Designing and Evaluating Deep Learning Methods for Cancer Classification on Gene Expression Data

Arif Canakoglu, Luca Nanni, Artur Sokolovsky, Stefano Ceri

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy
first.last@polimi.it

Keywords: Deep learning, Feature engineering, Gene expression.

Abstract. Gene expression levels, measuring the transcription activity, are widely used to predict abnormal gene activities, in particular for distinguishing between normal and tumor cells. This problem has been addressed by a variety of machine learning methods; more recently, this problem has been approached using deep learning methods, but they typically failed in meeting the same performance as machine learning. In this paper, we show specific deep learning methods that can achieve similar performance as the best machine learning methods.

1 Introduction

Cancer classification using gene expression data has been addressed by a variety of conventional machine learning methods [1, 2]; the binary classification problem (determining if a tissue is from a normal or cancer cell) is typically solved with high accuracy. More recently, various attempts have been made of using deep learning methods; three main challenges in the training and testing of deep learning models for expression are (a) the lack of training samples (b) the unbalanced populations of the different classes and (c) high dimensionality of the problem. The first challenge could possibly be tackled by a suitable generation of synthetic data; however, such practice is not easy in the case of gene expression. The second challenge can be approached by sampling techniques but this is not recommended in the general lack of training data. Finally, the huge dimensions of the search space can be dealt with some pre-filtering technique on the genes [3], but this might cause the omission of relevant information.

For addressing these challenges, in this work we describe three innovative deep learning methods and compare them to baseline methods; our attempt is to match the quality of classical machine learning for binary classification, in the 99% accuracy range, thereby also testing their applicability to more difficult cancer classification problems. Innovation addresses two orthogonal perspectives: the data dimension and the provisioning of additional information. Along the first perspective, we can use *feature engineering* to simplify the training task or *data augmentation* to increase the information used for training. Along the second one, we can use external information for training, either in the form of *biological knowledge* or of *other compatible datasets*.

Table 1: Four neural network methods described in this work

	No information	**External information**
Feature engineering	*Feed Forward Network*	*Ontology-Guided CNN*
Data augmentation	*Ladder Network*	*Transfer Learning*

We will next describe the datasets used for gene expression, then apply the baseline models in order to classify normal and tumor cells, then present each of three interesting cases of baseline augmentations (Table 1), and finally, we will compare their performance. All the considered models take as input gene expressions and produce as output the normal/tumor labels for three datasets corresponding to specific tumors.

2 Materials and Methods

2.1 Datasets

We used RNA-seq data from the TCGA public dataset [4]: we downloaded Illumina HiSeq 2000 log2 scaled data matrix from the Xena Browser[1] in December 2017. Due to the lack of samples and the imbalance between normal and tumor samples in most cancer types, we considered the three most represented cancer types by origin tissue, which are: 1. **Breast**: Breast invasive carcinoma (BRCA); 2. **Lung**: Lung adenocarcinoma (LUAD) and Lung squamous cell carcinoma (LUSC); 3. **Kidney**: Kidney Chromophobe (KICH), Kidney renal clear cell carcinoma (KIRC) and Kidney renal papillary cell carcinoma (KIRP). We also performed a principal components analysis (PCA) on the three selected datasets (Fig. 1) displaying the first two components and the relationship with the sample label (normal or tumor).

Table 2: TCGA sample counts for each tissue

Tissue	Total	Normal	Tumor	Normal (%)
Breast	1218	114	1104	9.36%
Kidney	1020	129	891	12.65%
Lung	1129	110	1019	9.74%

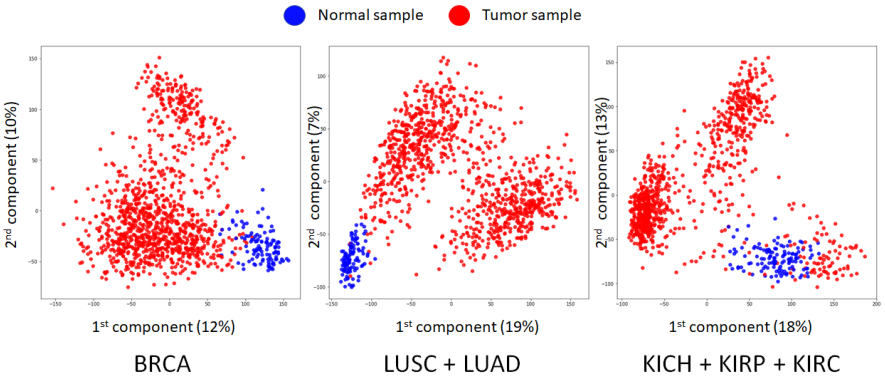

Figure 1: First two components of Principal Component Analysis (with their percentage of explained variance) of the three cohort datasets considered in this study.

