CPEN 502 Assignment Part 1a – Backpropagation Learning

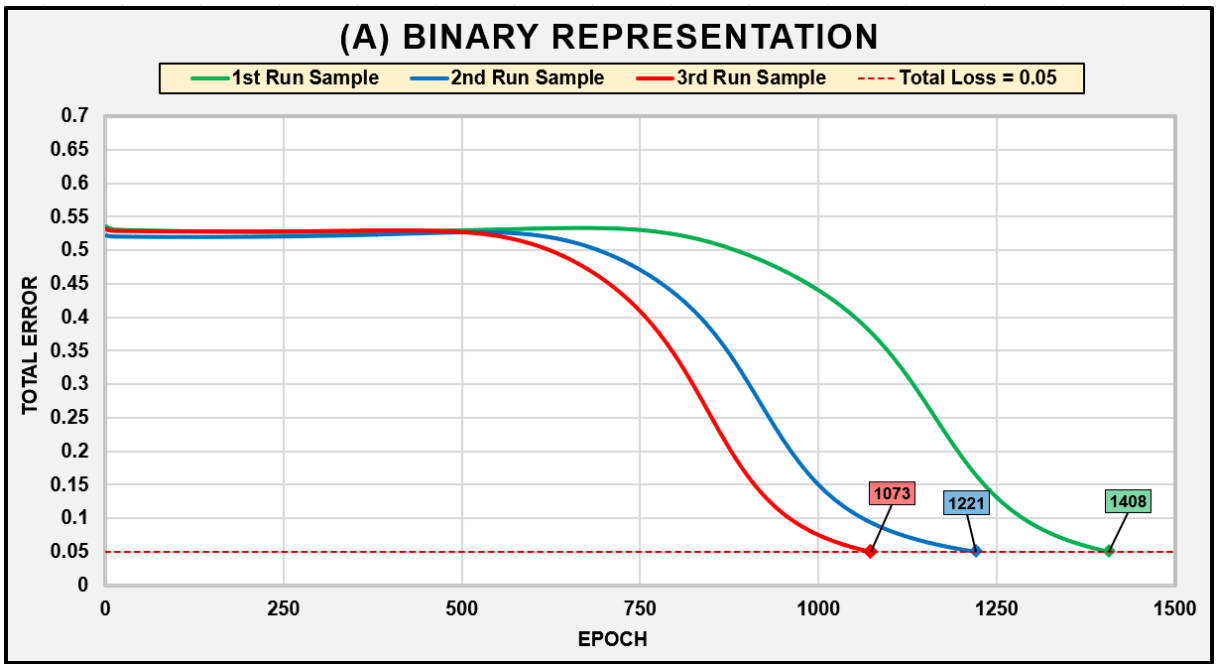
Student Name: Chao-Wu Chu
Student Number: 85406312

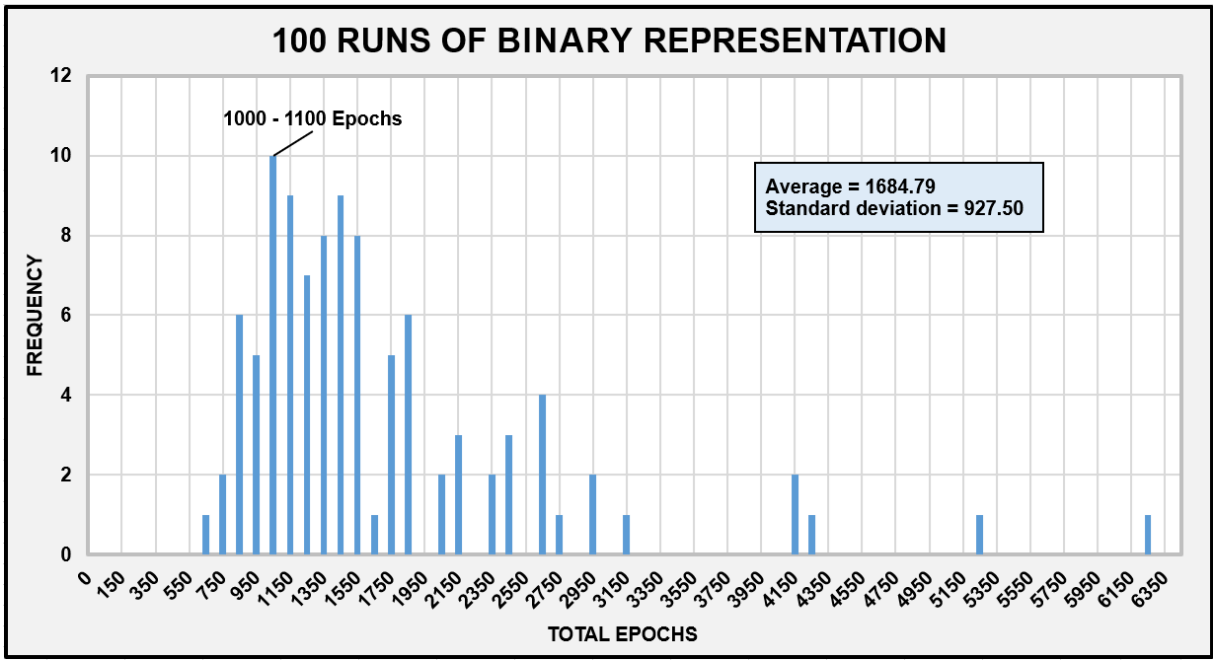
(1) Set up your network in a 2-input, 4-hidden and 1-output configuration. Apply the XOR training set. Initialize weights to random values in the range -0.5 to +0.5 and set the learning rate to 0.2 with momentum at 0.0.

