Machine Learning: Supervised Classification

Mr. Michaud

Marist School

Computational Perception and Artificial Intelligence

Types of Machine Learning Problems

- Supervised Learning
  - Training Algorithms on Data sets with Attributes and Given Classifications
    - Decision Trees
    - Support Vector Machines
    - Artificial Neural Networks
    - K-Nearest Neighbor (Like Sum of Squared Distances)
- Unsupervised Learning: Grouping Data without Classifications
  - Clustering Algorithms
  - Dimension Reduction
- Reinforcement Learning

Focus of this Assignment: Supervised Learning with Artificial Neural Network

- Given a Data Set *D*
  - Attributes: X
  - Classification Y
- In each Instance of data in D:
  - X_i are the attributes
    - $X_1, X_2, X_3 \ldots . X_i$
    - Y is the "Classification" or Label

$$D \rightarrow \{X_i, Y\}$$

	X_1	X_2	X_3	X_4	Y
	A	B	C	D	E
1	**Sepal Length**	**Sepal Width**	**Petal Length**	**Petal Width**	**Species**
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	7	3.2	4.7	1.4	versicolor
8	6.4	3.2	4.5	1.5	versicolor
9	6.9	3.1	4.9	1.5	versicolor
10	5.5	2.3	4	1.3	versicolor
11	6.5	2.8	4.6	1.5	versicolor
12	6.9	3.2	5.7	2.3	virginica
13	5.6	2.8	4.9	2	virginica
14	7.7	2.8	6.7	2	virginica
15	6.3	2.7	4.9	1.8	virginica
16	6.7	3.3	5.7	2.1	virginica

Goal of Supervised Learning

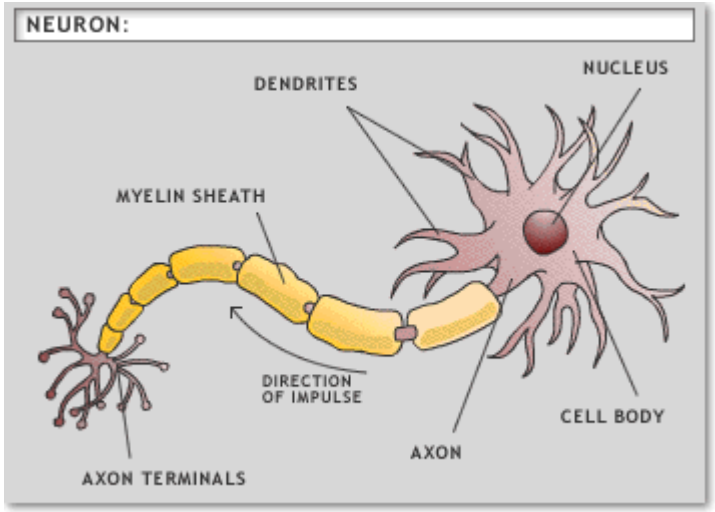
- Given a Data set D with Attributes X_i and Classification Y
  - Build a Learner that will accurately predict the classification Y for a given set of attributes Xi

- We build this Learner by training with a data set D with thousands of samples and creating a Hypothesis (H).

- Then with Hypothesis H we can pass in a set of Attributes X_i and get a prediction Y

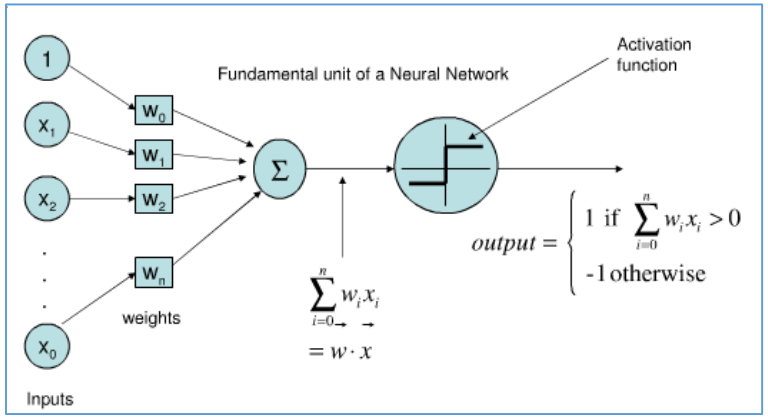
Learner: Artificial Neural Network (ANN)

