Deep Learning in Pan-Cancer Early Detection based on Gene Expression

Chao Li and Meng Zhang
Department of Computer Science
Stanford University
cl5@stanford.edu, meng.zhang@stanford.edu

Abstract

Early detection of cancer significantly increases the chance of successful treatment. Traditional cancer diagnosis relies on physicians' experience to identify morphological abnormalities which firstly do not capture atypical tumors and secondly appear relatively late compared to altered molecular signatures such as sequence variants and gene expression levels. In this project we built a five-layer neural network classifier to predict cancer versus non-cancer based on gene expression data, aiming to detect cancer at the earliest stage. During the process of training the classifier, we used the Integrated Gradients method to identify genes that are most relevant to cancer detection. These identified genes can be further validated to be used as biomarkers and also hold the promise of providing insights into biomedical research and potential therapeutic targets for all types of cancer.

1 Introduction

Thanks to the progress in biomedical research and advances in technology, humans have achieved the power to defeat many previously lethal diseases. However, there is still a long list of life-threatening diseases. Only exceeded by heart disease, cancer is the second leading cause of death, resulting in 8.8 million deaths worldwide every year. Despite recent addition of immunotherapies to the limited cancer treatment options, early diagnosis and accurate prognosis remain the most powerful weapons in the war against cancer as they significantly increase the chance of successful treatment. The traditional diagnoses of cancer rely on human skills to identify abnormal patterns in histo-pathological and radiological evidence. However, there are a few drawbacks in these methods: 1.They are subject to human error, inaccurate, time-consuming, and labor intensive. 2.Inevitable challenges lie in the fact that cancer morphology displays a wide spectrum and many tumors are atypical or lack morphological features that are useful for differential diagnosis [1]. 3.Most importantly, by the time that these abnormal patterns are detectable by humans it is very likely that the cancer may already be at critical stage. Therefore, there have always been calls for alternative approaches.

Molecular signatures hold the promise of precise, objective, and systematic cancer diagnosis and classification. During pathological conditions, the cellular activity is dynamically regulated through changes in gene expression. Therefore, the specific gene expression profiles are valuable signatures that are helpful for early diagnosis because gene expression abnormalities always appear before morphological abnormalities can be observed [2]. Indeed there have been studies that established correlations between gene expressions and early cancer development [3].

Therefore, our goal in this project is to build a classifier to automate the early detection of cancer based on gene expression levels. **The input to our algorithm is the transcriptome profiling data of the patient. We then use a five-layer neural network to output a prediction of cancer/non-cancer.**

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Developments in next generation sequencing in the past decade have revolutionized the field by not only increasing the sequencing depths and accuracy, but also reducing the time and cost to an affordable level for individual cancer patients. Therefore, gene expression profiling has become a feasible option for guiding cancer diagnosis and prognosis. Many efforts have been made in building such models with promising results. For example, Luque-Baena et al. explored six standard and machine learning-based techniques (Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Naïve Bayes (NB), C-MANTEC Constructive Neural Network, K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP)) to identify genes with high cancer predictive capability and biological relevance [4], and Ahmad et al. built a genetic algorithm – artificial neural networks for early diagnosis of breast cancer [5]. Most recently, Salem et al. presented an intelligent decision support system for breast cancer diagnosis using gene expression profiles which outperformed previous models [6].

However, there are still unresolved challenges in these studies: 1. Gene expression profiles contain large numbers of genes but all of the previous studies only had access to relatively small amount of samples ranging from less than a hundred to a few hundred. Although they utilized dimensionality reduction techniques, the limitation in samples numbers was still an obstacle to the generality of their prediction models. 2. The different genes that are relevant to cancer diagnosis and prognosis are usually correlated and have complex interactions that may affect the application of conventional machine learning classifiers [7].

The developments of deep neural network (DNN) in recent years aim to resolve these challenges. DNN is a type of neural network that models abstracted high-level data features using multiple non-linear and complex processing layers, and provides feedback for updating the parameters and insight on interpretation of the input features via back-propagation [8]. Nowadays DNN is widely applied in image classification, object detection, text processing and many other fields. And DNN offers the novel opportunities for gene expression profiling data where it perfectly fits the need for high dimensional data processing and capturing gene-gene interactions.

