Classifying the Stage of Hepatocellular Carcinoma Using Transfer and Deep Learning

Rachel Theriault
Queen's University, School of Computing
21-25 Union St., Kingston ON, CA
15rlt@queensu.ca

Abstract

Liver cancer is one of the most common and fatal malignancies world-wide with hepatocellular carcinoma making up a majority of its reported cases. The classification of the stage of liver cancer has a large impact on the treatment provided, and survival prognosis. Recently, connections between a subset of genes and potential pathways have been linked to detecting stages of hepatocellular carcinoma however, analysis has remained linear and classification models have yet to be proposed. I propose use of a deep learning model for classification of the stage of hepatocellular carcinoma from genetic features selected via a lasso model versus a variational autoencoder. To address the concern of large feature numbers and small datasets in genetics, transfer learning from pan-cancer data was used for the variational autoencoder portion. It is expected deep feature detection and linear feature detection will result in different classification performances, and detect different genetic patterns for cancer stage classification. If this proves true, deep learning may be a valuable tool to further explore genetic patterns related to the stage of hepatocellular carcinoma, and for enhancing cancer stage-classification.

The code is available at: https://github.com/theriaultr/CISC867_Project

1 Introduction

Liver cancer is the most fatal malignancy worldwide and hepatocellular carcinoma makes up 90% of all liver cancer cases[1]. The survival prognosis of hepatocellular carcinoma is highly correlated with stage of cancer at time of diagnosis. The stage of hepatocellular carcinoma determines treatment plans[1] and as a result, misclassifications have a large patient and clinical level impact. Current gold-standard detection of cancers is physical analysis of tissue samples by a pathologist which is a time consuming process and known to have variations amongst cancer types. Detecting the stage of cancer is a crucial step in the cancer diagnosis pipeline. To try to enhance accuracy and decrease variation, other metric and data modalities for identifying cancer stage have been explored. A promising modality is genetic analysis.

There are many forms of genetic data analysis such as copy number variants that detect differences in repeats of genes, mutation analysis, methylation of genes as an indirect measure of gene activation, and finally gene expression. Gene expression can be determined by analyzing the number of micro RNA (miRNA) or messenger RNA (mRNA) fragments associated with a gene within a cell. miRNA inhibit the processing of gene copies and therefore increased levels suggest gene inhibition of their associated genes. Alternatively, mRNA expression represent activation of a gene. Through cellular machinery mRNA are translated to proteins, which directly impact cellular activity and behaviour. Therefore, monitoring mRNA is monitoring cellular activity. A common way to detect mRNA is through RNA sequencing analysis (RNAseq)[2]. After alignment to a reference genome, the expression of a gene can be compared across samples.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

As the technology and availability of genetic data is increasing, researchers are turning to genetic data to uncover mysteries of diseases such as cancer. Using RNAseq data, Sarathi and Palaniappan discovered relationship of specific genes to detecting the stage of hepatocellular carcinoma. Although only linear analysis was performed, results were promising for the ability of genetics to detect cancer stages[3]. Linear analysis is often the first stage of gene discoveries however, it is limited in the complexity of patterns it can detect.

Application of deep learning to genetic research has been shown to be a promising technique for continuing discoveries. A benefit of deep learning is that a deep network enables discovery of complex non-linear relationships within the data. In genetic analysis, genes are considered as features, and individual samples are observations. Using point mutations, Sun et al were ables to reach high accuracies in predicting cancer-type between different tissue[4]. The use of gene expression data and a deep learning pipeline also displayed high-accuracy subtype performance[5]. The addition of deep genetic analysis for gene expression data has also been shown to enhance accuracy of previous classifiers. For example, compared to analyzing only tissue sample images, addition of a deep learning genetic and clinical data pipeline enhanced accuracy of survival prediction in the PageNet model[6]. Although a promising technique, the use of deep learning for genetic data faces many challenges.

