Machine Learning Exercise Sheet 07

Deep Learning

In-class Exercises

See the recording of the in-class exercise for the discussion of the code in the notebooks.

Problem 1: See notebook `exercise_inclass_07_vectorization_numerics.ipynb` on Moodle.

Problem 2: See notebook `exercise_inclass_07_backpropagation.ipynb` on Moodle.

Homework

Problem 3: In machine learning you often come across problems which contain the following quantity

$$y = \log \sum_{i=1}^{N} e^{x_i}$$

For example if we want to calculate the log-likelihood of neural network with a softmax output we get this quantity due to the normalization constant. If you try to calculate it naively, you will quickly encounter underflows or overflows, depending on the scale of x_i. Despite working in log-space, the limited precision of computers is not enough and the result will be ∞ or $-\infty$.

To combat this issue we typically use the following identity:

$$y = \log \sum_{i=1}^{N} e^{x_i} = a + \log \sum_{i=1}^{N} e^{x_i - a}$$

for an arbitrary a. This means, you can shift the center of the exponential sum. A typical value is setting a to the maximum ($a = \max_i x_i$), which forces the greatest value to be zero and even if the other values would underflow, you get a reasonable result.

Your task is to show that the identity holds.

Upload a single PDF file with your homework solution to Moodle by 08.12.2021, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

This is called the *log-sum-exp trick* and is often used in practice.

$$
\begin{aligned}
y &= \log \sum_{i=1}^{N} e^{x_i} \\
e^y &= \sum_{i=1}^{N} e^{x_i} \\
e^{-a} e^y &= e^{-a} \sum_{i=1}^{N} e^{x_i} \\
e^{y-a} &= \sum_{i=1}^{N} e^{-a} e^{x_i} \\
y - a &= \log \sum_{i=1}^{N} e^{x_i - a} \\
y &= a + \log \sum_{i=1}^{N} e^{x_i - a}
\end{aligned}
$$

Problem 4: Similar to the previous exercise we can compute the output of the softmax function $\pi_i = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}}$ in a numerically stable way by shifting by an arbitrary constant a:

$$
\frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} = \frac{e^{x_i - a}}{\sum_{i=1}^{N} e^{x_i - a}}
$$

often chosen $a = \max_i x_i$. Show that the above identity holds.

For some arbitrary constant $C > 0$ we have

$$
\frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} = \frac{C e^{x_i}}{C \sum_{i=1}^{N} e^{x_i}} = \frac{e^{x_i + \log(C)}}{\sum_{i=1}^{N} e^{x_i + \log(C)}}
$$

Since C is arbitrary, we can set $\log(C) = -a$ and get $\frac{e^{x_i - a}}{\sum_{i=1}^{N} e^{x_i - a}}$.

Problem 5: Load the notebook `exercise_07_notebook.ipynb` from Moodle. Fill in the missing code and run the notebook. Export (download) the evaluated notebook as PDF and add it to your submission.

We have implemented several helper functions for checking the correctness of your code in a small library `nn_utils.py` that can be downloaded from Moodle as well.

This week's programming assignment is closely connected to the contents of the in-class exercises. Make sure that you have a look at the in-class exercises before starting working on the homework task.

Upload a single PDF file with your homework solution to Moodle by 08.12.2021, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks and how to convert them to other formats, consult the Jupyter documentation and nbconvert documentation.

The solution notebook is uploaded to Moodle.