Understanding Cancer Phenomenon at Gene-Expression Level by using a Shallow Neural Network Chain

Pietro Barbiero[1], Andrea Bertotti[2], Gabriele Ciravegna[3], Giansalvo Cirrincione[4], Elio Piccolo[3], and Alberto Tonda[5]

Politecnico di Torino, DISMA, Turin, Italy
pietro.barbiero@studenti.polito.it
Università degli studi di Torino, Dipartimento di Oncologia.
Candiolo Cancer Institute - FPO, IRCCS, Italy
andrea.bertotti@ircc.it
Politecnico di Torino, DAUIN, Turin, Italy
elio.piccolo@polito.it
University of Picardie Jules Verne, Lab. LTI, Amiens, France.
University of South Pacific, Suva, Fiji
exin@u-picardie.fr
UMR 782, Université Paris-Saclay, INRA, Thiverval-Grignon, France
alberto.tonda@inra.fr

Abstract. Exploiting the availability of the largest collection of Patient-Derived Xenografts from metastatic colorectal cancer annotated for response to therapies, this manuscript aims to characterize the biological phenomenon from a mathematical point of view. In particular, we design an experiment in order to investigate how genes interact with each other. By using a shallow neural network model, we find some sub-spaces where the resistance phenomenon may be much easier to understand and analyze.

Keywords: artificial neural networks, classification, colorectal cancer, deep learning, DNA microarray, high dimensionality, manifold analysis, Patient-Derived Xenograft

1 Biological Introduction

The cancer phenomenon seems to be the result of a different sequence of genetic alterations. In this difficult setting, clinical treatments add an external complexity to the tumor behavior. In recent years, Patient-Derived Xenografts (PDXs) have emerged as powerful tools for biomarker discovery and drug development in oncology [1][2][3]. PDXs are obtained by propagating surgically derived tumor specimens in immunocompromised mice. Through this, cancer cells remain viable ex-vivo and retain the typical characteristics of different tumors from different patients. Hence, they can effectively recapitulate the intra- and inter-tumor heterogeneity that is found in real patients. Based on this idea, the PDX technology has been leveraged to conduct large-scale preclinical analyses to identify reliable correlations between genetic or functional traits and sensitivity to anti-cancer drugs. In this context, during the last decade we have been assembling the largest collection of PDXs from metastatic colorectal cancer (mCRC) available worldwide in an academic environment. Such resource has been widely characterized at the molecular level and has been annotated for response to therapies, including cetuximab, an anti-EGFR antibody approved for clinical use [4][5][6]. Such multi-layered information has been already leveraged to reliably anticipate clinical findings [7] with major therapeutic implications. Here it is proposed to exploit and combine available transcriptional data obtained from mCRC PDXs of our collection through the Illumina bead array technology [8].

2 Data set

The data consists in a DNA microarray composed of the expression of 20023 genes in 403 CRC murine tissues. Each cancerous tissue is associated with a Boolean variable describing the tumor response to cetuximab, as two classes, responsive and not responsive to treatments, as described in previous works [9]. For the purpose of the analyzes, genes are considered as features while each patient corresponds to a sample. The data set is normalized using a column statistical scaling (zscore) on features.

3 Objective

Previous works concerning patient classification on this data set have shown remarkable results reaching very high accuracy. In [10], authors have dealt with the high-dimensional space of genes by using dimensionality reduction algorithms and manifold analyzes. In [11], they have shown how several machine learning classifiers are able to assess patient response to drugs reaching similar performances in cross-validated test sets. In this work, going beyond the previous analyzes, we try to investigate the structure of the biological phenomenon at the gene-expression level from a mathematical point of view. In particular, we are interested in understanding how the cancer resistance to treatments can be explained by using only the information contained in a DNA-microarray. At the same time, we take advantage of the availability of real word data to make some considerations about the behavior of shallow neural networks dealing with high dimensional spaces.

