Feature Selection

Richard Pospesel and Bert Wierenga

Introduction

- Preprocessing
- Peaking Phenomenon
- Feature Selection Based on Statistical Hypothesis Testing
- Dimensionality Reduction Using Neural Networks

Outlier Removal

- For a normally distribution random variable
  - 2*σ covers 95% of points
  - 3* σ covers 99% of points
- Outliers cause training errors

Data Normalization

- Normalization is done so that each feature has equal weight when training a classifier

$$\bar{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik}, \qquad k = 1, 2, \ldots, l$$

$$\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ik} - \bar{x}_k)^2$$

$$\hat{x}_{ik} = \frac{x_{ik} - \bar{x}_k}{\sigma_k}$$

Data Normalization (cont)

- Softmax Scaling
  - "squashing" function mapping data to range of [0,1]

$$y = \frac{x_{ik} - \bar{x}_k}{r\sigma_k}, \qquad \hat{x}_{ik} = \frac{1}{1 + \exp(-y)}$$

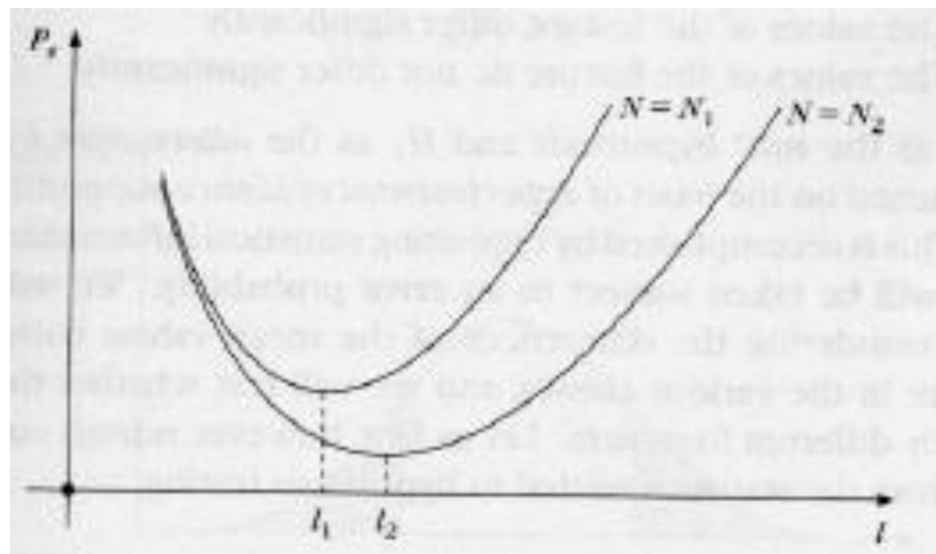
Missing Data

- Multiple Imputation
  - Estimating missing features of a feature vector by sampling from the underlying probability distribution per feature

Peaking Phenomenon

- If for any feature *l* we know the pdf, than we can perfectly discriminate the classes by increasing the number of features
- If pdfs are not known, than for a given *N*, increasing number of features will result in the maximum error, 0.5

- Optimally: $l = N / \alpha$
  - $2 < \alpha < 10$
- For MNIST:
  - $784 = 60{,}000 / \alpha$
  - $\alpha = 60{,}000 / 784$
  - $\alpha = 76.53\ldots$



Feature Selection Based On Statistical Hypothesis Testing

- Used to determine if the distributions of values of a feature for two different classes are distinct using a t-test
- If they around found to be distinct within a certain confidence interval, than we include the feature in our feature vector for classifier training

Feature Selection Based On Statistical Hypothesis Testing (cont)

- Test statistic for Null hypothesis (assuming unknown variance)

$$q = \frac{(\bar{x} - \bar{y}) - (\mu_1 - \mu_2)}{s_z \sqrt{\frac{2}{N}}}$$

- where

$$s_z^2 = \frac{1}{2N - 2} \left(\sum_{i=1}^{N} (x_i - \bar{x})^2 + \sum_{i=1}^{N} (y_i - \bar{y})^2 \right)$$

- Compare q to the t-distribution with 2N – 2 degrees of freedom to determine confidence that two distributions are different
- Simpler version for when we "know" the variance which compares q against a Gaussian

Feature Selection Based On Statistical Hypothesis Testing Example:

level $\rho = 0.05$.

