Embedding structured sequences with rNN/dNN

Emotion recognition
from speech

Motivation

Structured data

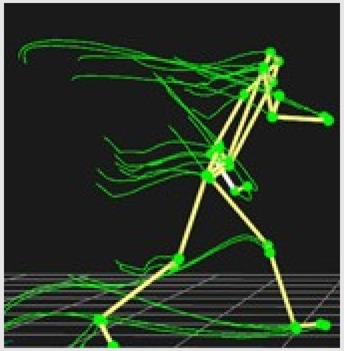
Data that consists of several parts, and not only the parts themselves contain information, but also the way in which the parts belong together.

Automatic Translation

Source X	**Target Y**
Word sequence length T	Word sequence length T'
„at the end of the“	„a la fin du“
„parts of the world“	„gions du monde“
„the past few days“	„cours des tout derniers jours“

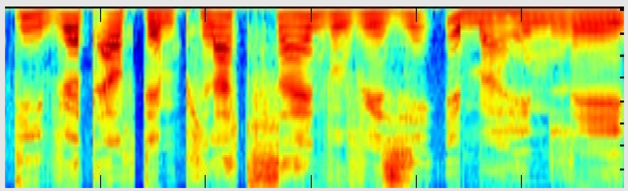
Motivation

Guesture recognition

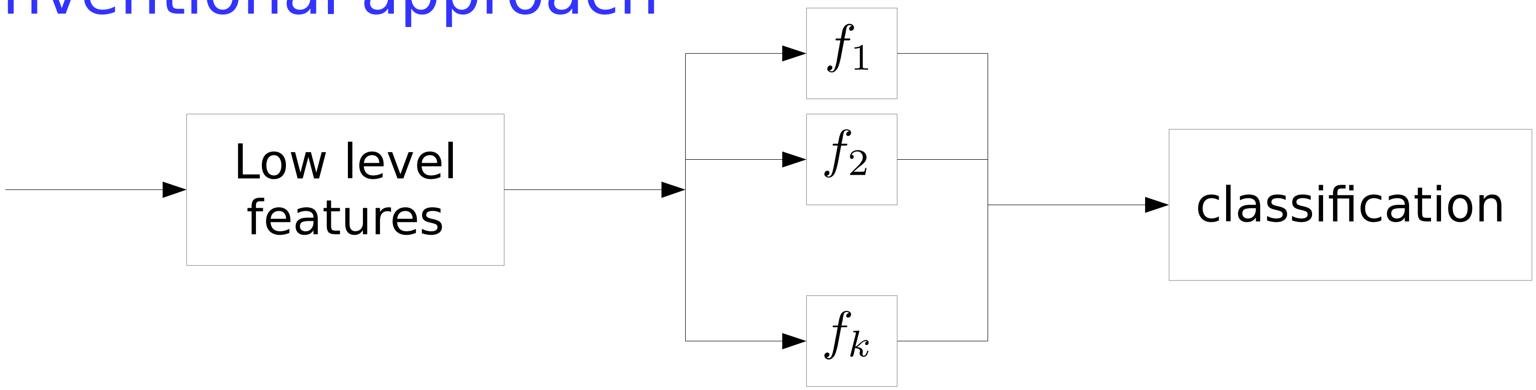
Source X	Target Y
Motion sequence	Guesture sequence
Velocity and position of POIs 	Walking, Running, Answering phone call, ...

Motivation

Emotion recognition from speech

Source X	**Target Y**
Spectral sequence	Emotion label
Spectrogram of sentence 	Neutral, Happy, Angry, Bored, ...

Conventional approach



- Ignores temporal order
- Computational complex

Overview

Dealing with structured data

- Conventional approach
- Approach using recurrent Neural Networks

The Recurrent Neural Network (rNN)

- Structure
- Embedding and reconstruction
- Long-time Short-time Memory Cells

Network architecture for Emotion recognition

- Problems with EMODB database
- Using cross-database training for regularization
- Used optimization methods

Results on EMODB

Dealing with structured data

Conventional approach

Find in ML sense:

$$f : X \to Y$$

X ...input sequence

Y ...label sequence

Define Energy function:

$$E(x, y) = \sum_i w_i^T \phi'(x_i, y_i) + \sum_{i,j} w_{i,j}^T \phi(y_i, y_j)$$

$\phi(x_i, y_i)$, $\phi(y_i, y_j)$ ...compatibility functions

Output probability:

$$p(x, y) = \frac{1}{Z} \exp -E(x, y)$$

Inference via ML estimation:

$$f(x) = \arg\min E(x, y) + \log Z$$

Problems:

- Inference from x to y is intractable in general (because of partition function Z)
- Restrictions on signal length

Dealing with structured data

Mathematical description

Wanted:

$f : X \rightarrow H$ ...embedding function
$g : H \rightarrow Y$ ...generative model

Model the conditional probabilities:

$$p(y_k, y_{k-1}, ..., y_0 | x_l, x_{l-1}, ..., x_0) = \underbrace{p(y_k, y_{k-1}, ..., y_0 | h)}_{generative} \underbrace{p(h | x_l, x_{l-1}, ..., x_0)}_{embedding}$$

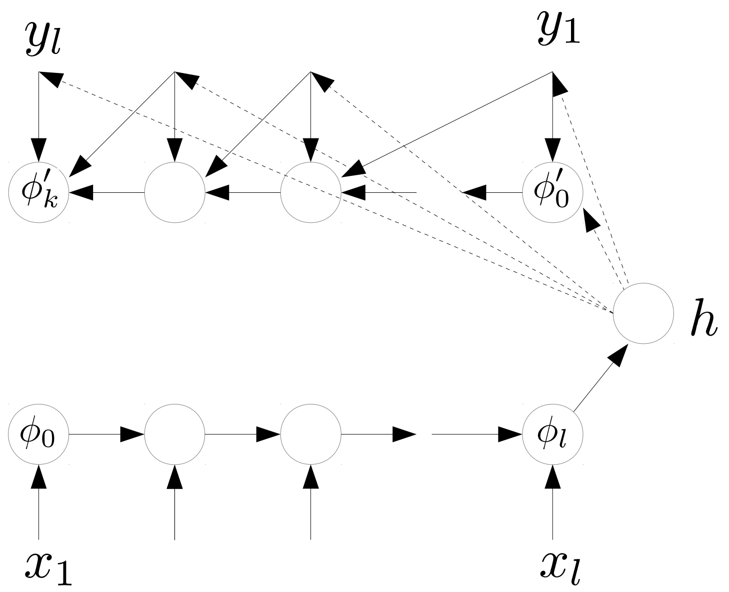
$$= \frac{1}{Z} \exp - \sum_k w_k^T \phi'_k(y_k, y_{k-1}, h_l) - w_e^T \phi_l(h_{l-1}, x_l)$$

Advantages:

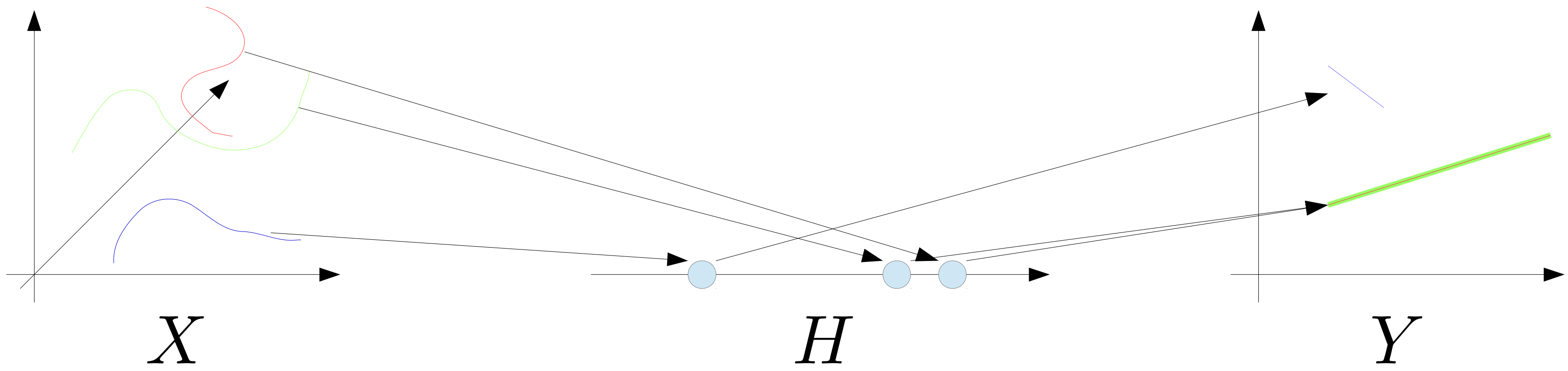
- No restriction on signal lengths
- Exact inference for h with given sequence of x is possible
- Exact inference for sequence y with given h is possible