4 Shallow Neural Network

4.1 Mathematical Model of the Network Architecture

The neural network architecture we used in the following experiments is known as *Adaline* [12][13][14]. It has 20023 inputs corresponding to the input features (genes) and one output neuron equipped with a linear output function. The network has not hidden layer of neurons. In forward propagation, the network computes the dot product between the weight vector w and the i^{th} sample $x^{(i)}$ plus the bias b. This corresponds to a weighted sum of the inputs with bias correction (as in a linear regression model):

$$z^{(i)} = w^T x^{(i)} + b \tag{1}$$

$$\hat{y}^{(i)} = f(z^{(i)}) = z^{(i)} \tag{2}$$

where w is the weight vector, b the bias, f the activation function and $\hat{y}^{(i)}$ the network output.

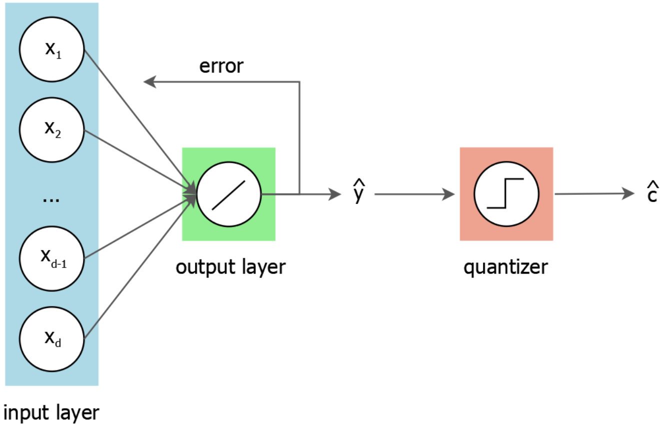

Fig. 1. Shallow neural network architecture.

4.2 Objective Function

The squared error function evaluates the performance of the algorithm on an individual sample:

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = (y^{(i)} - \hat{y}^{(i)})^2 \tag{3}$$

where $y^{(i)}$ is 1 if the i^{th} sample belongs to class 1 and 0 if it belongs to class 0. In order to evaluate the global performance of the classifier, we use a cost function with L2 regularization of the weights. The L2 regularization is a technique that applies to objective functions in ill-posed optimization problems [15][16]. The proposed neural model is ill-posed since the solution is not unique and it changes continuously according to the initial conditions and cross-validation randomness. Appending a term in the cost function that penalizes large weights, the search space reduces, as well as the number of solutions, and the problem becomes less prone to initial conditions:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}||w||_w^2 \tag{4}$$

where λ is the regularization parameter and $||w||_w^2$ is the L2 norm of the weight vector. For big values of λ the regularization is stronger, increasing the penalization related to weights. As a result, the weights which are useless for the purpose of minimizing the MSE (i.e. the first part of the objective function) are shrunk towards zero. On the contrary, for low values of λ the regularization effect is weaker[1]. In order to provide a quantitative measure of the network performance, we transform the regression outcomes into class labels by using an Heaviside step function:

$$\hat{c}^{(i)} = \frac{d}{d\hat{y}}\max\{0, \hat{y}^{(i)}\} \tag{5}$$

and we compute the accuracy as it was a classification task.

4.3 Parameter Optimization

Since the cost function states how bad are the current predictions, the problem of the learning process is equivalent to the minimization of the cost function. Whereas the training samples are fixed, the cost function depends only on the networks parameters (weights and bias). So, the cost function minimization is equivalent to the optimization of the network parameters. For the following analyzes we use the Adaptive momentum estimation optimizer (also known as Adam). Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [17]. It is a variant of the gradient descent algorithm designed to combine the advantages of two recently popular methods: AdaGrad and RMSProp. According to [17] some of Adam's advantages are that its step-sizes are approximately bounded by the learning rate, it does not require a stationary objective, it works with sparse gradients, and it naturally performs a form of step size annealing. In the context of feed-forward neural networks, the objective function to be minimized is the cost function $J_t(\theta)$, where t denotes the t^{th} epoch and θ is a label for w and b. The authors identify with g_t the gradient, i.e. the vector of partial derivatives of J_t, w.r.t w and b evaluated at epoch t (6). This estimate is then used to update two exponential moving averages of the gradient (m_t, (7)) and the squared gradient (v_t, (8)). The two hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages. High values for βs reduces the time-window size of the moving averages, resulting in low inertial effect and greater oscillations. On the contrary, low values of βs increase the time-window size, providing a stronger smoothing effect. The first moving average m_t is an

