Deep Learning # 1
Universität Bern

Hamadi Chihaoui (hamadi.chihaoui@unibe.ch)
Llukman Cerkezi (llukman.cerkezi@unibe.ch)

Due date: 22/04/2024

1 Introduction

In this assignment you need to upload a zip file to ILIAS which includes: 1) A Jupyter Notebook file Assignment1.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

Pneumonia is a medical condition characterized by inflammation and infection of the air sacs in one or both lungs. Detecting pneumonia can be done using various methods like CT scans, pulse oximetry, and others, with the most common method being X-ray

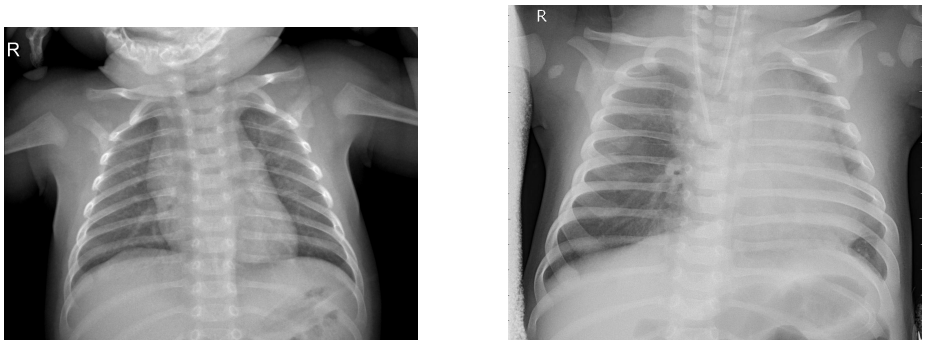

Figure 1: An illustrative example of Chest X-ray in patients with clear lungs without any areas of abnormal opacification (**Left**) and an example of Chest X-ray with bacterial pneumonia (**Right**).

imaging. However, interpreting chest X-rays (CXR) can be challenging and subject to differences in interpretation. In this task, our aim is to develop a model for pneumonia detection that determines whether a given chest X-ray has pneumonia or not. The dataset is accessible here.

Tasks:

1. **[Total 60 pts] Train a fully connected neural network and convolutional neural network for binary classification.**

   You will implement and train a fully connected neural network and convolutional neural network for binary classification to predict whether a given chest X-ray has pneumonia or not (see Fgiure 1). To access the data, first extract `data.zip`. Folders `train` and `val` contain the train and validation datasets respectively.

   - **[10 pts]** Create Dataset and DataLoader objects for provided training and validation data (folders train and val). Visualize few images from each class.
   - **[10 pts]** Implement the MLP model according to the definition below:
     – Fully connected layer, out_features=128
     – Activation function ReLU
     – Fully connected layer, out_features=128
     – Activation function ReLU
     – Fully connected layer, out_features=128
     – Activation function ReLU
     – Fully connected layer, out_features=128
     – Activation function ReLU
     – Fully connected layer, out_features=2

- **[15 pts]** Implement a convolutional model according to the definition below:
  – Convolutional layer, kernel size 3x3, stride 1, 32 channels
  – Max Pooling layer, kernel size 3x3, stride 2, ceil mode=True
  – Activation function ReLU
  – Convolutional layer, kernel size 3x3, stride 1, 64 channels
  – Max Pooling layer, kernel size 3x3, stride 2
  – Activation function ReLU
  – Convolutional layer, kernel size 3x3, stride 1, 64 channels
  – Max Pooling layer, kernel size 2x2, stride 2
  – Activation function ReLU
  – Convolutional layer, kernel size 2x2, stride 1, 128 channels
  – Activation function ReLU
  – Convolutional layer, kernel size 3x3, stride 1, 256 channels
  – Activation function ReLU
  – Convolutional layer, kernel size 3x3, stride 1, 256 channels
  – Activation function ReLU
  – Convolutional layer, kernel size 1x1, stride 1, 2 (output) channels
- **[20 pts]** Write the training code and train the network you implemented.
  – Train for 30 epochs with a batch size of 32.
  – Optimize the cross entropy loss.
  – Use Adam optimizer with learning rate 1e-3.
- **[5 pts]** Include plots for the training and validation losses and accuracies.

2. **[Total 30 pts] Add Regularization to your convolutional (CNN) model**

   Regularization is a common technique used in deep learning to prevent overfitting in models. In this task, you should choose two popular regularization techniques.

   - **[15 pts]** Train a convolutional neural network with the first regularization technique you have chosen.
   - **[15 pts]** Train a convolutional neural network with the second regularization technique you have chosen.

3. **[10 pts] Comment the results.**

   At the conclusion (last cell) of your Jupyter notebook, please provide the final validation accuracies for the following scenarios:

   - Validation accuracy for a fully connected neural network.
   - Validation accuracy for a convolutional neural network.
   - Validation accuracy for a convolutional neural network with the first regularization technique applied.

- Validation accuracy for a convolutional neural network with the second regularization technique applied.

Additionally, comment your results. Identify which model achieved the highest score and explain the reasons behind its success. Discuss potential enhancements that could be implemented to improve accuracy. Lastly, analyze why the performance of other models may not have matched that of the best-performing one.