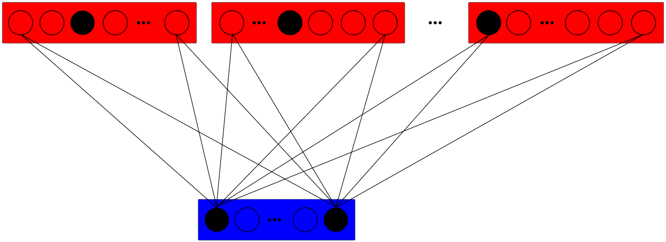
Continuous Space Language Models using Restricted Boltzmann Machines

Jan Niehues and Alex Waibel

Institute for Anthropomatics



Motivation

N-gram based language models

- Make use of large corpora
- Can be trained efficiently

Domain Adaptation

- Language models trained on small corpora are needed
- Language model has to back-off to smaller contexts
- Continuous space language models always use same context size
- Longer training time not as problematic
- Aim: Application during decoding

Related work

First approaches of neural network language models using word categories in the 90s (Nakamura et al.,1990)

CSLM for speech recognition (Bengio et al., 2003, Schwenk et al., 2002)

SOUL Language model by Le et al., 2011

RBM-based language model (Mnih and Hinton, 2007)

Overview

Restricted Boltzmann Machine

RBMs for Language Model

- Calculating probabilities efficiently

Evaluation

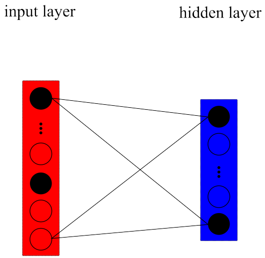
- German-English
- English-French

Conclusion

Restricted Boltzmann Machine

Layout

- 2 layer neuronal net
- Binary units
- Weighted connections between the layers
- No connections within the layer



Restricted Boltzmann Machine

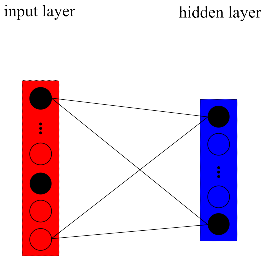
Layout

Probability

- Probability defined by the energy

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \quad (1)$$

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (2)$$



- Problem: Hidden state needs to be known

Restricted Boltzmann Machine

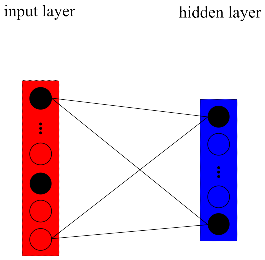
Layout

Probability

- Probability defined by the energy

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \quad (1)$$

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (2)$$



- Problem: Hidden state needs to be known

Restricted Boltzmann Machine

Layout

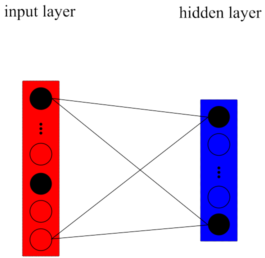
Probability

- Probability defined by the energy
- Probability using free energy

$$p(v) = \frac{1}{Z} e^{-F(v)} \tag{3}$$

$$F(v) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} log(1 + e^{x_j}) \tag{4}$$

$$x_j = b_j + \sum_{i \in visible} v_i w_{ij} \tag{5}$$



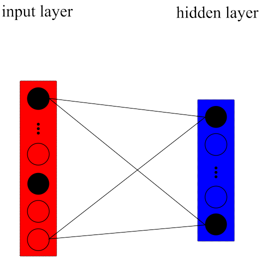
Restricted Boltzmann Machine

Layout

Probability

Training

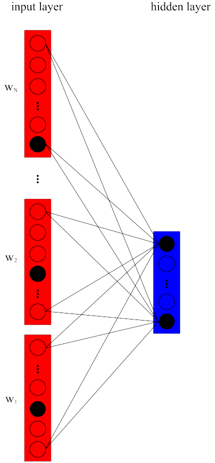
- Contrastive Divergence
- Increase probability of seen training example



