Technische Universität Kaiserslautern
Fachbereich Informatik
RHRK
Dr. habil. Josef Schüle

High Performance Computing with GPUs
Exercise Sheet No. 3

Exercise Neuronal Networks
In the directory `/scratch/GPU/src` you find a CPU program for neuronal networks. You may copy the files, make a binary and call the program with
`./neuron -p /scratch/GPU/test.dat -t /scratch/GPU/train.dat -n /scratch/GPU/network.inp`
(there is no need to copy the dat-files).
All functionality to be changed is mainly located in the function `network.c`. Other files should not be touched to start with. Data layout may be found in `net_include.h`. I suggest to keep this layout and organize the data on the GPU similarly.
Your task is to port this program to be used with a GPU. To start with, only the expensive back propagation has to be ported. It is the function `grad` located in `network.c`. Multiplying matrices with vectors and including a function for the rank-1 update (named sger) should be an easy task with help of the cublas-library. The most important arrays are allocated in function `net_init`. Please note, that there is one allocation for all layers, that is for example,

```
size=sizeof(float)*2*network.l_prod;
cuda_Memcpy(d_net[1].w_in,net[1].w_in,size,cudaMemcpyHostToDevice);
```

will copy all weights and all derivates of the weights from host to device for all layers.
In `network.inp` the network layout is defined. It contains 784 neurons as input layer. This number reflects the size of each image which is 28×28. Each image contains a hand written digit. Therefore the number of output neurons is 10, according to the 10 digits. The output neuron with largest value is the one selected.