Deep Learning to Analyze RNA-Seq Gene Expression Data

D. Urda[1,3(✉)], J. Montes-Torres[2,3], F. Moreno[2], L. Franco[2,3], and J.M. Jerez[2,3]

[1] Andalucía Tech, ETSI Informática (España), Universidad de Málaga, Málaga, Spain
durda@lcc.uma.es
[2] Departamento de Lenguajes y Ciencias de la Computación, ETSI Informática (España), Universidad de Málaga, Málaga, Spain
[3] Inteligencia Computacional en Biomedicina (España), Instituto de Investigación Biomédica de Málaga (IBIMA), Málaga, Spain

Abstract. Deep learning models are currently being applied in several areas with great success. However, their application for the analysis of high-throughput sequencing data remains a challenge for the research community due to the fact that this family of models are known to work very well in big datasets with lots of samples available, just the opposite scenario typically found in biomedical areas. In this work, a first approximation on the use of deep learning for the analysis of RNA-Seq gene expression profiles data is provided. Three public cancer-related databases are analyzed using a regularized linear model (standard LASSO) as baseline model, and two deep learning models that differ on the feature selection technique used prior to the application of a deep neural net model. The results indicate that a straightforward application of deep nets implementations available in public scientific tools and under the conditions described within this work is not enough to outperform simpler models like LASSO. Therefore, smarter and more complex ways that incorporate prior biological knowledge into the estimation procedure of deep learning models may be necessary in order to obtain better results in terms of predictive performance.

Keywords: Deep learning · RNA-Seq · Personalized medicine · Machine Learning · Biomarkers discovery

1 Introduction

In the last years, artificial neural networks have raised back interest of the research community on this family of Machine Learning (ML) models under the tag "deep learning" [11]. Behind this recent interest, there are well-known companies, such as Google or Microsoft among other private and public entities, that have made big investments to succeed applying deep neural networks into several Artificial Intelligence (AI) areas [6,9,10,15]. The implementation of

I. Rojas et al. (Eds.): IWANN 2017, Part II, LNCS 10306, pp. 50–59, 2017.
DOI: 10.1007/978-3-319-59147-6_5

new initialization and training procedures for this ML models [8], supported by the high computing resources available at these entities, has finally allowed to overcome the barrier that artificial neural networks were facing ten years ago.

Deep learning is actively used today in a wide range of fields, including Bioinformatics and Computational Medicine. Its strength working with graphical information has motivated many researches in the last few years to incorporate this ML models in their works. Thus, it has been successfully applied in medical image processing, where deep convolutional neural network have been proven to be robust pixel classifiers [3–5]. Indeed, solving image classification problems is not the only way deep learning can assist biomedical researches. As some recent works show [12,18], deep neural networks are being used for predictive modeling, using RNA-Seq data as input. We may ask ourselves, though, whether the use of deep learning, when it comes to produce predictive models, is as straight forward as it is in other kind of problems.

Despite the increasing amount of papers referencing the use of deep learning models in biomedical related areas, the authors of this work consider that there is still a long way to go in order to achieve a relevant improvement with respect to classical models in certain applications, like predicting the outcome for patients with gene expression datasets. This work aims to provide a first approximation of how to use a multi-layer feed-forward artificial neural network to analyze RNA-Seq gene expression data. For this purpose, three public RNA-Seq dataset are considered in order to predict the vital status of a patient at time t. Two deep learning models, which differ in the feature reduction procedure applied, are compared to a standard linear model with l_1-regularization (LASSO with homogeneous priors). Furthermore, feature selection, models estimation, selection and evaluation are performed using an honest validation scheme.

The rest of the article is organized as follows. Section 2 describes the datasets and ML models considered within the analysis as well as a description of the validation strategy used to compare the performance of the models. Then, Sect. 3 shows the results obtained with each model on the studied datasets. Finally, Sect. 4 provides some conclusions for this work.

