Lecture 15

Artificial Neural Networks Presentation (3 of 4): Pattern Recognition using Unsupervised ANNs

Monday, February 21, 2000

Prasanna Jayaraman
Department of Computing and Information Sciences, KSU
http://www.cis.ksu.edu/~prasanna

Readings:
"The Wake-Sleep Algorithm For Unsupervised Neural Networks"
- Hinton, Dayan, Frey and Neal

Presentation Outline

- **Paper**
  - **"The Wake-Sleep Algorithm For Unsupervised Neural Networks"**
  - **Authors: Hinton, Dayan, Frey and Neal**
- **Necessity of this Topic**
  - **Supervised learning algorithm for multi-layer network suffers from**
    - **Requirement of a teacher**
    - **Requirement of an error communication method**
- **Overview**
  - **Unsupervised learning algorithm for a multi-layer network**
    - **Wake-Sleep Algorithm**
    - **Boltzmann and factorial distribution**
    - **Kullback-Leibler divergence**
    - **Training algorithms**

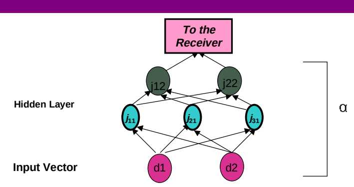
The Core Idea

- **Goal**
  Economical representation and accurate reconstruction of input.
- **Aim**
  To minimize the "description length".
- **Idea**
  Driving the neurons of ANN by the appropriate connection in the corresponding phase achieves the desired goal.
- **A Few Basic Jargons**
  - **ANN Connections**
    - **Recognition connections convert the input vector into a representation in hidden units.**
    - **Generative connections reconstruct an approximation to the input vector from its underlying representation.**

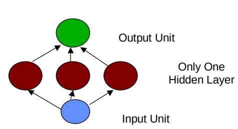
Sleep & Wake Phases

- **Wake Phase**
  - **The units are driven *bottom-up* using the *recognition weights*, producing a representation of the input vector in all the hidden layers.**
  - **This "total representation" is used to communicate the input vector, d, to the receiver.**
  - **Generative connections are adapted to increase the probability that they would reconstruct the correct activity vector in the layer below.**
  - **Only generative weights learn in this phase.**
- **Sleep Phase**
  - **Neurons are driven *top-down* by *generative connections* which reconstruct the representation in one layer from the representation in the layer above.**
  - **Recognition connections are adapted to increase the probability that they would produce the correct activity vector in the layer above.**

Explanatory Figures

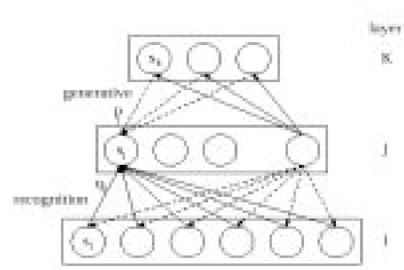


Fundamentals of Wake - Sleep Algorithm



Basics of Other Training Algorithms

Sample Figures



Wake - Sleep Algorithm

- Wake phase is invoked initially to create the total representation of the inputs.
- Stochastic binary units are chosen for training the 2 basic connections of ANN.
- The probability that the unit is on is:

$$\text{Prob}(s_v = 1) = \frac{1}{1 + \exp(-b_v - \sum_u s_u w_{uv})}$$

- The binary state of each hidden unit, j, in total representation α is s_j^α
- Activity of each unit, k, in the top hidden layer is communicated using the distribution

$$\left(p_k^\alpha, 1 - p_k^\alpha\right)$$

- Activities of the units in each lower layer are communicated using the distribution

$$\left(p_j^\alpha, 1 - p_j^\alpha\right)$$

Wake - Sleep Algorithm

- The description length of the binary state of unit "j" is:

$$C\left(s_j^\alpha\right) = -s_j^\alpha \log p_j^\alpha - \left(1 - s_j^\alpha\right)\log\left(1 - p_j^\alpha\right)$$

- The description length for the entire input vector "d" is:

$$c(\alpha, d) = c(\alpha) + c(d \mid \alpha) = \sum_{l \epsilon L} \sum_{j \epsilon L} c\left(s_j^\alpha\right) + \sum_i c\left(s_i^d \mid \alpha\right)$$

- All the recognition weights are turned off and the generative weights drive the units in the top-down fashion.
- As the hidden units are stochastic, this produces a "fantasy" vectors on the input units.
- Generative weight is adjusted in proportion to minimize the expected cost and to maximize the probability that the visible vectors generated by the model would match the observed data.
- Then, only the recognition weights are adjusted to maximize the log probability of recovering the hidden activities that actually caused the fantasy.

