Designing and Evaluating Deep Learning Models for Cancer Detection on Gene Expression Data

Arif Canakoglu(✉), Luca Nanni(✉), Artur Sokolovsky, and Stefano Ceri

Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, Milan, Italy
{arif.canakoglu,luca.nanni,artur.sokolovsky,stefano.ceri}@polimi.it

Abstract. Transcription profiling enables researchers to understand the activity of the genes in various experimental conditions; in human genomics, abnormal gene expression is typically correlated with clinical conditions. An important application is the detection of genes which are most involved in the development of tumors, by contrasting normal and tumor cells of the same patient. Several statistical and machine learning techniques have been applied to cancer detection; more recently, deep learning methods have been attempted, but they have typically failed in meeting the same performance as classical algorithms. In this paper, we design a set of deep learning methods that can achieve similar performance as the best machine learning methods thanks to the use of external information or of data augmentation; we demonstrate this result by comparing the performance of new methods against several baselines.

Keywords: Deep learning · RNA-seq · Machine learning · Cancer detection

1 Introduction

Next Generation Sequencing technologies enabled in the last years the creation of constantly growing whole-transcriptome datasets, allowing the researchers to understand the underlying mechanisms of gene expression and their relationship with sample phenotype. Large gene expression databases like The Cancer Genome Atlas [20] or the Genotype-Tissue Expression project [13] became the basis on which a lot of computational works have defined novel methodologies to extract information from gene expression datasets.

Machine Learning (ML) is the study of statistical techniques which enable computers to extract relevant patterns from data and use this information to solve specific tasks without the need of manually specify the set of instructions. Deep Learning (DL) is a specific form of ML in which algorithms are trained to

A. Canakoglu and L. Nanni—Co-primary authors.

M. Raposo et al. (Eds.): CIBB 2018, LNBI 11925, pp. 249–261, 2020.
https://doi.org/10.1007/978-3-030-34585-3_22

learn an increasingly abstract hierarchy of *feature representations* of the original data, where the first layers represent *low-level* (i.e., concrete) features and last layers represent *high-level* (i.e., abstract) features.

DL models have the ability to learn expressive representations of the data without the need of specific pre-processing steps [11]. After training, the learned data representation is usually able to embed very complex non-linear relationships between the sample features, which can then be used to perform the classification task. The power of this approach has been widely demonstrated in many fields like computer vision, speech recognition, and natural language processing. Most of DL models achieve this by relying on Neural Network architectures [16], which usually require more parameters to be trained than classic ML algorithms. This reflects in an increased need of training data, and for this reason, only very recently researchers have started to apply DL architectures to biological tasks.

Three main challenges in the training and testing of DL models with gene expression data are (a) the lack of training samples, (b) the unbalanced populations of the different classes, and (c) high dimensionality of the problem. In this paper, we show a set of specific deep learning methods that, thanks to their ability to adapt to the problem, can achieve similar performances as the best machine learning methods.

2 Scientific Background

Gene expression levels, measuring the transcription activity, are widely used to predict abnormal gene activities, in particular for distinguishing between healthy and tumor cells [22]. This is a classical problem that has been addressed by a variety of ML methods, e.g., see [6,10,18]. There is a long story of learning algorithms applied to gene expression datasets: relevant applications count identifying gene expression signatures specific to a cancer type or sub-type [17,23], differentiating between tumor characteristics like grade or stage [14] and predicting clinical outcomes [21].

More recently, new artificial intelligence methods have emerged, and some attempts have been made of using DL methods for cancer detection and classification [5,12].

As discussed above, the three main challenges using DL models with gene expression data are (a) the lack of training samples, as the open datasets available in The Cancer Genome Atlas [20] are distributed across 33 tumor types, resulting in very few samples for most of them (b) the unbalanced populations of the different classes, as very few normal cells are available in comparison with cancer cells, and (c) high dimensionality of the problem, as the number of features (coding and noncoding genes) may be of the order of 100 K depending on the technology used.

The first challenge could possibly be alleviated by a suitable generation of synthetic data; however, such practice is not easy in the case of gene expression [3]. 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 [8], but this might cause the omission of relevant information.

To address these challenges, in this work we describe three innovative DL methods and compare them to standard ML methods and a baseline DL method; 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* to improve model training.

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.

3 Materials and Methods

3.1 Datasets

We used RNA-seq data from the TCGA public dataset [20]: 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). In Table 2, we reported the number of available data and 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).

