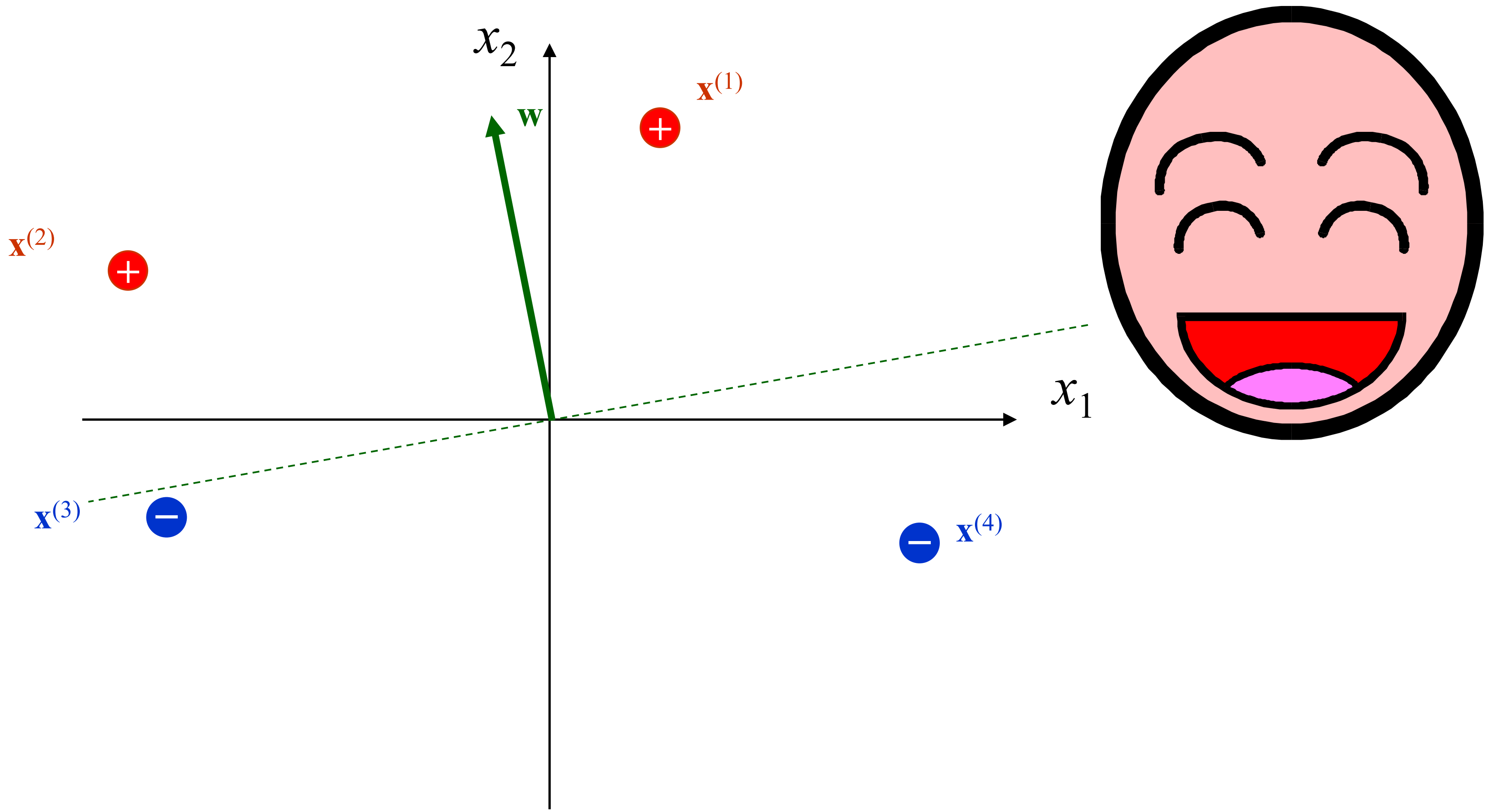


The Learning Scenario



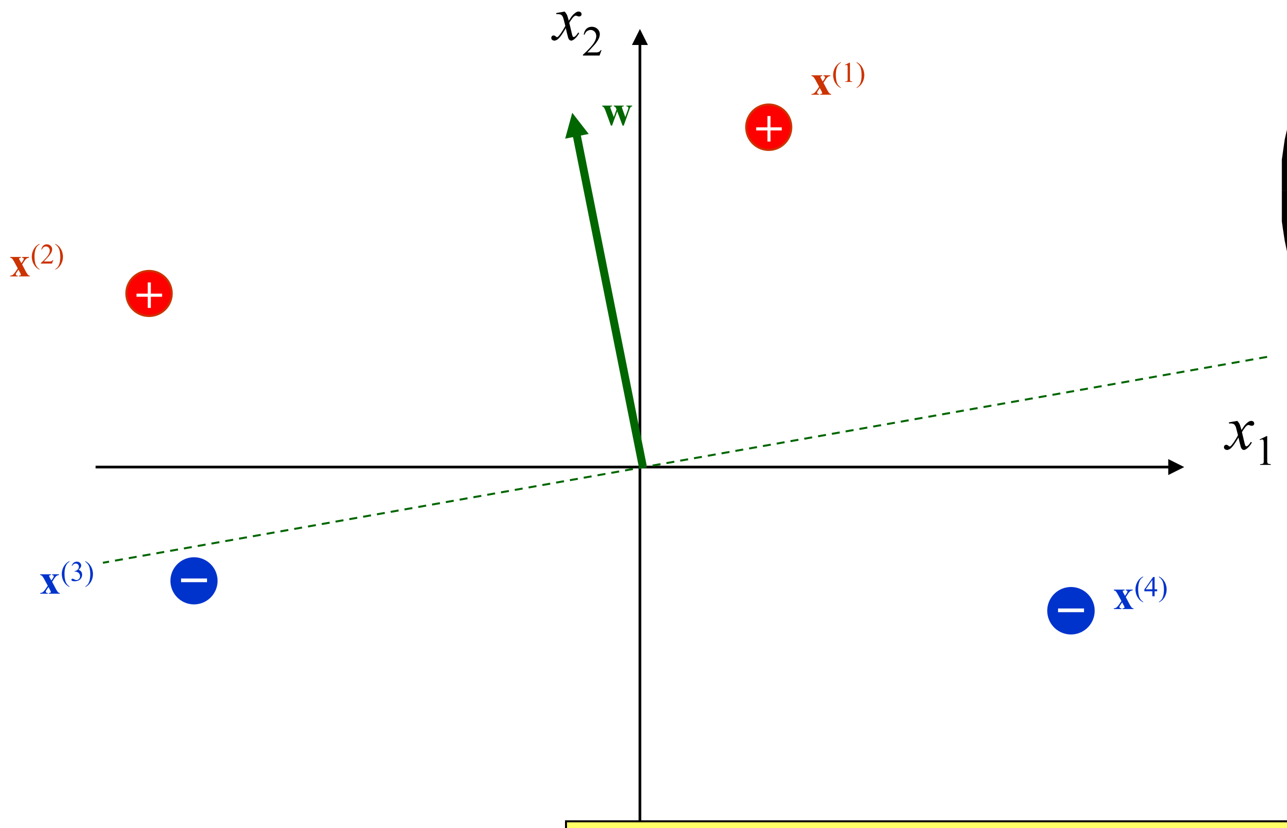
$$\mathbf{w}_4 = \mathbf{w}_3$$

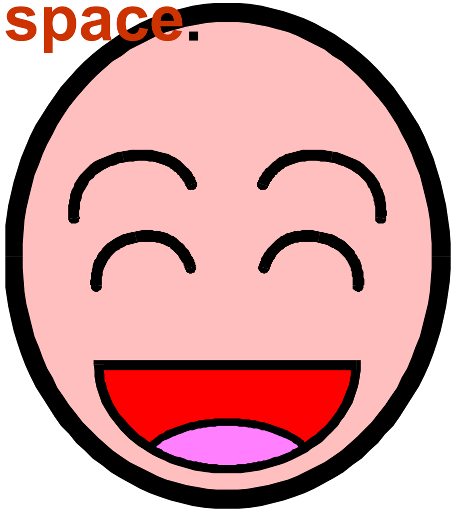
The Learning Scenario



The Learning Scenario

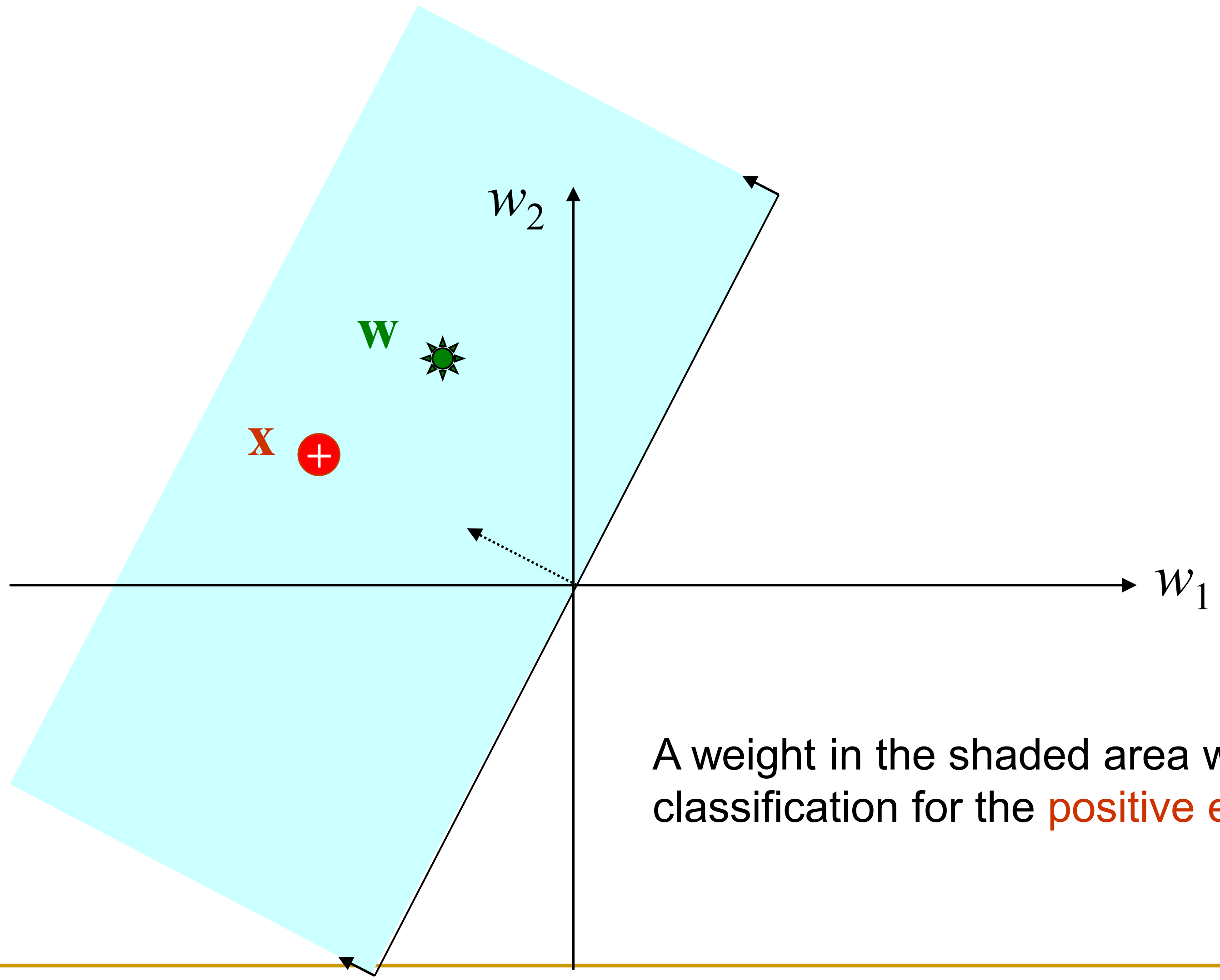
The demonstration is in augmented space.