Dealing with structured data

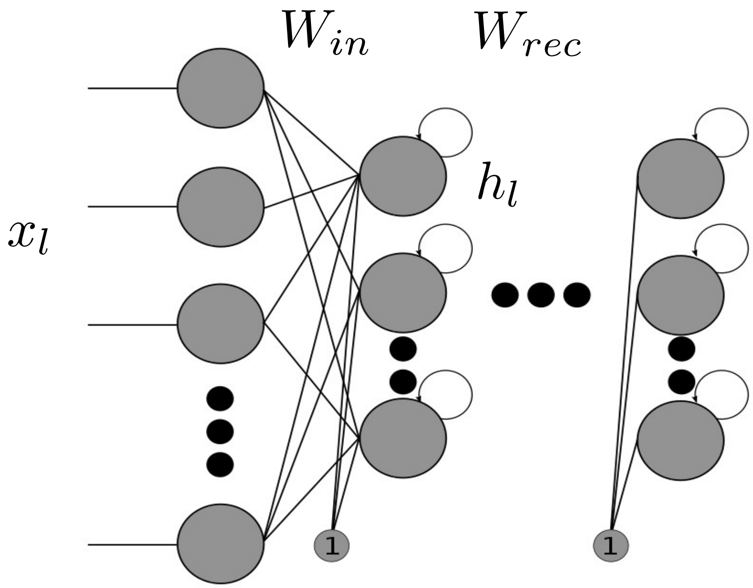
Learn sequence representations



Intuitive explanation



The Recurrent Neural Network (rNN)



Mapping function:

$$h_l = W_{rec}\sigma(h_{l-1}) + W_{in}x_l + b$$

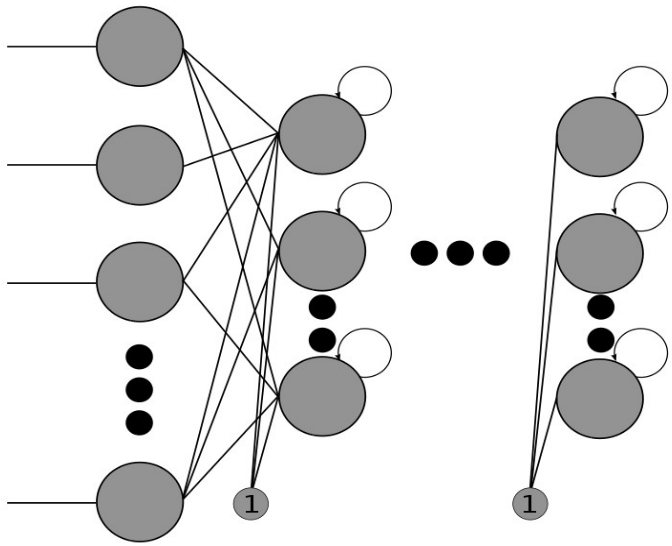
Using softmax activation:

$$p(h_j = 1|x_l, h_{l-1}, ..., h_1) = \frac{\exp w_j h_l}{\sum_{j'} \exp w_{j'} h_l}$$

$$= \frac{1}{Z} \exp w_e \phi_l(h_{l-1}, x_l)$$

→ RNN can be used as compatibility function

The Recurrent Neural Network (rNN)



$$\frac{\partial r}{\partial \theta} = \sum_l \frac{\partial r_l}{\partial \theta}$$

$$\frac{\partial x_l}{\partial x_k} = ... = \prod_{l \geq i \geq k} W_{rec}^T diag(\phi'(x_{i-i}))$$

