

# NML502 HW #3

*Elliot Smith, Eugen Hruska, Warum Suriyanarayana*

*1/31/2018*

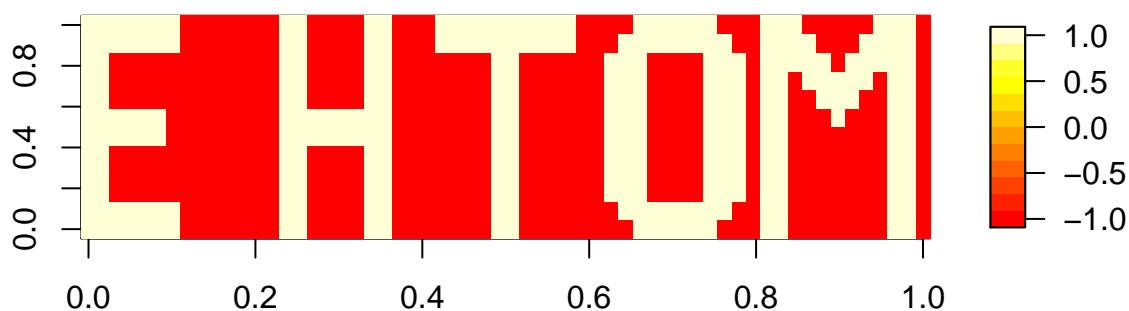
## Problem 3

### Problem 3a

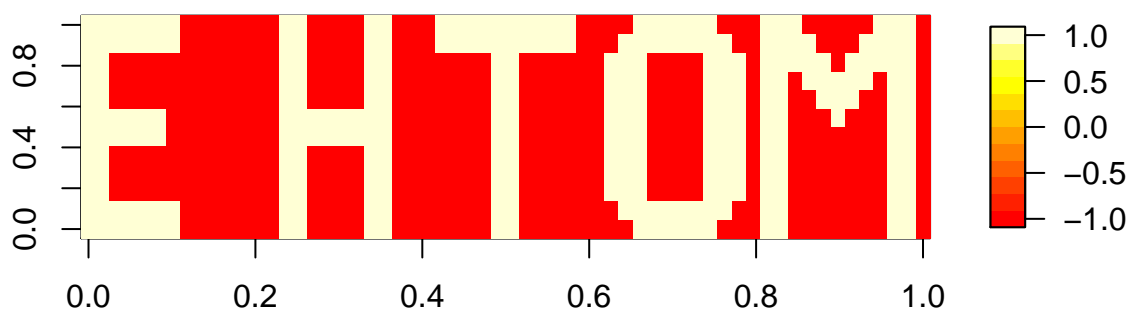
#### Recall Accuracy

- Learning Rate: 0.001
- Max Number of Learning Steps: 1000
- Error Threshold: 0.0000001
- Error Measure: As defined in errcorr.m

#### Original Input and Desired Output



#### Memory Recall



## Summary

Please see above for our parameter selections. Neither our data nor memory matrix in this iteration were corrupted in any way. Our maximum number of learning steps is 1000, however, if we reached our error tolerance of 0.0000001 then we would immediately terminate the learning process and return the memory matrix. Our learning rate acts as a factor by which to influence the change of weights, we felt that 0.001 was

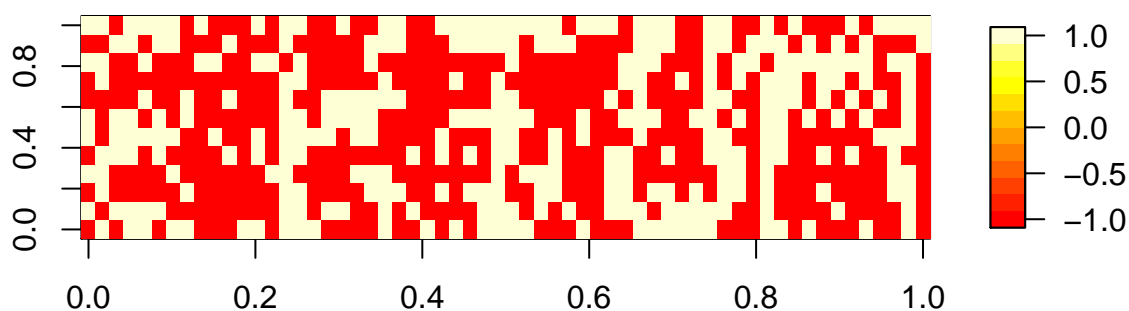
appropriate for us. There was no provided, thresholded, recalled image in this case because our recall was perfect.