From the foregoing we have

$$\omega_1: \quad \bar{x} = 3.73 \quad \hat{\sigma}_1^2 = 0.0601$$

$$\omega_2: \quad \bar{y} = 3.25 \quad \hat{\sigma}_2^2 = 0.0672$$

For $N = 10$ we have

$$s_z^2 = \frac{1}{2}(\hat{\sigma}_1^2 + \hat{\sigma}_2^2)$$

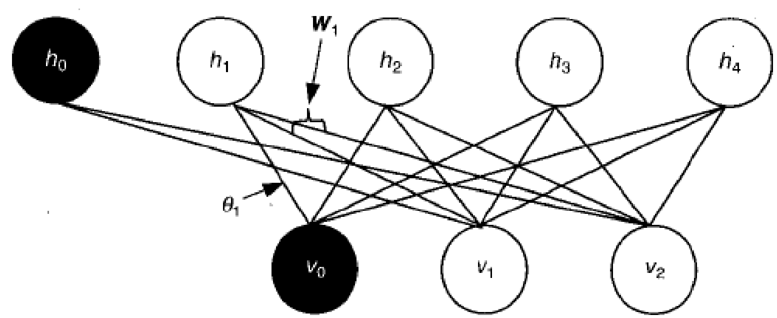
$$q = \frac{(\bar{x} - \bar{y} - 0)}{s_z\sqrt{\frac{2}{N}}}$$

$$q = 4.25$$

Degrees of freedom	$1 - \rho$	0.9	0.95	0.975	0.99	0.995
10		1.81	2.23	2.63	3.17	3.58
11		1.79	2.20	2.59	3.10	3.50
12		1.78	2.18	2.56	3.05	3.43
13		1.77	2.16	2.53	3.01	3.37
14		1.76	2.15	2.51	2.98	3.33
15		1.75	2.13	2.49	2.95	3.29
16		1.75	2.12	2.47	2.92	3.25
17		1.74	2.11	2.46	2.90	3.22
18		1.73	2.10	2.44	2.88	3.20
19		1.73	2.09	2.43	2.86	3.17
20		1.72	2.09	2.42	2.84	3.15

Reducing the Dimensionality of Data with Neural Networks

- Restricted Boltzmann Machine
  - Stochastic variant of a Hopfield Network
  - Two Layer Neural Network



  - Each Neuron is "Stochastic Binary"

$$p_{vi} = p(v_i = 1) = \frac{1}{1 + \exp(-\sum_j w_{ij} h_j)}$$

$$p_{hj} = p(h_j = 1) = \frac{1}{1 + \exp(-\sum_i w_{ij} v_i)}$$

Reducing the Dimensionality of Data with Neural Networks (cont)

