Deep Network for Speech Emotion Recognition

—A Study of Deep Learning—

Zhuowei Han

Institut für Signalverarbeitung
und Systemtheorie

Universität Stuttgart

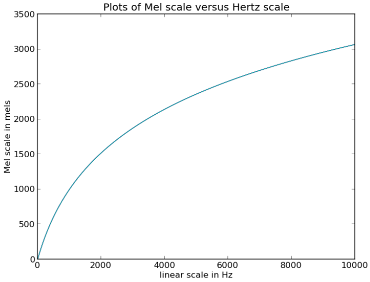
16/04/2015

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



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

Framework

- Pre-processing of emotion data to extract MFCC features
- Model data distribution based on MFCCs via unsupervised learning
- Classification with supervised learning

Emotion Recognition Approaches

Traditional Approaches

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

Deep Learning Approaches

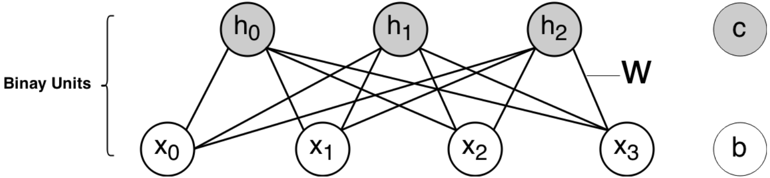
- learning representations from high-dim data
- extracting appropriate features without hand-crafting
- low-level features are used to build high-level features as network gets deeper
- frame-based classification

Table of Contents

Restricted Boltzmann Machine

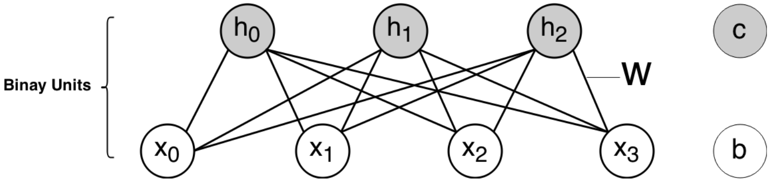
- Generative graphical model, capture data distrbution $P(\mathbf{x}|\boldsymbol{\theta})$
- Trained in unsupervised way, only use unlabeled input sequence $\mathbf{x}$ for learning.
  - automatically extract useful features from data
  - find hidden structure (distribution).
  - learned features used 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



$$\mathbf{x} \in \{0,1\}$$
$$\mathbf{h} \in \{0,1\}$$

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

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

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

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

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

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

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

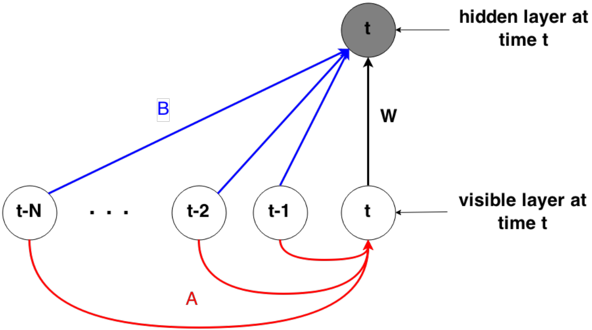
$$P(x_i = 1 \mid \mathbf{h}) = sigmoid(\sum_j W_{ij} h_j + b_i)$$

Conditional RBM

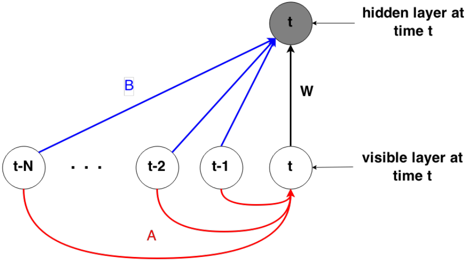
- Consider visible units from previous time step as additional bias for current visible and hidden layer
- A and B are weight parameter of visible (history) - visible and visible (history) - hidden connections
- Visible layer is linear units with independent Gaussian noise to model real-valued data,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

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

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

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

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

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

Training of Energy-based Model

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

Training of Energy-based Model

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

Kullback-Leibler Divergence:

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

$Q(\mathbf{x})$, 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

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

Kullback-Leibler Divergence:

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

$Q(\mathbf{x})$, 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

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

Training of Energy-based Model

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

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

$\mathbf{x}$, input (visible) data space

$\tilde{\mathbf{x}}$, all possible vectors in the data space, generated by model.

