ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA INFORMÁTICA

Ingeniería de la Salud

Aprendizaje profundo aplicado a la bioinformática
Deep learning for bioinformatics

Realizado por
Guillermo López García

Tutorizado por
José Manual Jerez Aragonés
Cotutorizado por
Francisco Javier Veredas Navarro

Departamento
Lenguajes y Ciencias de la Computación,
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2018

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DE LA SALUD

Aprendizaje profundo aplicado a la bioinformática
Deep learning for bioinformatics

Realizado por

Guillermo López García

Tutorizado por

José Manual Jerez Aragonés

Cotutorizado por

Francisco Javier Veredas Navarro

Departamento

Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2018

Fecha defensa:

El Secretario del Tribunal

Resumen:

Actualmente, el aprendizaje profundo (*deep learning*) constituye una de las tecnologías del campo de la Inteligencia Artificial (IA) que goza de mayor éxito y popularidad. En campos como el procesamiento de imágenes y el análisis de datos secuenciales, su uso se encuentra bastante extendido, formando parte del núcleo de sistemas de vanguardia como los vehículos de conducción automática o los sistemas de reconocimiento facial. Sin embargo, y a pesar de sus grandes capacidades representacionales y predictivas, su aplicación a problemas, como el análisis de datos de expresión para su empleo en tareas de clasificación de cáncer, en los que el número de variables (N) supera con creces el número de muestras (M) o patrones del conjunto de datos ($N \gg M$), constituye un verdadero reto todavía sin resolver. Con el objetivo de resolver este problema entre el número de variables y de muestras, diferentes ténicas de aprendizaje automático de reducción de la dimensionalidad de los datos han sido aplicadas. Aunque esta técnicas consiguen reducir el número de variables, el rendimiento en predicción de los modelos de aprendizaje autoemático tradicionales es moderado, ya que el número reducido de muestras empleado para el entrenamiento de los métodos de reducción de la dimensionalidad no les permite extraer las características adecuadas para mejorar el rendimiento en predicción de forma significativa. Para resolver estos problemas y mejorar la habilidad predictiva de los métodos clásicos de aprendizaje automático, proponemos un enfoque basado en el aprendizaje profundo para reducir la dimensionalidad de los datos de expresión, que emplea aprendizaje supervisado y no supervisado para hacer uso de todas las muestras de tumores presentes en una base de datos para resolver una tarea de clasificación en cáncer concreta. Empleando la predicción del subtipo intrínseco de cáncer de mama como ejemplo de tarea de clasificación en cáncer, los resultados obtenidos muestran que el rendimiento de los enfoques basados en aprendizaje profundo y en técnicas tradicionales de aprendizaje automático son muy similares a la hora de reducir la dimensionalidad de los datos de expresión génica para su empleo en tareas de clasificación en cáncer. Sin embargo, aunque algunos enfoques tradicionales parecen superar el rendimeinto del enfoque basado en aprendizaje profundo, para concluir cuál es el enfoque más efectivo más trabajo es necesario. Por otro lado, comparando el rendimiento de los diferentes modelos de aprendizaje profundo implementados, aunque con mucha prudencia, podemos decir que cuanto mas profundo el modelo mejor rendimiento obtuvo, mostrando el poder representacional

de estos modelos para la extracción de una jerarquía de representaciones abstractas utiles para la resolucion de tareas de clasificación.

Palabras claves: aprendizaje profundo, aprendizaje automático, inteligencia artificial, datos de expresión génica, reducción de la dimensionalidad, cáncer de mama, calsificación

---

Abstract:

Deep learning has become one of the most promising Artificial Intelligence (AI) technologies nowadays. It has been very successfully applied to areas such as computer vision or natural language processing. However, although the great representational and predictive capabilities exhibited by these models, their feasibility to be applied to problems such as gene expression data analysis for cancer classification, in which the number of input variables (N) far exceeds the number (M) of samples ($N \gg M$), remains a challenge yet to be solved. In order to solve this balancing problem, several traditional machine learning dimensioanlity reduction techniques have been applied. Although these techniques scale down the input feature space, the prediction performance of the traditional machine-learning models is moderate, as the reduced number of samples used to train both the dimensionality reduction methods and the classifiers does not allow them to extract the hidden patterns in the gene expression data in a way that improves the prediction performance significantly. In order to solve these problems and improve the prediction ability of the traditional machine-learning models, we propose a deep learning approach for reducing the dimensionality of gene expression data, which uses both unsupervised and supervised learning to make the most of the entire tumor data available in a database to solve a concrete cancer classification task. Using breast cancer intrinsic subtype prediction as an example of a cancer clasification task, the obtained results showed that the performances of both deep learning and traditional machine-learning approaches are very similar when reducing the dimensionality of gene expression data for the purpose of cancer classification. However, though some traditional approaches seem to outperform the deep

learning strategy, to conclude which is the most effective approach more work needs to be done. On the other hand, comparing the performance of the different autoencoders, although very cautiously, we could say that the deeper the model the better perfomance was obtained, showing the representational power of deep learning to extract a hierarchy of abstract representations useful for solving classification tasks.

Keywords: deep learning, machine-learning, artificial intelligence, gene expression data, dimensionality reduction, breast cancer, classification

Contents

1 Introduction

1.1 Motivation

Artificial Intelligence (AI) is changing our society in an unprecedented way, a fact no one can deny. Shopping, driving, web searching, cooking, nearly all human activities can be computationally automated, and that has made AI one of the most successful areas in Computer Science. This is evidenced by the great amount of resources the technology giants, such as Google, Facebook or Amazon, are investing in this field, with some experts claiming that we are now at the Fourth Industrial Revolution [1].

Driving all this progress, a state-of-the-art Machine Learning technique stands out from the others, and that is deep learning. This technology has reached astonishing performance in domains where data has spatial or sequential information, such as computer vision or natural language processing [2]. This way, the image processing area has been revolutionized by a deep learning architecture called Convolutional Neural Network (CNN) [3]. Since 2012, all winning models of the ImageNet contest, a competition where the cutting-edge algorithms in visual recognition tasks meet annually, have been based on CNN architecture. By 2015, human-level performance was reported to be exceeded in several object recognition tasks from the contest [4]. This has allowed deep learning models to be at the core of some of the most advanced artificial visual perception systems, such as the ones used in self-driving cars [5].

Another area to which deep learning has contributed the most is natural language processing. Recurrent Neural Networks (RNNs) are specially suited for tasks where data is presented as a sequence of elements, as they are able to integrate the context where an element is presented to the network in its internal representation. Thus, RNNs have been widely applied to language translation and speech recognition tasks [6, 7]. When combined with CNN models, they can be used for image captioning, the process of generating a natural language description of an image, in the way we see in Figure 1.

Up to now, we have described deep learning models that solve predictive tasks. However, in recent years, new deep generative models have been created, which are able to learn the true data distribution from a training set in order to generate new data. In 2014,

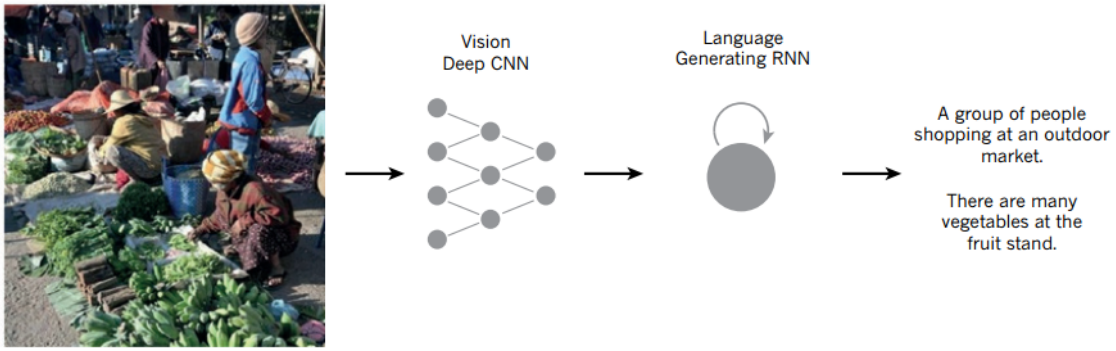

Figure 1: Natural language description produced by an RNN model taking as input the representations extracted by a CNN, taken from [2].