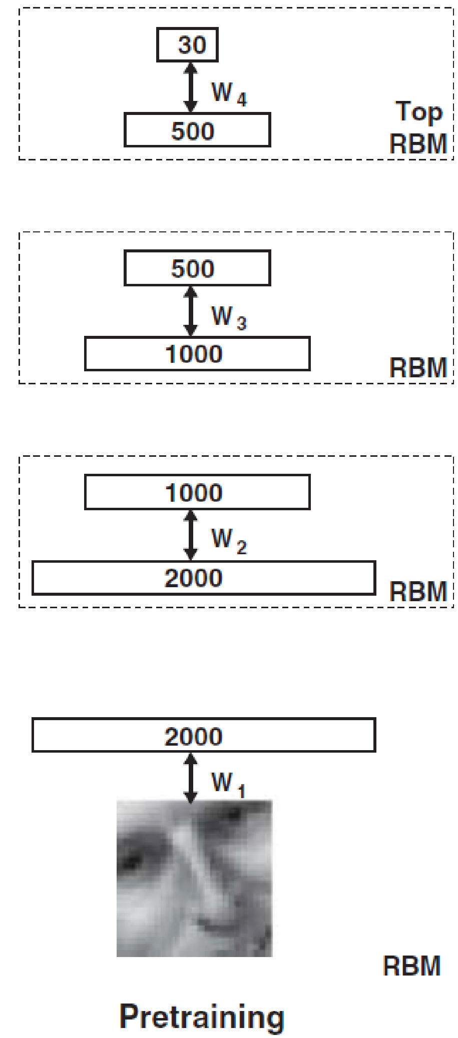
- Easy unsupervised descent training algorithm:

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}}\right)$$

  - Minimizes the "Free Energy"
- Allows the RBM to learn features found in input data

Reducing the Dimensionality of Data with Neural Networks (cont)

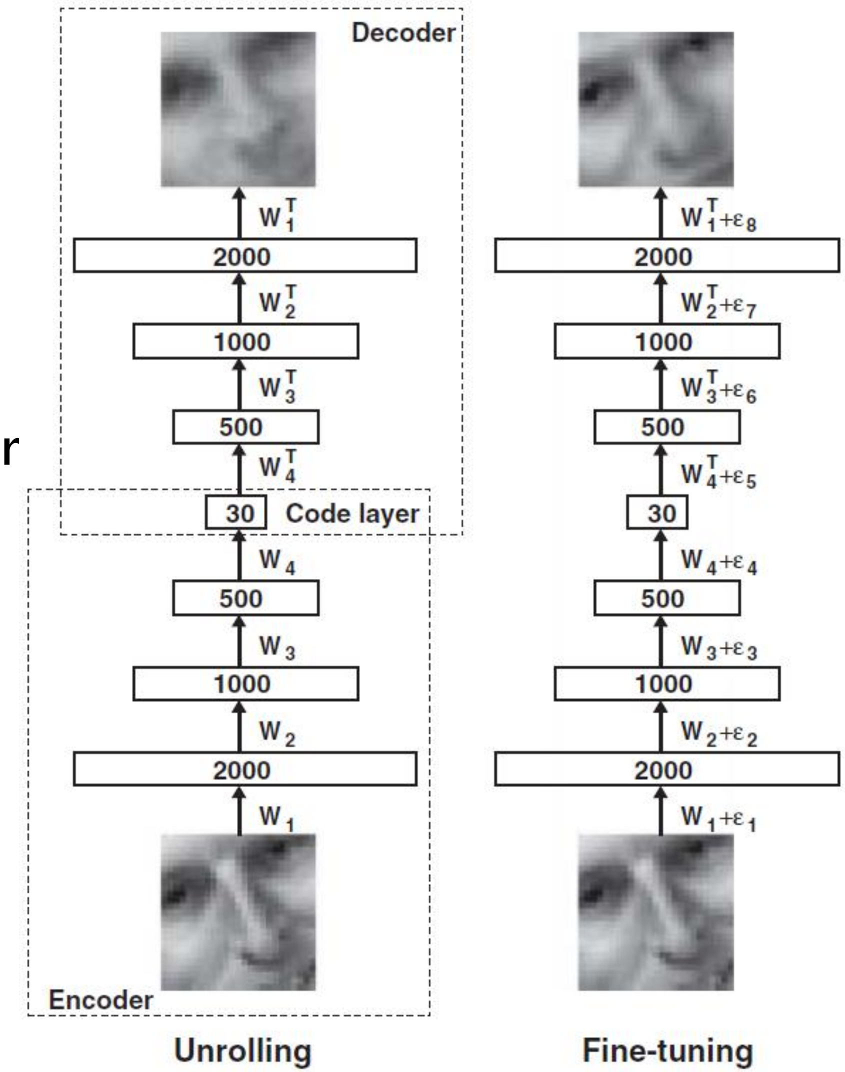
- RBMs can be stacked into a "Deep Belief Network"
  - Hidden neurons remain Stochastic Binary, but Visible neurons are now Logistic
- By stacking RBMs with decreasing sized Hidden Layers, we can reduce the number of dimensions of the underlying data.
- First RBM uses data as input
  - Each successive RBM uses output probabilities of previous RBM's hidden layer as training data.