2.2 Baselines

As baseline for classic machine learning, we trained a *Support Vector Machine with linear kernel (SVM)*, known as a high performance model for this task [1]. We also evaluated a classical *5-nearest neighbors classifier* and a *10-tree random forest*, which were used for clinical outcome prediction with gene expression data.

As a baseline for deep learning, we trained a classic *Feed Forward Network* (FFN). As FFN is used as baseline, the network architecture is as simple as possible: for each considered dataset, we selected the top 5000 most variant genes and performed a Min-Max normalization of the expression values. The architecture is composed by 2 hidden layers that contain 100 and 20 neurons respectively, with the ReLu activation function. We used binary cross entropy as loss function to be minimized:

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1)$$

We anticipate from the discussion that the performance of machine learning baselines is generally superior to FFN, used as deep learning baseline. We study new deep learning models with the objective of improving over the relative baseline and achieve comparable results with the machine learning models.

[1] http://xena.ucsc.edu

2.3 Ladder Network

A ladder network [5] is a semi-supervised method which contains both supervised and unsupervised parts in a single deep neural network; training of both parts is simultaneous, without using layer-wise pre-training. The network has three passes, two feed forward passes and one decoding denoising pass that connects the two forward passes. The supervised part is a feed forward network, whose weights are normalized by the unsupervised encoder weights with denoising cost for each layer. The unsupervised part, which contains the decoding and the corrupted feed forward passes, works as Denoising Auto Encoder (DAE), in which the Gaussian noise is added to each hidden layer. The loss function of the whole network is a weighted sum of the supervised and unsupervised paths. The former is cross entropy cost on top of the standard forward pass and the latter is sum of denoising square error costs of each layer of the decoder.

We tuned the network by using different parameters, the most relevant ones are the number of layers (single layer or 2, 3, 5, 7 and 10 hidden layers) and the training feed size (10, 20, 30, 40, 60, 80 and 120 labeled data) as in [6]. We selected 5 hidden layer with 5000, 1500, 500, 250, 10 neuron sizes in the forward paths and inverse sizes on the decoding path and we feed 120 labeled data which contains equals number element from each class to the supervised path; all samples are feeding the unsupervised path.

2.4 Ontology-driven Convolutional Neural Network

Convolutional Neural Networks (CNN) exploit the spatial relationships between the input features to derive high-level representations which are then provided as inputs of a classical feed-forward network (FFN). The convolutional layer uses a set of kernels which transform the input features by aggregation, locally applied to near features.

The concept of neighborhood in the context of biology can be applied to an extremely vast set of entities, which are *near* if they share a common behavior, or present a similar pattern, or are semantically correlated. We enriched the prediction capability of a CNN by guiding the convolution layers using distance relations between genes derived from prior biological knowledge.

A distance matrix $\mathbf{D}$ is a symmetric matrix in which the diagonal elements are zero and the other non-negative elements contain the distance d_{xy} between genes g_x and g_y; the k-neighbours of a gene g_x are the genes g_k for which d_{xk} is in the set of the k lowest distances between x and all the genes with $i \neq k$. In order to compute distances, the Genomic and Proteomic Data Warehouse (GPDW) [7] is used, which contains gene datasets, their annotations from the Gene Ontology [8] (GO: cellular component, molecular function and biological process), and also the relationships between them. The minimum distance information extracted from GPDW for each gene pair $< g_x, g_y >$ is:

$$d\left(g_x, g_y\right) = \begin{cases} \min\limits_{o_i \in \Omega_x, o_j \in \Omega_y} \text{hops}\left(o_i, o_i\right) & \text{if } x \neq y \wedge \Omega_x \cap \Omega_y \neq \emptyset \\ 0 & \text{if } x = y \\ +\infty & \text{otherwise} \end{cases}$$

