Histopathology Classifier (Project 1)

Sofya Filippova (s3848979), Mitchell Gust (s3782095)

1 Approach

1.1 Model 1 - Classifying Cancerous Cells

1.1.1 Baseline Model

Implementation

A basic MLP model was chosen for our baseline. It is simple, easy to understand and can model complex non-linear relationships between inputs and outputs. On the contrary, decision trees and rule learning algorithms are inherently limited in their ability to represent non-linear patterns. On top of this, MLP models reduce the need for manual feature engineering, unlike the other mentioned methods.

We make use of both the main and extra data .csv file to help build a more robust model with better generalisation and greater accuracy.

A hidden layer of 256 nodes was used in order to increase complexity, as opposed to using smaller values. A model with higher complexity has more potential to capture complex patterns within the data and generally leads to better performance.

Stochastic Gradient Descent is an efficient and easy to implement optimisation technique for training models. Its iterative computation on small sets of data minimises the duration of computation and seems like a good fit for our baseline model. Though the Adam optimization algorithm is also efficient, Adam may be more susceptible to overfitting and may not work as well on our simple binary classification model.

A Sigmoid Function is used as the activation function of our output layer due to its ability to map input values within the range 0 and 1. These mapped values can be interpreted as belonging to one of two classes and thus a good fit for our binary classification problem.

Evaluation - Examining Figure 1 in Appendix.

The training and validation loss both descend in value with an immediate start. It reaches a low value and begins to stabilise around 0.35. Some level of convergence behaviour is shown however, it is clear bumpy behaviour is present. Following the general trend, we observe slight underfitting. On top of this, we also observe the accuracy line plot beginning to stabilise at 0.86. This value is okay but could be better. This could indicate that the data is not perfectly represented by the model. Thus, a neural network with more hidden layers or a larger amount of neurons, i.e. a deep learning network may be more suited to represent the data. Therefore, we can increase the number of neurons in the hidden layer and the number of hidden layers of the neural network to create a better representative model.

1.1.2 Model 1 - VGG 1

Implementation

Our next approach is a Convolutional Neural Network with a VGG architecture. Included are 3 VGG blocks. We used the ReLU activation function in all the hidden layers, as it introduces non-linearity into the neural network, allowing the algorithm to learn complex patterns and relationships in the data. Furthermore, compared to many other popular activation functions, it reduces the vanishing gradient problem. Also specified is a padding strategy which adds extra values to preserve the original input dimensions. In each of the VGG blocks, max pooling is then performed to downsample the images dimensions. Though this involves losing information, it may improve computation time and improve identification of prevalent features by selecting maximum value pixels. In the output layers, the feature map is flattened for processing, sigmoid is used to output binary classification, and a dropout layer is used to improve generalisation and hopefully minimise overfitting.

Evaluation - Examining Figure 2 in Appendix.

Shown in the outputting metrics, we observe an accuracy that is high in training (0.97) but low and quite static in validation values (0.88). Also shown is a large and increasing gap between the training and validation data. These are two symptoms of overfitting and could be the result of our model not capturing and learning from underlying patterns. This model may perform better when regularisation is applied to the loss function so that it could become less influenced by particular training data and more reliant on generalised patterns.

1.1.3 Model 1 - VGG 2

Implementation

This iteration implements L2 regularisation on all convolution layers (Convo2D) and the model's output layer in an attempt to regulate reliance on individual weights and encourage generalisation. It is also important to note that we initialised the regularisation using a generally moderate reg_lambda value of 0.001. Furthermore, ReLU was used as the final output function for reasons discussed above. During Hyperparameter tuning, it proved to be a better fit for our model than sigmoid. Furthermore, the number of epochs was reduced from 50 to 25, as after epoch 25 output graphs showed overfitting and potential memorization of training data.

Evaluation - Examining Figure 3 in Appendix.

This modification reached high levels of accuracy which appears to be continually rising beyond 0.89. Our loss value depicts a negative trend which features some correlation to an eventually stabilised value of 0.30. Both plots exhibit a level of good fit, the loss value is low and our accuracy at the end of the 25 epochs has reached a high value.

