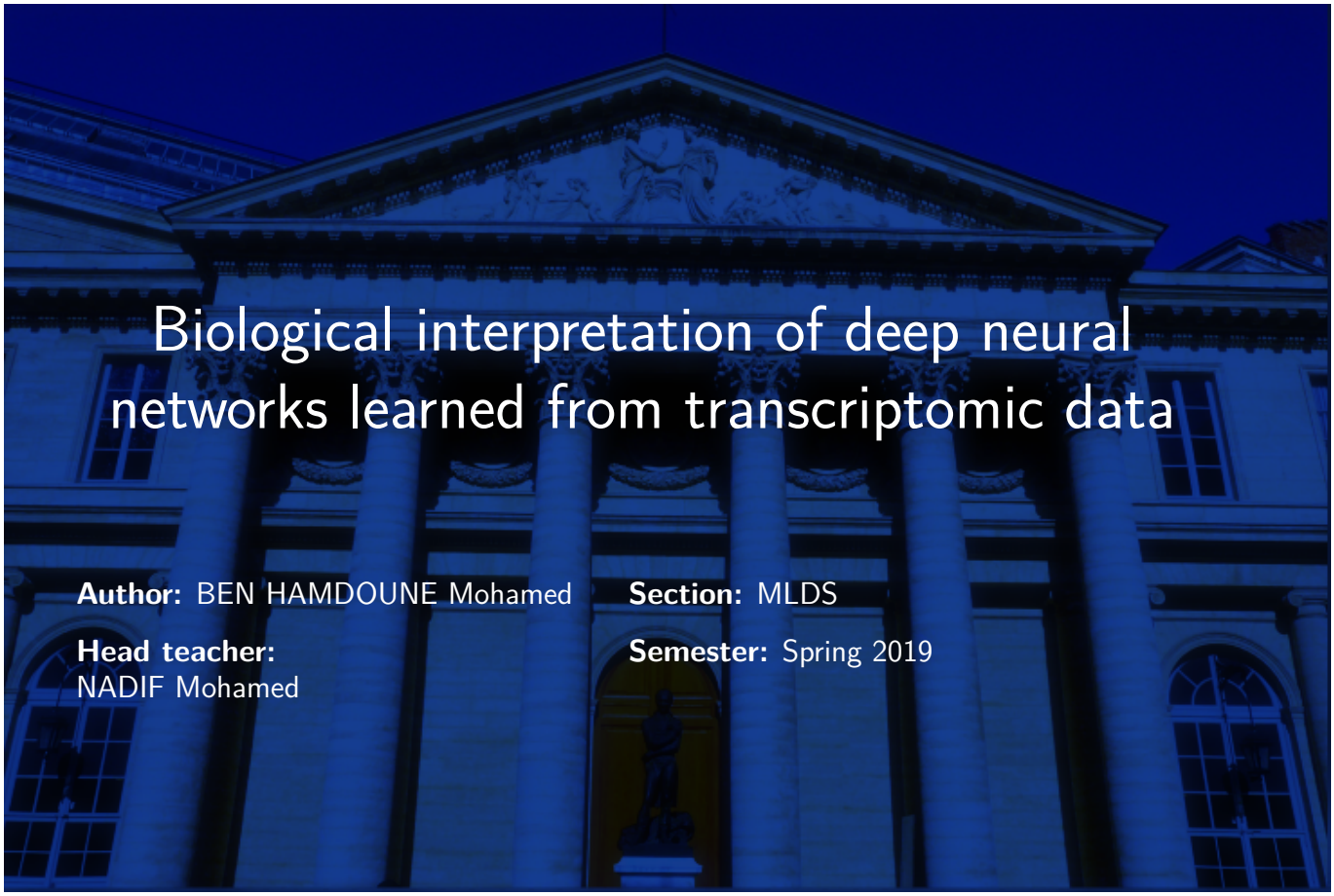

Biological interpretation of deep neural networks learned from transcriptomic data

Author: BEN HAMDOUNE Mohamed

Section: MLDS

Head teacher:
NADIF Mohamed

Semester: Spring 2019

Summary

The interpretation of neural networks and their predictions is a major challenge. Neural networks are considered as "black boxes", in which patient data are injected as input and a prediction is calculated as output without explanation. The European Union recently adopted a legal text for users of learning algorithms to be able to explain the decisions of a predictive model. There is therefore a real need to make neural networks more interpretable and this is particularly true in the medical field. It is important to ensure that the neural network bases its predictions on reliable patient representation and does not focus on irrelevant artefacts present in the training data. Without explanation of the predictions, physicians can not trust the model whatever its performances.

Laboratory: Laboratoire IBISC
Place: Evry
Tutor: HANCZAR Blaise

Keywords:

Deep Learning
Relevance Propagation
Self-Explanation
Gene Ontology

Acknowledgements

This internship was an opportunity for me to discover a little more about research but also on deep learning and biology. So, I would like to thank Hanczar Blaise for helping me to gain skills in this area of research.

I would also like to express my very great appreciation to Johan Arcile and Beji Lotfi for the discussions we held.

I am particularly grateful for the fruitful conversations and assistance given by Louis Becquey throughout my internship, and Tina Issa and Ghiwa Khalil for the biological interpretation of my results.

I would also like to thank Cristel Dos Santos Catarino and Victoria Bourgeais for their help and availability. I also warmly thank Michal Strikek and Junkai He for their sympathy, their professionalism.

I particularly wish to thank Baptiste Lemoine, Kevin Decroos, for their invaluable assistance in proofreading and correcting my memoir. In addition, I extend my sincere thanks to my family, and all my relatives and friends, who accompanied, helped, supported and encouraged me throughout the realization of this memoir.

The members of the AROBAS (Algorithmique et Recherche Opérationnelle, Bio-informatique, et Apprentissage Statistique) and COSMO (Communications Spécifications Modèles) teams who were able to bring their experience and their advices to the various trainees.

Finally, I didn't have the opportunity to formalize my academic thanks in another report than this one, therefore I would also like to take advantage of this report to particularly acknowledge the importance of some of my teachers like Lazhar Labiod, François Role, and also Mohamed Nadif.
I cannot thank them enough for having given birth to a passion and a vocation. Until then, I had only little idea of what I was doing, and I would only have kept scratching the surface.

Abstract

We live in a time where possibilities are endless and deep learning technology can help us to make new technological advances. It greatly increases our understanding of biology, including genomics, proteomics, metabolomics, and immunomics. Despite that, we are still facing the problem of deep learning comprehension and interpretation, and the latest legislation requires us to have a network reliable and interpretable.

The internship focuses on the design of a model of neural networks including on one hand, the learning part with biological constraint, and interpretation on the other hand. The goal is to remove the black box aspect of deep learning models. We must be able to trust them before we adopt them because deep learning networks are leading on the analysis of complex data structures. We will follow a methodology by creating a model that we will name InOmicNet (Interpretable Omic Network).

The subject of this memoir is divided into three parts. The first is about all state of art since it involves applying methods of explaining networks of deep neurons. This part will mainly constitute the techniques that are currently used.

In the second part, we will focus on the approach and we will build an architecture integrating biological information and a custom cost function to consider this information. The number of layers will also be determined according to the distribution of GO functions per depth. We are in the case of s*elf-explanation* model and not on a *a posteriori* interpretation model.

The last part will be on biological interpretation. It is necessary to associate with relevant neurons, biological functions, metabolic pathways or diseases. it will validate existing methods of explanation in the field of medicine precision. We can conclude with a probability test whether our network makes conclusions in a hazardous way or the network has learned the biological elements involved in cancer. This will be the subject of the experimental part of the dissertation.

Contents

Chapter 1

Introduction

1.1 The IBISC Laboratory

Created in 2006, the IBISC (Informatique, Bio-informatique et Systèmes Complexes) laboratory is a strong STIC (Sciences et technologies de l'information et de la communication) pole on the University of Evry Val d'Essonne. Its tutors are the University of Evry Val d'Essonne and ENSIIE (École nationale supérieure d'informatique pour l'industrie et l'entreprise). It is also part of Genopole.
The IBISC laboratory is made up of 4 teams (AROBAS, COSMO, IRA2 (Interaction, Réalité virtuelle Augmentée, Robotique Ambiante), SIAM (Signal, Image et AutoMatique)) whose activities fall into two scientific areas STIC & SMART SYSTEM and STIC & VIVANT.

- STIC & SMART SYSTEM: The research defined in this axis deals with the design of autonomous and intelligent systems. The notion of system refers to both fleets of road and air vehicles, robots, distributed and communicating software and services, or intelligent hardware components with interacting sensors.
- STIC & VIVANT: this interdisciplinary research covers a wide spectrum of biology issues at different scales of the living: analysis of biological and biomedical data and signals, modelling of biological systems, learning of surgical gestures and assistance to the individual.

The AROBAS team focuses on three areas: Algorithms and Operational Research, Bioinformatics, and Statistical Learning.

1.2 Internship Context

The European regulation imposes specific consent requirements on the profiling and use of analytical solutions: the consent must be informed, recorded and presented to the auditors details on the use of profiling. This right to the explanation, inscribed in the GDPR (General Data Protection Regulation), becomes a thorny subject for the world of artificial intelligence.

In the name of transparency, companies using artificial intelligence will be forced to popularize the process of their algorithms. Mounir Majhoubi, secretary of state in charge of the digital explained to the National Assembly, "if an algorithm cannot be explained, it cannot be used in the public service".
In this context, I carried out my internship at the IBISC laboratory, since it is essential to have a transparency on the model when we use it for a critical area such as health or defense.

Artificial intelligence, and more precisely deep learning, has become popular thanks to the technological

evolution of our machines and their components, which allows us to build complex models with a consequent number of parameters, and to be able to have our results in a reasonable time. As I write this memoir, evolution continues to progress very rapidly with increasingly powerful GPUs (Graphics Processing Unit) and techniques associated with deep learning in several areas, such as computer vision or natural language processing, that are currently being used on our daily lives.

These aspects which are the progress of the capacities of our computational powers as well as the new consideration in our current legislations are the main reasons that put our work forward, and will lead to the establishment of a future decision assistant for physicians.

1.3 Plan

The subject of this dissertation is therefore to provide tools in line with the new GDPR directives of the European Union. Like we said previously, to be put into production in a sensitive area such as medicine or defense, the model must be interpretable, but in the case of deep learning it is more complex.
We use transcriptomic data (gene expressions). Many methods for interpreting the results of neural networks applied to image analysis are already functional, but we are facing a challenge because we will not use a convolutional network and our inputs are omics data.

In the medical field, data are very scarce but there can be conversely a very large number of input variables. In our case, dimensionality of input data does not pose a big concern to neural networks in a general way. To solve this problem of interpretation we will pass first by the biological knowledge, by defining what a gene as well as the transcriptomic data are, and a view on the cancer too.
Then we will discuss the approach of learning techniques in the context of omics data and more precisely on the case of deep learning in the context of our method. We will use a dense network with some additional layers useful for the improvement of the learning during training.

An in-depth study of the current state of the art to argue what methods are developed and used in the case of interpretation. We will introduce the two methods that are based on perturbations and gradient methods with their variants.
After studying the main methods made popular by their effectiveness and their abilities to adapt to different problems, we will develop in detail the methodology used to make transparent the prediction of a network when it predicts a cancer for a patient.

We will first explain the gene ontology (GO), as an acyclic directed graph which contains the information of the biological functions. We will then associate the descendants of each GO function with a large matrix of adjacency, in order to stick each level of the graph to a layer. This allows the matrix to learn to adapt to non-descendant connectivity when updating weights at each of its backpropagation (i.e., we will add this information to a custom cost function by adding a coefficient regularized term).
It will also be a question of developing a vast monitoring system which will serve as a demonstration on the reliability of the model and its consistency and from its good return we will then be able to develop on the biological interpretation.

We will recover the biological functions associated with the best scores according to a method of relevance scores then also a multitude of backpropagation analysis according to the best neurons per layer in order to recover the best probes used in input (the 5000 best probes for example).

Chapter 2

Definitions

An interpretation [1] is a projection of an explicit notion (e.g. an expected category) into a domain that humans can create meaning of and an explanation is a set of characteristics of an interpretable domain that have led to a decision-making process (e.g. classification or regression).

Interpretation is a very broad field in deep learning. To define what this term means, we will use the definitions of [2]. However, we will not repeat the same terms. Indeed, in the article by [2], it is a question of interpretation and explanation. To avoid confusion, we will talk about interpreting a model and interpreting a prediction.

They are trying to refine [2] the discourse on interpretability. Many articles suggest interpretability as a way of generating confidence. But what is confidence, though ? Does it relate to the belief in the results of the model, the robustness or some other ownership of the choices it produces ? Does interpretability merely imply a low-level mechanistic knowledge of our designs ? If so, does it apply to the features, parameters, models, or training algorithms ? Other publications suggest a connection between an interpretable model and one wich uncovers causal an interpretable model and one which uncovers causal structure in data. Some publications equate interpretability with understandability or intelligibility, i.e, that we can grasp how the models work. The problem can also arise when the dynamics of the deployment environment differ from the training environment. In all instances, interpretations represent those goals that we consider significant but are struggling to correctly model.

2.0.1 Interpreting models

The interpretation of a model (interpretation) consists in making the link between an abstract concept (a predicted class) and a domain that would make sense for a human being and we must be careful not to be confused with the interpretation of a prediction.
More concretely, this means that the interpretable model must be able to explain what a class is, using a process understandable by humans. Humans have an innate ability of their own to be able to interpret the facts surrounding them.

The interpretation of a model is sometimes called Model Transparency ([3], [4]). Indeed, the interpretation consists in solving the problem of the black box, hence the expression "transparency". Confidence in a model can be defined objectively such as metrics (precision, recall, ...).
In this sense, we might care not only about how often a model is right but also for which examples it is right. We will now see some important facts on model interpretation:

- Causality: The job of inferring causal relationships from observational data has been widely researched. But these techniques appear to depend on powerful judgments about previous

understanding.

- Informativeness: While the goal of the device may be to decrease mistake, the real-world goal is to provide helpful data. The most evident manner the model conveys information is through its inputs. An interpretation may demonstrate informative even without shedding touch on the internal functioning of the model.
- Fair and Ethical Decision-Making: Algorithms conform to ethical standards [5]. Moreover, the same regulations suggest that algorithmic decisions should be contestable.
- Algorithmic Transparency: Modern deep learning techniques, on the other side, absence this kind of algorithmic transparency. While heuristic optimization procedures for neural networks are demonstrably powerful, we do not understand how they operate and cannot at current ensure a priori that they will work on fresh issues. Note, however, that human beings do not display these types of transparency.
- Post-hoc Interpretability: Although post-hoc interpretability often fails to clarify exactly how a model operates, they may nevertheless provide helpful data for professionals and end consumers of teaching techniques.

2.0.2 Interpreting predictions

The interpretation of a prediction is the set of variables in the interpretable domain that contributed to the decision making for a given example. For example on images to explain the decision making, we highlight the pixels of our images that have been determining.
We discuss of post-hoc interpretability, or else [4] speak of Model functionality to describe the interpretation of a prediction.

There exist many auxiliary criteria that one may wish to optimize. Privacy implies that the technique is used to protect delicate information in the records. Properties such as quality and robustness determine whether algorithms achieve certain concentrations of efficiency in the presence of entry parameter variability. Causality means that the expected yield shift owing to disruption will happen in the actual scheme and that the available technique provides data.

In the field of psychology literature, where [6] notes "explanations may highlight an incompleteness" we argue that interpretability can assist in qualitatively ascertaining whether other desiderata such as fairness, privacy, reliability, robustness, causality, usability and trust are met. For example, one can provide a feasible explanation that fails to correspond to a causal structure, exposing a potential concern.

We contend that the need for interpretability stems from incompleteness in the formalization of the issue, generating a basic obstacle to optimization and assessment. Note that the incompleteness is separate from the error: the merged estimation of the place of a missile may be unsure, but this error can be rigorously quantified and officially justified.

2.1 Biological notions

2.1.1 Gene

A gene is a DNA (DeoxyribNnucleic Acid) sequence encoding a character, i.e., an ordered sequence of nucleotides having a beginning and an end. DNA is therefore the medium of genetic information. The comparison of various DNA fragments of different species shows that the only parameter that

varies within a DNA molecule is the order in which the nucleotides succeed each other and shows that they are identifiable not only by the sequence of nucleotide sequences, but also by the length of the sequence.

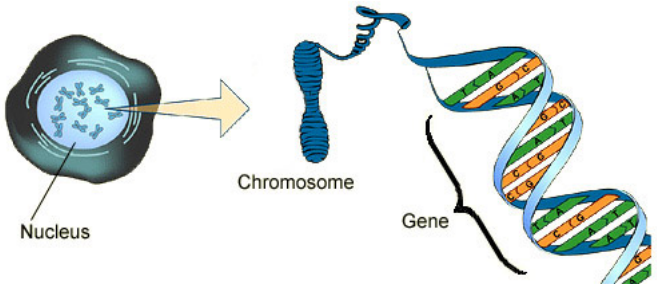

Figure 2.1: Illustration of DNA.