where Ω_x, Ω_y are the sets of ontological terms annotated respectively to gene g_x and g_y and hops is a function giving the minimum distance between them calculated as the sum of the number of the edges towards a common ancestor. We then used this distance measure to derive the sets of genes on which apply the convolution. For each gene, we derived the nearest 4 genes and applied a 1-dimensional convolutional filter to each set of 5 genes using a stride of 5. In this way, we derive an aggregated representation of the neighborhoods of genes, which we then input to a FFN having 2 hidden layers of 200 and 50 neurons (characterized all by a ReLu activation function). A similar technique was applied to the classification of inflammatory bowel disease (IBD) using the phylogenetic tree of bacteria for the hierarchical characterization of the features [9].

2.5 Transfer Learning using a Combined set of Tumors

Transfer learning is a classic machine learning method where a general model developed for a task is reused as the starting point for a model on a second, target task; this approach is applicable to our problem due to the general scarcity of training datasets. However, the model design has to be careful: when the two datasets do not have a strong affinity or the target dataset has strong linearity properties with respect to the label, then the hybrid model performance will be degraded.

We apply transfer learning to our problem as follows. If we want to detect the presence of cancer for tumor t, we first train a general network model M_g on all the tumor types except t. We then train a hybrid model M_h composed by the concatenation of M_g and a small neural network M_s, where the weights of M_g are kept unchanged during the training. One can see this procedure as a function composition $M_h = M_g \circ M_s$ where the various models are functions $M(x) = \tilde{x}$ which extract relevant features from the input vector except for the final model, which produces as output the class probability $y = \tilde{x} = M(x)$.

In our case, the general model is constituted by a FFN with four fully connected hidden layers respectively with 500, 200, 100 and 50 neurons. We use the ReLu activation function between all the layers except the output layer, which applies a Sigmoid function. The small network used for the fine tuning of the hybrid network on the specific tumor type has two hidden layers of 50 an 10 neurons. Like in the FFN case, we pre-filtered the genes used for training and testing. taking only the top 5000 variants.

3 Results

All the methods were evaluated using a 5 times repeated 5-fold cross validation strategy and taking the mean of the various metrics as aggregated performance score. For each fold we computed accuracy, F_1 score, precision and recall. Each fold splits the data in training and test set and a portion of the training data (25%) is taken as validation set. All the models were programmed in Python using the Keras library.

Table 3: Performance measures for the four baseline models.

Linear SVM	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.994913	0.973241	0.964699	0.982609
LUAD+LUSC	0.996986	0.984847	0.9739	0.996364
KIRC+KIRP+KICH	0.996668	0.986865	0.988303	0.986031

KNN	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.991622	0.957749	0.921784	0.998261
LUAD+LUSC	0.991849	0.960519	0.925095	1
KIRC+KIRP+KICH	0.996866	0.987742	0.983696	0.992185

Random Forest	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.987843	0.932008	0.965448	0.903083
LUAD+LUSC	0.992561	0.960789	0.972691	0.950909
KIRC+KIRP+KICH	0.992943	0.970947	0.99202	0.952062

Feed Forward Network	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.988511	0.945621	0.910416	0.991304
LUAD+LUSC	0.995039	0.97576	0.957081	0.996364
KIRC+KIRP+KICH	0.996079	0.984266	0.989459	0.979815

Table 3 compares the baselines. A linear model like SVM achieves high accuracy in

all the cohorts that we selected for the study, thanks to an intrinsic linear separability property of the cohorts that we selected, confirmed by PCA analysis (Fig. 1). KNN and Random Forest show lower performance metrics with exception for recall; this is principally due to the shape of the clusters that the two classes form in the datasets, which enables a neighbor-based approach like KNN to achieve better recognition capability for the normal samples. Random Forest shows lower recall but better precision than KNN; prediction accuracy is lower than all the other baselines. FFN achieves in general worse performance than the linear SVM, including a worse precision. This is expected because of difficulty of training the model with a small amount of samples; on the other hand, it is remarkable that such a simple architecture and pre-filtering of genes can achieve comparable results with state-of-art machine learning methods.

Table 4: Performance measures for three deep learning models.