## Problem 3b

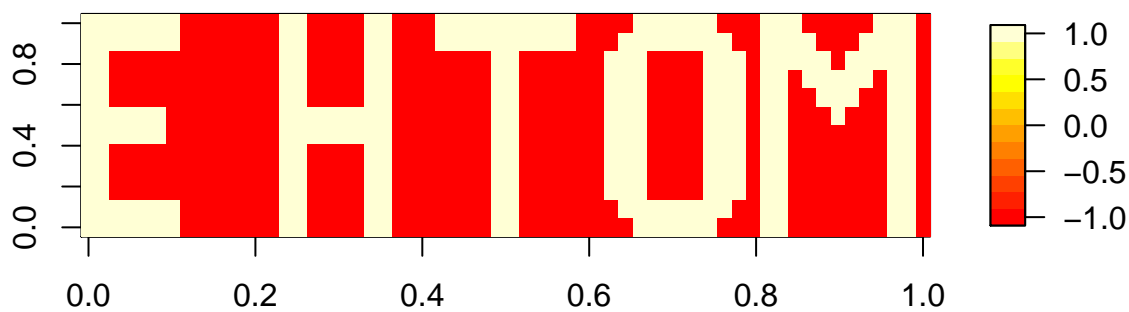
### 25% Input Corruption Recall Accuracy

- Learning Rate: 0.001
- Max Number of Learning Steps: 1000
- Error Threshold: 0.0000001
- Error Measure: As defined in errcorr.m