2 Materials and Methods

2.1 Datasets

Free-public RNA-Seq gene expression datasets can be easily downloaded from The Cancer Genome Atlas (TCGA) website[1]. In particular, this work analyzes three datasets that have already been pre-processed to take into account batch effects and normalized through the RSEM procedure [13]. The first dataset is linked to Breast Invasive Carcinoma (BRCA) containing 199 cases and 1013 controls. The second database contains 81 cases and 245 controls of Colon Adenocarcinoma (COAD). The last dataset corresponds to a joint cohort of Kidney Chromophobe, Kidney renal clear cell carcinoma and Kidney renal papillary cell carcinoma (KIPAN) with 267 cases and 753 controls. Each sample in

[1] https://cancergenome.nih.gov/.

Table 1. Information of the RNA-Seq datasets: number of samples (N), number of genes (P) and class distribution ($control = 0$, $cases = 1$).

Name	N	P	Controls	Cases
BRCA	1212	20021	1013	199
COAD	326	19467	245	81
KIPAN	1020	20144	753	267

these datasets is finally described by approximately 20000 genes after applying a sanity check procedure where those genes that appeared to be constant across the sample are removed. Additionally, a $log_2(exp + 1)$ transformation of the genes expression levels was performed to make their distribution look as close as possible to a normal distribution. Table 1 shows the overall description of each dataset where the event of interest considered is the vital status of a given patient (0 = "alive" are controls, 1 = "dead" are cases).

2.2 Methods

This work uses two different machine learning models to learn a given dataset $D = \{\boldsymbol{x_i}, y_i\}$ of N samples, where $i \in [1, N]$, $\boldsymbol{x_i}$ represents a vector of P genes expression level describing the i-th sample, and y_i is the class label for the i-th sample. On one hand, a linear model is used assuming that the independent variable y_i can be represented as a linear combination of the dependent variables $\boldsymbol{x_i}$. On the other hand, another model that enables to capture non-linear relationships is also considered as a possible alternative to linear models in order to push forward the predictive performance of this family of models. Next, we describe the models considered in this work:

- ***Lasso***: this model is the baseline model in this work and corresponds to a standard LASSO model [16] with homogeneous priors. LASSO is a well-known linear model in the bioinformatics community and it is widely used for several and diverse tasks. LASSO tries to optimize the minimization problem depicted in Eq. 1:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} (y_i - f(\boldsymbol{x_i}, \boldsymbol{\beta}))^2 + \lambda \sum_{j=1}^{P} |\beta_j| \tag{1}$$

  In contrast to linear or logistic regression, this model includes an l_1-penalty term to set as many features as possible to zero unless the data tells us not to do it. Moreover, this term is controlled by a regularization parameter λ ($\lambda = 0$ would exactly correspond to the objective function in linear or logistic regression). Therefore, LASSO is an embedded method that performs feature selection at the same time that the model is adjusted to data. The R package *glmnet* [7] has been used to estimate a LASSO model due to its