Gene expression refers to the set of biochemical processes by which the hereditary information stored in a gene is read to lead to the production of molecules that will have an active role in cell function, like proteins or RNAs (RiboNucleic Acid). Even if all the cells of an organism share the same genome, certain genes are expressed only in certain cells, at certain time during the life of the organism or under certain conditions.

The regulation of gene expression is therefore the fundamental mechanism for cell differentiation, morphogenesis and adaptability of a living organism to its environment.

2.1.2 Transcriptome

The transcriptome is defined as the set of transcripts present in a cell at a given moment and under given conditions. It is an image of the functional state of the genome.
DNA microarrays make it possible to study the transcriptome by simultaneously observing the expression of several thousand genes in a given cell or tissue, thus measuring the changes in the different cell states.
The microarray technique is based on the hybridization principle [7] which states that two complementary nucleic acid fragments can associate and dissociate reversibly under the action of heat and the saline concentration of the medium.

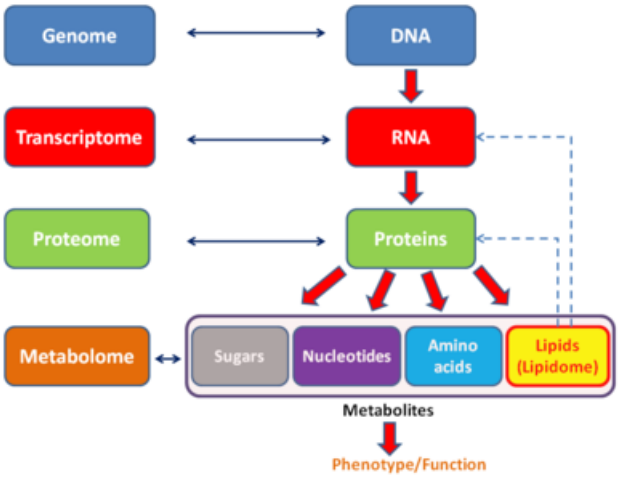

Figure 2.2: Illustration of Transcriptome position

Specifically, a DNA chip is a rigid support (glass or nylon) of a few square centimetres, on which

short DNA sequences have been deposited.
These short sequences are termed "probes" corresponding to synthetic oligonucleotides or PCR (polymerase chain reaction) products, then the probes have the distinction of having been chosen to be specific to a single gene and this microdevice is brought into contact with the RNAs extracted from the samples to be analysed called "targets".
These targets are marked by incorporation of radioelements or fluorochromes. After acquisition of the hybridization images, the quantification of the hybridization signals reflects the level of expression, in the initial sample, of each of the genes represented on the chip.

The first effort to collect a temporary animal transcriptome was released in 1991 [8] and 609 mRNA samples from the human brain were reference. In 2008 [9], two mammalian transcriptomes, made up of millions of transcript-derived sequences spanning 16,000 genes, were reference. Transcriptomic techniques are very simultaneous and involve big calculations to generate significant information for DNA chip and RNA-Seq tests.

In RNA-seq analysis [10], we use filtering criteria to reduce the consideration of differentially expressed genes whose expression levels could be tolerably hypothesized to be below the level essential to affect cellular function or phenotype. Thus, we are permitted to emphasize on those that are most likely to be biologically significant [11]. Normalization is an imperative step in the RNA-seq data analysis. The amount of reads which are monitored per gene relies on the expression level and the length of the gene, and on the RNA composition of the sample ([12], [13]).

The effect of gene length and total sample RNA composition is lessened by the normalization process. As a result, a direct clear representation of the targeted gene expression level is shown by the normalized read counts. Several normalization methods are used in RNA-seq analysis. Some of them are Total Count (TC), Upper Quartile (UQ), Median (Med), DESeq (Differential gene expression analysis based on the negative binomial distribution), Quantile (Q) and Reads PerKilobase per Million mapped reads (RPKM).

2.1.3 Cancer

Our body is made up of thousands of billions of cells grouped together to form tissues and organs. The genes in the nucleus of each cell are essential for regulation of growth, metabolism, division and apoptosis tell it when to grow, work, divide and die. Normally, our cells function in harmony and we remain healthy.
But when our DNA is altered or harmed, genes can mutate. Mutated genes do not function correctly because the DNA directives are mistaken. Then, the cells that need to be removed or die can split and develop into a disease that can contribute to cancer.

Cancer is no longer a disease from an organ, but a disease defined by the precise biological nature of its cellular composition.
The biological and clinical study of cancer nowadays involves the molecular characterization of tumours, DNA mutations and gene expression are indicative of the mechanisms of tumour progression and illuminate both the understanding of cancer biology and the therapeutic choices.

In this context, the emergence of new technologies of systematic molecular and subcellular investigation has revolutionized the research against cancer.
These technologies generate huge amounts of data, such as DNA chips (2 million measurements per chip), massive cell phenotyping, or next-generation sequencing (about 10 Gigabytes and a few hundred million sequences per experiment).

The volumes, the completeness, and the resolution of the information brought by these technologies lead us to propose new strategies of analysis and new concepts, with a rigorous scientific approach.

Laboratory tests measure the composition of certain substances in the body such as blood, urine and other liquids. Laboratory tests are important and provide information on the functioning of body organs, but one cannot rely solely on laboratory tests to make a diagnosis. Imaging studies include computed tomography (CT), ultrasound, magnetic resonance imaging (MRI), positron emission tomography (PET) and radiography. Biopsy is an operation performed to collect a sample of tissue or tumour from the body to find out if it contains cancer cells, the sample is examined under a microscope. The diagnostic process may seem long and discouraging, but it is important that the health care team eliminate any other possible cause of the health problem before making a diagnosis of cancer.

Because there is not one but multiple cancers and because each patient is unique, different types of treatments exist and the choice of treatments is adapted according to each situation. The main types of cancer treatments are:

- Surgery.
- Chemotherapy.
- Targeted Therapies.
- Radiotherapy.
- Hormone Therapy.

All these treatments are aimed at eliminating cancer cells. They act either locally, on all cells but more effectively on cancer cells present in the body.

2.2 Learnings on Omics Data

There are different types of learning techniques and in the context of this memoir, we will focus on supervised learning.

Here we are dealing omics data and the problems encountered, the experimental conditions, as well as the objectives targeted vary considerably: pre-selection of genes, considering of kinetics, importance of covariates, presence of phenotypic variables...
Nevertheless, the major specificity of these data, the one that most questioned the good use and know-how of the statistician, is the very high dimensionality of the number of genes whose expression is observed on a comparatively very small number of biological samples. Formally, the problem arises as the observation of a variable, transcriptomics expression (or quantity of messenger ribonucleic acid - mRNA- produced), in experimental situations that cross at least two factors: the gene and the type of biological sample (healthy or pathological tissue, wild-type or modified cell ...).

At present, teaching designs play a significant part in biological and biomedical research, particularly in the assessment of big omics information. Application issues are creating new difficulties:

- Large omics information collections can be highly unbalanced owing to the trouble of acquiring sufficient beneficial samples of these unusual mutations or illnesses.

- Many learning techniques are black boxes, which is enough for social applications. However, in biological or biomedical fields, knowledge of the molecular mechanisms underlying any disease or biological study is necessary to deepen our understanding.
- The genotype-phenotype association is a white index captured in conventional massive data studies but identifying causality rather than association would be more useful for physicians or biologists, as it can be used to determine an experimental target as a topic for future research.

Therefore, in order to simultaneously improve phenotypic discrimination and genotype interpretability for complex diseases, it is necessary to design and implement new learning technologies in order to integrate prior knowledge from gene ontology for example. The idea is to develop new theories and methods based on deep neural networks to find a compromise between precision and interpretability of learning in biomedical and biological fields.

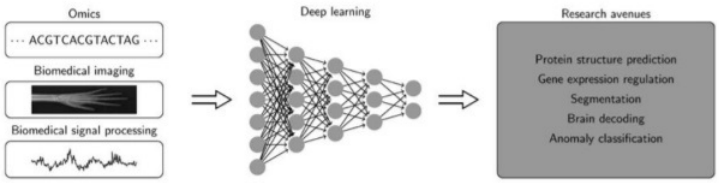

Figure 2.3: Deep Learning in Bioinformatics

The purpose of the statistical analysis is then to extract relevant information concerning the effect of various factors on the functional state of the cell.

The goal is to learn from experience and create a hierarchy of concepts that helps to understand something complex.
For example, an image can be understood through different concepts can be shapes or colors and in our case it's about omics data. The analysis and understanding of an input will be done through the prioritization of these concepts.

We will not cover here everything about deep learning, since it is a very vast field. But, in order to understand the topic of the dissertation and the course of the internship, we will discuss some important notions about neurons.

A formal neuron is a mathematical model that calculates an output from input data. The output corresponds to the dot product between its input vector x and a vector of weight w.
Following this, we add a bias b to this product. The result of this calculation will then go through an activation function f that will determine its output. It is said to be activated when its output value exceeds a threshold j.

The formula for calculating its output value is:

$$y = f(x.w + b) \tag{2.1}$$

The f most currently represented activation function in neural networks is the ReLu (Rectified Linear Unit) function. We can write it this way:

$$f = max(0, x) \tag{2.2}$$

Then, the output value of the neuron becomes:

$$y = max(0, x.w + b) \tag{2.3}$$

The neurons are then layered together to form a network.
This allows to learn more complex functions.
So we have an MLP (MultiLayer Perceptron), with an input layer, for the variables of the problem, a hidden layer, whose neurons are connected to all the previous ones, and then an output layer with a neuron for each class to predict.

In a context of supervised learning, prediction consists of giving a label to an example. Imagine that we have our function G which must classify images. This function represents the classification of the different data. Our neural network, which approximates G, must have roughly similar results. To make a prediction, our neural network propagates information from the input layer (pixels of the image), through the hidden layers, to our output layer (also called logit layer). The prediction will correspond to the neuron of the output layer having the highest value. This is called the forward.

A model or classifier is a network of neurons that has been trained. That means a weight and a bias were calculated for each neuron. This is done through the gradient. It is calculated after a prediction and we differentiate between the values of neurons corresponding to the predicted class and the real class. This difference is used during the learning phase to retro-propagate the mistake and calculate new weights and bias for our neurons. In a sense, the gradient represents the direction (it is also obtained following a derivative) to be taken for our prediction to be correct, it's called backpropagation [14].

We also use a method to improve the strength of the neural network, the batch normalization [15] normalizes the performance of the past input section by subtracting the batch average and separating it by the batch normal deviation.

Consequently, batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a "standard deviation" parameter (gamma) and add a "mean" parameter (beta). In other words, batch normalization lets SGD do the denormalization by changing only these two weights for each activation, instead of losing the stability of the network by changing all the weights.

We also use the famous Dropout [16] technique in order to improve the robustness of the network against the future example not known in advance by the network. Dropout is a technique that is intended to prevent overfitting on training data by dropping units in a neural network. In practice, neurons are either dropped with probability *p* or kept with probability *1-p*.

Moreover, even if we are in the case of a classification problem with two classes and we can also talk about binary classification, we do not use sigmoid activation function because for the future techniques of retro-propagation we must dissociate between the neurons predicting the healthy class and the sick class. So, we have in last layer, two neurons and our activation function is softmax.

Here is the formula of softmax:

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2.4}$$

The Softmax function, or normalized exponential function, is a generalization of the sigmoid function. In our case and even often, we have an unbalanced dataset because there are about 66% of patients being classified as cancerous and the rest healthy.
There is a multitude of techniques to counter this problem like the change of metric and also the oversampling or under sampling techniques but in our case, we will simply take into account the coefficient of each class when learning with Keras [17].

Chapter 3

State of the art

In this section, the goal is to explain why an example is part of a certain concept (class). For that, one is interested in the variables which contributed to take a decision.
This takes the form of a relevance score R_i that is attributed to the neuron i. We are talking about relevant methods. If $\mid R_i \mid$ is relatively high, so the variable is decisive. If $R_i > 0$ then the neuron goes in the direction of the prediction. That is, it validates the predicted class.

On the other hand, if $R_i < 0$, then the neuron goes against the prediction. In the case of transcriptomic data, this could mean that the activation of certain genes goes against a disease but this is still a hypothesis. For the assignment of relevance score to the different neurons, there are two main families of algorithms specific to deep learning. These are the disturbances method and gradient-based methods. The second one is sometimes also called Backpropagation-based methods.

3.1 Interpretation Techniques

3.1.1 Prediction Interpretation Techniques

Visualisation: Another way to overcome the problem of the black box is visualization. This has been studied in different articles, in different ways. The first one is called t-Distributed Stochastic Neighbor Embedding (t-SNE) [18]. The principle is like PCA (Principal Component Analysis). The aim is to visualize, most often in two dimensions, dataset of very large dimensions. Even if it is not really a mean for interpretation, it can help us to read the model learned. One last visualization method is to generate an abstract image from an example [19]. This helps to understand the abstract concept learned and predicted by the network. In this article, the authors have even demonstrated the possibility of regenerating the original image from its abstraction.

by example:

Even if it is not directly interpreting the data, this method of explanation derived from human reasoning which also works sometimes by analogy. So, the network will use examples that it has already learned to explain a concept that may be new to him. That's what has been made by [20]. In addition, they also kept the learning data from the network. So, they made the connection between the hidden neuron activations for the examples and for the prediction. This link was made via the k nearest neighbors algorithm. [21] also started from the same principle, but concerning text analysis. So, for a learned word, the model puts it in link with "close" words and it corresponds to find synonyms.

Explanatory text: This method assumes that human beings often justify or explain their decisions with text [22]. Indeed, they first led a model based on reinforcement learning to be able to play a

video game (Mario Bros).
Following this, they constructed another model that will be used to make the link between the state of the previous model and a textual explanation. Additionally, [23] who have also explored this possibility. In this article, we are talking about a system of recommendations:
For example, in an e-commerce website, a product may be recommended for a customer. The idea of this paper, is to predict a grade to a recommendation system, usually assigned by the client. Except that this prediction will be explained textually by another model. This is to improve the model in charge of the recommendations.

Local explanations: The last approach is also quite popular because it does not try to make the connection between the network and all the concepts that it has been able to learn. In general, these are inexpensive techniques in terms of calculation. Indeed, we try to look "locally" the choice of the model. You can do that, through the input data, the activations, the gradient, and so on. This is the case of [24], who set up the concept of *Saliency Map*. This consists in multiplying the gradient of our output which corresponds to the real class, with the input data.

This allows to highlight the inputs that greatly influence the activation of output neurons. They therefore have a huge impact during a change. The following methods are part of so-called local explanations and they all have in common the fact of assigning a relevance score for neurons. The higher the score, the more important the neuron has been in decision making. We will first talk about the LIME (Local Interpretable Model-Agnostic Explanations) method [25] and it is popular because it applies to any type of model such as *Random Forest* or *SVM*.

Thus, it does not rely on the architecture of our network. The overall idea of LIME is first to generate examples that will approximate our input data. This consists of a set of disturbances of our entry because this will make it possible to calculate a more "transparent" model that will approximate another more complex model. Finally, LIME will assign a relevance score to the input variables so this method is also useful for detecting confused data that will "noisy" our learning data.
LIME is popular because it applies to any type of model.

However, for the purposes of this internship, we also wanted to have an interpretation of the network and of the hidden neurons. For this, there are other methods that always calculate relevance scores, but for any neuron. They are therefore specific to deep learning, sometimes specific to one type of architecture, but also very fast. We always consider them as local interpretations because they consider our input variables.
These methods are: Simple Gradient Method [26], LRP [2], DeepLIFT [26], Integrated Gradients [27] and Guided Backpropagation [28]. These methods are part of Gradient-Based Methods.
These methods can also be compared to perturbation methods [29], [30]. This involves disturbing the variables to find out which ones are important. These last two sub-families of methods Gradient-based methods and Perturbations methods will be presented in detail in the following chapter.

3.1.2 Model Interpretation Techniques

Link between the input layer and the hidden layers: First, [31] have developed a method to reduce the black boxes aspect of neural networks. For this, the authors have proposed one of the first methods to understand what neurons hidden layers have learned.

Next, [29] took over this concept to adapt it to convolutional networks. This method is called deconvolutional networks. The goal is to link the input data with the neurons of the upper layers. This allowed them to improve the accuracy of their network but also to see that the depth of a model is important for network performance. Interpretation is therefore also a tool to improve our network.
[19], [32] and [33] have further explored this subject and discovered that the deeper the layers, the more they learned abstract concepts.

Prototype representing an abstract concept: Another approach is to generate an image that would be a prototype of an abstract concept or class. This is the case of [24] who developed a method called Activation-Maximization. The goal is to generate a prototype that will maximize the value of the output neuron. In this way, the "preferred" network image is obtained for a given concept and [33] did the same thing but for each neuron. So, a realistic image can be generated to understand the role of a neuron.