Training of Energy-based Model

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

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

$\mathbf{x}$, input (visible) data space
$\tilde{\mathbf{x}}$, all possible vectors in the data space, generated by model.

objective function by averaging log-likelihood over data:

$$-\left\langle \frac{\partial \log P(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_{\mathbf{x}} = \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{\mathbf{x}} - \left\langle \frac{\partial \mathcal{F}(\tilde{\mathbf{x}})}{\partial \boldsymbol{\theta}} \right\rangle_{\tilde{\mathbf{x}}}$$

Gibbs sampling

$$\mathbf{x}^{(1)} \sim P(\mathbf{x})$$

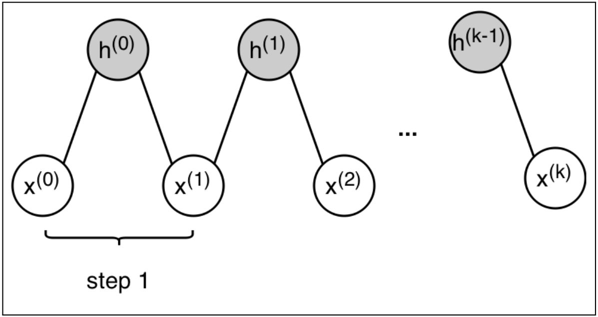
$$\mathbf{h}^{(1)} \sim P(\mathbf{h}|\mathbf{x}^{(1)})$$

$$\mathbf{x}^{(2)} \sim P(\mathbf{x}|\mathbf{h}^{(1)})$$

$$\mathbf{h}^{(2)} \sim P(\mathbf{h}|\mathbf{x}^{(2)})$$

$$\vdots$$

$$\mathbf{x}^{(k)} \sim P(\mathbf{x}|\mathbf{h}^{(k-1)})$$



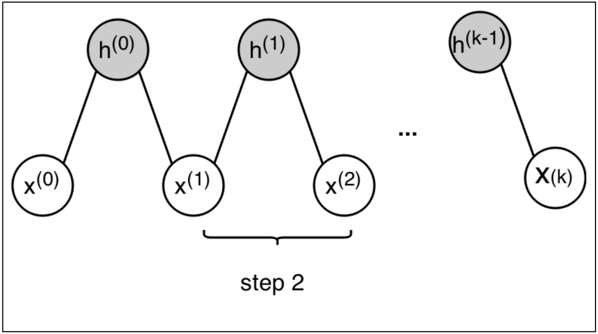
Gibbs sampling

$$\mathbf{x}^{(1)} \sim P(\mathbf{x})$$
$$\mathbf{h}^{(1)} \sim P(\mathbf{h}|\mathbf{x}^{(1)})$$

$$\mathbf{x}^{(2)} \sim P(\mathbf{x}|\mathbf{h}^{(1)})$$
$$\mathbf{h}^{(2)} \sim P(\mathbf{h}|\mathbf{x}^{(2)})$$

$$\vdots$$

$$\mathbf{x}^{(k)} \sim P(\mathbf{x}|\mathbf{h}^{(k-1)})$$



Gibbs sampling

$$\mathbf{x}^{(1)} \sim P(\mathbf{x})$$

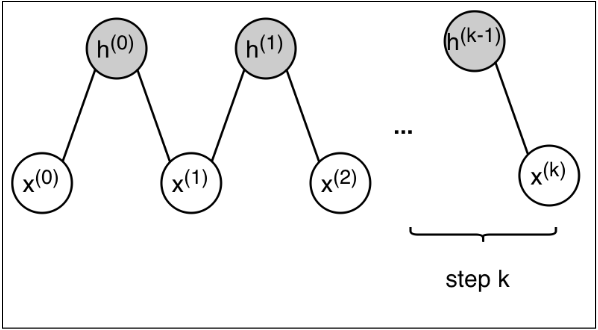
$$\mathbf{h}^{(1)} \sim P(\mathbf{h}|\mathbf{x}^{(1)})$$