Your submission should be a brief document clearly showing the graphs requested. Please number your graphs as above and also include in your report an appendix section containing your source code.

(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.

Answer:



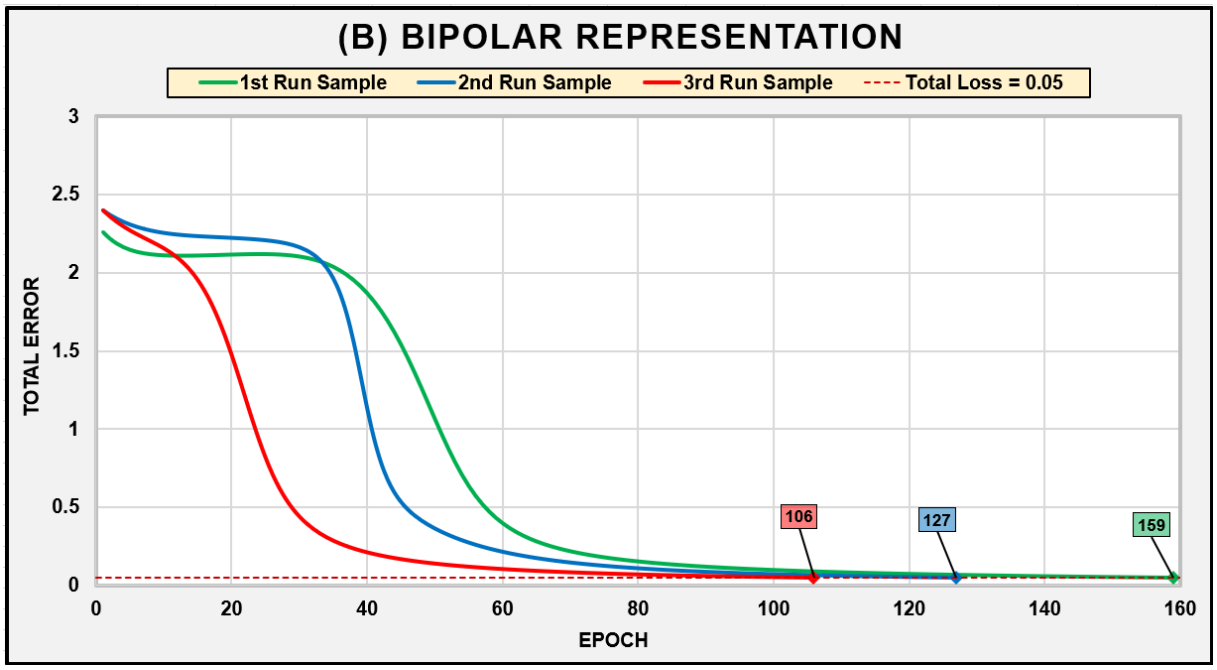


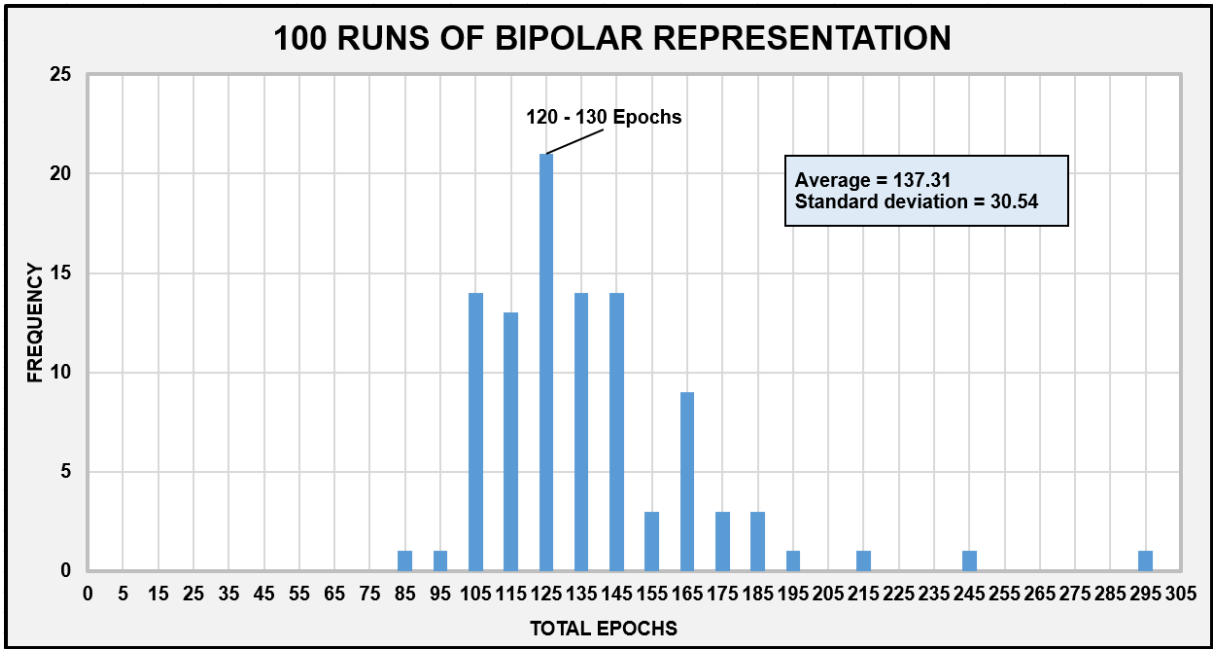
For binary representation, I picked 3 of the most representative datasets to plot. The green, blue, and red curves reached a total loss of less than 0.05 at 1408, 1221, and 1073 epochs respectively. On average with 100 runs, the average is 1684.79 epochs to reach a total loss of less than 0.05, and the standard deviation is 927.50.