Still in the interpretation, [34] have started from the principle of disturbing some input data through the descent of the gradient to improve the activation of certain selected neurons in the hidden layers. Disturbed inputs can highlight what has been learned.
This method is difficult to classify because in a certain way it makes it possible to interpret a model and not a prediction. However, this interpretation follows a prediction. It is also classified among the model functionality. Following this, an input image is disturbed after a prediction.
This perturbation reveals what the model has learned and the neurons whose activation has been improved. We can thus observe the different concepts learned, which neurons look for in this image.

Learning-oriented interpretation of learning: Another possibility of interpretation is to take an interest in the learning data [35]. This method consists in altering the basic data to understand their importance in learning the model and then in its decision-making.
It also detects examples that can mislead our model. Similarly, it allows to create adversarial example. These are modified data but whose changes are imperceptible by humans eye. However, the predictions are still strongly impacted.

Approximate our neural network by a transparent model: We will be approximating our neural network by a transparent or more interpretable model and it is the case of a linear model or a decision tree.
This has been done in [36], a decision tree is extracted from a neural network to understand what it has learned and it also allows you to extract rules that are used to decide.

3.2 Disturbance method

Disturbance methods were applied by [29]. They consist in directly calculating a relevance score for each variable, by deleting, hiding or modifying the input data.
Following this, we redo a prediction (a forward), in order to calculate the difference with the true prediction. The objective is to measure the impact of a variable on the prediction. More formally, we define x as an example and x_i the ith variable of x. Thus, $Sc(x)$ will be the output value after the forward, for our example x, so we obtain the following equation:

$$R_i = S_c(x) - S_c(x_{[x_i=0]}) \tag{3.1}$$

One of the problems with this method is the computation time needed for a single example. Indeed, the time increase according to the number of variables. For example, for a single image it may take several hours. In order to decrease this time, it is possible to hide several variables at a time. It is applicable in the field of images, because a pixel is directly related to its neighbors.

We cannot do likewise on transcriptomic data because we have not yet determined link between the different variables. These disturbance methods are called $Occlusion - n$, where n is the number of variables corrupted by iteration.

3.3 Gradient based method

Gradient-based methods are also part of the relevant methods. They also allow to assign a relevance score R_i to a neuron.
This is undoubtedly one of the most popular types of interpretive methods. Indeed, they are specific to learning methods and allow to rely on the architecture of the model (layers and neurons), and its instantiation (weight and bias). They rely on the retro-propagation of the value of the output neuron.
Thus, the interpretation of a single example often takes less than a minute. In order to present them, we will try to see them in a coherent order.

3.3.1 Activation-Maximization

In this section, we will explain the *Activation-Maximization* method. All information concerning this algorithm is presented in the article of [2]. The neural network will take a given x input and will then propagate via the different layers and activations of the information neurons to the output layer. Following this, we can calculate the probability that x belongs to the class w_c. This probability is written $P(w_c|x)$.

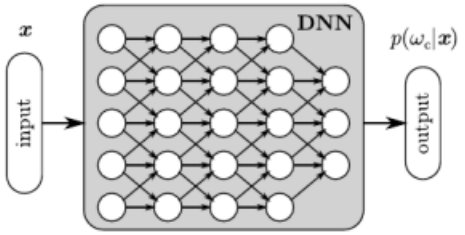

Figure 3.1: Example of a neural network with a probabilistic vision [37]

The goal of the Activation-Maximization method is to maximize the activation of the output neuron of the predicted class. This will increase its probability. The example created following this maximization will be the prototype of an abstract concept. Formally, we consider data x that is associated with a set of classes $(w_c)_c$. We still have our probability $P(w_c|x)$. The prototype x^* will be calculated by optimizing the following equation:

$$\max_x \log p(w_c|x) - \lambda \|v\|^2 \tag{3.2}$$

In the first part of the equation, we find our probability. The term right is a regularization term [38], called *l2-norm regularization*. This is usually a term that helps to avoid overfiting. In this context, it makes it possible to favor inputs close to the origin, even if the final prototype is grayed

out (in the context of image analysis). It is possible to improve the obtained prototype by replacing the regularization term by an expert $p(x)$. It is a model of learning and the equation becomes:

$$\max_x \log p(w_c|x) + \log p(x) \tag{3.3}$$

This method will make it possible to obtain strong class response (the value of the neurons of the output layer) and create a prototype much closer to the data. The problem of the expert is sometimes that it is difficult to make it accurate. In this case, we go through a so-called generative model. The principle is to generate examples from a simple distribution. These examples will then be linked to our data via a decoding function and it corresponds to a prediction model. It sounds complicated, but in reality, it corresponds to the GAN (Generative Adversarial Network) [39]. Indeed, the principle of GAN is to generate data that seems real. For this, we have a model that generates examples and another that validates or invalidates the likelihood of the example. The first model is driven until it can generate plausible data. It's a bit like if the created data passed the Turing test. To illustrate the concept of Activation-Maximization, we can use the following figure:

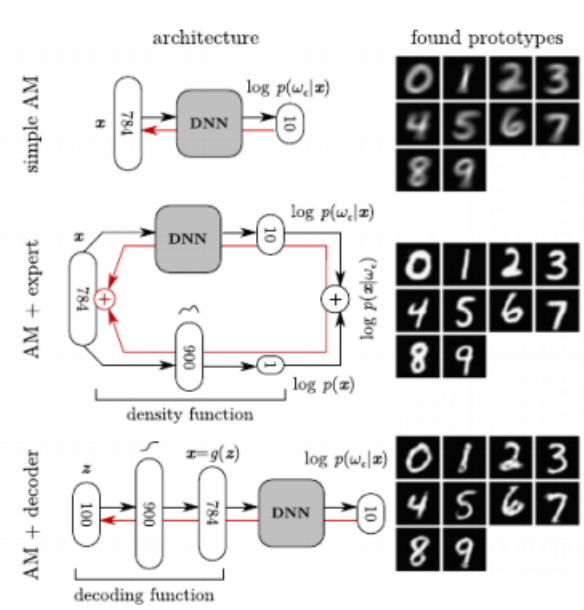

Figure 3.2: Different architectures for the interpretation method *Activation-Maximization* [37]

In Figure 3.2, we can see the three types of architectures possible for our method of interpretation. They have been applied to MNIST data. These are images of handwritten figures and there are ten classes, from 0 to 9. On the left, you can see the architectures, and on the right the prototypes created from them. These are images of 28 pixels by 28 pixels so 784 input variables. The DNN (deep neural network) in grey is the network that we want to interpret. So, we see that the prototypes obtained by simple activation maximization are a little fuzzy, unlike other architectures. The last two have similar results. It must be understood that prototypes are created images that represent an abstract concept, a class, which is here a number. This idea of prototype is here an overall interpretation of the different concepts that the model has learned. Then, when the concept is too abstract it is possible that it makes no sense. It would be more interesting at that time to make local interpretations. This amounts to generating different prototypes for the same concept.

3.3.2 Gradients × Inputs

The Gradients × Inputs method was introduced by [24]. In order to explain clearly what is going on, we will make a formal definition. We take an example x, which can be an image or omics data.
We have the score $Sc(x)$ which corresponds to the value of the output neuron. If we were dealing with a linear model, we could write the following equation:

$$S_c(x) = w_c^T x + b_c \tag{3.4}$$

w_c and b_c are the weights and bias of the model, respectively. In this case, the weights could represent the importance corresponding to each element of x. However, the linearity of the model is "broken" because of the different activations functions which apply to the output of neurons. To approximate our nonlinear model, we use a series of Taylor. Indeed, we use the terms first order:

$$S_c(x) \approx w^T x + b \tag{3.5}$$

where

$$w = \frac{\partial S_c}{\partial x} \tag{3.6}$$

w is our gradient.

3.3.3 Simple Taylor Decomposition

The Simple Taylor Decomposition method was introduced in [40]. It is also explained by [2]. This takes a bit of the principle of the previous method. We have our non-linear model that corresponds to a function f.
The idea is to break down this function into a sum of relevance scores. These scores are obtained following the identification of the terms of a series of Taylor of first order at a root point x_0, such that $f(x_0) = 0$. This root removes the information in the input layer that makes $f(x)$ positive.
Now, we have our example from which we removed the reasons that would be responsible for our prediction. We will call this root point, a reference. The Taylor series allows you to rewrite the function as follows:

$$f(x) = \sum_{i=1}^{d} R_i(x) + O(xx^T) \tag{3.7}$$

where the scores are expressed as follows:

$$R_i(x) = \frac{\partial f}{\partial x_i}\Big|_{x=\bar{x}} \cdot (x_i - x_i') \tag{3.8}$$

$R_i(x)$ are first order terms and $O(xx^T)$ are higher order terms. This method does not use back-propagation. Finally, a variant of applying this layer by layer and which is called *Deep Taylor Decomposition* belongs to this family of methods.

3.3.4 Layerwise Relevance Propagation

Layerwise Relevance Propagation (LRP) [37] is an algorithm that consists in retro-propagating a relevance score of a neuron from the output layer to the input layer.

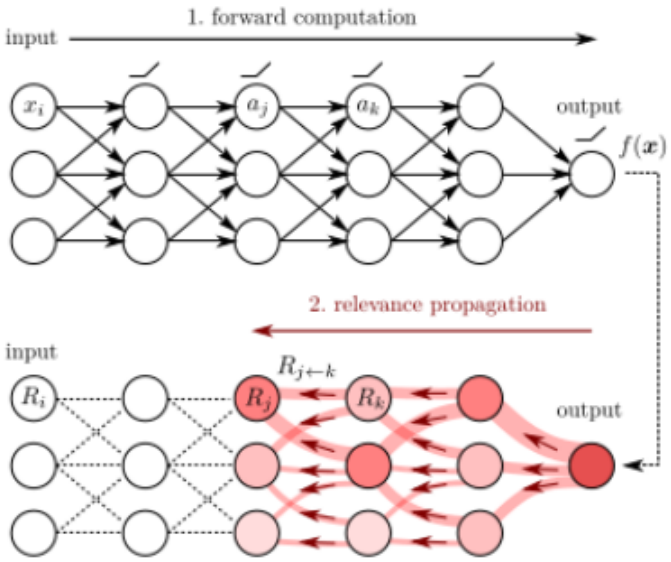

Figure 3.3: Schematic of the LRP procedure [37]

Figure 3.3 schematizes the LRP procedure quite simply. First there is the propagation phase (forward). This one consists in making the prediction starting from the variables x_i. We get the output $f(x)$. This will be our relevance score. This is the one we are going to retro-propagate. This is the relevant spread as shown in the diagram and LRP respects a principle of conservation. Indeed, the sum of the relevance of the neurons of the same layer is equal to the sum of the relevance of the other layers.

We have just seen the LRP principle, but we have not defined how to retro-propagate a relevance score of a neuron to previous neurons. Since the goal is to highlight the relevant neurons, we should not split this score evenly.
We then establish rules of propagations. As seen previously, we can use the Simple Taylor Decomposition method, this will become now the Deep Taylor Decomposition.
However, much more intuitive rules exist: we divide this score according to the activation of the previous neuron and its associated weight. This neuron will be more rewarded if it participates more than the other neurons of its layer. More formally, take two layers j and k. We can define the following propagation rule:

$$R_j = \sum_k (\alpha \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} - \beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-}) \tag{3.9}$$

with R_j the total relevance of the j layer, a and w are the activations and weights, respectively. $()^+$ and $()^-$ are the positive and negative weights between the j and k layers. This distinction makes it possible to highlight the neurons that go in the direction of prediction and those that go against it. You should also know that $\alpha - \beta = 1$ and $\beta \geq 0$. We can add terms to avoid any division by 0. This rule is called $\alpha\beta - rule$. We modify α and β according to the application domain. Some variants are suitable for the input layer and others for the hidden layers. The notation $LRP - \alpha_n\beta_m$ is used when assigning the values n and m respectively to α and β. It has been proved that $LRP - \alpha_1\beta_0$ corresponds to the *Deep Taylor Decomposition* method. For more information on the theory as on the implementation, one can refer to the document of [2].

3.3.5 DeepLIFT

DeepLIFT is a method introduced in the article [26]. It was thought after highlighting some defects of disturbance methods, but also that based solely on the gradient.

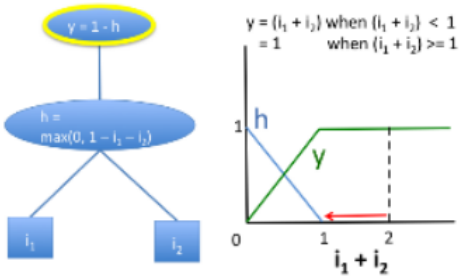

Figure 3.4: Example of failure of disturbance or gradient-based methods [26]

In Figure 3.4, we have an example where previously seen methods fail. Indeed, in the context of an $Occlusion - 1$ perturbation, no change is produced and it is impossible to know which of i_1 or i_2 has contributed the most. Similarly, the gradient will be 0 when $i_1 + i_2 > 1$.

To solve this problem, DeepLIFT will do a mix the disturbance and LRP methods. The main idea is to calculate the difference between our example and a reference. Note t the activation of our example, and t^0 that our reference. We calculate δt which is the difference between the example and the reference: $\delta t = t - t^0$. Following this, we take the principle of LRP and retro-propagate this difference to previous layers. This is done via propagation rules.

After that, because of the number of rules specific to the types of layers, it is better to refer to the [26] document. The choice of the reference can be complicated in some cases. In that of the MNIST data, a white image is used. So, the difference between the example and the reference, will allow to focus only on the number. For text analysis, it will be an empty string and for omics data it is not so simple. The choice of this reference is decisive for the final interpretation. In this way, DeepLIFT solves the problems of discontinuities encountered with the gradient methods.

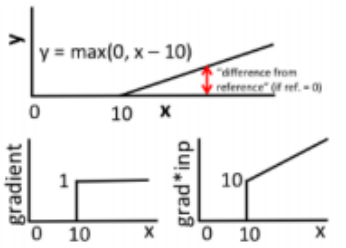

Figure 3.5: Example of gradient discontinuity [26]

In Figure 3.5, we take the activation of a neuron with a ReLU activation function, with a bias of -10. That is the output of our neuron is $y = \max(0, x - 10)$. It is only when $y \geq 0$ that the relevance score attributed to the input neurons will be positive. We note that the relevance score will be discontinuous for the gradient, gradient $\times$ inputs, unlike the difference from reference, represented by the red arrow. As you may have noticed, this method is classified in the category of gradient-based methods whereas it is seen as an improvement.

However, in the literature DeepLIFT is considered to use what is called a "discrete" gradient [27]. Moreover, to avoid complicating the field of interpretation more than it already is, I chose not to create another sub-category among the relevant methods.

3.3.6 Integrated Gradients

The Integrated Gradients method [27] was created via an axiomatic approach. The designers of this method first defined the axioms that must be respected. To understand the different axioms, it should be noted that Integrated Gradients use just like DeepLIFT a reference.

The first axiom is called *sensitivity*. It consists in assigning non-zero relevance when an example and its reference differ in one variable, and the prediction is different. This has been highlighted in the article on DeepLIFT [26] and explained in Figure 3.4.
The second axiom is the *invariance implementation*. Formally, two networks are functionally identical if for each input we obtain the same output, even if they do not have the same architecture. It must be the same for the relevance attribution method. For two functionally identical networks then one must obtain the same interpretation.

Integrated Gradients therefore respects these two axioms. Now let's define how this algorithm works. We have x our example and x_0 our reference. The reference is sometimes called baseline and the model is represented by a function F. For calculate the i-th dimension of x (a variable), we apply the following formula:

$$IntegratedGrads_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \tag{3.10}$$

Intuitively, Integrated Gradients is the sum (via the integral) of gradients for all the points that belong to the "path" between the x example and its x' reference. Defined as this, the method is not necessarily very understandable, but especially it is very time consuming. There is the approximation of Riemman and we sum the gradients at regular intervals (m step).
We find here our sum of gradients for all possible paths because of that, it is moreover for this reason that we also speak of *path methods*.

3.3.7 Guided Backpropagation

Guided Backpropagation is not an improvement of other methods and this algorithm was introduced briefly by [28]. This approach is a gradient-based visualization technique designed to highlight what parts of the input contribute to a given neuron in a neural network. The method back propagates the gradient with relation to the input image while masking negative values. This results in only positive gradients being conserved. The advantage of retaining only positive gradients is to prevent a backward flow of negative signals corresponding to neurons which inhibit activation of the higher-level neuron.