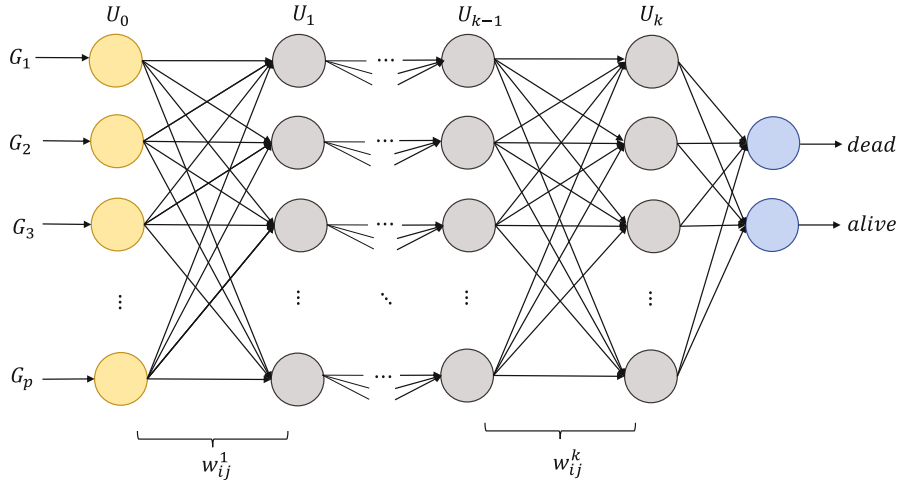

Fig. 1. General architecture of a multi-layer feed-forward artificial neural network to perform binary classification. The input layer composed of P genes expression levels is connected to units of subsequent k-hidden layers through synaptic weights ω_{ij}^k.

easy interface that automatically allows to learn the regularization parameter λ through cross-validation.

– ***DeepNets***: among the different deep learning models available, a multi-layer feed-forward artificial neural network was chosen to learn existing non-linearities between the input and output spaces. Figure 1 shows the classical architecture of these artificial neural networks. In concrete, it shows an architecture of $K+1$ layers (one input layer and K hidden layers) and a vector $\boldsymbol{U} = \{U_0, U_1, ..., U_k\}$ representing the number of units in the 0-th layer (input layer) and the k-th hidden layer, respectively.

In this type of models, it can be easily inferred that the number of parameters to be learned ω_{ij}^k (synaptic weights), where $k \in [1, K]$, $i \in [1, U_k]$ and $j \in [1, U_{k-1}]$, increases drastically in comparison to the number of parameters $\boldsymbol{\beta}$ of the standard LASSO. The more layers and the more neurons per layer we add, the more number of parameters ω_{ij}^k will be obtained and needed to be learned. Conversely, this number remains equal to the number of genes in the input space for the LASSO model. Therefore, deep nets require to introduce strategies to avoid overfitting such as regularization or dropout. Regularization aims to impose some constraint in the optimization procedure, being the l_1-penalty (lasso penalty: set as many ω_{ij}^k as possible to zero) or the l_2-penalty (ridge penalty: avoid setting ω_{ij}^k to high values) the most known and used ones. The dropout strategy [14] aims to "disconnect" some of those links between neurons of different layers to decrease the number of parameters that needs to be learned. Both strategies are complementary and applicable jointly. The R package *h2o* [1] has been used in this work to fit a deep net to data using regularization and dropout strategies thus dealing with overfitting issues.

Table 2. Subset of deep nets' parameters included in the *h2o* framework implementation. A list with the name of each parameter tuned within a random search procedure is provided. For each studied dataset and parameter, a list of tried values (in brackets) or range of values (in squared brackets) is shown.

Parameter	BRCA COAD KIPAN
Activation function	{Rectifier, Tanh, Maxout}
Number of hidden layers	{2, 3, 4}
Number of units per layer	[10, 200]
L1 regularization	[0.001, 0.1]
L2 regularization	[0.001, 0.1]
Input dropout ratio	[0.001, 0.1]
Hidden dropout ratios	[0.001, 0.1]

Furthermore, two different feature dimensionality reduction techniques were applied prior to estimating a deep net in order to reduce the input space and, therefore, the number of parameter to be learned:

- $DeepNet_i$: this procedure applies a univariate t-test to compare if the difference of genes expression levels in controls and cases are statistically significant. Genes under a p-value threshold of 0.001 are retained and then given to a correlation feature reduction procedure that gets rid of highly correlated genes until the number of retained genes is similar to the average number of genes kept in the standard LASSO.
- $DeepNet_{ii}$: this procedure uses a standard LASSO model to retrieve the most important genes for the given outcome.

In both cases, the selected genes are given as input to fit a deep net to data, discarding the remaining genes from the analysis. Additionally, a random search was performed to tune some parameters linked to a deep net model in *h2o*, where Table 2 shows the parameters considered to be tuned together with their respective ranges for each dataset analyzed.

