Machine Learning Assignment 2: Project 1 - Colon cancer classification

Rhys Reid Mallia, S3656436 **The size requirement forces me to not use spacing, please forgive me :)**

Introduction

The goal of this project was to build a series of classifiers in order to attempt to accurately identify two categories of image from a modified version of the CRCHistoPhenotypes dataset, which contains 22444 27x27 images of colon cells and two associated data label excel documents .

The first category, isCancerous, is a binary classification of a colon cell which represents cancerous cells or not. The second category, cellType, is a n-ary classification of a colon cell which identifies the type of cell from a basis of 4 different cell types, fibroblast, inflammatory, epithelial, and others.

The task was achieved through various means taught through the course. First, the data was prepared for the machine learning algorithms, and a Principal Component Analysis (PCA) and associated T-distributed stochastic neighbor embedding (t-SNE) was used to investigate the dimensionality of the data and to visualize the decision boundaries of each class within the two categories of prediction so that correct models could be used.

Ultimately, the four models used for both tasks were the same.

- Support Vector Machine (SVM).
- k-nearest neighbor classifier (KNN).
- Random Forest of Decision Trees (RFC).
- Convolutional neural network (CNN).

The models which were considered include …

- Regularized Logistic Regression
- Gradient Boosted Tree
- Naive Bayes

Each of these models was tuned via their hyper parameters and trained and tested on an unforeseen holdout dataset.

Ultimately, in each of the two categories, the most effective technique was found to be the Random forest of choice (RFC), followed by the K-nearest neighbors (KNN), support vector machine (SVM) and the convolutional neural network (CNN), in that order.

The KNN, which achieved the best scores on the test dataset, did not perform as well on the holdout dataset, often losing a third of its performance, which can be attributed to overfitting on the test data.

Data exploration

Through an exploration of the data, it was found that the dataset was somewhat unbalanced. ***(fig 1)***

In the isCancerous dataset, there was a bias towards images classified as isCancerous[0], which is a non-cancerous classification.

CellType contained a bias as well, this time towards a classification of cellType[2], while at the same time the cellType[3] class was underrepresented in the overall dataset.

Both of these bias' will result in a model which cannot accurately predict unrepresented data, and which will overpredict classicaitions with a bias.

The mean image ***(fig2)*** and variance of the dataset was calculated and found that each image was accurately centred and that a majority of the information which was different between images was centered as well, however some data was important on surrounding points of the mean. This means that augmentation such as horizontal and vertical movement would risk obscuring important information, and an augmentation such as horizontal flip would be better for training purposes.

Principal component analysis (PCA) was applied to both datasets.
PCA is used to find the strongest trends that exist within the data, followed by the next strongest and so forth. Each of these are mapped linearly to a lower dimension, which means that decision boundaries can exist and be used in these lower dimensions.
For both datasets, an attempt to reduce the images to a two dimensional and three dimensional fit was made. However, with both datasets, the tightly coupled trends showed only an increase in complexity upon being reduced, and would only further increase the difficulty in finding a correct line of fit for the machine learning models ***(fig 3)***
This was more visible in the cellType category, where the 4 classes were highly coupled and mixed together, even considering outliers, each reduction only further tightened the classes together.
No clear decision boundaries were visible, and this was interpreted as an indicator that higher dimensional data would produce a better result and a better fitted model.

T-distributed stochastic neighbour Embedding ***(fig 4)*** was also applied to both datasets.
t-SNE is a technique for representing higher-dimensional data whose features exist on lower dimensional subsets. It works by converting the data into a matrix of similarities and converting the distance between data points into conditional probabilities. While not being suitable to use as a dimensionality reduction technique, it can be helpful for determining decision boundaries.
This method was not helpful for our tightly coupled data, and did not show any clear decision boundaries on either dataset. This was seen as an indicator that a higher dimensional dataset may be more helpful in classification, and that the models which should be selected would need to be able to specialise in non-linear decision boundaries.