As opposed to usual backpropagation, this can act as an additional guidance signal when traversing the network. The output of guided backpropagation is an RGB image of the same dimensions as the input image. This method works for visualizing neurons in convolutional as well as fully connected layers. It is explained via the following schema:

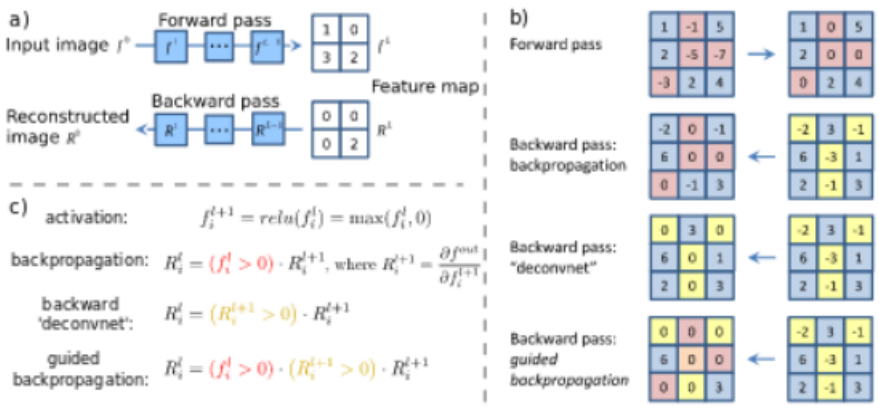

Figure 3.6: Diagram of Guided Backpropagation [28]

In Figure 3.6, we can see that the algorithm is inspired by the techniques of back-propagation and deconvnet [29]. The only thing to understand is that the allocation of the relevance score fllows that reasoning: If the relevance of the superior neuron is positive and the activation of the current neuron is also positive, then the relevance of the current neuron is equal to that of the higher neuron.
It's finally a simplistic method and from the experimentation that we have done, it comes down in the majority of cases to do:

$$R_i = \max(0, gradient \times input(x_i)) \tag{3.11}$$

We call this method guided backpropagation because it adds an additional guidance signal from the higher layers to usual backpropagation. This prevents backward flow of negative gradients, corresponding to the neurons which decrease the activation of the higher layer unit we aim to visualize.

3.4 Comparison of interpretation methods

In this section we will first summarize the different methods. It must be understood that there is not one better than the other, in some cases, we will favour a method.

Methods	Reference	Sensitivity	Implementation Invariance	Fragile
Occlusion-1	No	No	Yes	No
Gradient x inputs	No	No	Yes	Yes
LRP	No	No	No	Yes
DeepLIFT	Yes	Yes	No	Yes
Integrated Gradients	Yes	Yes	Yes	Yes
Guided Backpropagation	No	No	Yes	Yes

The different criteria come from the article by [27]. There are some errors, as it is said that LRP uses a reference and which is not the case. So LRP cannot respect the *sensitivity* axiom. We have a proof that LRP returns to the same thing as gradient $\times$ inputs [26].

Methods	Architectures	#Forward-Backward
Occlusion-1	All	n (number of variables)
Gradient x inputs	All	1
LRP	All	1
DeepLIFT	All	1
Integrated Gradients	All	m steps (generally 50)
Guided Backpropagation	ReLU Networks	1

The proof is based on a single rule of propagation, while there is almost a dozen. The constitution of a synthesis of the different methods quickly becomes tedious. The fragility of interpretation is discussed in the article by [41]. This fragility is defined by a different interpretation for a disturbed example whose disturbance is invisible to the naked eye. This problem is put forward as a security breach and especially in image analysis. Disturbance methods are not affected by this problem because of their nature.

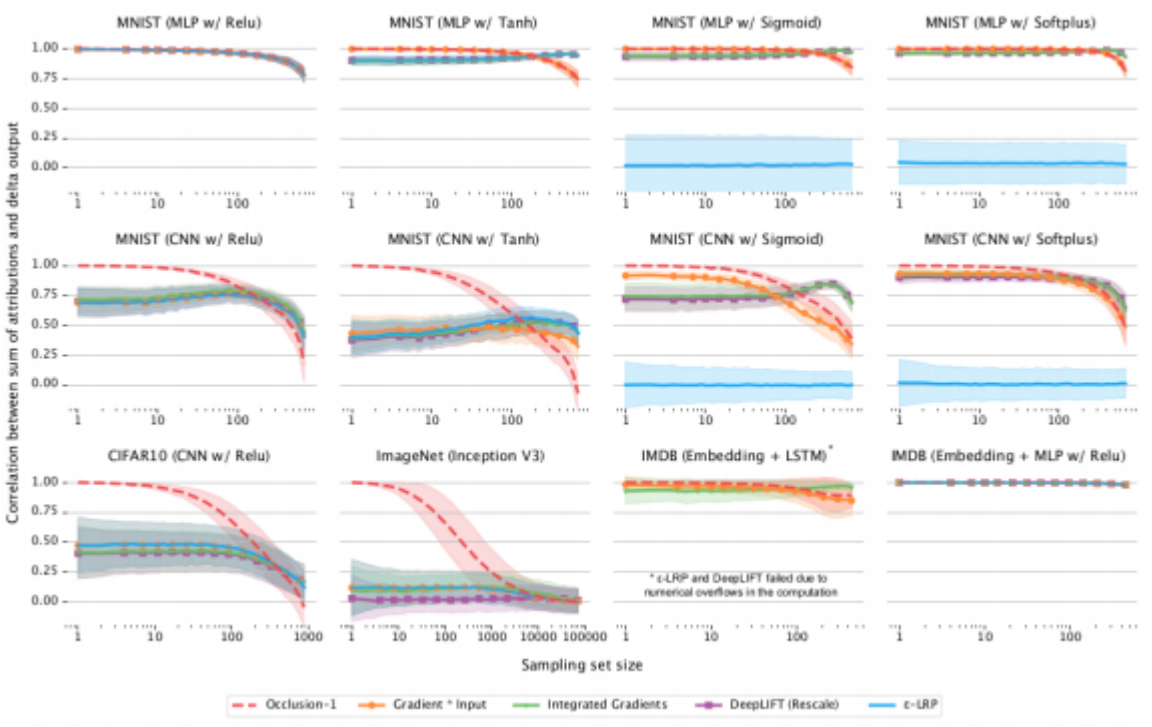

Figure 3.7: Comparison of the correlations between the sum of the attributions of relevance and the variation of the target class (delta output), applied on images [42]

In Figure 3.7, the different relevance score assignment methods have been applied on images (MNIST, IMDB, CIFAR10 and ImageNet). They have been tested on different architectures: MLP, CNN (convolutional neural network) and LSTM (long short-term memory). The image compares the correlations between each method.

Chapter 4

InOmicNet

4.1 Introduction of the method

InOmicNet is a shortcut for "Interpretable Omic Network", the purpose of this memoir is to develop a neural network able to learn and evolve the weights of the following layers by biological information introduced with a knowledge base outside.

The first step of the method will therefore consist of extracting the knowledge from the set of probes, we must recover the biological functions associated with these probes. We will use the available base package Bioconductor [43] that will allow us to recover first biological functions. We will also note the loss of information related to the search for biological functions and also according to the current gene ontology.
Then we will define a directed acyclic graph from this one and evaluate the incoming and outgoing nodes as well as the number of functions at each depth of the graph in order to get an idea about the number of hidden layers to use.

After defining the number of hidden layers for the neural network, we will introduce a custom cost function. This graph will be put in the form of an adjacency matrix that will serve both the interpretation of the network a priori but also as a constraint to the cost function. This custom cost function will consider the ascendancy of the biological functions in order to better learn from this external information, and we will also implement a callback monitoring system that will allow the evolution of weights and losses according to the l_1-norm and l_2-norm.

It is important to note that the technique allowing interpretation is included in InOmicNet. As discussed there [44], most recent work on interpretability of complex learning models has focused on estimating a a posteriori explanation for previously trained models around specific predictions. Self-explaining models where interpretability plays a key role already during learning have received much less attention. To give a correct value, we will carry out a verification of the evolution of the penalties in order to see if the network considers its information with also a grid search of the regularizing coefficient α.

For the interpretability part, we will use the LRP and with the assistance of a simple dense network with the predefined layer number, we will copy the weights of this one in order to use the package and check if the scores in a first follow a Gaussian distribution.
And finally, a series of several techniques such as taking the best score per layer for each prediction and look at what biological functions are and other ideas that we will develop later.

4.2 Gene Ontology

Ontologies as they are used in computer science (because the concept is philosophical above all) were first developed for artificial intelligence. Their purpose is to describe what exists and the formal definition is quite complex.

In its definition, an ontology is a structured set of terms and concepts of a domain, specifying the relationships between these terms and their properties. Each term in an ontology must have a definition to be sure of the meaning associated with it. An ontology has a hierarchical structure and the set of terms is anchored by a high-level term, the root.

For example, the ontology that must be the best known in biology is Gene Ontology. It focuses on describing what genes and their products are with 3 different namespaces (3 subparts of the ontology): molecular functions, cellular compartments and biological processes.

It is widely used for the annotation of genomes in order to harmonize the characteristics associated with genes whatever the species. Then, again thanks to this vocabulary, when one needs to search for a gene or function, it suffices to use the corresponding GO Term. Gene Ontology is used by many software and databases that need concepts on genes and it makes it easier for users to use them because they only need to know this reference.

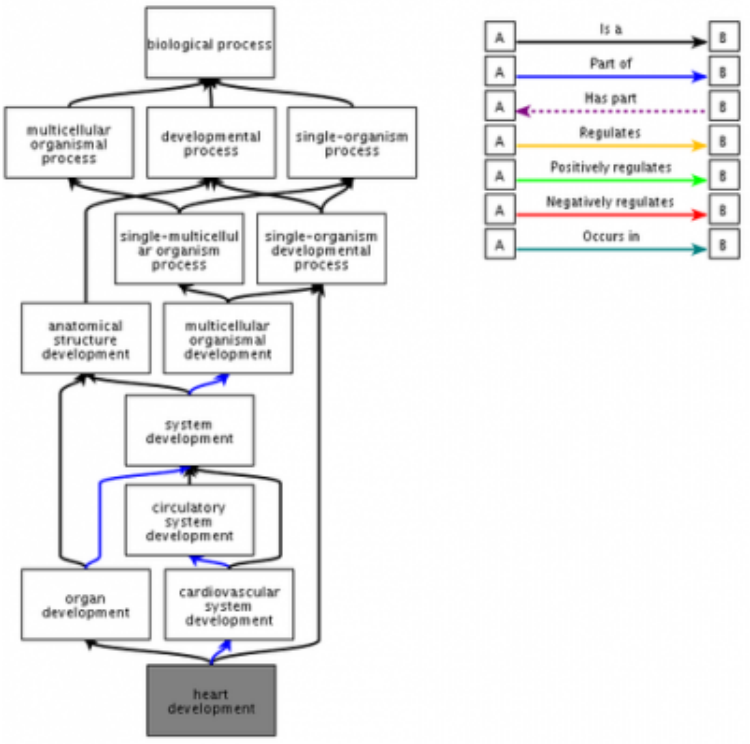

Figure 4.1: Illustration of the different part of the gene ontology.

The goal of the Consortium is to produce a structured, precisely defined, common, controlled vocabulary for describing the roles of genes and gene products in any organisme. GO terms are connected into nodes of a network, thus the connections between its parents and children are known and form what are technically described as directed acyclic graphs. One reason for this is that state of biological knowledge of what genes and proteins do is very incomplete and changing rapidly.

We need to be able to organize, describe, query and visualize biological knowledge at vastly different stages of completeness. Any system must be flexible and tolerant of this constantly changing level of knowledge and allow updates on a continuing basis. The vagueness of the word feature when introduced to genes or proteins was identified as a specific issue, as the word commonly used to define biochemical operations, biological functions and cellular structure.

4.2.1 Extraction of knowledge

Our work is to be used in medicine and therefore must be precise. We will argue on the next chapter that despite the loss of information, in the end we have enough to carry out the methodology.

I basically worked on *Jupyter Notebook* (*Jupyter* is a web application used to program in more than 40 programming languages and it is an evolution of the *IPython* project and, it allows you to make notebooks) using the famous programming language used in the field of data science domain: *Python*. The worry is that the bioinformatician prelims language is *R*, so I used *rpy2* to make a binding between *R* and *Python* and allow the efficient and fast use of other packages specific to *Python*. To retrieve data from the Affymetrix HG-U133Plus2 device, I had to use the **hgu133plus2.db** package available in the *Biocmanager* library.

From this starting point, the library allows us to extract the functions *GO*, *ENTREZID*, *ENZYME* and all the *PROTEINS* associate with a probe. So we have 54.675 probes identifier at the start, but when we want to find the genes associated with the probe identifier, we realize that 18,89% are lost and that finally according to the [45], the 23.437 genes are not all recognized because there are 1.435 missing genes (a loss of 6,12%).

Then I coded a routine to retrieve all GO functions associated with these probes as inputs, returning 8.198.615 functions and when we only filter the biological functions (BP), we have 5.474.170 functions. With these biological functions, we can associated 36.587 unique probes. This number is interesting because we retrieve 16.037 GO biological process functions which will be the size of our future adjacency matrix at the beginning. The loss of genes information is about 27,34% because we can fetch only 17.027.

We download the gene ontology directed acyclic graph from *OBOLibrary* official website, we will use two *Obonet* and *Networkx* libraries to manipulate the *obo* file as well as the directed acyclic graph. There is (at the time of writing this memoir) 45.006 nodes (GO function) with 92.809 arcs. We retrieve the gene ontology functions extracted from the library previously (16.037) and then perform a simple operation: we subtract the nodes not included in our list of GOID GO base graph.

Then there is another problem, the graph of GO functions is updated regularly and therefore the information retrieved by the package **hgu133plus2.db** is obsolete, so 26 GO functions are not known and we remove them from the graph. In addition, it was necessary to add two arcs.

```
    list_go = file_goall.GOALL.unique()
    list_nodes_useless = list(graph.nodes - list_go)
    list_go_old = [x for x in list_go if x not in graph.nodes]
    list_go_uptaded = [x for x in list_go if x not in list_go_old]
    graph.remove_nodes_from(list_nodes_useless)
    graph.remove_nodes_from(list_go_old)
    graph.add_edge("GO:0099180","GO:0071577")
    graph.add_edge("GO:0036520","GO:0007267")
```

Finally, we have a directed acyclic graph containing 16.011 nodes with 37.912 arcs, an incoming and outgoing average of 2,3679. From this moment on, we retrieve several informations, for

example the fact that we have a loss of 33,08% of incoming probes. Since the number of input variables is very important despite this loss of information, we have decided to ignore future probes that we will not be able to interpret. Here is a summary table including the depth of the graph relative to the root **GO:0008150** with is associated to term BP (*biological process*).

d°_mēan_in	*d°_mēan_out*	*mēan_probes*	*nb_envidence_associated*	*nb_go*	*Depth*
0.0	29.0	36587.000000	114569	1	0
1.172414	15.827586	1248.448276	556058	29	1
2.054830	7.563969	94.477807	1039316	383	2
2.582438	3.664643	16.754787	1396427	2141	3
2.610091	2.550564	7.824297	1270356	4519	4
2.448291	1.851227	7.399430	653485	4564	5
2.049626	1.705396	11.908625	317339	2539	6
1.879163	1.381875	14.103021	91318	1291	7
1.900709	1.115839	22.943262	27050	423	8
1.359223	1.048544	31.883495	6098	103	9
1.055556	0.666667	24.277778	721	18	10

Then we generate two CSV, the first contains for each function GO its associated depth and for each probe its associated depth in the graph. This first step is used to build an information model that will be used to build a model of neural networks and to guide learning later.

4.3 Learning constraint

Our model is a self-explaining one where interpretability is built-in architecturally and enforced through regularization. This learning constraint is useful because, since we add meaning to each neuron in our network, it is nothing more than an abstract mathematical value, including a weight and a bias, but a biological function originating initially from the probes.

The distribution of GO functions through the depths of the graph follows a Gaussian distribution, so we decided to take 5 layers (including 2539, 4564, 4519, 2141, 383 neurons), there are obviously the 54.675 neurons in inputs and the 2 neurons output.
During implementations, we use custom layers because we implement some constraint learning with an introduction of a connective matrix and also a custom loss function for learning. During my internship, there was the release of Tensorflow 2.0 with basic Keras included as well as other functionality, one should not mix. Tensorflow 2.0 introduced many things like eager execution and a better use of RTX graphics card due to CUDA 10 which was an occasion to use the new server with 4x RTX 2080Ti.

Construction of the connection matrix: To render the network interpretable, we use an adjacency matrix constructed with *Networkx* (NetworkX [46] is a *Python* suite for creating, manipulating, and studying the design, dynamics, and features of complex networks.) containing for each GO function the ascendancy to the node in the graph.
It will help us during the learning to adapt weight of neurons provided by the gene ontology. In input, we have the 54.675 probes and from them we build the matrices following:

- 54675x2539
- 2539x4564
- 4564x4519

- 4519x2141
- 2141x383
- 383x2

As can be seen, there is a big drop between the number of input variables and the first hidden layer. In the model summary A, it follows that there are about 138 million parameters. Then the number of GO functions decreases to 383. Number of parameters from the batch normalization are 56.584.

