Exercises

2. Return to the examples from Table 4.2. Hand-simulate the perceptron learning algorithm's procedure, starting from a different initial set of weights than the one used in the table. Try also a different learning rate.

Give It Some Thought

2. Explain in what way it is true that the 1-NN classifier applied to a pair of examples (one positive, the other negative) in a plane defines a linear classifier. Invent a machine learning algorithm that uses this observation for the induction of linear classifiers. Generalize the procedure to n-dimensional domains.

Computer Assignments

2. Create a training set consisting of 20 examples described by five binary attributes, $x_1, \ldots, x_5$. Examples in which at least three attributes have values $x_i = 1$ are labeled as positive, all other examples are labeled as negative. Using this training set as input, induce a linear classifier using perceptron learning. Experiment with different values of the learning rate, η. Plot a function where the horizontal axis represents η, and the vertical axis represents the number of example-presentations needed for the classifier to correctly classify all training examples. Discuss the results.

3. Use the same domain as in the previous assignment (five boolean attributes, and the same definition of the positive class). Add to each example N additional boolean attributes whose values are determined by a random-number generator. Vary N from 1 to 20. Observe how the number of example-presentations needed to achieve the zero error rate depends on N.