Deep Learning Course - Final Project

Nitsan BenHanoch (208585927), Nir Koren (316443902), Tsvi Tabach (311292304)

Submitted as a final project report for the DL course, BIU, 2024

1 Introduction

Deep-Learning in healthcare is a hot topic. As deep neural networks become increasingly involved in patient care, these networks must meet several requirements. Such requirements include high precision rate, transparency in their decision-making, and performing effectively even when valuable data is missing.

1.1 The Dataset

Kaggle's chest-xray-pneumonia dataset consists of 5,856 lungs X-Ray images of men and children. Each image is labeled either "healthy", "bacterial-infected" or "viral-infected". The data is split between train/set sets. One important issue to note is the classes are not balanced; each set contains a different ratio of images from the different classes.

1.2 Our Tasks

In this project, we demonstrate several different deep-learning techniques we learned during the course.
Task 1.a. Binary Classification: we train a CNN on healthy/sick labeled x-ray images of human lungs;
Task 1.b. Multiclass classification: we further improve the CNN to distinguish between bacterial/viral infection;
Task 2. Embedding Space: we apply KNN and t-SNE on embedding vectors we get from the aforementioned networks;
Task 3. Anomaly Detection: having trained on healthy lungs alone, we use auto-encoders to tell if lungs are sick;
Task 4. Explainability: we decipher the model's decision making, e.g. what's important to each layer and neuron.

1.3 Related Works

Previous research, such as CheXNet, study by Rajpurkar et al. (2017), has shown the advantage of using CNN in medical diagnostics. The model proposed in the paper used CNN to output probability of pneumonia with a heat-map localizing the pneumonia areas from chest X-ray images.

2 Task 1.a: Binary Classification

2.1 Solution

2.1.1 The Task

In this task, we treat both bacterial and viral images as *sick*, and train a model to classify *healthy/sick* lungs x-ray.

2.1.2 General approach

We use several convolution layers, followed by a few dense layers. It is the field's classic and go-to way to classify images. Furthermore, the limited dataset size (approximately 5K images) posses a serious restriction on the complexity of the network, as overly powerful architectures risk significant over-fitting.

2.1.3 Design

Our network's architecture mimics the great: we took inspiration from the architecture of the VGG networks family. We tailored their layer sizes to our needs after thorough experimentation, maximizing accuracy, while battling over-fitting. The final network structure is portrayed in Figure 1. We built the model as a Keras Sequential on a kaggle notebook. It took ~30 minutes to train using NVIDIA Tesla P100 GPU. For Loss, we used the classic binary-crossentropy, as it's classic in the context of binary classification. For optimizer we used Adam with initial learning rate of 1e-4, but also installed a callback to further lower it on plateau. As said, over-fitting was a major challenge, due to the limited size of the dataset (~5K images);

224 x 200 x 1 (Grayscale)

222 x 198 x 32 · 220 x 196 x 32 · 110 x 98 x 32 · 108 x 96 x 64 · 106 x 94 x 64 · 53 x 47 x 64 · 51 x 45 x 128 · 49 x 43 x 128 · 24 x 21 x 128 · 22 x 19 x 256 · 20 x 17 x 256 · 18 x 15 x 256 · 9 x 7 x 256 · 512 · 256 (the **embedding** vector) · probability lungs are sick

Input Image | 3x3 Conv (32 dims) + Relu | 3x3 Conv (32 dims) + Relu | 2x2 MaxPool | 3x3 Conv (64 dims) + Relu | 3x3 Conv (64 dims) + Relu | 2x2 MaxPool | 3x3 Conv (128 dims) + Relu | 3x3 Conv (128 dims) + Relu | 2x2 MaxPool | 3x3 Conv (256 dims) + Relu | 3x3 Conv (256 dims) + Relu | 3x3 Conv (256 dims) + Relu | 2x2 MaxPool | Dense (512 neurons) + Relu | Dense (256 neurons) + Sigmoid | Dense (1 neuron) + Sigmoid
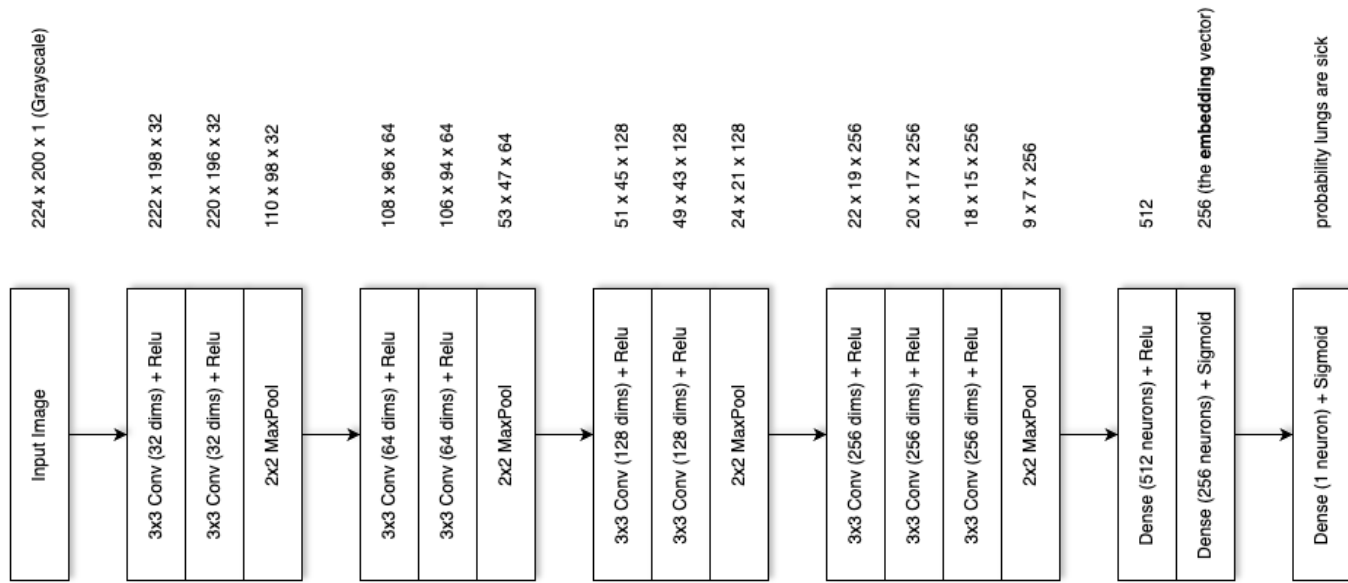
Figure 1: Binary Classification Model