For the graphical section, the total loss only begins to show a significant decrease when it reaches about half of the total epochs. During the decline, the degree of decrease is relatively gentle, and the trend is S-shaped (the slope increases to its maximum before starting to decrease). The whole training process would take more epochs (or time) to reach our target.

(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?

Answer:





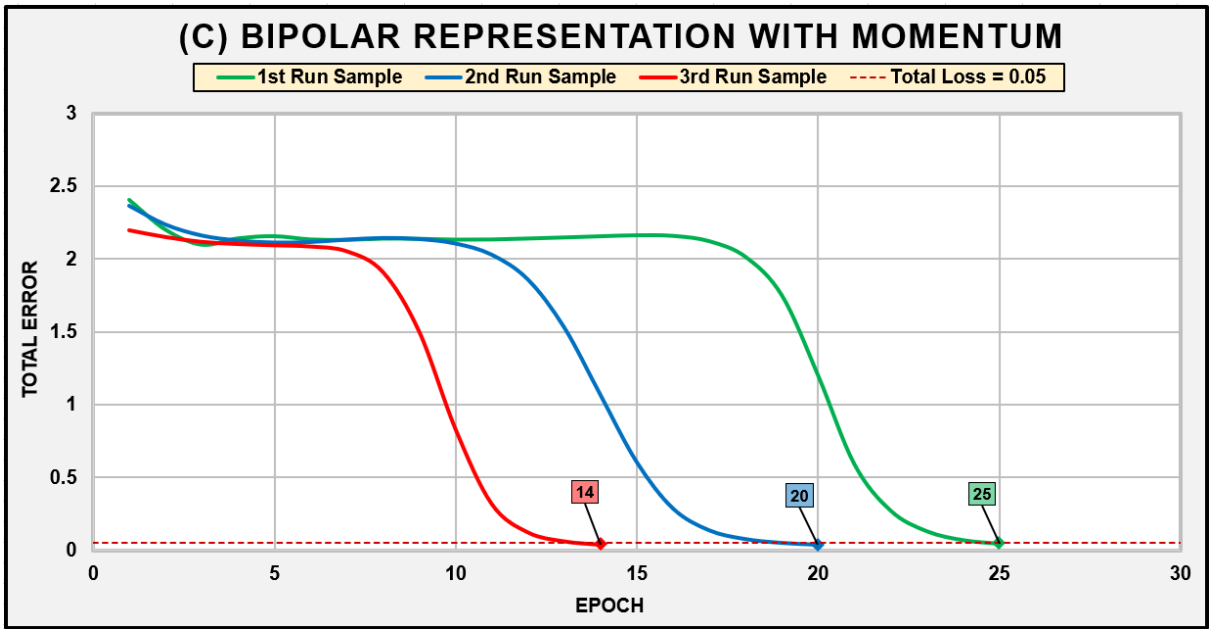
For bipolar representation (without momentum), I picked 3 of the most representative datasets to plot. The green, blue, and red curves reached a total loss of less than 0.05 at 159, 127, and 106 epochs respectively. On average with 100 runs, the average is 137.31 epochs to reach a total loss of less than 0.05, and the standard deviation is 30.54.