Ian Goodfellow *et al.* presented Generative Adversarial Networks (GANs) [8], a model composed of two deep neural networks, a discriminator and a generator, which compete with each other through a process called adversarial training. One of the most relevant applications of GANs is the production of photorealistic high-resolution images, such as the ones showed in Figure 2. Once trained, the generator network is able to create these realistic images receiving just noise as input data. This has strong implications in many other fields like Philosophy, Psychology or Art, as the ability to be creative may no longer be attributed only to humans.

Figure 2: High-resolution images generated by GANs, taken from [9].

There are three main reasons why deep learning models have achieved all the success mentioned above. Firstly, the theoretical model behind this technology: Representation learning, a set of methods that allow a machine to be fed with raw data and to automatically discover the representations needed for detection or classification tasks [2]. The key is that they do it in a data-driven way: The transformations needed to solve a certain task are learnt from the raw data itself, using a general-purpose learning algorithm, avoiding the manual feature extraction performed by humans. Deep learning is a group of representation-learning models, and the biologically inspired neural architecture of these models allows them to stack many layers forming what is called a deep architecture. Using this architecture, the model processes the data transforming its raw representation into a hierarchy of progressively more abstract representations, which makes the model very good at discovering intricate structure in high-dimensional data [10]. However, the main drawback is having a very complex model with so many parameters to be learnt, which can in turn cause problems such as overfitting and vanishing gradient problem. In recent years, several technical improvements, such as dropout, batch-normalization or rectified linear units [11, 12, 13], have contributed to overcome these difficulties.

As we have just said, as a model gets deeper, the number of parameters to be learnt increases enormously, needing huge amount of data to train the model in an effective way. Nevertheless, when having enough data, in contrast to traditional machine-learning algorithms, deep learning models show scalable properties that allow them to make the most of the data in terms of performance, i.e. they increase their performance as the amount of training data increases. Hence, the second engine driving deep learning to success is Big Data [14]. In the digital era we live in, nearly all human and non-human activities are monitored by sensors that collect vast amounts of data. The combination of huge datasets and methods such as deep learning able to handle them, promises to extract all the information hidden in the data [15], turning two things that separately are almost useless into a useful symbiosis.

Finally, the last reason why deep learning is such a great success is the development of modern and more capable hardware, specially GPUs. The theory behind deep learning is not new, actually the first experiments with artificial neural networks were conducted in the 1940s [16]. Even so, the field remained unpopular until faster CPUs and general

purpose GPUs were made available. Graphical Processing Units (GPUs) are specially suited for the optimization of the matrix operations needed to train deep models, which is the reason why they have become so popular in the field, performing the training 20 times faster than CPUs [17].

In what follows, we shall proceed to enumerate the main contributions of these models to the field of bioinformatics, paying special attention to gene expression analysis, the area to which we aim to apply deep learning in this work.

1.2 Deep learning in bioinformatics

The contributions of deep learning to the bioinformatics domain can be seen in two main areas: biological image analysis, and omics data processing.

Because of the great success deep learning models have obtained in the image processing domain, one of the most straightforward applications of deep learning in bioinformatics is the analysis of biological images. In 2012, Ciresan *et al* used a deep convolutional model to segment neuronal membranes in electron microscopy images, classifying each pixel as membrane or non-membrane [18, 19]. These authors used the same model in 2013 to detect mitosis in breast histology images [20]. An interesting work by Zhang *et al* was pusblished in 2015, in which they showed that a convolutional model pre-trained on natural images from ImageNet could be further fine-tuned using *in situ* hybridization images to improve the prediction of *Drosophila melanogaster* developmental stages [19, 21]. This is an example of what is called Transfer Learning (TL), a machine-learning technique in which a model pre-trained on a *base* dataset and task is further adjusted using a *target* dataset to be used to solve a *target* task (notice that *base* and *target* refer to different datasets and tasks). This strategy has been widely and successfully applied to solve image processing tasks using deep learning models [22].

The second area where deep learning has been applied in the bioinformatics domain is omics data analysis, one of the most promising areas in bioinformatics nowadays. The Next Generation Sequencing (NGS) methods are revolutionizing biology, generating an unprecedented vast amount of genomic data that, jointly with other molecular data,

is analyzed by the omics disciplines, such as genomics, transcriptomics, proteomics or epigenomics. The integration of all this data from a single organism can be very valuable in fields such as medicine, where this data is used to get a holistic view of the molecular state of a patient, driving the progress of what is called P4 medicine (predictive, preventive, personalized and participatory), considered by the experts to be the medicine of the future [23].

The contributions of deep learning to this domain are mainly found in genomics (a key area in P4 medicine), due to the adaptation of the convolutional models applied in computer vision to DNA sequence data. Instead of processing 2-D images with three color channels, a DNA sequence is considered as a 1-D sequence with four channels, one for each type of nucleotide (A, C, T, G) [24]. In 2015, this approach was used by Alipanahi *et al* to find useful motifs in DNA sequences to predict sequence specificities of DNA- and RNA-binding proteins [25]. One year later, the same convolutional strategy was used by Zhou and Troyanskaya to find effective motifs for predicting the effects of non-coding variants in DNA sequences [26].

Apart from genomics, gene-expression data analysis (transcriptomics) is becoming one of the most important omics disciplines in P4 medicine, due to the advent of high-throughput sequencing technologies such as RNA-Seq [27]. In areas such as oncology, gene expression data offers a completely new way of describing the molecular state of a patient. As cancer is considered a genetic disease, a gene expression sample from a patient (which describes the genetic changes responsible for the progression of the disease, such as the over-activity or the repression of genes) contains information of paramount importance for the prevention, diagnosis and treatment of this malignant disease. For example, in breast cancer (one of the most heterogeneous cancers with many intrinsic subtypes) the information hidden in the gene expression data, when properly extracted, can be used to diagnose the concrete subtype in a precise and effective way [28]. An accurate diagnosis is extremely important for the development of a personalized treatment, as the molecular and specific therapies as well as the predicted prognosis strongly depend on the intrinsic breast cancer subtype of the patient [29].

The contributions of deep learning to gene expression analysis for cancer prediction are extremely scarce. The reason being that, although deep learning models have demon-

strated to be able to extract the hidden patterns in extremely complex data, gene expression data present some problems that make the application of deep learning models a difficult challenge yet to be solved. Up to now, in all the successful deep learning applications we have mentioned, the data had spatial or local information (text sequences, images, biological sequences, etc.); however, this is not the case with gene expression data. To make matters worse, the dimensionality of the input feature space, i.e. the number of input features (N), is extremely high (10K-60K) in gene expression datasets. However, in clinical tasks such as cancer detection, the number of available samples (M) is very low (300-1K). This enormous imbalance between the number of input features and the number of available samples ($N \gg M$) makes the learning process extremely difficult, and it is known as the *curse of dimensionality* [30], a common problem not only for deep learning models, but for traditional machine-learning algorithms as well.

In fact, in order to solve different cancer classification tasks using gene expression data, various traditional machine-learning models have been applied, such as logistic regression, decision trees, support vector machines, swallow artificial neural networks, etc. [31, 32]. But again, the main problem faced by these algorithms is the high dimensionality of the gene expression data compared to the lack of available samples ($N \gg M$). To reduce the number of input features, distinct classic dimensionality reduction techniques have been used, such as feature selection and extraction methods [33]. Although these techniques scale down the input feature space, the prediction performance of the traditional machine-learning models is moderate, as the reduced number of samples used to train both the dimensionality reduction methods and the classifiers does not allow them to extract the hidden patterns in the gene expression data in a way that improves the prediction performance significantly. This is mainly due to a scalability problem, as, for example, when using traditional machine-learning methods to predict the intrinsic breast cancer subtype from gene expression data, these supervised models cannot take advantage of any other tumor data but breast cancer, using only a few hundred samples to train the models.

In order to solve these problems and improve the prediction ability of the traditional machine-learning models, we propose a deep learning approach for reducing the dimensionality of gene expression data, which uses both unsupervised and supervised learning to make the most of the entire tumor data available in a database to solve a concrete cancer

classification task, such as predicting the intrinsic breast cancer subtype of a patient.

Now, we shall proceed to review the state of the art in deep learning for cancer detection using gene expression data.

