Deep Neural Network for Speech Emotion Recognition

—A Study of Deep Learning—

Zhuowei Han

Institut für Signalverarbeitung
und Systemtheorie

Universität Stuttgart

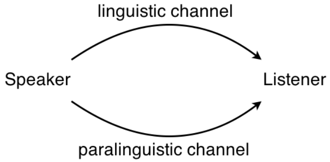
16/04/2015

Motivation

Why 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 Network Applications

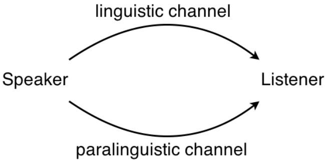
- Handwriting Digit Recognition
- Image Recognition

Motivation

Why 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 Network Applications

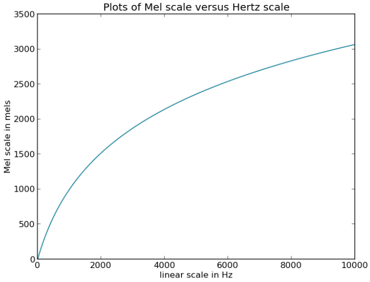
- Handwriting Digit Recognition
- Image Recognition

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



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

$$f_{Hz} = 700\left(\exp(f_{mel}/1125) - 1\right) \quad (2)$$

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 classfication

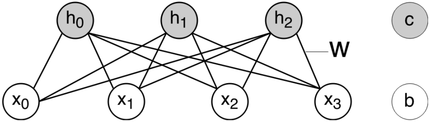
Table of Contents

Product of Experts

Character

- Generative model, capture data distrbution $P(\mathbf{x}|\boldsymbol{\theta})$
- Undirected graphical model, good at modeling high-dimensioanl data (speech emotion)
- 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
- Potential to be extend to capture temporal information
- Binary Units of input and outpu layer
- No interconnections within the same layer

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

$$P(\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h})$$
$$P(\mathbf{h}) = \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{h})$$
$$P(\mathbf{h}|\mathbf{x}) = \frac{P(\mathbf{x},\mathbf{h})}{P(\mathbf{x})}$$
$$P(\mathbf{x}|\mathbf{h}) = \frac{P(\mathbf{x},\mathbf{h})}{P(\mathbf{h})}$$
$$P(h_j = 1 \mid \mathbf{x}) = sigmoid(\sum_i x_i W_{ij} + c_j)$$
$$P(x_i = 1 \mid \mathbf{h}) = sigmoid(\sum_j W_{ij} h_j + b_i)$$

Conditional RBM

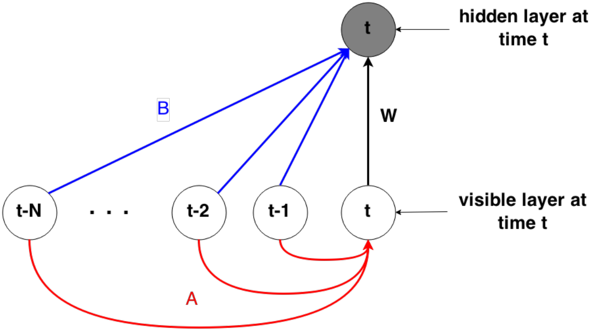
- Linear input units with independent Gaussian noise
- Real-valued data, e.g. spectral features

Conditional RBM

- Linear input units with independent Gaussian noise
- Real-valued data, e.g. spectral features

Conditional RBM

- Linear input units with independent Gaussian noise
- Real-valued data, e.g. spectral features



Conditional RBM

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

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

Optimization Method: Maximum Likelihood

$$P(\mathbf{x}) = \frac{e^{-\mathcal{F}(\mathbf{x})}}{Z}$$

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

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

Training of Energy-based Model

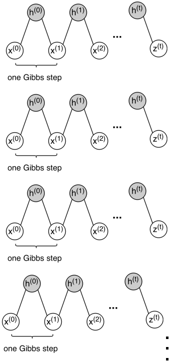
Optimization Method: Maximum Likelihood

$$P(\mathbf{x}) = \frac{e^{-\mathcal{F}(\mathbf{x})}}{Z}$$

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

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

Training of Energy-based Model



$$\begin{aligned}
\mathbf{x_1} &\sim \hat{P}(\mathbf{x}) \\
\mathbf{h_1} &\sim \hat{P}(\mathbf{h}|\mathbf{x}_1) \\
\mathbf{x_2} &\sim \hat{P}(\mathbf{x}|\mathbf{h}_1) \\
\mathbf{h_2} &\sim \hat{P}(\mathbf{h}|\mathbf{x}_2) \\
&\vdots \\
\mathbf{x_{t+1}} &\sim \hat{P}(\mathbf{x}|\mathbf{h}_t)
\end{aligned} \tag{3}$$

