Assignment 4: Neural Networks

Due to: May 7, 2020

Goal: Get familiar with neural networks by implementing them and applying them to image classification.

Submission: The assignment consists of two parts: implementation and analysis. You are supposed to present results for both parts in the following manner:

1. Upload your code.
2. Prepare a report with an analysis of results obtained at home.

The code and the report must be uploaded due to the deadline to Canvas.
UPLOAD A **SINGLE FILE** (a zip file) containing your code and the report. Name your file as follows: [vunetid]_[assignment number].

Introduction

In this assignment we are going to learn about neural networks (NNs). The goal is to implement two neural networks: a fully-connected neural network, a convolutional neural network, and analyze their behavior.

The considered task is image classification. We consider a dataset of small natural images (see the additional file) with multiple classes. We aim at formulating a model (a neural network) and learning it using the negative log-likelihood function (i.e., the cross-entropy loss) as the objective function, and the stochastic gradient descent as the optimizer.

In this task, **the code must be implemented in PyTorch.**

Part 1: Implementation

Data Loaders

First, take a look at the provided additional file and use it for loading data (data loaders).

Fully-Connected Network (FCN)

1. Implement a FCN with the following structure:

   $$x \rightarrow h \in IR^{300} \rightarrow y,$$

   that is, it contains 300 neurons, and two linear layers. Use ReLU as the first non-linarity, and the softmax for the output.
2. Train this model and monitor:
   a. Training loss (using the training set).
   b. Validation loss (using the validation set).

   Plot the learning progress.
3. Once the model is trained, evaluate the FCN using the test set using the objective function and the classification error.

Please remember that the learning rate is an important hyperparameter and it should be relatively small (e.g., 0.001).

Convolutional Neural Network (CNN)

1. Implement a CNN with the following structure:

   input → Conv2d + ReLU→ Pooling → Linear + ReLU → Linear + Softmax → output.

   Please take 32 filters in Conv2d, stride equal 1, kernel size 3x2, and padding equal 1.
   Pooling is 2x2.
   First linear layer outputs 300 neurons.
2. Train this model and monitor:
   a. Training loss (using the training set).
   b. Validation loss (using the validation set).

   Plot the learning progress.
3. Once the model is trained, evaluate the CNN using the test set using the objective function and the classification error.

Part 2: Analysis

1. Compare the performance of FCN and CNN.
2. How does the learning rate (the step size) affect learning?
3. How can you explain the differences in performance of these two models?