1.2.1 Deep learning for gene expression data analysis

Although, as it was said before, the contributions of deep learning to cancer prediction using gene expression data are just starting to emerge and there are not yet numerous examples, several works are worth mentioning, as they clarify how unsupervised deep learning models (essentially autoencoders) can be adapted to reduce the dimensionality of gene expression data for the purpose of cancer classification. One of the most inspiring and cited ones was done in 2013 by Fakoor *et al* [34]. In this work, they used a combination of PCA and simple autoencoder architectures (sparse and two-layers stacked autoencoders) to perform dimensionality reduction, and a softmax output layer on top of the autoencoder architecture during the classification stage. Although the autoencoders are constrained by the features extracted by PCA, they use a very simple linear classifier and only 2K samples for training the deep models, this is the first work using deep autoencoders and gene expression data from different tumors during the feature learning step. In 2016, Danaee *et al* used stacked denoising autoencoders for feature extraction, and evaluated the extracted representations performing supervised breast cancer detection [35]. Even though such a deep model was trained using only 1K breast cancer samples, this was the first time stacked denoising autoencoders were applied to gene expression data. Finally, in 2018, Way and Greene employed variational autoencoders (VAEs) to extract a latent feature space using 10K gene expression samples [36]. Though they did not use the extracted features to perform any cancer detection task, they showed that the extracted features represented biological signals.

In this final project, basing on these previous works, we will try to use deep autoencoders to reduce the dimensionality of the gene expression data, as well as a transfer learning approach that allows us to train the deep learning models using a whole database of tumor samples, and use the extracted features to perform cancer classification tasks such as the prediction of breast cancer intrinsic subtypes.

1.3 Objectives

Thus, the main objective of this project is to use a deep learning approach to reduce the dimensionality of gene expression data, as well as analyzing its effectiveness when applied to cancer classification tasks.

This principal objective can be divided into two more specific objectives:

1. Adapting deep learning models to gene expression data particular difficulties. In order to solve the great imbalance between the dimensionality of the data (N) and the number of samples ($N \gg M$), we will try different autoencoders architectures (sparse, stacked sparse and denoising stacked) to reduce the number of input features. In addition, we will use a transfer learning approach to increase the number of samples used during the feature extraction stage.

2. Comparing the obtained results using a deep learning approach with the ones obtained using traditional machine-learning dimensionality reduction techniques. In order to compare both approaches, we will use the extracted features to solve a cancer classification task, such as the prediction of the breast cancer intrinsic subtypes. To do that, we will use three classic supervised machine-learning algorithms: Logistic regression, support vector machines and swallow artificial neural networks.

1.4 Document structure

In this section, we give a brief description of the structure of the document:

- Methods: This section describes each of the phases of a traditional data-mining metodology followed to carry out this project, such as data extraction, data preprocessing, dimesnionality reduction, etc. In addition, it contains a description of the algorithms used to perform all these stages.

- Results: Here, the project final results are presented and discussed. Special attention will be paid to the comparison of the results obtained using the deep learning approach and the traditional machine-learning strategy to reduce the dimensionality of the data.

- Conclusion: Finally, the last section contains a conclusion of the work both in English and Spanish, with a final subsection dedicated to describe future works and research lines.

2 Methods

In this project, we compare two different strategies of solving a cancer classification task using gene expression data, in our case the prediction of breast cancer intrinsic tumor subtypes. The two approaches reduce the high dimensionality of the data in a different way, one using deep learning algorithms and the other using traditional machine-learning techniques.

2.1 Deep learning strategy

The first strategy uses deep unsupervised learning models (autoencoders) to perform feature extraction. Using a transfer-learning approach, the models are pre-trained on a large compendium of gene expression samples, and then fine-tuned using a small dataset for classification purpose. Hence, we distinguish two phases: feature learning and classification learning.

2.1.1 Feature learning

Data extraction

To pre-train the deep models that reduce the dimensionality of the gene expression data, any of these public data sources can be used:

- The Cancer Genome Atlas (TCGA) platform is a collaboration between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI), that has generated one of the most complete genomic studies up to now, known as *PanCancer Atlas* [37]. This data contains muti-dimensional omics data (DNA methylation, gene and protein expression data, etc.) of 33 different tumor types from 11K patient samples.

- The UCSC Xena portal allows to access 1521 multi-omics datasets from 135 different cohorts. The largest dataset to which they provide access to is the TCGA-TARGET-GTEx dataset, a data integration from three different platforms into a

unique dataset free of computational batch effects [38]. It comprises 20K gene expression samples, from which almost the 50% of them come from cancer patients, and the other 50% from healthy (control) patients. This makes TCGA-TARGET-GTEx one of the largest and most balanced gene expression datasets, something specially useful when performing cancer detection tasks such as predicting whether a sample comes from a cancer or healthy patient.

- The Gene Expression Omnibus (GEO) is a public functional genomics data repository that stores and freely distributes microarray, next generation sequencing and many other high-throughput functional genomics data provided by the scientific community. Platforms such as $ARCSh^4$ [39], provide access to all the RNA-Seq gene expression data available in GEO, processing the data from different platforms uniformly. Concretely, 187,946 samples are accessible through $ARCSh^4$ with 103,083 mouse and 84,863 human.

Due to performance reasons, the dataset used in this work to pre-train the models is the *Pan-Cancer* gene expression dataset. Although our initial intention was to use the TCGA-TARGET-GTEx dataset, it contains almost twice (20K) the number of samples of the *Pan-Cancer* (11K), which makes it too large considering our hardware resources (see section).

Data preparation

The original *Pan-Cancer* dataset contains 11K samples and 60K variables (gene transcripts). However, our hardware resources cannot process so many input variables (see section). Hence, instead of using the original dataset, we used the data from [36] (henceforth called pan-cancer dataset), accessed through `https://github.com/greenelab/tybalt`. This dataset contains all the 11K samples from 33 tumor types but only includes the 5K most variably expressed genes, defined by median absolute deviation (MAD). In addition, the unit of the gene expression data is log2(FPKM + 1) transformed RSEM values.

Actually, rather than using the whole pan-cancer dataset for pre-training the models, we split the data into two distinct sub-datasets: one containing only the breast cancer tumor samples (BRCA pan-cancer dataset, 1K samples) and the other containing the re-

maining samples from the rest of the 32 tumor types (non-BRCA pan-cancer dataset, 10K samples). As in our transfer-learning approach the cancer classification task we want to solve is the prediction of the intrinsic breast cancer subtype, the BRCA pan-cancer data is only used during the classification learning phase, as it contains the subtypes information, whereas the non-BRCA pan-cancer dataset is used during the feature learning phase to pre-train the deep models in an unsupervised way. The rationale behind this is that we do want to use totally different data to perform the pre-training and the fine-tuning during classification learning phase.

On the other hand, regarding normalization, the non-BRCA pan-cancer data is normalised using the standard centering and scaling method (zero mean and unit variance).

Dimensionality reduction

To reduce the high dimensionality of the gene expression data we use autoencoders, an unsupervised feature extraction method.

- Theoretical model

An autoencoder, in its simplest form, is a feedforward neural network with three layers: an input, a hidden and an output layer. It is an unsupervised learning method in which the main goal is, given an input, to reconstruct an output layer representation as closely as possible to the initial input layer representation. This is done by training the network using backpropagation method to minimize the reconstruction error, a function that computes the difference between the input and the output.

For example, as it is shown in Figure 3, given a set of k unlabeled training samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$, where $x^{(i)} \in \Re^6$, an autoencoder tries to learn a function $\hat{x} \approx x$ [40]. The non-linear function that transforms the input into a hidden representation is called *encoder*, and can be expressed as $h(x) = f(Wx + b)$, where f is the hidden activation function, such as sigmoid or tanh, W is the hidden weight matrix and b is the bias vector of the hidden layer. The matrix W is of dimensions $n \times d$, where n is the dimension of the input data (number of units in the input layer), and d is the dimension of the encoded representations (number of hidden units). On the other hand, the non-linear function

that takes the hidden representations and transforms them into the reconstructed input representations is called *decoder*, and can be expressed as $\hat{x}(h) = g(W'h + b')$, where g is the output activation function, W' is the output weight matrix and b' is the bias vector of the output layer. As opposed to W, the matrix W' is of dimensions $d \times n$.