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

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%

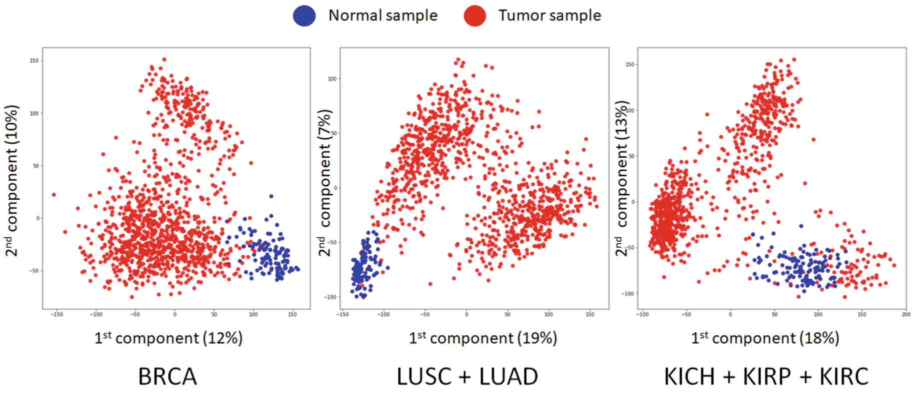


Fig. 1. First two components of Principal Component Analysis (with their percentage of explained variance) of the three cohort datasets considered in this study.

3.2 Baselines

As baseline for classical machine learning, we trained a *Support Vector Machine with linear kernel (SVM)*, known as a high-performance model for this task [6]. We also evaluated a classical *5-nearest neighbours classifier* and a *10-tree random forest*, which were used for clinical outcome prediction with gene expression data. The PCA of Fig. 1 shows that the problem is intrinsically separable; hence, improving over the l-SVN baseline is very hard.

As 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:

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

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.

3.3 Ladder Network

A ladder network [15] combines both supervised and unsupervised parts in a single deep neural network; the training of both parts is simultaneous, without using layer-wise pre-training.

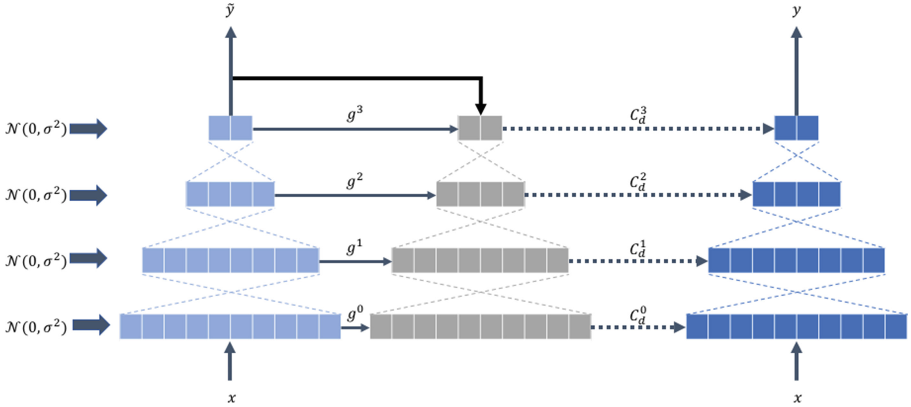

$\tilde{y}$
y
$\mathcal{N}(0,\sigma^2)$
$\mathcal{N}(0,\sigma^2)$
$\mathcal{N}(0,\sigma^2)$
$\mathcal{N}(0,\sigma^2)$
g^3
g^2
g^1
g^0
C_d^3
C_d^2
C_d^1
C_d^0
x
x

Fig. 2. Ladder networks are constituted by the combination of three components. The first one (on the left) is an encoder, corrupted with Gaussian noise ($\mathcal{N}(0, \sigma^2)$); the second one (in the middle) is a decoder, with a denoising functions (g^n); the third one is an encoder without corruption. Denoising cost (C_d^l) is used to calculate the loss function of the system.

In the Fig. 2, ladder network has three paths; two of them are feed forward paths, one standard (on the left) and one corrupted (on the right). In the middle, a denoising decoding path connects two forward passes. The unsupervised part, which contains corrupted feed forward pass and decoding pass, works as Denoising Auto Encoder (DAE) [19], in which the Gaussian noise is added to each hidden layer of corrupted forward pass. The supervised part is the feed forward network on the right; its weights are batch normalized by the unsupervised part, by using a denoising cost specific to each layer.

The loss function of the network is weighted sum of the supervised and unsupervised parts. The former is one is cross entropy cost on the top of the standard forward pass(y) and cost functions of each layer (C_d^l). The unsupervised cost is denoising square error costs weighted by a hyper parameter at each layer of the decoder.