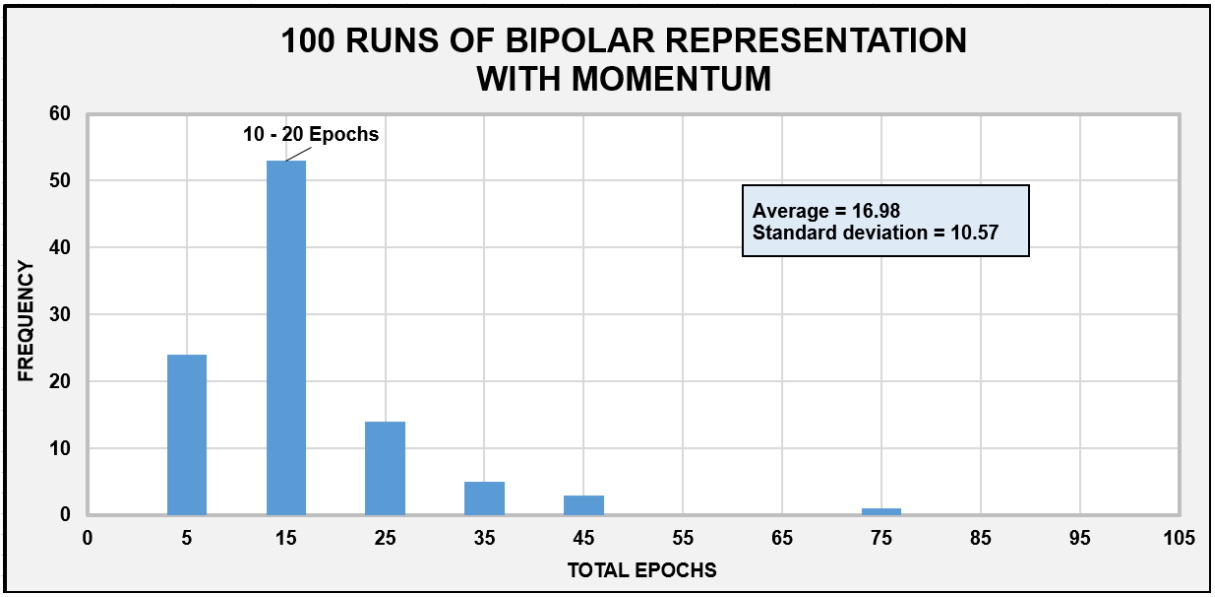
In the graphical section, the total loss already begins to show a significant decrease when it reaches about 1/4 of the total epochs. During the decline, the trend of the graph is also S-shaped, but compared to the binary representation, the slope for bipolar is higher, allowing it to reach the target value within the acceptable total error faster (one-order faster).

Bipolar representation (+1 and -1) compared to binary representation (0 and 1), when provided with the same input values, can produce higher output values in its differential equation. In each epoch, each neuron can change the weights by a larger magnitude, accelerating the training process.

(c) Now set the momentum to 0.9. What does the graph look like now and how fast can 0.05 be reached?

Answer:





For bipolar representation with momentum, I picked 3 of the most representative datasets to plot. The green, blue, and red curves reached a total loss of less than 0.05 at 25, 20, and 14 epochs respectively. On average with 100 runs, the average is 16.98 epochs to reach a total loss of less than 0.05, and the standard deviation is 10.57.

In the graphical section, the total loss also begins to noticeably decrease around 1/4 of the total epochs. During the decline, the graphical trend is similarly S-shaped, but compared to the bipolar representation without momentum, the slope for bipolar with momentum is higher, allowing it to reach the target value within the acceptable total error faster (one-order faster).

When the momentum parameter is introduced, each weight changes more rapidly due to the inertial influence of the changing gradient, significantly accelerating the overall training process.