Design of a Multilayer Feedforward Neural Network Classifier and Approximator

version 28 March 2022

ECE 5730 Foundations of Neural Networks

Do not use any references to complete this project other than Haykin [1] and the course text(s). You must write your own code; use of code libraries e.g. MATLAB™ toolboxes are not allowed.

Part 1: Classifier Design (this problem is based on computer experiment 6.9 in [1])

1. You will design a MFNN classifier to classify a two-dimensional feature vector. There are two classes, with probability density functions
   a. CLASS 1: $\mathrm{p}(\boldsymbol{x}|C_1) = \frac{1}{2\pi\sigma_1^2}\mathrm{e}^{-\frac{1}{2\sigma_1^2}||\boldsymbol{x}-\boldsymbol{\mu}_1||^2}$

      $\boldsymbol{\mu}_1 = \begin{bmatrix}1\\1\end{bmatrix}$

      $\sigma_1^2 = 1$

   b. CLASS 2: $\mathrm{p}(\boldsymbol{x}|C_2) = \frac{1}{2\pi\sigma_2^2}\mathrm{e}^{-\frac{1}{2\sigma_2^2}||\boldsymbol{x}-\boldsymbol{\mu}_2||^2}$

      $\boldsymbol{\mu}_2 = \begin{bmatrix}-1\\-1\end{bmatrix}$

      $\sigma_1^2 = 4$
2. Generate 500 samples from each class. Plot the samples in the plane, using different colors for class 1 and class 2.
3. The *optimal* classifier classifies any **x** within a circle of radius 1.9227 centered at (5/3, 5/3) as class 1. Any **x** outside this region is considered to be class 2. Estimate the number of equal samples of each class needed to find the probability of miscalculation to within 1%.
4. Design a multilayer feedforward neural network (MFNN) classifier using this training set by generating a training set consisting of equal numbers of samples from class 1 and 2. You may generate an additional 100 points for class 1 and class 2 to test for overfitting during training. You must use your own code to implement the Error Backpropagation (EBP) algorithm. *You are urged to first check the accuracy of your EBP code by checking your results for a known less complex problem.*
5. Test your design using a test set consisting of an equal number of class 1 and class 2 examples not used in the training or validation set. The number of examples is determined from step 3.

Part 2: Approximator Design (similar examples are found in [2])

1. Download the following ASCII files from the online ECE 5730 syllabus approx1t.dat and approx1v.dat.
2. The file approx1t.dat contains examples of a one-dimensional mapping $\{t, x(t)\}$. Use these examples to design a MFNN approximator.

3. Test your design using approx1v.dat. **YOU MAY NOT USE THIS DATA TO UPDATE NETWORK WEIGHTS**; you may use this data to terminate training.
4. **EXTRA CREDIT:** Design an electronic circuit to implement your approximator design utilizing operational amplifiers and passive components. Test your circuit using a circuit simulator. Compare the actual and desired performance of the circuit.

Maximum report length is 6 pages using the syllabus prescribed format. Attach your code as an appendix to the report. **No collaboration is allowed on this project – complete this project on your own.**

References

[1] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, 1st edition, 1994.

[2] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.