$$\mathbf{x}^{(2)} \sim P(\mathbf{x}|\mathbf{h}^{(1)})$$

$$\mathbf{h}^{(2)} \sim P(\mathbf{h}|\mathbf{x}^{(2)})$$

$$\vdots$$

$$\mathbf{x}^{(k)} \sim P(\mathbf{x}|\mathbf{h}^{(k-1)})$$



Contrastive Divergence

- k=0, $P_0(\mathbf{x})$ is true data distribution, independent of parameter $\boldsymbol{\theta}$
- Performing k-Gibbs steps to generate $P_k(\mathbf{x}|\boldsymbol{\theta})$, with $k \to \infty$ the Markov chain converges to stationary distribution:

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

Contrastive Divergence

- k=0, $P_0(\mathbf{x})$ is true data distribution, independent of parameter $\boldsymbol{\theta}$
- Performing k-Gibbs steps to generate $P_k(\mathbf{x}|\boldsymbol{\theta})$, with $k \to \infty$ the Markov chain converges to stationary distribution:

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

Rewrite objective function:

$$-\left\langle \frac{\partial \log P(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0(\mathbf{x})} = \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0(\mathbf{x})} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_\infty(\mathbf{x}|\boldsymbol{\theta})}$$

Contrastive Divergence

- k=0, $P_0(\mathbf{x})$ is true data distribution, independent of parameter $\boldsymbol{\theta}$
- Performing k-Gibbs steps to generate $P_k(\mathbf{x}|\boldsymbol{\theta})$, with $k \to \infty$ the Markov chain converges to stationary distribution:

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

Rewrite objective function:

$$-\left\langle \frac{\partial \log P(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0(\mathbf{x})} = \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0(\mathbf{x})} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_\infty(\mathbf{x}|\boldsymbol{\theta})}$$

Contrastive Divergence: Perform CD-1

$$
\begin{aligned}
&-\frac{\partial}{\partial \boldsymbol{\theta}}(P_0 \| P_\infty^{\boldsymbol{\theta}} - P_1^{\boldsymbol{\theta}} \| P_\infty^{\boldsymbol{\theta}}) \\
&= \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_1^{\boldsymbol{\theta}}} + \frac{\partial P_1^{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \frac{\partial (P_1^{\boldsymbol{\theta}} | P_\infty^{\boldsymbol{\theta}})}{\partial P_1^{\boldsymbol{\theta}}}
\end{aligned}
$$

Contrastive Divergence: Perform CD-1

$$
\begin{aligned}
&-\frac{\partial}{\partial\boldsymbol{\theta}}(P_0\|P_\infty^{\boldsymbol{\theta}} - P_1^{\boldsymbol{\theta}}\|P_\infty^{\boldsymbol{\theta}}) \\
&= \left\langle\frac{\partial\mathcal{F}(\mathbf{x})}{\partial\boldsymbol{\theta}}\right\rangle_{P_0} - \left\langle\frac{\partial\mathcal{F}(\mathbf{x})}{\partial\boldsymbol{\theta}}\right\rangle_{P_1^{\boldsymbol{\theta}}} + \frac{\partial P_1^{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}}\frac{\partial(P_1^{\boldsymbol{\theta}}|P_\infty^{\boldsymbol{\theta}})}{\partial P_1^{\boldsymbol{\theta}}}
\end{aligned}
$$