Ladder Network	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.988998	0.944120	0.899841	0.992982
LUAD+LUSC	0.992737	0.964003	0.932088	0.998182
KIRC+KIRP+KICH	0.995294	0.981595	0.971168	0.992248

Ontology-Guided CNN	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.993436	0.965749	0.950609	0.982609
LUAD+LUSC	0.99292	0.965749	0.935173	1
KIRC+KIRP+KICH	0.998039	0.992305	0.992593	0.992308

Transfer Learning	**accuracy**	**F_1 score**	**precision**	**recall**
BRCA	0.990963	0.953823	0.920312	0.990909
LUAD+LUSC	0.993974	0.970087	0.948913	0.992727
KIRC+KIRP+KICH	0.995683	0.982932	0.982227	0.984431

Table 5: Transfer learning results

BLCA Dataset	**accuracy**	**F_1 score**	**precision**	**recall**
Linear SVM	0.992016	0.899937	0.948667	0.876667
Feed Forward Network	0.991069	0.872635	0.914000	0.856667
Transfer Learning	0.989668	0.897123	0.871524	0.950000

Table 4 compares our three proposed methods. Ladder Network achieves better recall than both linear SVM and FFN, but does not manage to compete at the level of accuracy. Note that this model does not use any gene pre-filtering, as it extracts the relevant inner features from the *whole* set of genes, characterizing itself as a good knowledge extraction method for gene expression. Ladder network has equal number of samples for the supervised learning path, this leads to better recall and lightly worse precision.

The ontology-guided CNN approach achieves better performance than FFN in all the metrics with the exception of the LUNG dataset. In addition it is able to overcome machine learning methods in the Kidney dataset and almost same performance in the other tissues, while it is always superior in recall. We checked if the performance improvement could be caused by the convolutional layer by itself by testing the classification accuracy with a randomly generated distance matrix, but we achieved worse performance. Therefore, we conclude that the GO ontological network is able to guide the convolution to a better inner representation of the features.

The transfer learning approach provides accuracy results comparable with FFN; on the BRCA datasets, it improves in all the measures w.r.t. the machine learning baselines,

including linear SVM. We also applied transfer learning to a different cancer type, **Bladder Urothelial Carcinoma** (BLCA), for which very few normal datasets are available. In Table 5, we show better performance with respect to the FFN baseline for what concerns recall (10% improvement) and F_1 score; transfer learning proved to be a valid tool when applied to cancer classification in the case of severe lack of samples.

4 Conclusions

To the best of our knowledge, this work presents the first systematic evaluation of machine and deep learning methods for the cancer classification problem on gene expression data, improving earlier work by [10]. We generally demonstrated that three deep learning architectures can compete with machine learning methods even in the presence of few training samples, population unbalancing and high problem dimensionality. In our future work, we will also approach the *stage classification problem* – cancer stages are provided in TCGA for some cancer types and can also be predicted. In our preliminary results, deep learning models actually prove superior to machine learning ones for this specific problem.

Availability The source of all methods are available in `https://github.com/DEIB-GECO/CIBB2018`.

Acknowledgment. This work is supported by the ERC Advanced Grant 693174 *GeCo* (Data-Driven Genomic Computing) .

References

[1] T. S. Furey *et al.*, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[2] A. Statnikov *et al.*, "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification," *BMC Bioinformatics*, vol. 9, no. 1, p. 319, 2008.

[3] I. Guyon *et al.*, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, Jan 2002.

[4] J. N. Weinstein *et al.*, "The cancer genome atlas pan-cancer analysis project," *Nature Genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.

[5] A. Rasmus *et al.*, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

[6] G. Golcuk *et al.*, "Exploiting ladder networks for gene expression classification," in *International Conference on Bioinformatics and Biomedical Engineering*, pp. 270–278, Springer, 2018.

[7] A. Canakoglu *et al.*, "Integrative warehousing of biomolecular information to support complex multi-topic queries for biomedical knowledge discovery," in *BIBE*, pp. 1–4, IEEE, 2013.

[8] M. Ashburner *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, p. 25, may 2000.

[9] D. Fioravanti *et al.*, "Phylogenetic convolutional neural networks in metagenomics," *BMC Bioinformatics*, vol. 19, p. 49, Mar 2018.

[10] S. Agrawal and J. Agrawal, "Neural network techniques for cancer prediction: A survey," *Procedia Computer Science*, vol. 60, pp. 769–774, 2015.