NEURAL NETWORK MODELLING FOR BREAST CANCER SUBTYPES PREDICTION USING RNA-Seq DATASET

VICARI Celia, GHEZAIEL Morad - M2 Bioinformatique DLAD

Context

Nowadays, breast cancer is the most prevalent cancer in women all around the world. Depending on the subtype, invasive tumorigenesis (ductile carcinoma) and metastatic phenotype can lead respectively to breast ablation and death. In that way, effective early diagnosis may contribute to survival rate improvement. Currently, breast cancer subtypes are classified with the mean of histological datas, immunophenotyping of estrogen, progesterone and HER receptor expression[1]. Even if histopathological observations are the main indicators of the cancer class[2], various subtypes are still poorly characterized and gene expression analysis appears to be relevant for subtype classification and discovery[3]. In this study, we tried to take advantage of the publicly available breast cancer RNA-Seq dataset (n=801) to build a neural network-based model for subtype prediction. First, statistical based feature selection permits us to reduce the number of feature from more than 20000 genes to 15. During the cross-validation step performed using 33% of the shuffled training set, the model has reached 98 % accuracy with a "quasi exponential" loss convergence (loss<0.01 at 50 epochs).

1. Material and methods

Human breast cancer dataset containing transcriptome profiling (n = 20532 genes) of 801 individuals were pulled from the UCI Donald Bren School of Information and Computer Sciences website. Subtypes are not uniformly distributed in the dataset with frequencies following approximately the naturally observed occurrence, from highly represented (BRCA = 37 %) , common (KIRC, PRAD, LUAD = 18% each) to less represented ones (COAD = 10%). Learning processes were applied using a python script available here. This script takes advantages of the Keras deep-learning library (Tensorflow backend) to perform dataset preprocessing, model initialization, training, and validation step. Datas in csv format were embedded in a pandas Dataframe. Basic math tools and statistical tests were provided by numpy and the sci-kit learn package respectively. Finally, cross validation and comparative plots were drawn using the matplotlib library. Neural network topology and parameters are presented below part (see Topology and parameter tuning)

2. Feature selection

The dataset containing more than 20000 genes, and giving the fact that human genome is composed of nearly 30000 genes, we emitted the hypothesis that a significative number of these are not important to characterize the 5 cancer subtypes. In a statistical approach, we decide to plot the histogram of feature variance (figure 1 first panel). As expected, variances values spread from 0 to 40 with a high number of features displaying low variance ranging from 0 to 1 (50% of all features). In that way, we chose to filter our data according to a variance threshold of 1 leaving only 9180 features in the dataset. However, this

number of features is still high and not all of these genes can be considered as relevant transcriptomic cancer signature. To increase dimensionality reduction, we chose to apply a Chi2 independence test on our data. Based on the null hypothesis and similarly to pearson R^2 coefficient, this test computes an independence score for each features depending on sample labels. According to this score, it removes all but the k-highest scoring features from the dataset. Assuming that features with high variances are important to explain cancer subtypes signature, we chose to plot the mean of each feature variances for k ranging from 9180 to 5. Results (figure 1 second panel) show a high increase of the feature variances mean for low k values. Consistently with our previous observations, these results prompted us to only keep the 15 highest scoring features for the modelling step.