1.2 Model 2 - Classifying Cell Type

1.2.1 Model 2 - VGG 3

Implementation

The model is based on the final model produced for the binary classification of cancerous cells, however adjustments are made to represent the non-binary problem. Due to the greater complexity of a non-binary problem a fourth VGG block was added to the model and the number of neutrons were doubled. These two specific changes aimed to increase performance for an arguably more complex task.

Evaluation - Examining Figure 4 in Appendix.

Viewing the resulting line charts, we observe the validation accuracy value shows signs of correlation with the training values. It follows an incremental positive trend and reaches 0.75 by epoch 50. We observe an immediate and negative descent in the loss value, reaching 0.7. There is little to no sign of plateauing and it could be the case that the loss values could continue to descend.

1.2.2 Model 2 - VGG 4, Improved using Extra Data

Implementation

Since the extra data was unlabeled with cellType, semi-supervised learning was used to perform transfer learning. After VGG 3 was trained on the main data provided, its weights were saved and then used to predict labels for extra_data.csv. The two datasets were then concated and used to train VGG 4. More data generally leads to better performance and higher generalisation. Due to the improved dataset, it seems no longer required to have such a complex model, therefore VGG 4 had 1 less block of hidden layers.

Evaluation - Examining Figure 5 in Appendix.

Analysing the graph we can see an increase in learning and accuracy. However, when compared to the previous iteration for model two we can see a much higher jump in accuracy, signifying that combining transfer learning with extra data did improve the model. The number of epochs did have to be lowered in order to avoid run time errors, especially since working with a large dataset. Furthermore, this was done to reduce overfitting.

2 Ultimate Judgement and Evaluation

Examining Figure 6 and 7 in Appendix. For classifying isCancerous: VGG2 model was the best iteration for this task. It had the best accuracy and loss, as well as no visible signs of underfitting or overfitting. Examining the confusion matrix in figure 6, it is clear to see we have an extremely high number of true negatives, meaning our model is extremely effective in classifying patients who don't have cancer. The model was not as effective at classifying positive cases of cancer. This could have been due to the complexity of cancer diagnosis.

For classifying cellType: VGG4 model was the best iteration for this task by achieving the highest accuracy. Examining the confusion matrix in figure 7, we can see the model was the best at classifying epithelial. Examining the EDA, a possible cause of this was class imbalance. If given the opportunity to redo the assignment, techniques such as under-sampling would be used. This involves reducing the number of samples in the epithelial category to prevent the model having a bias towards it.

The precision, recall and F1-score were also low, especially when compared to the results of Sirinukunwattana et al. [1]. The main improvement in this study over ours, is the use of detection before classification. Detection minimises the search space, focusing only on regions of interest. This in turn, improves efficiency and robustness against low visibility.

3 References

[1] K. Sirinukunwattana, S. E. A. Raza, Y. Tsang, D. R. J. Snead, I. A. Cree and N. M. Rajpoot,"Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1196-1206, May 2016, doi:10.1109/TMI.2016.2525803

