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

Theoretical Task Due Date: 28.11.2019
Practical Task Due Date: 28.11.2019

Assistants: Soumajit Majumder

1 Theoretical Task (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 being 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?

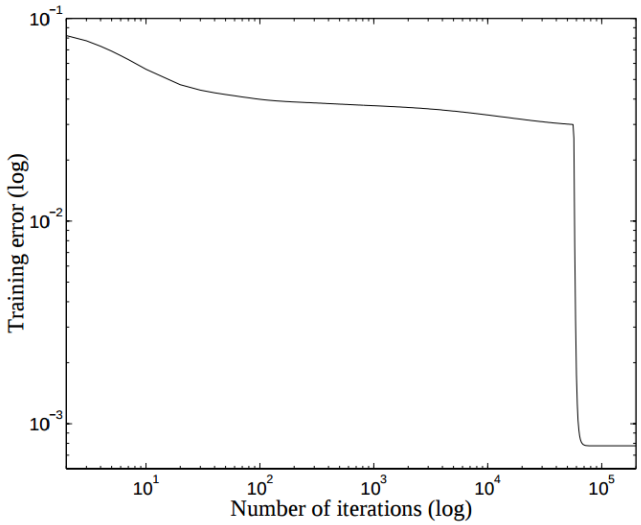

Figure 1: K.Fukumizu, S.Amari. "Local minima and plateaus in hierarchical structures of multilayer perceptrons."

2 Programming Exercises (15 pts)

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

For this task, you have to design a MLP with **2 hidden layers** with **512 hidden units** in each layer with **ReLU** as the activation function. The second hidden layer is followed by a **softmax layer**. Use cross-entropy loss as the loss function; it is often the most appropriate when working with logistic or softmax output layers. Using this MLP as a starting point, create the following variants by modifying the training-loss function and parameter update routines as shown in the table.

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. Keep the number of epochs fixed at 25.

1. Plot the validation loss vs epochs for each of the variants (MLP I to IV) on the same plot. Which variant converges the fastest in training ?

2. Report the classification accuracy for each of the variants on the MNIST dataset.

Also, feel free to play with the hyper-parameters and report the network with the highest classification accuracy on the test set. Finally, change the number of hidden units in each of the layers from 512 to 2048 and 512 respectively. Then re-run the same set of experiments for ORL_FACES subset (previously provided).