Genetic data suffers from the curse of dimensionality - an mRNA sample contains the expression of over 20,000 genes and often datasets larger than a magnitude of 100 are unavailable for genetic cancer data. As deep learning using high dimensionality data contains a significant number of hyperparameters to train, developers have had to develop creative network and training alternatives to enhance algorithm robustness and generalizability. Kim et al demonstrated how use of transfer learning enhanced robustness of their algorithm for miRNA data. Authors pre-trained a variational auto encoder (VAE) on genetic data from multiple cancers and transferred feature reduction layers to a feed forward network for survival prediction[7]. Alternatively, inclusion of previous biological knowledge has been a popular technique to decrease the number of features and enable interpretability of results, particularly through the use of gene set enrichment analysis (GSEA) [8]. By identifying the pathways genes are associated with, GSEA enables sparse encoding and biologically-relevant dimensionality reduction of the data. This is a software package that based on genes present can identify biological pathways active in the data. This has been shown to aid in biological interpretation of selected genes in classification tasks[5] and incorporated into multiple networks as a form of data encoding[9, 6]. Despite its growing prevalence in the field of genetics, limited work seems to have been done in use of deep learning for classifying stages of cancer.

For this project, I hypothesize using deep learning feature reduction and linear feature reduction via lasso will result in identification of different gene patterns and therefore, different accuracies of hepatocellular carcinoma stage prediction. Use of a lasso method for feature reduction will enable comparison to results found by Sarathi and Palaniappan[3]. I will extend their work by using feature selection in a deep fully connected network for stage classification. I will compare this method to use of a variational autoencoder for feature selection. I will use the same VAE transfer learning method proposed by Kim et al. [7]. I will transfer learned weights to a deep fully connected network to classify hepatocellular carcinoma, fine-tuning all layers for the dataset.Results of the deep network will be compared to the lasso methods to determine if deep learning enhanced cancer stage classification. If successful, this project will act as a proof of concept for use of deep learning in detecting stage of hepatocellular carcinoma, and for identifying complex genetic patterns. If the lasso method proves more successful, this will suggest either not enough data is available or, deeper patterns do not enhanced classification and monotonic/linear patterns should be explored in greater depth.

2 Related Work

Although it has lead to enhanced predictive abilities, the use of deep learning for processing genetic data poses an interesting set of challenges. Primarily, genetic data has an extremely high number of features and typically small datasets. This can make production of generalizable deep models extremely challenging. Explainability of deep learning networks also poses a challenge, especially in the medical domain where explainability is crucial for a clinically deployable system. Due to these challenges, researchers have had to become creative in the application of deep learning to genetic data and still, many groundbreaking discoveries first occur at the linear level for which the results can be more easily interpreted.

In 2019 Sarathi and Palaniappan proved the predictive power of hepatocellular carcinoma at the gene level through linear analysis[3]. Using both a linear model and pairwise comparisons, the group identified genes that were able to predict cancer stage based on their expression levels in RNAseq data. From the linear model, genes with the largest coefficients were selected as important genes for differentiating between cancer stages. based on their coefficients for each stage, their up or down-regulation were interpreted. Each stage was also pair-wise compared to control samples and the genes that displayed largest differentiation in expression were selected. These genes were then used to compare the stage to all other stages to determine significance in differentiation between the stages. Using numerical statistical cutoffs of significance, the final genes were selected as important for differentiating between stages however,some genes were selected as important in multiple stages but assigned to only the one it displayed the most significance in predicting for. This discovery proved that gene level correlations to cancer stage exist. The importance of genes in multiple stages suggest non-simplistic patterns may exist in these correlations - something deep learning is designed to detect and that was not accounted for in the described analysis. The analysis did not utilize a classification method to investigate effectiveness of the selected features in detecting cancer stages. Application of deep learning to stage prediction would reveal non-linear patterns for prediction. The challenge in working at the gene level, is how to enable interpretability and robustness of the model given the large dimensionality and small datasets available in cancer genetics.

