**Abstract**

* Relate an input image to a desired set of output values
* Hebb and Pseudo summary
* Noise exists in practical applications, desire to correctly predict noisy images

**Method**

1. Training Phase:
   * First create image as matrix..
   * ..then convert image to a vector..
   * ..store each input vector, target output vector, and weight matrix to an array..
   * ..use input vector paired with desired output target vector to train weight matrices

* After weight matrices are trained for all input vectors in all systems, training phase is complete
* At this point, may pass some (uncorrupted) input vectors to the network to verify correct output

1. Computing Phase:
   * For each weight matrix obtained during the Training Phase..
   * ..apply noise to 2 pixels for each input vector used to train the current weight..
   * ..and supply these noisy inputs to the network.
   * Repeat this process 10 times, recording results each time to calculate an error percentage
   * After 10 tests ran with 2 noisy pixels, repeat the entire process with 4, then 6, noisy pixels

**Result**

* Analyze error wrt:
  + Each network
  + Each group of N-noisy pixels
  + Number of unique input vectors (stored digits column in the provided instructions)
  + Size of input vectors
  + Size of output vectors

**Conclusion**

* Why is performance different
* Predict how network could be more correct/efficient
* Limitations of one or both networks

**References**