[1] The described shallow neural network model is equivalent to a linear regression model with an L2 regularization of the parameters also known as Ridge Regression [16].

estimate of the 1^{st} order moment (the mean) of the gradient. The second one instead is an estimate of the 2^{nd} order moment (the uncentered variance) of the gradient. Since these moving averages are initialized as vectors of 0's, the moment estimates are biased towards zero during the initial time-steps (especially when the decay rates are small, i.e. the βs are close to 1). This annoying issue can be alleviated by the bias correction shown in (9) and (10). The ratio of the two moving averages corresponds to a standardization of the first order moment of the gradient. The network parameters are finally updated by using the usual formula of the gradient descent in (11). The term ϵ (typically 10^{-8}) ensures that the denominator is always non-zero, avoiding numerical issues.

$$g_t = \nabla_\theta J_t(\theta_{t-1}) \tag{6}$$

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \tag{7}$$

$$v_t = \beta_2 \cdot m_{t-1} + (1-\beta_2) \cdot g_t \odot g_t \tag{8}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{9}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{10}$$

$$\theta = \theta - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{11}$$

The initial conditions are: $m_0 = 0$, $v_0 = 0$ and $t = 0$. Typical values for βs are $\beta_1 \approx 0.9$ and $\beta_2 = 0.999$. Overall, Adam is a very efficient algorithm, requiring very few computations and memory space, which is crucial given the data set size.

5 Experiment and Discussion

The experiment consists in a chain of cross-validated training of the neural model described above. In particular, at each iteration the neural network is trained 30 times, each of which using a 10-fold cross validation with random folds. The neural network hyperparameters are heuristically fixed to:

$$\lambda = \frac{1}{\#\text{samples}} \tag{12}$$

$$\alpha = \frac{1}{\lambda + L + 1} \tag{13}$$

$$\tag{14}$$

where L is the maximum sum of the squares over all samples [18][19]. Since the objective function to minimize contains the L2 norm of the weights (see equation 4), then those who are not fundamental for the classification purpose are shrunk towards zero by the optimizer [16]. Fig. 2 and 3 show respectively the histogram and the notched box plot of the weights after the training process in the first iteration. Notice that most of the weights are zero or very close to zero. This means that their contribution in the weighted sum in equation (1) is almost negligible. Exploiting this result, for each fold we take note of the input features (i.e. the genes) which corresponds to weights having an absolute value w_j after a training process:

$$|w_j| > 2\sigma_w \tag{15}$$

where σ_w is the variance of the weights distribution (see fig. 3). At the end of the 30 iterations, we found that some input features are chosen more frequently than others.

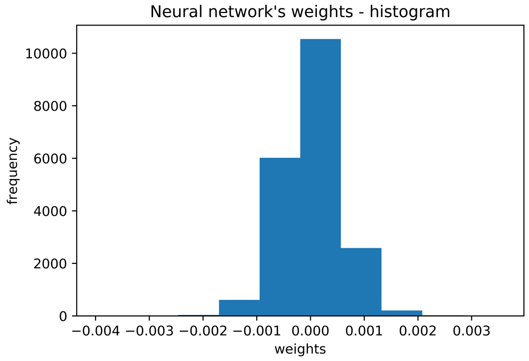

Fig. 2. An example of histogram of the neural network weights after the first training iteration.

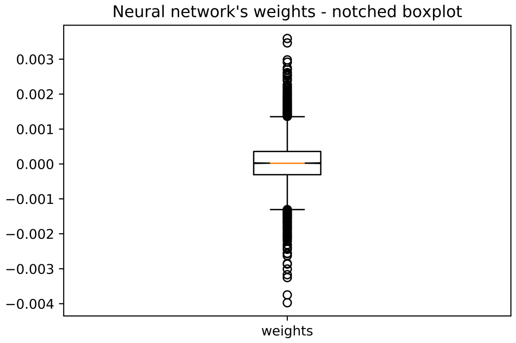

Fig. 3. An example of notched box plot of the neural network weights after the first training iteration.