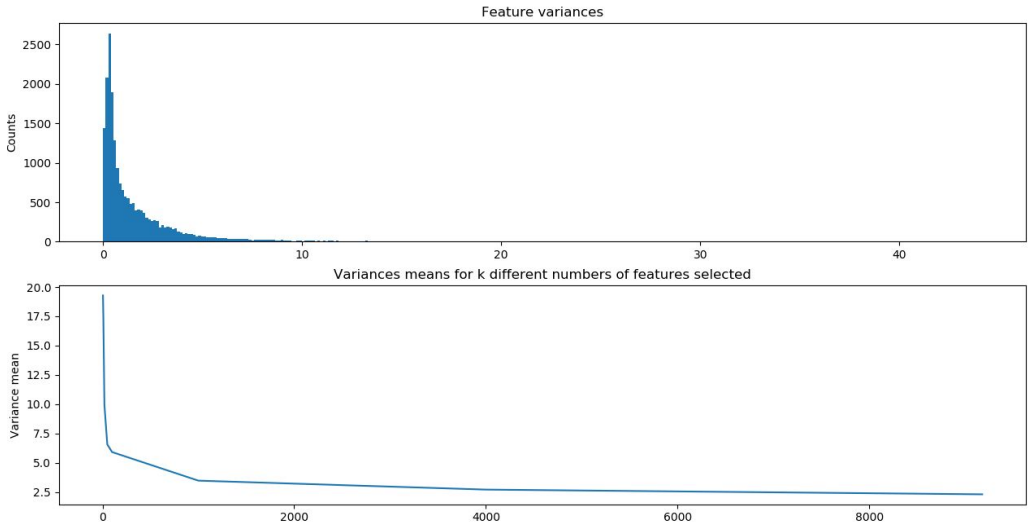
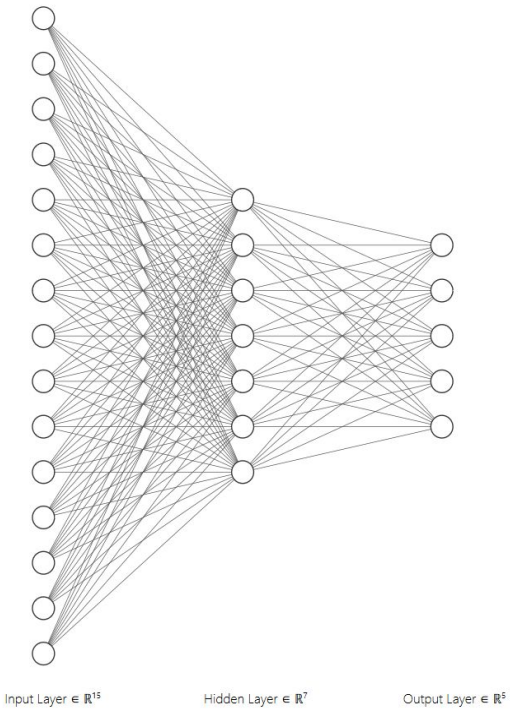


Fig1: First panel: Histogram of feature variance in the whole dataset. Second panel: Variances means from k best features ranging from 9000 to 5.

3. Topology



Compared to most of machine learning algorithms, neural networks offer a relatively high degree of freedom depending on their architectures. Giving the low number of features we obtained from the features selection step, we chose to build a fully connected feedforward shallow neural network (i.e: composed of a single hidden layer). The input layer is made up of 15 input nodes ,7 hidden units and 5 for the output. Dealing with a classification problem, we chose to use a softmax activation function for the output layer. The choice of the objective function, activation one and optimizer are discussed below (see Parameter tuning)

Fig2: Neural network topology used for breast cancer subtype prediction

4. Parameter tuning

1. Objective fonction

For error calculation and according to our classification problem, we chose to use the binary-cross entropy loss :

$$CE = -t_1 log(f(s_1)) + (1 - t_1) log(1 - f(s_1))$$

For C = 5 classes, binary-cross entropy computes the sum of the above formula from C = 0 to 4, with s1 = the input value, t1 the binary prediction, and f the sigmoid function. The choice of this loss was highly motivated by its documented use in multiclass problems [4].

2. Activations

Activation functions are mathematical links joining layers together. The choice of this function may be critical to model efficiency depending on the data, the type of node, and more generally on the network architecture. More generally, activations can be classified in two types : linear and non-linear functions. In order to choose the best activation for the hidden layer, we have proceeded to a cross-validation step using various known activations. Indeed, 33% of the shuffled dataset were assigned as a validation set at the end of each epoch (n = 100), the other 66% being used for training. This procedure was done using 7 known optimizer algorithms with their default parameters.

