Advanced Topics in Machine Learning
Assignment # 3
Universität Bern

Due date: 19/05/2020

1 Introduction

In this assignment you need to upload a zip file to ILIAS which includes: 1) A Jupyter Notebook file Assignment3.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 Unsupervised Learning and Transfer Learning

In many real world applications the amount of labelled training data will be very limited since it is expensive to obtain. Often you will have access to a large amount of

unlabelled data however. In this assignment you will explore techniques to leverage large amounts of unlabelled data to learn image representations. These image representations can then be used for novel tasks (e.g., classification) via supervised transfer learning. If the number of labelled examples for the target task is small, unsupervised feature learning can substatially improve the performance on the target task.

2.1 Data

In this assignment you are given two types of training sets:

- **A large unlabelled training set:** This data will be used to learn a data representation with some unsupervised feature learning method of your choice (100K unlabelled images in total).
- **A small labelled training set:** This data will be used to perform transfer learning where the previously learned representation is repurposed for image classification (4K labelled images in total).

Additionally you will find a labelled validation set (2K images) and an unlabelled test set for Kaggle (10K images). Images are of size 80×80 and belong to 200 object classes. The setup is similar to Assignment 2.

2.2 Choice of Unsupervised Learning Task

You are free to choose any method to learn features in an unsupervised manner. Possible examples from the lectures include:

- **Autoencoders:** You can learn image representations in the encoder of some variant of the autoencoder model (e.g., DAE, VAE, etc. . .).
- **GAN:** You can train a generative adversarial network and transfer the features of the discriminator.
- **Self-Supervised Learning:** You can use a pretext task where labels come "for free" with the data. Some simple examples are "Rotation Prediction" and "Exemplar-CNN".

2.3 Tasks:

1. **[10 Points] Prepare the data**: You will need to create the following datasets:
   - Unlabelled training set (optionally including self-supervised labels)
   - Labelled training set
   - Labelled validation set
   - Unlabelled test set (for Kaggle)

2. **[20 Points] Define the CNN architecture:** The architecture should be designed such that the first k layers (the backbone) of the network are identical during unsupervised pre-training and supervised transfer learning. The remaining layers (the network head) can be different for unsupervised and supervised training. You are free to reuse (part of) the network from Assignment 2.

   Example: If you choose to train an autoencoder, the encoder layers would be shared between unsupervised pre-training and supervised transfer. The decoder layers would be removed and replaced by some number of fully-connected layers during transfer to classification.

3. **[20 Points] Implement the unsupervised training code for your model.** You are not allowed to use a pre-trained model or labelled data (must train from scratch on the unlabelled training set). Monitor the performance on the pre-training task to check for convergence. Save the network parameters after pre-training (you will need them during transfer experiments).

4. **[40 Points] Transfer Learning Experiments** Perform the following set of transfer learning experiments:

   - **Fixed Features:** Initialize the first k layers with the parameters learned in Task 3 and randomly initialize the layers of the classification head. **Train only the layers of the classification head!**
   - **Finetuned Features:** Initialize the first k layers with the parameters learned in Task 3 and randomly initialize the layers of the classification head. **Train all the layers!**
   - **Random Features Fixed:** Randomly initialize all the layers for the classification task (do not use parameters learned in Task 3). **Train only the layers of the classification head!**
   - **Random Features Finetuned:** Randomly initialize all the layers for the classification task (do not use parameters learned in Task 3). **Train all the layers!**

   Tune and evaluate each of those models on the validation set and summarize the results in a Table.

5. **[10 Points + 10 Bonus]** Competition time! Compute predicitons for examples in the test set. 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. **Achieve at least 15% accuracy to score 10 points. The top 5 students will obtain an additional 10 bonus points. Provide a short description of the model used for the Kaggle submission in the report**