Deep Network for Speech Emotion Recognition

—A Study of Deep Learning—

Zhuowei Han

Institut für Signalverarbeitung
und Systemtheorie

Universität Stuttgart

16/04/2015

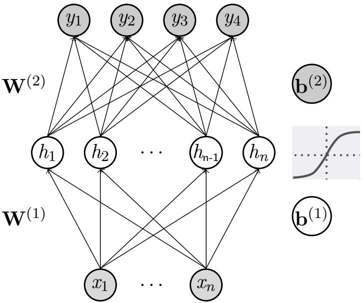
Table of Contents

Table of Contents

Structure and Function

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

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

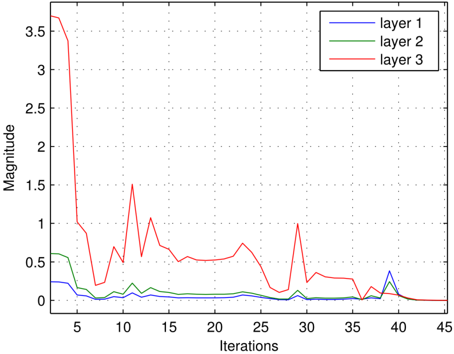
- loss function $l(\hat{y}(\mathbf{x}^{(m)};\boldsymbol{\theta}), y^{(m)})$
  for sigmoid activation $l(\boldsymbol{\theta}) = \sum_m \frac{1}{2}\left\| y^{(m)} - \hat{y}^{(m)} \right\|^2$
- regularizer $\lambda\Omega(\boldsymbol{\theta})$

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

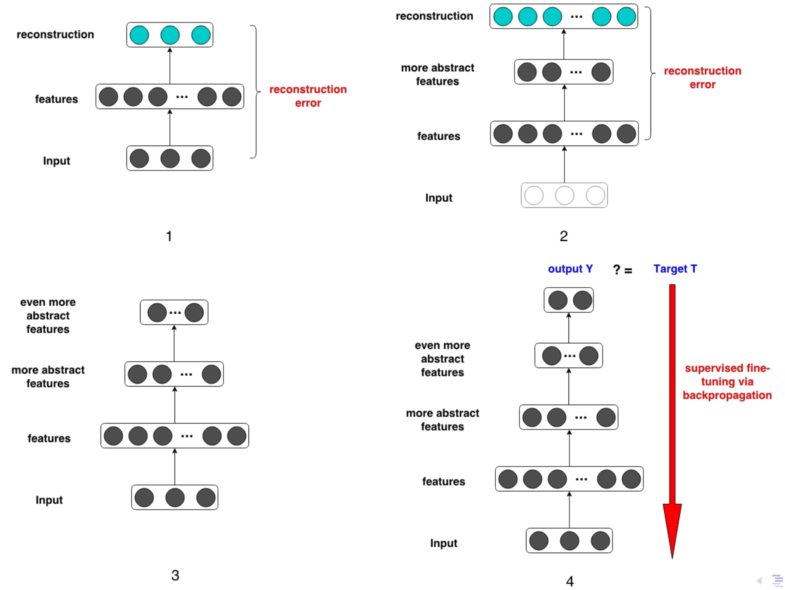
Vanishing Gradient

- 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

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

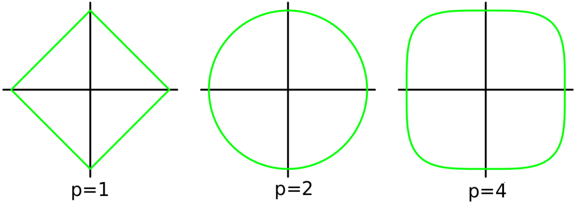
- In convex optimization:

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

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 $p = 2$)

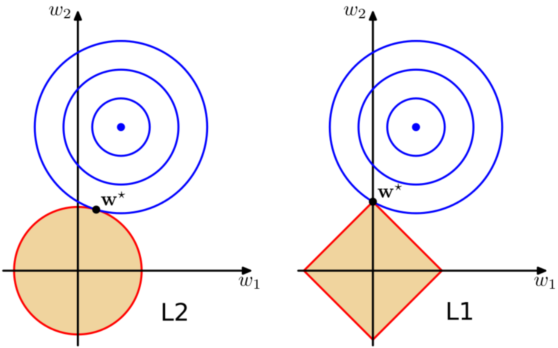


Table of Contents

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!