Lehrstuhl Informatik 6
RWTH Aachen University
Prof. Dr.-Ing. H. Ney

Aachen, 29. 06. 2018

R. Schlüter,
K. Irie,
A. Zeyer

Automatic Speech Recognition

6. Exercise

This exercise reviews some basic concepts in neural networks.
Submission Deadline: 12. 07. 2018 at the beginning of the exercise session. (Thursday!).
IMPORTANT: Any electronic submission should be sent to:
`zeyer@i6.informatik.rwth-aachen.de`
with a copy (cc.) to `irie@i6.informatik.rwth-aachen.de` .

Task 6.1 Introduction to classical activation functions in neural networks

(a) Consider an L-layer multi-layer perceptron (MLP) with $\tanh$ as activation function for each layer except for the $\mathrm{softmax}$ output layer. Show that such a network can be re-parametrized into a MLP with the same architecture but with the standard sigmoid (σ) as the activation function instead of $\tanh$. Specify the transformation of the weight matrices and biases needed to transform the $\tanh$-based MLP into an equivalent sigmoid σ-based MLP. Remember the following identity: $\tanh(x) = 2\sigma(2x) - 1$ (5 P)

(b) When the task is a binary classification (output size of two), the softmax is not the most appropriate output activation function. Explain why and suggest an alternative activation function. (5 P)

Task 6.2 (*Bonus task) Illustration of the vanishing gradient

Consider the standard recurrent neural network. The hidden state vector $\boldsymbol{h_t}$ at time t is computed from the input $\boldsymbol{x_t}$ at time t and the hidden state at previous time step $\boldsymbol{h_{t-1}}$ as follows:

$$\boldsymbol{h_t} = \tanh(\boldsymbol{W}\boldsymbol{x_t} + \boldsymbol{R}\boldsymbol{h_{t-1}} + \boldsymbol{b})$$

where $\boldsymbol{W}$ and $\boldsymbol{R}$ are weight matrices and $\boldsymbol{b}$ is a bias vector.

(a) Assume that the matrix $\boldsymbol{R}$ is diagonal and all entries on the diagonal are equal to a real number α (i.e. $\boldsymbol{R} = \alpha\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix).
Write down the derivative of $\boldsymbol{h_t}$ w.r.t. $\boldsymbol{h_{t-1}}$: $\frac{\partial \boldsymbol{h_t}}{\partial \boldsymbol{h_{t-1}}}$.
What happens during the backpropagation through time if $0 < \alpha < 1$? What if $1 < \alpha$? (*5 P)

Task 6.3 Practical task: hyperparameter tuning for neural network training

Please first get the materials for this exercise by doing as follows:

- Download the software `rwthlm` from:
  https://www-i6.informatik.rwth-aachen.de/web/Software/rwthlm.php
  If you are a student worker at our institute, you can also use the binary:
  `/work/asr2/irie/adv-asr-exercise/rwthlm`
  The description of all flags for this software can be looked up on the same website. In particular, please note that the network topology should be indicated in the model file name.
- Download the script and data files from the lecture website. These will allow you to train neural network-based language models. The data size is intentionally chosen to be small such that the training can be observed in real time.

In the following questions, you will be asked to modify the hyperparameters for training neural language models and observe the effect. In this exercise, there will be no need to change the options `--sequence-length` and `--word-wrapping`. Please remember to give a new file name to the model for each modification. When you report a perplexity, please round it to one decimal place.

(a) Open the vocabulary file `vocab.rwthlm`. Note that the right column indicates the word class index. Report the vocabulary size and number of classes. (1 P)

(b) Open the script `train.sh`. Note that the model file name ends in `i100-m100`: this means that your model consists of a projection layer of size 100 and a LSTM layer of size 100 (followed by the output layer). Run the script `train.sh`. Describe how the learning rate is modified during the training. Report the validation perplexity at convergence. (2 P)

(c) In the script used in (b), change the number of nodes to 200 for the LSTM layer and run the scripts. What do you observe? (1 P)

(d) In the script used in (b), change the batch size to 4. What do you observe? (1 P)

(e) In the script used in (b), change the initial learning rate to 3e-3 and run the script. What do you observe? (1 P)

(f) In the script used in (b), add one more layer of LSTM. Run the script and report the new perplexity. (1 P)

(g) There is no pre-determined recipe for training a neural network: it's up to you to find your own recipe to achieve the best perplexity. Modify the hyperparameters to improve the best perplexity you got in the previous questions. Report any effect you observe by modifying these parameters. Report your best validation perplexity and your configuration. (3 P)