Assignment 3: Optimization and Regularization
MA-INF 2313: Deep Learning for Visual Recognition

Theoretical Task Due Date: 24.11.2017
Practical Task Due Date: 01.12.2017

Assistants: Soumajit Majumder majumder@cs.uni-bonn.de
Fadime Sener sener@cs.uni-bonn.de

1 Theoretical Exercises (15 pts)

1 Fundamentals of Unconstrained Optimization : 8 points
(a) Compute the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ of the Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Show that $x^* = (1, 1)^T$ is the only local minimizer of this function, and that the Hessian matrix at that point is positive definite.

(b) Show that the function $f(x) = 8x_1 + 12x_2 + x_1^2 - 2x_2^2$ has only one stationary point, and that it is neither a maximum or minimum, but a saddle point.

2 Case Study : 7 points

For this question, we consider a MLP with one hidden layer: two input units, three hidden units, one output unit. The non-linearity that is used is sigmoid, the error function is given by the square of the euclidian distance. A gradient descent is performed with 'online' learning, meaning that the error is backpropagated after each sample iteration. Below is the learning curve of the error on the training set.

(a) For each training sample, the error function can be seen as a function of the parameters of the network. If the network has n parameters, the graph of this function gives a hypersurface in $\mathbf{R}^{n+1}$, also called the error surface. How would you describe what this error surface looks like around the parameters obtained between the times corresponding to iteration 100^{th} and iteration $10,000^{th}$? How is such an area of the error surface usually called ?

(b) If you could choose a property that the Hessian matrix of the error function would satisfy at, say, iteration 100, what would you choose?

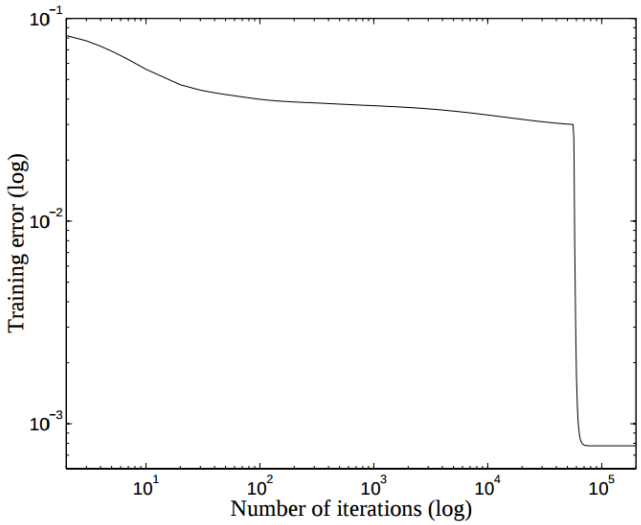

Figure 1: K.Fukumizu, S.Amari. "Local minima and plateaus in hierarchical structures of multilayer perceptrons."

2 Programming Exercises (15 pts)

For this programming exercise, you will study the **impact of regularization and optimization on the training and classification performance of a Multi-Layered Perceptron (MLP)** on the MNIST Dataset.

[Task 1 : 8 pts]

For this task, you have to first design a MLP with **2 hidden layers** with **800 hidden units in each layer**. We will use the **linear rectifier as the activation unit**. The second hidden layer is followed by a **softmax output layer of 10 units** - same as the number of classes in MNIST. Additionally, we introduce **dropout layers** - we apply a 20% dropout to the input data and a 50% dropout following each hidden layer. Furthermore, we will use **cross-entropy as the error or loss function** [1]. Cross-entropy is often the most appropriate when working with logistic or softmax output layers. In each epoch, we do a full pass over the training data and update our parameters based on the mini-batches.

Using this MLP as a starting point, create the following variants of this MLP by modifying the training-loss function and parameter update routines as shown in the table.

[1] Refer to tutorial python notebooks

Type	REG	OPT
MLP I	-	SGD
MLP II	-	NESTEROV
MLP III	L1	NESTEROV
MLP IV	L2	NESTEROV

Here we have two different type of parameter update routine : 'SGD' which refers to stochastic gradient descent and 'Nesterov' which refers to SGD with Nesterov momentum. To safeguard against over-fitting, we also use the l_1 and l_2 regularization.

[Performance Analysis : 7 pts]

Keep the number of iterations fixed at $10,000$.

1. Plot the **training loss vs iteration** for each of the variants (MLP I to IV) on the same plot. Also the **validation accuracy vs epoch** for all the variants. *Which variant converges the fastest in training ? Does the same variant also simultaneously arrive at the lowest training error amongst the four ?* Explain your results.
2. Report the classification accuracy for each of the variants on the MNIST dataset. *Does having regularizations and smarter optimization routines signficantly change the performance on this dataset ?*

Also, feel free to play with the hyper-parameters and report the network with the best performance on the test set.
