EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATH. - NATURWISS. FAKULTÄT
FACHBEREICH INFORMATIK
KOGNITIVE SYSTEME · PROF. A. ZELL

Deep Neural Networks

Assignment 8

Assignment due by: 03.07.2018, Discussions on: 10.07.2018

Question 1 Backpropagation (4 + 4 + 4 points)

In this assignment we will go back to native Python to manually implement Backpropagation. We work with the MNIST dataset again and we use a neural network with a single hidden layer. The structure will be the following:

$$\mathcal{L} = H(\boldsymbol{y}', \boldsymbol{y})$$
$$\boldsymbol{y} = \mathrm{Softmax}(\boldsymbol{u}^{(2)})$$
$$\boldsymbol{u}^{(2)} = \boldsymbol{z}^{(1)}\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}$$
$$\boldsymbol{z}^{(1)} = \sigma(\boldsymbol{u}^{(1)})$$
$$\boldsymbol{u}^{(1)} = \boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}$$

Where $\sigma(x)$ is the logistic sigmoid function. The derivative of the Softmax function has already been discussed in assignment 5, so in this exercise we will focus on the rest of the backpropagation process. Thus you can use $\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(2)}}$ without expanding it further.

To perform gradient descent we need the gradients, i.e. the partial derivatives w.r.t. the weights and biases, i.e. $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(i)}}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(i)}}$ for $i \in \{1, 2\}$.

(a) Just using the chain rule, expand $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(i)}}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(i)}}$ into a chain of partial derivatives (without calculating the derivatives, just write down the correct derivative expressions).

(b) Calculate the derivatives and write out the resulting formulas (Except $\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(2)}}$). You can write the derivative of $\sigma(x)$ as $\sigma'(x)$.

(c) Find common subexpressions that could be used to avoid duplicate calculations. Write down the order of calculations to compute all the gradients efficiently.

Notes: Remember that $\frac{\partial f}{\partial x} = \sum_i \frac{\partial f}{\partial z_i}\frac{\partial z_i}{\partial x}$ when the intermediate variable is a vector. For this question you don't need to take into account that you are processing multiple inputs at the same time.

Question 2 Implementing Backpropagation (8 points)

The file `backprop.py` contains most of the code for training a 1-hidden-layer neural network on the MNIST dataset. Your task is to add the appropriate operations to calculate the partial derivatives and the gradients that are then used in gradient descent.

For this task you need to take into account that the batches consist of multiple images, so some of the arrays you are dealing with have an extra dimension that you have to average over at the end.