Announcements:

- There will be no lectures and no tutorials next week (week 5). The last tutorials will be held in week 6, or as specified by your tutor.
- This is the last exercise. Different to the previous exercises, you have two weeks time so that the due date is the 4th of February.

Exercise 22 - Neural Networks

Your task is to solve a multi-class problem (MNIST, $K = 10$ classes) using a one hidden layer neural network.

$$f_k(w,u)(x) = \sum_{i=1}^{m_1} w_{ki}\,\sigma\Big(\sum_{j=1}^{d} u_{ij}x_j\Big), \qquad k = 1,\ldots,K.$$

where $\sigma(z) = \max\{0, z\}$, that is you use the *ReLU* (rectified linear unit) as activation function and m_1 is the number of hidden units. As loss you use the cross-entropy loss, that is you minimize

$$\min_{w\in\mathbb{R}^{10\times 20}, u\in\mathbb{R}^{20\times 784}} \frac{1}{n}\sum_{i=1}^{n} L\Big(y_i, f(w,u)(x_i)\Big).$$

That means we use no regularization.

a. **(5 Points)** Write a function

```
[wnew,unew,Obj,trainError,testError]
=SGDReLUNetwork(X,Y,Xtest,Ytest,wcur,ucur,alpha,batchsize,epochs)
```

which minimizes the loss based on stochastic gradient descent with variable batchsize for a fixed number of epochs (one epoch = one pass over the data without replacement) and `wcur` and `ucur` are the starting points. Your update step of w and u has the form

$$w \mapsto w - \alpha\frac{1}{B}\sum_{i=1}^{B}\nabla_w L(y_i, f(w,u)(x_i)), \quad u \mapsto u - \alpha\frac{1}{B}\sum_{i=1}^{B}\nabla_u L(y_i, f(w,u)(x_i)),$$

where here $(x_i, y_i)_{i=1}^B$ is the mini-batch you use for each step of stochastic gradient descent.

For each iteration of SGD you compute also training/test error and the objective and return this together with the latest weights.

b. **(3 Points)** Apply the function to the MNIST dataset with $m_1 = 20$ and batchsize 20 for 10 epochs. Use stepsize parameter $\alpha = 1$. Plot training error, test error in one plot as a function of the iterations (not epochs) and in a separate plot the objective value.

The MNIST dataset can be downloaded from *http://yann.lecun.com/exdb/mnist/* and be loaded in Matlab via scripts from *http://ufldl.stanford.edu/wiki/index.php/Using_the_MNIST_Dataset*

Exercise 23 - Semi-supervised Learning

Given a graph G with non-negative weights, the unnormalized Laplacian is defined as

$$L = D - W,$$

where D is a diagonal matrix with degree of vertices on the diagonal and W is the weight matrix.

a. **(4 Points)** Consider the following objective for semi-supervised learning:

$$F(f) := \sum_{i \in T} d_i (y_i - f_i)^2 + \frac{\lambda}{2} \sum_{i,j \in T} w_{ij} (f_i - f_j)^2,$$

where T is the set of all points (x_i, y_i) and d_i is the degree of the i^{th} vertex. Note that $y_i = 0$ for unlabeled points. What is the interpretation of this objective?

Show that the minimizer f^* of F is given by

$$f^* = \Big(\mathbb{1} + \lambda(\mathbb{1} - D^{-1}W) \Big)^{-1} Y$$

b. **(3 Bonus Points)** The minimizer f^* of $F(f)$ can be rewritten as

$$f^* = (1 - \alpha) \sum_{r=0}^{\infty} \alpha^r S^r Y,$$

where $\alpha = \frac{\lambda}{1+\lambda}$ and $S = D^{-1}W$ (no need to derive this part).

Derive the solution f^* in the limit of $\lambda \to 0$ (resp. $\alpha \to 0$) where the graph is sparse (in particular it is not fully connected); that is derive for each vertex i its label (note that the final classification of vertex i is done with $\text{sign}(f^*(i))$.

Hints:

- For the last part you have to consider the limit $\lambda \to 0$. Setting $\lambda = 0$ will yield a different result ! It is helpful to expand the sum in the solution $f^*(i)$ for one vertex i. What happens as $\alpha \to 0$? Which terms in the sum are non-zero ? Which terms dominate the solution as $\alpha \to 0$? What is the meaning of $(S^r)_{ij}$ (think of paths in the graph) ?

Submission instructions

- We accept both handwritten and electronic submissions. So you can choose what is more convenient for you. In any case, you should specify full names and immatriculation IDs of all team members. Obviously, programming tasks you can submit only electronically.
- Handwritten submissions should be submitted in the lecture hall of Monday's lecture (before the lecture starts).
- Electronic submissions should be zipped, containing the m-files (`Basis` etc.), your plots (png files) and the matlab data files (.mat) and emailed to the corresponding tutor:

  a. Apratim Bhattacharyya (Wednesday 8-10): abhattac@mpi-inf.mpg.de
  b. Maksym Andriushchenko (Thursday 8-10): s8mmandr@stud.uni-saarland.de
  c. Max Losch (Friday 16-18): mlosch@mpi-inf.mpg.de

  If not all 3 students belong to the same tutorial group, then you should email your submission to **only** one tutor (e.g. to the tutor of the first author of your homework), so please do not put other tutors in copy of the email.

  The email subject must have the following form: "[ML18/19 Exercise] Sheet X", where X is the number of the current exercise sheet. Then please specify in the email full names and immatriculation IDs of all team members. Then please attach all your files as a single zip archive, which consists of your immatriculation IDs, e.g. "2561234_2561235_2561236.zip".