We tuned the network by using different parameters as in [7], 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). We selected 5 hidden layers with 5000, 1500, 500, 250, 10 neuron sizes in the forward paths and we feed 120 labelled data which contains an equal number of elements from each class to the supervised path.

3.4 Ontology-Driven Convolutional Neural Networks

The concept of neighborhood in the context of biology can be applied to an extremely vast set of entities and it is usually associated with the concept of *interaction*. Biological entities can be thought to be *near* if they share a common behaviour, or present a similar pattern, or are semantically correlated.

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 transforming 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.

In order to calculate neighbour genes, we need to first define the distances between them. A distance matrix **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 . In order to compute distances, we used the Genomic and Proteomic Data Warehouse (GPDW) [4], which contains gene datasets(e.g., Entrez gene, Ensembl gene, ...), their annotations from the Gene Ontology [2] (all three aspects: cellular component, molecular function and biological process), and also the ontological relationships between them. The minimum distance information extracted from GPDW for each gene pair $<g_x, g_y>$ is:

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

where Ω_x, Ω_y are the sets of ontological terms directly annotated respectively to gene g_x and g_y, oe is a semantic expansion function towards its hypernyms and od is a function giving the minimum distance between two ontology concepts calculated as the number of the edges between them. 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 $i \neq k$.

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). In Fig. 3 a schematic representation of the pipeline is presented.

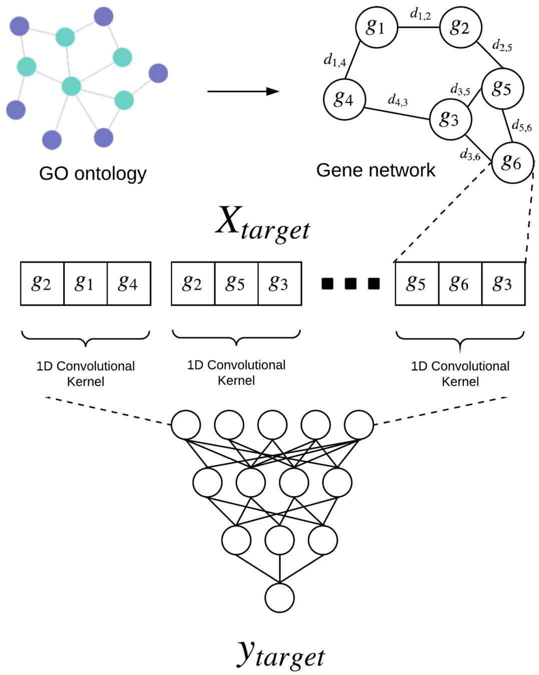


Fig. 3. Schematic representation of the ontology-driven convolutional neural network pipeline. As a first thing, we extract from the GO ontology a gene interaction network. Then, for each gene we extract its k neighbours in the graph and we perform a 1D convolution step on the feature vector composed by the gene and its neighbours. The resulting values are finally provided to a Feed Forward Neural Network

3.5 Transfer Learning Using a Combined Set of Tumors

The lack of data when concentrating on a single tumor for classification produces often unreliable and low performing models. To deal with this issue, we designed a transfer learning procedure that makes use of external information provided by a generic classification problem on the whole set of tumors.

Transfer learning is widely used in deep learning setups both academically and in industry. It provides a relatively simple method for enriching datasets submitted to the learning task. However, the model designer has to be particularly careful in associating datasets which share common characteristics and distributions. When instead the general model is trained with a dataset without a strong affinity to the target one, the hybrid model performances are degraded. In addition to this, if the target dataset has strong linearity properties with respect to the label, then the addition of the general model will yield to an over-complicated hybrid model, which will also get a performance degradation.

Transfer learning occurs as follows: Suppose that 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