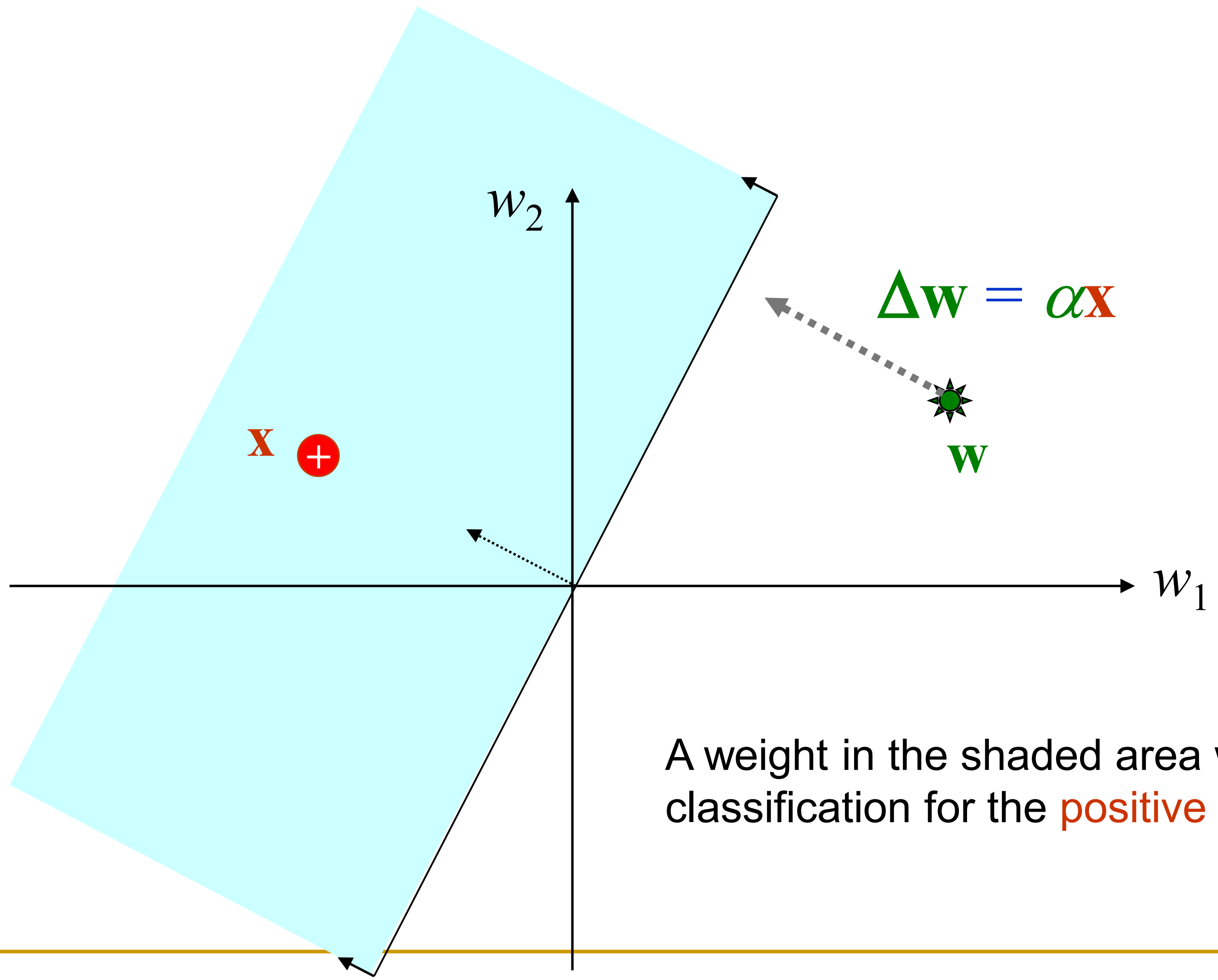
Conceptually, in augmented space, we adjust the weight vector to fit the data.

Weight Space



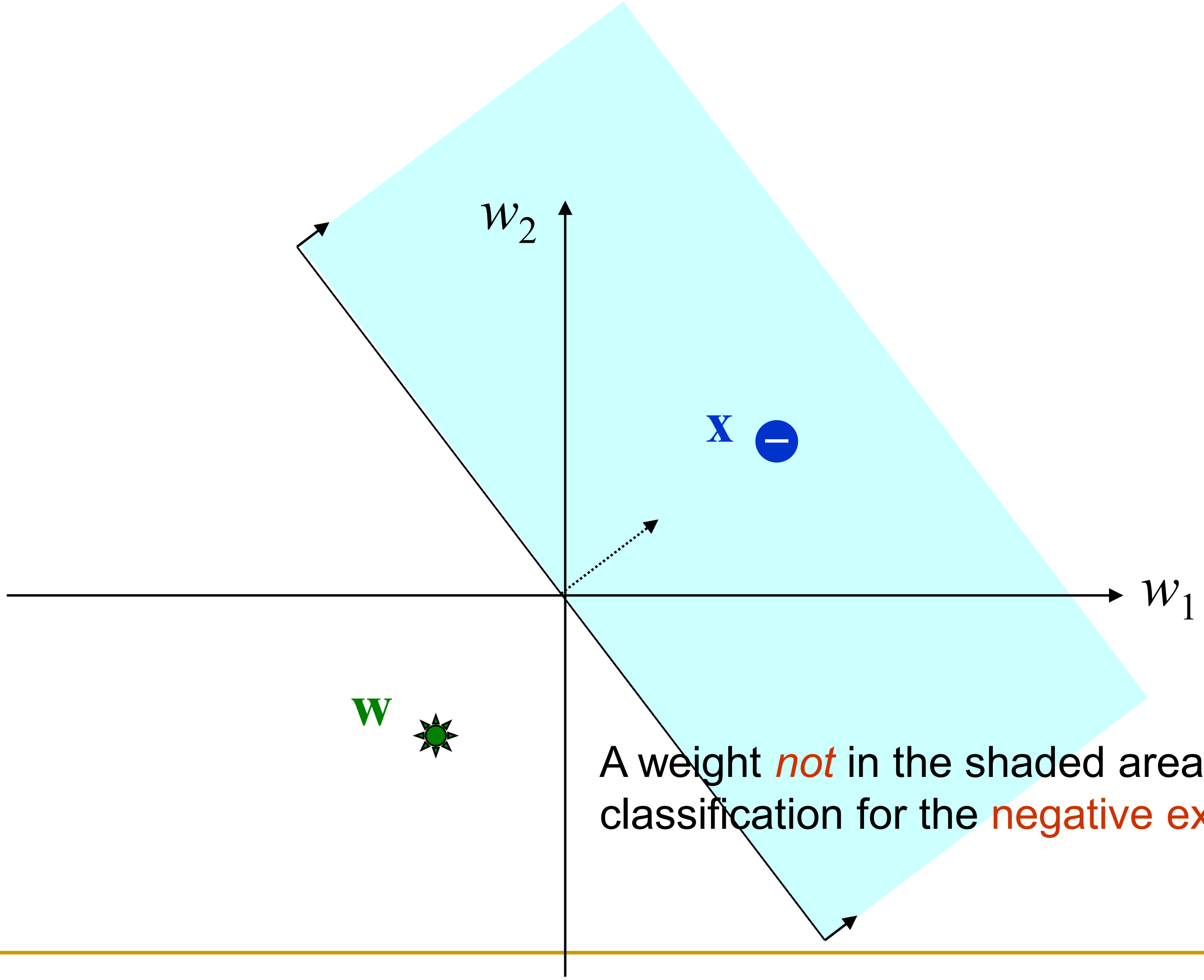
A weight in the shaded area will give correct classification for the positive example.

Weight Space



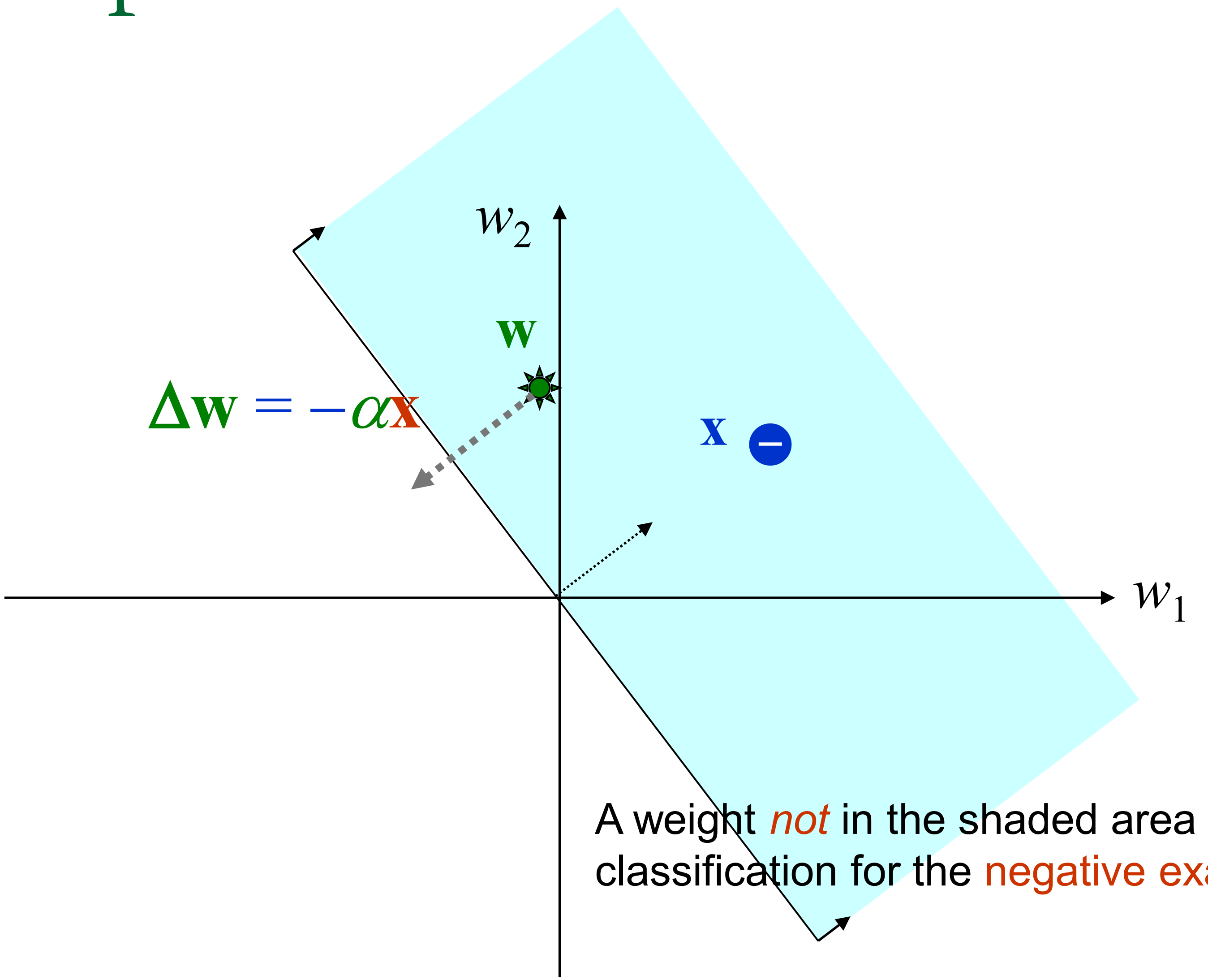
A weight in the shaded area will give correct classification for the positive example.