Contrastive Divergence: Perform CD-1

$$
\begin{aligned}
&-\frac{\partial}{\partial \boldsymbol{\theta}}(P_0 \| P_\infty^{\boldsymbol{\theta}} - P_1^{\boldsymbol{\theta}} \| P_\infty^{\boldsymbol{\theta}}) \\
&= \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_1^{\boldsymbol{\theta}}} + \frac{\partial P_1^{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \frac{\partial (P_1^{\boldsymbol{\theta}} | P_\infty^{\boldsymbol{\theta}})}{\partial P_1^{\boldsymbol{\theta}}} \cancel{+ \frac{\partial P_1^{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \frac{\partial (P_1^{\boldsymbol{\theta}} | P_\infty^{\boldsymbol{\theta}})}{\partial P_1^{\boldsymbol{\theta}}}}
\end{aligned}
$$

Constrasive Divergence

Contrastive Divergence: Perform CD-1

$$
\begin{aligned}
& -\frac{\partial}{\partial \boldsymbol{\theta}}(P_0 \| P_\infty^{\boldsymbol{\theta}} - P_1^{\boldsymbol{\theta}} \| P_\infty^{\boldsymbol{\theta}}) \\
& = \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_1^{\boldsymbol{\theta}}} + \frac{\partial P_1^{\theta}}{\partial \theta} \frac{\partial (P_1^{\theta} | P_\infty^{\theta})}{\partial P_1^{\theta}} \cancel{+ \frac{\partial P_1^{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \frac{\partial (P_1^{\boldsymbol{\theta}} | P_\infty^{\boldsymbol{\theta}})}{\partial P_1^{\boldsymbol{\theta}}}}
\end{aligned}
$$

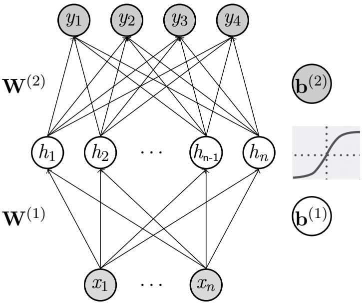
Parameter Update

$$
\Delta \boldsymbol{\theta} \sim \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_0} - \left\langle \frac{\partial \mathcal{F}(\mathbf{x})}{\partial \boldsymbol{\theta}} \right\rangle_{P_1^{\boldsymbol{\theta}}}
$$

Table of Contents

Hidden layer pre-activation:

$$\mathbf{a}(\mathbf{x}) = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$$
$$a_j(\mathbf{x}) = \sum_i w_{ji}^{(1)} x_i + b_j^{(1)}$$



Hidden layer activation:

$$\mathbf{h} = f(\mathbf{a})$$

Output layer activation of single hidden layer:

$$\hat{y}(\mathbf{x}) = o(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)})$$

Output layer activation of N hidden layers:

$$\hat{y}(\mathbf{x}) = o(\mathbf{W}^{(N+1)}\mathbf{h}^{(N)} + \mathbf{b}^{(N+1)})$$

Empirical Risk Minimization

- learning algorithms

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{M} \sum_{m} l(\hat{y}(\mathbf{x}^{(m)}; \boldsymbol{\theta}), y^{(m)}) + \lambda \Omega(\boldsymbol{\theta})$$

- loss function $l(\hat{y}(\mathbf{x}^{(m)}; \boldsymbol{\theta}), y^{(m)})$

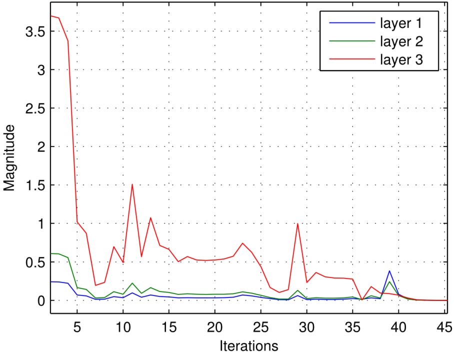
  for sigmoid activation $l(\boldsymbol{\theta}) = \sum_m \frac{1}{2} \left\| y^{(m)} - \hat{y}^{(m)} \right\|^2$
- regularizer $\lambda \Omega(\boldsymbol{\theta})$

Optimization

- Gradient calculation with Backpropagation
- Stochastic/Mini-batch gradient descent

Vanishing Gradient

- Training time increases as network gets deeper
- Gradient shrink exponentially and training end up local minima
- Caused by random initialization of network parameters