- Models the interaction of neurons in a biological system.
- Brain cells:
  - Thousands of connections to other brain cells via dendrites.
  - Thought happens (modelled) through signals passed along the connections between neurons

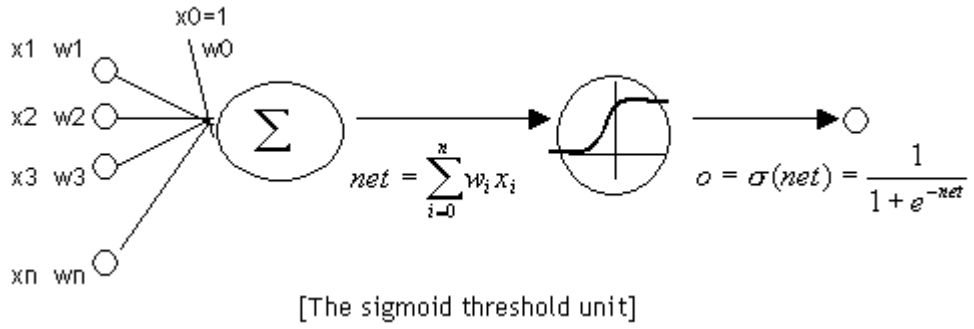


Perceptron: The “neuron” of an ANN

- Perceptron is the mathematical model of a neuron in an Artificial Neural Network.
- Two types of Perceptrons:
  - Binary
  - Sigmoid

