Advanced Topics in Machine Learning
Assignment # 2
Universität Bern

Due date: 14/04/2020

1 Introduction

In this assignment you need to upload a zip file to ILIAS which includes: 1) A Jupyter Notebook file Assignment2.ipynb completed with code and answers and 2) a Jupyter Notebook exported to HTML (File / Export Notebook as / HTML). The zip file name must be FirstName LastName.zip. If your implementation requires auxiliary functions, you must implement those functions inside a corresponding .py file. Please state your name at the beginning of the notebook.

1.1 Notes on code and submission quality

In addition to answering the different questions, you are also expected to provide well written submissions. Here are some recommendations to take into consideration.

- Please answer the question in the same order as in the assignment and use the same question numbers.
- Don't answer the questions in the code comments. Use the text cells in your notebook.
- Remove clutter such as unused code lines instead of turning them into comments.
- Make sure the right execution order of the notebook cells is from top to bottom. A TA should be able to reproduce your results by simply clicking "Run All" without having to guess which cells should be executed first.

Poorly written submissions might result in points deduction.

2 Problem

ImageNet is a well known dataset with 1000 image classes. We will be working on a subset of the dataset (60k images, 100 classes, 600 images per class 80×80 pixels, RGB) and train a model to classify an image into one of the 100 classes. The

dataset is located under the "data" directory. Training and validation data splits are under "data/train" and "data/val" directories respectively. Both splits consist of 100 directories, each representing an object category.

Tasks:

1. **[10 Points]** Implement a **ImageNetLimited** class for data loading in a **datasets.py** file.

2. **[20 Points]** Design and implement a Convolutional Neural Network architecture for image classification in a **ConvNet** class in the notebook. Some examples of popular classification models are: AlexNet, VGG, ResNet, ...Justify your design choices in the report. The input to your model must be an image of size 64×64 pixels.

3. **[30 Points]** Implement training and evaluation code for your model. Choose an appropriate loss function and evaluate the model on the validation set using classification accuracy. You are not allowed to use a pre-trained model (must train from scratch on the provided data).
   Your model should achieve an accuracy of at least 40.0% on the validation set.

4. **[30 Points]** Try to find the best performing model by tuning the model design and hyper-parameters on the validation set. Perform ablation experiments to illustrate the effect the most important hyper-parameters. Some examples of ablations: training parameters (e.g., optimizer, learning rates, batch size), network architecture (e.g., number of layers, number of units, activation function, normalization layers), model regularization (e.g., data augmentation, dropout, weight decay, early stopping), test-time augmentation, etc...**Perform at least 5 ablations and report the performance of each on the validation set.**

5. **[10 Points]** Evaluate the trained model on the validation set and plot 10 random mistakes that your model made.

6. **[10 Bonus Points]** Competition time! Read the images from "data/test" folder. There are no labels for these images. Run your best model on these images and save the image IDs (names) and predicted label in a file LastName.csv. You will receive a link via email to upload the CSV file to an online system which will give you the score of your model on the held-out test set. Top 5 students with at least 40% classification accuracy will obtain bonus points.