Deep Network for Speech Emotion Recognition

—A Study of Deep Learning—

Zhuowei Han

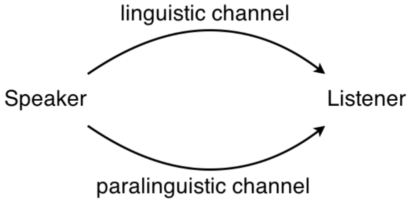
Institut für Signalverarbeitung
und Systemtheorie

Universität Stuttgart

16/04/2015

Speech Emotion Recognition

- Most current work focuses on speech processing based on linguistic information, e.g.: Skype Translator
- More natural human-machine interaction requires paralinguistic information such as age, gender, emotion.
- Speech Recognition / Speeker Identification / Emotion Recognition



Deep Learning

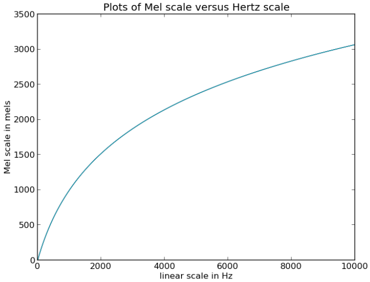
- Deep architecture for extracting complex structure and building internal representations from input
- New research area of machine learning (from shallow to deep structure)
- Widely applied in vision/audition processing, e.g. handwriting recognition (Graves, Alex, et al. 2009), traffic sign classification (Schmidhuber, et al. 2011), text translation (Google, 2014)

Table of Contents

Table of Contents

Mel Frequency Cepstral Features

- short-term power spectrum
- mel-scale approximate human perception
- widely-used in speech recognition tasks
- Transformation between Mel and Hertz scale



$$f_{mel} = 1125 \ \ln\left(1 + f_{Hz}/700\right)$$
$$f_{Hz} = 700\left(\exp(f_{mel}/1125) - 1\right)$$

Emotion Recognition Approaches

Traditional Approaches

- pre-selected features
- supervised training
- low-level features not appropriate for claasification
- shallow structure of classifiers

Deep Learning Approaches

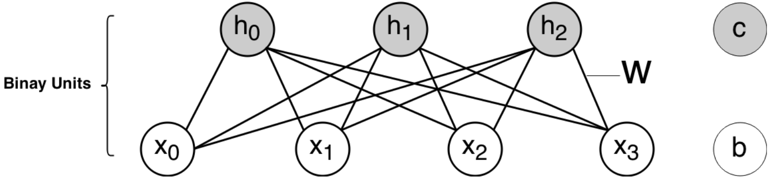
- learning representations from high-dim data
- extracting appropriate features without hand-crafting
- low-level features are used to build high-level features as network gets deeper
- frame-based classification

Table of Contents

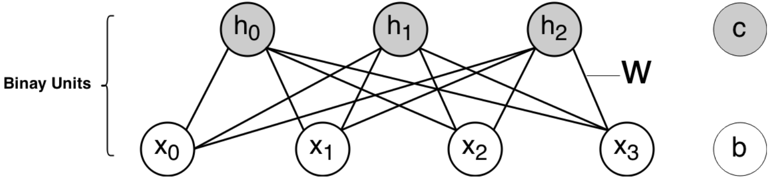
Concepts

- Generative graphical model, capture data distrbution $P(\mathbf{x}|\boldsymbol{\theta})$
- Trained in unsupervised way, only use unlabeled input sequence$\mathbf{x}$ for learning.
  - automatically extract useful features from data
  - Find hidden structure (distribution).
  - Learned features used for prediction or classification
- Successfully applied in motion capture (Graham W. Taylor, Geoffrey E. Hinton, 2006)
- Potential to be extend to capture temporal information

Structure



Structure



Energy Function: $E_{\boldsymbol{\theta}} = -\mathbf{x}^{\mathbf{T}}\mathbf{W}\mathbf{h} - \mathbf{b}^{\mathbf{T}}\mathbf{x} - \mathbf{c}^{\mathbf{T}}\mathbf{h}$

Joint Distribution: $P^{RBM}(\mathbf{x}, \mathbf{h}) = \dfrac{1}{Z} e^{-E_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{h})}$