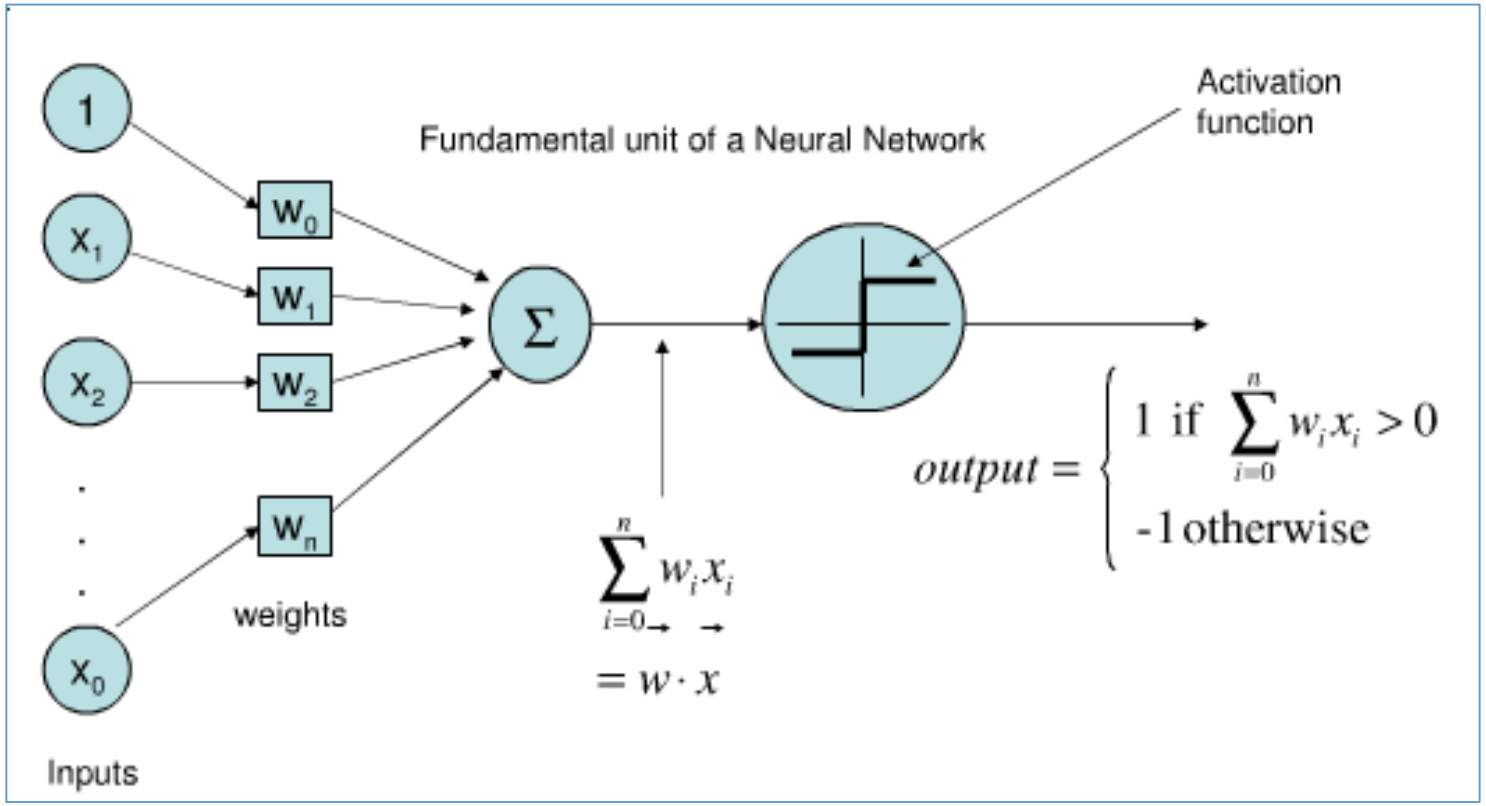


Binary

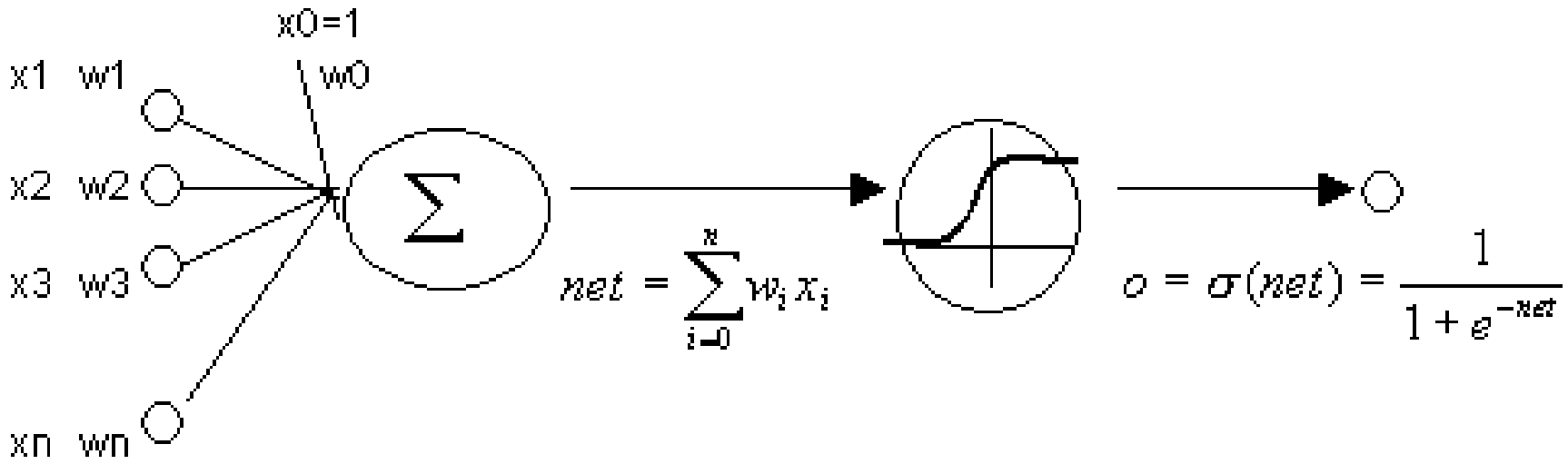


Sigmoid

Binary Perceptron