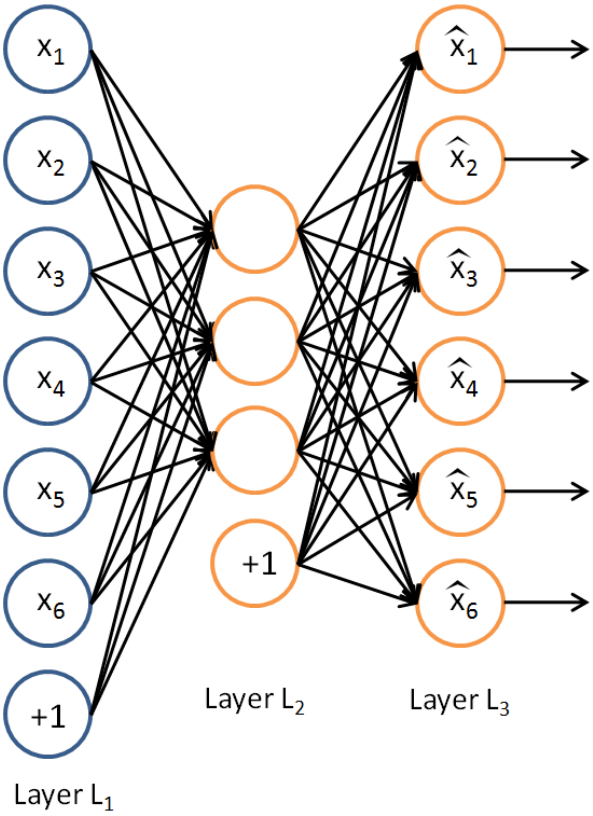

Figure 3: Example of a simple autoencoder architecture, taken from [40].

Having a hidden layer with fewer units than the input and the output layers ($d < n$), forces the autoencoder to compress the input representation into a lower dimensional representation, which can be reconstructed to its initial representation. That is why it is used as a dimensionality reduction method.

Constraining the network, such as using a small number of hidden units, has demonstrated to force the network to extract more abstract and meaningful features in the hidden representations. In addition to reduce the number of hidden units, another popular way of constraining the network is using what is called a *sparsity* penalty [40]. This penalty creates sparse representations, in which hidden units tend to be inactive most of the time, i.e. close to zero if the hidden activation function is the sigmoid or ReLU

function and close to -1 if it is the tanh activation function. Hence, the main effect of the sparse penalty is to favour the distributed hidden (encoded) representations and the units specialization, as each input pattern is encoded by the activation of a relatively small set of hidden neurons and each neuron responds (activates) to a small set of inputs. This penalty is generally implemented using L1-regularization in the hidden layer, which is added to the reconstruction error function. Hence, if the mean squared reconstruction error is used, the overall loss function minimized during the learning procedure can be expressed as:

$$\left[\frac{1}{m}\sum_{i}^{m}||x_i - \hat{x}_i||^2\right] + \lambda \sum_{j}^{n}\sum_{l}^{d}|w_{jl}| \tag{1}$$

where m is the batch size, n is the number of input and output units, d is the number of hidden units, w_{jl} is the weight connecting the input unit j to the hidden unit l and λ is the L1-regularizer penalty. The first term corresponds to the input reconstruction error, whereas the second term represents the L1-regularization, which tends to decrease the magnitude of the weights, acting as a *sparsity* constraint. The *sparsity* penalty is widely used in image processing domain, where it has shown to produce very good results [41].

Another widely used approach for constraining the network is known as *denoising* autoencoders [42]. During training, noise is added to the input data, and the difference between the input reconstruction and the original noiseless data is minimized using back-propagation. Hence, the goal of the network is to obtain a hidden representation robust to the introduction of noise in the input layer. In order to be able to reconstruct the input correctly, the corruption of the input data forces the network to extract more abstract and meaningful features in the hidden layer. A simple denoising autoencoder architecture can be seen in Figure 4.

Finally, the last approach used to force the network to extract more abstract features is *stacked* autoencoders. This strategy simply consists on "stacking" several autoencoders into a deep autoencoder model, as the one shown in Figure 5, which is the result of stacking two simple autoencoders, having a deep model of one input layer, two hidden encoder layers, and two decoder layers. The hidden abstract representation is always the

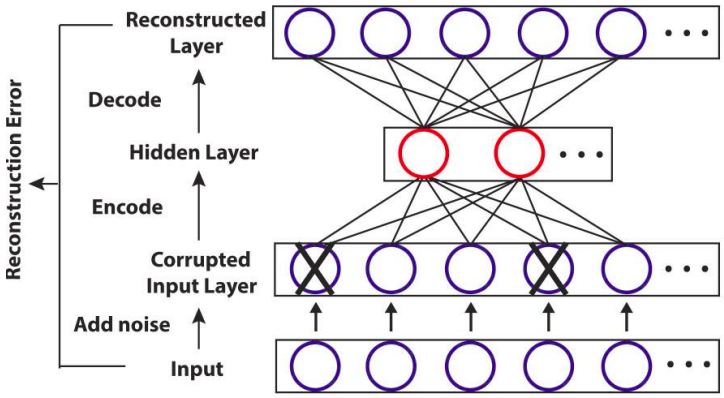

Figure 4: Example of a simple denoising autoencoder architecture, taken from [43].

one encoded by the middle layer, the last encoder layer.

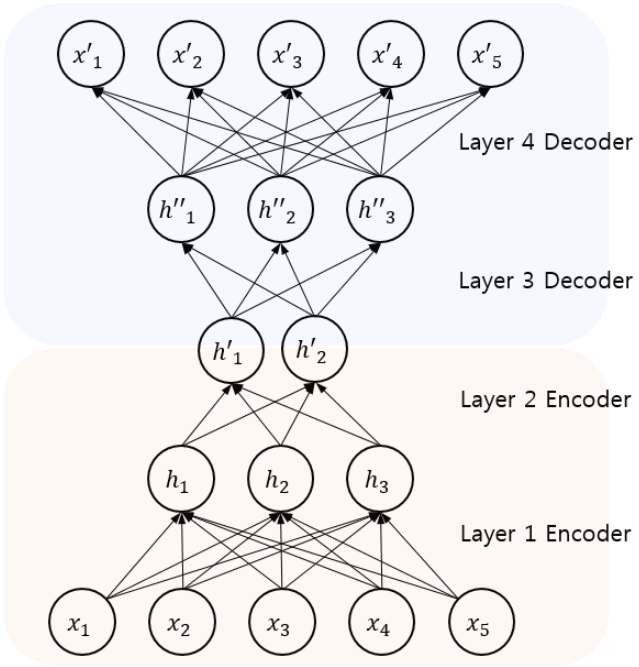

Figure 5: Example of *stacked* denoising autoencoder architecture, taken from [44].

Besides, different constraining approaches can be combined into one, such as sparse denoising autoencoders, stacked sparse autoencoders, or stacked denoising autoencoders. All these autoencoder models have been successfully applied in image processing domain, where they are able to transform an image into an encoded lower dimensional representation. However, although the application of a deep learning approach to reduce the dimensionality of the gene expression data has been explored in some previous works (see section 1.2.1), much works is still needed to be done to really know the feasibility of these

models when applied to gene expression data.

- Model implementation

In this work, we have implemented three different autoencoder architectures: sparse, stacked-sparse and stacked-sparse-denoising autoencoders. To train these models we used the non-BRCA pan-cancer dataset (see data preparation phase). Besides, for each architecture, we needed to tune several hyperparameters, such as L1 penalty term, number of epochs, batch size, etc. Though the best practice for tuning the hyper-parameters is to use a search method such as Grid-Search or Randomized-Search, this would have been too computationally intensive considering our hardware resources. For that reason, in order to tune the hyper-parameters of each model, we extracted the 10% of the samples of the non-BRCA pan-cancer dataset to define a validation set, and we used the remaining 90% of the samples as the training set. In this way, we tried different hyper-parameters configurations and examined both the training and the validation loss curves to select the best values, the ones that minimize both errors and prevent overfitting.

For the sparse architecture with one hidden layer, the hyper-parameters used in that layer and the values we tried are shown in Table 1. Batch Normalization is used in the hidden layer as a regularizer and to help to prevent vanishing gradient problems. The number of extracted features is 100 in all the implemented autoencoder models.

Hyper-parameter	**Possible values**
L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
Activation function	ReLU
Batch Normalization	Yes
Number of hidden units	100
Learning algorithm	Adam
Learning rate	$\{0.001, 0.005, 0.01\}$
Number of epochs	$\{40, 60, 80\}$
Batch size	$\{80, 100, 120\}$

Table 1: Hyper-parameter space of the sparse one-hidden layer autoencoder model.

Layer	Hyper-parameter	Possible values
Hidden one (encoder)	L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
	Activation function	ReLU
	Batch Normalization	Yes
	Number of hidden units	2000
Hidden two (encoder)	L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
	Activation function	ReLU
	Batch Normalization	Yes
	Number of hidden units	100
Hidden three (decoder)	L1 regularization	No
	Activation function	ReLU
	Batch Normalization	No
	Number of hidden units	2000
Whole model	Learning algorithm	Adam
	Learning rate	$\{0.0001, 0.0005, 0.001\}$
	Number of epochs	$\{100, 125, 150\}$
	Batch size	$\{100, 150, 200\}$

Table 2: Hyper-parameter space of the stacked-sparse autoencoder model.