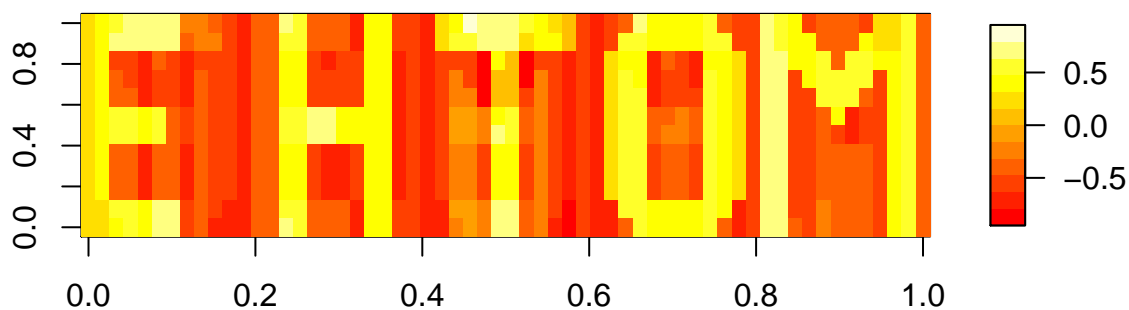
Original Input



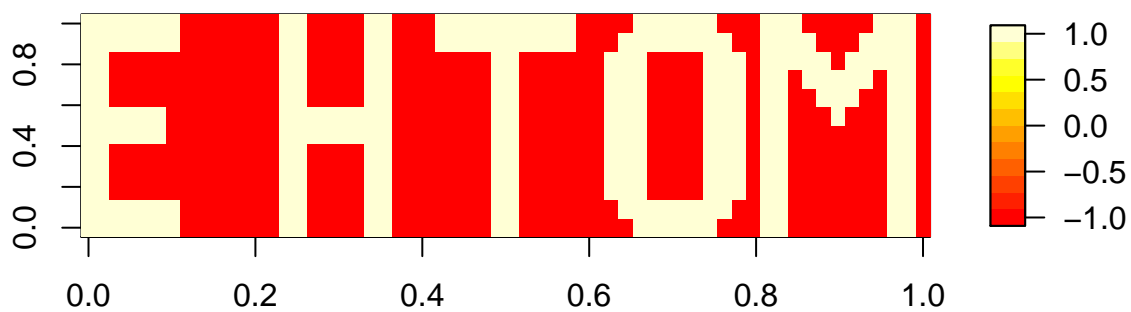
Desired Output



Memory Recall



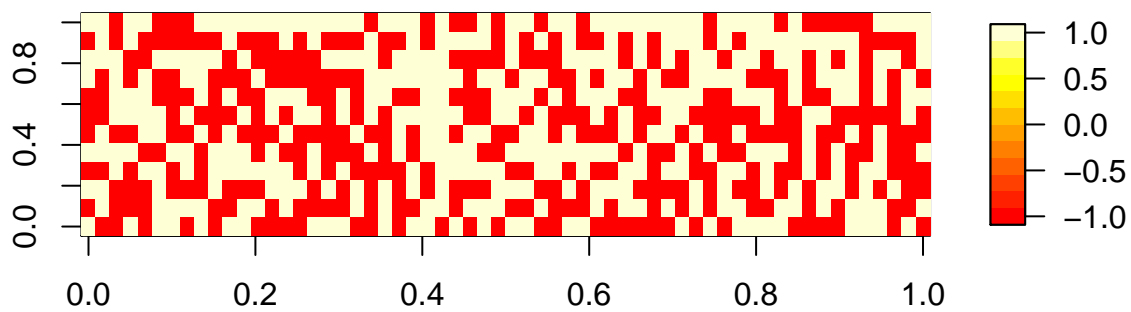
Thresholded Memory Recall



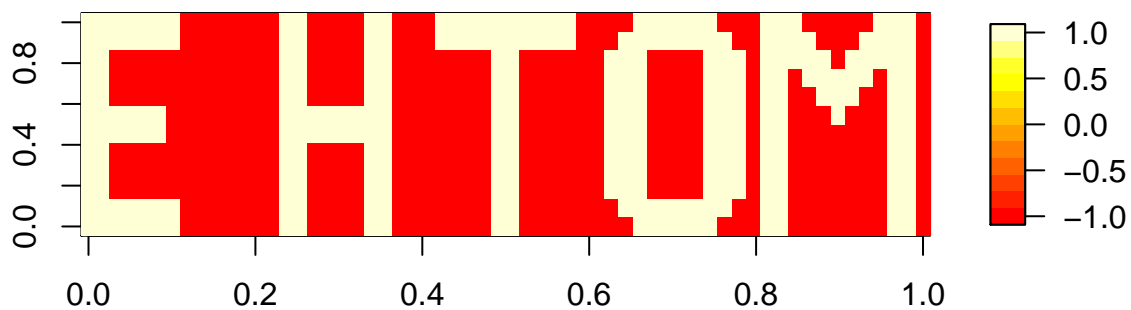
50% Input Corruption Recall Accuracy

- Learning Rate: 0.001
- Max Number of Learning Steps: 1000
- Error Threshold: 0.0000001
- Error Measure: As defined in errcorr.m