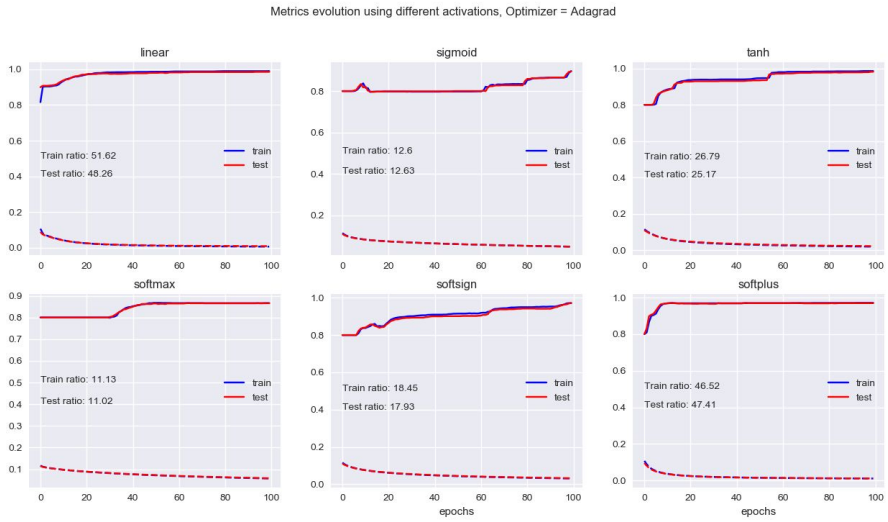


Fig3: Cross validation step for 6 known activation functions. Adagrad optimizer was used for each, with default learning rate (lr=0.01). For each epoch, 33% of the shuffled dataset was used as a validation set. Accuracy and loss curves were plotted and the ratio between their AUC (area under the curved) was calculated (Acc/Loss). Batch size =32. Results from other optimizers are obtainable by running activation comparaison pipeline(see output folder).

We chose to calculate for each activation an Area Under Curve (AUC) ratio between accuracy and loss (Accuracy / Loss) for training and testing step. As we can see, 2 activations are displaying a high training and testing AUC ratio compared to others : linear, and softplus functions. The fact that both display a similar profile can be understood by the fact that softplus is a quasi linear function for high input values higher than 1. On the contrary, non linear units such as sigmoid and hyperbolic tangent display either low ratios or unexpected behaviour. These observations are consistent with previous studies on deep learning modelization evoking the vanishing gradient problem of non linear units[5]. Indeed, non linear units use to map input values into a relatively small range ([0,1]), leading to non efficient update of coefficients during backpropagation. In those case, small variations between input values are not effectively mapped in the [0,1] interval. As a consequence, these results show that input features and the 5 breast cancer subtypes are related in a very subtle way. Finally, the softplus function seems to be the best choice. Its test ratio is higher than its train one, leading us to think that increasing the number of samples could improve its generalization ability. Giving this non overfitting behaviour, we chose to use softplus activation for the optimizer comparison part.

3. Optimizers

Optimizers are algorithms that search to optimize a fully differentiable loss function. Various kind of optimizers can be used depending on the problem, however they all share the same way to optimize the loss function: the gradient descent. In this method, we look for the deepest error value by the mean of partial derivative calculus. For this step, particular parameters as the learning rate are critical for effective optimization. When a too high learning rate is set it can lead to divergence from the minimum of the valley, a too low one can cause the error to never converge. To gain more insight into which optimizer we should use for the training of our network, we chose to calculate areas under the training and testing accuracy curves after having trained our model with various known optimizers. These measurements were done for learning rates ranging from 0.01 to 0.1 with a step of 0.01 using the softplus activation. To get a better sight into the results, we chose to draw heatmaps for training and test values.

