Modelling Language with Recurrent Neural Networks

Thomas Hummel, Sebastian Springenberg and Abtin Setyani
Natural Language Systems Department, UHH Hamburg, Hamburg, Germany

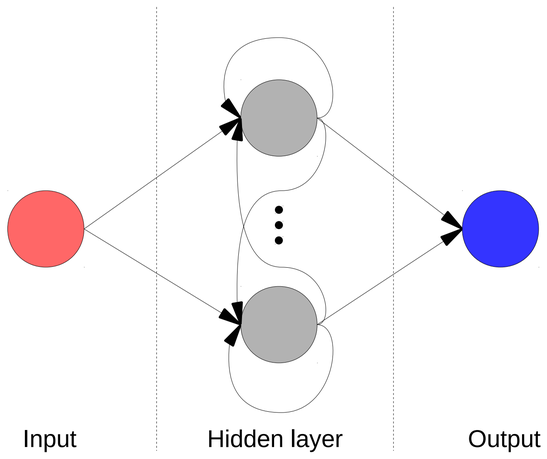
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Introduction

- A language model
  - Captures the prominent statistical characteristics of the distribution of sequences of words in a natural language
  - Allows to make probabilistic predictions of next words given preceding ones
- Problem: curse of dimensionality
  - Sequence of 10 words with vocabulary of 10 000 words has 10^{40} different possibilities
- Approaches: N-gram models, Neural networks, ...

Modelling Language with Recurrent Neural Networks

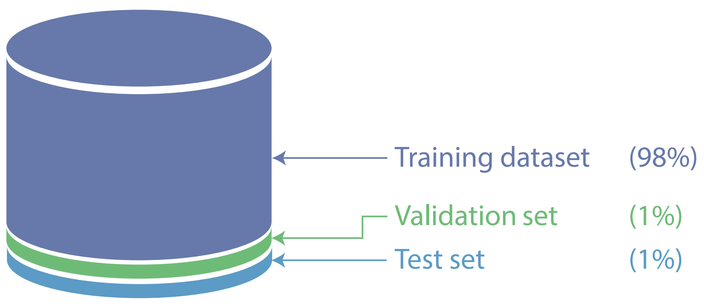
- Recurrent neural networks (RNNs):
  - Represent time recursively by recurrent connections among hidden units associated with a time delay
  - Building up of an internal context of previous states
- RNNs can naturally model the sequential character of language
- Many different parameters



$\Rightarrow$ **Which parameters influence the performance most?**

Dataset

- European Parliament Proceedings Parallel Corpus (Europarl) [1]
- Only a monolingual English subset of the Corpus used for the following language modelling tasks
- Dataset consists of 2 218 201 words constructing 53 974 751 English sentences
- Dataset partitioning:
  - 98 % of the data reserved as training data
  - 1 % of the data used for validation, 1 % for testing
  - Training dataset fractions of different size are used in the following experiments to test effects of training data size on model performance



Experiments

- Implementation used: Faster RNNLM (HS/NCE) toolkit (`https://github.com/yandex/faster-rnnlm`)

Experiment 1: Perplexity with respect to training data size

- Train a RNN with hidden layer size of 100 NCE GRU units on dataset fractions of different size
- Investigate the effect of training set size on the networks performance in terms of perplexity

Experiment 2: Perplexity obtained by different network architectures

- Train RNNs with different architectures on 1 % of the dataset
- Investigate the effects of different activation functions and hidden layer sizes
- RNNs used:
  - Rectified linear unit (relu) activation function, varying hidden layer size
  - Sigmoid activation function, varying hidden layer size
  - Gated recurrent unit (gru) activation function, varying hidden layer size

Experiment 3: Perplexity with respect to the learning rate

- Train RNNs with 100 NCE GRU units on 1 % of the training dataset
- Varying learning rate to investigate its effect on performance

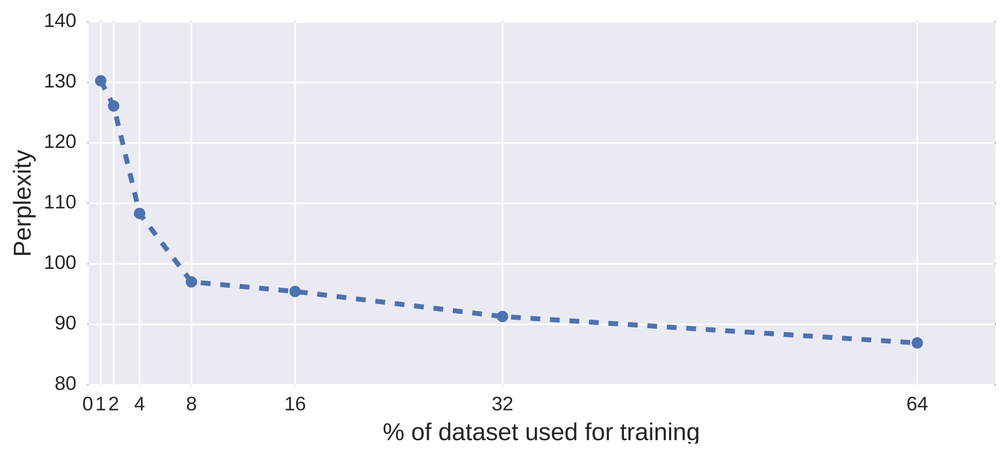
Evaluating the Model: Perplexity

- The best language model is one that best predicts an unseen test set
  - Given the highest P(sentence)
- Perplexity is the inverse probability of the test set, normalized by the number of words