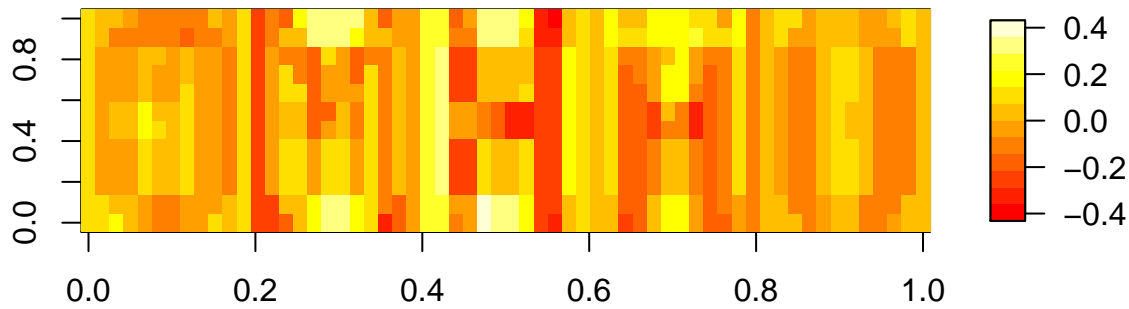
Original Input



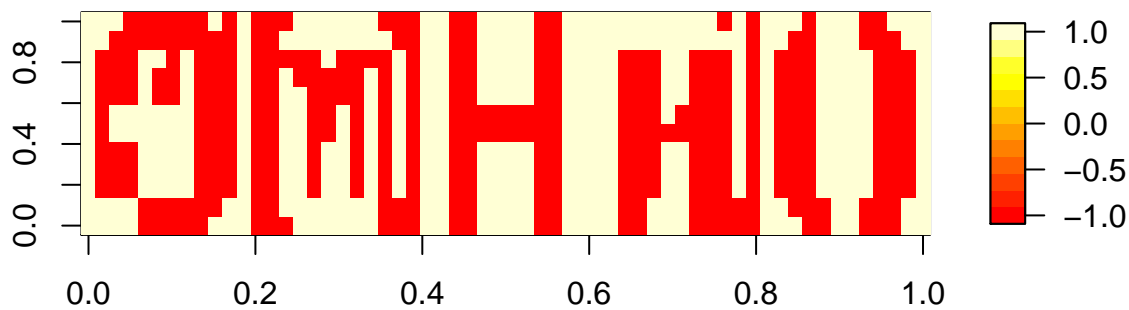
Desired Output



### Memory Recall



### Thresholded Memory Recall



### Summary

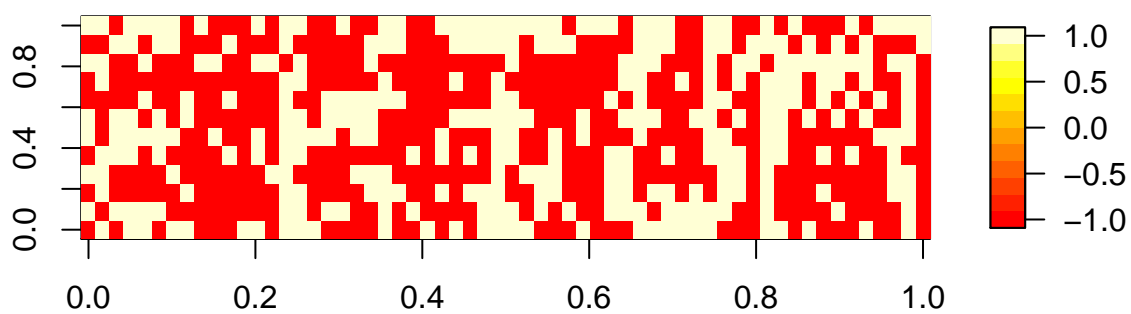
Please see above for our parameter selections. In our first iteration in this section, we randomly select 20% of the input values for each pattern and corrupted them by switching their signs; in the second iteration we increased that number to 50%. Our maximum number of learning steps is 1000, however, if we reached our error tolerance of 0.0000001 then we would immediately terminate the learning process and return the memory matrix. Our learning rate acts as a factor by which to influence the change of weights, we felt that 0.001 was appropriate for us. Our threshold methodology was to set the value to 1 if a value was greater or equal to 0 and to change it to -1 in all other cases. As can be seen by our recalled image and recalled thresholded image of the 20% corruption, we had great results overall and the image can still be interpreted very clearly. However, our 50% corruption did not have great results trying to reproduce the desired output, we can see some what appears to be some of our letters, but it is quite vague.