Unsupervised Layerwise Pre-training

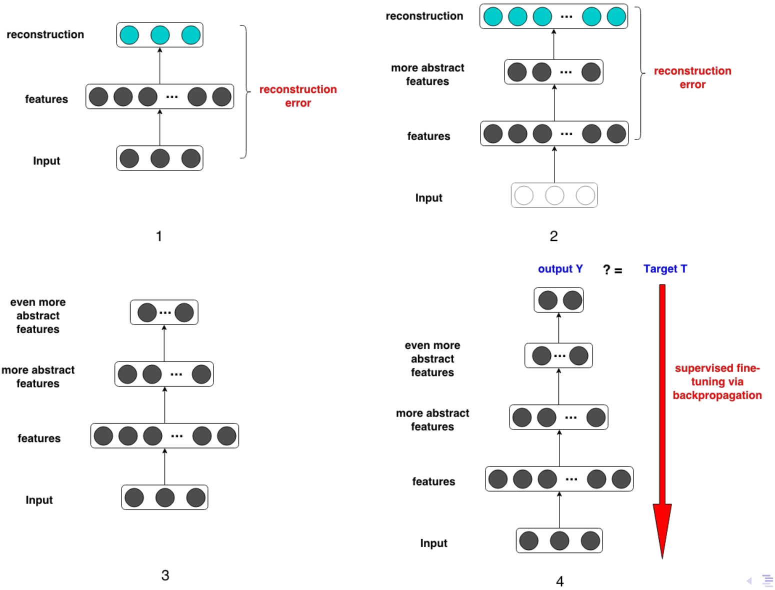
Vanishing Gradient

- Training time increases as network gets deeper
- Gradient shrink exponentially and training end up local minima
- Caused by random initialization of network parameters

Unsupervised layerwise pre-training

- Pretrain the deep network layer by layer to build a stacked auto-encoder
- Each layer is trained as a single hidden layer auto-encoder by minimizing average reconstruction error:
  $\min\ l_{AE} = \sum_m \frac{1}{2} \left\| \mathbf{x}^{(m)} - \hat{\mathbf{x}}^{(m)} \right\|^2$
- Fine-tuning the entire deep network with supervised training

Pre-training



Overfitting

- Huge amount of parameters in deep network
- Not enough data for training
- Poor generalization

Overfitting

- Huge amount of parameters in deep network
- Not enough data for training
- Poor generalization

Regularization

- Add weight penalization $\lambda \left\| \mathbf{w} \right\|_p$ to loss function

$$\text{arg } \min_{\boldsymbol{\theta}} \frac{1}{M} \sum_m l(\hat{y}(\mathbf{x}^{(m)}; \boldsymbol{\theta}), y^{(m)}) + \lambda \left\| \mathbf{w} \right\|_p$$

- In convex optimization:

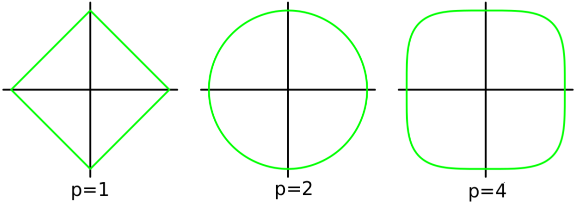
$$\text{arg } \min_{\boldsymbol{\theta}} \frac{1}{M} \sum_m l(\hat{y}(\mathbf{x}^{(m)}; \boldsymbol{\theta}), y^{(m)}), s.t. \left\| \mathbf{w} \right\|_p \leq C$$

Regularization

P-Norm

$$\|\mathbf{w}\|_p := \left(\sum_{n=1}^{n} |w_i|^p\right)^{1/p} = \sqrt[p]{|w_1|^p+, ..., +|w_n|^p}$$

Widely used: L1- and L2-regularization ($p = 1$ and $p = 2$)



P-Norm

$$\|\mathbf{w}\|_p := \left(\sum_{n=1}^{n} |w_i|^p\right)^{1/p} = \sqrt[p]{|w_1|^p+, ..., +|w_n|^p}$$