Weight Space



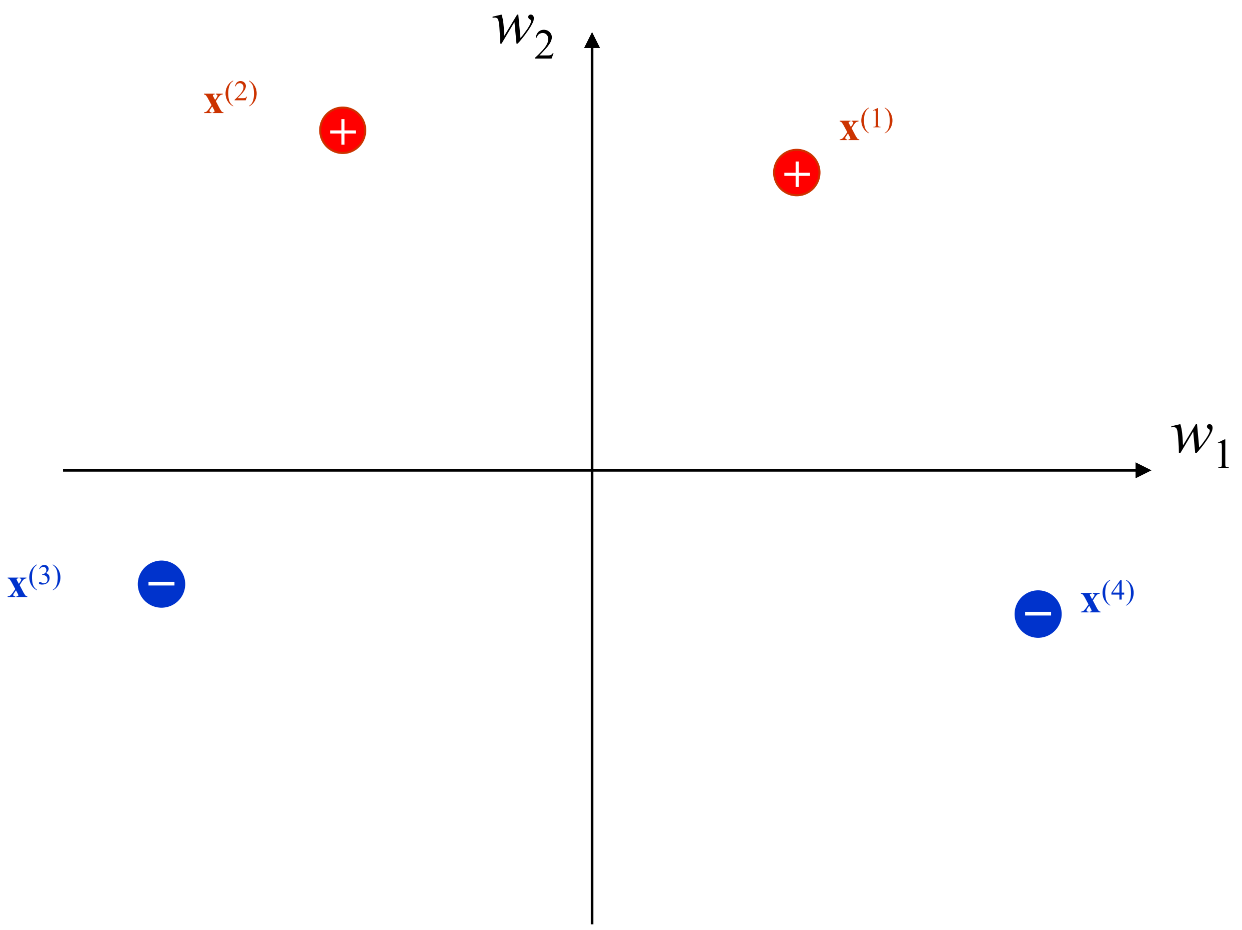
A weight *not* in the shaded area will give correct classification for the negative example.

Weight Space

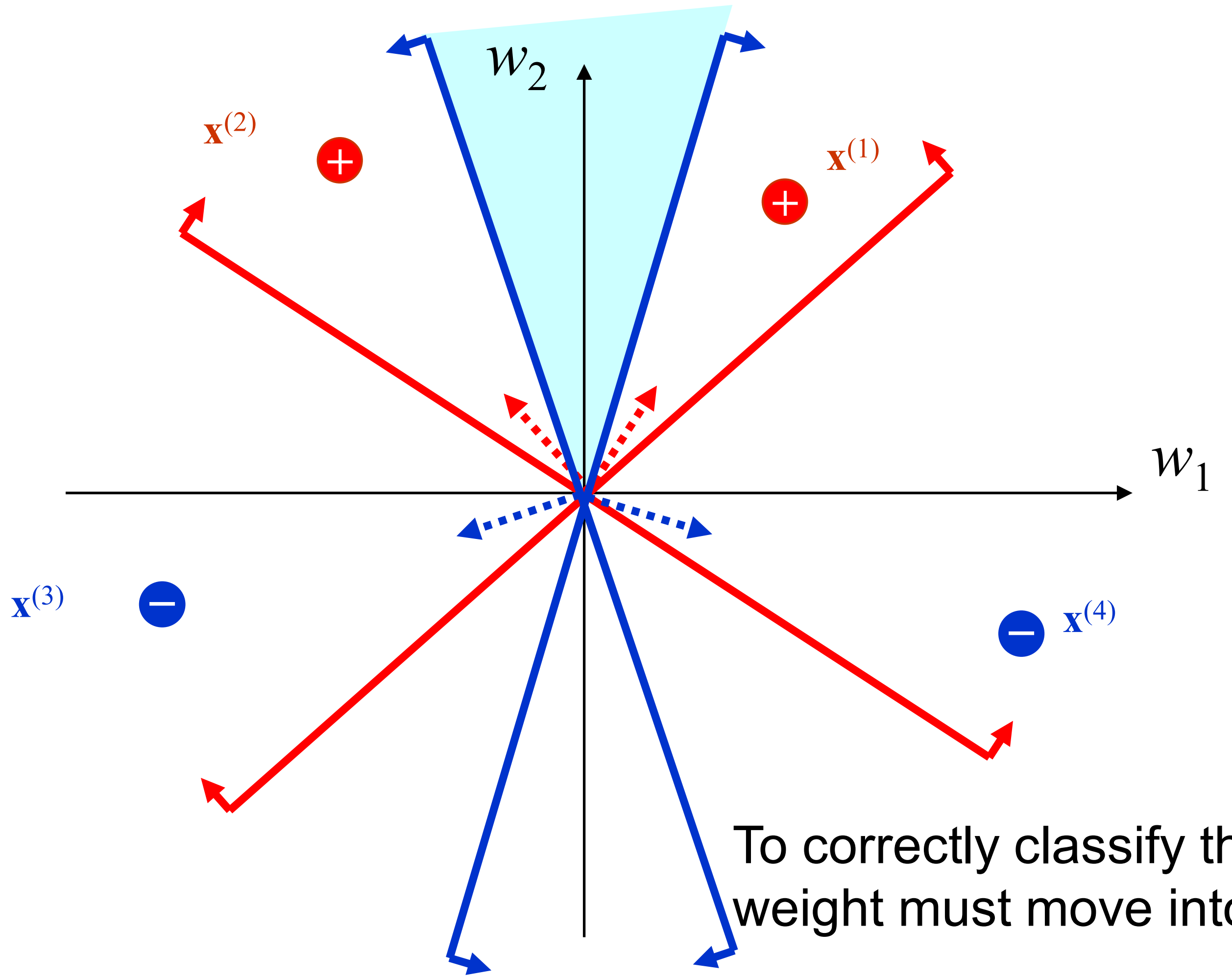


A weight *not* in the shaded area will give correct classification for the negative example.

Learning Scenario in Weight Space

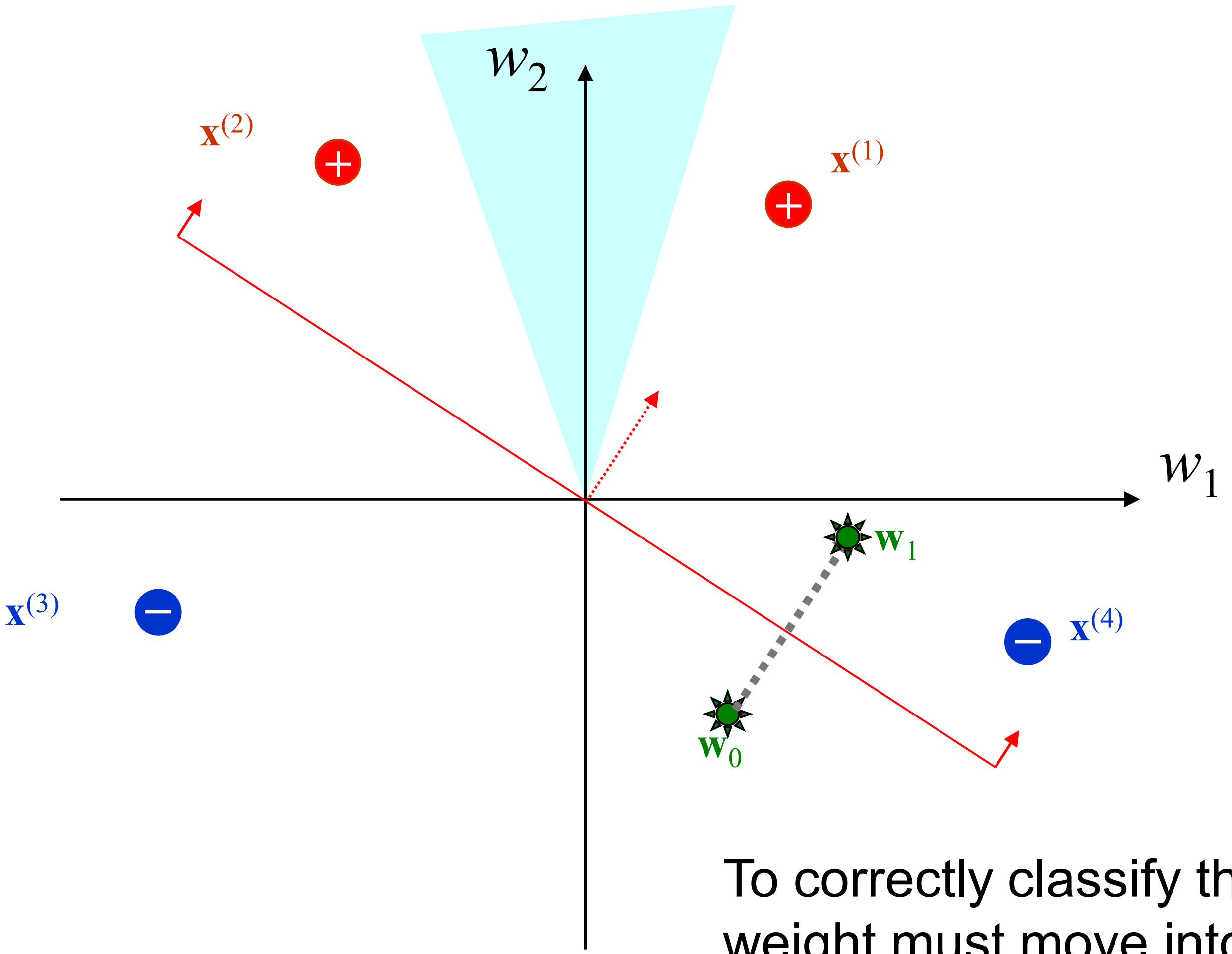


Learning Scenario in Weight Space



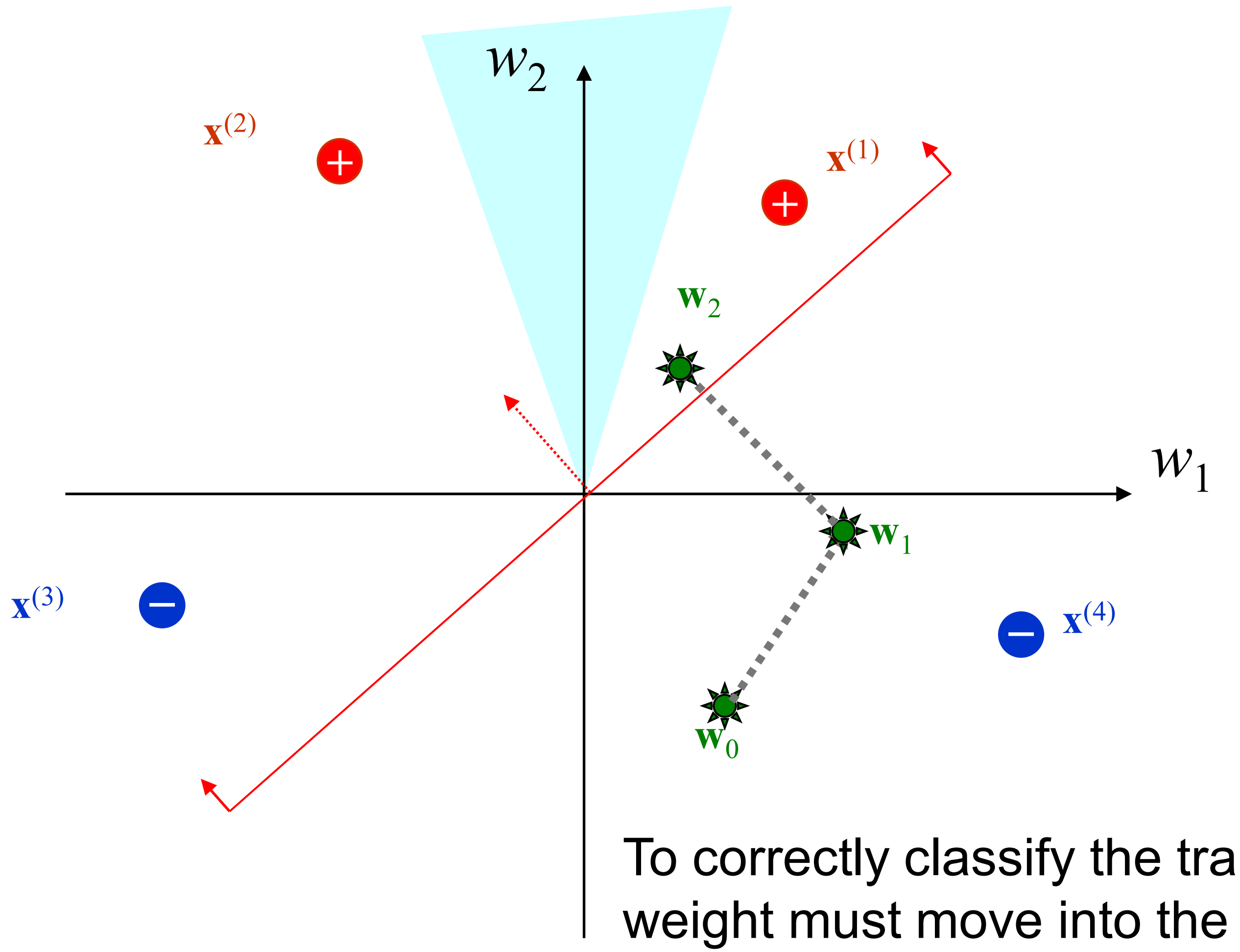
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