$t = 1$, Gibbs step $\rightarrow$ Constrastive Divergence

Table of Contents

Computing net-activation

$$\begin{aligned}\underline{z}_k^{(l+1)} &= \mathbf{W}^{(l)}\underline{a}_k^{(l)} + \underline{b}^{(l)} \\ \underline{a}_k^{(l+1)} &= \underline{\Phi}\left(\underline{z}_k^{(l+1)}\right) \\ \underline{\hat{y}}_k &= \underline{a}_k^{(ol)}\end{aligned}$$

- Arbitrary non-linear mapping from $\underline{x}_k$ to $\underline{\hat{y}}_k$ possible
- Relation $N \Leftrightarrow$ Complexity
- Deep Architectures $(l\uparrow)$ more efficient than shallow ones $(l\downarrow, N_l\uparrow)$

Training objective

$$J(\mathbf{W}, \underline{b}) = \sum_{\forall k} \frac{1}{2} ||\underline{y}_k - \hat{\underline{y}}_k||^2 + \frac{\lambda}{2} \sum_{\forall l} ||\mathbf{W}^{(l)}||_F^2 \quad (4)$$

$$\mathbf{W}, \underline{b} = \arg\min_{\mathbf{W}, \underline{b}} J(\mathbf{W}, \underline{b}) \quad (5)$$

Numerical minimization

- Gradient calculation with Backpropagation
- Stochastic gradient descent
- **L**imited memory **B**royden-**F**letcher-**G**oldfarb-**S**hanno (L-BFGS)

Problems

- Optimization problem non-convex
  $\Rightarrow$ getting stuck in poor local minima
- Diffusion of gradients
- Large p small n problem $\Rightarrow$ overfitting

Solutions

- Layerwise Pre-training

- Layerwise Pre-training

Table of Contents

Concepts of RNN

- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.
- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

Concepts of RNN

- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.
- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

Concepts of RNN

- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

Concepts of RNN

- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.
- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

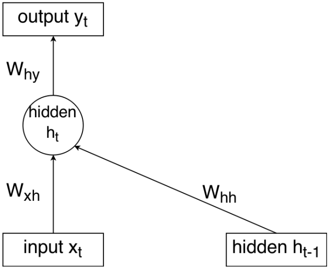
Concepts of RNN

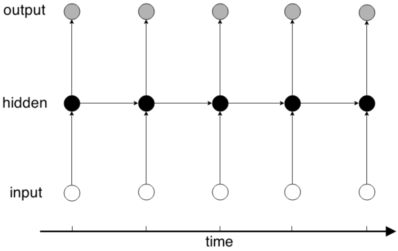
- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.
- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

Concepts of RNN

- modelling sequential data, emotion in speech .
- Same Structure as MLP but differs from feed-forward network, enabling nonlinear mapping.
- Feedback connection between previous hidden units and current hidden units, enabling memory past hidden state.
- Potentially to model arbitary dynamic system.
- Trained with **b**ack**p**ropagation **t**hrough **t**ime (BPTT)

Concepts of RNN





From RNN to LSTM

Problems with RNN

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

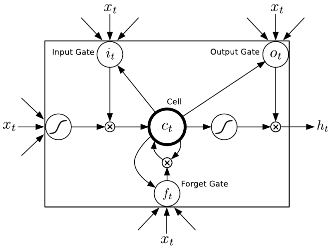
Solutions

-

Long short term memory

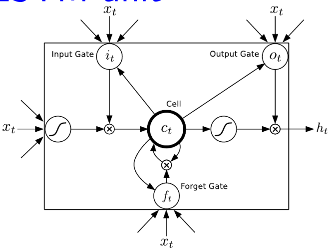
S. Hochreiter and J. Schmidhuber, Lovol. 9, pp. 1735-1780, 1997.

LSTM unit



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

LSTM unit



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

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.