Custom Loss: We use a custom cost function for each layer of the neural network:

$$-\left(x\log(p)+(1-y)\log(1-p)\right)+\frac{\alpha}{\lambda}*\|\mathbf{W_i}*\mathbf{C_i}\| \tag{4.1}$$

α represents the weight of the l_2 regularization and λ is the number of layers then we realize the product of the weight matrix W_i and the connectivity matrix containing the biological information defined with C_i.

Normally, we use l_2-norm only on the weight matrix when we want to train the network to the classification of cancer cases and the results are around 95% on the validation set.

$$-\left(x\log(p)+(1-y)\log(1-p)\right)+\frac{\alpha}{\lambda}*\|\mathbf{W_i}\| \tag{4.2}$$

4.4 Monitoring

When learning the model, we want to set up a complete monitoring system that will allow us to have a visualization of the results in a clear way and it will allow us to have a verification system in terms of coherence. Let C_i, the connectivity matrix for a i layer then we have two types of penalties:

- *go* : penality C_i.
- *no_go* : penality 1 - C_i, the one used for learning.

We also follow the weight values for each layer by stocking with the l_1 norm and the l_2 norm. It was initially chosen not to include the connectivity matrix in the l_2 norm but we then chose to consider because the weight of this one was not very important and moreover it is necessary to take into account that the matrices are very separate. We could have used the l_1-norm which highlights the sparsity of the matrix ahead, but it does not have any consequences in the end on the learning and the evolution of the weights of penalties.

$$\|\mathbf{W_i}\| \tag{4.3}$$

We perform a custom loss monitoring according to the *go* or *no_go* penalty of each i layer:

$$\frac{\alpha}{\lambda}*\frac{\|\mathbf{W_i}*(1-C_i)\|}{|\ (1-C_i)\ |} \tag{4.4}$$

Then the last monitoring according to the l_1-norm, l_2-norm and *cr* for cross-entropy.
Then:

$$-\left(x\log(p)+(1-y)\log(1-p)\right)+\frac{\alpha}{\lambda}*loss\_penality\_norm \tag{4.5}$$

We retrieve the values at the beginning of each training session and at the end of each epoch, in order to see later the evolution of the different values that we wish to follow. This system allows us to validate the methodology and then adjust the formula later. Our methodology must be rigorous in our case.

Chapter 5

Results

In this section, we will discuss the different results acquired during the various implementations realized on the InOmicNet network. We will first see the results on the simple classifier and InOmicNet to see if there is a difference in accuracy or not. Then we will carry out a study on the regulator coefficient in order to see when and how the network considers the secondary terms of the loss and also the consequences on the weights of the penalty. Finally, we will make a study on the different results of the penalties scores of relevance.

5.1 Input data

In the case of learning methods, it goes without saying that the importance of data is primordial. Indeed, in biological research, the data are a very rare commodity because it is very expensive to make biological pre-treatments on patients and ethically questionable.
It is indeed necessary to go through a selection of people who are cancerous or not, then to carry out the administrative procedures for the treatment running tests. Moreover, the length can vary between several months or several years depending on the cases treated. In our case, they compiled a human gene expression dataset from approximately 40.000 publicly available Affymetrix HG-U133Plus2 arrays [45].

The creation of this dataset has required a lot of work, and it remains that it weighs several gigabytes. After strict quality control and data standardization, the data were quantified in the expression matrix of 20.000 genes and 28.000 samples. The assessment recognized 1.285 genes with targeted expression of genes changes in cancer.
Raw Affymetrix HG-U133Plus2 device function information was recognized and accessed from the ArrayExpress data archive. The initial data set consisted of 40.871 CELs from 69 experiments and included samples from a variety of normal and non-healthy cell and tissue types as well as cell lines. Replicated information documents have been recognized by comparing document dimensions and MD5 checks and deleted.
54.675 probe sets (referring to 23.437 genes) and 27.887 samples, continuously annotated by EFO, but introduced more data from other databases.

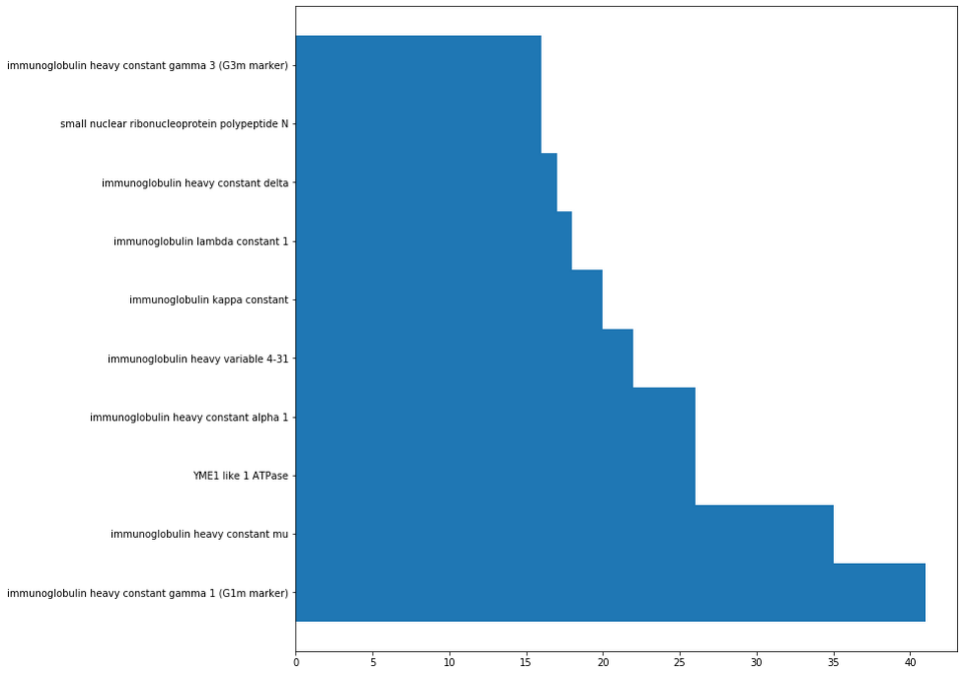

Figure 5.1: Top 10 Genes from the probes.

In Figure 5.1, I extracted the 10 most frequent genes from the probes in entries for information for reader informed.

5.2 The Efficiency Of The Classifier.

We will compare the precision of the model according to the function cost to use or the value of the coefficient of regularization. All the plots are available at the end of this memoir on appendix. The different exercises of the models are carried out with a learning step of 10^3 and an epoch number set to 100. We will use 3 values of α such as 10^6, 1, 10^{-5}. Taking the case where α is at 10^6 5.2, we can see that the precision is locked at 66% and this is due to the ratio of the data set containing 66% belonging to the cancer class with a conventional l_2-norm cost function. The α is too big and the number of epoch low, so it is normal that the model cannot learn because the cost function is not balanced.

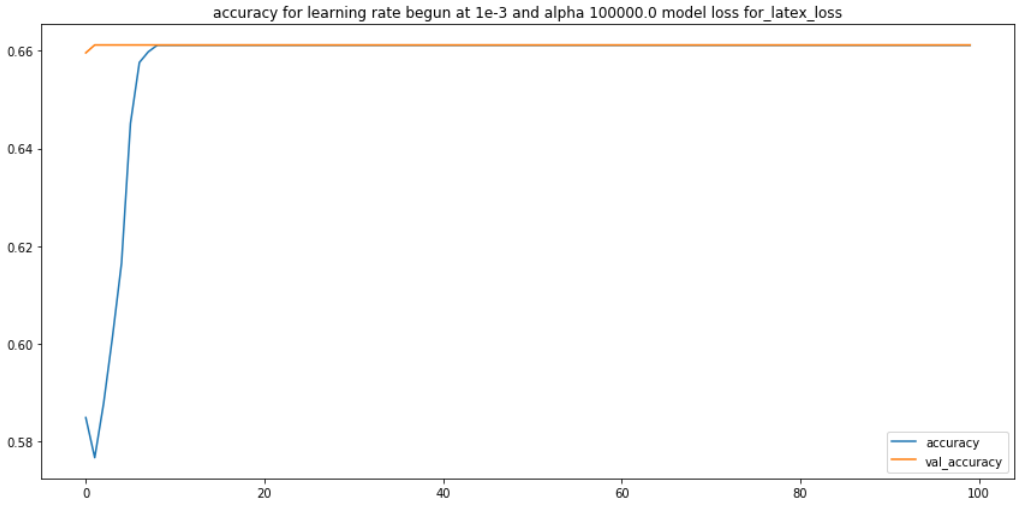

Figure 5.2: Accuracy for $\alpha = 10^6$ with l_2-norm.

Whereas on the contrary with the cost function considering the information of gene ontology, one reaches 90% E with the validation set. This comparison shows that this cost function is more adapted to our problem and is better able to model the problem we are confronting. The second case when α is 1, we have two similar graphs, that is, the precision on the dataset is around 95% even though we can see some instability in the growth.

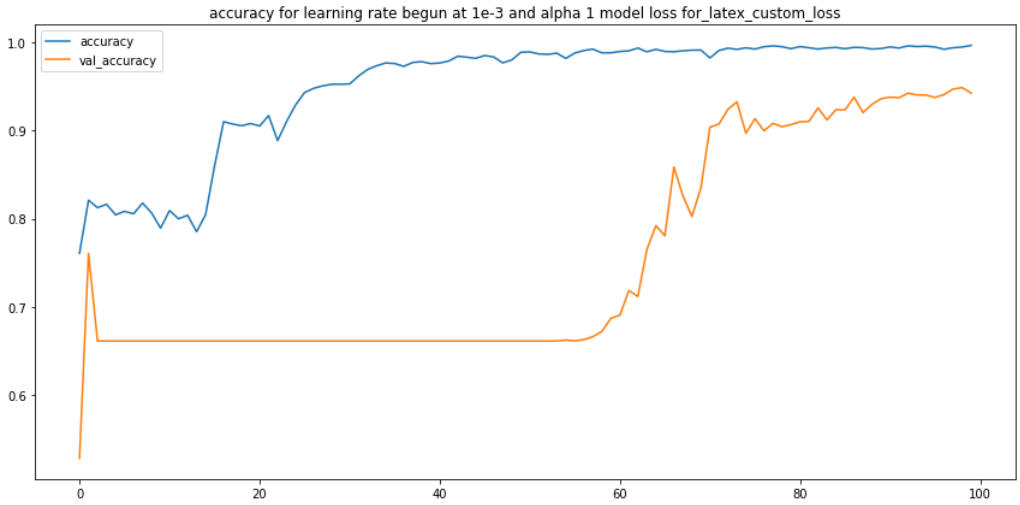

Figure 5.3: Accuracy for $\alpha = 1$ with Custom Loss.

Finally, for a low value α at 10^{-5}, we notice that the progression of the precision is stable and that the difference of the loss formula has no effect, which is explained by the fact that in both cases the cost function approximate cross-entropy.
By studying the different graphs of the cross entropy we realize that when the α is large then the model tends to reduce the other part of the loss formula, it means that the implementation of the model is correct and that monitoring is a necessary development step to prove the stability of the model.

5.3 Variations Of The Regularization Coefficient.

In this section, we are interested in knowing for which values of α there is a consequent variation. Since we have very sparse matrices, I made this method using the l_1-norm because it looks attentively at this problem. Then I generate values from 10^{-7} to 10^{7} in steps of powers of 10. As can be seen there is a tangent to 10^{-4}, which means that any training smaller than this value is not good enough, whereas for 10^{-5} and others the penalty is taken into account because it tends to zero.

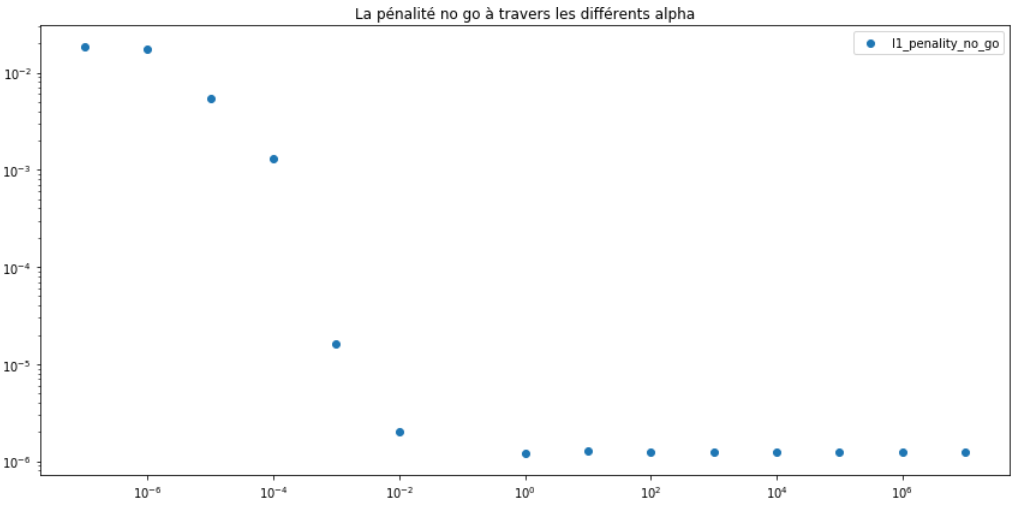

Figure 5.4: l_2 NO GO penalties across the different α.

This simply means that when learning the model with the custom cost function, the model considers the information from the non-ascendancy biological function ontology and it helps reducing the error and assigning the weights of the neural networks according to the prediction of the network.

5.4 Penalty Weight Curves.

We have two types of penalties as we have already discussed before, and we will see the results of the feedback following the various trainings.
Every plot described are available at the end of this memoir. We start by studying the GO penalty, normally the values of these weights should not decrease some values of α and this is confirmed through the different figures as can be seen.

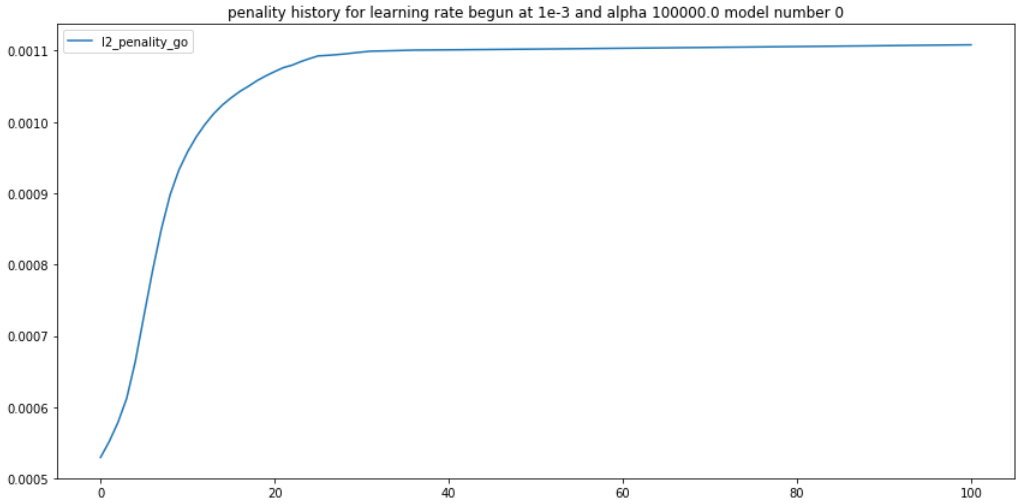

Figure 5.5: l_2 Penalty GO when $\alpha = 10^6$ with Custom Loss.

When the α is 10^{-5}, we see a slight decrease because the second member of the formula is not well enough considered.
However, for the *no_go* penalty, we can see that the implementation is correct because when the α is 1 and 10^5, we have a curve that tends to 0 very quickly.

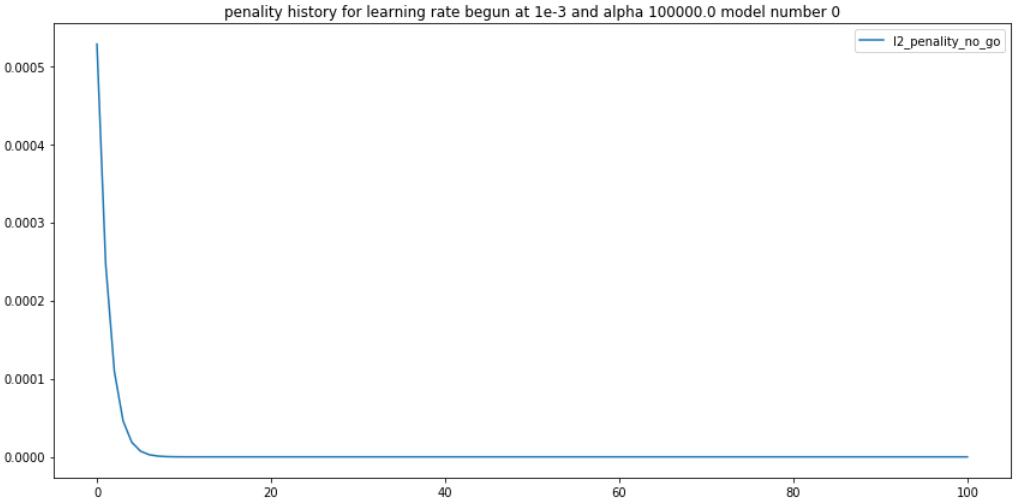

Figure 5.6: l_2 Penalty NO GO when $\alpha = 10^6$ with Custom Loss.