On the other hand, with the developments of DNA sequencing technology, large amounts of sequencing data from cancer patients have become available and provide unprecedented opportunities for building DNN models for cancer diagnosis and prognosis. Specifically, the Cancer Genome Atlas (TCGA) have collected gene expression profiles from thousands of cancer patients and made the data publicly available together with the patients' clinical information.

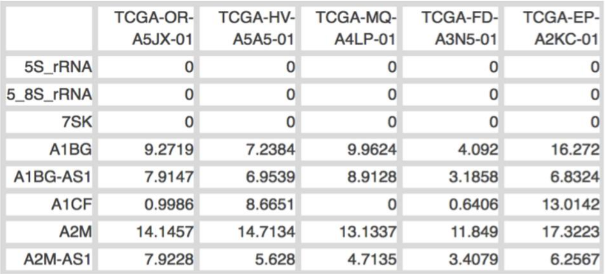
A couple groups took advantages of the TCGA data in cancer diagnosis, including Danaee et al. from Oregon State University who took a deep learning approach for breast cancer detection using TCGA gene expression profiles [9], and Nikhil Cheerla and Olivier Gevaert at Stanford who created a pan-cancer diagnosis system based on TCGA microRNA expression profiles [10]. However, no such DNN models have been established for pan-cancer diagnosis making use of the 10,000 gene expression profiling data available on TCGA.

Our project, for the first time, adapt the deep learning technique to predict diagnosis for pan-cancer based on gene expression profiling and identified relevant genes in prediction of all types of cancer.

3 Dataset and Features

The pan-cancer transcriptome profiling data were downloaded from the University of California Santa Cruz Xena data center [11], which were processed from the raw data on TCGA. The gene expression profiles were measured experimentally using the Illumina HiSeq 2000 RNA Sequencing platform by the University of North Carolina TCGA genome characterization center. The datasets show the gene-level transcription estimates, as in log2 (x+1) transformed RSEM (RNA-seq by Expectation Maximization) normalized count [13]. Genes are mapped onto the human genome coordinates using UCSC Xena HUGO probeMap. The dataset includes pan-cancer patients (n = 9,807) and normal samples (n = 856) and covers 58,582 transcripts.

The biggest challenge in this project is the high dimensionality of the gene expression data and the relatively small amount of samples. Therefore, we took two approaches for dimensionality reduction —- prior knowledge and auto-encoder (details described in the Methods section).



	TCGA-OR-A5JX-01	TCGA-HV-A5A5-01	TCGA-MQ-A4LP-01	TCGA-FD-A3N5-01	TCGA-EP-A2KC-01
5S_rRNA	0	0	0	0	0
5_8S_rRNA	0	0	0	0	0
7SK	0	0	0	0	0
A1BG	9.2719	7.2384	9.9624	4.092	16.272
A1BG-AS1	7.9147	6.9539	8.9128	3.1858	6.8324
A1CF	0.9986	8.6651	0	0.6406	13.0142
A2M	14.1457	14.7134	13.1337	11.849	17.3223
A2M-AS1	7.9228	5.628	4.7135	3.4079	6.2567

Figure 1: A sample look of the TCGA transcriptome profiling data. Row: sample ID (01-09: cancer, 11: non-cancer), Column: transcript ID. (58,582 transcript IDs X 10,663 samples, 37 different types of cancers)

4 Methods

First of all, we built our models with total data without feature reduction to check their performances. Since our cancer/non-cancer prediction is a binary classification task, we firstly built a logistic regression model, a support vector machine, and a random forest as the baseline models. The transcriptome profiling data was split into training set and test set using 7-fold cross-validation. Since the data is very imbalanced (about 90% cancer samples and 10% non-cancer samples), accuracy is not an optimal metric to evaluate model performance. Instead we used F1 score which combines precision and recall to represent the model performance.

After building the baseline models, we split the data into 90% training set and 10% test set and built neural networks with various layers. Since our data contains far more cancer samples than healthy samples, we used a weighted cross entropy cost function to balance precision and recall:

$$J = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}log(a^{(i)})pos\_weight + (1-y^{(i)})log(1-a^{(i)}))$$

We used the most common RELU activation function in our hidden layers of the neural network, and sigmoid in our output layer since this is a binary classification problem. We used Adam Optimizer to minimize the cost function in training the model.

After building out the initial architecture, we introduced feature reduction approaches to see if they can further improve our model. We tried two methods – prior knowledge and auto-encoder.

