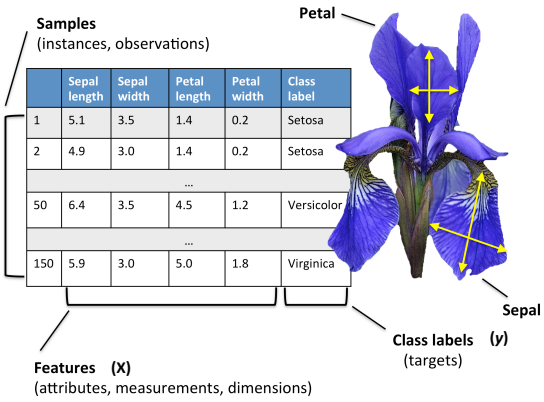
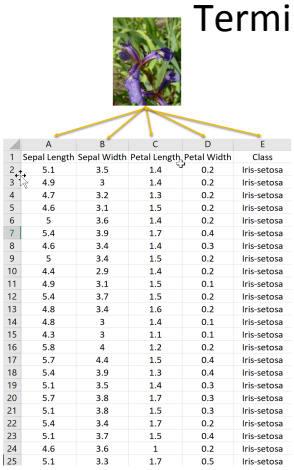
Perceptron Model and Adaline Model

Basic Terminology



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Terminology & Notation



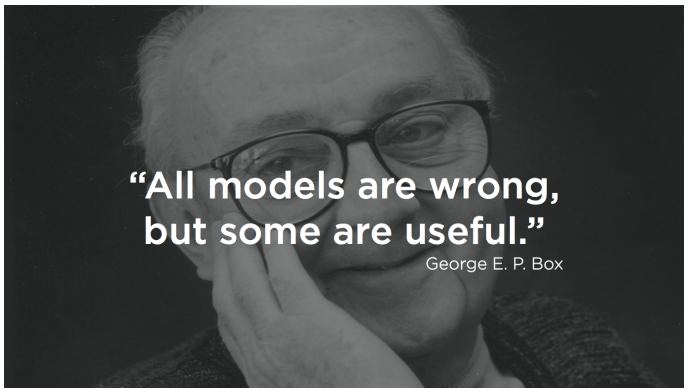
	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa
8	4.6	3.4	1.4	0.3	Iris-setosa
9	5	3.4	1.5	0.2	Iris-setosa
10	4.4	2.9	1.4	0.2	Iris-setosa
11	4.9	3.1	1.5	0.1	Iris-setosa
12	5.4	3.7	1.5	0.2	Iris-setosa
13	4.8	3.4	1.6	0.2	Iris-setosa
14	4.8	3	1.4	0.1	Iris-setosa
15	4.3	3	1.1	0.1	Iris-setosa
16	5.8	4	1.2	0.2	Iris-setosa
17	5.7	4.4	1.5	0.4	Iris-setosa
18	5.4	3.9	1.3	0.4	Iris-setosa
19	5.1	3.5	1.4	0.3	Iris-setosa
20	5.7	3.8	1.7	0.3	Iris-setosa
21	5.1	3.8	1.5	0.3	Iris-setosa
22	5.4	3.4	1.7	0.2	Iris-setosa
23	5.1	3.7	1.5	0.4	Iris-setosa
24	4.6	3.6	1	0.2	Iris-setosa
25	5.1	3.3	1.7	0.5	Iris-setosa

Feature Table	
Feature # 1	Sepal Length
Feature # 2	Sepal Width
Feature # 3	Petal Length
Feature # 4	Petal Width

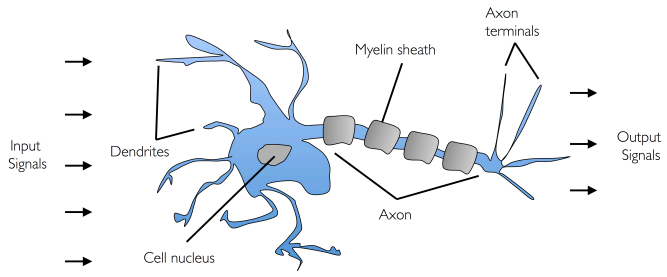
INPUT: The features of the 150 samples can be written as a 150*4 matrix.