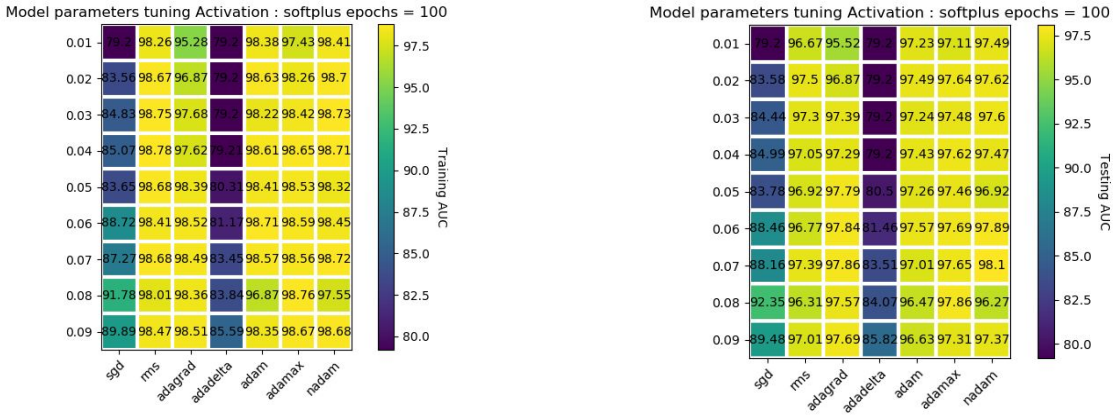


Fig4: Heat maps of the compared training (first heat map) or testing (second heat map) accuracy AUC using 7 optimizers and 10 learning rates ranging from 0.01 to 0.1. Softplus activation was used for the hidden layer. At each epoch, 33% of the shuffled dataset was used as a validation set. Accuracy AUCs (area under the curved) were calculated for training and validation. Batch size = 32.

According to the results in figure 5, stochastic gradient descent optimizer and adadelta display the worst AUCs compared to others. In addition, both do not display significative differences between AUC values for the train step and the test step. In the same way, even if adam, adamax and nadam show the highest AUC values, no relevant differences are present between the two steps. However, interesting observations can be done on Adagrad. Indeed, its test AUC values are higher than its train ones for a learning rate set to 0.01, equals if set to 0.02. However, the ratio is inverted for alpha > 0.02 prompting us to think that choosing an alpha of 0.01 may permit to keep two important abilities:

1) being enrichable with new informations
2) avoid overfitting by overlearning.

In that way, in order to preserve the above aspects, we chose to keep the adagrad optimizer with a learning rate set to 0.01.

5. Cross validation

Giving results from the previous parts, we chose to keep the following configuration for our model : softplus activation function for the hidden layer, and Adagrad (alpha=0.01) for the optimizer. Batch size was setted to 64 and number of epoch to 100. Confusion matrix was calculated at the end of the 100 epochs. As expected, the overall prediction seems good with quasi perfect specificity and sensitivity.

Predictions	BRCA	KIRC	PRAD	LUAD	COAD
Labels					
BRCA	98	0	0	0	0
KIRC	0	47	0	0	0
PRAD	0	0	45	0	0
LUAD	0	0	0	43	0
COAD	1	0	0	0	28

Fig5: Cross table representing the number of right predictions and wrong ones among test dataset.

Learning curves for accuracy and losses are drawn figure 7. As it is shown, the error value converge fastly to 0 at 80 epochs. Training accuracy appears higher than test one but do not tend to 100%, keeping a stable behaviour for n epoch > 60.

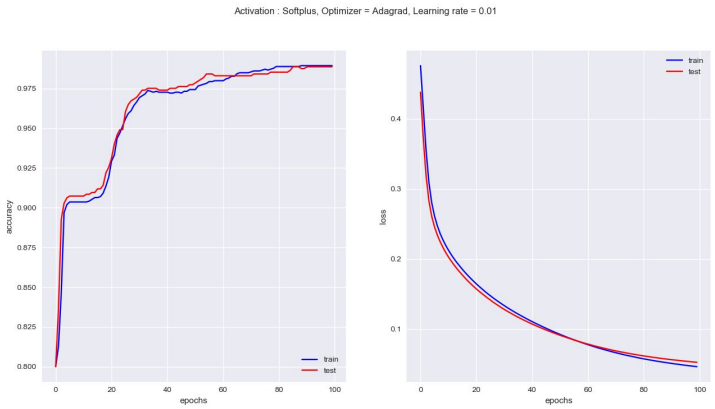


Fig6: Learning curve for the trained model. The left curves represent train and test accuracies, the right curves represent train and test losses.