We combated it using several techniques, including many forms of augmentation, limited number of epochs, and more. We spared 20% of the train set for validation. Also, we dealt with unbalanced classes by removing excess data until the number of representatives of each class is equal.

2.2 Experimental results

Provide information about your experimental settings. What alternatives did you measure? Make sure this part is clear to understand, provide as much details as possible. Provide results with tables and figures.

2.3 Discussion

Provide some final words and summarize what you have found from running the experiments you described above. Provide some high level insights.

Note - your project will be evaluated for aspects, including the technique you selected, the rational of the experiments you decided to run, the insights you learned from this process and more. Remember, for the purpose of this course, the process that you demonstrate is very important.

2.4 Code

Please provide a link to your colab notebook.

3 Task 1.2: Multiclass Classification

3.1 Solution

3.1.1 General approach

Describe your preferred approach to solve the problem. what alternatives you plan to try and why.

3.1.2 Design

Provide some general information about your code, platform, how long it took you to train it, technical challenges you had, Loss functions, Optimizers, Architecture, and more.

3.2 Experimental results

Provide information about your experimental settings. What alternatives did you measure? Make sure this part is clear to understand, provide as much details as possible. Provide results with tables and figures.

3.3 Discussion

Provide some final words and summarize what you have found from running the experiments you described above. Provide some high level insights.

Note - your project will be evaluated for aspects, including the technique you selected, the rational of the experiments you decided to run, the insights you learned from this process and more. Remember, for the purpose of this course, the process that you demonstrate is very important.

3.4 Code

Please provide a link to your colab notebook.

4 Task 2: Model Embeddings and Visualization

4.1 Solution

4.1.1 General approach

Describe your preferred approach to solve the problem. what alternatives you plan to try and why.

4.1.2 Design

Provide some general information about your code, platform, how long it took you to train it, technical challenges you had, Loss functions, Optimizers, Architecture, and more.

4.2 Experimental results

Provide information about your experimental settings. What alternatives did you measure? Make sure this part is clear to understand, provide as much details as possible. Provide results with tables and figures.

4.3 Discussion

Provide some final words and summarize what you have found from running the experiments you described above. Provide some high level insights.

Note - your project will be evaluated for aspects, including the technique you selected, the rational of the experiments you decided to run, the insights you learned from this process and more. Remember, for the purpose of this course, the process that you demonstrate is very important.

4.4 Code

Please provide a link to your colab notebook.

5 Task 3: Anomaly Detection

5.1 Solution

5.1.1 General approach

Describe your preferred approach to solve the problem. what alternatives you plan to try and why.

5.1.2 Design

Provide some general information about your code, platform, how long it took you to train it, technical challenges you had, Loss functions, Optimizers, Architecture, and more.

5.2 Experimental results

Provide information about your experimental settings. What alternatives did you measure? Make sure this part is clear to understand, provide as much details as possible. Provide results with tables and figures.

5.3 Discussion

Provide some final words and summarize what you have found from running the experiments you described above. Provide some high level insights.

Note - your project will be evaluated for aspects, including the technique you selected, the rational of the experiments you decided to run, the insights you learned from this process and more. Remember, for the purpose of this course, the process that you demonstrate is very important.

5.4 Code

Please provide a link to your colab notebook.

6 Task 4: Explainability in Deep Learning

6.1 Solution

6.1.1 General approach

Describe your preferred approach to solve the problem. what alternatives you plan to try and why.

6.1.2 Design

Provide some general information about your code, platform, how long it took you to train it, technical challenges you had, Loss functions, Optimizers, Architecture, and more.

6.2 Experimental results

Provide information about your experimental settings. What alternatives did you measure? Make sure this part is clear to understand, provide as much details as possible. Provide results with tables and figures.

6.3 Discussion

Provide some final words and summarize what you have found from running the experiments you described above. Provide some high level insights.

Note - your project will be evaluated for aspects, including the technique you selected, the rational of the experiments you decided to run, the insights you learned from this process and more. Remember, for the purpose of this course, the process that you demonstrate is very important.

6.4 Code

Please provide a link to your colab notebook.

Good luck!!