5.5 Relevance scores.

For this part, we will study the results obtained in return for a perturbation technique. Here, there is two informations like the fact that we use the package *innvestigate* [47] then for the technique we use *LRP* with an ϵ at 10-7. The distribution of the scores by layers follows a Gaussian, for each graph we distinguish the class cancer (1) and the healthy class (0).

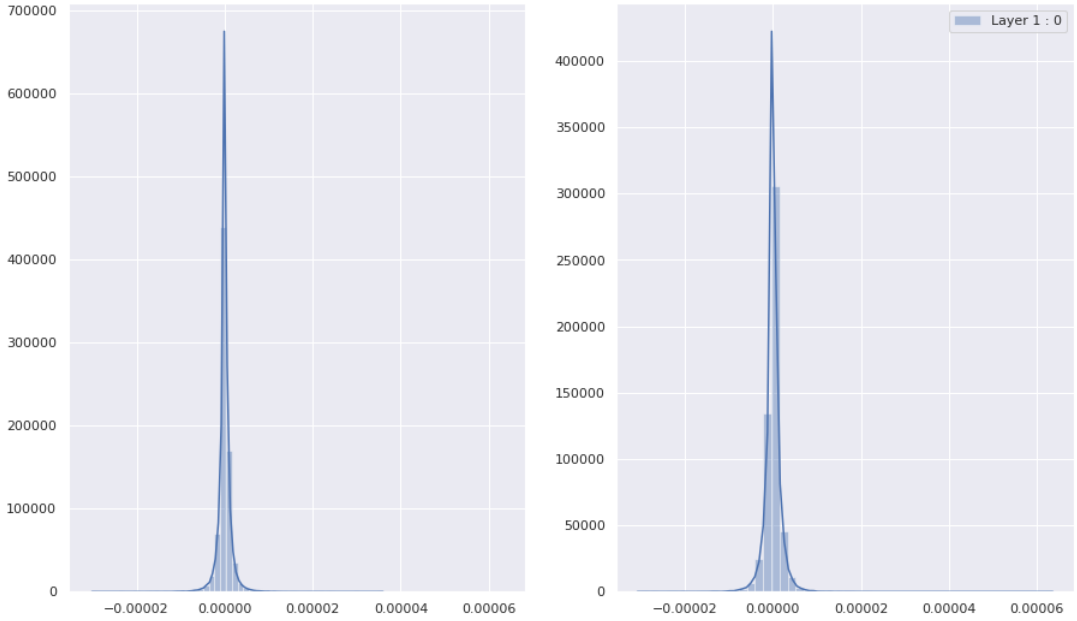

Figure 5.7: Distribution of relevance score for layer 1.

It is interesting to see the scores of layers 1 have scores at 10^{-5} while for layers 6 and 2, we are around 10^{-3}. For layer 3 and 4, we are around 10^{-4}.
We found that the amount of parameter per layer alone is not sufficient to explain these results. Regarding the standard deviation of the distribution by layer and by classes, several are distinguished. For layer 1, the so-called normal class seems to vary more than the sick class and the same reasoning for layer 2, there is a slight greater variation. Then for layer 3, we see that there is clearly a larger standard deviation, scores for the normal predicted class and the same for layer 4.
For layer 5, the opposite happens, the sick class has a bigger variation whereas the layer 6, has a bigger defeat for class 0.

Chapter 6

Biological Interpretation

6.1 Top 10 biological information by layers

In this section, we get the top 10 frequencies when the network predicts cancer for each layer. The goal is to find out the presence of biological functions related to cancer. This will tell us if the network globally makes predictions of cancer cases on functions well related to our problem.

For the input layer, we have the probe **210729_at** which stands out for neuropeptide Y receptor Y2) and it is a member of the neuropeptide Y receptor family of G-protein coupled receptors. Neuropeptide Y (NPY) was first identified from porcine brain in 1982 [48], and plays its biological functions in humans through NPY receptors (Y1, Y2, Y4 and Y5).
NPY receptors are known to mediate various physiological functions and are involved in multiple human diseases in most human diseases, such as obesity, hypertension, epilepsy and metabolic disorders. This review [49] summarizes the current knowledge on the expression incidence and density of NPY receptors in various cancers, the selective ligands for different NPY receptors subtype and their application in cancer diagnosis and treatment.

For layer 2, the function **GO:1905271** is called the proton-transporting ATP synthase activity.
As we do some research [50], we can see that they talk about how V-ATPase dysregulation contributes to cancer growth, metastasis, invasion and proliferation with the potential link between V-ATPase and the development of drug resistance.

For layer 3, we have in first place the function **GO:0071679** corresponding to commissural neuron axon guidance.
We found one paper [51], who said that recently they found several axon guidance genes, including Netrin (Net) and Deleted in Colorectal Cancer (DCC), have been implicated in human cancers.

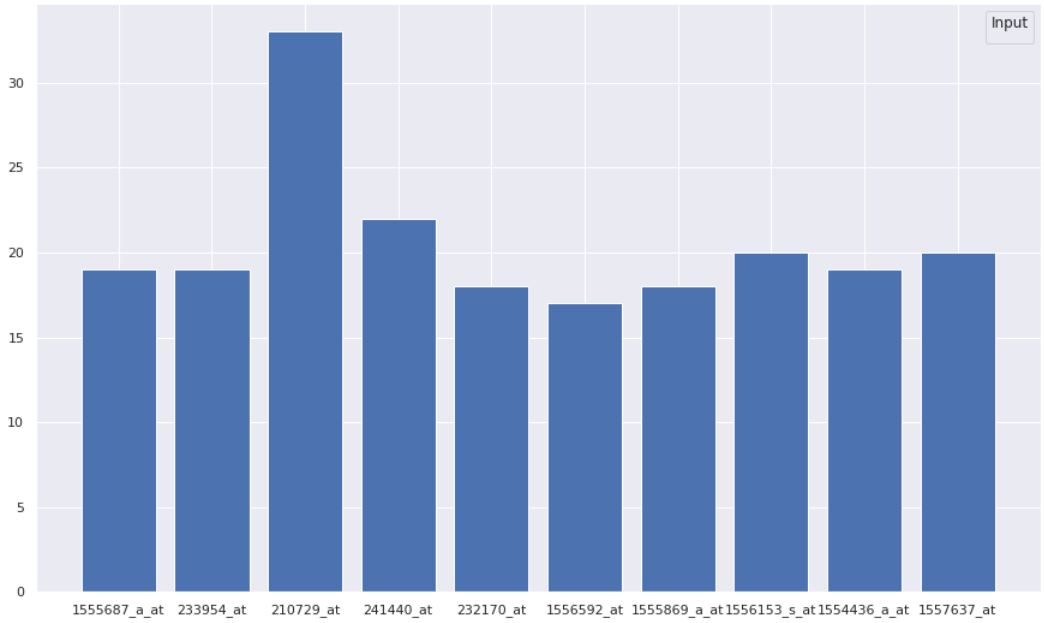

Figure 6.1: Top 10 GO functions on Layer 1.

For layer 4, **GO:0003404** is optic vesicle morphogenesis and it is about the developmental process pertaining to the formation and shaping of the optic vesicle. This process begins with the specific processes that contribute to the appearance of the vesicle and ends when the vesicle has evaginated. The optic vesicle is the evagination of neurectoderm that precedes the formation of the optic cup. We can't rely that directly to cancer.

Next for the layer 5, the biological function that has the greatest frequency is **GO:0010259** and it is associated with the name of multicellular organism aging and its description is an aging process that has as a participant in a whole multicellular organism.
It includes loss of functions, as well as disease, homeostasis, and fertility, as well as wear and tear. Multicellular organisms aging includes processes like cellular senescence and organ senescence but is more inclusive.
In this paper [52], it is mentioned that the diverse cell types are required for multicellularity, but also created liabilities in the form of points of vulnerability that when mutated or dysregulated facilitate the development of cancer.

Finally, for the layer 6, the biological function **GO:0051606** is associated with the stimulus detection action and the series of events in which a stimulus is received by a cell or organism and converted into a molecular signal and its definition is the series of events in which a stimulus is received by a cell or organism and converted into a molecular signal.

6.2 Biological functions used for prediction

6.2.1 Best average neurons

We carry out a new analysis with *innvestigate* using predicted cancer cases, we seek to know which are on average the best neurons that have the highest score in absolute value. At the end of this reasoning we have this feedback: **220842_at**, **GO:0071898**, **GO:0048469**, **GO:1903925**, **GO:1904019**, **GO:0014821**.

For the probe, the official name is *AHI1* and it means *Abelson Helper Integration Site 1*. This gene is apparently required for both cerebellar and cortical development in humans. It mutation causes specific forms of Joubert syndrome-related disorders [53]. Alterations in Wnt signalling have long been linked to numerous cancer types, including breast, prostate, lung and colorectal.

For the first hidden layer, there is **GO:0071898** corresponding to any process that modulates the frequency of estrogen receptor binding. The main pathology related to a malfunction of estrogen and its receptors is hormone-dependent breast cancer. Nearly 60% of breast cancer tumours initially have estrogen-dependent growth.

Then we have the second hidden layer mainly associated with the biological function **GO:0048469**, it is associated with cell maturation corresponding to a developmental process, independent of morphogenetic (shape) changes, that is required for a cell to reach its fully functional state.
A concern for cell differentiation may be involved in the development of cancer.

For the following biological function **GO:1903925**, it represents any process that results in a change of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of a bisphenol A stimulus. There is some research about his link with cancer [54] where they say it is dangerous for the health [55]. Bisphenol A (BPA) is present in many commonly used products. It is one of the constituents of rigid plastics such as polycarbonate. The ability of BPA to mimic estrogen is well documented.

After that we have, the biological function corresponding to epithelial cell apoptotic process (**GO:1904019**). An epitheliocyte is an epithelial cell grown in vitro and apoptosis is a complex, tightly regulated and I found one paper related with cancer and this biological function [56].

Finally, the function **GO:0014821** determines a process in which force is generated within a smooth muscle tissue, resulting in a change in muscle geometry.
It is interesting to note that for the last layer, we are facing a biological function related to the muscle. Here we can maybe see the intuition behind the neural networks to which we associate the last layer the concrete concepts related to our problem. Just as an image representing a cat will contain at its last layer the eyes, mouth, paws of the cat then in our case.
But this hypothesis can also be an overrepresentation because the accumulation of the layers can also be related to an enrichment of the biomechanical process or else of myosin type genes.

The rest of the charts detailed in this section are available at the end of this document. It is important to relativize the number of occurrences in relation to the number of total examples of the validation set. The total number of examples is 5578 in the validation dataset with 3742 cases of cancer so it is about 67.08%.

6.2.2 Hypergeometric test on the best probes

Our network contains 23 layers in total (see A) and we are going to realize here several cuts of our network with each index of layers belonging to a dropout and to carry out an analysis using LRP. We will select the neuron that had previously the best average (these values change each time we start a training model).

From here, we have a list of the 5000 best probes per LRP analysis.
We use again the *Bioconductor* package in order to be able to link PROBEID and ENTREZID and GOID (being biological functions) useful to the package *Gostats*. This will allow us to perform the 5 hypergeometric tests to see what are the biological functions associated back to the biological function of the best average neurons that we took.

GOBPID	p-value	OddsRatio	ExpCount	Count
GO:0045190	1.378852e-01	1.58750971	6.8597015	10
GO:0050685	7.158130e-01	0.8146025	3.5940299	3
GO:0042542	0.8608389173	0.75855933	16.574951	13
GO:0072576	9.092333e-01	0.4120025	2.2271252	1
No matching found	#	#	#	#

I explicitly chose not to perform the tests with a p-value of 0.05 because some functions were not present. Indeed, as can be seen from the table above, the p-value is very large and increases as one goes deeper into the network. The odds ratio (OR) is a statistical measure often used in epidemiology, expressing the degree of dependence between qualitative random variables. We note that the odds ratio decreases as we move deeper into the network, which means that the biological function is less and less likely to appear among the 5000 probes we have selected before.

GOBPID	Size	Term	Position	Dropout_Layer
GO:0045190	46	isotype switching	1437	1
GO:0050685	28	positive regulation of mRNA processing	6727	2
GO:0042542	129	response to hydrogen peroxide	8089	3
GO:0072576	16	liver morphogenesis	9363	4
No matching found	#	#	#	#

Regarding the biological information associated with the 4 function to which we found a match, we have in the first-place *isotype switching*. Isotypic switching is a process that, when a B lymphocyte matures, changes the isotype of the immunoglobulins produced. This is a change in the heavy chain of antibodies. This allows the humoral response to be more efficient and adapt to the type of pathogen.

Secondly, the term, *positive regulation of mRNA processing* and some research has been done [57]. Alternative splicing of mRNA precursors is a nearly ubiquitous and extremely flexible point of gene control in humans. It provides cells with the possibility to create protein isoforms of differing, even opposing, functions from a single gene. Cancer cells often take advantage of this flexibility to produce proteins that promote growth and survival.

About the next layer being associated with *response to hydrogen peroxide*, this study among others has shown that [58] the production of hydrogen peroxide is connected with cancer. Fibroblasts, can provide the necessary "fertilizer", accelerated aging, DNA damage, inflammation and cancer metabolism, in the microenvironment tumor. By secreting hydrogen peroxide, cancer cells and

fibroblasts are mimicking the behavior of immune cells (macrophages/neutrophils), driving local and systemic inflammation, via the innate immune response (NFκB).

Finally, we have the last biological function (**GO:0072576**), *liver morphogenesis*, this function having as link a part of the body and its morphogenesis corresponds to cell growth whereas cancer represents cell division. Then regarding the last layer, no correspondence was found, that can be explained by the fact that the last layer containing only 383 neurons, there is indeed little of *ENTERZID* which will be then associated with the biological function.

Chapter 7

Conclusion

I learned a lot during this internship, passing the knowledge related to deep learning and in biology. Although it was an internship in a public research laboratory, my work was not only to focus on the theory and reflections on publications. I worked also on implementation in several segments with different types of framework related to deep learning and a material that cannot be envied even in some self-proclaiming boxes of artificial intelligence with unsuitable material and low investment.

This memory contains, a beginning of progression with the interpretability of neural networks in the context of transcriptomic data. We have seen that biological information about the predictions of cancer cases is not far from the current research that is done. However, the probabilities that the batch of biological functions in association with the probes belong to the neurons with the highest relevance score indicate to us that the model is very far from reliable and that especially when it is more and more deep.

There is still some way to go before a physician decision assistant is in large-scale production and follows the requirements and current legislation. But we still have ideas to explore, such as techniques from the ensemble methods to improve the accuracy of the neural network but also the modification of the neural network architecture because we currently tested only with 5 layers.

This internship was a first step in the world of research where I was able to follow a rigorous methodology step by step process that took me several months to reach the result that we could detail previously.
It has also been a professional experience where I collaborate with many researchers, for example at monthly meetings when we discuss and exchange about the latest knowledge on the field of deep learning.
Finally, with this first experience in research, I would like to continue directly via a CIFRE thesis towards the sector of the applied research, in order to perfect my knowledge in this field which is an important development of the future.

7.1 Summary of the results on the work

The results are a start in the field of interpreting omics data with the use of neural network. It has been found that the neural network does not completely realize its predictions on artefacts or biological functions totally disconnect from cancer. On the other hand, the results vary a lot with each learning of the network and the results are not robust and then the hypergeometric tests show us that the values of the p-value are much too high to be considered actually in production.

One solution would be to use the techniques related to the ensemble methods that we will apply to the neural network, for example if we train 100 different networks and apply the same reasoning in order to extract the common biological functions and we keep the one with the most functions in common, so we can already say that we have gone one step further in the consistency of our results. Then as a last remark, I only use *LRP* as a relevance score technique with a ϵ fixed at 10^{-7}, even if in our case it was not demonstrated any differences real we could also perform tests with other techniques such as DeepLIFT.

7.2 Observations

Research around interpretation is growing, as the legislation has evolved, it is interesting to note the idea of *algorithmic responsibility*. For example, DARPA, the US military research agency, allocated $6.5 million in May 2017 to researchers to provide a "visual or written explanation" of its decisions.

Yet despite this research and wishful thinking, deep learning will probably be able to be transparent and traceable to the good old computer code. But is it really a problem? "Psychologists have shown that when you ask a human to explain a choice, it will build a justification after the fact that will probably not be the real reason," said Peter Norvig, the director of research of Google, June 2017. According to him, the same approach should be applied to Deep Learning: "One could lead an algorithm to generate an automatic explanation of their results based on the data that were provided." Our own reactions are often irrational, fruit of instinct, our subconsciousness or our habits.

These are all the questions we need to face today if we want artificial intelligence to be at the heart of our lives and to help us increasing our productivity and improve our way of life.