The deep stacked-sparse architecture we used is the same as the shown in Figure 4, with one input layer, three hidden layers and one output layer. The hyper-parameters and their configurations are shown in Table 2.

Finally, the deep stacked-sparse-denoising architecture we used has two more layers than the previous model: one input layer, five hidden layers and one output layer, whose hyper-parameters and their possible values can be seen in Table 3. For corrupting the input data, we set the values of a random fixed proportion of genes (input variables) to zero (we call it noise ratio). This was implemented using dropout in the input layer [11].

As the gene expression values are real numbers, this correponds to a regression problem, so the three distinct models use the mean squared error (MSE) as the loss function, and linear activation function in the output layer. Besides, the number of input and output units in all models is 5000 (the number of input features).

2.1.2 Classification learning

Classification task

As we said at the beginning of the previous section, to performe classification, we used the

Layer	**Hyper-parameter**	**Possible values**
Hidden one (encoder)	L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
	Activation function	ReLU
	Batch Normalization	Yes
	Number of hidden units	2500
Hidden two (encoder)	L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
	Activation function	ReLU
	Batch Normalization	Yes
	Number of hidden units	1000
Hidden three (encoder)	L1 regularization	$\{1 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$
	Activation function	ReLU
	Batch Normalization	Yes
	Number of hidden units	100
Hidden four (decoder)	L1 regularization	No
	Activation function	ReLU
	Batch Normalization	No
	Number of hidden units	1000
Hidden five (decoder)	L1 regularization	No
	Activation function	ReLU
	Batch Normalization	No
	Number of hidden units	2500
Whole model	Noise ratio	$\{0.1, 0.15, 0.2\}$
	Learning algorithm	Adam
	Learning rate	$\{0.00005, 0.0001, 0.0005\}$
	Number of epochs	$\{150, 175, 200\}$
	Batch size	$\{200, 225, 250\}$

Table 3: Hyper-parameter space of the stacked-sparse-denoising autoencoder model.

BRCA pan-cancer dataset (1K samples). In particular, the variable to be predicted is the PAM50 BRCA subtype. PAM50 is a widely used 50-gene breast cancer intrinsic subtype predictor [45] that, applied to our RNA-Seq data, group the samples in four subtypes: Luminal A, Luminal B, Basal-like and Her-2 enriched, which are the main breast cancer subgroups from a clinical point of view. Hence, our classification problem consist on a multi-class classification problem of 4 different classes.

Fine-tuning and classification

Once the autoencoders are pre-trained on the non-BRCA pan-cancer dataset, following our transfer-learning approach, the encoders need to be fine-tuned. Hence, using a 5-fold nested Cross-Validation (CV) procedure, and extracting the encoder layers of each pre-trained autoencoder, the models are fine-tuned on the BRCA pan-cancer dataset using a *softmax* layer on top of the last encoder layer. Once tuned, we use them to encode the gene expression data, and use the reduced dimensional data as input of three classification algorithms: Logistic Regression (LR), Support Vector Machines (SVM) and swallow Artificial Neural Network (ANN). Actually, we use the outer CV for fine-tuning the encoders and training the classification algorithms, and for evaluating the models using the average accuracy measure (ACC). The inner CV procedure is used to tune some hyper-parameters of both the encoders and the classification algorithms, using Randomized-Search method. We also standard scaled the data (zero mean and unit variance) before feeding it into the autoencoders and before applying the classifiers.

For sparse, stacked-sparse and stacked-sparse-denoising autoencoders, their fine-tuning hyper-parameters and their configurations can be seen in tables 4, 5 and 6 respectively. The number of freezed layers corresponds to the last encoder layer to be freezed, i.e. set as non-trainable during fine-tuning. The Dropout2 refers to the dropout used in the last encoder layer, and corresponds to the proportion of units set to zero during training. As the deep stacked-sparse and stacked-sparse-denoising models may suffer from overfitting because of their complexity, an additional Dropout1 parameter indicates the dropout ratio used in the hidden encoder layer specified by DropoutPos.

Hyper-parameter	**Possible values**
Learning algorithm	Stochastic Gradient Descent
Learning rate	$\{0.0001, 0.0005, 0.001\}$
Momentum	$[0.5, 0.9]$
Dropout2	$\{0, 0.3, 0.5\}$
Number of epochs	$[10, 30]$
Batch size	$[20, 60]$

Table 4: Fine-tuning hyper-parameter space of the sparse one-hidden layer encoder.

Hyper-parameter	**Possible values**
Number of freezed layers	$\{0, 1\}$
Learning algorithm	Stochastic Gradient Descent
Learning rate	$\{0.0001, 0.0005, 0.001\}$
Momentum	$[0.5, 0.9]$
Dropout2	$\{0, 0.3, 0.5\}$
DropoutPos	1
Dropout1	$\{0, 0.3, 0.5\}$
Number of epochs	$[30, 60]$
Batch size	$[30, 80]$

Table 5: Fine-tuning hyper-parameter space of the stacked-sparse encoder.

Hyper-parameter	**Possible values**
Number of freezed layers	$\{1, 2\}$
Learning algorithm	Stochastic Gradient Descent
Learning rate	$\{0.0001, 0.0005, 0.001\}$
Momentum	$[0.5, 0.9]$
Dropout2	$\{0, 0.3, 0.5\}$
DropoutPos	$\{1, 2\}$
Dropout1	$\{0, 0.3, 0.5\}$
Number of epochs	$[40, 70]$
Batch size	$[50, 100]$

Table 6: Fine-tuning hyper-parameter space of the stacked-sparse-denoising encoder.

For the Logistic Regression, Support Vector Machine and swallow Artificial Neural Network classification algorithms, their hyper-parameters are their possible values are shown in tables 7, 8 and 9.

Hyper-parameter	**Possible values**
Norm penalization	{L1, L2}
Multiclass	{One-Versus-Rest, Multinomial}

Table 7: Hyper-parameter space of the Logistic Regression model.

Hyper-parameter	**Possible values**
Kernel	{Radial Basis Function, Polynomial}
C penalty	$\{0.1, 1, 10, 100, 1000\}$
Gamma	$[1 \times 10^{-4}, 1 \times 10^{-1}]$
Polynomial kernel degree	$[2, 5]$

Table 8: Hyper-parameter space of the Support Vector Machine model.

Hyper-parameter	**Possible values**
Number of units in the hidden layer	$\{20, 40, 60\}$
Hidden layer activation function	{sigmoid, tanh, ReLU}
Learning algorithm	Stochastic Gradient Descent
Learning rate	$[0.001, 2]$
Momentum	$[0.2, 0.75]$
Maximum number of iterations	$\{100, 200\}$

Table 9: Hyper-parameter space of the swallow Artificial Neural Network (one-hidden layer) model.

2.2 Traditional Machine-Learning approach

In order to evaluate the performance of the deep learning approach, we compared this strategy with a traditional machine-learning approach for reducing the dimensionality of the gene expression data, with the purpose of performing a cancer classification task.

Data preparation

The same dataset used during the classification learning stage of the previous section is used here, the BRCA pan-cancer dataset (1K samples) with the 5K most variably expressed genes as input variables. Also, the same multi-class classification task is performed, the prediction of the PAM50 breast cancer intrinsic subtypes.

Dimensionality reduction and classification

For reducing the dimensionality of the gene expression data, we used different classical techniques: two feature selection methods, and a feature extraction technique. The feature selection methods correspond to filter methods, one using ANOVA F-values and the other using mutual information values [33, 46]. The feature extraction method is Principal Components Analysis (PCA), a widely used dimensionality reduction technique in

many different domains [47]. Just like when using the deep learning approach, the gene expression data was reduced to 100 features.

To be consistent with what we did when using the deep learning strategy, we again used 5-fold nested CV to evaluate the performance of the models, using the average accuracy measure. This time no fine-tuning is needed, and the features extracted by the traditional dimensionality reduction methods are used by the same classification algorithms as in the previous section: LR, SVM and swallow ANN. In this way, the inner CV is again used to tune some hyper-parameters of the classifiers (see previous section for details) using Grid-Search (LR) and Randomized-Search (SVM and ANN), and the outer CV is used to train the best selected models and evaluate their performance, by first reducing the dimensionality of the data. We also standard scaled the data (zero mean and unit variance) before feeding it into the classifiers.