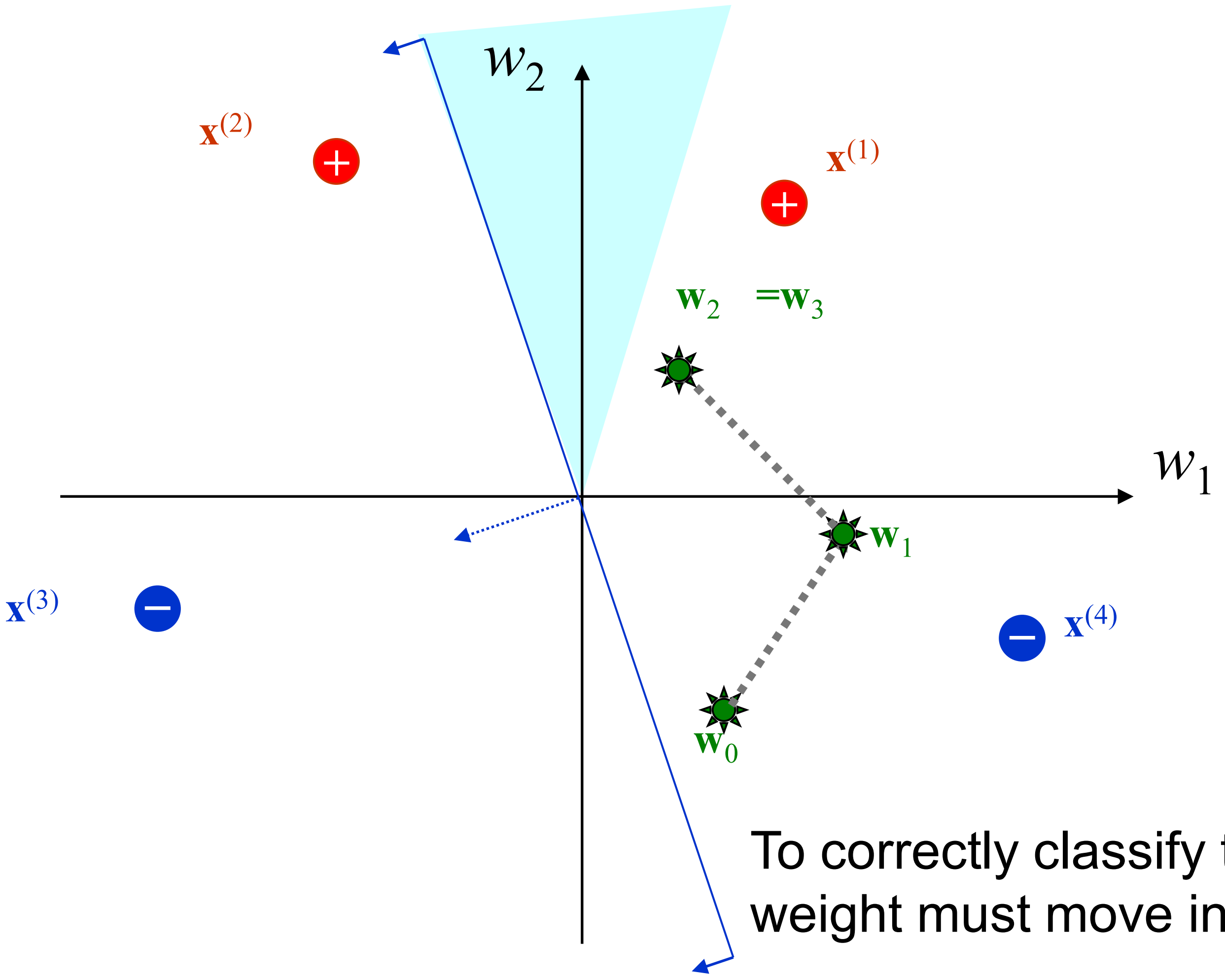
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



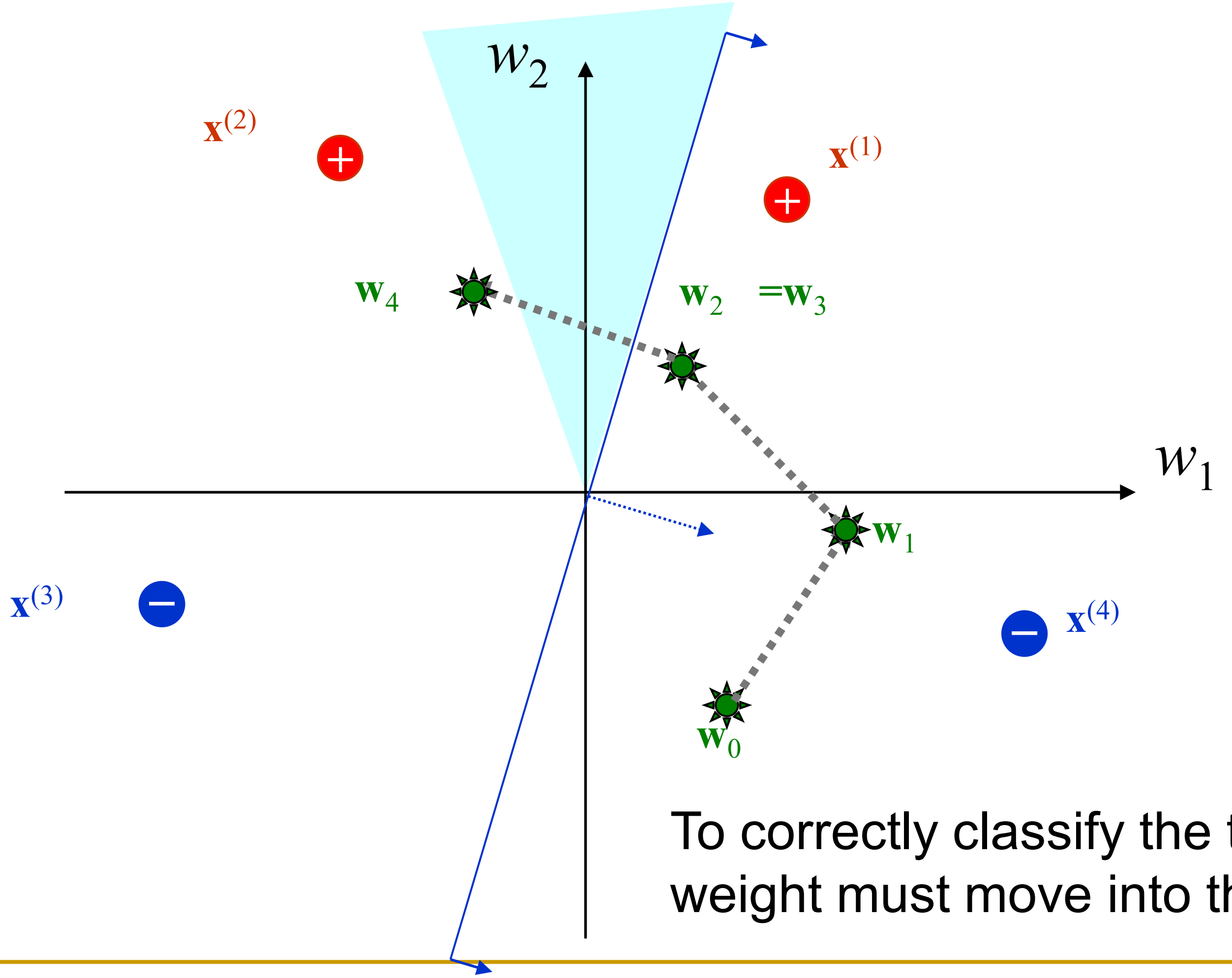
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



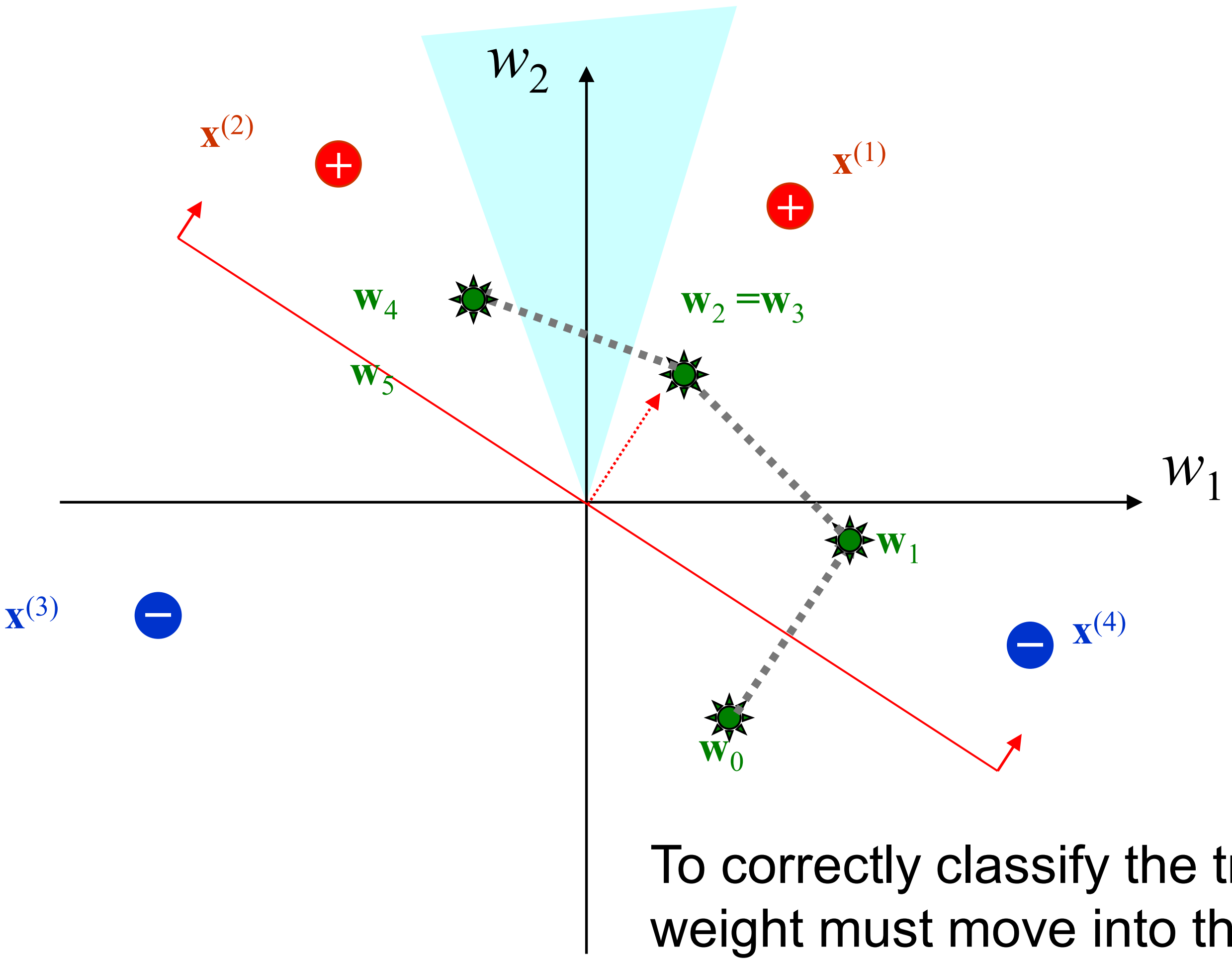
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