RBMs for Language modeling

Layout

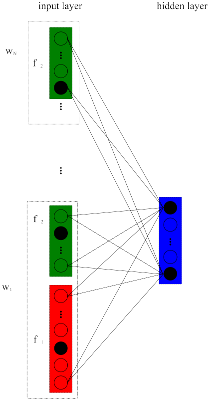
- N input blocks
- Each of the block has V units
- H hidden units
- N*V*H weights and N*V + H biases



RBMs for Language modelling

Layout

- N input blocks
- Each of the block has V units
- H hidden units
- N*V*H weights and N*V + H biases
- Easy integration of additional word factors
- Replace block by W sub blocks

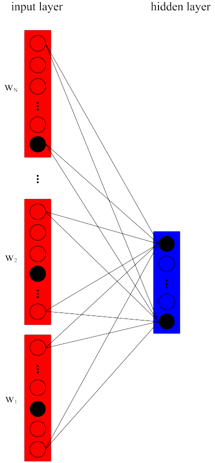


RBMs for Language modeling

Layout

Training

- Contrastive Divergence
- Random order of the n-grams in training data
- 1 iteration over all the data
- 1 iteration of Gibbs sampling for collection samples



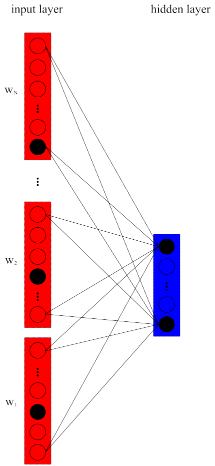
N-gram Probability

Use language model in the decoder
$\rightarrow$ efficient calculation needed

- Use free energy instead of probability
    - No normalization needed
- Complexity

$$F(v) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} log(1 + e^{x_j})$$

$$x_j = b_j + \sum_{i \in visible} v_i w_{ij}$$



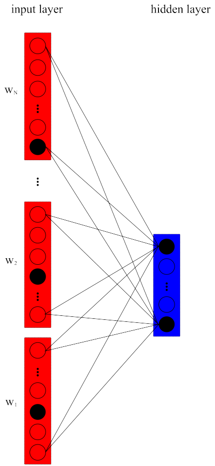
N-gram Probability

Use language model in the decoder
$\rightarrow$ efficient calculation needed

- Use free energy instead of probability
  - No normalization needed
- Complexity

$$F(v) = -O(N) - \sum_{j \in hidden} log(1 + e^{x_j})$$

$$x_j = b_j + O(N)$$

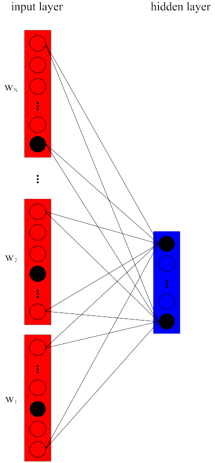


N-gram Probability

Use language model in the decoder
$\rightarrow$ efficient calculation needed

- Use free energy instead of probability
  - No normalization needed
- Complexity

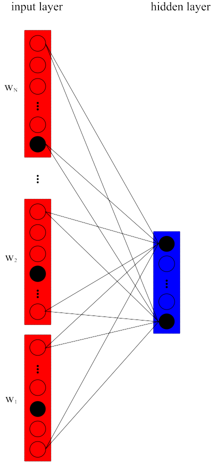
$$F(v) = -O(N)$$
$$- O(H * N)$$
$$x_j = b_j + O(N)$$



N-gram Probability

Use language model in the decoder
$\rightarrow$ efficient calculation needed

- Use free energy instead of probability
  - No normalization needed
- Calculate free energy in $O(H * N)$
- Independent of the vocabulary size



Sentence Probability

Feature describing probability of the whole sentence

$$\sum_{j \in L+N-1} F\left(w_{j-N+1} \dots w_j\right) \tag{6}$$

- Sum over free energy of all n-grams

<s>	<s>	<s>	I	I	go	home	</s>
<s>	<s>	I	go	**go**	**home**	**</s>**	**</s>**
<s>	I	go	home	**home**	**</s>**	**</s>**	**</s>**