Partition Function: $Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{h})}$

Free Energy: $\mathcal{F}(\mathbf{x}) = -\log \sum_{h} e^{-E(\mathbf{x},\mathbf{h})}$

Inference

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h})$$

$$P(\mathbf{h}) = \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{h})$$

Inference

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h})$$

$$P(\mathbf{h}) = \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{h})$$

$$P(\mathbf{h}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{h})}{P(\mathbf{x})}$$

$$P(\mathbf{x}|\mathbf{h}) = \frac{P(\mathbf{x}, \mathbf{h})}{P(\mathbf{h})}$$

Inference

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h})$$

$$P(\mathbf{h}) = \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{h})$$

$$P(\mathbf{h}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{h})}{P(\mathbf{x})}$$

$$P(\mathbf{x}|\mathbf{h}) = \frac{P(\mathbf{x}, \mathbf{h})}{P(\mathbf{h})}$$

$$P(h_j = 1 \mid \mathbf{x}) = sigmoid(\sum_i x_i W_{ij} + c_j)$$

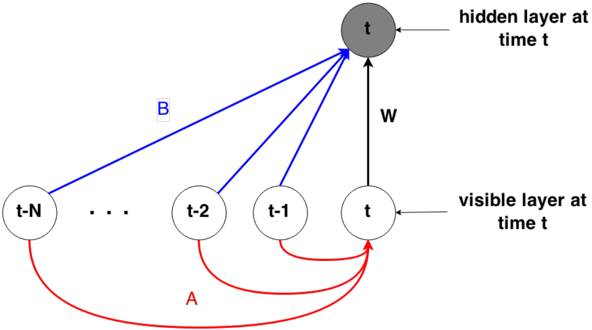
$$P(x_i = 1 \mid \mathbf{h}) = sigmoid(\sum_j W_{ij} h_j + b_i)$$

Conditional RBM

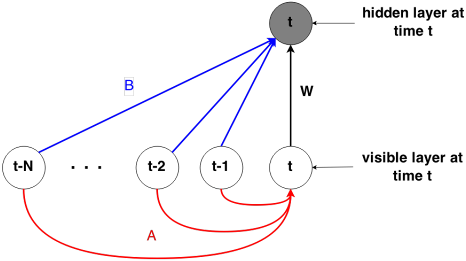
- Consider visible units from previous time step as additional bias for current visible and hidden layer
- A and B are weight parameter of visible (history) - visible and visible (history) - hidden connections
- Visible layer is linear units with independent Gaussian noise to model real-valued data, e.g. spectral features

Conditional RBM

- Consider visible units from previous time step as additional bias for current visible and hidden layer
- A and B are weight parameter of visible (history) - visible and visible (history) - hidden connections
- Visible layer is linear units with independent Gaussian noise to model real-valued data, e.g. spectral features



Conditional RBM



Energy Function: $E_{\boldsymbol{\theta}}^{CRBM}(\mathbf{x}, \mathbf{h}) = \left\| \frac{\mathbf{x} - \tilde{\mathbf{b}}}{2} \right\|^2 - \tilde{\mathbf{c}}^T \mathbf{h} - \mathbf{x}^T \mathbf{W} \mathbf{h}$

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{A} \cdot \mathbf{x}_{<t}$$

$$\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{B} \cdot \mathbf{x}_{<t}$$

$$\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{b}, \mathbf{c}\}$$

Free Energy: $\mathcal{F}(\mathbf{x}) = \left\| \mathbf{x} - \tilde{\mathbf{b}} \right\|^2 - \log(1 + e^{\tilde{\mathbf{c}} + \mathbf{x} \cdot \mathbf{W}})$

Conditional RBM

Energy Function: $E_{\boldsymbol{\theta}}^{CRBM}(\mathbf{x}, \mathbf{h}) = \left\| \frac{\mathbf{x} - \tilde{\mathbf{b}}}{2} \right\|^2 - \tilde{\mathbf{c}}^T \mathbf{h} - \mathbf{x}^T \mathbf{W} \mathbf{h}$

Free Energy: $\mathcal{F}(\mathbf{x}) = \left\| \mathbf{x} - \tilde{\mathbf{b}} \right\|^2 - \log(1 + e^{\tilde{\mathbf{c}} + \mathbf{x} \cdot \mathbf{W}})$

