Part 1a – Backpropagation Learning: Instructions For this part of the assignment you are to implement a multi-layer perceptron and train it using the error-backpropagation algorithm. You will not yet need to use Robocode, however you should keep in mind that your neural net will later be used in your robot tank. It will be useful if your neural network software is able to accept as a parameter the number of hidden neurons. To help, I have provided Java interfaces for you to start with: https://courses.ece.ubc.ca/592/PDFfiles/NeuralNetInterface.java.pdf https://courses.ece.ubc.ca/592/PDFfiles/CommonInterface.java.pdf Now follow the instructions below: Submission instructions 1) Set up your network in a 2-input, 4-hidden and 1-output configuration. Apply the XOR training set. Initialize weights to random values in the range -0.5 to +0.5 and set the learning rate to 0.2 with momentum at 0.0.

a) Define your XOR problem using a binary representation. Draw a graph of total error against number of epochs. On average, how many epochs does it take to reach a total error of less than 0.05? You should perform many trials to get your results, although you don’t need to plot them all. b) This time use a bipolar representation. Again, graph your results to show the total error varying against number of epochs. On average, how many epochs to reach a total error of less than 0.05? c) Now set the momentum to 0.9. What does the graph look like now and how fast can 0.05 be reached? Your submission should be a brief document clearly showing the graphs requested about. Please number your graphs as above and also include in your report an appendix section containing your source code