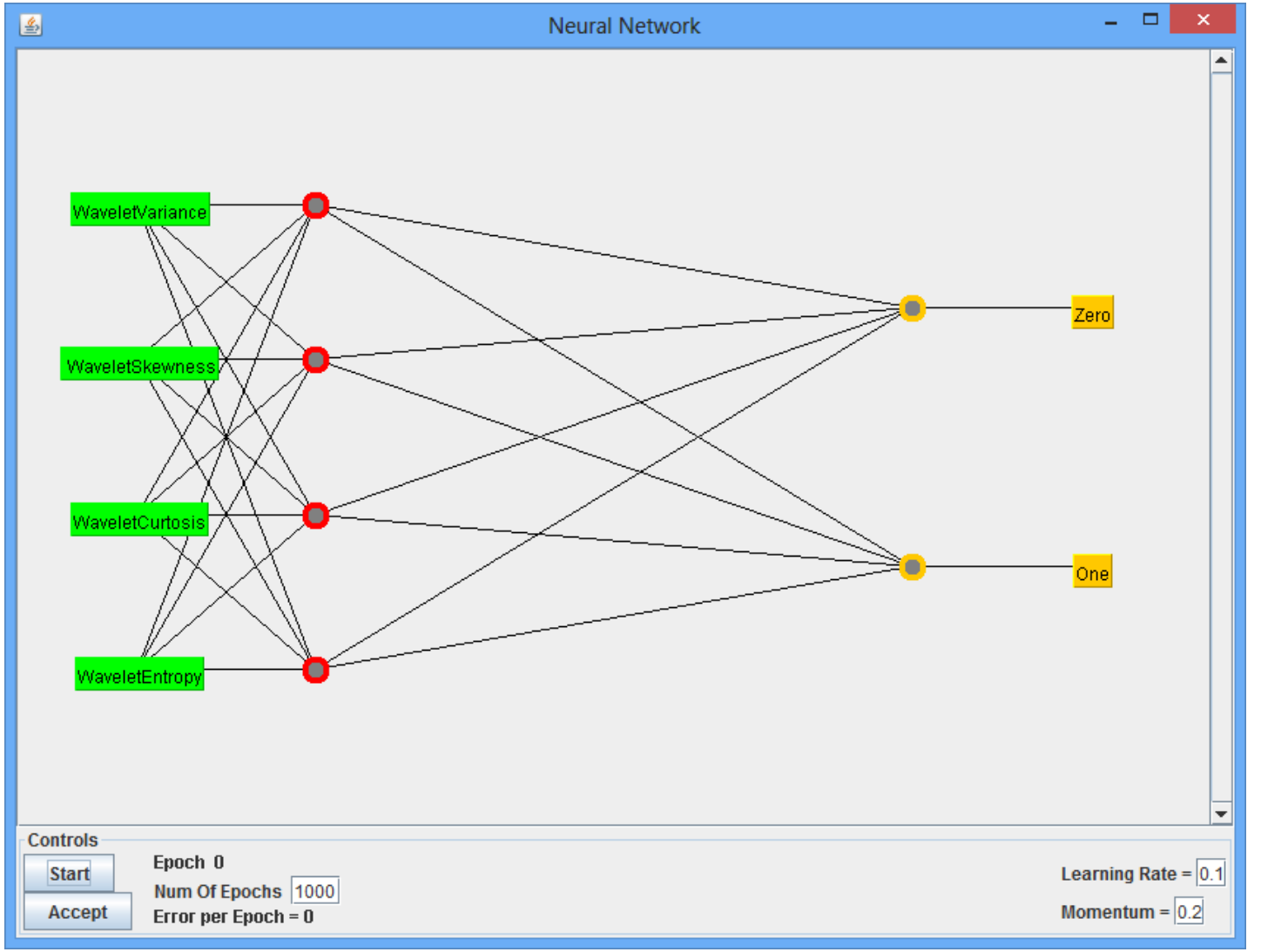


Sigmoid Perceptron

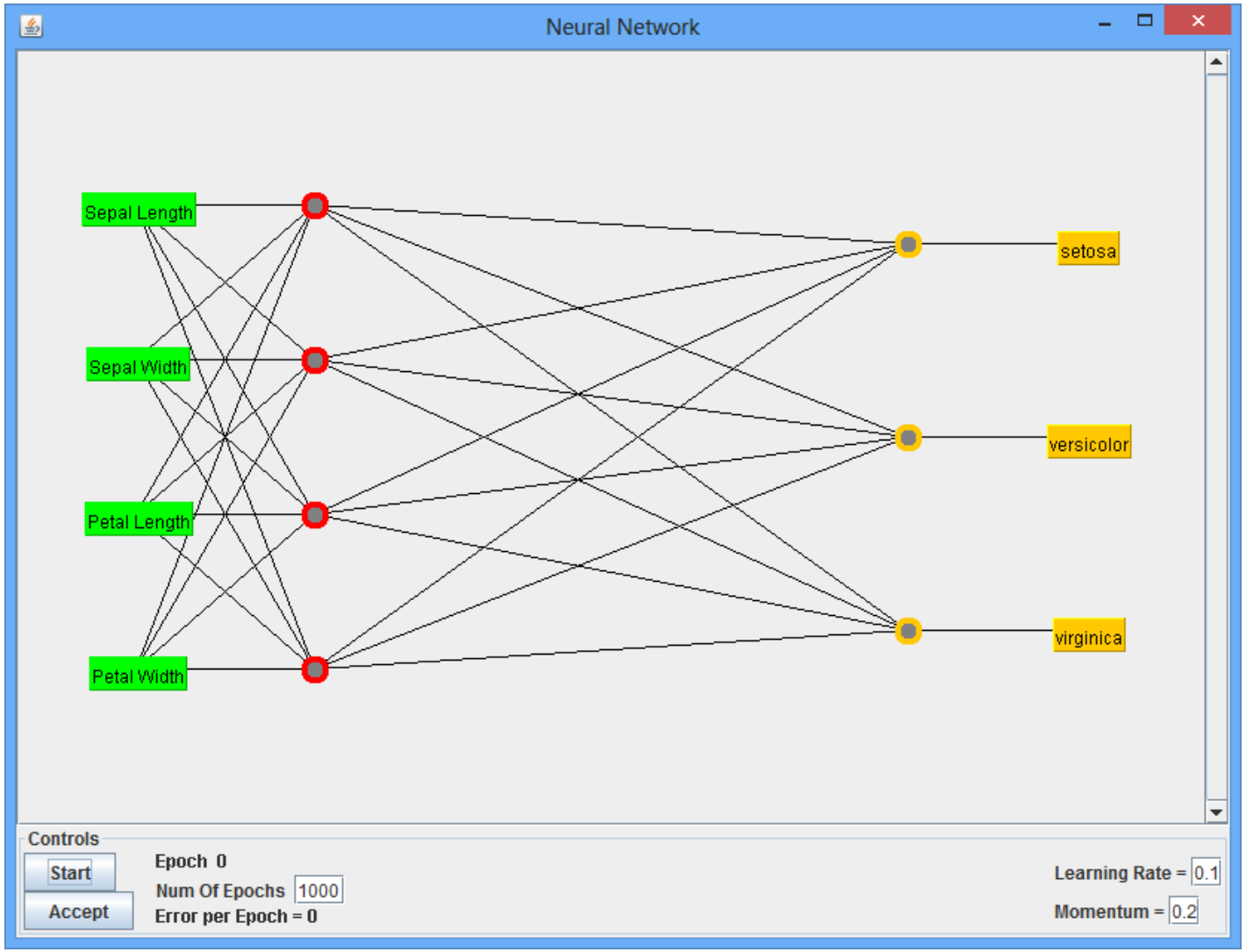


[The sigmoid threshold unit]

Diagram of Small Neural Network



Another Example . . .



Training equations for Perceptrons

Basic Idea:

$$w_i^m = w_i^{m-1} + \propto (y_j - f(x_j))$$

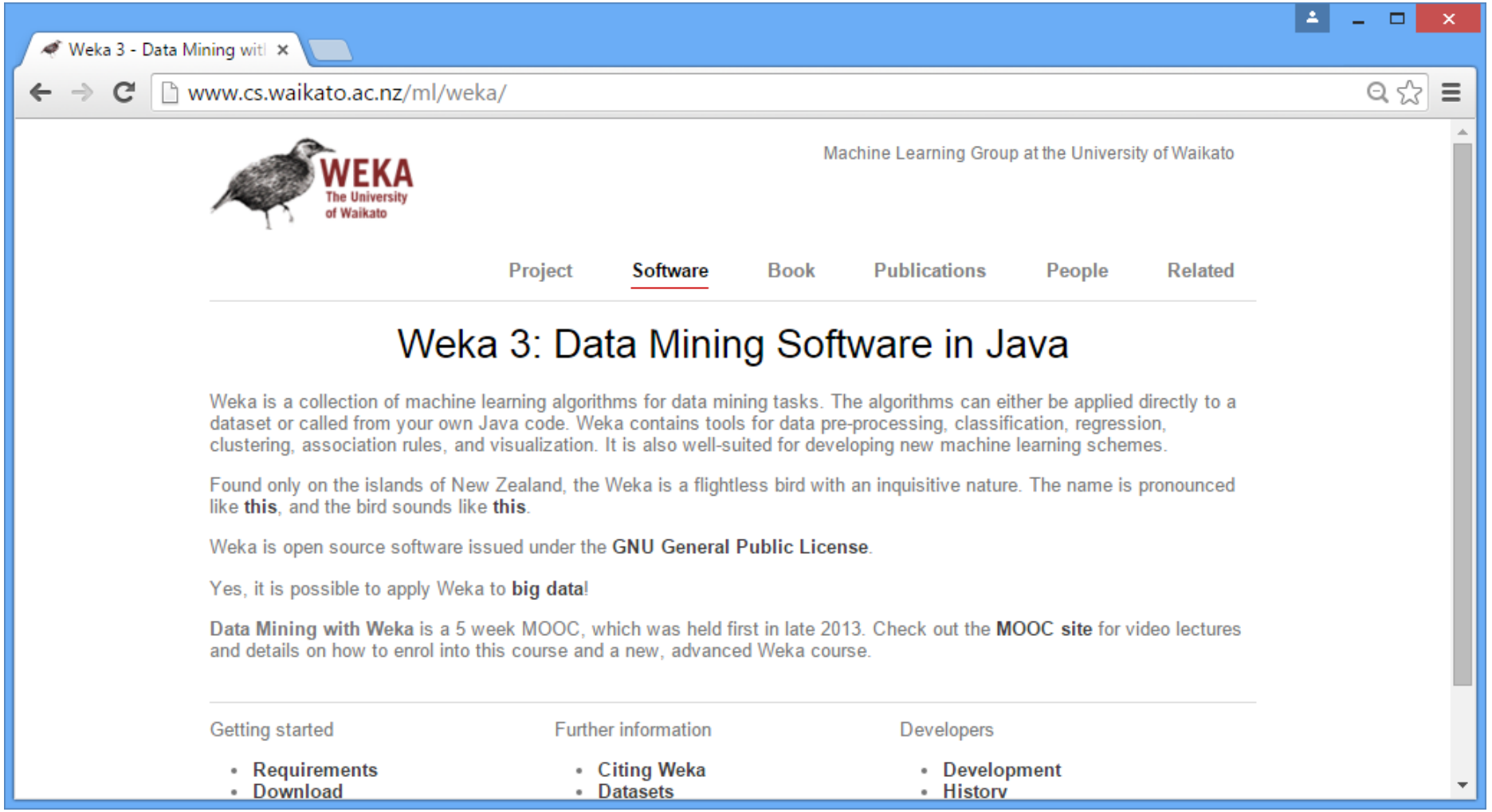
Where:
w -> weight at connection of perception
$\propto$ -> Learning Rate – like the slope of the learning
y -> Expected (correct) output given the training example
f(x) -> Output actual from the neural network

Error = $(y_j - f(x_j))$

So – how does this work???

- Algorithm called ‘Back Propagation’ that runs the training samples through the neural network thousands of times.
  - You will see this later . . .
  - It is like the Chain Rule in Calculus
- Each time (epoch) the sample is run, the weights adjust (converge) towards the ‘correct’ Hypotheses.
- The converged Hypotheses then is the model for the Neural Network.

Tool: WEKA Machine Learning Library



Setup: Machine Learning with Java and WEKA

- Download PS11.zip and extract in your directory
- Open Eclipse and set workspace to PS11
- Create a new Java Project called PS11
- Import the weka.jar file and place on build path
- Create directory “iris_data” in your PS11 Java Project
- Copy the iris.csv file into the “iris_data” directory
- (Detailed Directions at: http://nebomusic.net/perception/PS11.pdf)

Procedure for Establishing ANN in Java / WEKA

- Load Training Data
- Load Testing Data (Optional)
- Set Class Index
- Build Neural Network
  - Set Momentum
  - Set Training Rate
  - Set Hidden Layers
- Train Network
- Test Network / Evaluate Network
- Create System using Learner (after training and testing)