4 Appendix

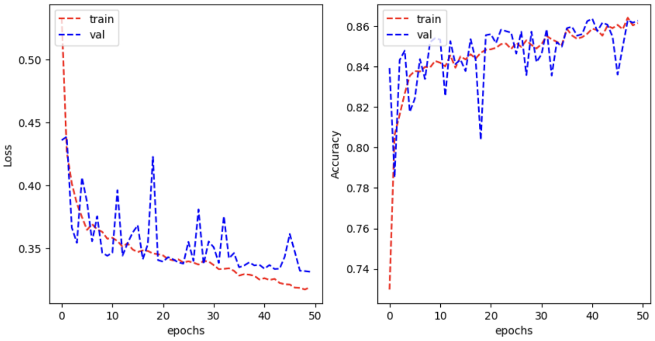


Figure 1: Model 1 - Baseline

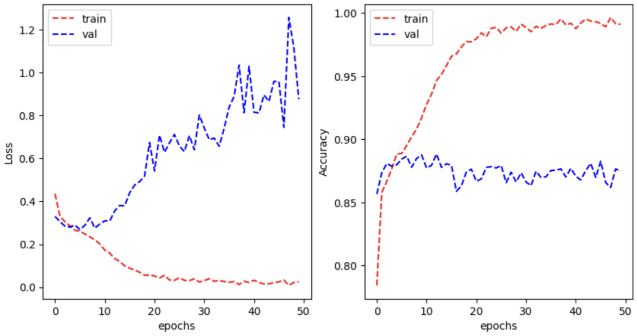


Figure 2: Model 1 - VGG 1

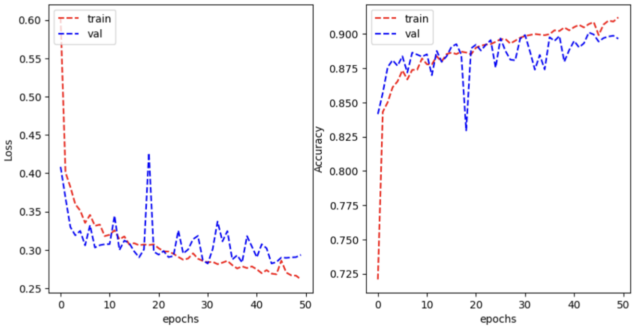


Figure 3: Model 1 - VGG 2

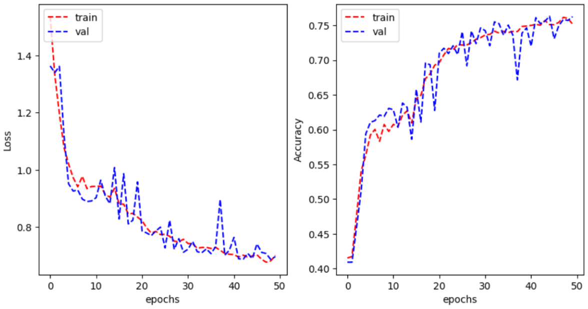


Figure 4: Model 2 - VGG 3

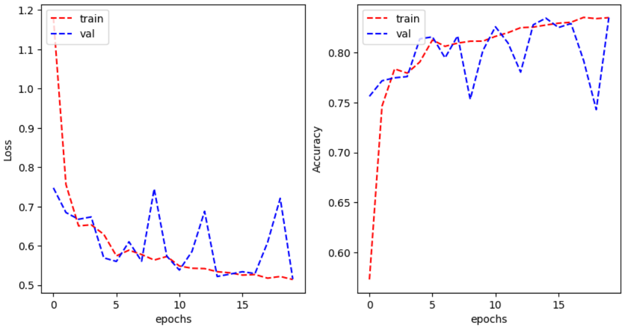
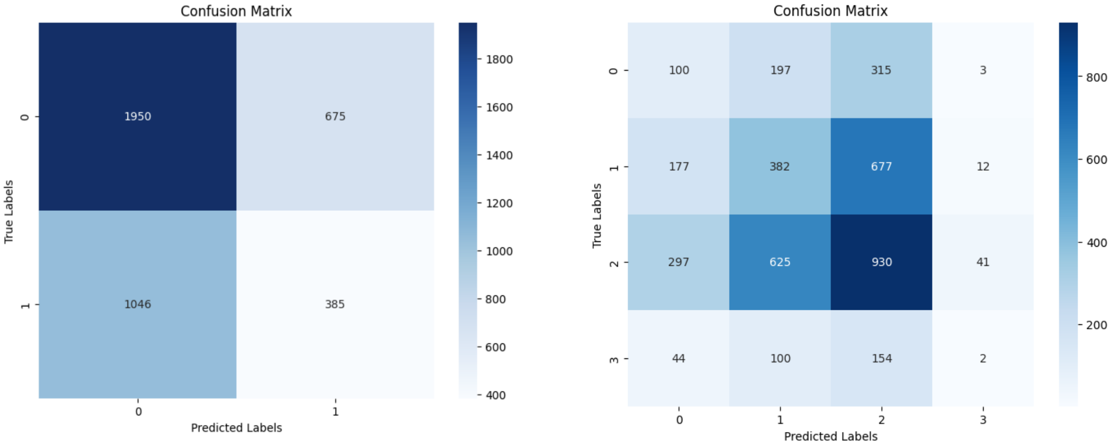


Figure 5: Model 2 - VGG 3 using Extra Data



(a) Model 1 - Confusion Matrix Testing

(b) Model 2 - Confusion Matrix Testing

Figure 6: Model Evaluation