2.3 Validation Strategy

A known and valid evaluation strategy is always required in order to estimate generalization error and compare the performance of the models considered in the analysis. In particular, this work implements Z repetitions of k-fold cross-validation, where $Z = 20$ and $k = 10$. For a given repetition, this evaluation strategy divides the complete dataset into k non-overlapping folds of equal sizes and applies an iterative procedure that uses $k - 1$ folds to fit the models and the unseen fold left apart to test the performance (rotating train and test folds on each iteration). The utilization of this validation strategy rather than other

Algorithm 1. Pseudocode of our methodological approach

```
dataset ⇐ {"BRCA", "COAD", "KIPAN"} //choose one option
model ⇐ {"lasso", "deepnet"} //choose one option
filtering ⇐ {"ttest − cor", "lasso"} //choose one option
X ⇐ load_design_matrix(dataset)
Y ⇐ load_outcome(dataset)
partitions ⇐ load_partitions(dataset)

for Z = 1 → 20 do
   folds ⇐ get_folds(partitions, Z)
   for k = 1 → 10 do
      Xtrain ⇐ get_design_matrix(X, folds, k, "train") //training data
      Xtest ⇐ get_design_matrix(X, folds, k, "test") //test data
      Ytrain ⇐ get_outcome(Y, folds, k, "train")
      Ytest ⇐ get_outcome(Y, folds, k, "test")

      if (model=="deepnet") then
         retained_genes ⇐ apply_filtering(Xtrain, Ytrain, filtering)
         Xtrain ⇐ Xtrain[, retained_genes]
      end if

      fitted ⇐ fit_model(Xtrain, Ytrain, model) //performs model selection internally
      predictions ⇐ predict(fitted, Xtest)
      measures[k] ⇐ performance(Ytest, predictions)
   end for
   results[Z] ⇐ mean(measures)
end for

print(results)
```

well-known strategies such as leave-one-out, bootstrapping, holdout, etc., is motivated on (i) its simplicity and small computational resources needed, and (ii) the proved that there is no universal unbiased estimator of the variance of k-fold cross-validation [2]. Algorithm 1 contains a high-level description of the methodological approach used to carry out the analysis.

The Area Under the Curve (AUC) was computed to compare the performance of each model since the three studied datasets are highly imbalanced (see Table 1). Additionally, both the number of genes retained by each of the feature reduction procedures considered and the total time (in minutes) required to execute the validation strategy described were computed to open a discussion over the results.

3 Results

This work has analyzed three cancer-related RNA-Seq datasets using the models described in Sect. 2.2. The quantitative results are shown in Table 3. In general, it can be seen that the predictive performance in terms of AUC is relatively poor independently of the model used. On two out of three databases, BRCA and COAD, the performance measured by the AUC is not over 0.65, and particularly in the COAD dataset the prediction of the vital status of a patient from RNA-Seq gene expression profiles turned out to be quite difficult (AUC under 0.6, close to random predictions). Conversely, the predictive performance of a simple linear model on the KIPAN dataset seems to be good with AUC values around 0.77.

Table 3. Average AUC results and number of retained genes for 20 repetitions of 10-fold cross-validation over each RNA-Seq dataset using the three models proposed: standard *Lasso*, $DeepNet_i$ and $DeepNet_{ii}$. 95% CI and standard deviation are shown for the AUC and #genes columns respectively. The last column shows the total number of minutes required for the corresponding analysis.