6. Discussion

In this supervised approach, we developed a neural network that is able to classify samples giving 15 RNA seq features. During the feature selection process, we chose to take advantage of a chi2 test of independence to select 15 features. However, this step could be improved by comparing these results with those from a pearson correlation test. Concerning parameter tuning, we observed that the use of a softplus activation for the hidden layer worked better than others :

$$f(x) = \log(1 + e^x),$$

Softplus activation is linear for x>2, projecting input values in a smaller interval. Consistenly, plots of the feature means before and after softplus activation (figure 7) show that softplus activation increases the feature mean variability making them more discriminative for the classification task.

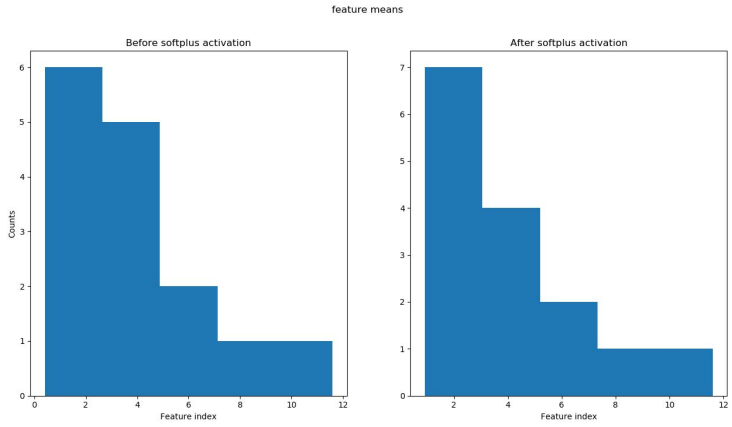


Fig6: Learning curve for the trained model. The left curves represent train and test accuracies, the right curves represent train and test losses.

In addition, we found that Adagrad optimizer performed well on our model with a learning rate of 0.01. This result is consistent with studies stating that it works well on low learning rate and sparse datas. However, results for SGD and Adadelta optimizers were not fully investigated.
Finally, results from the final cross validation step have shown a really good specificity and sensitivity, making the choice of the softplus activation and the Adagrad optimizer well suited for our model.
To put it in a nutshell, neural network technology seems to be an incredible tool for medical diagnosis using RNA seq datas . Additionally, recent studies have shown really good model performance for multiple cancer classification using histological datas [6]. An idea could be to build a model that take account of RNA seq features and histological datas. Consistently with the concept of black box concerning neural network architecture, building such a model could help us to decipher hidden links between differential expression datas and histological phenotypes. In that way, several approaches that have not been described in this study are currently in development concerning medical interpretation of neural networks[7].

6. References

1. Wesolowski, Robert and Bhuvaneswari Ramaswamy. “Gene expression profiling: changing face of breast cancer classification and management” *Gene expression* vol. 15,3 (2018): 105-15

2. Malhotra, Gautam K et al. “Histological, molecular and functional subtypes of breast cancers” *Cancer biology & therapy* vol. 10,10 (2010): 955-60.

3. Zhang, Yu-Hang et al. “Identifying and analyzing different cancer subtypes using RNA-seq data of blood platelets” *Oncotarget* vol. 8,50 87494-87511. 15 Sep. 2017

4. https://gombru.github.io/2018/05/23/cross_entropy_loss/

5. https://blog.paperspace.com/vanishing-gradients-activation-function/

6. Sepandi, Mojtaba et al. “Assessing Breast Cancer Risk with an Artificial Neural Network” *Asian Pacific journal of cancer prevention : APJCP* vol. 19,4 1017-1019. doi:10.22034/APJCP.2018.19.4.1017

7. Zhang, Zhongheng et al. “Opening the black box of neural networks: methods for interpreting neural network models in clinical applications” *Annals of translational medicine* vol. 6,11 (2018): 216.