Selected models
The following models were selected based on the requirements found in the data analysis ...
Support Vector Machine (SVM)
The SVM was picked for its ability to model non-linear boundaries, and because of their robust nature to avoid overfitting in a high dimensional space. They do suffer from a high memory usage and an upper limit on the size of the data, however in this case neither of those constraints were an issue, this may not be true in implementation.
K-Nearest neighbors (KNN)
The KNN classifier was picked for its simple implementation and robust ability to find the correct search space. This was important as our data could not be linearly separated. This classifier also has the advantage of quick operation and the ability to add new instances quickly and efficiently.
Random Forest of Decision trees (RFD)
The RFD classifier was picked for its good performance, the RFD as with the other models also handles modelling non-linear decision boundaries well, and is very robust when it comes to scale, handling outliers and their natural hierarchical structure.
Convolutional Neural network (CNN)
The CNN was an obvious choice due to it’s proven track record when it comes to its performance in high-dimensional data and identification of nonlinear decision boundaries. The CNN is also extremely generaliziable when it comes to unforeseen data and is able to be modified to far greater extent than any of the other models, however, this is also an

obvious downside and great care must be taken when creating the CNN, it also benefits from a far greater size of dataset compared to our small sample.

Evaluation of performance

The evaluation of the performance of each classifier on the two datasets is surprising ***(fig 5),*** however, common to both datasets is the outcomes of the classifiers.

The random forest was found to have the most accuracy on predicting the unseen data. Followed by the KNN, the SVM and finally the CNN.

This was obviously due to the robust nature of KNNs which allow outliers and scale to be accurately considered, the SVM surprisingly did not perform as well, which may be related to the kernel chosen, however in most industry applications a random forest is usually preferred and this has reflected those ideals correctly.

Each non-CNN classifier was tuned and examined in the same way. Using a grid search and validating through K-cross validation, the correct hyper parameters with the greatest performance could accurately be found and tested.

Each classifier other than the CNN showed heavy overfitting during the k-fold validation, however this seems to not have negatively affected the prediction on the holdout data, which is surprising.

The CNN performed with a high level of accuracy during the training phase, and showed little overfitting to the validation data, however, when put up against the testing data, a clear overfit to the testing was found. Both CNN models failed to maintain their accuracy and suffered heavily, gaining the last spot out of all the models.

Given the nature of CNN's, this can be accurately attributed to the creation and development of the CNN models. In its current form, each layer was modeled and inspected for their performance on the datasets, and given that data could only be pooled 4 times before the data is 1x1, a high number of convulsions was explored **(fig 6)** and tested, which only increased accuracy at a near parallel rate, indicating an overall low overfitting of the model.

Adan was chosen as an optimizer as it allowed quickly training better models with a good balance of performance for low computational power.

Each CNN model was put through early stop and model checkpointing before being run over 200 epochs in order to properly consider the entire growth of the model.

Ultimately, the CNN showed lower scores in F1, recall and precision as well as accuracy based on the other models, and giiven more time and consideration, a better and highly effective CNN model may be able to be created, however, in its current form the CNN cannot be recommended as an accurate classifier for either dataset, a random forest is preferred.

Independent evaluation

In contrast to my own approach, the following two papers approached the problem much differently.

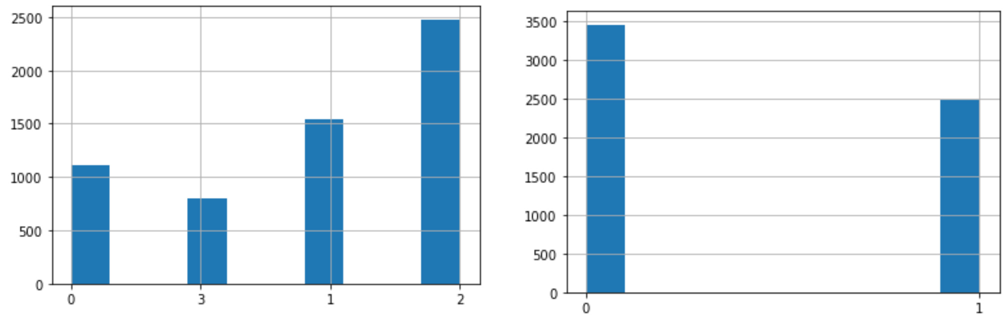
The first paper, 'Deep learning with sampling in colon cancer histology', created an algorithm that consisted of two neural networks working in series. The first simply identified the cancer cell, the output of which was a cell map, and passed the coordinates to a second CNN which categorized each cell as one of the four categories. The model itself, based on the cifar10 model, used a much higher batch size, and a lower learning rate, which may contribute to some of its successes other than the CNN design.

