Armando Carrasquillo

2056125

**Individual Project 3**

**Introduction:**

Three separate methods were used to train a set of weights and bias (W1, W2, b1, and b2) with a set number of hidden layers, a threshold (stopping criterion) an alpha(adaptation rate), and in part III a gamma. The results from the training includes a mean squared error and evolution of 5 of the weights/biases for each epoch displayed on a line graph. These trainings also have a max of 3000 epochs. These weights were trained with the targets (TTEKG) and the patterns (PPEKG) provided.

**Part I: “Basic” Backpropagation**

To run the “Basic” Backpropagation, weights need to be created with a set number of hidden layers. The code snippet below creates weights.

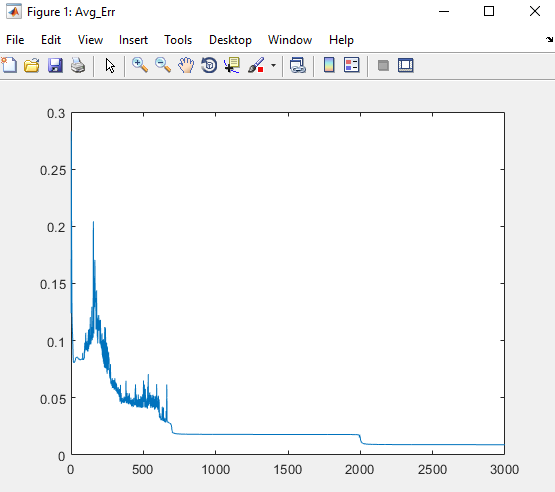
W1 = 0.25 \* randn(10,47); W2 = 0.25 \* randn(1,10);

b1 = 0.25 \* randn(10,1); b2 = 0.25 \* randn(1,1); alp = 0.01;

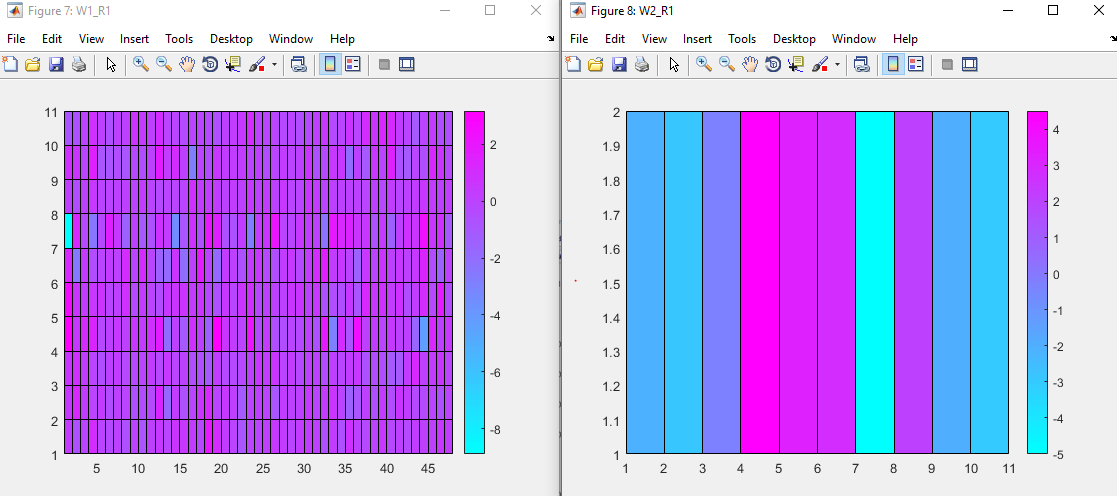
With the randomly generated weights/bias, running the function “tester” with the target TTEKG, and the patterns PPEKG would begin the backpropagation training. To get the hit rates, the function hittable needs to be ran.