Chapter 8

Personal review

This internship was my first on a public research laboratory.
Indeed, I had already completed an internship in Master I, as data scientist intern at Eiffage. I consider that this internship brought me a lot, both technically and theoretically.
First, about technical skills. I worked for six months on python programming. In addition, I used tools specific to deep learning, so I was able to acquire and consolidate technical knowledge which I did not have the opportunity elsewhere.

Outside of practice, I had the opportunity to make a state of the art on a very large and growing topic. This allowed me to learn a lot about the field of deep learning. Also, Mr Hanczar invited me to a meeting where the different participants presented their work or their last reading. It helped me to have a broader vision and to get a little out of my internship topic.

However, the most important thing for me is that this internship allowed me to know what I wanted to do later. I knew I wanted to work in the field of computing. The vastness of the latter makes it difficult to choose. Thus, this master and this final internship allowed me to discover learning and made me want to learn more and work in it.

Moreover, it was my first experiences in the field of research. Having already experienced business experiences through internships and alternating, I highly appreciated the world of research. Therefore I now aim to obtain a CIFRE thesis. This internship will have helped me to set goals, a career plan, but also a credibility to achieve what I want.

Appendix A

Model Summary

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	(None, 54675)	0
first_layer (MyLayer)	(None, 2539)	138822364
dropout_20 (Dropout)	(None, 2539)	0
batch_normalization_20 (Batc	(None, 2539)	10156
re_lu_20 (ReLU)	(None, 2539)	0
second_layer (MyLayer)	(None, 4564)	11592560
dropout_21 (Dropout)	(None, 4564)	0
batch_normalization_21 (Batc	(None, 4564)	18256
re_lu_21 (ReLU)	(None, 4564)	0
third_layer (MyLayer)	(None, 4519)	20629235
dropout_22 (Dropout)	(None, 4519)	0
batch_normalization_22 (Batc	(None, 4519)	18076
re_lu_22 (ReLU)	(None, 4519)	0
fourth_layer (MyLayer)	(None, 2141)	9677320
dropout_23 (Dropout)	(None, 2141)	0
batch_normalization_23 (Batc	(None, 2141)	8564
re_lu_23 (ReLU)	(None, 2141)	0
fifth_layer (MyLayer)	(None, 383)	820386
dropout_24 (Dropout)	(None, 383)	0
batch_normalization_24 (Batc	(None, 383)	1532
re_lu_24 (ReLU)	(None, 383)	0
last_layer (MyLayer)	(None, 2)	768
softmax (Softmax)	(None, 2)	0

Total params: 181,599,217
Trainable params: 181,570,925
Non-trainable params: 28,292

Figure A.1: Details for each layer of the neural network.

Appendix B

LRP Distribution Score

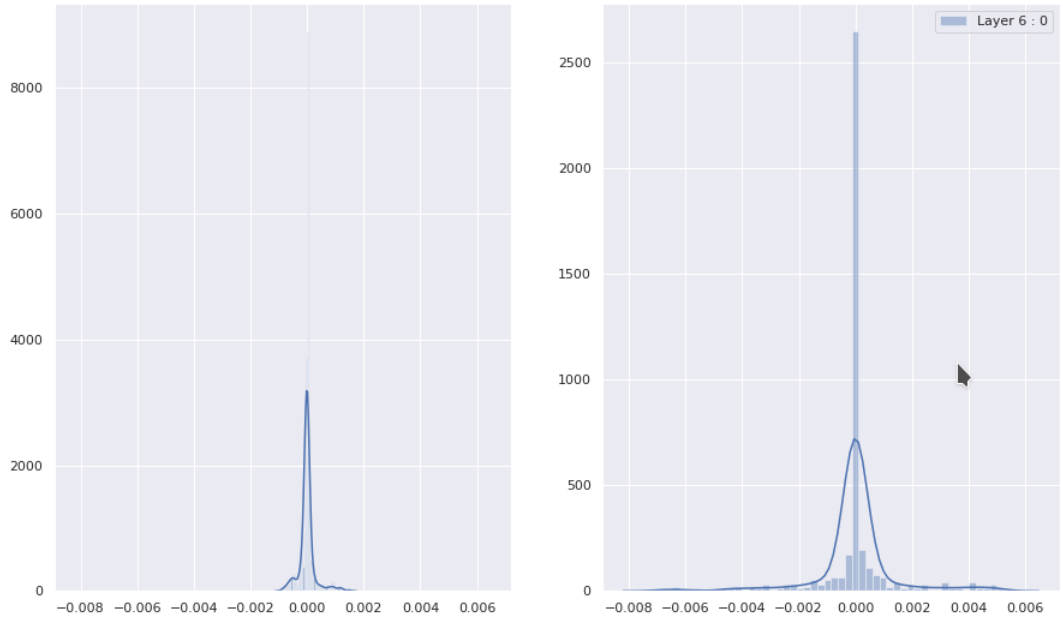

Figure B.1: Distribution of relevance score for layer 6.

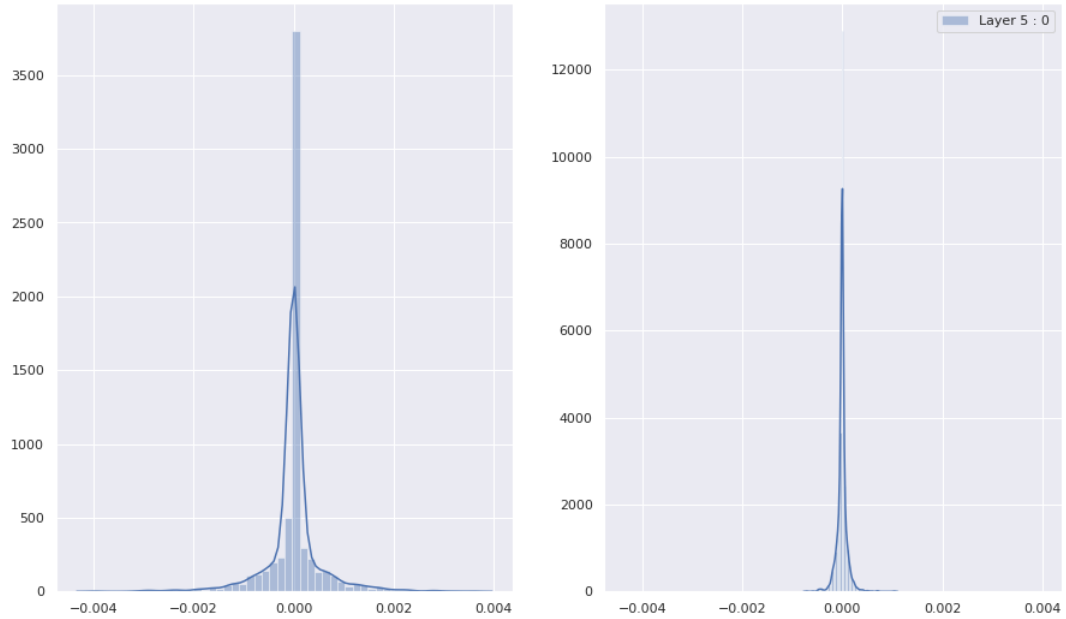

Figure B.2: Distribution of relevance score for layer 5.

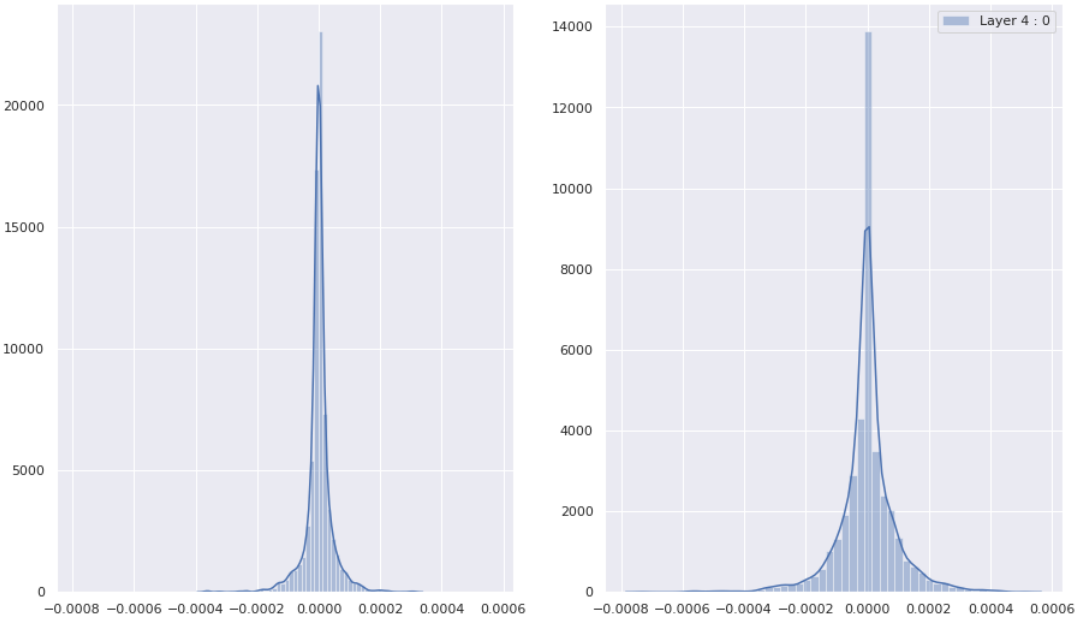

Figure B.3: Distribution of relevance score for layer 4.

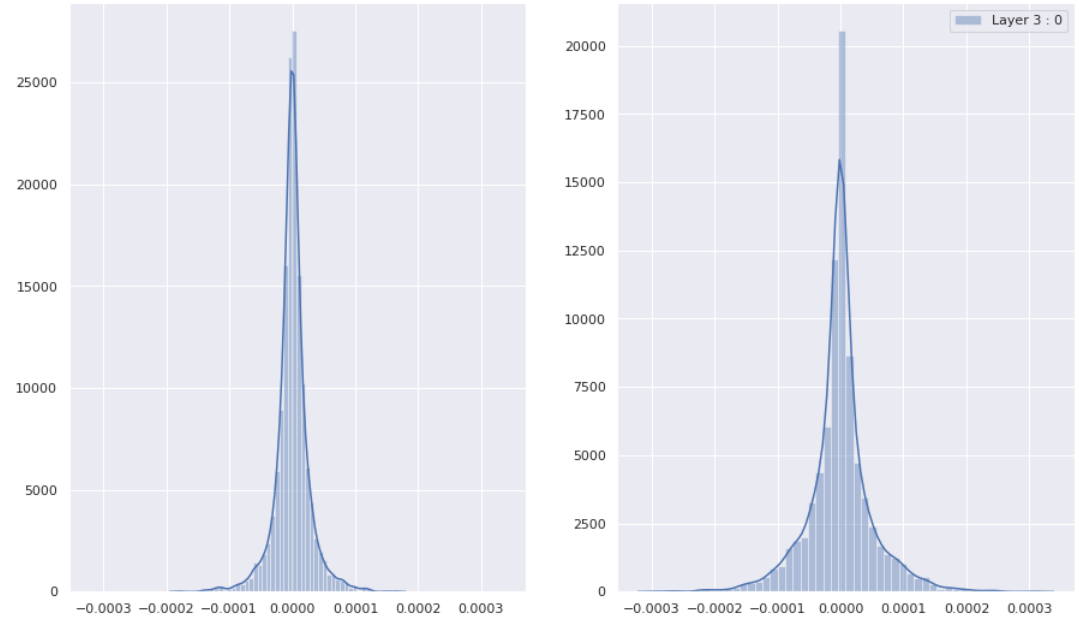

Figure B.4: Distribution of relevance score for layer 3.

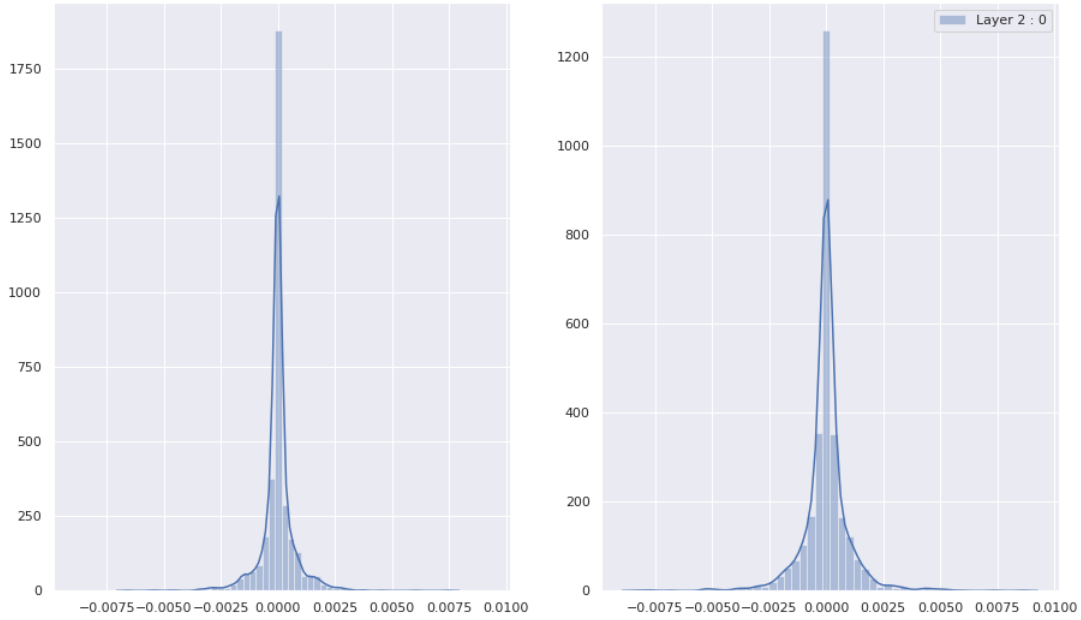

Figure B.5: Distribution of relevance score for layer 2.

Appendix C

Distribution of biological

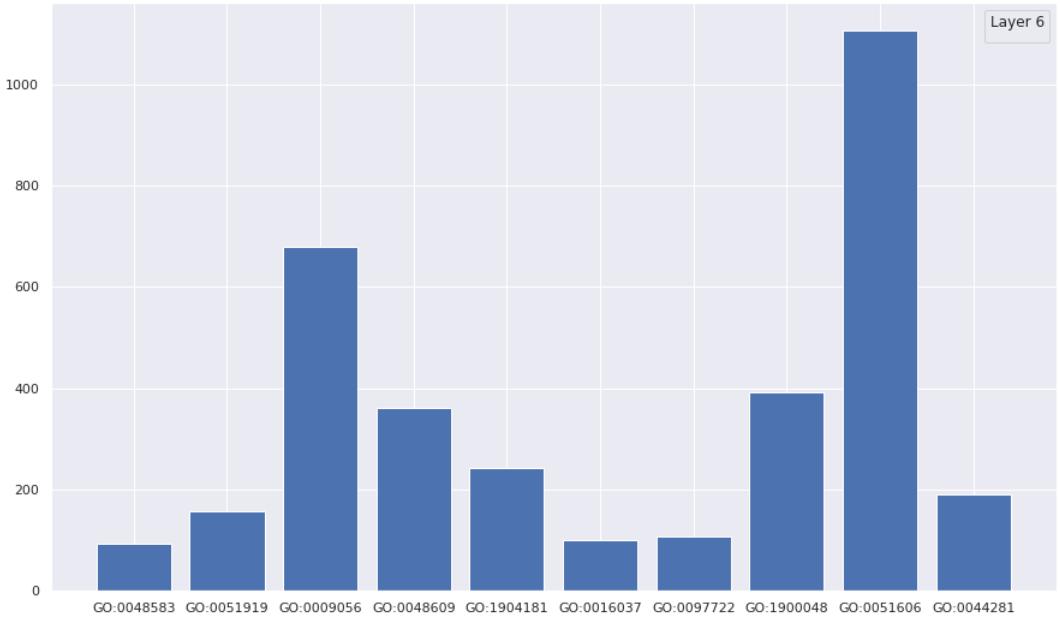

Figure C.1: Top 10 GO functions on Layer 6.