1. Prior knowledge: Out of the 58,582 transcripts, most of them might be irrelevant to the task of diagnosing cancer. Besides, since this study uses pan-cancer data, We would focus on the shared features among different types of cancers. Based on prior knowledge, genes that are involved in cell cycle, cell death/apoptosis, and cell adhesion are more likely to be relevant. Thus we used the previously identified gene sets to select transcripts that are included in relevant biological pathways.

2. The autoencoder is a type of neural networks that helps extract and compose robust features from the input data [14]. An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. It is trying to learn an approximation to the identity function, so as to output $\hat{x}$ that is similar to x. The dimensionality reduction is achieved by limiting the number of hidden units. The autoencoder we used contains four hidden layers (relu -> relu -> linear -> relu -> sigmoid) with the latent space having 512 nodes. And we used RMSE (root mean square error) loss function:

$$L = \|\hat{x} - x\|^2$$

Finally we used the Integrated Gradients method to study input attributions [15]. Integrated Gradients is a method for attributing a neural networks' prediction to its input features by examining the gradients of inputs obtained by interpolating on a straight-line path between the input and a baseline input, and then aggregate these gradients together following the below equation:

$$IntegratedGrads_i^{approx}(x) ::= (x_i - x'_i) \times \sum_{1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$$

Here m is the number of steps in the Riemman approximation of the integral.

5 Results and Discussion

We tested multiple neural network architectures: three layers (1000 neurons in the first hidden layer and 500 neurons in the second hidden layer), five layers (1000 , 500, 500, 250 respectively in the four hidden layers), and nine layers (20, 10, 15, 12, 8, 6, 4 neurons in the seven hidden layers respectively). We also built deeper neural networks with more than 10 layers. However, as we grew the number of layers, the results did not show significant improvement and were mediocre at best. This suggested that a medium size neural network was the optimal architecture in solving this particular problem.

When building the deep neural network model, we primarily focused on tuning three hyperparameters: learning rate, number of epochs, and pos_weight. For learning rate, we wanted to find an appropriate rate that can both converge to minimum fast and also do not miss the target. We experimented with rate from 0.000001 to 0.1, and found 0.00001 to yield the best results (Figure 2A). For number of epochs, we observed that the cost decreasing rate flattened out around 500, and thus decided to pick 500 as our number of epochs in training the model (Figure 2B). Lastly, in our data set, we labeled 0 as cancer and 1 as healthy samples. If the model predict 0 for all samples, it would already achieve 90% accuracy. Thus, we introduced pos_weight in our cost function, and set it greater than 1 to penalize false negative count. We experimented with pos_weight values ranging from 2 to 20, and found 2 to yield the best result in balancing precision and recall (Figure 2C).

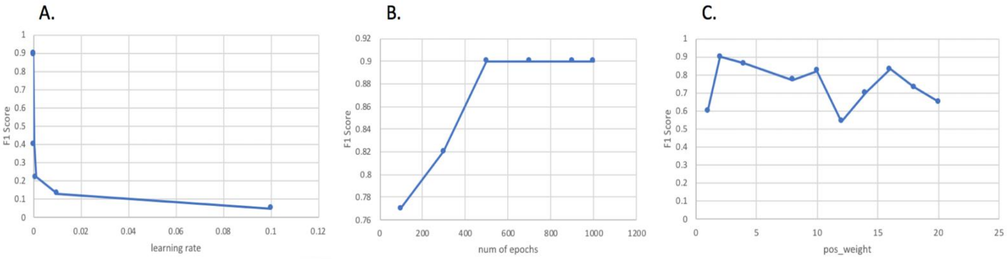
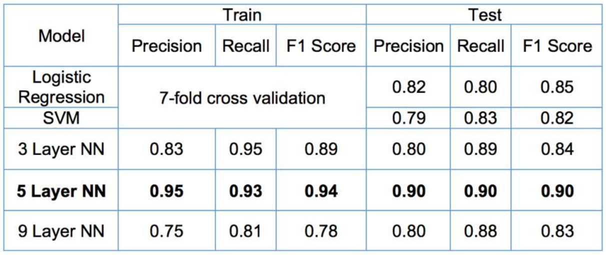