3 Results

3.1 Deep models pre-training

As we stated before, the three autoencoders models were pre-trained using the non-BRCA pan-cancer dataset. For tuning several hyper-parameters of each architecture, we simply split the data into a training (90% of the samples) and a validation set (10%), and then examine the loss curves trying different values configurations.

In this way, when using the sparse architecture, we selected the values showed in Table 10. Using these values, the training and validation loss curves are the ones showed in Figure 6. The training process seems to be quite stable, though some oscillations are observed. However, overfitting clearly exists, as we can see from the distance between the training and validation loss curves.

Hyper-parameter	**Possible values**
L1 regularization	1×10^{-5}
Learning rate	0.001
Number of epochs	40
Batch size	100

Table 10: Hyper-parameters selected values of the sparse one-hidden layer autoencoder model.

For the stacked-sparse model, the selected hyper-parameters values configuration is described in Table 11. As we can see from Figure 7, the training process seems to be quite stable too, and overfitting is again observed.

Finally, when using the stacked-sparse-denoising autoencoder, the selected values of the hyper-parameters are shown in Table 12. The loss curves in Figure 8 show an incredibly stable learning process, with no observable fluctuation, as well as the almost overfitting inexistence, something really difficult to obtain in such a deep model like this one (five hidden layers). This overfitting reduction in comparison with the other two models is due

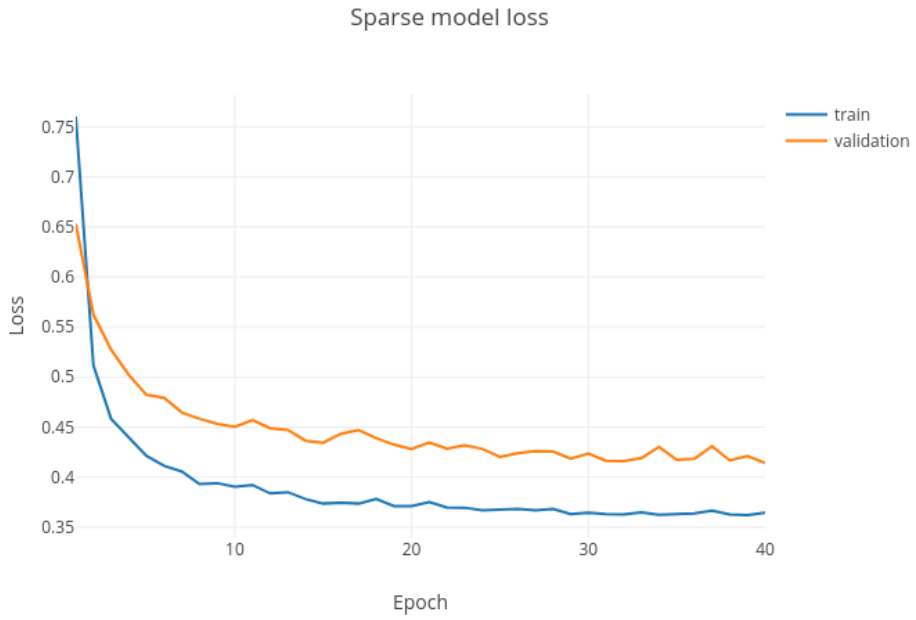

Figure 6: Training and validation loss curves of the sparse architecture.

Layer	**Hyper-parameter**	**Possible values**
Hidden one (encoder)	L1 regularization	1×10^{-5}
Hidden two (encoder)	L1 regularization	1×10^{-5}
Whole model	Learning rate	0.0005
	Number of epochs	150
	Batch size	200

Table 11: Hyper-parameters selected values of the deep stacked-sparse three-hidden layers autoencoder model.

to the noise used in the input layer of the denosing model, implemented using dropout, one of the most effective techniques to reduce overfitting in complex deep networks [11].

3.2 Classification results

The performance of our deep learning approach in breast cancer subtype classification is summarized in Table 13, whereas the Table 14 contains the performance of the classical machine-learning approach. All the values contained in both tables represent the average classification multi-class accuraccy obtained accros the 5 iterations of our 5-fold nested Cross-Validation procedure, and the performance of the models is evaluated in terms of the average test ACC.

Layer	Hyper-parameter	Possible values
Hidden one (encoder)	L1 regularization	1×10^{-5}
Hidden two (encoder)	L1 regularization	1×10^{-5}
Hidden three (encoder)	L1 regularization	1×10^{-5}
Whole model	Noise ratio	0.15
	Learning rate	0.0001
	Number of epochs	200
	Batch size	225

Table 12: Hyper-parameters selected values of the deep stacked-sparse-denoising five-hidden layers autoencoder.

Autoencoder	Classifier	Test ACC	Train ACC
Sparse	Softmax	0.867 ± 0.013	0.997 ± 0.003
	LR	0.864 ± 0.018	0.995 ± 0.005
	SVM	0.879 ± 0.022	0.999 ± 0.001
	ANN	0.849 ± 0.025	0.993 ± 0.014
Stacked-sparse	Softmax	0.887 ± 0.012	1 ± 0
	LR	0.878 ± 0.012	1 ± 0
	SVM	0.888 ± 0.021	1 ± 0
	ANN	0.888 ± 0.015	1 ± 0
Stacked-sparse-denoising	Softmax	**0.903 ±0.019**	1 ± 0
	LR	**0.900 ±0.011**	1 ± 0
	SVM	**0.893 ±0.010**	1 ± 0
	ANN	**0.889 ±0.011**	1 ± 0

Table 13: Performance of the classification algorithms using the deep learning approach for dimensionality reduction. In addition to the four traditional ML classifiers, we also include the softmax used to perform fine-tuning.

Dimensionality reduction	Classifier	Test ACC	Train ACC
Anova	LR	0.897 ± 0.023	0.971 ± 0.006
	SVM	0.894 ± 0.019	0.956 ± 0.028
	ANN	0.899 ± 0.021	0.983 ± 0.014
Mutual-Information	LR	0.908 ± 0.018	0.972 ± 0.007
	SVM	0.895 ± 0.021	0.987 ± 0.016
	ANN	0.901 ± 0.010	0.981 ± 0.021
PCA	LR	0.902 ± 0.021	0.982 ± 0.011
	SVM	0.906 ± 0.030	0.999 ± 0.001
	ANN	0.905 ± 0.018	0.977 ± 0.014

Table 14: Performance of the classification algorithms using the traditional machine-learning approach for dimensionality reduction.

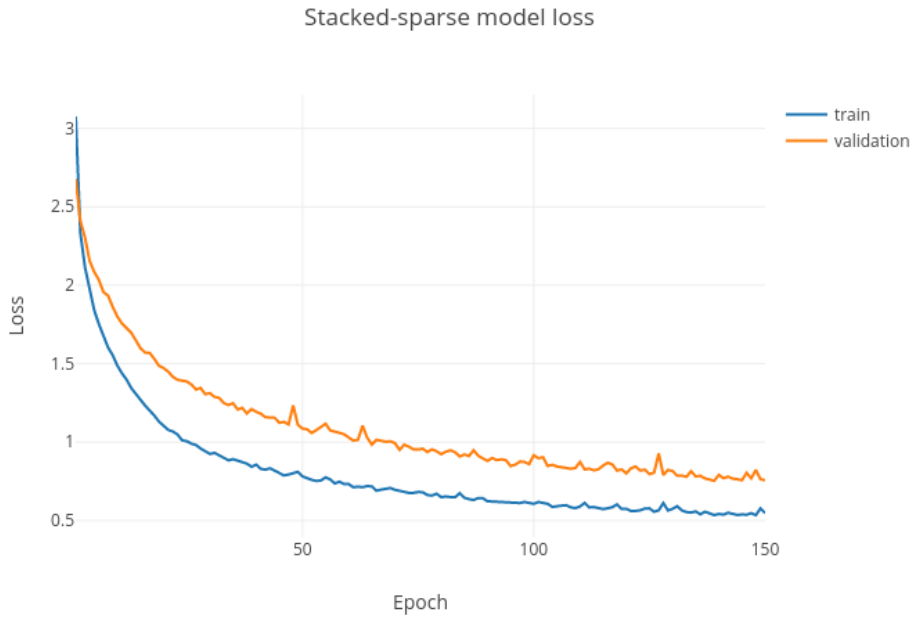

Figure 7: Training and validation loss curves of the stacked-sparse architecture.