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{A} \cdot \mathbf{x}_{<t}$$

$$\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{B} \cdot \mathbf{x}_{<t}$$

$$\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{A}, \mathbf{B}, \mathbf{b}, \mathbf{c}\}$$

Training of Energy-based Model

Maximum Likelihood Estimation $P(\mathbf{x}|\boldsymbol{\theta})$

Note that KL is non-negative

Training of Energy-based Model

Maximum Likelihood Estimation $P(\mathbf{x}|\boldsymbol{\theta})$

Kullback-Leibler Divergence:

$$
\begin{aligned}
KL(Q(\mathbf{x})\|P(\mathbf{x}|\boldsymbol{\theta})) &= \int_{-\infty}^{\infty} Q(\mathbf{x}) \cdot \log \frac{Q(\mathbf{x})}{P(\mathbf{x}|\boldsymbol{\theta})} \mathrm{d}\mathbf{x} \\
&= \int_{-\infty}^{\infty} Q(\mathbf{x}) \cdot \log Q(\mathbf{x}) \mathrm{d}\mathbf{x} - \int_{-\infty}^{\infty} Q(\mathbf{x}) \cdot \log P(\mathbf{x}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x} \\
&= \langle \log Q(\mathbf{x}) \rangle_{Q(\mathbf{x})} - \langle \log P(\mathbf{x}|\boldsymbol{\theta}) \rangle_{Q(\mathbf{x})}
\end{aligned}
$$

$Q(\mathbf{x})$, true data distribution
$P(\mathbf{x}|\boldsymbol{\theta})$, model distribution
$\langle \cdot \rangle_{Q(\mathbf{x})}$, expectation w.r.t. $Q(\mathbf{x})$
Note that KL is non-negative

Training of Energy-based Model

$$-\log P(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{F}(\mathbf{x}) + \log \sum_{\mathbf{x}} \sum_{\mathbf{h}} e^{-E_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{h})}$$ Free Energy

Training of Energy-based Model

$$-\log P(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{F}(\mathbf{x}) + \log \sum_{\mathbf{x}} \sum_{\mathbf{h}} e^{-E_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{h})}$$ Free Energy

$$-\frac{\partial \log P(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} - \sum_{\tilde{\mathbf{x}}} P(\tilde{\mathbf{x}}) \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \boldsymbol{\theta}}$$ ←intractable!

Training of Energy-based Model

$$-\log P(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{F}(\mathbf{x}) + \log \sum_{\mathbf{x}} \sum_{\mathbf{h}} e^{-E_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{h})}$$ Free Energy

$$-\frac{\partial \log P(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} - \sum_{\tilde{\mathbf{x}}} P(\tilde{\mathbf{x}}) \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \boldsymbol{\theta}}$$ ←intractable!

$$-\frac{\partial \log P(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} - \frac{1}{|\mathcal{N}|} \sum_{\tilde{\mathbf{x}} \in \mathcal{N}} P(\tilde{\mathbf{x}}) \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \boldsymbol{\theta}}$$ sampling

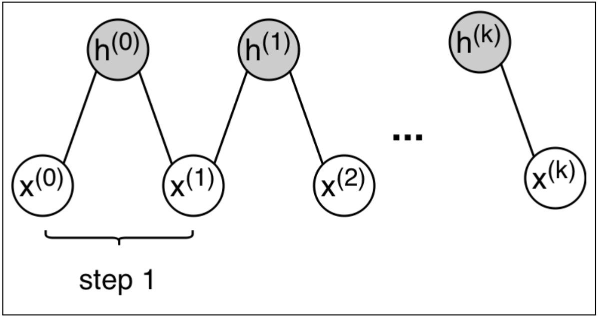
t steps-Gibbs sampling

$$\mathbf{x_1} \sim \hat{P}(\mathbf{x})$$
$$\mathbf{h_1} \sim \hat{P}(\mathbf{h}|\mathbf{x}_1)$$

$$\mathbf{x_2} \sim \hat{P}(\mathbf{x}|\mathbf{h}_1)$$
$$\mathbf{h_2} \sim \hat{P}(\mathbf{h}|\mathbf{x}_2)$$

