* Introduction
  + Automated handwriting recognition definition
    - Unsupervised learning versus hand built feature detectors
    - Advantage of massive amounts of unlabeled data
  + Restricted Boltzmann machine definition
    - bipartite graph
    - gibbs sampling
    - contrastive divergence
    - universal approximator
  + Convolutional Variant
    - Comparison and advantages
    - GPU implementation
    - probabilistic max pooling
      * Reason it was omitted here
    - Sparsity requirements
      * prior implementation
  + No discrimination done in this paper
* Prior Work
  + Work on MNIST
    - unsupervised feature learning most promising
    - less hardcoded information
    - generative models
  + First restricted boltzmann machine
    - contrastive divergence made it possible
  + Greedy layer wise for DBN
    - generative and discriminative
  + Introducing the CRBM
    - very good features extracted in 1 and 2 layers
    - probabilistic max pooling
      * neighborhoods
        + only one detector on
    - sparse representation
* Technical Methods
  + RBM
    - minibatches
    - contrastive divergence
    - mind vs brain states
    - momentum
    - weight decay
    - universal approximator
      * model can converge on any data
  + CRBM
    - smaller batches and less training examples
    - no pooling
    - penalty for large gradient
    - slower
    - sparsity penalty
      * implemented globally
      * use cross-entropy for loss function
        + Honglak used loss function more closely related to sparse coding

squared expectation of hidden unit

* + - * + Has to update and apply sparsity penalty per feature map

also applies the penalty via the bias instead of the weights

* + - * My implementation, applying cross entropy to all feature maps in parallel before pooling and penalizing both the weights and bias with sparsity penalty, has success at controlling sparsity
        + better because less updates and computation
        + possible global sparsity of a CDBN can be achieved layerwise this way
        + future work
* Results
  + images
    - pixel error was main way to tell how well
      * checked average pixels off per image
    - crbm does as well in one epoch as vanilla rbm does after 10 epochs
      * however the training is much slower unless the code is ran on the gpu
    - scrbm does the best
      * reaches target sparsity in approximately 5 batches and oscillates around the target
      * more parameters in the loss function the more easy the algorithm gets stuck
    - The nine that got fixed
    - Images of the reconstruction comparison
    - filter maps over time
      * overcomplete which would be remedied by max pooling
  + This process could be used to more efficiently train sparse representations
    - global sparsity control could be experimented on in deeper networks
    - Promise that with correct parameterization we can train systems with less overhead
  + Importance of sparsity
    - neural codes
    - competition
  + Convolution is most promising system and any advances in the training algorithm allows it to be applied more widely
    - More elegant training algorithm
  + Overall great experience