Widely used: L1- and L2-regularization ($p = 1$ and $p = 2$)

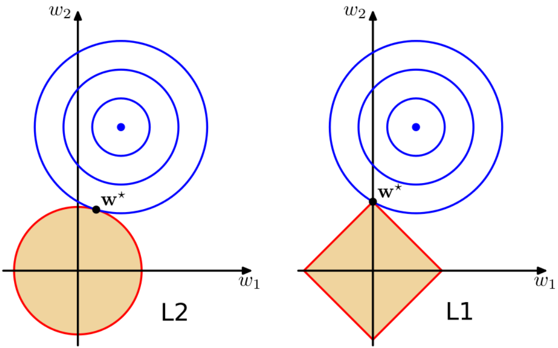


Table of Contents

Problems with RNN

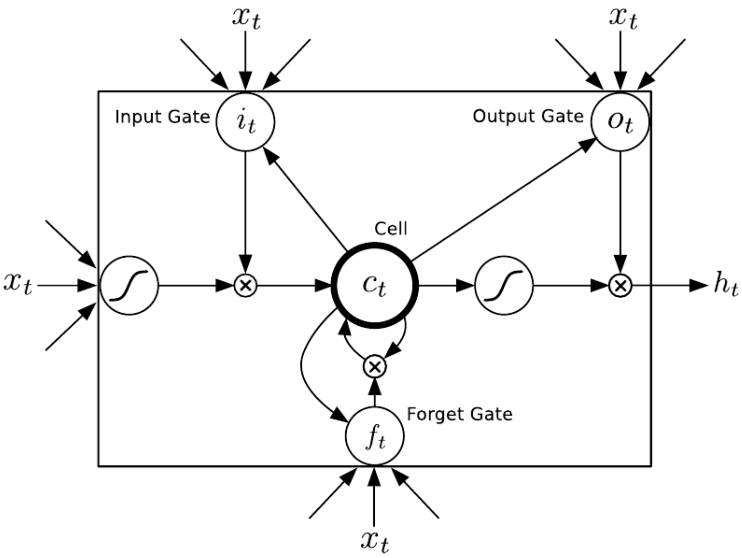
- gradient vanishing during backpropagation as time steps increases ($>$100)
- difficult to capture long-time dependency (which is required in emotion recognition)

Problems with RNN

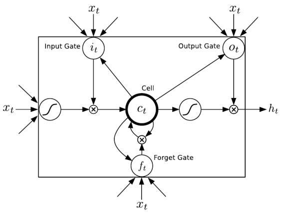
- gradient vanishing during backpropagation as time steps increases ($>$100)
- difficult to capture long-time dependency (which is required in emotion recognition)

S. Hochreiter and J. Schmidhuber, Lovol. 9, pp. 1735-1780, 1997.

Long short term memory



Long short term memory



$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

Features in LSTM

- gates are trained to learn when it shoud be open/closed.
- Constant Error Carousel
- preserve long-time dependency by maintaining gradient over time.