Gene set enrichment analysis (GSEA) developed by Subramanian et al. is a common method of accounting for both genetic interpretability and enabling sparse encoding in genetic networks [8]. Authors produced software that can predict active pathways based on genetic expression. Pathway analysis is higher-level analysis than gene analysis and contains a much smaller feature set; therefore, GSEA can be used in 2 ways: 1) biological interpretation of results or 2) network simplification. For example, GSEA was used by Sarathi adn Palaniappan to determine potentially active pathways based on their genetic-level linear analysis of genes and hepatocellular carcinoma stage[3]. GSEA can also be used to develop a pathway layer in a network with sparse encoding from input genes to pathways. This used in the CoxPasNet network which interprets genetic and clinical data for survival prediction [9]. Similarly Gao et al. implemented functional encoding using GSEA in the classification of molecular subtypes of cancers [5]. In both models, use of GSEA enabled biologically interpretable results, and decreased network complexity. Inclusion of GSEA also lead to enhanced robustness and superior performance of both algorithms. Incorporating GSEA as prior knowledge results in pathway-level analysis and acts to ensure only useful gene patterns are explored and decreases the number of parameters for the remainder of the network. The challenge with use of GSEA is one must know which pathways should be explored for the problem at hand. In exploratory work, this is not always known and instead, using GSEA to analyze results like Sarathi and Palaniappan[3], may be more feasible in the context of pattern discovery.

An alternative form of feature reduction is use of deep learning algorithms called autoencoders for feature selection. Autoencoders reduce dimensionality of data. They are symmetric networks that evaluate performance by reconstructing the input data from reduced data and comparing the results using a loss metric. This enables data to be reduced in a way that minimizes information loss. This acts as an alternative to linear analysis techniques such as linear regression and lasso [10]. Auetonecoders can be applied without prior knowledge to identify important features in the data however, they contain many weights and as a result, require large quantities of data to train. Despite the challenges in small genetic datasets, Danaee et al. demonstrated stacked denoising autoencoders enabled successful feature reduction of genetic breast cancer data by accounting for small unbalanced datasets via synthetic minority oversampling[11]. To perform synthetic minority oversampling in biologically meaningful way, one must have strong biological understanding of their data. Alternatively, Kim et al instead employed pan-cancer transfer learning [7]. Using a VAE, authors trained using 20 cancer datasets to identify general cancer important features. Authors then transferred the reduction layers to a network for survival prediction. This enhanced both robustness and performance of the network. Authors suggested use of prior knowledge to further avoid over-fitting however, this cannot be used in discovery-based networks such as the one proposed in this paper. Although linear regression models do not generally require as much data as deep learning models, feature selection does not investigate the non-linear relationships in the same way as deep methods such as VAE. In both methods presented, biological interpretation of the results was not discussed. Using an autoencoder for dimensionality reduction, GSEA can be applied to interpretation of the results.

3 Methods

The following section contains a very rough draft of the methods section. It will be enhaned for the final report.

3.1 Data preprocessing

The data used for this project were downloaded from Broad GDAC Firehose [12], a TCGA-associated website. RESM normalized alignment [13] files were used. For training of the VAE, all available datasets were used. For classification and lasso analysis, only the LIHC dataset was used and only the samples used by Sarathi and Palaniappan were considered (399 samples of 20531 genes). After visualization of the data, it was determined the data needed to be log transformed to enhance analysis(Figure 3). After log transformation, the data was standardized using z-score feature-wise methodology mimicking Kim et al [7] for use in the VAE. The same standardization was used for lasso analysis. This ensured all feature expression was zero-mean with a standard deviation of 1. Following the methodology of Sarathi and Palaniappan [3], all substages were collapsed into their parent class for further analysis; for example, stage ia and ib were combined into a single stage i class.

The methodology that will be employed to handle missing values for training of the VAE is still under investigation - currently missing expression values are replaced with a value 0.

3.2 Lasso

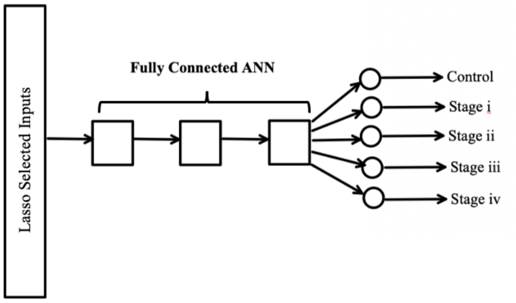


Figure 1: The architecture used for the Lasso-based classification network are displayed. Lasso was completed in MATLAB and selected features were used as inputs to the model.