Reducing the Dimensionality of Data with Neural Networks (cont)

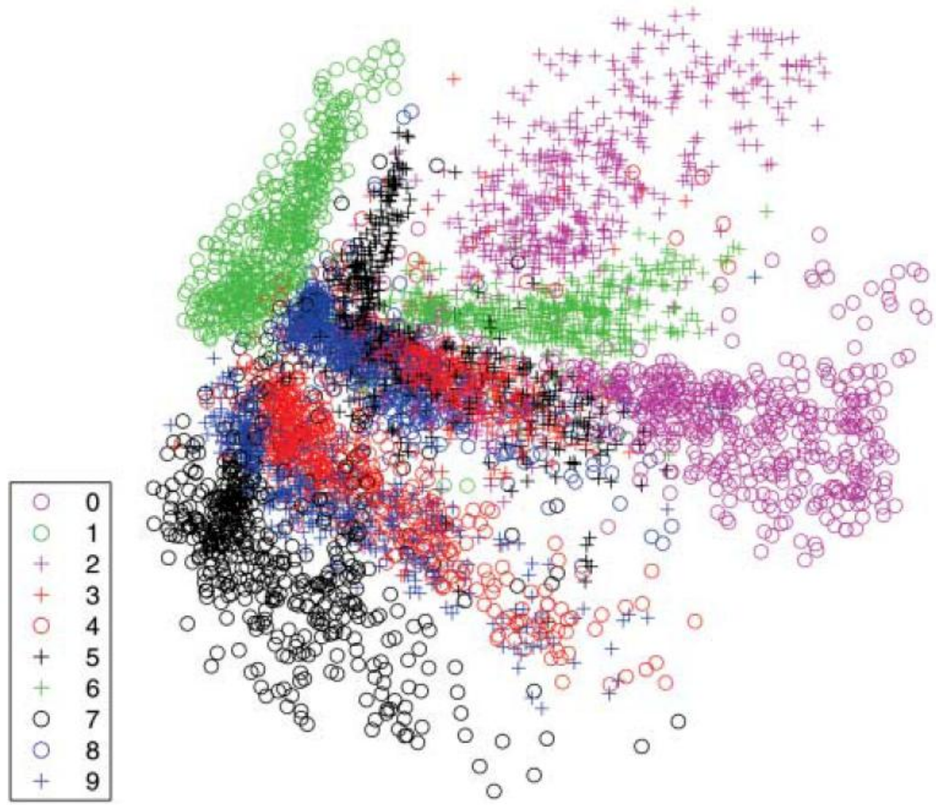
- Once a DBN Encoder network has been trained in the layer wise fashion, we can turn it around to make a DBN Decoder network

- This Encoder-Decoder pair can then be "Fine Tuned using Backpropagation



Reducing the Dimensionality of Data with Neural Networks (cont)

- 784-1000-500-250-2 AutoEncoder MNIST Visualization



Reducing the Dimensionality of Data with Neural Networks (cont)

- Run Demo

References

- G. Hinton and R. Salakhutdinov. "Reducing the dimensionality of data with neural networks" *Science* Vol. 313, No. 5786, pp. 504-507, 28 July 2006
- H Chen and A. Murray. "Continuous restricted boltzmann machine with an implementable training algorithm" *IEEE Proceedings* Vol. 150, No. 3 June 2003