Inference using the rNN:

$$p(y|h) = \prod_k p(y_k|y_{k-1}, ..., y_1, h)$$

Training:

$$r = E[l(\hat{y}_n, y_n)p(y_n|x_n)]$$

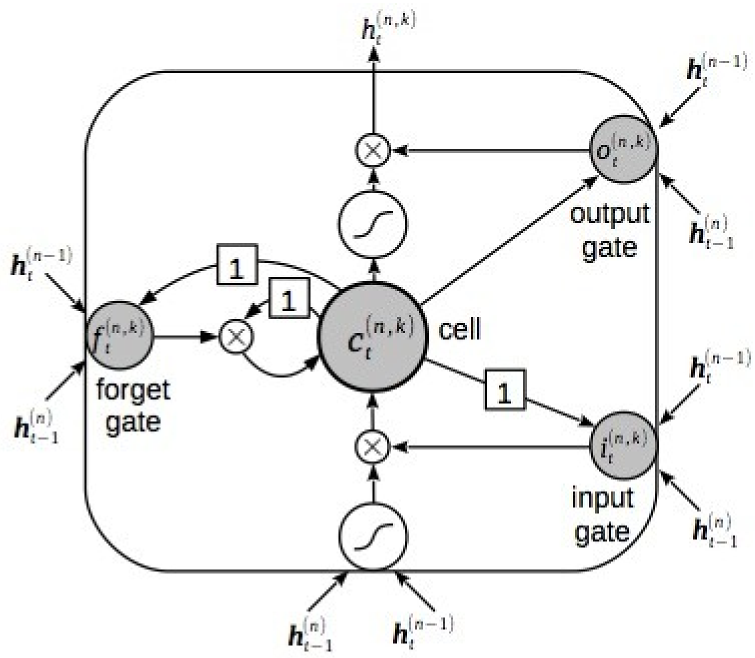
$$\theta = \arg\max \frac{1}{N} \sum \log p_\theta(y_n|x_n) + \beta||\theta||^2$$

Practical problems:

- Vanishing gradients
- Only short-time dependencies can be learned

How to implement the rNN

LSTM units



$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_1)$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