Sentence Probability

Feature describing probability of the whole sentence

$$\sum_{j \in L+N-1} F\left(w_{j-N+1} \dots w_j\right) \tag{7}$$

- Sum over free energy of all n-grams
- proportional to approximation of the geometric mean of all language model probabilities with contexts $\leq$ N
  - terms depending on the sentence length are ignored
- easy to integrate into the decoder

Evaluation

System description

German to English

- TED Translation task
- Phrase-based SMT system
- Training data: EPPS, NC, TED, BTEC
- In-domain: TED
- Additional word factor: automatic word classes generated by MKCLS (50 classes)

English to French

- System built during IWSLT 2012 Evaluation

Evaluation

German to English

Table: *Experiments on German to English*

System	Iterations	BLEU Score Dev	BLEU Score Test
Baseline		26.31	23.02
+ RBMLM H32	1	27.39	23.82
+ RBMLM H32	10	27.61	24.47
+ FRBMLM H32	1	27.54	24.15
Baseline+NGRAM		27.45	24.06
+ RBMLM H32	1	27.64	24.33
+ RBMLM H32	10	27.95	24.38
+ FRBMLM H32	1	27.80	24.40

Evaluation

Number of hidden units

Table: *Experiments using different number of hidden units*

System	Hidden Units	BLEU Score Dev	BLEU Score Test
NGRAM		27.09	23.80
RBMLM	8	25.65	23.16
	16	25.67	23.07
	32	26.40	23.41
	64	26.12	23.18

Evaluation

Karlsruhe Institute of Technology

Training iterations

Table: *Experiments using different number of training iterations*

System	Iterations	No Large LM		Large LM	
		Dev	Test	Dev	Test
NGRAM		27.09	23.80	27.45	24.06
RBMLM	1	26.40	23.41	27.39	23.82
	5	26.72	23.38	27.40	23.98
	10	26.90	23.51	27.61	24.47
	15	26.57	23.47	27.63	24.22
	20	26.16	23.20	27.49	24.30

Evaluation

Karlsruhe Institute of Technology

English to French

Table: *Experiments on English to French*

System	BLEU Score	
	Dev	Test
Baseline	28.93	31.90
RBMLM	28.99	31.76
FRBMLM	29.02	32.03

Conclusion

- Continuous space language model using Restricted Boltzmann Machines
- Approximations to efficiently calculate language model score
- Language model score is independent of vocabulary size
- Integration into decoding
- Factor language model
- Experiments on two TED translation tasks
  - Detailed experiments on German-English
  - Slight improvements on English-French

Example

$$P_3(S) = P(I|<s><s>) * P(go|<s>I) * P(home|Igo) * P(</s>|goho$$

$$P_2(S) = P(I|<s>) * P(go|I) * P(home|go) * P(</s>|home)$$

$$P_1(S) = P(I) * P(go) * P(home)$$

$$\begin{aligned}
P(S) &= \sqrt[3]{P_3(S) * P_2(S) * P_1(S)} \\
&= \frac{P(<s><s>I)}{P(<s><s>)} * \frac{P(<s>I)}{P(<s>)} * P(I) \\
&* \frac{P(<s>Igo)}{P(<s>I)} * \frac{P(Igo)}{P(I)} * P(go) \\
&* \frac{P(Igohome)}{P(Igo)} * \frac{P(gohome)}{P(go)} * P(home) \\
&* \frac{P(gohome</s>)}{P(gohome)} * \frac{P(home</s>)}{P(home)}
\end{aligned}$$

Example

$$
\begin{aligned}
&= \frac{P(<s><s>I)}{P(<s><s>)} * \frac{P(<s>I)}{P(<s>)} * P(I) \\
&* \frac{P(<s>Igo)}{P(<s>I)} * \frac{P(Igo)}{P(I)} * P(go) \\
&* \frac{P(Igohome)}{P(Igo)} * \frac{P(gohome)}{P(go)} * P(home) \\
&* \frac{P(gohome</s>)}{P(gohome)} * \frac{P(home</s>)}{P(home)} \\
&= P(<s><s>I)P(<s>Igo) * P(Igohome) \\
&* P(gohome</s>) * P(home</s>) \\
&= P(<s><s>I)P(<s>Igo) * P(Igohome) \\
&* P(gohome</s>)P(home</s></s>)
\end{aligned}
$$