Appendix:

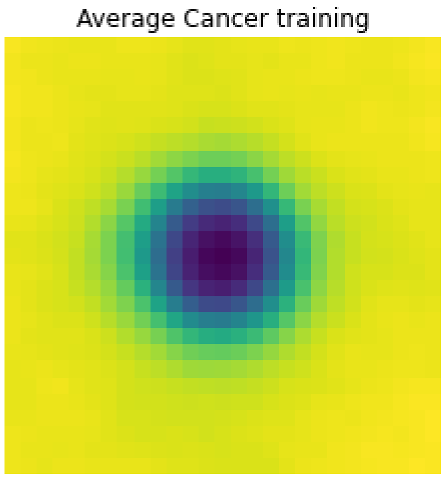
Citations

AUTHOR=Shapcott Mary, Hewitt Katherine J., Rajpoot Nasir
TITLE=Deep Learning With Sampling in Colon Cancer Histology
JOURNAL=Frontiers in Bioengineering and Biotechnology
YEAR=2019
PAGES=52
URL=https://www.frontiersin.org/article/10.3389/fbioe.2019.00052
DOI=10.3389/fbioe.2019.00052
ISSN=2296-4185

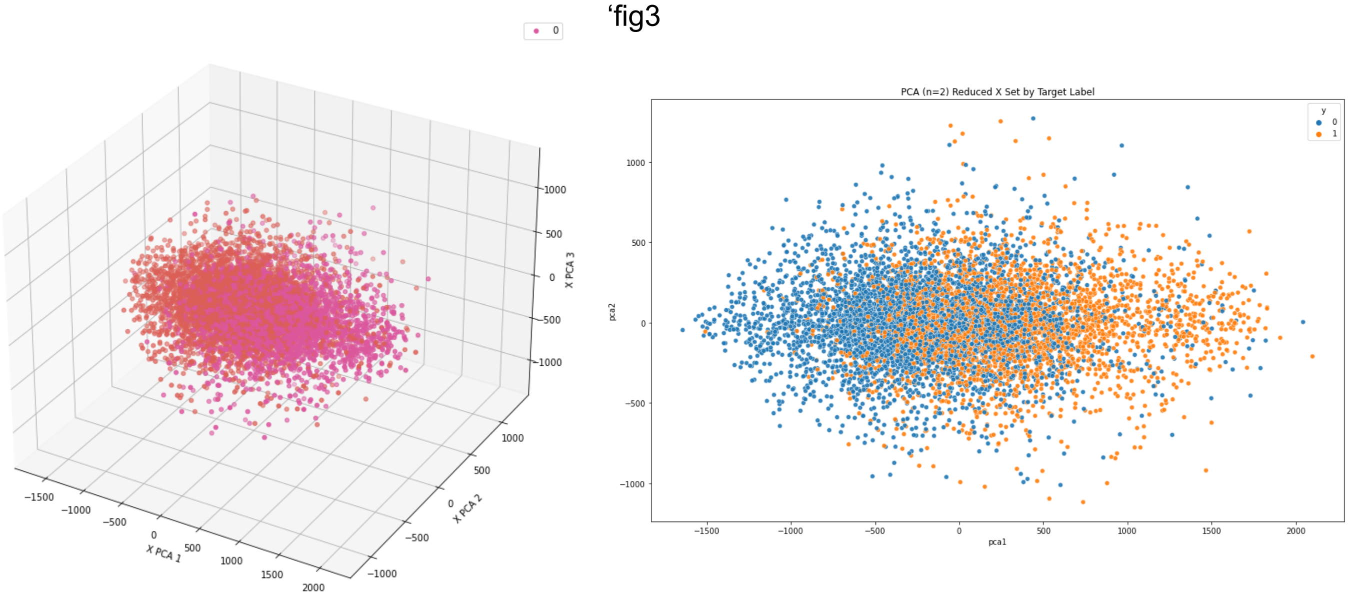
(fig1)