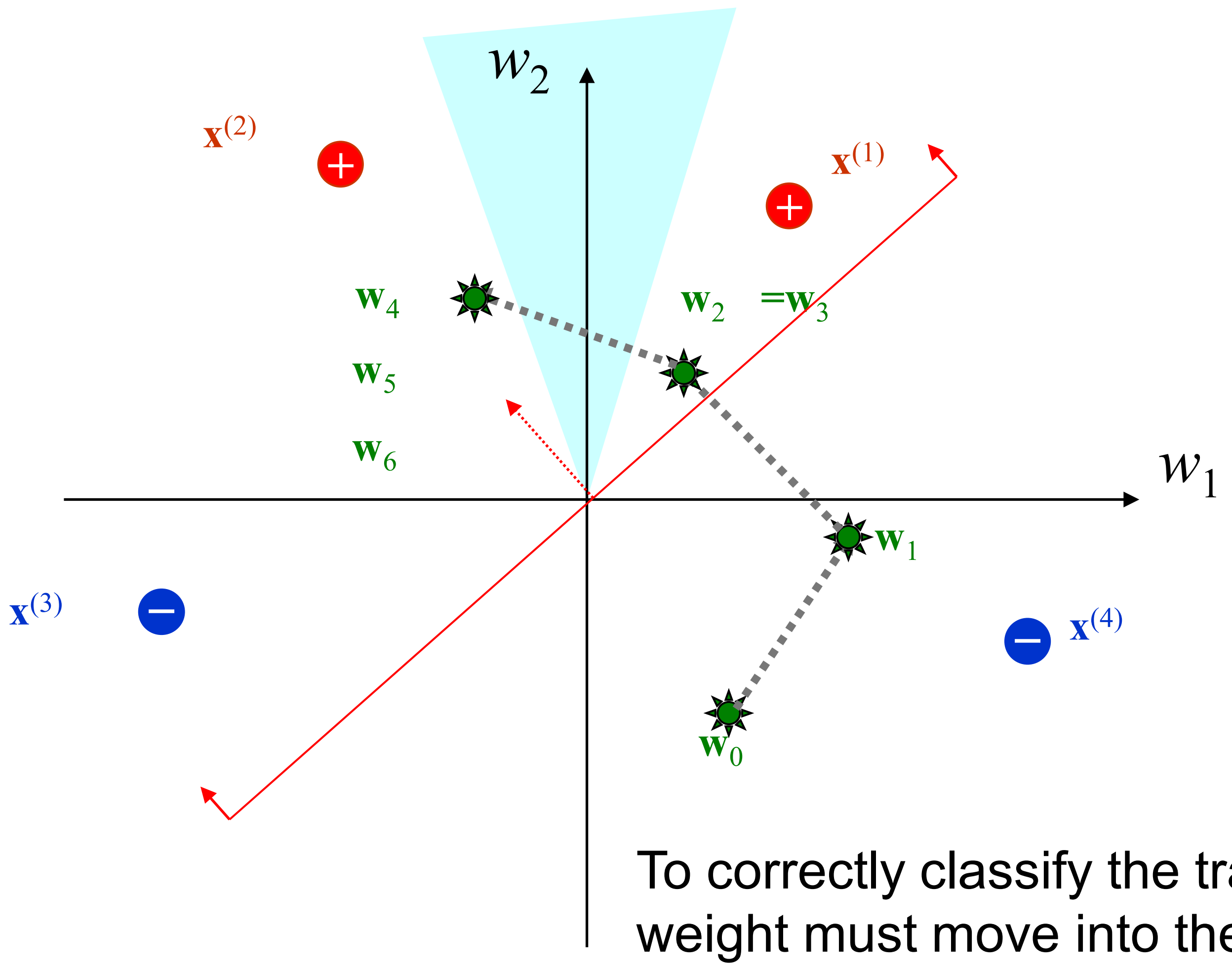
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



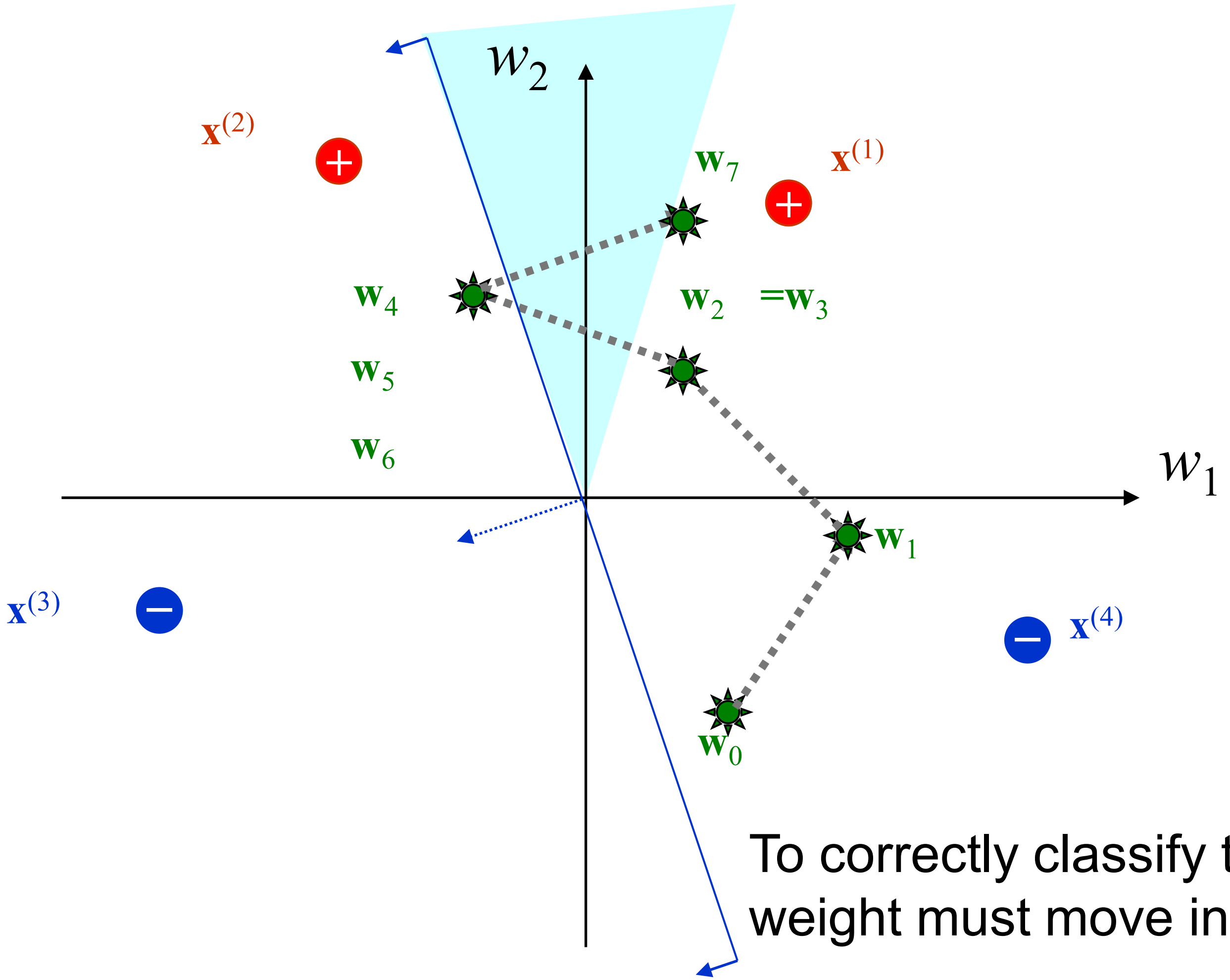
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



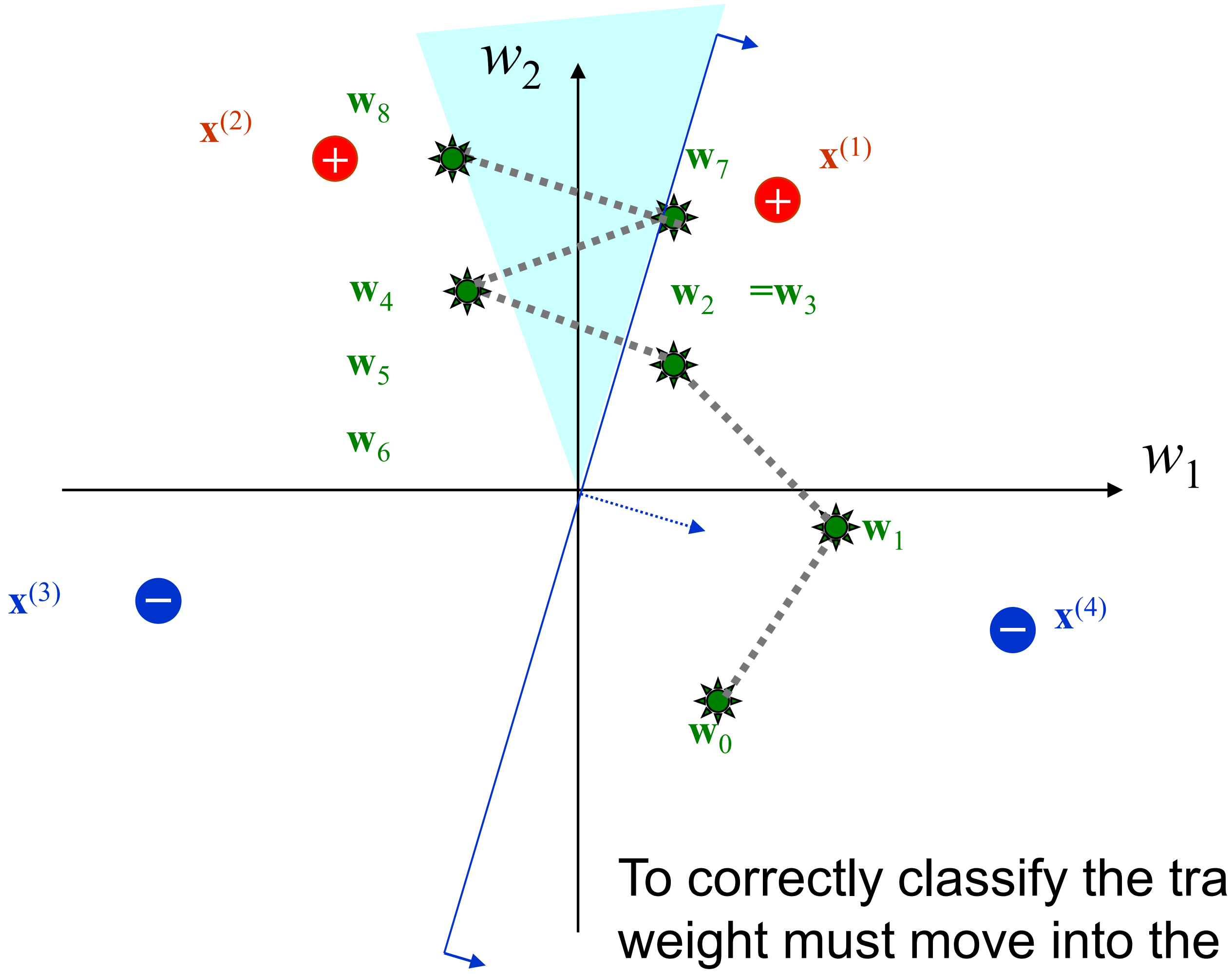
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



