New Applications

Ian Buck
Stanford University
Neural Networks

• Commercial Use
  – Handwriting recognition
    • Mail Sorting
    • User Interface

• Research Use
  – Network Design (Compute Heavy)
    • Training on large datasets
    • Parameter search
Neural Networks

Input (Pixels) => Hidden Layer => Output (Probabilities)

Any non-linear mapping function: tanh(), sigmoid()

Chapter 20, Russel, Norvig
Different weight per arrow

Hidden Layer
Neural Networks

Matrix * Vector

$\mathbf{a} = \mathbf{W} \mathbf{in}$

Function Map

Matrix * Vector

Function Map
Backward Propagation Learning

• Iterative process over learning samples

• Weight Matrix Update
  \[ W_{i,j} += \alpha \cdot a_j \cdot \Delta_i \]

• Error Propagation
  \[ \Delta_j = \text{act'}(in_j) \cdot W^T \cdot \Delta_i \]
Brook

• Efficient linear algebra implementation
  – Matrix * vector
    
    ```c
    for (i=0; i<rows; i++)
    ... dot product reduce ...
    ```
  – Need for multi-dimensional reductions
    • Reduce to stream

![Diagram of Reduce operation]

```plaintext
float a<100,200>  float b<100,1>
```
Gaussian Mixture Models

- Given a set of nD data
- Fit m nD gaussians

\[ p(x|\Theta) = \sum_{i=1}^{M} \alpha_i p_i(x|\theta_i) \]

\[ \theta = (\mu, \Sigma) \]
Gaussian Mixture Models

• Applications
  – Speech
    • Phoneme detector
    • Speaker identification
  – Vision
    • Skin detection
  – Sales and Marketing
    • Searching for sales trends
Gaussian Mixture Models

• Applications
  – Speech
    • Phoneme detector
    • Speaker identification
  – Vision
    • Skin detection
  – Sales and Marketing
    • Searching for sales trends

Big Data Problems
Algorithm

- Expectation Maximization (EM) Algorithm

\[ p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X}). \]

\[ \Theta^* = \arg\max_{\Theta} \mathcal{L}(\Theta|\mathcal{X}). \]

\[ \Theta = (\mu, \sigma^2, \alpha) \]

Blimes, A Gentle Tutorial of the EM algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models
EM Algorithm

Iterative Algorithm

\[ p_\ell(x | \mu_\ell, \Sigma_\ell) = \frac{1}{(2\pi)^{d/2}|\Sigma_\ell|^{1/2}} e^{-\frac{1}{2}(x-\mu_\ell)^T \Sigma_\ell^{-1}(x-\mu_\ell)}. \]

\[
\alpha^\text{new}_\ell = \frac{1}{N} \sum_{i=1}^{N} p(\ell | x_i, \Theta^g)
\]

\[
\mu^\text{new}_\ell = \frac{\sum_{i=1}^{N} x_i p(\ell | x_i, \Theta^g)}{\sum_{i=1}^{N} p(\ell | x_i, \Theta^g)}
\]

\[
\Sigma^\text{new}_\ell = \frac{\sum_{i=1}^{N} p(\ell | x_i, \Theta^g)(x_i - \mu^\text{new}_\ell)(x_i - \mu^\text{new}_\ell)^T}{\sum_{i=1}^{N} p(\ell | x_i, \Theta^g)}
\]

Use Bayes Rule to compute \( p(\ell | x_i, \Theta^g) \)
EM Algorithm

• Gaussian Evaluation
  – Lots of ops

• Series computation
  – Sum reduction
  – Bandwidth

• Iterative
  – Lots of work
Hidden Markov Models (HMM)

- State machine producing outputs
  - Each state contains different prob. dist. for output

\[ \pi_i = p(Q_1 = i) \]
\[ A = \{a_{i,j}\} = p(Q_t = j | Q_{t-1} = i) \]
\[ b_j(o_t) = p(O_t = o_t | Q_t = j) \]
\[ B = \{b_j(\cdot)\} \]
HMM Applications

• Speech
  – Recognition
    • Combine with GMM
    • Training and evaluation

• Vision
  – Handwriting
  – Gesture & lip reading

• Text
  – Translation
EM Algorithm for HMM

- Training HMM with EM

\[
\tilde{\pi}_i = \gamma_i(1)
\]

\[
\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}
\]

\[
\tilde{b}_i(k) = \frac{\sum_{t=1}^{T} \delta_{o_t,v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}
\]

\[
\gamma_i(t) = \frac{\alpha_i(t) \beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t) \beta_j(t)}
\]

\[
\xi_{ij}(t) = \frac{\gamma_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t + 1)}{\beta_i(t)}
\]

\[
1. \beta_i(T) = 1
\]

\[
2. \beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t + 1)
\]

\[
1. \alpha_i(1) = \pi_i b_i(o_1)
\]

\[
2. \alpha_j(t + 1) = \left[ \sum_{i=1}^{N} \alpha_i(t) a_{ij} \right] b_j(o_{t+1})
\]
HMM

- Series Computations
  - Reductions
  - Bandwidth wins
- Iterative
  - Lots of Work
- Combine with GMM
  - Lots of ops