ML Assignment 2 Report

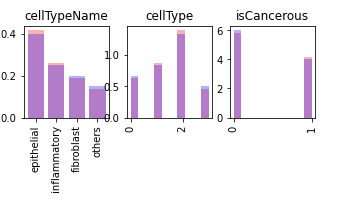
By Jack Allen s3832293 and Vijay

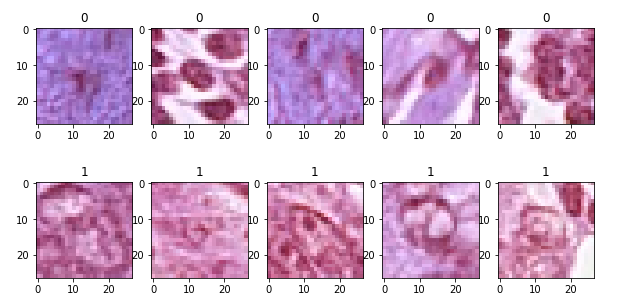
The task was to develop two machine learning models that take an image of a cell as input. One model is required to accurately predict if a cell is cancerous, and the other model can predict the type of cell. Initially both models were approached in a similar way. We began by developing a base model using a traditional neural network. From this we expanded on each model with VGG deep neural networks.

In creating ML models that can predict if a cell is cancerous, we decided to use the data labels main data csv file as well as the data labels extra data. We took this approach because the more data we have the better the model can be trained. When conducting the EDA, we began by looking at how the data is distributed. We did this because if we have a lot of data in one area but not others our model may be better at predicting certain things and not others. This was also done to check if the data is distributed well between the train and test data sets. Appendix 1 shows that there are many more epithelial cell types than others in the data and slightly more non-cancerous cells than cancerous. Though the test set has a very similar distribution of data on each cell and whether the cell is cancerous as the train set. We also decided to look at the images of non-cancerous and cancerous cells to see if there is any clear indicator of a cancerous cell that can be picked up by the human eye, these images can be seen in appendix 2(0 are non-cancerous cells and 1 are cancerous). As can be seen in appendix 2, there is no obvious way of telling which cells are cancerous and which are not based on the images.

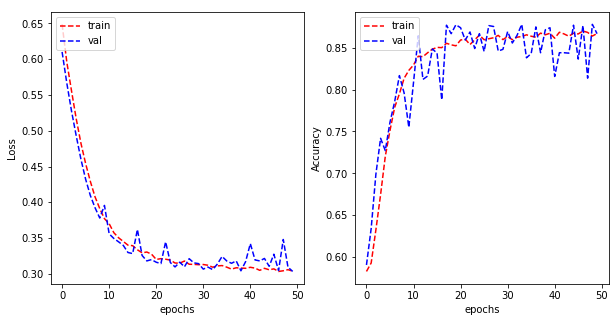
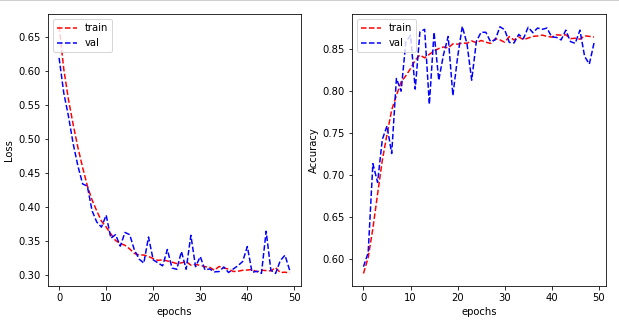
Now it was time to start the process of creating and testing models. The traditional neural network was used to begin with to establish a base level of accuracy that the other models can be compared to. In creating the base model, we also took the opportunity to test how different numbers of neurons in the hidden layer affect the accuracy. We tested with 256, 512 and 724 neurons. The respective learning curves for these models can be found in appendix 3 in order. All 3 models performed very similarly, thus we concluded that the number of neurons in the hidden layer had little affect on the accuracy. In further testing 724 neurons seemed to perform the best so we used that for further models. The aim was then to build on this with a more complex model and to ultimately achieve a greater accuracy. VGG was used because it is one of the most used image-recognition architectures. The VGG model we created used 3 blocks and 50 epochs. This model outperformed the base model but was overfitting as can be seen in appendix 4. To prevent overfitting, we created a total of 9 models testing different regularization and dropout values and learning rates. After this testing was completed, we ended up with a final VGG model that can be seen in appendix 5. This model had the highest accuracy (0.9035) out of those tested and did not overfit or underfit.

**Appendices**

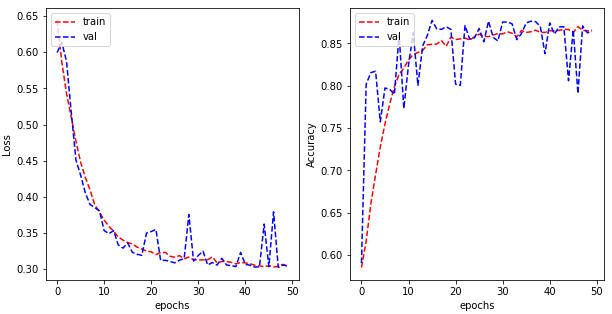
**1**

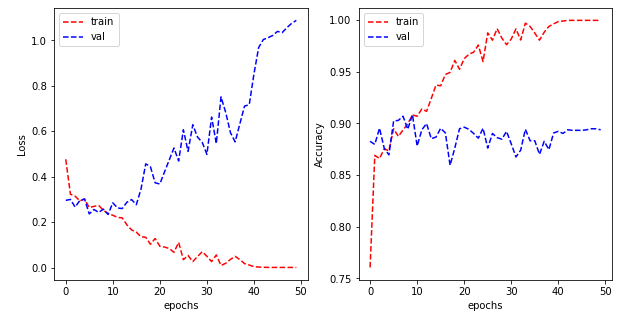


**2**



**3**







**4**

**5**