Figure 2: Hyperparameters vs. F1 Score

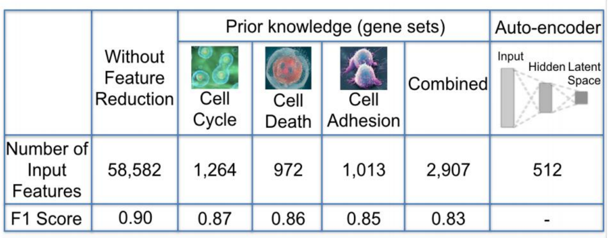
After comparing several neural network architectures and optimizing the hyperparameters, we found the model fitted the training set well, but did poorly on the test set. This was because our data has way more features than samples. Thus, we incorporated L1 and L2 regularization with different scales into the model. We found L2 regularization to do a better job than L1, and was able to reduce overfitting effectively. As shown in Figure 3, 5 layer neural network generated the best results. It yielded 90% F1 Score and beat all the baseline models.



Model	Train			Test		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Logistic Regression	7-fold cross validation			0.82	0.80	0.85
SVM				0.79	0.83	0.82
3 Layer NN	0.83	0.95	0.89	0.80	0.89	0.84
5 Layer NN	**0.95**	**0.93**	**0.94**	**0.90**	**0.90**	**0.90**
9 Layer NN	0.75	0.81	0.78	0.80	0.88	0.83

Figure 3: Results from different baseline models and different neural network architectures

For dimensionality reduction, we experimented both prior knowledge and auto-encoder. Research suggested that genes in cell cycle, cell death, and cell adhesion are among the most relevant in cancer prediction. Thus we handpicked 1,264 cell cycle genes, 972 cell death genes, and 1,013 cell adhesion genes, and used these subsets to feed into our model. As shown in Figure 4, reducing the feature space down to these subsets actually achieved decent prediction results (around 85% F1 score), but they were not able to beat our model without feature reduction. However, our attempt to employ autoencoder yielded a much worse performance compared to no feature reduction, suggesting that future investigations and more thorough search for autoencoder architectures are needed.



	Without Feature Reduction	Prior knowledge (gene sets)				Auto-encoder
		Cell Cycle	Cell Death	Cell Adhesion	Combined	Input Hidden Latent Space
Number of Input Features	58,582	1,264	972	1,013	2,907	512
F1 Score	0.90	0.87	0.86	0.85	0.83	-

Figure 4: Feature Reduction Methods

With a decent neural network model in hand, we were curious to see what genes were the most important indicators of predicting pan-cancer. We used the Integrated Gradients method to achieve this purpose. The resulting attributions tell us which genes are most responsible for predicting cancer and we used them for gene ontology analysis to identify pathways they are involved in. As shown in figure 5, in addition to the commonly known gene sets related to cell cycle and cell adhesion, other pathways stood out to be important: mRNA catabolic process, immune response, fatty acid oxidation, to name a few. This possibly explains why our feature reduction using prior knowledge did not yield superb results, as these genes were otherwise overlooked by prior knowledge.

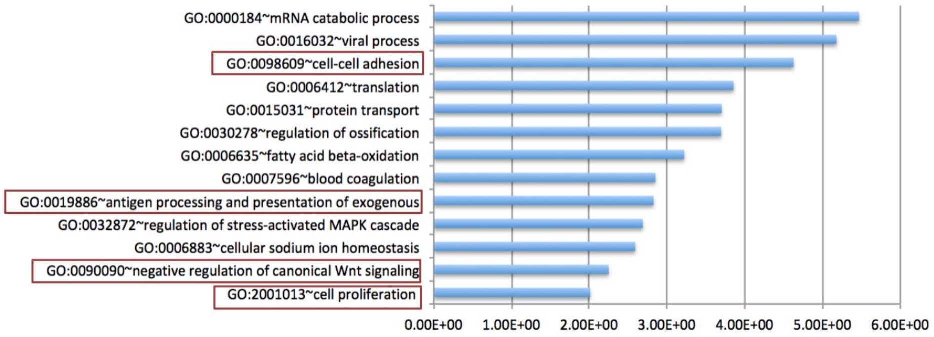


Figure 5: Integrated Gradients

6 Conclusion and Future Work

Deep learning has enabled us to predict cancer precisely using transcriptome profiling data. Instead of relying solely on imaging and human expertise, our model can help physicians detect cancer at a much earlier stage and thus greatly improve the potential of patient treatment. In particular, our neural network model with 5 layers was able to achieve a prediction F1 score of 90%. Neural network has the power to capture gene-gene interaction that are usually not captured in traditional machine learning models and thus renders satisfying results.