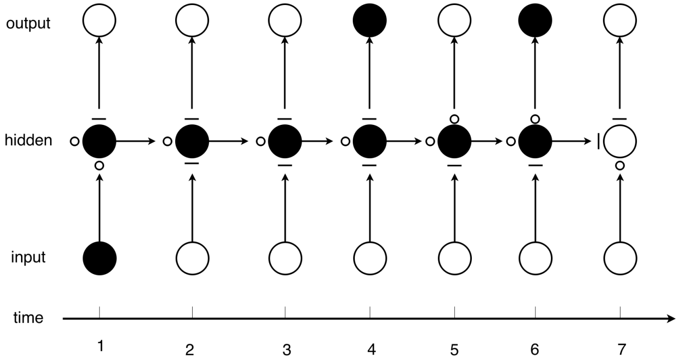


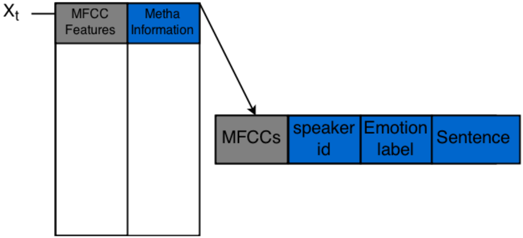
Table of Contents

Experiment Setup

EmoDB Database

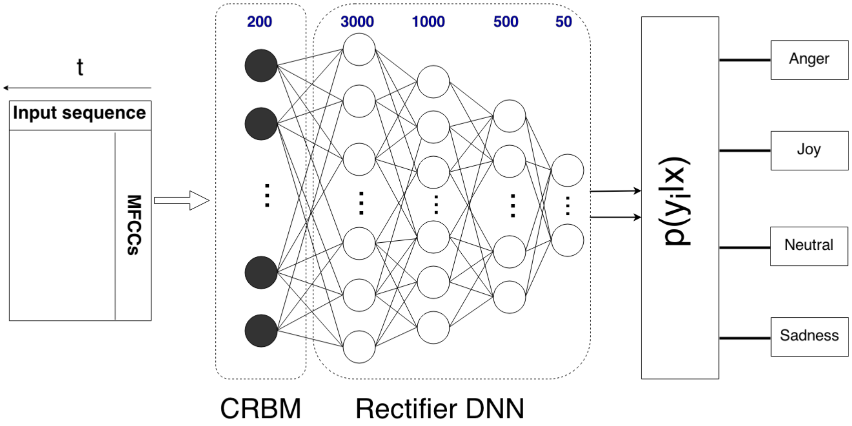
	Joy	Neutral	Sadness	Anger	Total
No. of sentences	71	79	62	127	339
Percent (%)	21	23.2	18.3	37.5	100