Since the effects of deep feature extraction for stage classification of hepatocellular classification are currently unknown, lasso was performed as a comparison method. Lasso analysis was implemented in MATLAB. Lasso was tested using the LIHC dataset and the same samples used by Sarathi and Palaniappan in their linear analysis of the same dataset [3]. This enabled direct comparison of linearly determined important genes for the classification of each stage.

Lasso was implemented in a classification approach - binary classification was performed in a pair-wise fashion of each stage to control sample only, and for each stage to control and all other stages. The selected features were from only the comparison of each stage to control and all other stages. Selected genes were compared to those found by Sarathi and Palaniappan [3] and the standardized, log-transformed data was stored in a CSV for only the lasso-selected features. The CSV was then transferred to Python and used as input to the classification network displayed in Figure 1.

The classification network was used to predict stage from the reduced LIHC dataset. The layers, and number of neurons in each layer will be tuned during the official training process, but the current proposed architecture is 3 hidden layers and 1 output layer. The input layer will be the number of genes (features) selected by Lasso. The current proposed size of the hidden layers are 512, 124, and 32. The output layer consists of 5 nodes that predict the stage of the cancer (or if it is a control/healthy sample) using the pytorch cross entropy function. The classification was assigned as the maximum output. Mimicking training of a cancer subtype classification ANN by Gao et al [5], all weights were initialized using Xavier method.

Current training of the classification network was completed using a 80:10:10 training:validation:testing split however, this may be adjusted to cross-validation testing due to the small dataset. The method is trained using stochastic gradient descent and the momentum value will be tuned for the final dataset (currently it is proposed to be between 0.5 and 0.9). An early stopping method was implemented into model training so training would end once validation error stopped decreasing for a user-specified number of epochs. Evaluation of the model will use metrics of RMSE, sensitivity, and accuracy. A confusion matrix will also be provided.

3.3 Variational autoencoder

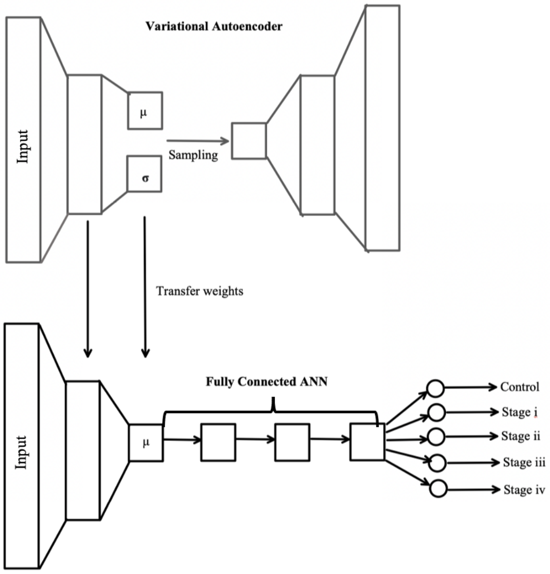


Figure 2: The architecture used for the VAE and transfer learning to classification network are displayed. The VAE is outlined in the top section and is taken from Kim et al [7]. The first hidden and mean layer will be transfered as initial layers to the ANN network and fine-tuned for the LIHC dataset.

3.3.1 Variational autoencoder training

The VAE code provided by Kim et al was used for the pre-training of a variational autoencoder on pan-cancer dataset. The model consisted of an input layer, 2 sequential encoding layers, and 2 sequential decoding layers. Encoding layers used tanh activation and function decoding layers the linear activation function.

The data first had to be formatted into expected table format. This was completed in MATLAB using the log-transformed z-score feature-wise standardized dataset. Gene names were made to represent columns, and sample names (TCGA identification codes) were placed along the rows. This is the expected input format for the model.

Training began with using the exact model presented in the paper. To test efficiency of the model, the model was fine-tuned to a single dataset (the LIHC data set). RMSE was used to evaluate initial model performance. Different learning rates, and dropout rates were tested. The number of nodes in the middle layer was increased to 1000 nodes as both Sarathi and Palaniappan [3] and Lasso results suggest up to 1000 genes may be important int he classification. Two training methods were made availabel in the provided code: SGD and Adam. Kime et al suggested use of Adam, however both were tested and presented in the preliminary results.