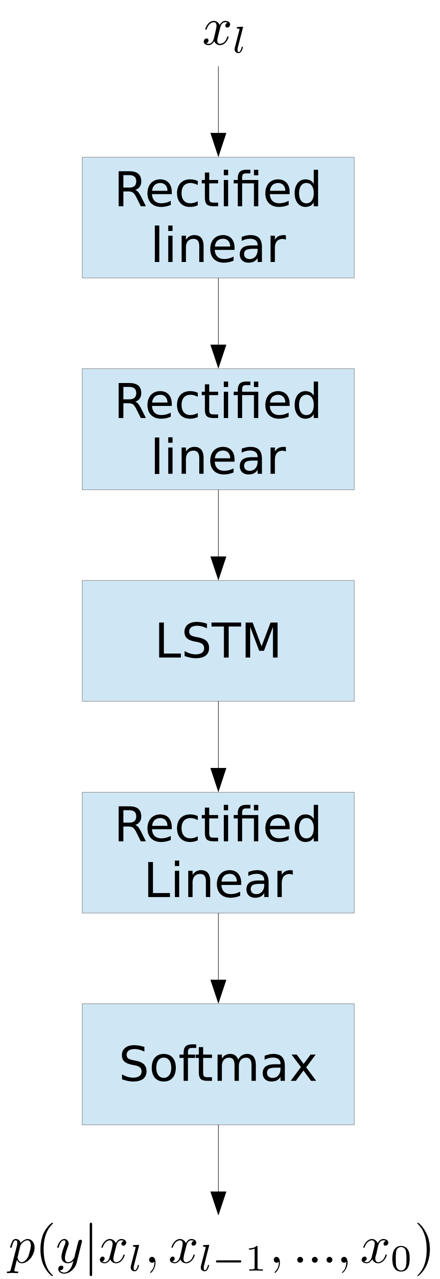
$$C_t = i_t \tilde{C}_t + f_t C_{t-1}$$

$$h_t = o_t + \tanh C_t$$

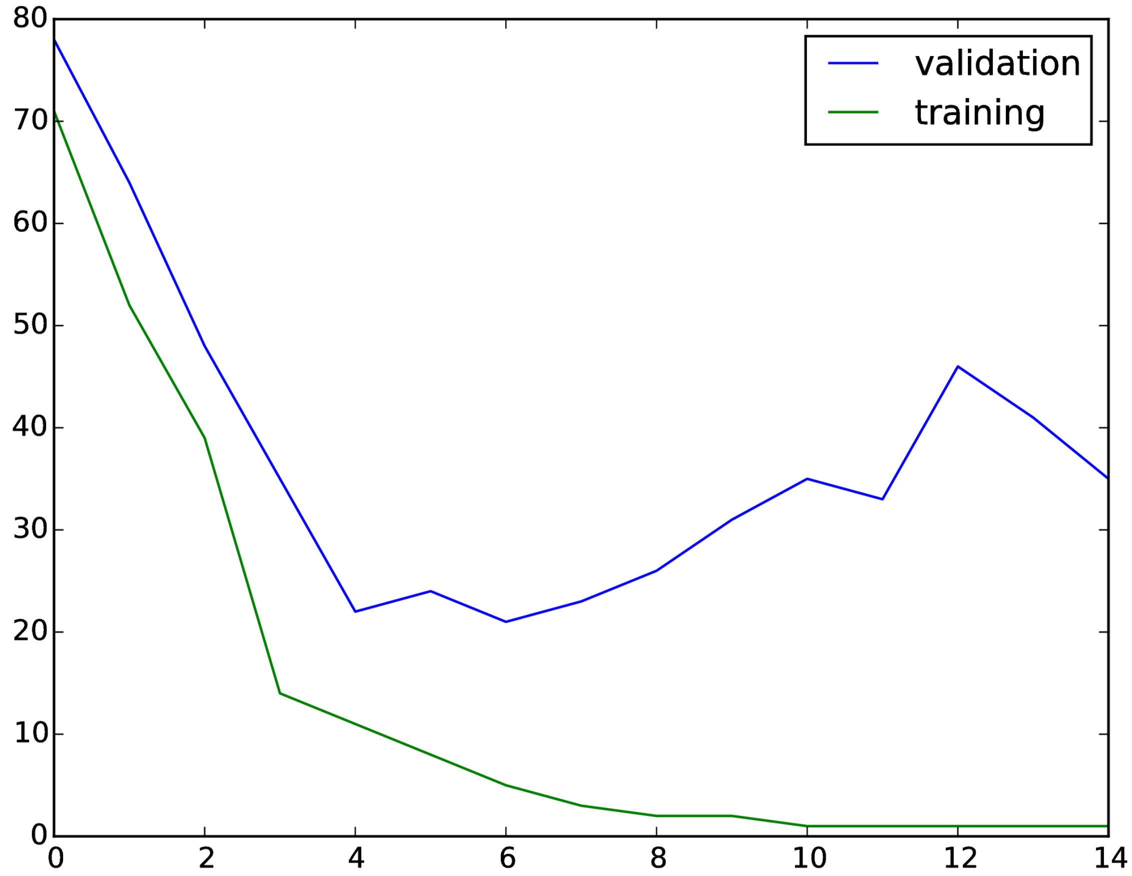
Advantages over conventional rNN

- It can actually be trained!
- No vanishing or exploding gradients during BPTT
- Can model long-time dependencies due to error carousel

Setup for Emotion Recognition



Case of 4 emotions on EMODB



Problem of overfitting

Using cross-database training for regularization

- EMODB contains very few training examples (488 sentences)

- Embedding layer may learn very complex representation (no natural clustering)

- Following layers have to map those representations to the labels

  → results in a very complex decision boundary

Idea:

- There is a second database recorded at the ISS (with 2321 sentences)
- Use this database to regularize the network

Regularization of the embedding layer

Introducing an additional regularization term

For Lossfunction L and similarity matrix W

$$r(x) = r(y_n, \hat{y}_n) + \lambda \sum_{i,j} L(h(x_i, \theta), h(x_j, \theta), W_{ij})$$

