Machine Learning for Language Modelling

Homework

Marek Rei

UNIVERSITY of TARTU 1632

UNIVERSITY OF CAMBRIDGE

Homework

Two parts:

- Implement an n-gram language model
- Complete a neural network language model

Deadline:

5. May 2015, 23:59 Estonian time
(3 weeks from the end of the course)

Homepage:

http://www.marekrei.com/teaching/mllm

Homework dataset

A prepared dataset from Wikipedia

- Separated into training, dev, and test
- Tokenised, lowercased and sentence split
- Training set has 10M words
- For neural networks, I recommend using smaller sections of the data, as otherwise training will take long

LM task

- Use your language models to assign scores to sentences, based on their quality
- The task is to differentiate between the correct and incorrect versions of the same sentence (higher score for better sentence)
- Equal scores for both sentences = counted as incorrect
- Download input file and upload your score file to evaluate your system: http://www.marekrei.com/teaching/lmtask

LM task

Dataset contains 10,000 sentences - 5,000 sentence pairs

- Sentences from Wikipedia, where two randomly selected words are switched
- Sentences from language learners with mistakes, where the correct versions are manually created by experts

Everything is tokenised, lowercased, and sentence split

Homework: N-gram LM

- Use any programming language that you wish

- Implement an n-gram language model with bigrams or higher order n-grams

- Use some smoothing to handle zero-probability words. “Stupid” backoff is the easiest option that works well enough.

Homework: N-gram LM

My implementation:

- Bigrams
- unk 100
- “Stupid” backoff
- <s> and </s>
- Trains and tests on 500MB of memory in 30 seconds
- LM test accuracy: 0.733
- Likely not the most optimal settings, you are encouraged to experiment further

Homework: Neural LM

- There is a Java skeleton code provided for a neural LM

- Using JBlas for matrix operations

- Fill in the gaps for feedforward and backpropagation processing (apply matrix operations, based on what you’ve learned in the lectures)

- OR implement from scratch in any language

Homework: Neural LM

My implementation:

- 3-gram model (2 words of context)
- unk 100
- Word representation size: 30
- Hidden layer size: 30
- Activation: sigmoid
- Trained on 40K lines in 57 minutes
- Used 1K lines for development evaluation
- LM test accuracy: 0.6924
- Likely not the most optimal settings, you are encouraged to experiment further

Why neural nets don't win here?

- There are many strategies for speeding up neural nets that we don't use here
  - Class-based architecture
  - Hierarchical softmax
  - Noise contrastive estimation
  - Parallel processing with GPUS
- We have to train small models on less data, to finish in reasonable time
- You are welcome to experiment with more advanced options

Homework submission

For your submission of both language models:

- Include the source code
- Include the output file for the LM task
- Include your achieved accuracy on the LM task. I expect at least 0.68 with either model
- Include instructions on how to compile and run your system, to reproduce the result
- Do not use an external library for language modelling or neural network optimisation. Matrix algebra libraries are fine.
- Package everything up, upload it (eg Dropbox) and e-mail to me: marek.rei@gmail.com
- I recommend not leaving it on the last minute