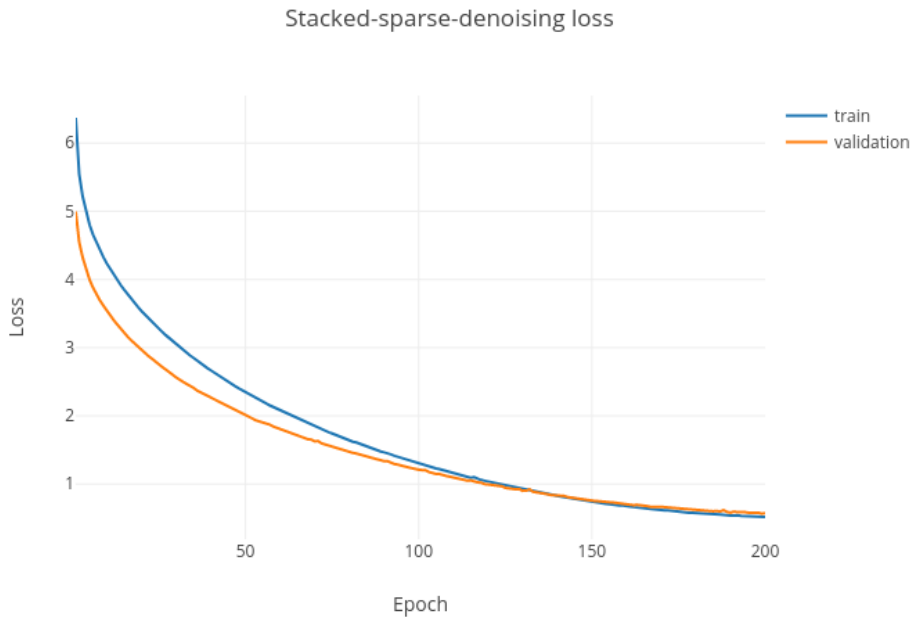

Figure 8: Training and validation loss curves of the stacked-sparse-denoising architecture.

When comparing the different autoencoder arquitectures, the deep models outperform the single hidden layer sparse autoencoder using any of the four classifiers. Besides, the deep stacked-sparse-denoising model obtains the best results, showing that its deep architecture and the corrupted data used during training allows the model to extract the most

useful features for classification purpose. However, caution is needed, as the differences between the 5-fold CV average accuracies are not statistically significant enough to draw categorical an definitive conclusions. On the other hand, comparing the classification algorithms, although the best test ACC value is obtained when using softmax classifier and stacked-sparse-denoising model, SVM seems to be most consistent when combined with autoencoder models, being the classifier that obtains the best performance when using sparse and stacked-sparse architectures.

Analyzing the traditional ML approach, though the maximum test ACC value is obtained when using mutual-info (and LR), PCA seems to be the most consistent method.

Comparing Table 13 and Table 14, the traditional ML strategy seems to outperform our DL approach. However, the deep stacked-sparse-denoising arquitecture outperforms Anova using any classifier. Hence, from the obtained results, we can say that, though some traditional approaches (mutual-info and PCA) seem to outperform the deep learning strategy, to conclude which is the most effective approach more work needs to be done.

One of the main reasons why DL strategy does not seem to outperform the traditional ML approach is overfitting. Comparing train and test ACC, there is a much bigger difference between those values in Table 13 than in Table 14. Although dropout technique was used, other regularization methods that prevent overfitting may be needed in order to boost the performance of DL models in this particular cancer classification task.

It is worth mentioning that our hardware limitations may has something to do with the fact that traditional machine-learning methods seems to outperform our deep learning approach. For the stacked-sparse and stacked-sparse-denoising deep architectures, we could only execute very few iterations of the Randomized-Search procedure to tune the hyper-parameters of the models, which could have been tuned much more effective using a greater number of iterations, possibly improving the prediction performance of those models.

4 Conclusion

In this project, we have tried to adapt deep learning, one of the most promising and successful AI technologies nowadays, to bioinformatics domain, in particular to gene expression analaysis for cancer classification. Our main goal was to give a solution to the enormous imbalance between the number of input features (N) and the number (M) of tumor gene expression available samples ($N \gg M$) using deep learning models, for the purpose of performing breast cancer intrinsic subtypes classification. In this way, we have used three distint types of autoencoders, an unsupervised feature learning technique, to reduce the dimenionality (N) of the gene expression data by a factor of 50. Besides, using a transfer-learning approach, we were able to pre-train the models using a large compendium of tumor data, different from the data used to perform the cancer classification task, and hence increasing the potential number of samples (M) used to train the models in an unsupervised way. This also allows deep learning models to extract generic features from a large tumor dataset that may be useful for solving a concrete cancer classification task such as breast cancer subtype prediction. Using a nested Cross-Validation strategy, we also compared the perfromance of the deep learning approach with a traditional machine learning strategy for reducing the dimenionality of the data.

The obtained results showed that the performances of both deep learning and traditional machine-learning approaches are very similar when reducing the dimensionality of gene expression data for the purpose of breast cancer subtype classification. However, though some traditional approaches seem to outperform the deep learning strategy, to conclude which is the most effective approach more work needs to be done. On the other hand, comparing the performance of the different autoencoders, although very cautiously, we could say that the deeper the model the better perfomance was obtained, showing the representational power of deep learning to extract a hierarchy of abstract representations useful for solving classification tasks.

4.1 Future work

In order to solve the difficulties faced by gene expression analysis classification tasks, different deep learning approaches are yet to be explored. Apart from transfer-learning, generative models such as GANs or VAEs could be used to generate artificial gene expression samples, and hence balancing the enormous disproportion of input features and available samples ($N \gg M$) mentioned above.

On the other hand, much more work needs to be done to exhaustively analyze the feasibility of the transfer-learning approach used in this project. We have used the extracted features by the pre-trained models to solve only one cancer classification task, but there are many more cancer prediction tasks were generic features extracted by deep learning models from a large compendium of tumor data could be very valuable.

Finally, another of the most unexplored areas of deep learning is model intrepretability. Although many efforts have been made, most of deep learning models are still considered as "black-boxes". In areas such as bioinformatics or medicine, if we want to apply these models, interpretability must not be a lacking quality, but a characteristic.

Finally, to sum up, it should be noticed that the two global objectives of the project have been greatly accomplished, and the project has served as my first experience in writing and carrying out a scientific project, in which many of the concepts learnt throughout the Bioinformatics degree have been successfully applied.

5 Conclusión

En este proyecto, se ha tratado de adaptar el aprendizaje profundo, una de las tecnologías más exitosas de la Inteligencia Artificial (IA) en la actualidad, al dominio de la bioinformática, concretamente al análisis de datos de expresión génica para su uso en tareas de predicción de cáncer. Nuestro principal objetivo era el de emplear el aprendizaje profundo para proporcionar una solución al problema del enorme desequilibrio entre el número de variables de entrada (N) y el número (M) de muestras disponibles ($N \gg M$) que presentan los datos de expresión génica, cuando se emplean en tareas como la predicción del tumor intrínseco de cáncer de mama que presenta un determinado paciente. De esta forma, se han empleado tres tipos diferentes de *autoencoders* (ténica de aprendizaje no supervisada) para reducir la dimensionalidad (N) de los datos en un factor de 50 (respecto al número de variables original). Además, empleando un enfoque basado en la técnica de transferencia de aprendizaje, ha sido posible pre-entrenar los modelos usando un enorme conjunto de datos de muestras tumorales, diferente al conjunto usado en la tarea de clasificación, consiguiendo de esta manera aumentar el número de potenciales muestras (M) empleadas para entrenar los modelos. Todo ello también permite a los modelos de aprendizaje profundo extraer carácterísticas genéricas a partir de un gran conjunto de datos de muestras tumorales, las cuales pueden ser de gran utilidad a la hora de resolver tareas de clasificación en cáncer como la predicción del tumor intrínseco de cáncer de mama. Además, emplando una estrategia de validación cruzada anidada, se ha podido comparar el rendimiento del enfoque basado en modelos de aprendizaje profundo con una estrategia basada en técnicas tradicionales de aprendizaje automático para la reducción de la dimensionalidad de los datos.