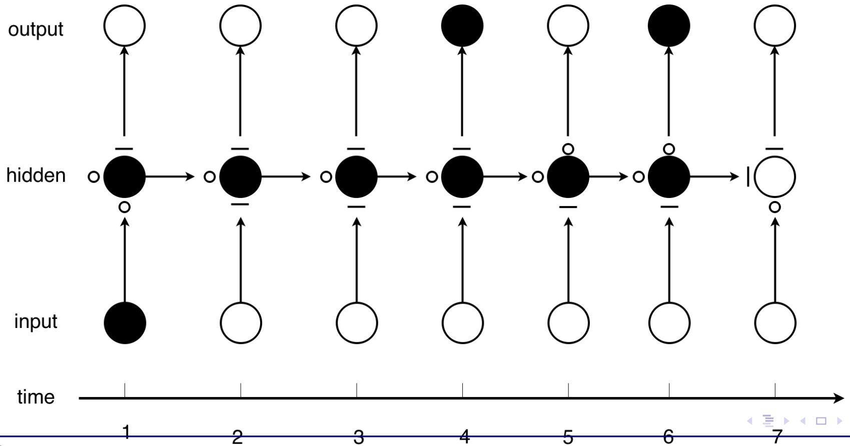
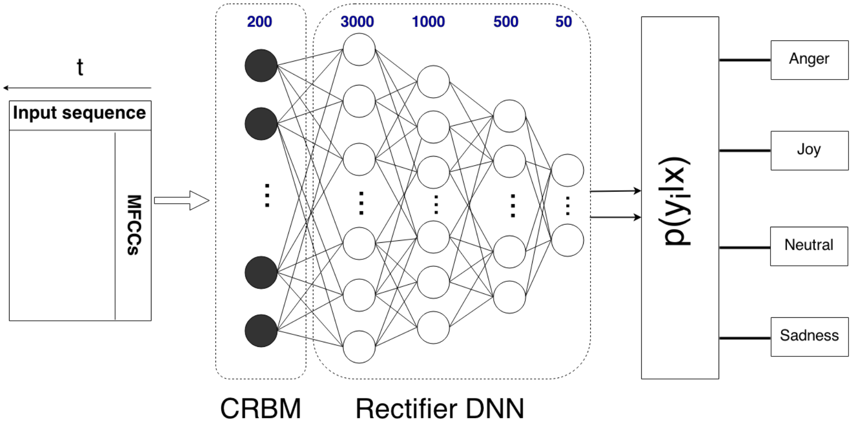


Table of Contents

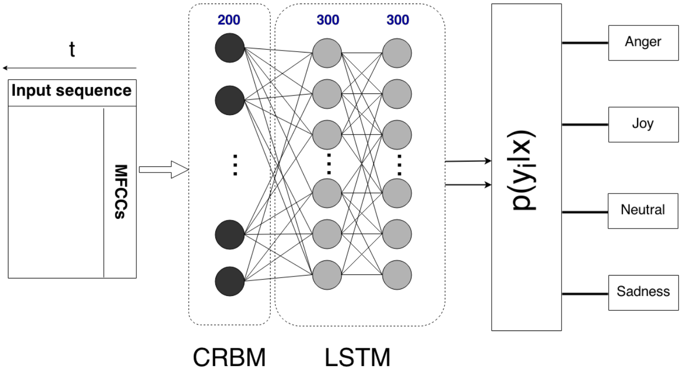
Experiment Setup

- CRBM-DNN



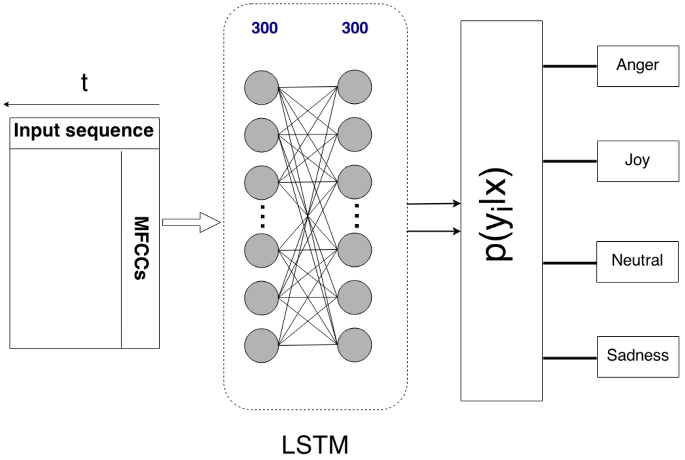
Experiment Setup

- CRBM-LSTM



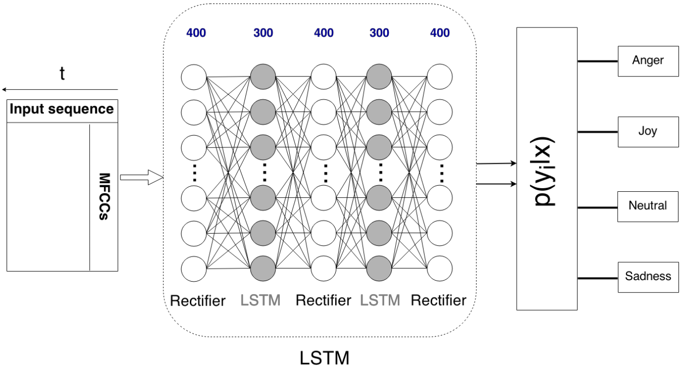
Experiment Setup

- 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%

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

- Model with long-term dependencies shall be used for speech emotion.
- CRBM is appropriate for short time modelling, stacked CRBM can model longer dependency
- LSTM can model long time dependency, get in the task.
- 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
- Traing CRBM with more/larger data base
- Second order optimization to speed up learning process
- Bi-directional LSTM, capturing future dependencies

End

Thank You!