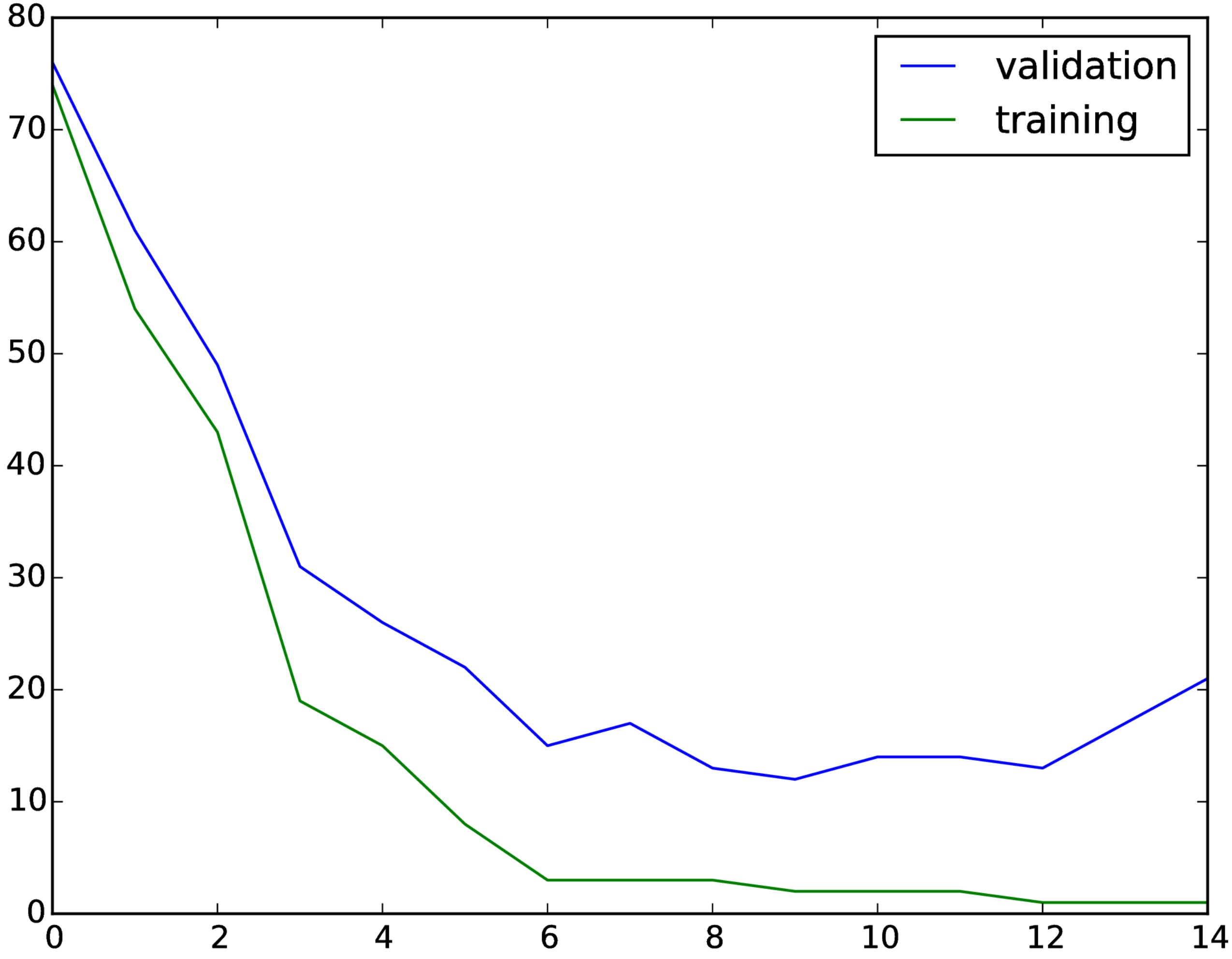
x_i ...in-domain sequence
x_j ...additional out-of-domain sequences

Use margin-based loss to enforce clustering (Hadsell et al. 2006)

$$L(h_i, h_j, W_{ij}) = \begin{cases} ||h_i - h_j||^2 & \text{if } W_{ij} \equiv 1 \\ \max(0, m - ||h_i - h_j||^2) & \text{if } W_{ij} \equiv 0. \end{cases}$$

Regularization of the embedding layer

Case of 4 emotions



Results for 4 emotions

Confusion matrix

	FR	NT	TR	WT
FR	60	2	0	38
NT	3	94	3	0
TR	0	7	93	0
WT	3	1	0	96

Training and validation error rates

$E_t = 1\%$ $\qquad$ $E_v = 12\%$

Comparison to conventional approach

$E_v = 15.5\%$ (Altun and Polat 2009)

Results for 6 emotions

Confusion matrix

	AN	FR	GW	NT	TR	WT
AN	74	2	9	4	2	9
FR	19	44	0	4	0	33
GW	4	2	62	20	12	0
NT	8	0	15	74	2	1
TR	0	0	11	4	85	0
WT	5	6	0	0	0	89

Training and validation error rates

$E_t = 3\%$ $\qquad$ $E_v = 25\%$

Comparison to conventional approach

$E_v = 14\%$ (Masterthesis Gruber)

Conclusion Recognition

- Deep Neural Networks with Recurrent Embedding Layers can be used for emotion recognition from speech
- Good results can be achieved in case of the 4 base emotions
- Tends to overfitting for higher order of emotions
- Using data based regularization on embedding layer can reduce overfitting