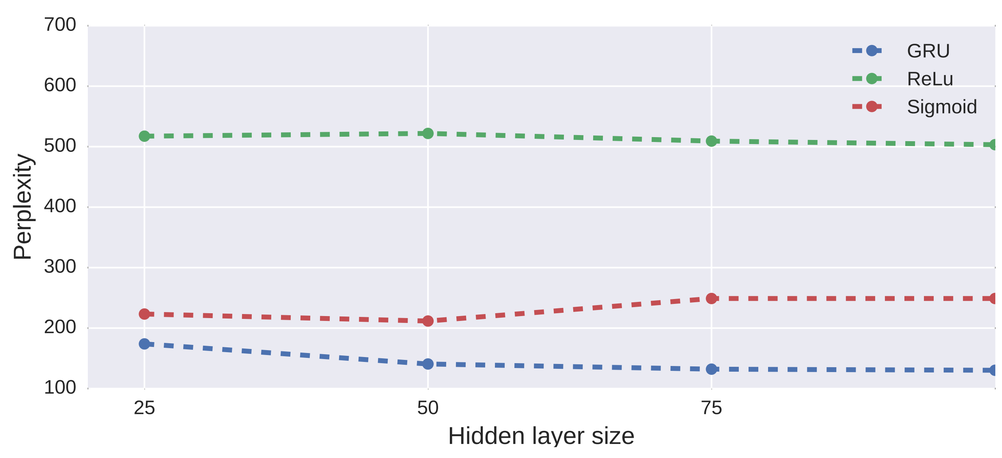
$$PP(W) = \sqrt[n]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \quad (1)$$

- Minimizing perplexity is the same as maximizing probability
- Lower perplexity = better model

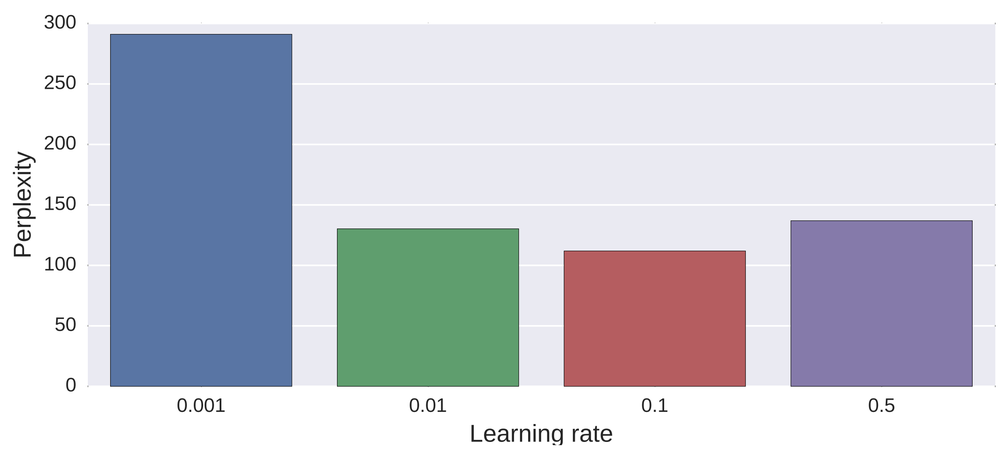
Results: Perplexity with respect to training data size



Results: Perplexity obtained by different network architectures



Results: Perplexity with respect to the model's learning rate



Conclusion

- An increasing training data size has a positive effect on the model's performance which however saturates when having supplied enough training examples (8 % of the training dataset)
- Distinct activation functions perform differently well, more sophisticated activation functions (gru) achieve better results than simple ones (relu)
- Hidden layer size does not have a significant impact on performance
- When choosing an appropriate learning rate in the range of 0.01 to 0.5, differences in performance are negligible

References

- P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, pp. 79–86, 2005.
- X. Chen, X. Liu, M. J. Gales, and P. C. Woodland, "Recurrent neural network language model training with noise contrastive estimation for speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5411–5415, IEEE, 2015.

https://nats-www.informatik.uni-hamburg.de/ {5hummel,5springe,4asgharp}@informatik.uni-hamburg.de