CSE 474
Pattern Recognition Sessional

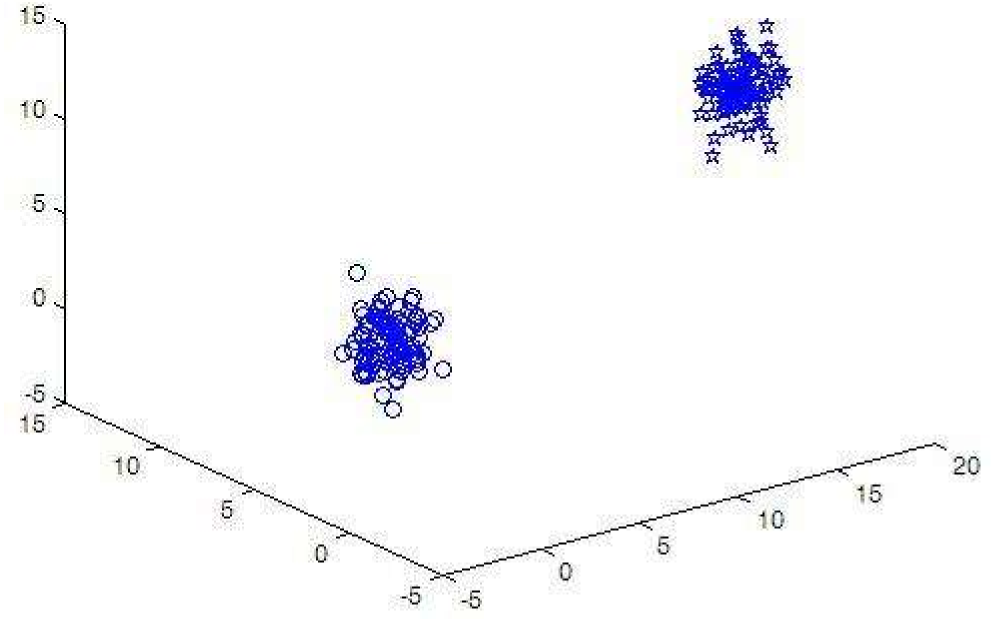
Lab# 2: Implementation of Perceptron and its variants

Lab Objective

- Implement 2-class classifiers using
  - Basic perceptron algorithm
  - Reward and punishment (RP) algorithm
  - Pocket algorithm

Training Data

- Assume the following training set
  - 2 classes
  - 3 features



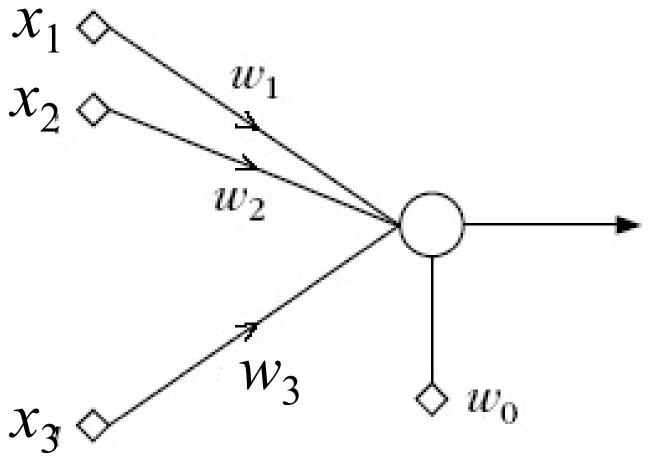
What to do

- Assume the following training set
  - All numerical data
  - Features are real numbers
  - Classes are integers

F1	F2	F2	Class
9.4512	7.3199	6.4664	1
10.7276	9.6067	5.9398	2
10.1960	9.3145	8.3873	1
15.7777	1.5879	11.4440	1
15.8685	2.7902	11.2532	1
14.9448	0.7798	12.7481	2

What to do

- Use the training set to learn a perceptron for each of the 3 variants (e.g., *basic*, *RP* and *Pocket*)



Perceptron

Training Set

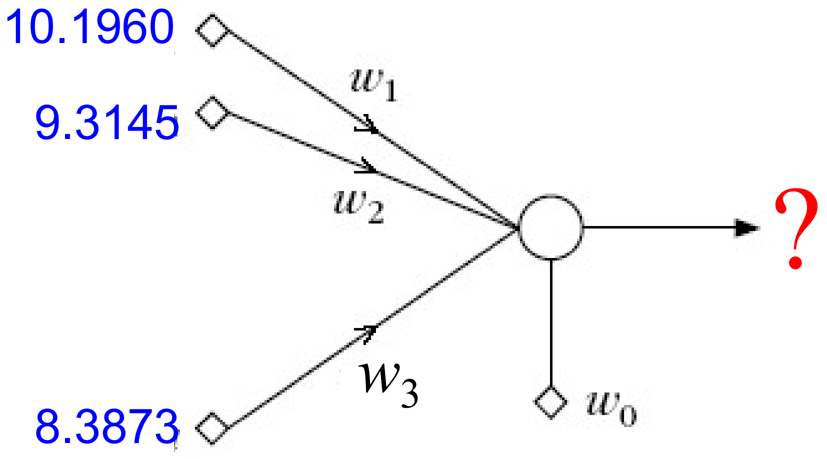
F1	F2	F2	Class
9.4512	7.3199	6.4664	1
10.7276	9.6067	5.9398	2
10.1960	9.3145	8.3873	1
15.7777	1.5879	11.4440	1
15.8685	2.7902	11.2532	1
14.9448	0.7798	12.7481	2

What to do

- Given an unknown sample,

  $[x_1, x_2, x_3]$ = [10.1960 9.3145 8.3873]

Predict its class!



Training Files

- 2 Training Files
  - trainLinearlyNonSeparable.txt *for pocket algorithm*
  - trainLinearlySeparable.txt *for other 2 algorithms*
  - File format:
    - first line contains 3 integers *d m n*. *d*= no. of features, *m*= no. of classes, *n*=no. of samples
    - Each of next *n* lines contains *d*+1 numbers: *d* feature values followed by its class in integer

Testing Files

- 2 Testing files
  - testLinearlyNonSeparable.txt *for pocket algorithm*
    testLinearlySeparable.txt *for other 2 algorithms*
  - Format is identical but the first line is removed

Output

- For each of the 3 algorithms
  - Learn the perceptron using corresponding training file
  - Use the corresponding testing file to identify all misclassified samples and report as follows

sample no.   feature values   actual class   predicted class

  - % of accuracy

Other information

- Your program must be able to handle variable no. of features. Hard coded feature representation will NOT be accepted.
- Submission deadline is 26/09/2020 at 9 am
- Sample training and testing files are available in the moodle
- Follow the algorithms and notations of your text book (e.g., *Pattern Recognition* by S. Theodoridis)
- You can use your own data to judge your code
- Different files will be used during evaluation