Target: [Setosa, Versicolor, Virginica]

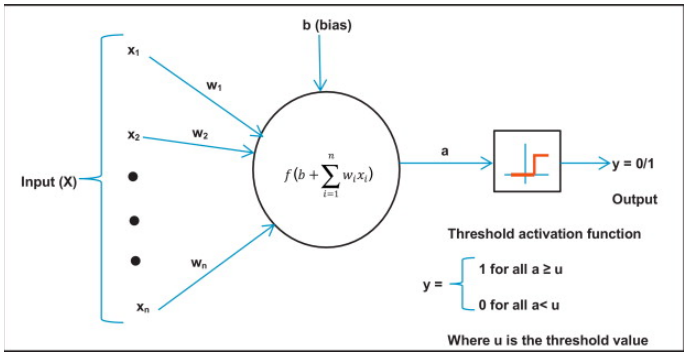
TARGET: The targets/labels of the 150 samples can be written as a 150-dimensional column vector.



Biology Neuron



Perceptron Model



x: input w: weight

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

$$z = w_1 x_1 + \ldots + w_m x_m$$

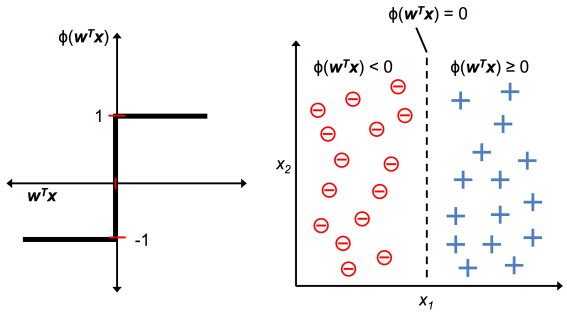
Unit step function

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

$$z = w_0 x_0 + w_1 x_1 + \ldots + w_m x_m = \boldsymbol{w}^T \boldsymbol{x}$$

Add a bias unit and the step function becomes

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$



The Perceptron Learning Rule

1. Initialize the weights to 0 or small random numbers.
2. For each training sample $\boldsymbol{x}^{(i)}$:
   a. Compute the output value $\hat{y}$.
   b. Update the weights.

Here, the output value is the class label predicted by the unit step function that we defined earlier, and the simultaneous update of each weight w_j in the weight vector **w** can be more formally written as:

$$w_j := w_j + \Delta w_j$$

The value of Δw_j, which is used to update the weight w_j, is calculated by the perceptron learning rule:

$$\Delta w_j = \eta \left(y^{(i)} - \hat{y}^{(i)} \right) x_j^{(i)}$$

Exercise: Perceptron Learning

- Calculate the weight updates over 1 epoch for given the last two samples in the Iris dataset.

	0	1	2	3	4
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Adaline: Adaptive Linear Neuron

- Minimizing a cost function: Sum of Squared Errors

$$J(\boldsymbol{w}) = \frac{1}{2} \sum_i \left(y^{(i)} - \phi\left(z^{(i)} \right) \right)^2$$

Where:

$\phi\left(\boldsymbol{w}^T \boldsymbol{x} \right) = \boldsymbol{w}^T \boldsymbol{x}$ is the identity activation function

The Adaline Learning Rule

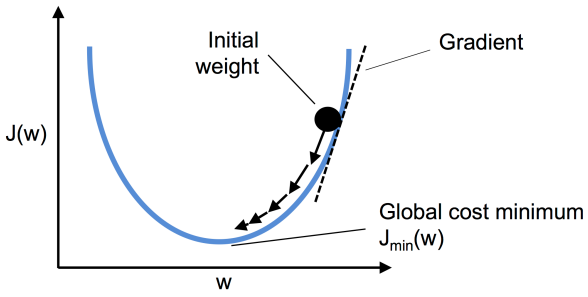
Based on the partial derivative of the cost function with respect to each weight:

$$\Delta w_j = -\eta \frac{\partial J}{\partial w_j} = \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)} \right) \right) x_j^{(i)}$$

Weights re updated simultaneously so we have:

$$\boldsymbol{w} := \boldsymbol{w} + \Delta \boldsymbol{w}$$

Use Gradient Descent to implement Adaline



Perceptron V.S. Adaline

- Both are classifiers for binary classification
- Both have a linear decision boundary

- Perceptron uses the class labels (predicted output) to learn model coefficients
- Adaline uses continuous predicted values (from the net input) to learn the model coefficients.
- In their original form, Perceptron updates the weights every sample, and Adaline updates the weights based on the whole sample set.