Once it was made clear the variational autoencoder could perform well on a single dataset, all datasets were combined into 1 csv file (still to be performed). All datasets were sent through the VAE and once again, tuning of the model was complete. The final model was saved for the next section.

3.3.2 Classification network training

The classification network will be evaluated uwith and without transfer learning from the VAE. This will be used as an evaluation of enhanced model performance and robustness that was claimed by Kim et al [7], however has not been tested for mRNA data. This will confirm the utility of transfer learning from pan-cancer data for mRNA.

For the transfer learning portion, the weights learned from the hidden and final encoding layer of the variational autoencoder will be transferred to the first and second hidden layers of the model as shown in Figure 1 (not yet completed as final tuning has not completed). The same model architecture that has been implemented for Lasso feature classification model will be implemented with the addition of VAE layers. The architecture has already been outlined from the lasso portion. Only LIHC data will be used for training the model and only the samples used by Sarathi and Palaniappan will be used[3]. Evaluation of the model will use metrics of RMSE, sensitivity, and accuracy. A confusion matrix will also be provided.

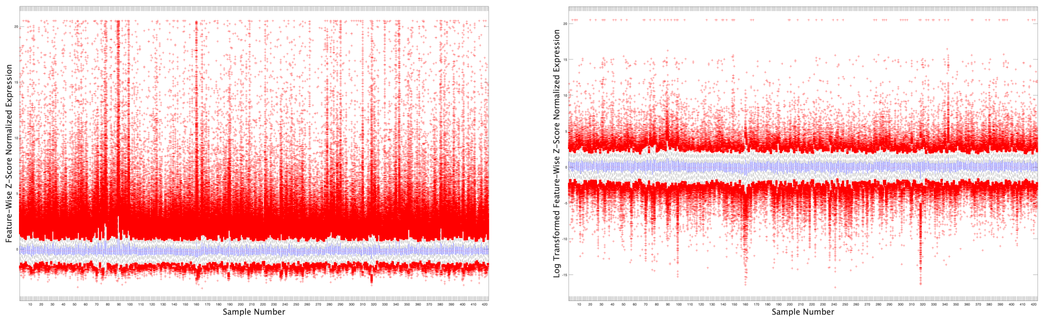
4 Results

Only preliminary results are presented here. Everything will be enhanced for the final paper submission but this is so it is clear what has been completed thus far.

4.1 Data preprocessing

A total of 33 datasets are planned to be used in the data processing. All data samples were used, and missing data values were replaced with 0 (this may be changed depending on final VAE results)

The RNASeq datasets were visualized as can be seen in Figure 3. Investigating normalized and log-transformed normalized data, the log transformed normalized data produced expected genetic results with more symmetric outlier distributions along thr horizontal axis. No batch effects were visualized in the LIHC dataset. Remaining datasets will be visualized in the same way to investigate batch effects. Assuming no batch effects are present, all data samples will be used for all datasets for VAE code, and only samples used by Sarathi and Palaniappan [3] will be used for both classification networks.



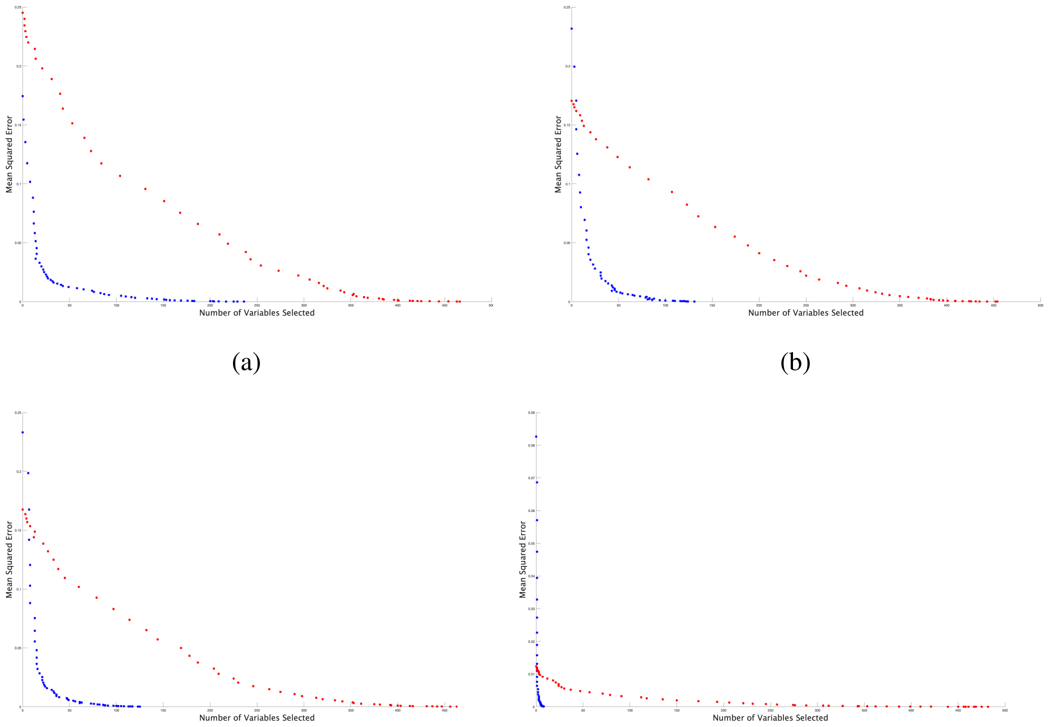
(a)