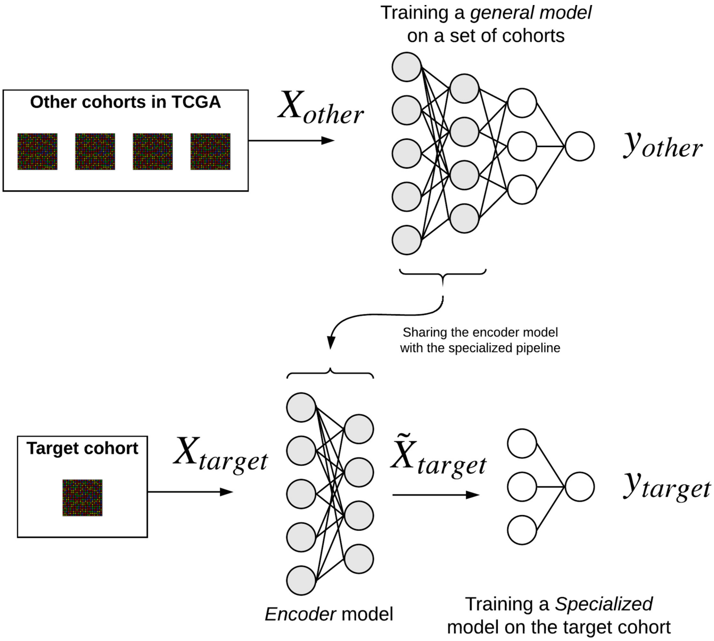
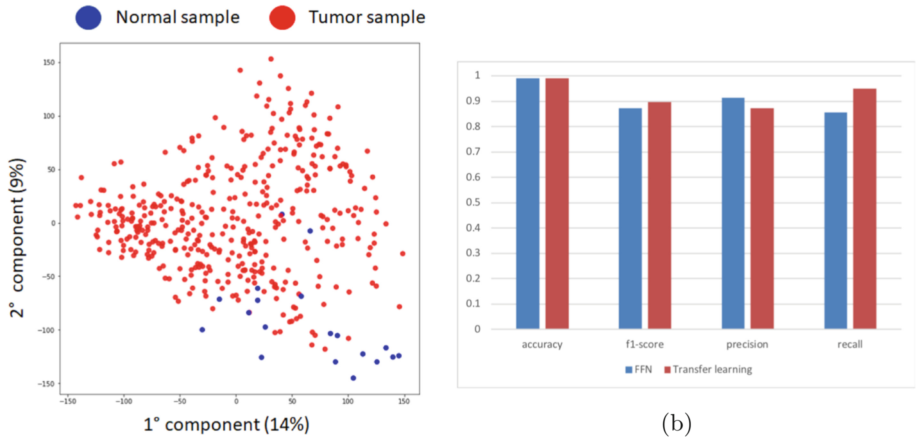

Training a *general model* on a set of cohorts
Other cohorts in TCGA
X_{other}
y_{other}
Sharing the encoder model with the specialized pipeline
Target cohort
X_{target}
$\tilde{X}_{target}$
y_{target}
Encoder model
Training a *Specialized* model on the target cohort

Fig. 4. Schematic representation of the transfer learning pipeline. The first step is to train a model on a set of TCGA cohorts with enough training samples of both classes. Then we share the encoding part of the model with a smaller neural network, which is trained on the few samples of the target cohort.

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 $y = \tilde{x} = M(x)$ (see Fig. 4).

For our application the general model is constituted by a FFN with four hidden layers respectively with 500, 200, 100 and 50 neurons for a total of 2625901 trainable parameters. Between all the layers the ReLu activation function is used with the exception of the output, 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 to be used for training and testing taking only the top 5000 variant ones. This speeds up the training and enables us to increase the performance of the model.

For this method we are going also to show the performance evaluation on an additional TCGA dataset characterized by non-trivial data distribution and low sample number: **Bladder Urothelial Carcinoma** (BLCA), whose PCA plot is shown in the Fig. 5a.



(a)

(b)

Fig. 5. (a) PCA graph for the BLCA dataset. (b) Performance measures of transfer learning on the BLCA dataset and comparison with FFN

4 Experimental Results

All the methods were evaluated using 5 times repeated 5-fold cross validation strategy and taking the mean of the various metrics as aggregated performance score.

When a classification task has a strong class imbalance, the choice of the correct evaluation metrics is particularly relevant [9]. In order to evaluate correctly the models, we must assess both the specificity and the sensitivity of the models. Therefore, together with the widely spread *accuracy* measure, we compute for each fold also the *precision*, *recall* and *f1-score*.

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 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 number 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 3. Performance measures for the four baseline models.

	Accuracy	F_1 score	Precision	Recall
Linear SVM				
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				
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				
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				
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 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)

Table 4. Performance measures for three deep learning models.

	Accuracy	F_1 score	Precision	Recall
Ladder Network				
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				
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				
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

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.

5 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 [1].

