Report Practical Exercise 03

**Task 1** : Lowdata Regime and Autoencoders

1a:

2. *Implement the calculation of accuracy in the test function. Is accuracy appropriate here? Examine the data distribution to support your argument.*

In our data distribution, there is an imbalance between classes, with a greater number of samples that indicating the presence of disease than healthy ones. In the training set, we have 134 positive samples and 166 negative samples, while in the test set, there are 10,505 positive samples and 11,928 negative samples. By looking at the evaluation on the test set, we have a true positive rate of 0.5207 and a true negative rate of 0.6341. Our model seems to be overpredicting the negative class. This will lead to a higher accuracy because there are more negative samples present in the data. If we look at the balanced accuracy (balanced accuracy: 0.5774 / accuracy: 0.581) we can see that it’s lower than the conventional accuracy, which occurs when either the sensitivity or specificity is low due to a bias in the classifier towards the dominant class. While the training set has a perfect balanced accuracy/ accuracy of 1, we don’t rely on its evaluations as it is overfitting to the training data.

3. *What would be a better way to select a model than a fixed number of steps that would allow us to set a very high number of epochs and not overfit too much? How would we select the best model using such a ”smart train” function if we are still worried about overshooting our optimum?*

If we want to set a high number of epochs while preventing overfitting, we can make use of early stopping. Here we do monitor the validation metric (such as e.g. validation loss) and stop training once the performance on the validation set stops improving or starts degrading. To fine-tune this approach and avoid prematurely stopping training, a patience parameter can be set to control the number of epochs to wait before stopping. If we are worried about overshooting our optimum due to factors like setting a high patience parameter, we can also do model checkpointing. Here we save the models weights whenever there is an improvement in the validation metric. This way, even if training continues for an extended period, you have checkpoints of the model at different stages, allowing you to revert to the best-performing model if needed. It generally should be added that early stopping is not always a good idea because the model does not always converge in a straightforward manner. In most cases it is a good idea to make training harder using regularization techniques, data augmentation or simply making the model less capable to prevent overfitting. So, it’s essential to consider the characteristics of the specific problem before deciding about if you should make use of early stopping.

1 (b)

1 (c)

1. We define a simple autoencoder which is compatible to our CNN from the previous task. Our autoencoder consist of an encoder and a decoder. The encoder contains four convolutional layers each of them followed by a ReLU activation, BatchNorm2d normalization and a Dropout with probability p. The decoder analogously contains four ConvTranspose2d layers again followed by ReLU activation, BatchNorm2d normalization and again Dropout with probability p.
2. As seen in the image below, the reconstruction of the autoencoder is recognizable but can still be improved a lot. A bigger dataset and more complex net could improve the results of the reconstructed images.

Ein Bild, das Farbigkeit, Screenshot, Grafiken, Grafikdesign enthält.

Automatisch generierte Beschreibung

1. It seems that both of our models (model pre and post transferring the weights) overfit on our dataset which results in very similar results on the test set (pre transferring weights acc: 0.582, post transferring weights acc: 0.588). The accuracy is slightly but not significantly better after transferring the weights from the autoencoder to our model.

**Task 2.**

**2 (a)**

Examine the Latent Space:

t-SNE and PCA of Train Embeddings (Latent Space Frozen Training):

Ein Bild, das Screenshot, Farbigkeit, Text enthält.

Automatisch generierte BeschreibungEin Bild, das Screenshot, Text, Farbigkeit, Diagramm enthält.

Automatisch generierte Beschreibung

The t-SNE Graph displays a clear cluster in the centre of the image (yellow), with (purple) dots mostly clustering around it. The separation and border between these clusters however is not so clear. Both classes also have multiple outliers in clusters of the other class.

The PCA Graph also shows two distinct clusters. Both clusters appear more compact, due to purple outliers far from both cluster-centres. The separation of both clusters is unclear, indicating that the model could struggle in that area.

To conclude: the separations of the clusters leave clear room for improvement, but considering the limited and the simplicity of the model, it is a good sign to identify clusters in both representations.

This (Transferring and Freezing layers) is often done in the literature. Why is that? What are we testing here?

Transferring the weights of the encoder and freezing them in the new model has multiple advantages. As a pretraining method, it is especially useful when there is limited data and or time to train on. The transfer of the encoder specifically can be explained intuitively as the encoder is expected to learn generic (hierarchical) features of the large dataset, features that are useful across tasks and domains.

Freezing the layers can be especially useful when time is sparse, as the number of parameters is limited, reducing the training duration. Additionally keeping the layers frozen can prevent the new training from destroying and overwriting already trained, good features at random.

This (examination of the latent space) is often done in the literature. Why is that? What are we testing here?

Analysis of the latent space can yield information regarding the model’s ability to generalize to the test set. Outliers, unclear clusters, and separations could indicate areas where the model has a hard time differentiating between classes and generalizing to the test set (or other tasks).

2 (b)

Examine the Latent Space again:

t-SNE and PCA of Train Embeddings (Latent Space Trainable):

Ein Bild, das Screenshot, Farbigkeit, Text enthält.

Automatisch generierte BeschreibungEin Bild, das Screenshot, Farbigkeit, Text enthält.

Automatisch generierte Beschreibung

t-SNE:

While we can detect two clusters for yellow and purple in the t-SNE visualization of the latent spaces, they both overlap, creating a poor separation. Additionally, some outliers of both classes reach far into the opposing cluster. These are details, that highlight areas where the model might fail to correctly identify data from the test set, but in general this is a satisfying result for the simplicity of the model.

PCA:

The PCA graph looks similar to the PCA graph of the model with frozen weights. The main difference is that there are fewer outliers far from the purple cluster in this version. There are clusters of both yellow and purple dots, with a significant intersection between them. This indicates that the model could struggle to generalize in that area.

What worked better? Freezing or leaving the weights trainable? Why?

Evaluating the plots, always includes a subjective component. Having said that, evaluating the PCA graph of the version with trainable weights indicates a slightly reduced vulnerability to outliers. Looking at the t-SNE graphs, the trainable model displays a more connected purple cluster, but also has a worse separation between the clusters. Both t-SNE graphs contain outliers in the cluster of the other class. Both methods seem to return similar results. Additionally considering the negative rate and (balanced)-accuracy leads us to believe the trainable model performed slightly better.

2 (c)

For completeness - Latent space (chest\_model):

Ein Bild, das Screenshot, gelb, Farbigkeit enthält.

Automatisch generierte BeschreibungEin Bild, das Screenshot, Farbigkeit, gelb enthält.

Automatisch generierte Beschreibung

We examined the latent spaces for the original chest model, the frozen pneumonia model, and the trainable pneumonia model.

The chest model was trained on few samples and heavily overfitted to the training data. The Latent space examination depicts this, as both graphs are chaotic with no discernible clusters.

The frozen pneumonia model utilized the pretrained encoder of the chest model that utilized the full dataset. The indirect access to this increased amount of data led to less overfitting and improved performance on the test set. The latent space displays two clearly recognizable clusters, with fuzzy borders and outliers.

Similarly, the trainable pneumonia model utilized the pretrained encoder but was allowed to optimize its weights. In our experiments this slightly improved the generalization abilities of the model compared to the frozen network. The latent spaces had fewer outliers and slightly better separations between clusters.

The latent space can be utilized to check how well the learned features in the encoder generalize to the test set. Unlike the chest\_model the pretrained encoder in the two pneumonia models had access to a larger dataset during training to avoid overfitting. This increased generalizability to the test set largely explains the drastic differences between the plots.

**Bonus**