## Problem 3c

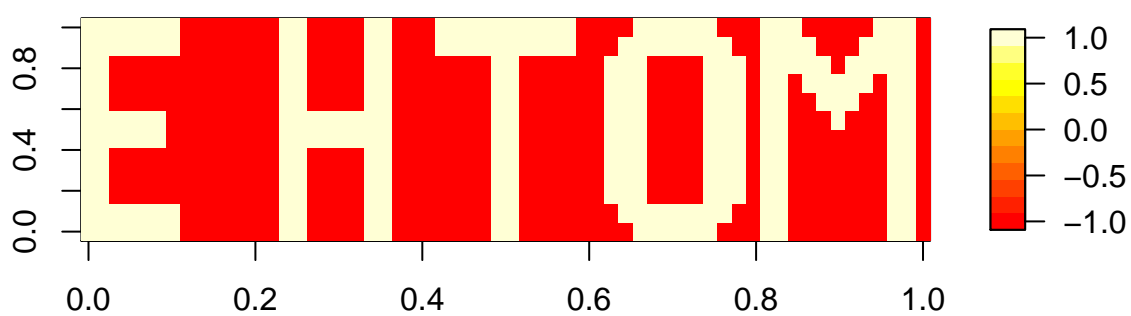
### Corrupted Memory Recall Accuracy

- Learning Rate: 0.001
- Max Number of Learning Steps: 1000
- Error Threshold: 0.0000001
- Error Measure: As defined in errcorr.m

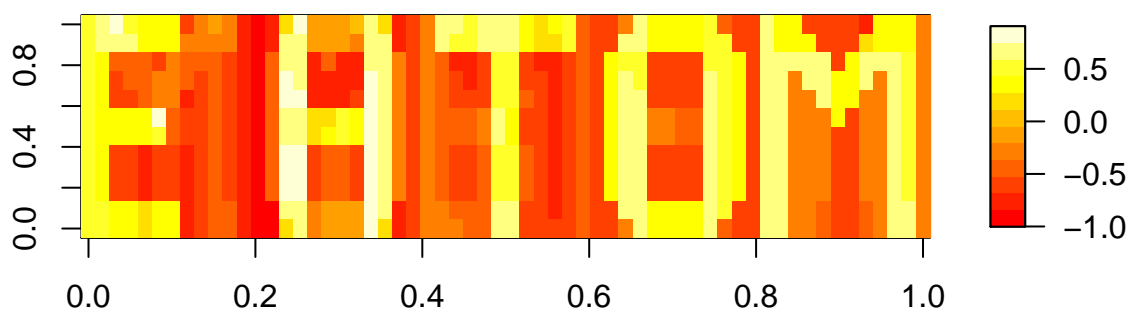
Original Input



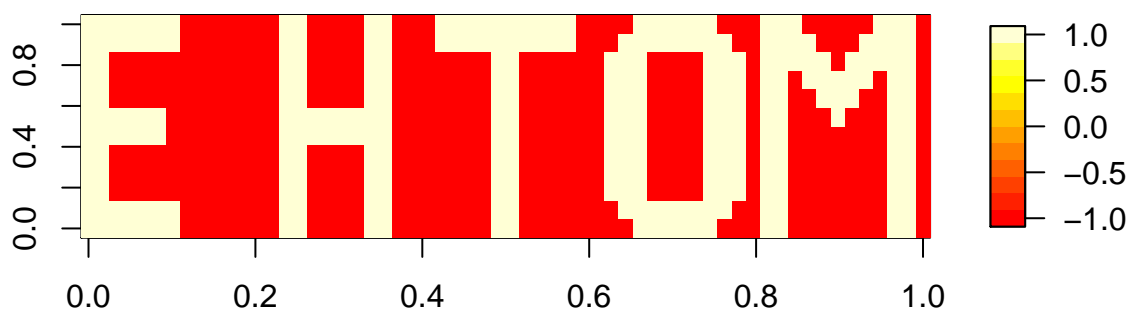
Desired Output



Memory Recall



Thresholded Memory Recall



## Summary

Please see above for our parameter selections. In this section, instead of corrupting the inputs to our network, we corrupted the inputs used to train our network; we corrupted 25% of the values from each input pattern. Our maximum number of learning steps is 1000, however, if we reached our error tolerance of 0.0000001 then we would immediately terminate the learning process and return the memory matrix. Our learning rate acts as a factor by which to influence the change of weights, we felt that 0.001 was appropriate for us. Our threshold methodology was to set the value to 1 if a value was greater or equal to 0 and to change it to -1 in all other cases. As can be seen by our recalled image and recalled thresholded image in this case, we had quite good results. In the original recalled image, we can clearly see the EHTOM pattern what we expect, and at the same time, our thresholded image looks to be perfectly recalled!