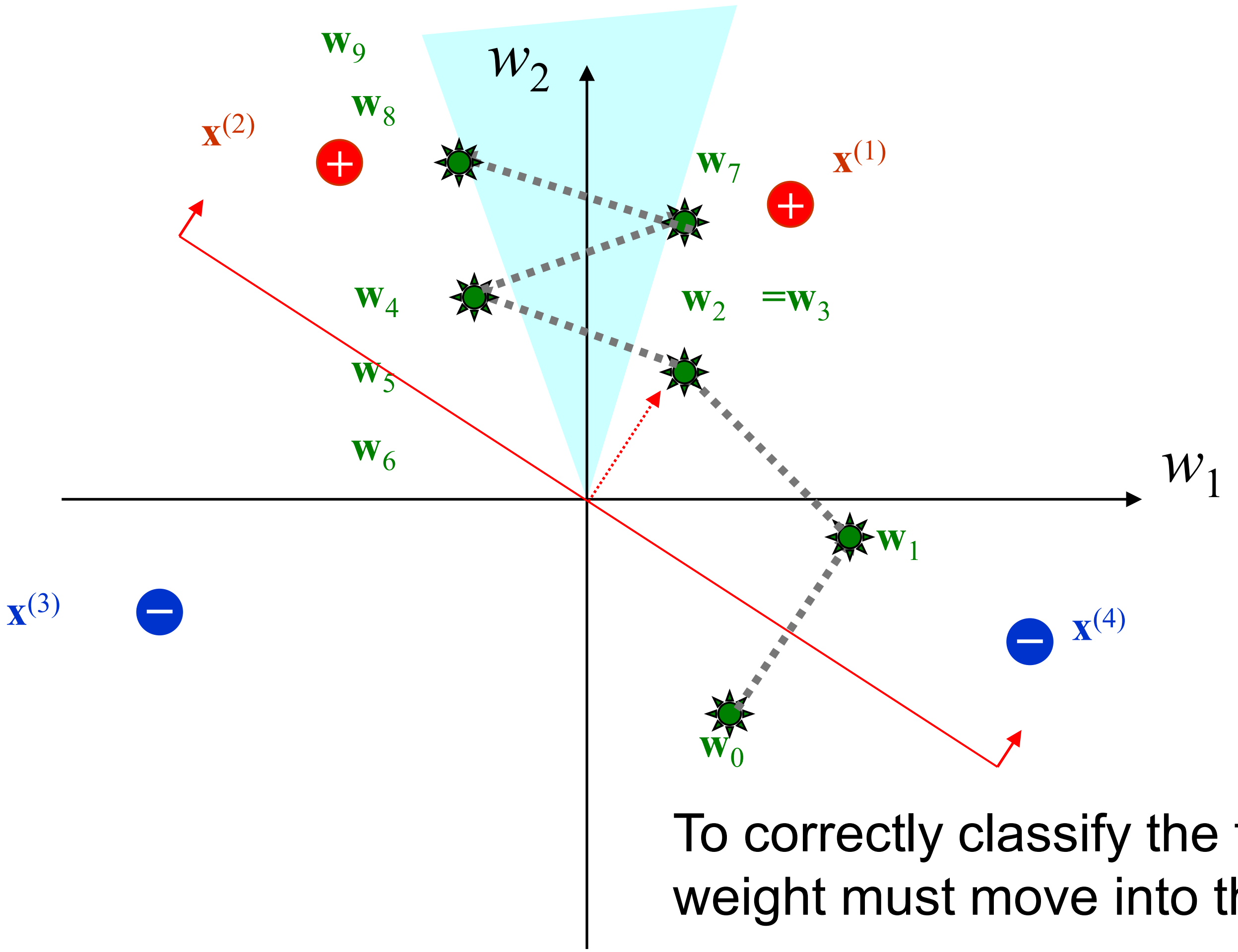
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



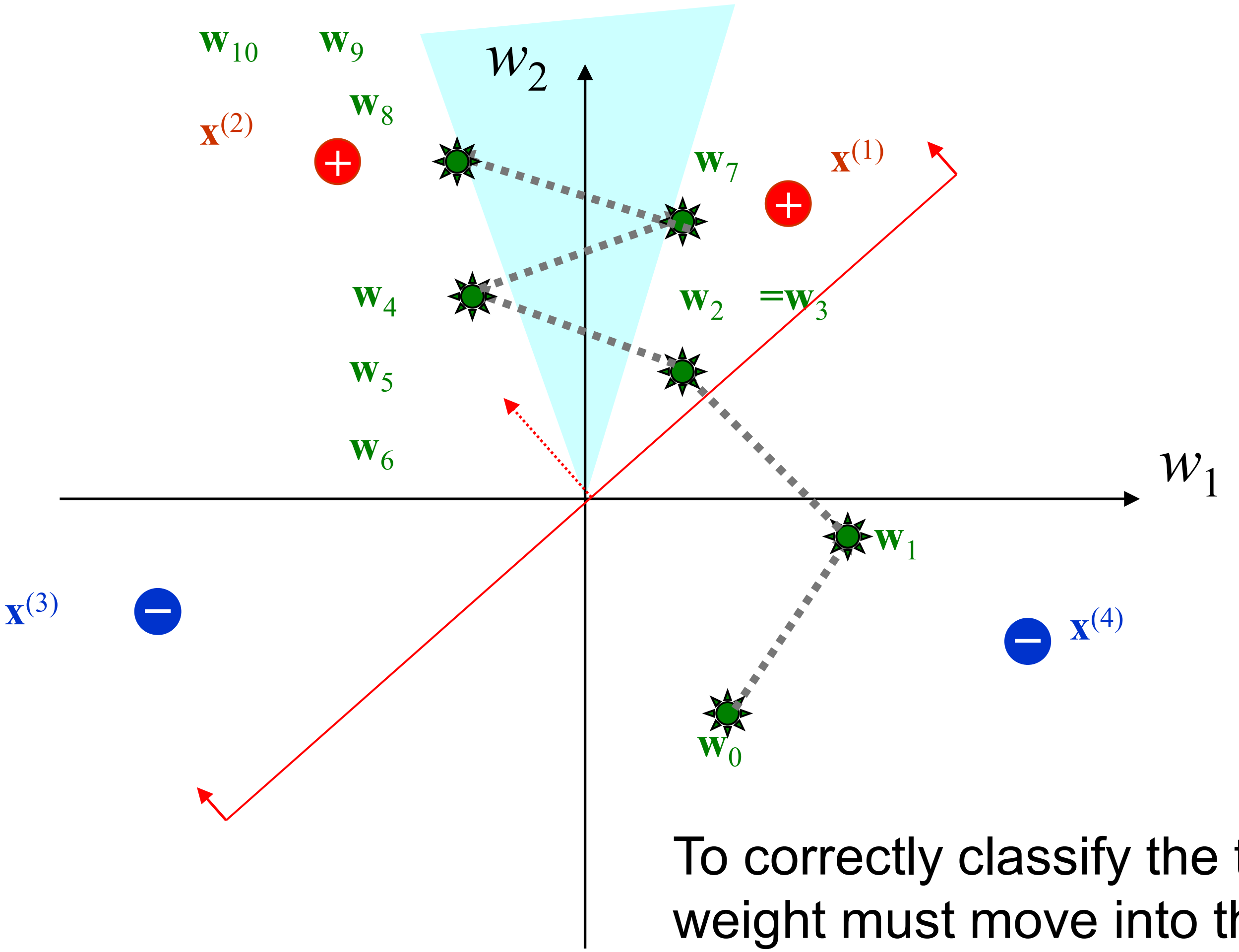
To correctly classify the training set, the weight must move into the shaded area.

The Learning Scenario in Weight Space



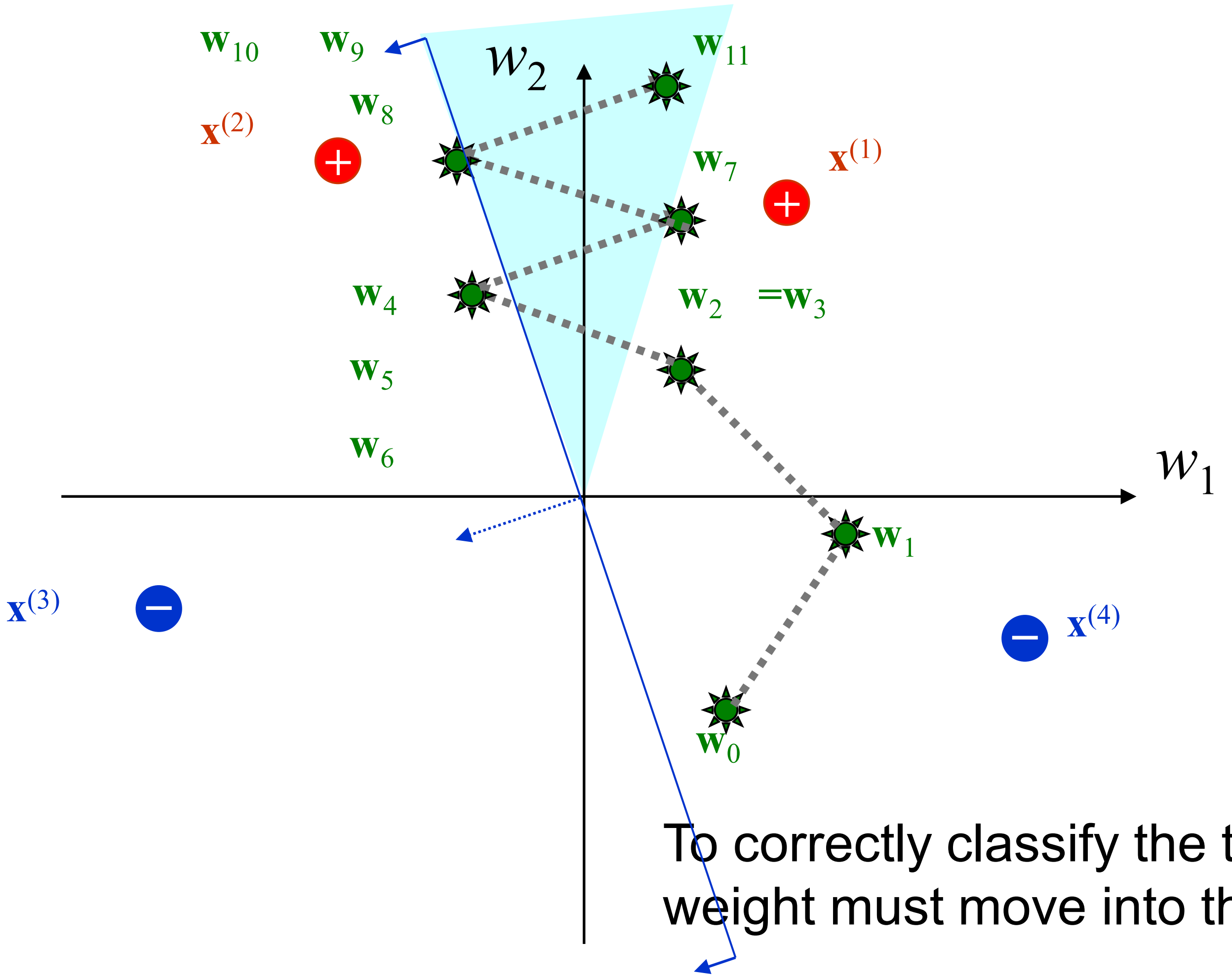
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space



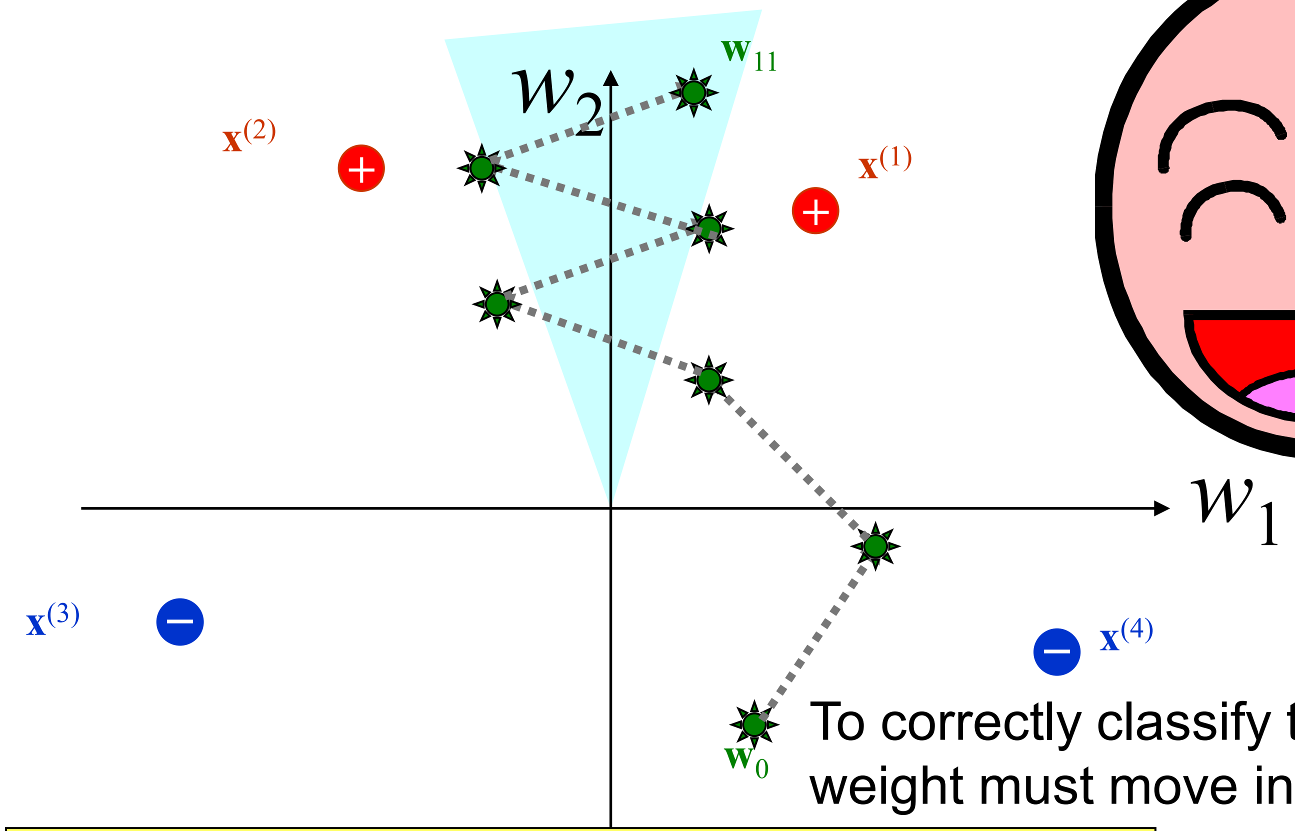
To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space