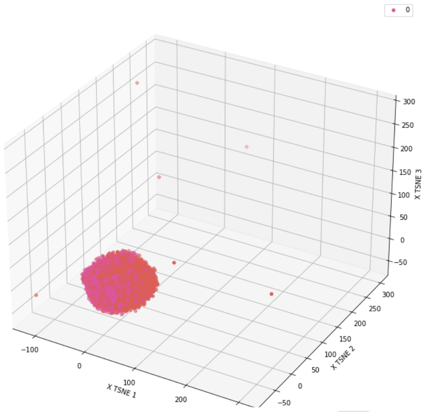


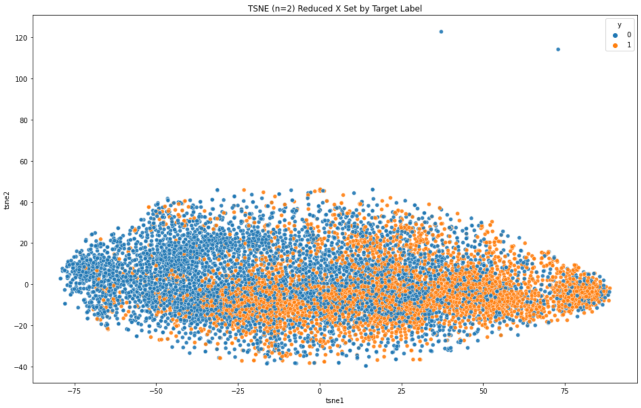
(fig2)



‘fig3







(fig 4)