Los resultados obtenidos muestran que el rendimiento de los enfoques basados en aprendizaje profundo y en técnicas tradicionales de aprendizaje automático son muy similares a la hora de reducir la dimensionalidad de los datos de expresión génica para su empleo en la predicción del subtipo intrínseco de cáncer de mama. Sin embargo, aunque algunos enfoques tradicionales parecen superar el rendimeinto del enfoque basado en aprendizaje profundo, para concluir cuál es el enfoque más efectivo más trabajo es necesario. Por otro lado, comparando el rendimiento de los diferentes *autoencoders* implementados, aunque con mucha prudencia, podemos decir que cuanto más profundo el modelo

mejor rendimiento obtuvo, mostrando el poder representacional de estos modelos para la extracción de una jerarquía de representaciones abstractas útiles para la resolución de tareas de clasificación.

5.1 Trabajos futuros

Con el objetivo de resolver las dificultades que plantea el análisis de datos de expresión génica, diferentes modelos de aprendizaje profundo permanecen todavía por explorar. Además de técnicas de transferencia de aprendizaje, modelos generativos como las GANs o los VAEs pueden ser empleados para generar de forma artificial muestras de datos de expresión, contribuyendo así a equilibrar el enorme desbalanceo entre el número de variables de entrada y el de muestras disponibles ($N \gg M$) mencionado con anterioridad.

Por otro lado, mucho trabajo queda por hacer para analizar de froma exhaustiva la viabilidad del enfoque basado en transferencia de aprendizaje empleado en este proyecto. Se han utilizado las características extraídas por los modelos pre-entrenados para resolver una única tarea de predicción de cáncer, pero existen muchas más tareas en las que las características genéricas extraídas por los modelos de aprendizaje profundo pueden ser de gran utilidad.

Finalmente, otra de las áreas más inexploradas del aprendizaje profundo se corresponde con el estudio de la interpretabilidad de los modelos. Aunque se han realizado numerosos esfuerzos, la gran mayoría de modelos basados en el aprendizaje profundo son todavía considerados como "cajas negras". En áreas como la bioinformática o la medicina, si pretendemos aplicar estos modelos, la interpretabilidad no ha de ser una carencia, sino una de las características indispensables de estos modelos.

Por último, cabe destacar que los dos objetivos globales de este proyecto han sido cumplidos con enorme éxito, y el proyecto ha supuesto mi primera experiencia escribiendo y llevando a cabo un proyecto científico, en el que he podido aplicar muchos de los conceptos adquiridos durante estos cuatro últimos años en el grado de Ingeniería de la Salud.

6 Resources

6.1 Software

All the software implemented for this project was coded in Python, using *Jupyter* notebooks. *Tensorflow*, *keras*, *sklearn*, *numpy*, *pandas* and *scipy* were the packages used to implement the deep learning models, machine-learning methods, nested-CV strategy, etc. Git was used as the version control system.

On the other hand, all the produced software (mainly *Jupyter* notebooks) are allocated in the next public repository: `https://github.com/guilopgar/DeepLearning-Bioinformatics`

6.2 Hardware

As it has been mentioned throughout the previous section, the hardware conditions have been a major limitation, specially for training the deep models and pre-processing the data.

Actually, I used my own personal machine for carrying out the project, an Ubuntu 14.04 system with 8GB RAM, 1TB hard disk and an Intel® Core™ i7-4510U CPU @ 2.00GHz 4 processor.

7 References

1. Y. N. Harari, *Homo Deus: A Brief History of Tomorrow* (Harvill Secker, 2015).
2. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
3. Y. LeCun *et al.*, Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems* **2**, 396–404 (1990).
4. K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ArXiv e-prints* (2015).
5. L. Fridman *et al.*, MIT Autonomous Vehicle Technology Study: Large-Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation. *ArXiv e-prints* (2017).
6. I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems* **27**, 3104–3112 (2014).
7. G. Hinton *et al.*, Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine* **29**, 82–97 (2012).
8. I. J. Goodfellow *et al.*, Generative Adversarial Nets. *Advances in Neural Information Processing Systems* **27**, 2672–2680 (2014).
9. T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation. *ICLR* (2018).
10. Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Machine Intell.* **35**, 1798–1828 (2013).
11. N. Srivastava *et al.*, Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15**, 1929–1958 (2014).
12. S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ICML* (2015).
13. V. Nair, G. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines. *Proc. 27th International Conference on Machine Learning* (2010).
14. V. Mayer-Schonberger, K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work and Think* (Houghton Mifflin Harcourt, 2012).
15. Q. Zhang, L. T. Yang, Z. Chen, P. Li, A survey on deep learning for big data. *Information Fusion* **42**, 146–157 (2018).
16. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016).

17. A. Coates *et al.*, Deep learning with COTS HPC systems. *International Conference on Machine Learning* (2013).

18. D. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Deep neural networks segment neuronal membranes in electron microscopy images. *MIT Press*, 2843–2851 (2012).

19. C. Angermueller, T. Pärnamaa, L. Parts, O. Stegle, Deep learning for computational biology. *Molecular Systems Biology* **12** (2016).

20. D. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Mitosis detection in breast cancer histology images with deep neural networks. *Medical Image Computing and Computer-Assisted Intervention–MICCAI*, 411–418 (2013).

21. W. Zhang *et al.*, Deep model based transfer and multi-task learning for biological image analysis. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1475–1484 (2015).

22. J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems* **27**, 3320–3328 (2014).

23. M. Flores *et al.*, P4 medicine: how systems medicine will transform the healthcare sector and society. *Personalized Medicine* **10**, 565–576 (2013).

24. H. Zeng, M. D. Edwards, G. Liu, D. K. Gifford, Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics* **32**, 121–127 (2016).

25. B. Alipanahi, A. Delong, M. T. Weirauch, B. J. Frey, Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33**, 831–838 (2015).

26. J. Zhou, O. G. Troyanskaya, Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2016).

27. Z. Wang, M. Gerstein, M. Snyder, RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics* **10**, 57–63 (2009).

28. X. Zhao, E. A. Rødland, R. Tibshirani, S. Plevritis, Molecular subtyping for clinically defined breast cancer subgroups. *Breast Cancer Research* **17** (2015).

29. O. Yersal, S. Barutca, Biological subtypes of breast cancer: Prognostic and therapeutic implications. *World Journal of Clinical Oncology* **5**, 412–424 (2014).

30. I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* **3**, 1157–1182 (2003).

31. K. Kourou *et al.*, Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **13**, 8–17 (2015).

32. A. Bashiri *et al.*, Improving the Prediction of Survival in Cancer Patients by Using Machine Learning Techniques: Experience of Gene Expression Data: A Narrative Review. *Iran. J. Public Health* **46**, 165–172 (2017).

33. Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**, 2507–2517 (2007).

34. R. Fakoor, F. Ladhak, A. Nazi, M. Huber, Using deep learning to enhance cancer diagnosis and classification. *Proceedings of the 30th International Conference on Machine Learning*, 1–7 (2013).

35. P. Danaee, R. Ghaeini, D. A. Hendrix, A deep learning approach for cancer detection and relevant gene identification. *Proceedings of the Pacific Symposium on Biocomputing* **22**, 219–229 (2016).

36. G. P. Way, C. S. Greene, Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Proceedings of the Pacific Symposium on Biocomputing* **23**, 80–91 (2018).

37. TCGA, *TCGA Releases The Pan-Cancer Atlas*, `https://cancergenome.nih.gov/newsevents/newsannouncements/pancancer_atlas`, [Online; accessed 15-June-2018].

38. J. Vivian *et al.*, Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology* **35**, 314–316 (2017).

39. A. Lachmann *et al.*, Massive mining of publicly available RNA-seq data from human and mouse. *Nature Communications* **9** (2018).

40. A. Y. Ng, *Unsupervised feature learning and deep learning - Autoencoders*, `http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/`, [Online; accessed 15-June-2018].

41. A. Coates, H. Lee, A. Y. Ng, An analysis of single-layer networks in unsupervised feature learning. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2011).

42. P. Vincent, H. Larochelle, Y. Bengio, P. A. Manzagol, Extracting and composing robust features with denoising autoencoders. *Proceedings of the Twenty-fifth International Conference on Machine Learning*, 1096–1103 (2008).

43. J. Tan, M. Ung, C. Cheng, C. S. Greene, Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. *Proceedings of the Pacific Symposium on Biocomputing* **20**, 132–143 (2015).

44. L. Gondara, Medical image denoising using convolutional denoising autoencoders. *IEEE 16th International Conference on Data Mining Workshops* (2016).

45. J. S. Parker *et al.*, Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology* **27**, 1160–1167 (2009).

46. J. R. Vergara, P. A. Estévez, A review of feature selection methods based on mutual information. *Neural Computing and Applications* **24**, 175–186 (2014).

47. H. Abdi, L. J. Williams, Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 433–459 (2010).