For future work, we plan to make use of transfer learning to build another two classifiers for cancer staging and survival prediction. Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on another task. Since the tasks for classifying cancer stages and predicting short-time survival are similar to the task for classifying cancer/non-cancer, we propose to reuse the trained model as the starting point for DNNs for pan-cancer staging and survival prediction. The clinical data containing staging and survival information will be collected from TCGA for each type of cancer individually and the input will still be the transcriptome profiling data after dimensionality reduction. Since cancer staging is a multi-class classification task we will replace the output sigmoid layer by a softmax regression layer.

After tuning the two new DNNs, we will test them on the held back test data and validate them using external data sets obtained from multiple studies deposited at the University of California Santa Cruz public hub. Once these two DNNs are successfully developed, they should be able to aim in precise staging and survival predication at diagnosis time, which can help physicians make better choices of treatment for cancer patients.

Code

```
https://github.com/cl9bs/CS230-Cancer-Genomics-Project/blob/master/CS230%
20project%20final.py
```

Contributions

We worked collaboratively on this project. Each team member has equally participated in all aspects of the projects, including literature research, data collection, data pre-processing, model building, and report writing.

References

[1] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., & Golub, T. R. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences* 98(26), 15149-15154. doi: 10.1073/pnas.211566398

[2] Okamoto, O. (2005) *DNA microarrays in cancer diagnosis and prognosis.* (Vol. 3).

[3] Ling, B., Chen, L., Liu, Q., & Yang, J. (2014) Gene Expression Correlation for Cancer Diagnosis: A Pilot Study. *BioMed Research International* 2014, 6. doi: 10.1155/2014/253804

[4] Luque-Baena, R. M., Urda, D., Subirats, J. L., Franco, L., & Jerez, J. M. (2014) Application of genetic algorithms and constructive neural networks for the analysis of microarray cancer data. *Theoretical Biology and Medical Modelling* 11(1), S7. doi: 10.1186/1742-4682-11-s1-s7

[5] Ahmad, F., Isa, N. A. M., Noor, M. H. M., & Hussain, Z. (2013, 5-7 June 2013). Intelligent Breast Cancer Diagnosis Using Hybrid GA-ANN. *Paper presented at the 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks*

[6] Salem, H., Attiya, G., & El-Fishawy, N. (2017). Early diagnosis of breast cancer by gene expression profiles. *Pattern Analysis and Applications* 20(2), 567-578. doi: 10.1007/s10044-016-0574-7

[7] Yuan, Y., Shi, Y., Li, C., Kim, J., Cai, W., Han, Z., & Feng, D. D. (2016). DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations. *BMC Bioinformatics* 17(17), 476. doi: 10.1186/s12859-016-1334-9

[8] Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing* 7(3–4), 197-387. doi: 10.1561/2000000039

[9]Danaee, P., Ghaeini, R., & Hendrix, D. A. (2016). A Deep Learning Approach for Cancer Detection and Relevant Gene Identification *Biocomputing* 2017 (pp. 219-229): WORLD SCIENTIFIC.

[10] Cheerla, N., & Gevaert, O. (2017) MicroRNA based Pan-Cancer Diagnosis and Treatment Recommendation. *BMC Bioinformatics* 18, 32. doi: 10.1186/s12859-016-1421-y

[11] UCSC Xena data center: https://xenabrowser.net/datapages/

[12] TCGA database: https://cancergenome.nih.gov/

[13] Li B, &Dewey CN. (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics.* 2011;12:323. doi:10.1186/1471-2105-12-323.

[14] Baldi, P. (2012, June). Autoencoders, unsupervised learning, and deep architectures. *In Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (pp. 37-49).

[15] Sundararajan, M., Taly, A., & Yan, Q. (2017). *Axiomatic attribution for deep networks. arXiv preprint arXiv*:1703.01365.

[16] Travis E, Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).

[17] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55.

[18] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010).

[19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David

Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011).

[20] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2018-03-23].

[21] Collet Francois and others. Keras. 2015. Github. https://github.com/keras-team/keras.

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.