Biologically speaking, this result suggests that the information contained in the DNA-microarray related to these genes may be relevant in understanding the cancer resistance to drugs. In order to investigate more deeply the biological phenomenon, we repeat the same of experiment modifying the database by keeping only the most frequently selected features, i.e. those which are selected at least half of the times after the 300 training processes. So, iteration by iteration we gradually reduce the number of input features used to train the neural network. Fig. 4 shows for each iteration the number of features used to train the neural network and the corresponding 10-fold cross validation accuracy. The blue bars correspond to the feature selection technique described above, while the violet ones to the ANOVA-F statistic method[2]. Notice that, initially, by using the original data set, the cross-validation accuracy is around 0.7. Such result may have two main explanations. First, the classes are not perfectly balanced, since the 66% of samples belong to class 0. Secondly, the high-dimensionality of data may generate a slight overfitting. However, by reducing the input features using the method described above, the cross-validation accuracy raises above 0.9, decreasing progressively as the number of features are further diminished.

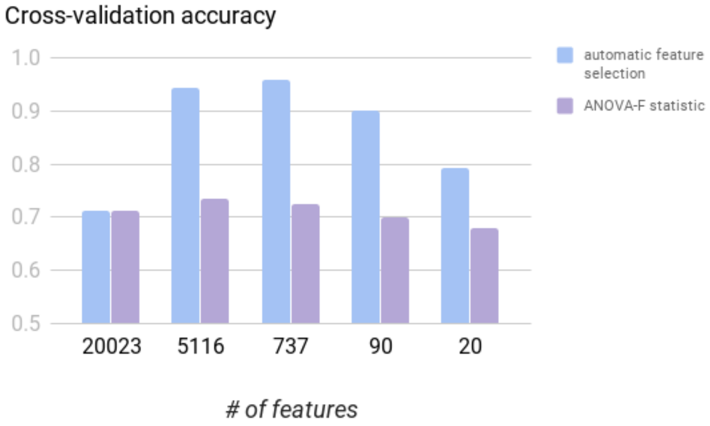

Fig. 4. Histogram displaying the 10-fold cross validation accuracy at each iteration of the experiment.

Notice that, the neural network used as classifier is linear, i.e. geometrically speaking it delimits the input space with an hyperplane in order to classify data. This means that it provides better results if the underlying phenomenon repre-

[2] The corresponding standard deviation is always in the order of few percentage decimals and it is not directly displayed since it is not relevant for the purpose of the discussion. However, you can reproduce the experiment by using our code if you need more precision.

sented by the input data set is also linear. The results in fig. 4 show how the shallow neural network classifier provides better results in the 737-dimensional space identified by the feature extracted using the method described above than in the original data set. So, it may suggest that the underlying biological phenomenon at the DNA-microarray level is more linear in the reduced space than in the original one. Practically speaking, a linear problem is much easier to understand and tackle because the superposition principle holds i.e. the net response caused by two or more stimuli is the sum of the responses that would have been caused by each stimulus individually. Therefore, from a biological point of view, these results may suggest that the above experiment generates sub-spaces of the input features where the cancer resistance to treatments can be studied more easily. In particular, in the 737-dimensional space the biological phenomenon is easier in the sense that it is more linear than in the original space; while in the 90 or 20-dimensional spaces it is easier because the genes involved are few tens.

6 Conclusion

In this manuscript we tried to investigate the biological phenomenon by using gene-expression information through a linear mathematical model for classification. Taking advantage of the weight penalization, the shallow neural network has found a way to tackle the course of dimensionality by reducing its optimization space. We exploited this property in order to design an experiment to understand more deeply how genes interact with each other. The results are critically interpreted in order to retrieve insights about the underlying phenomena. The linear nature of the classifier allowed us to make some considerations concerning the kind of interactions among genes. In particular we identified some sub-spaces of the data set where the biological phenomenon is much easier to understand and analyze because the superposition principle holds or the number of features is considerably restricted.