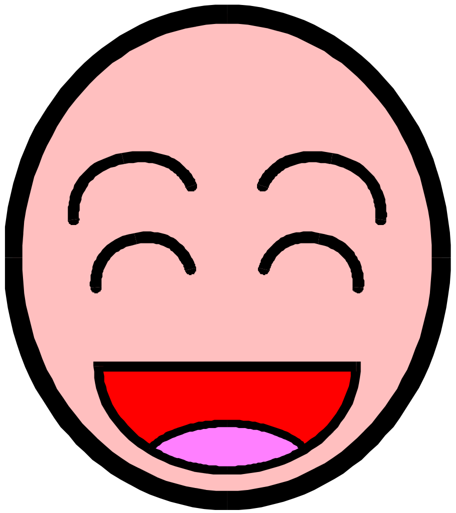


To correctly classify the training set, the weight must move into the shaded area.

Learning Scenario in Weight Space

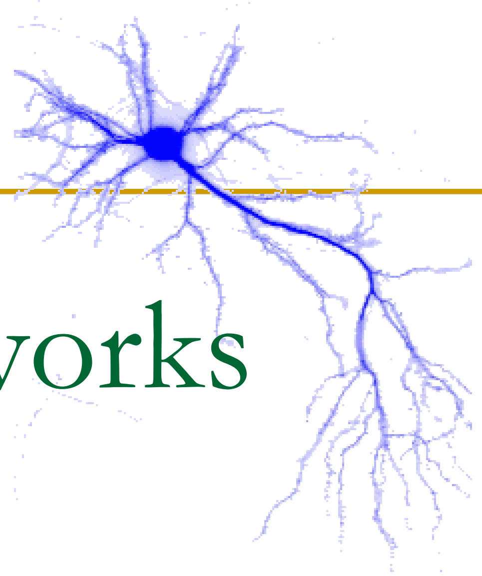


To correctly classify the training set, the weight must move into the shaded area.



Conceptually, in weight space, we move the weight into the feasible region.

Feed-Forward Neural Networks

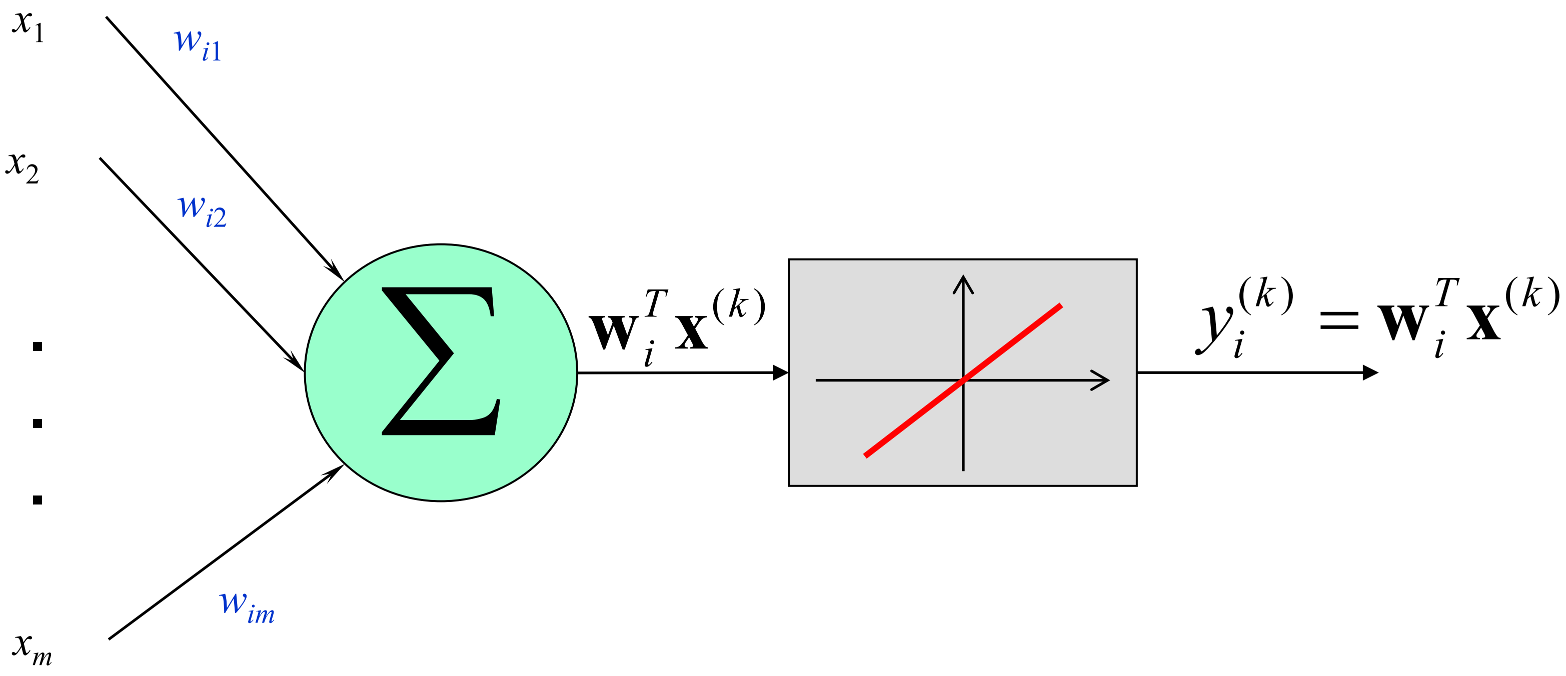


Learning Rules for
Single-Layered Perceptron Networks

- Perceptron Learning Rule
- Adaline Leaning Rule
- δ-Learning Rule

Adaline (Adaptive Linear Element)

Widrow [1962]



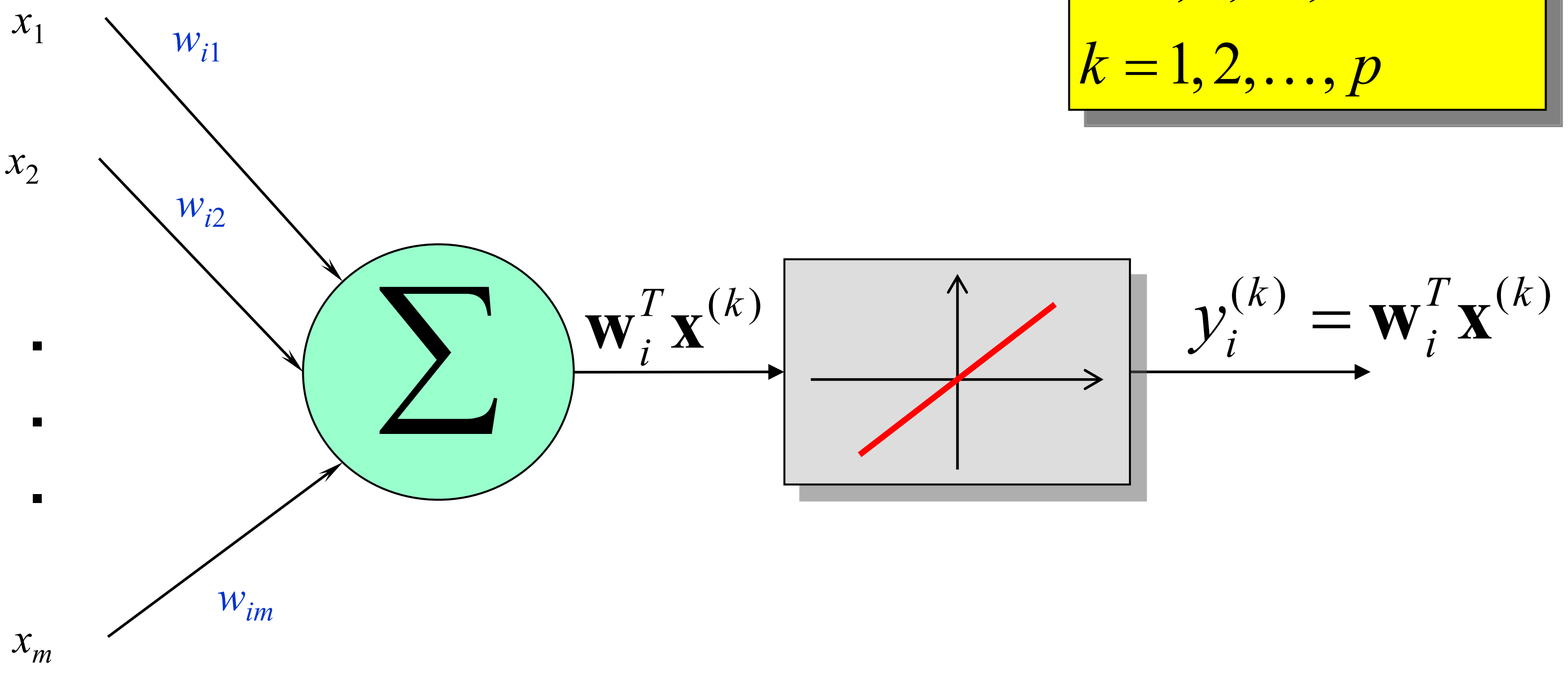
Adaline (Adaptive Linear Element)

In what condition, the goal is reachable?

Widrow [1962]

Goal:

$$y_i^{(k)} = \mathbf{w}_i^T \mathbf{x}^{(k)} = d_i^{(k)}$$
$$i = 1, 2, \ldots, n$$
$$k = 1, 2, \ldots, p$$