Dataset	Model	AUC	#genes	time (mins.)
BRCA	*Lasso*	0.65 [0.62, 0.67]	285.54 ± 25.83	501.79
	$DeepNet_i$	0.62 [0.58, 0.65]	242.02 ± 8.01	2294.83
	$DeepNet_{ii}$	0.65 [0.63, 0.68]	285.54 ± 25.83	9768.37
COAD	*Lasso*	0.57 [0.52, 0.63]	69.64 ± 11.63	30.89
	$DeepNet_i$	0.58 [0.54, 0.62]	37.29 ± 1.52	2699.84
	$DeepNet_{ii}$	0.57 [0.52, 0.61]	69.64 ± 11.63	2370.15
KIPAN	*Lasso*	0.77 [0.76, 0.78]	268.81 ± 32.54	93.60
	$DeepNet_i$	0.72 [0.68, 0.75]	201.64 ± 3.44	2633.52
	$DeepNet_{ii}$	0.75 [0.73, 0.78]	268.81 ± 32.54	9281.08

Focusing on the models considered in this work (*Lasso*, $DeepNet_i$, $DeepNet_{ii}$), the results obtained across the three databases confirmed us that the straightforward use of existing implementations of a multi-layer feed-forward artificial neural network (such as the R package *h2o*) will very rarely push the predictive performance further away compared to a simple regularized linear model. In two out of three databases, BRCA and KIPAN, deep learning models do not outperform the baseline model *Lasso*. On the other hand, in the COAD dataset a deep learning model estimated on the retained genes after applying a univariate t-test combined with a correlation filtering procedure turned out to slightly improve the AUC after executing 20 repetitions of 10-fold cross-validation (AUC from 0.57 to 0.58). Nevertheless, this tiny improvement is not statistical significant as indicated by the overlap observed for the 95% confidence intervals, indicating that our baseline model *Lasso* can also achieve similar predictive performance depending on the data used to fit the models.

Regarding the number of genes obtained after filtering reduction to finally estimate the models, two out of the three model (*Lasso* and $DeepNet_{ii}$) are using exactly the same average numbers since genes retained by the embedded method *Lasso* are used in both cases (approximately 275 genes in BRCA and KIPAN, or 70 genes in COAD). It turned out that the average numbers of genes retained by *Lasso* is slightly higher across the three databases, although the numbers of genes retained by the filtering procedure used in $DeepNet_i$ is close to the one in *Lasso* (242 in BRCA, 37 in COAD and 202 in KIPAN), thus making these results comparable. However, it can be highlighted the larger variability in terms of size of the genetic signatures obtained by *Lasso* in contrast to a simple t-test followed by a correlation filtering procedure. Analyzing the robustness of the genetic signatures found is beyond the scope of this work since it would lead to a complete different paper, although that type of analysis will constitute a

complement to these results in order to state the (un)suitability of *Lasso* as model for biomarkers discovery [17].

Running time is also an important factor to take into account when using deep learning models. Particularly, the fitting procedure of deep learning models considered within this work required much more time due to the number of parameters that need to be tuned (see Table 2), in contrast to the standard *Lasso* model. $DeepNet_i$ needed minimum five times more minutes (see BRCA dataset in Table 3) to achieve similar predictive performance than *Lasso*, going up to almost 100 times more minutes in the COAD dataset. The case of $DeepNet_{ii}$ is even worse in BRCA and KIPAN, where this model required 3 times more minutes than the $DeepNet_i$. Independently of the quantitative numbers, these results clearly show us how expensive the estimation procedure of deep learning models is in comparison to more simple models. Moreover, in this particular analysis and under the described conditions the use of deep learning models is not suggested since it will take minimum five times more minutes to finally obtain similar predictive performance.

4 Conclusions

This paper has presented a first approximation on the straightforward use of deep learning models existing implementations for the analysis of RNA-Seq gene expression profiles databases. In concrete, it considered a multi-layer feed-forward artificial neural network as deep learning model in combination with two different feature reduction techniques, and a standard LASSO (regularized linear model) as the baseline model to try to outperform. This work has used an honest validation strategy to analyze three public cancer-related databases, where both feature reduction and model estimation were performed in a train dataset and the resulting fitted model was evaluated in an independent test dataset.