(b)

Figure 3: a)normalized RNAseq gene expression data. b) log-transformed normalized RNAseq gene expression data.

4.2 Lasso

Figure 3 illustrates that for each stage, less variables are needed to differentiate the stage from control compared to the control and all other cancers. When combining the selected variables for each stage versus control and other stages, a total of 254 variables were selected as important for stage I, 315 for stage II, 275 for stage III and 79 for stage IV. When combined there was limited overlap between the stages and the final subset of genes was 824 genes displaying minimal overlap between the genes.



(a) (b)

(b) (b)

Figure 4: Plots of Lasso results for each stage versus control only (blue) and versus control and all other stages (red) for a) stage I, b) stage II, c) stage III, d) stage IV.

Training of the lasso classifier was tested using a subset of LIHC data. As only the first 1000 genes and 10 samples were used with a patience of 5, the model stopped training after 6 epochs. This was not surprising because the first 1000 genes do not contain information necessary for separating the cancer stages.

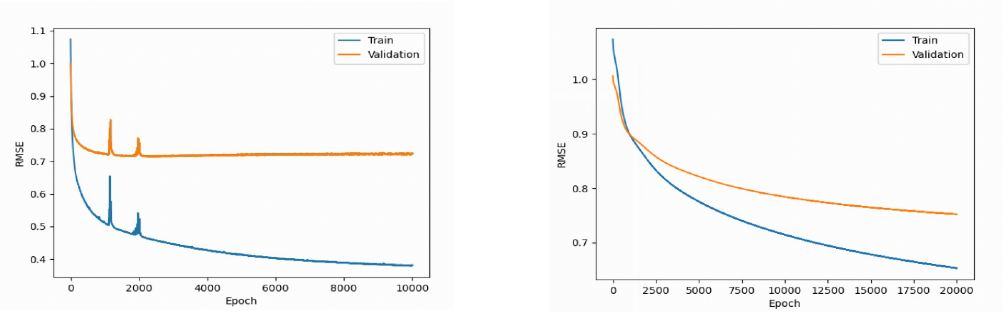
4.3 Variational autoencoder

Currently the VAE has only been run on a single dataset, but in the final paper results for running on the pan-cancer dataset will be presented.

4.3.1 Variational autoencoder training

When using the log-transformed data only the LIHC dataset, in 500 epochs the autoencoder received a training RMSE loss of 0.7275, validation loss of 0.814 and training loss of 0.8278. Considering this was trained with only 423 samples (1 out of the 20 datasets that will be used during true training) and with no hyperparameter tuning, this performed decently well.