In this paper we have presented, evaluated and compared four different deep learning models for cancer detection using gene expression data. We have also compared our results with three state-of-the-art machine learning models (SVM, KNN and Random Forest). We have used as benchmark three tissue datasets coming from the TCGA repository: Breast, Kidney and Lung. We have also used a challenging dataset (BLCA) from the point of view of cardinality and data distribution to test the capabilities of a transfer learning approach.

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. We have demonstrated that structuring the feature space using gene relationships given by the GO ontology and then using convolution to extract high level representations of the input is useful for improving the detection recall and accuracy. We have also proposed a framework of transfer learning which enables to increase the recall in the three tissues while keeping comparable accuracy and precision.

Our results show that, for the considered datasets, deep learning models provide results comparable with classical machine learning methods, but are not always able to improve their performance, which is highly affected by small fluctuations (for example, an accuracy decrease of 0.01 corresponds to the misclassification of 12.18 samples on average, which is rather significant); this is due to the small cardinality and high dimensionality of datasets. With the constant growth of the sizes of genomic databases, we expect these methods to become fully applicable.

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

Acknowledgment. This work is supported by the ERC Advanced Grant 693174 GeCo (Data-Driven Genomic Computing) and by the Amazon Machine Learning Research Award on Data-driven Machine and Deep Learning for Genomics.

References

1. Agrawal, S., Agrawal, J.: Neural network techniques for cancer prediction: a survey. Procedia Comput. Sci. **60**, 769–774 (2015). https://doi.org/10.1016/j.procs.2015.08.234
2. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. Nature Genet. **25**, 25 (2000)
3. Blagus, R., Lusa, L.: Smote for high-dimensional class-imbalanced data. BMC Bioinform. **14**(1), 106 (2013). https://doi.org/10.1186/1471-2105-14-106
4. Canakoglu, A., et al.: Integrative warehousing of biomolecular information to support complex multi-topic queries for biomedical knowledge discovery. In: BIBE, pp. 1–4. IEEE (2013)
5. Danaee, P., Ghaeini, R., Hendrix, D.A.: A deep learning approach for cancer detection and relevant gene identification. In: Pacific Symposium on Biocomputing 2017, pp. 219–229. World Scientific (2017)
6. Furey, T.S., et al.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics **16**(10), 906–914 (2000)
7. Golcuk, G., Tuncel, M.A., Canakoglu, A.: Exploiting ladder networks for gene expression classification. In: Rojas, I., Ortuño, F. (eds.) IWBBIO 2018. LNCS, vol. 10813, pp. 270–278. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78723-7_23
8. Guyon, I., et al.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1), 389–422 (2002). https://doi.org/10.1023/A:1012487302797
9. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239

10. Hijazi, H., Chan, C.: A classification framework applied to cancer gene expression profiles. J. Healthc. Eng. **4**(2), 255–284 (2013). https://doi.org/10.1260/2040-2295.4.2.255
11. LeCun, Y., et al.: Deep learning. Nature **521**, 436 (2015)
12. Liu, J., Wang, X., Cheng, Y., Zhang, L.: Tumor gene expression data classification via sample expansion-based deep learning. Oncotarget **8**(65), 109646–109660 (2017). https://doi.org/10.18632/oncotarget.22762
13. Lonsdale, J., et al.: The genotype-tissue expression (GTEx) project. Nat. Genet. **45**(6), 580 (2013)
14. Rahimi, A., Gönen, M.: Discriminating early-and late-stage cancers using multiple kernel learning on gene sets. Bioinformatics **34**(13), i412–i421 (2018)
15. Rasmus, A., et al.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems, pp. 3546–3554 (2015)
16. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
17. Shen, R., Olshen, A.B., Ladanyi, M.: Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics **25**(22), 2906–2912 (2009). https://doi.org/10.1093/bioinformatics/btp543
18. Statnikov, A., et al.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinform. **9**(1), 319 (2008)
19. Vincent, P., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(Dec), 3371–3408 (2010)
20. Weinstein, J.N., et al.: The cancer genome atlas pan-cancer analysis project. Nat. Genet. **45**(10), 1113–1120 (2013)
21. Yousefi, S., et al.: Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Sci. Rep. **7**(1), 11707 (2017)
22. Zhang, L., et al.: Gene expression profiles in normal and cancer cells. Science **276**(5316), 1268–1272 (1997). https://doi.org/10.1126/science.276.5316.1268
23. Zuyderduyn, S.D., et al.: A machine learning approach to finding gene expression signatures of the early developmental stages of squamous cell lung carcinoma. Cancer Res. **66**(8 Supplement), 431–432 (2006). http://cancerres.aacrjournals.org/content/66/8_Supplement/431.4