References

[1] Hidalgo M. et al. “Patient-derived Xenograft models: An emerging platform for translational cancer research”. In: *Cancer Discov.* 4 (2014), pp. 998–1013.

[2] Tentler J. J. et al. “Patient-derived tumour xenografts as models for oncology drug development”. In: *Nat. Rev. Clin. Oncol.* 9 (2012), pp. 338–350.

[3] Byrne A. T. et al. “Interrogating open issues in cancer precision medicine with patient derived xenografts”. In: *Nat. Rev. Cancer* (2017). DOI: `10.1038/nrc.2016.140`.

[4] Bertotti A. et al. “A molecularly annotated platform of patient- derived xenografts (’xenopatients’) identifies HER2 as an effective therapeutic target in cetuximab-resistant colorectal cancer”. In: *Cancer Discov.* 1 (2011), pp. 508–523.

[5] Zanella E. R. et al. "IGF2 is an actionable target that identifies a distinct subpopulation of colorectal cancer patients with marginal response to anti-EGFR therapies". In: *Science translational medicine* (2015). DOI: 10.1126/scitranslmed.3010445.

[6] Bertotti A. et al. "The genomic landscape of response to EGFR blockade in colorectal cancer". In: *Nature* 526 (2015), pp. 263–267.

[7] Sartore Bianchi A. et al. "Dual-targeted therapy with trastuzumab and lapatinib in treatment-refractory, KRAS codon 12/13 wild-type, HER2-positive metastatic colorectal cancer (HERACLES): a proof-of-concept, multicentre, open-label, phase 2 trial". In: *Lancet Oncol.* 17 (2016), pp. 738–746.

[8] "Illumina. Array-based gene expression analysis. Data Sheet Gene Expr." 2011. URL: http://res.illumina.com/documents/products/datasheets/datasheet_gene_exp_analysis.pdf.

[9] Isella C. et al. "Selective analysis of cancer-cell intrinsic transcriptional traits defines novel clinically relevant subtypes of colorectal cancer". In: *Nature Genetics* 8 (2017). DOI: 10.1038/ncomms15107.

[10] Barbiero P., Bertotti A., Ciravegna G., Cirrincione G., Pasero E., and Piccolo E. "Supervised Gene Identification in Colorectal Cancer". In: *Quantifying and Processing Biomedical and Behavioral Signals.* Springer International Publishing, 2018. ISBN: 9783319950945. DOI: 10.1007/978-3-319-95095-2_21.

[11] Barbiero P., Bertotti A., Ciravegna G., Cirrincione G., and Piccolo E. "DNA microarray classification: Evolutionary Optimization of Neural Network Hyperparameters". In: *Italian Workshop on Neural Networks (WIRN 2018).* (Vietri Sul Mare, Italy). June 2018.

[12] Widrow B. and Lehr M. A. "Artificial Neural Networks of the Perceptron, Madaline, and Backpropagation Family". In: *Neurobionics* (1993). DOI: 10.1016/B978-0-444-89958-3.50013-9.

[13] Haykin S. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 1998. ISBN: 0132733501.

[14] François Chollet et al. *Keras.* https://keras.io. 2015.

[15] Ng A. Y. "Feature selection, L_1 vs. L_2 regularization, and rotational invariance". In: *International Conference on Machine Learning.* 2004. DOI: 10.1145/1015330.1015435.

[16] Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics, 2009. ISBN: 0387848576.

[17] Kingma D. P. and Ba J. "Adam: A Method for Stochastic Optimization. *International Conference for Learning Representations*". In: (2017). URL: arXiv:1412.6980v9.

[18] Schmidt M., Le Roux N., and Bach F. "Minimizing Finite Sums with the Stochastic Average Gradient". In: *Mathematical Programming* (2013). DOI: 10.1007/s10107-016-1030-6.

[19] Defazio A., Bach F., and Lacoste-Julien S. "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In: *Advances in Neural Information Processing Systems* (2014). URL: `https://arxiv.org/abs/1407.0202`.