Data Structure



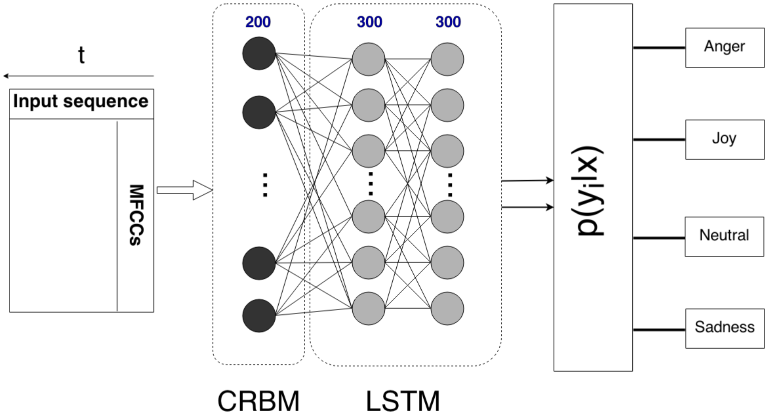
Experiment Setup

- CRBM-DNN



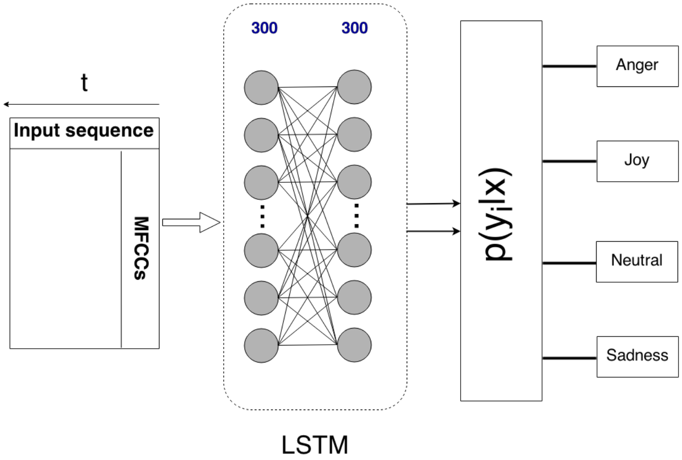
Experiment Setup

- CRBM-LSTM



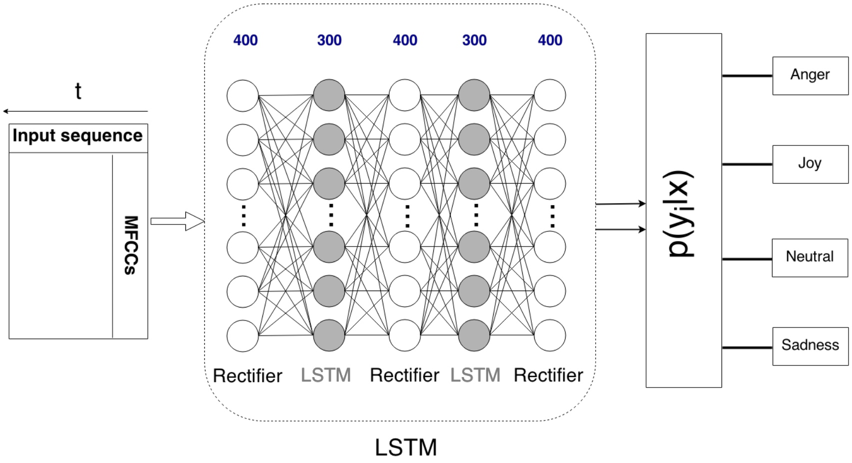
Experiment Setup

- LSTM



Experiment Setup

- LSTM with rectifier units



Result

Confusion matrix of CRBM-DNN result.

		Classfied			
		Joy	Neutral	Sadness	Anger
	Joy	57.7%	1.4%	0.0%	40.8%
True	Neutral	17.7%	54.4%	25.3%	2.5%
	Sadness	1.6%	27.9%	70.5%	0.0%
	Anger	39.4%	1.6%	0.0%	59.1%
		recognition rate:59.76%			

Result

Confusion matrix of CRBM-LSTM result.

		Classfied			
		Joy	Neutral	Sadness	Anger
	Joy	11.3%	9.9%	2.8%	76.1%
True	Neutral	0.0%	72.2%	17.7%	10.1%
	Sadness	0.0%	4.8%	88.7%	6.5%
	Anger	0.8%	1.6%	0.0%	97.6%
		recognition rate: 71.98%			

Result

Confusion matrix of pure LSTM result.

		Classfied			
		Joy	Neutral	Sadness	Anger
	Joy	66.2%	4.2%	0.0%	29.6%
True	Neutral	6.3%	79.7%	10.2%	3.8%
	Sadness	0.0%	19.7%	80.3%	0.0%
	Anger	12.6%	0.8%	0.0%	86.6%
		recognition rate: 81.59%			

Result

Confusion matrix of LSTM-Rectifier result.

		Classfied			
		Joy	Neutral	Sadness	Anger
	Joy	57.7%	7.0%	0.0%	35.2%
True	Neutral	6.3%	86.1%	6.3%	1.3%
	Sadness	0.0%	6.6%	93.4%	0.0%
	Anger	8.7%	0.0%	0.0%	91.3%
		recognition rate: 83.43%			

Table of Contents

Conclusion

- Capturing long-term dependencies is necessary for extracting speech emotion
- CRBM-DNN is inappropriate for modelling long-term dependencies (ER: 40.24%)
- LSTM is good at modelling long time dependencies
- Frame-based classification can also reach good result

Conclusion

- Capturing long-term dependencies is necessary for extracting speech emotion
- CRBM-DNN is inappropriate for modelling long-term dependencies (ER: 40.24%)
- LSTM is good at modelling long time dependencies
- Frame-based classification can also reach good result

Model	Temporal Dependency	Memory	Generaltive
DNN	-	-	-
RBM	-	-	✓
CRBM	✓	2-5	✓
AE	-	-	-
RNN	✓	1-100	-
LSTM	✓	1-1000	-

Conclusion

- Capturing long-term dependencies is necessary for extracting speech emotion
- CRBM-DNN is inappropriate for modelling long-term dependencies (ER: 40.24%)
- LSTM is good at modelling long time dependencies
- 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!