Helmholtz Machine

- The recognition weights determine a conditional probability distribution Q(. | d) over α.
- Initially, fantasies will have a different distribution than the training data.
- Helmholtz Machine
  - We restrict Q(. | d) to be a product distribution within each layer that is conditional on the binary states in the layer below and we can therefore compute it efficiently using a bottom-up recognition network. We call the model that uses a bottom-up recognition to minimize the bound as Helmholtz machine.
  - Minimizing the cost of representation can be done by generating a distribution sample from the recognition network and incrementing the top-down weight. This is a bit difficult but a simple approximation method could be generating a stochastic sample from the generative model and then we increment each bottom-up weight to increase the log probability that the recognition weights would produce the correct activities in the layer above. This way of fitting a Helmholtz machine is called the "wake-sleep" algorithm.

Factorial Distribution

- Boltzmann & Factorial Distribution
  - The recognition weights take the binary activities in one layer and stochastically produce binary activities in the layer above using a logistic function. So, for a given visible vector, the recognition weights may produce many different representations in the hidden layers but we can get an unbiased sample in a single pass.
  - C(d) is minimized when the probabilities of the alternatives are exponentially related to their costs by the Boltzmann distribution.
  - Make the recognition distribution as similar as possible to the posterior distribution to obtain the lowest cost representation.
  - The distribution produced by the recognition weights is a factorial distribution in each hidden layer because the recognition weights produce stochastic states of units within a hidden layer that are conditionally independent given the states in the layer below.

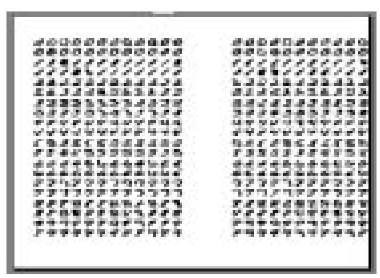
Kullback - Leibler Divergence

- Recognition distribution can not model non factorial distribution and hence it is impossible to exactly match the posterior distribution.
- Kullback - Leibler divergence between Q(. | d) and P(. | d) is the amount by which the description length using Q(. | d) exceeds - log P(d) .
- Kullback - Leibler divergence is

$$\sum_\alpha Q(\alpha \mid d) \log \frac{Q(\alpha \mid d)}{P(\alpha \mid d)}$$

- Unsupervised Training Algorithms
  - Principal Component Analysis
  - Competitive Learning or Vector Quantization or Clustering
  - In these approaches, there is only one hidden layer and there is no necessary to distinguish between the two kinds of weights as they are always the same.
  - This minimum description length approach treats the problem of learning as *statistical as it fits a generative model which accurately captures the structure in the input examples.*

Sample Figures



Summary Points

- ***Content Critique***
- ***Strengths***
  - *It is relatively an efficient method of fitting a multi layer stochastic generative model to a data.*
  - *In contrast to the normally available generative models, in addition to the top-down connections, this uses the bottom-up connections also to approximate the probability distribution over the hidden units given the data.*
- ***Weaknesses***
  - *Sleep phase creates a fantasy vector (close to the real vector) and then the wake phase, by adjusting the recognition weights trying to reconstruct the fantasy vector and not the real one.*
  - *Recognition weights produce only a factorial distribution of the hidden units but this demerit is weeded out or reduced by the use of generative weights in the wake phase, which minimizes the divergence.*

Summary Points

- **Presentation Critique**
  - **Audience: AI experts, ANN engineers, applied logic researchers, biophysicists**
  - **Application: Pattern Recognition in DNA sequence, Zip Code Scanning of postal mails etc.**
  - **Positive and exemplary points**
    - **Clear introduction to one of a new algorithm**
    - **Checking its validity with examples from various fields**
  - **Negative points and possible improvements**
    - **The effectiveness of this algorithm has to be compared with other predominant methods like base rate model, binary mixture model, Gibb's machine, mean field method etc. which can also be used for learning in multi layer network.**
    - **Experimental values depicting the training time, cost of representing the given input and compression performance could have been furnished for the various example problems, to leave an impression on the user's mind.**