Threshold: 0.0001, Hidden Layers: 10, Alpha: 0.01

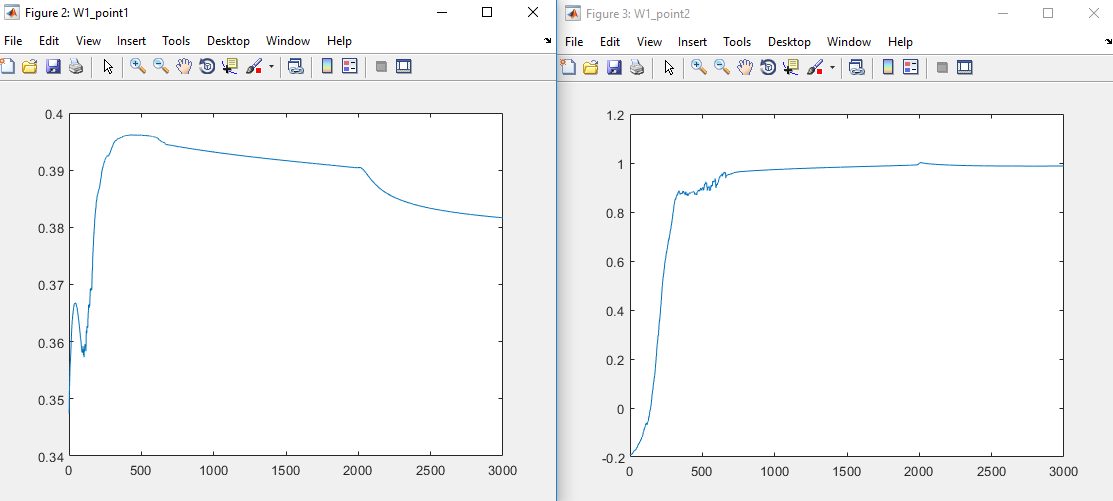
Avg Error:

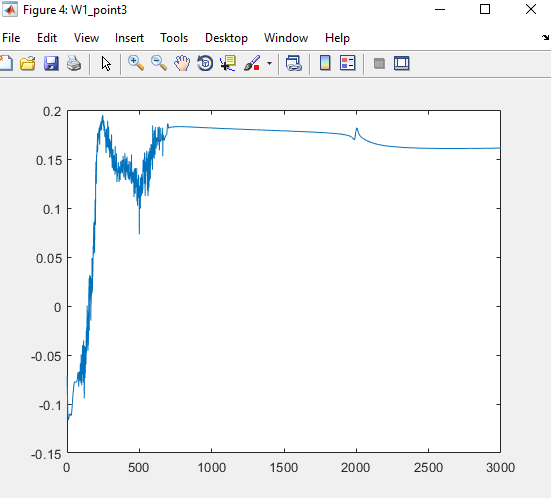


Trained Weights: W1 & W2

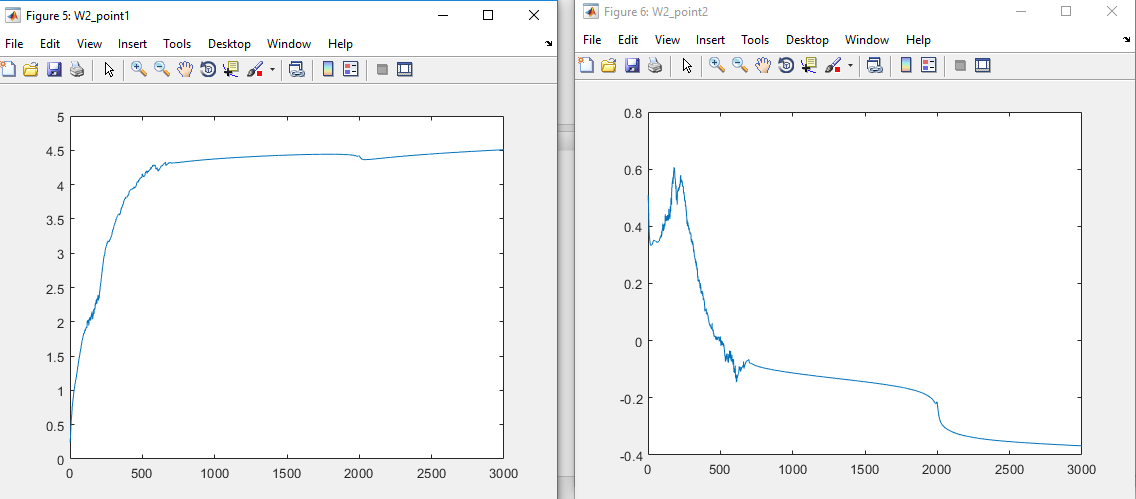


Evaluations from W1:

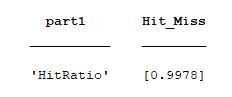




Evaluations for W2:



**Hit Ratio:**

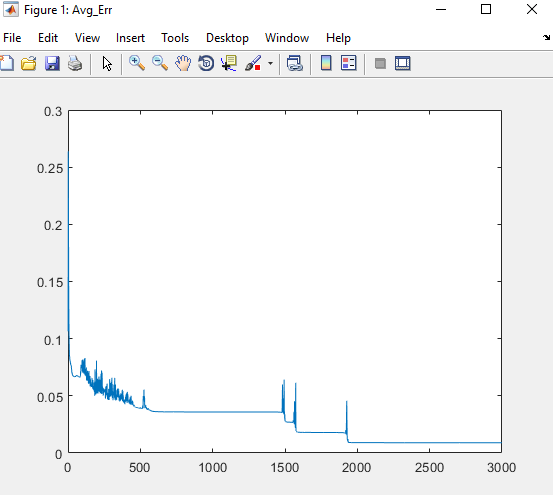


**Part II: Backpropagation with Nguyen-Widrow Initialization**

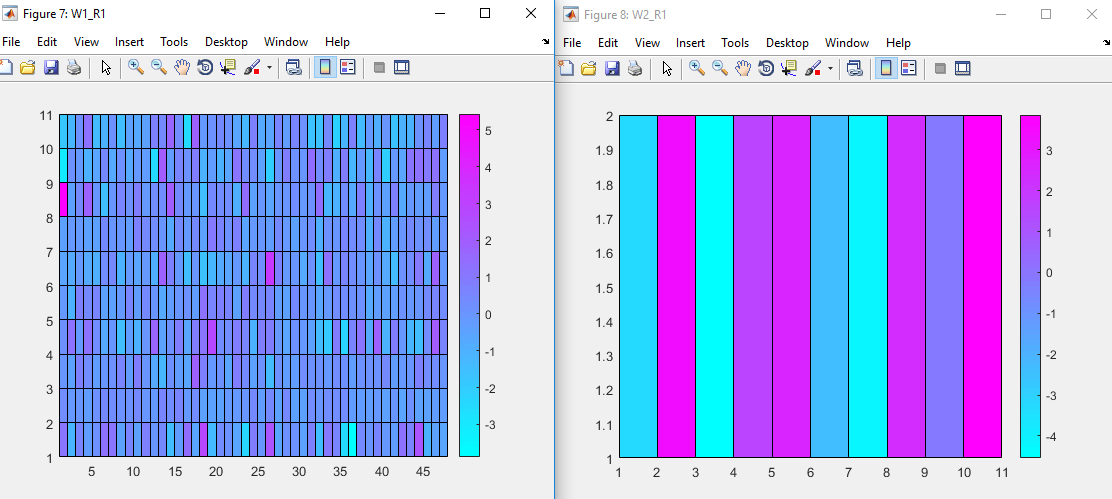
To run this test, the user needs to create weights by running the function nguyen(). With the created weights, PPEKG, and TTEKG the function tester needs to be run for the charts, and hittable for the hit ratio.