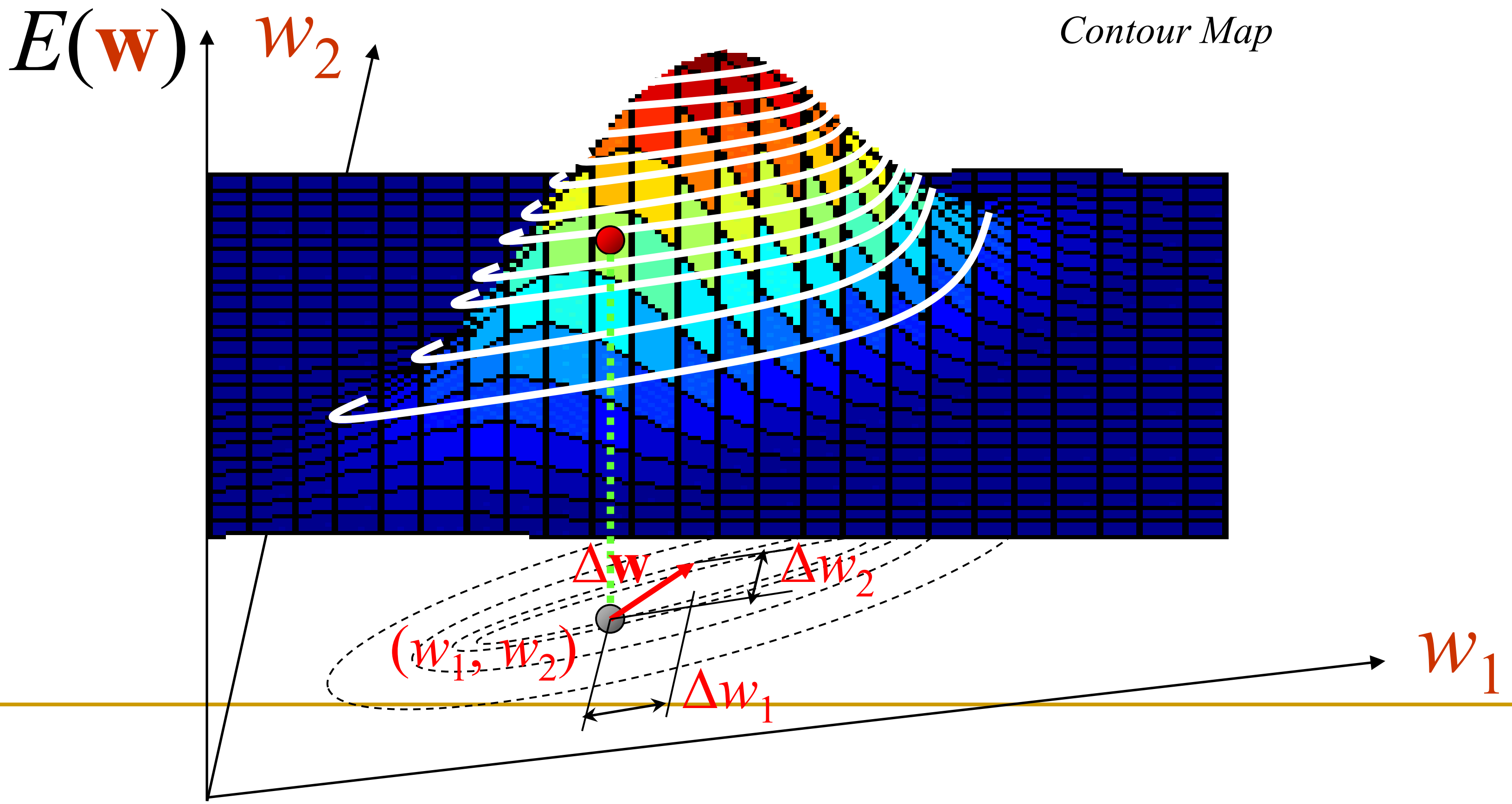
LMS (Least Mean Square)

Minimize the cost function (error function):

$$
\begin{aligned}
E(\mathbf{w}) &= \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - y^{(k)}\right)^2 \\
&= \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)^2 \\
&= \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - \sum_{l=1}^{m} w_l x_l^{(k)}\right)^2
\end{aligned}
$$

Gradient Descent Algorithm

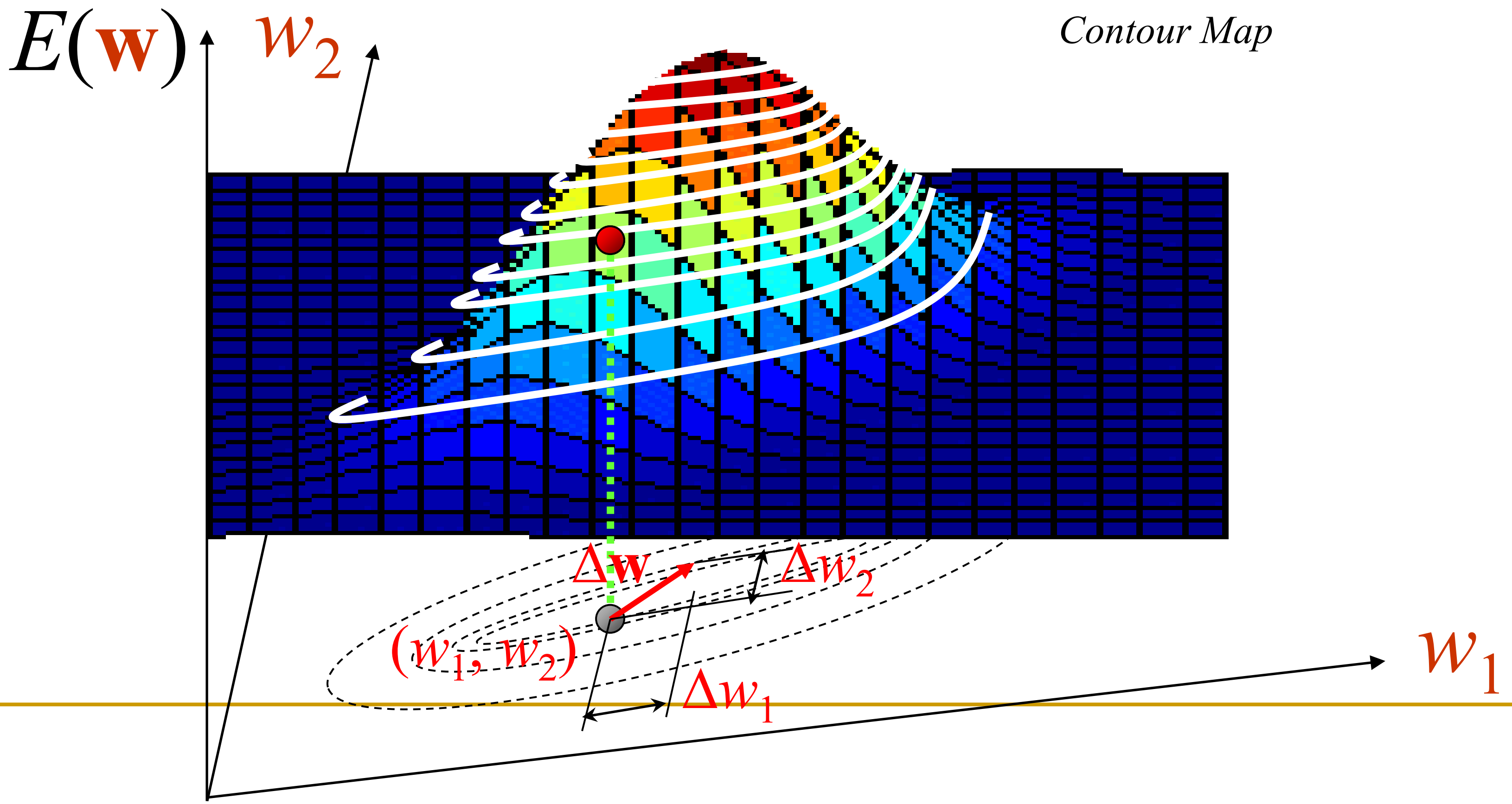
Our goal is to go *downhill*.



Gradient Descent Algorithm

Our goal is to go *downhill*.

How to find the steepest decent direction?



Gradient Operator

Let $f(\mathbf{w}) = f(w_1, w_2, \ldots, w_m)$ be a function over R^m.

$$df = \frac{\partial f}{\partial w_1} dw_1 + \frac{\partial f}{\partial w_2} dw_2 + \cdots + \frac{\partial f}{\partial w_m} dw_m$$

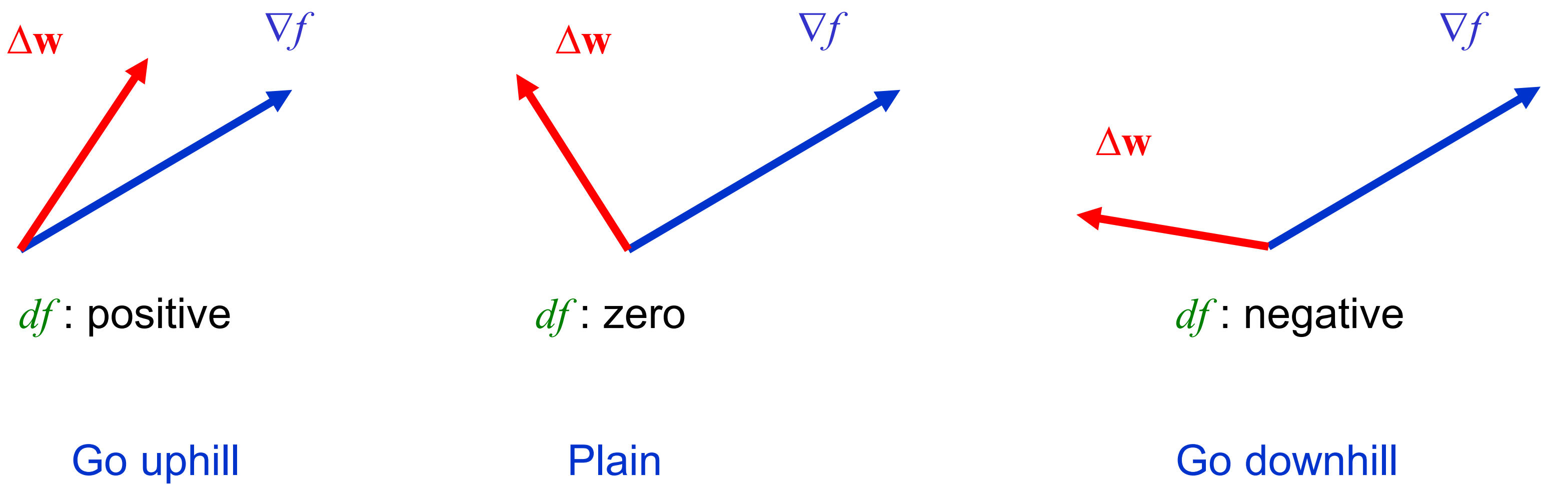
Define

$$\nabla f = \left(\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, \cdots, \frac{\partial f}{\partial w_m} \right)^T$$

$$\Delta \mathbf{w} = \left(dw_1, dw_2, \cdots, dw_m \right)^T$$

$$df = \langle \nabla f, \Delta \mathbf{w} \rangle = \nabla f \bullet \Delta \mathbf{w}$$

Gradient Operator

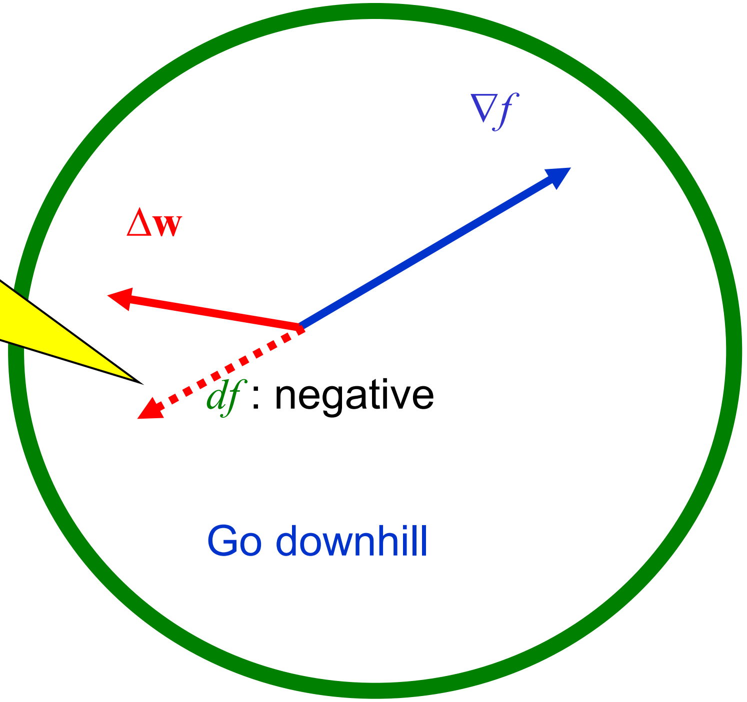


$$df = \langle \nabla f, \Delta\mathbf{w} \rangle = \nabla f \bullet \Delta\mathbf{w}$$

Steepest Decent Direction

To minimize f , we choose

$$\Delta \mathbf{w} = -\eta \nabla f$$



df : positive

Go uphill

df : zero

Plain

$$df = \langle \nabla f, \Delta \mathbf{w} \rangle = \nabla f \bullet \Delta \mathbf{w}$$

LMS (Least Mean Square)

Minimize the cost function (error function):

$$E(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{p}\left(d^{(k)} - \sum_{l=1}^{m} w_l x_l^{(k)}\right)^2$$

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = -\sum_{k=1}^{p}\left(d^{(k)} - \sum_{l=1}^{m} w_l x_l^{(k)}\right)x_j^{(k)}$$

$$= -\sum_{k=1}^{p}\left(d^{(k)} - \mathbf{w}^T\mathbf{x}^{(k)}\right)x_j^{(k)} = -\sum_{k=1}^{p}\overbrace{\left(d^{(k)} - y^{(k)}\right)}^{\delta^{(k)}}x_j^{(k)}$$

$$\frac{\partial E(\mathbf{w})}{\partial w_j} = -\sum_{k=1}^{p}\delta^{(k)}x_j^{(k)}$$

$$\delta^{(k)} = d^{(k)} - y^{(k)}$$