In general, the combination of deep learning models with the two considered feature reduction techniques very rarely outperformed a simple standard LASSO in terms of AUC. Furthermore, the estimation of the proposed deep learning models required minimum five times more minutes than LASSO due to the number of parameters that need to be tuned in such models, thus suggesting that using deep learning under the described conditions to predict the vital status of a patient from RNA-Seq data is not suggested. The exploration of this research line lead us to conclude that using a simple feature reduction procedure to reduce the number of genes and subsequently fit a deep learning model will take us much more execution time in order to obtain similar predictive performances.

Despite the discouraging results obtained in this work, there is no need to spread out a negative message in relation to the application of deep learning for RNA-Seq data analysis. Conversely, there is a big hope in pushing predictive performance forward with this type of models. To this end, this work has allowed us to realize that smarter use of deep learning models must be done to be successful in this research line. For instance, deep learning as stack auto-encoders could be used to somehow compress the information of 20000 genes into fewer

variables, thus allowing to use any other ML model with these new compressed features as inputs. On the other hand, imposing constrains in the optimization process of deep learning models in such a way that biological knowledge is taken into account may lead to better results in terms of performance. Finally, finding ways of making this type of models interpretable would be desirable from the clinical point of view, and for this purpose using published knowledge of relationships between Single Nucleotide Polymorphisms (SNPs), genes, pathways, proteins, etc., could be a possible way of defining the network architecture.

Acknowledgements. The authors acknowledge support through grants TIN2014-58516-C2-1-R from MICINN-SPAIN which include FEDER funds, and from ICE Andalucía TECH (Spain) through a postdoctoral fellowship.

References

1. Aiello, S., Kraljevic, T., Maj, P., with contributions from the H2O.ai team: h2o: R Interface for H2O (2016). https://CRAN.R-project.org/package=h2o. R package version 3.10.0.8
2. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of K-fold cross-validation. J. Mach. Learn. Res. **5**, 1089–1105 (2004)
3. Cadieu, C., Hong, H., Yamins, D., Pinto, N., Ardila, D., Solomon, E., Majaj, N., DiCarlo, J.: Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS Comput. Biol. **10**(12) (2014)
4. Ciompi, F., de Hoop, B., van Riel, S., Chung, K., Scholten, E., Oudkerk, M., de Jong, P., Prokop, M., van Ginneken, B.: Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Med. Image Anal. **26**(1), 195–202 (2015)
5. Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 411–418. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40763-5_51
6. Deng, L., Li, J., Huang, J.T., Yao, K., Yu, D., Seide, F., Seltzer, M.L., Zweig, G., He, X., Williams, J., Gong, Y., Acero, A.: Recent advances in deep learning for speech research at microsoft. In: ICASSP, pp. 8604–8608. IEEE (2013)
7. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1–22 (2010)
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Society for Artificial Intelligence and Statistics (2010)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., Ng, A.: Building high-level features using large scale unsupervised learning. In: International Conference on Machine Learning (2012)
11. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

12. Leung, M., Xiong, H., Lee, L., Frey, B.: Deep learning of the tissue-regulated splicing code. Bioinformatics **30**(12), I121–I129 (2014)
13. Li, B., Dewey, C.N.: RSEM: accurate transcript quantification from rna-seq data with or without a reference genome. BMC Bioinform. **12**(1), 323 (2011)
14. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
15. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR abs/1409.3215 (2014)
16. Tibshirani, R.: Regression shrinkage and selection via the lasso: a retrospective. J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) **58**(1), 267–288 (1996)
17. Urda, D., Aragon, F., Veredas, F., Franco, L., Jerez, J.M.: L1-regularization model enriched with biological knowledge. In: Proceedings of the 5th International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2017) (2017)
18. Wenger, Y., Galliot, B.: Rnaseq versus genome-predicted transcriptomes: a large population of novel transcripts identified in an illumina-454 hydra transcriptome. BMC Genomics **14**(1) (2013)