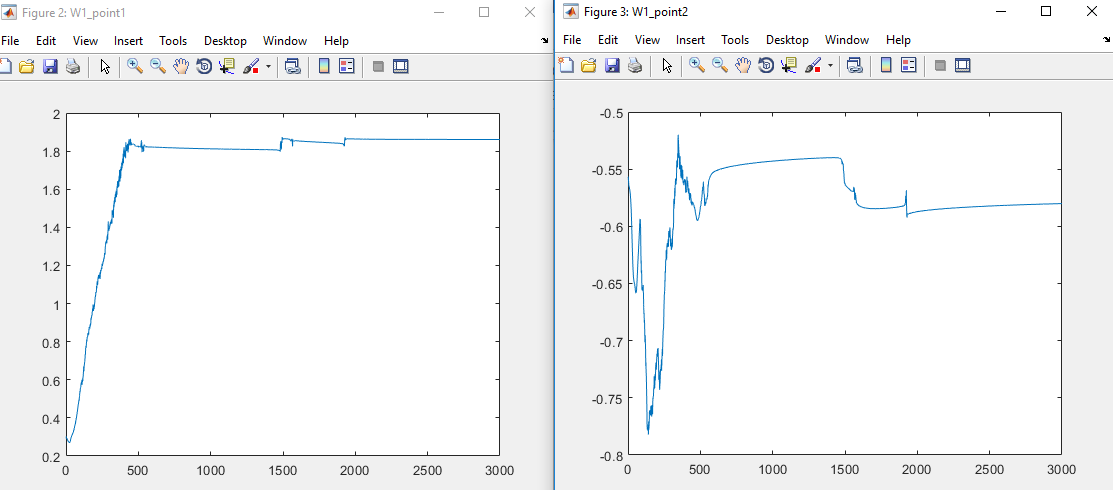
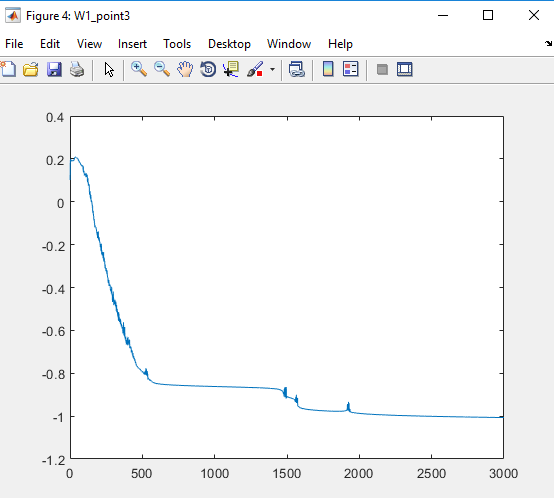
Threshold: 0.0001, Hidden Layers: 10, Alpha: 0.01 gamma = 0.9

Avg Error:

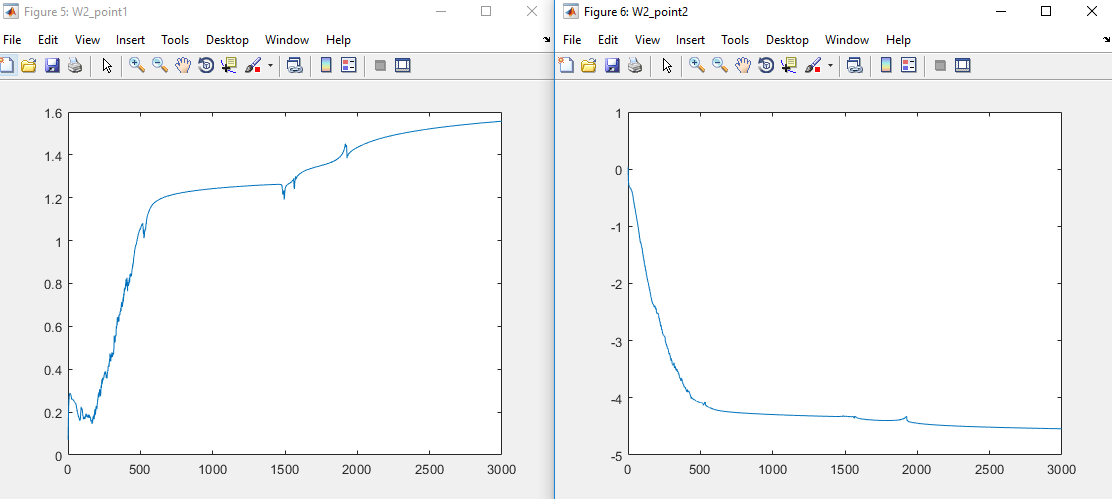


Trained Weights: W1 & W2

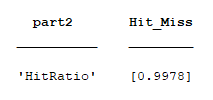


Evaluations from W1: 

Evaluations for W2:



**Hit Ratio:**

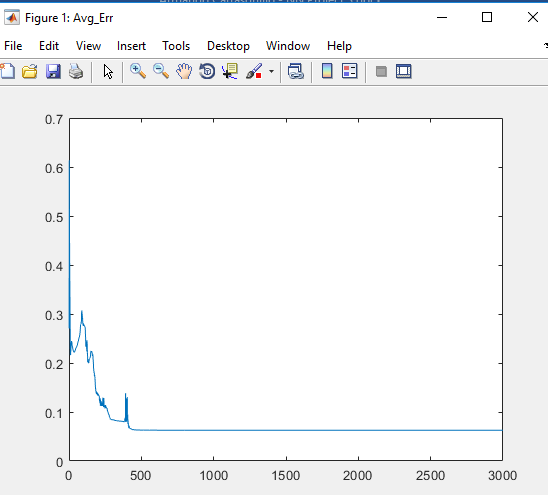


**Part III: Backpropagation with Momentum**

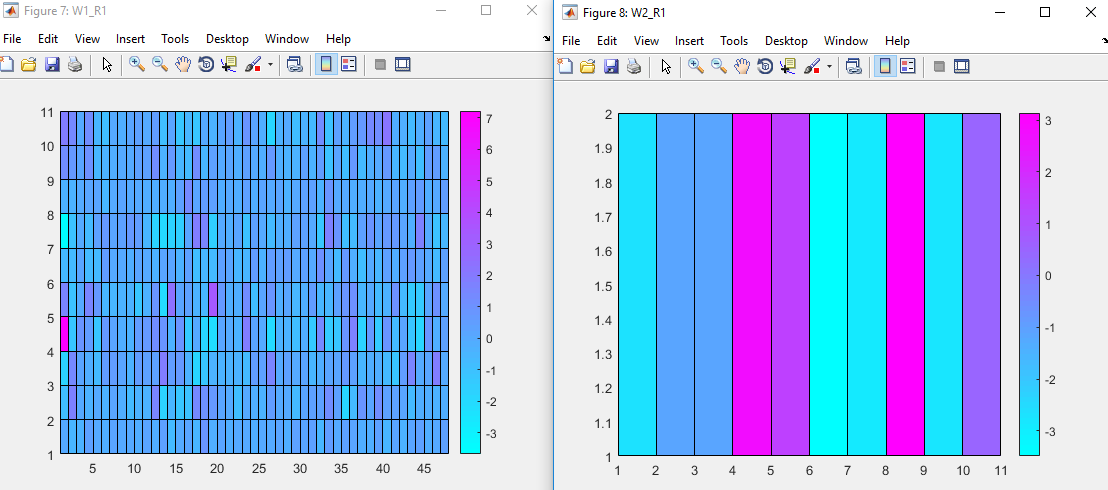
To run the Backpropagation with momentum, weights need to be created with a set number of hidden layers. The code snippet from part 1 can create these weights. With the randomly generated weights/bias, running the function “Mtester” with the target TTEKG, and the patterns PPEKG would begin the backpropagation training. To get the hit rates, the function hittable needs to be ran.

Threshold: 0.0001, Hidden Layers: 10, Alpha: 0.01 gamma = 0.9

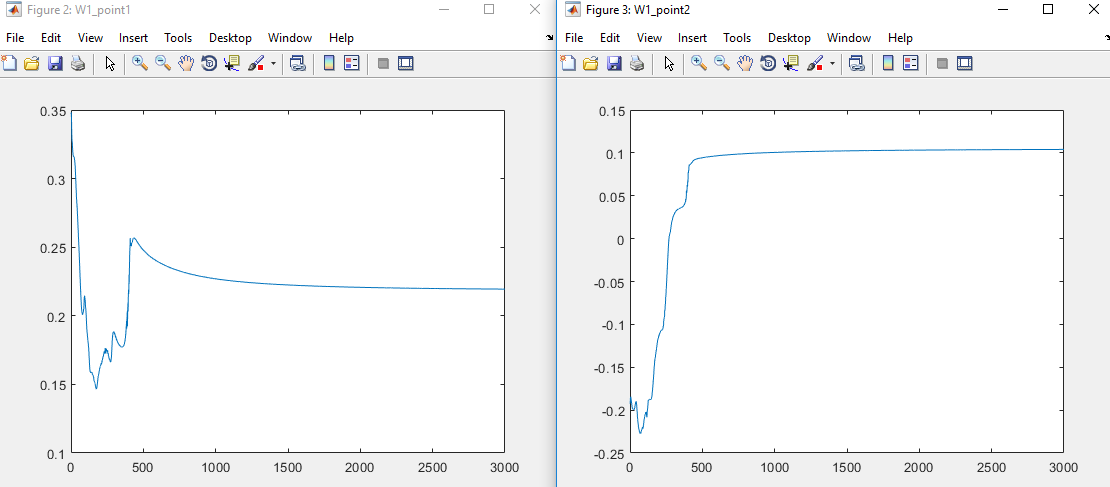
Avg Error:

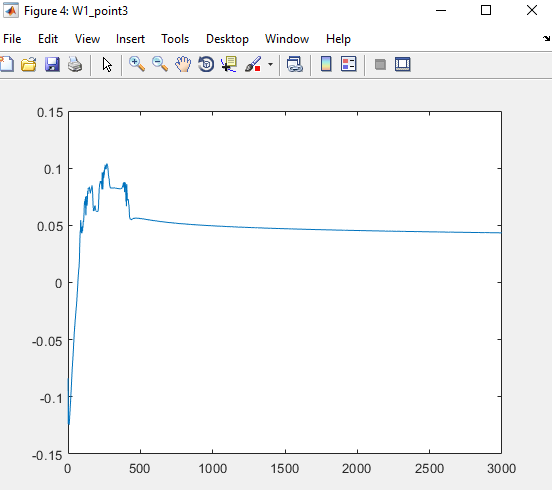


Trained Weights: W1 & W2

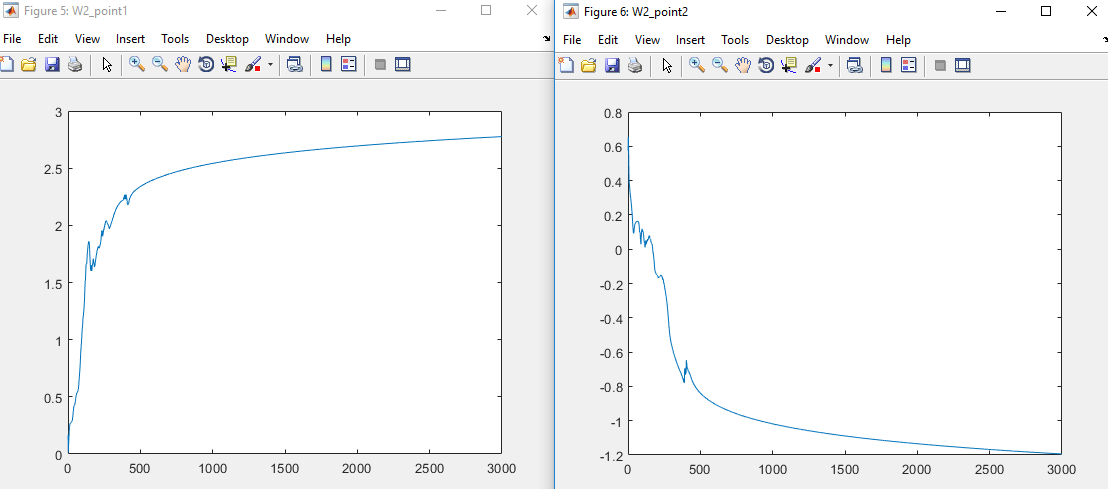


Evaluations from W1:

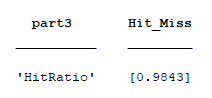




Evaluations for W2:



**Hit Ratio:**



**Part IV: Conclusions**

The three methods of backpropagation performed well. Basic backpropagation. Based on the recorded average squared errors, Backpropagation with Momentum (part 3) performed the training session with epochs 500, followed by Nguyen-Widrow(part 2) with 1900 epochs, and lastly the “basic” backpropagation(part 1)with over 2000 epochs. These methods were different versions of how to do backpropagation. All sets were done with an alpha of .01, 10 hidden layers, and for part 3, a gamma of 0.9.

Part 1 was using the basic backpropagation which resulted with a hit rate of 0.9978. This method is the one that the others should beat during the tests. The average squared error normalized after 2000 epochs, but not all the weights plateaued.

Part 2 used the Nguyen-Widrow method which had weights and bias that were constructed between -0.5 and .5. Along with that, the weights needed to follow a formula that includes calculating the magnitude of the rows and applying it in order to set them apart from the other method’s way of having initial weights (which was that they were randomly generated numbers). The hit ratio achieved in part 2 was 0.9978 which tied with the hit ratio achieved in part 1, but it did this with less epochs.

Part 3 deal using Backpropagation with Momentum, the goal for this type of backpropagation is to train the weights faster by using a delta Weight that is every epoch and is used in the following epoch to complete the training. This delta carries over and pushes the training to go even further.