After collecting results from Lasso, I discovered that 1000 variables may a more appropriate reduction. As a result I increased the nodes in the mean and standard deviation portions to 1024 (power of 2) and trained the network. The results of using both Adam and SGD optimizer on only a single small dataset are presented in Figure 5. The paper suggested use of Adam optimizerfor miRNA data however, for mRNA data it resulted in spikes in RMSE which is a reported finding of using Adam optimizer with RNAseq data. As a result, I decided to try using the SGD optimizer. The Adam optimizer reached minimum RMSE faster than SGD optimizer however the SGD optimizer resulted in smoother training. Due to its larger learning rate, it required similar training times. To determine the truly ideal model, all datasets will have to be used. The high validation RMSE could be a result of the very small validation data set for the dataset used (about 43 samples). These are preliminary tuning results that display that the model is functioning, however needs the full dataset and further tuning.



(a)

(b)

Figure 5: Training and validation for a)Adam optimizer with learning rate 0.0001 and dropout 0.1 over 10,000 epochs and b) SGD optimizer with learning rate 0.1, momentum 0.5, and dropout 0.1 over 20,000 epochs.

4.3.2 Classification network training

In this section the classification network using transferred weights, and not using transferred weights will be compared using sensitivity, specificity, F-score and accuracy measures. These will also be compared and contrasted amongst the stages.

5 Discussion

- Lasso results in figure 3 illustrated that to differentiate between cancer stages required more variables than to differentiate between each stage and control. Therefore, there is greater complexity differentiating between cancer stages than between cancer and control tissue. This suggests more complex feature selection and analysis methods may pull out some of these more subtle, and complex patterns.

- If either of these networks prove successful at classifying cancer stage, then it suggests genetic analysis is capable of assisting in cancer detection. My suggestion for future work would be to look at including clinical data, and ultimately integrating in a pipeline with pathology image processing mimicking PageNet for survival prediction[6]. PageNet was able to show enhanced classification of pathology images using genetic data, it is an empirical question if this is true for cancer staging - one considering the high variation in pathology staging, I think would be worth pursuing

- Future work should also include larger datasets (although this is challenging from public datasets at this time), to all substages (not collapsed into the parent class) and to other cancers to determine if these complex genetic patterns can be found in the data.

Acknowledgements

Thank you to Randy E. Ellis for his assistance in developing this project proposal and to Christian Muise for his assistance throughout the project.

This project is supported in part by the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-2018-04430 and in part by the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute.

References

[1] Anwanwan D, Singh SK, Singh S, Saikam V, Singh R: Challenges in liver cancer and possible treatment approaches. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer* 1873(1):188314, 2020

[2] Wang Z, Gerstein M, Snyder M: RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics* 10(1):57–63, 2009

[3] Sarathi A, Palaniappan A: Novel significant stage-specific differentially expressed genes in hepatocellular carcinoma. *BMC cancer* 19(1):1–22, 2019

[4] Sun Y, Zhu S, Ma K, Liu W, Yue Y, Hu G, Lu H, Chen W: Identification of 12 cancer types through genome deep learning. *Scientific reports* 9(1):1–9, 2019

[5] Gao F, Wang W, Tan M, Zhu L, Zhang Y, Fessler E, Vermeulen L, Wang X: Deepcc: a novel deep learning-based framework for cancer molecular subtype classification. *Oncogenesis* 8(9):1–12, 2019

[6] Hao J, Kosaraju SC, Tsaku NZ, Song DH, Kang M: Page-net: Interpretable and integrative deep learning for survival analysis using histopathological images and genomic data. *Pacific Symposium on Biocomputing*, volume 25, pp. 355–366. World Scientific, 2020

[7] Kim S, Kim K, Choe J, Lee I, Kang J: Improved survival analysis by learning shared genomic information from pan-cancer data. *Bioinformatics* 36(Supplement_1):i389–i398, 2020

[8] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* 102(43):15545–15550, 2005

[9] Hao J, Kim Y, Mallavarapu T, Oh JH, Kang M: Cox-pasnet: pathway-based sparse deep neural network for survival analysis. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 381–386. IEEE, 2018

[10] Tibshirani R: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288, 1996

[11] Danaee P, Ghaeini R, Hendrix DA: A deep learning approach for cancer detection and relevant gene identification. *Pacific symposium on biocomputing 2017*, pp. 219–229. World Scientific, 2017

[12] Center BITGDA: Analysis-ready standardized tcga data from broad gdac firehose 2016_01_28 run. *Broad Institute of MIT and Harvard* 2016

[13] Li B, Dewey CN: Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics* 12(1):1–16, 2011