Pattern Recognition with Deep Boltzman Machines

Natasha Parikh and Miranda Parker
Summary of Problem

- Pattern Recognition
Background on Deep Boltzmann Machines

Boltzmann Machine

Restricted Boltzmann Machine
Background on Deep Boltzmann Machines

Figure 2: Left: A three-layer Deep Belief Network and a three-layer Deep Boltzmann Machine. Right: Pretraining consists of learning a stack of modified RBM’s, that are then composed to create a deep Boltzmann machine.
Our Approach

- Training and testing sets
- Image processing
- Training with RBMs to create images
- Building a DBM to find classification errors
Our Approach - Image Processing

- Manual conversions for uniformity
  - 50x50 pixels
  - Grayscale
Our Approach-Image Processing

- Matlab processing
  - Ultimately convert to binary in .mat file
  - Save classification labels for each image
Our Approach-Neural Network

- Pre-train 3 layers of RBMs
  - 500 neurons in the first two layers, 2000 neurons in the last layer
Our Approach - Neural Network

- Making a Deep Boltzmann Machine
  - Use the defining values from the RBMs to classify the training and testing data
Our Results - Training 3 layers of RBMs

- Not the worst...
  - Layer 1:
  - Layer 2:
  - ...and Layer 3:
Our Results - Testing

- Not so good
- 63% misclassification rate (for 3 layers)
Future Work

- Improve code compatibility

---

CS 70: DATA STRUCTURES & PROGRAM DEVELOPMENT

FALL 2012

Elegance & Simplicity

Scenario

Quinn was asked to write a function to check an array of floating point numbers to make sure none are zero. The function will be used to check a column vector of coefficients in a scientific computation, which would fail if any of the coefficients were zero.
Future Work

- Comment the Code!
Future Work

- Improve performance
- Use more patterns!
Future Work

- Sanity check - no interference = good results?
References

http://deeplearning.net/software_links/