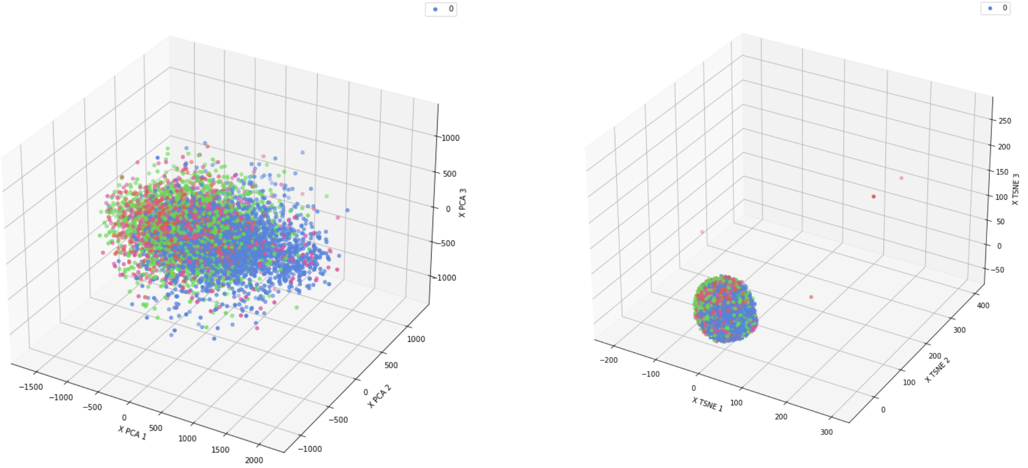


Fig5

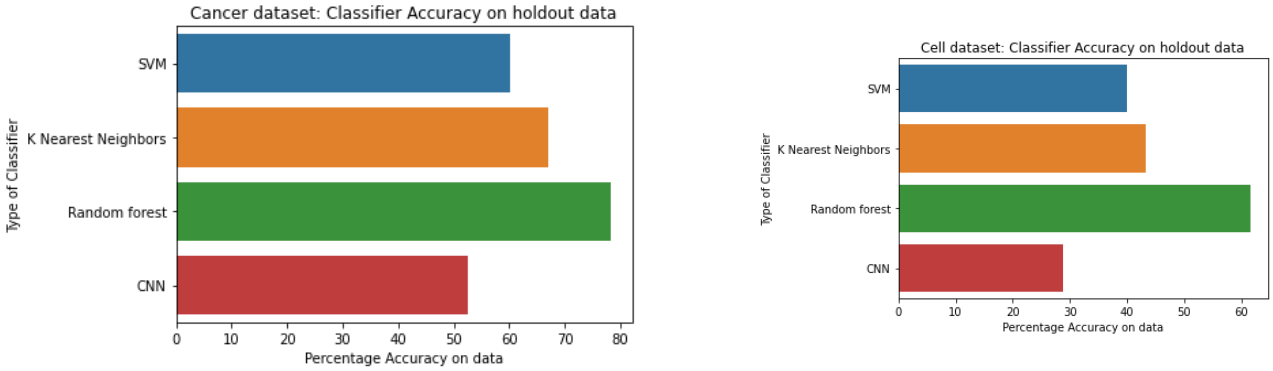
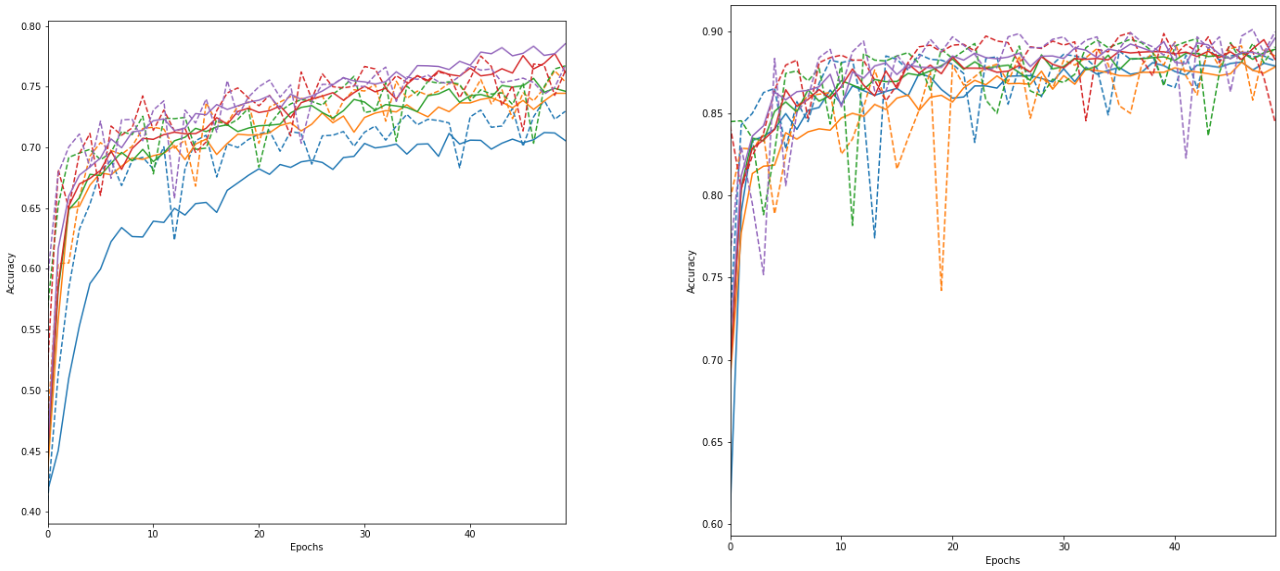


fig6


Accuracy
Epochs
Accuracy
Epochs