$$\vdots$$

$$\mathbf{x_{t+1}} \sim \hat{P}(\mathbf{x}|\mathbf{h}_t)$$



Training of Energy-based Model

t steps-Gibbs sampling

$$\mathbf{x_1} \sim \hat{P}(\mathbf{x})$$

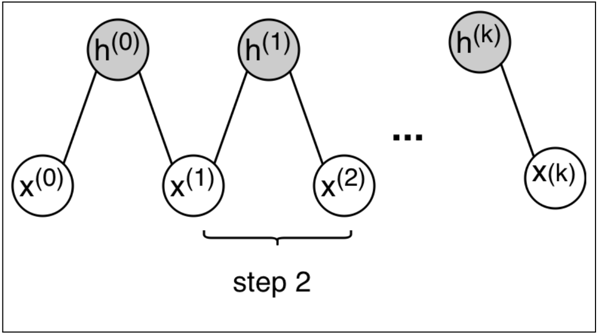
$$\mathbf{h_1} \sim \hat{P}(\mathbf{h}|\mathbf{x}_1)$$

$$\mathbf{x_2} \sim \hat{P}(\mathbf{x}|\mathbf{h}_1)$$

$$\mathbf{h_2} \sim \hat{P}(\mathbf{h}|\mathbf{x}_2)$$

$$\vdots$$

$$\mathbf{x_{t+1}} \sim \hat{P}(\mathbf{x}|\mathbf{h}_t)$$



t steps-Gibbs sampling

$$\mathbf{x_1} \sim \hat{P}(\mathbf{x})$$

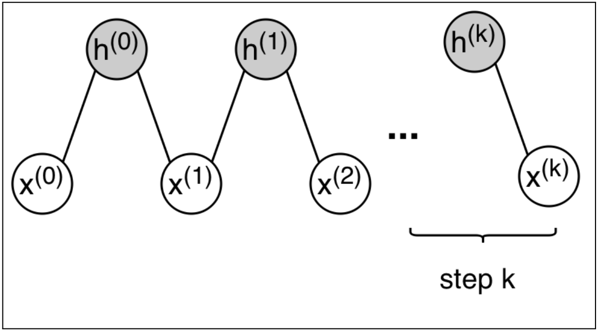
$$\mathbf{h_1} \sim \hat{P}(\mathbf{h}|\mathbf{x}_1)$$

$$\mathbf{x_2} \sim \hat{P}(\mathbf{x}|\mathbf{h}_1)$$

$$\mathbf{h_2} \sim \hat{P}(\mathbf{h}|\mathbf{x}_2)$$

$$\vdots$$

$$\mathbf{x_{t+1}} \sim \hat{P}(\mathbf{x}|\mathbf{h}_t)$$



Contrastive Divergence

- Performing k-Gibbs steps to generate $P_k(\mathbf{x}|\boldsymbol{\theta})$, approximation of model distribution
- Difference between approximation and true model distribution:

$$KL(P_k(\mathbf{x}|\boldsymbol{\theta})\|P(\mathbf{x}|\boldsymbol{\theta}))$$

- Constrastive Divergence (CD):

$$KL(Q(\mathbf{x})\|P(\mathbf{x}|\boldsymbol{\theta})) - KL(P_k(\mathbf{x}|\boldsymbol{\theta})\|P(\mathbf{x}|\boldsymbol{\theta}))$$

- With enough steps the Markov chain converges to stationary distribution:

$$P_{k\to\infty}(\mathbf{x}|\boldsymbol{\theta}) = P(\mathbf{x}|\boldsymbol{\theta})$$

- CD-1 performs well in practice

Constrasive Divergence

Parameter Update

$$\Delta\boldsymbol{\theta} \sim KL(P(\mathbf{x}|\boldsymbol{\theta})\|P_k(\mathbf{x}|\boldsymbol{\theta}))$$

Table of Contents

Conclusion

- Model with long-term dependencies shall be used for speech emotion
- CRBM is appropriate for short-term modelling, but not for long-term variation
- LSTM is good at modelling long time dependency
- Frame-based classification can also reach good result
  - CRBM-LSTM 71.98%
  - LSTM 81.59%
  - LSTM with rectifier layers 83.43%

Outlook

- Stacking CRBM to form deeper structure
- Train CRBM with more/larger database
- Second order optimization to speed up learning process
- Bi-directional LSTM, capturing future dependencies

Thank You!