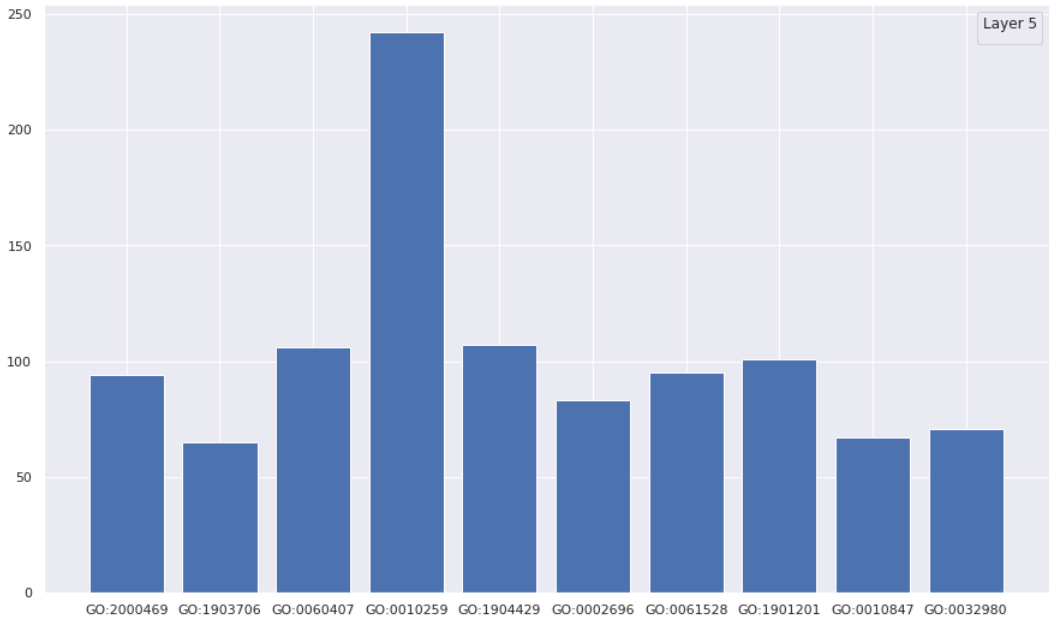

Figure C.2: Top 10 GO functions on Layer 5.

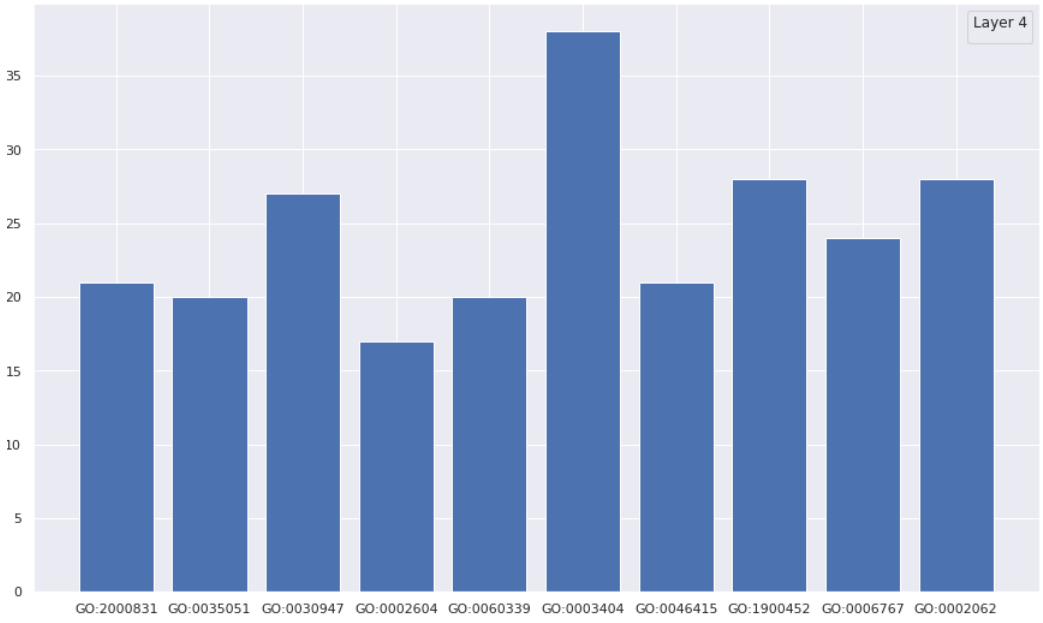

Figure C.3: Top 10 GO functions on Layer 4.

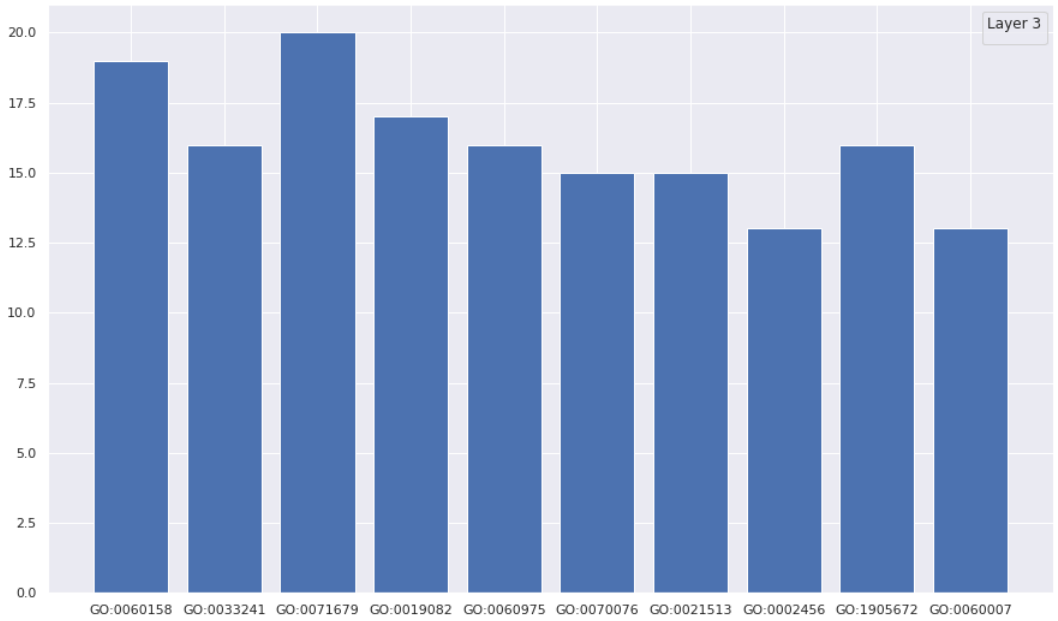

Figure C.4: Top 10 GO functions on Layer 3.

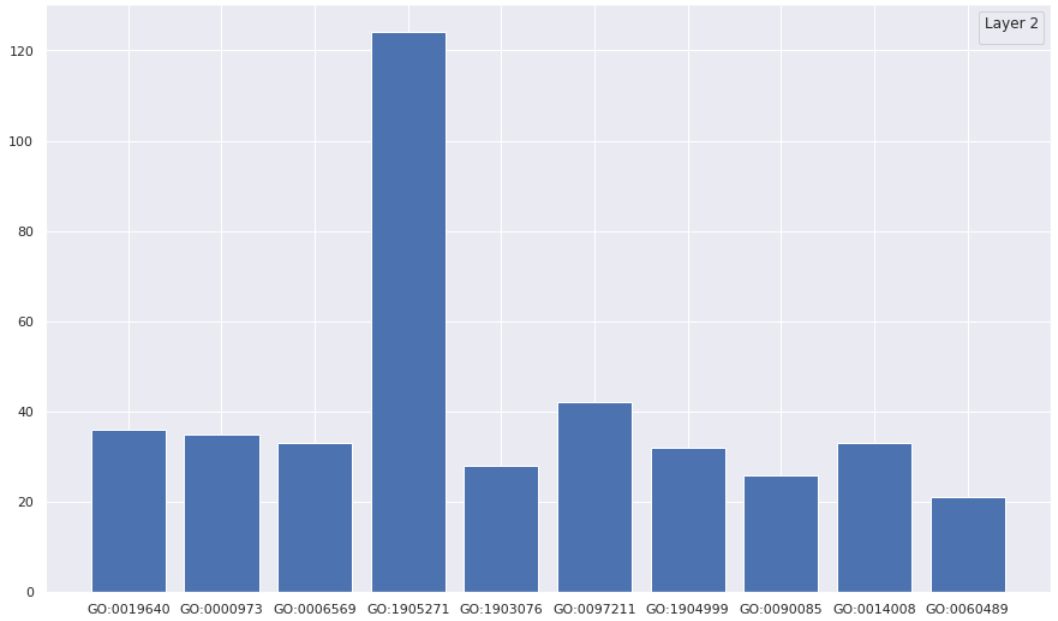

Figure C.5: Top 10 GO functions on Layer 2.

Appendix D

Accuracy plots

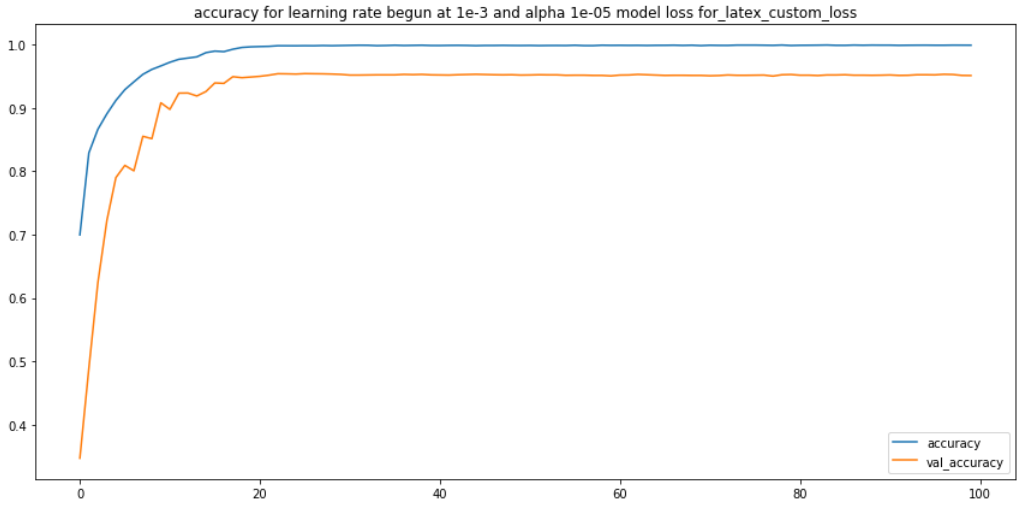

Figure D.1: Accuracy for $\alpha = 10^{-5}$ with Custom Loss.

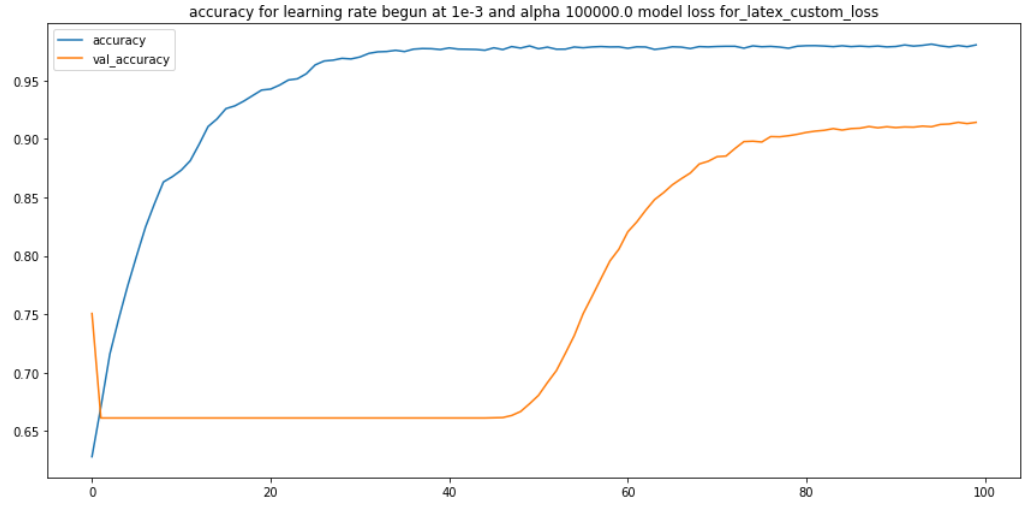

Figure D.2: Accuracy for $\alpha = 10^{6}$ with Custom Loss.

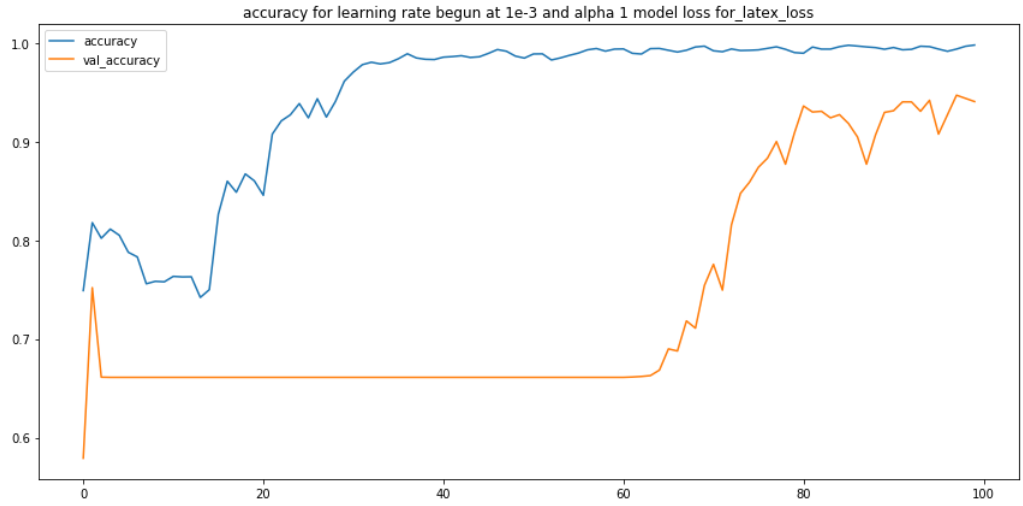

Figure D.3: Accuracy for $\alpha = 1$ with l_2-norm.

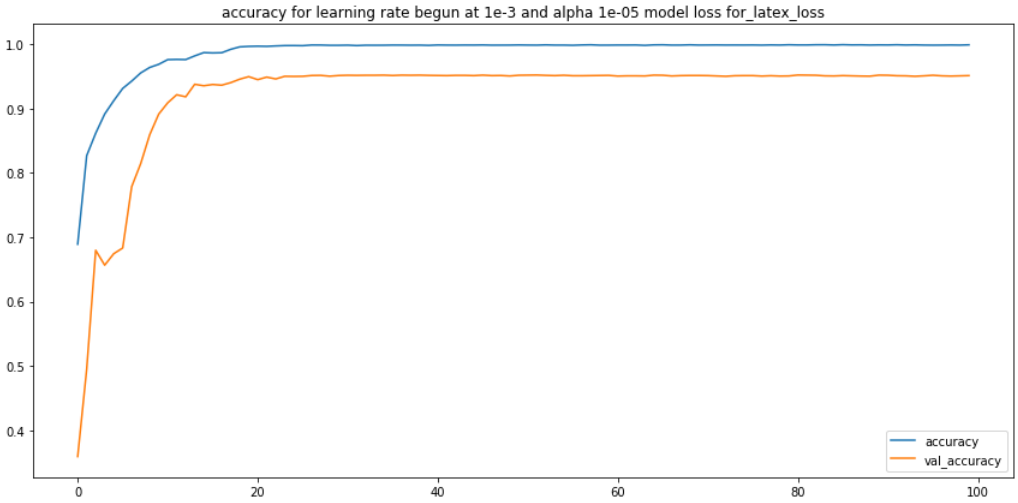

Figure D.4: Accuracy for $\alpha = 10^{-5}$ with l_2-norm.

Appendix E

Cross-entropy plots

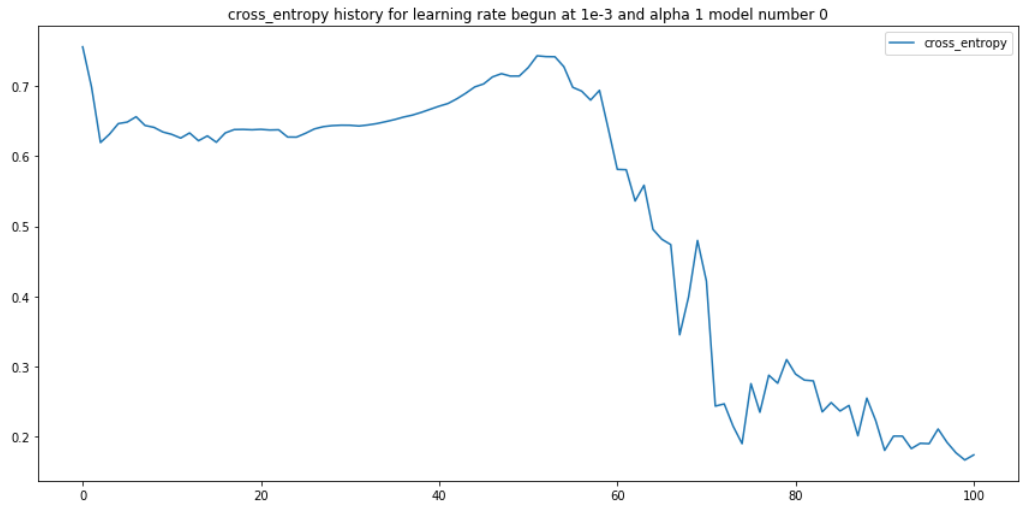

Figure E.1: Cross entropy for $\alpha = 1$ with Custom Loss.

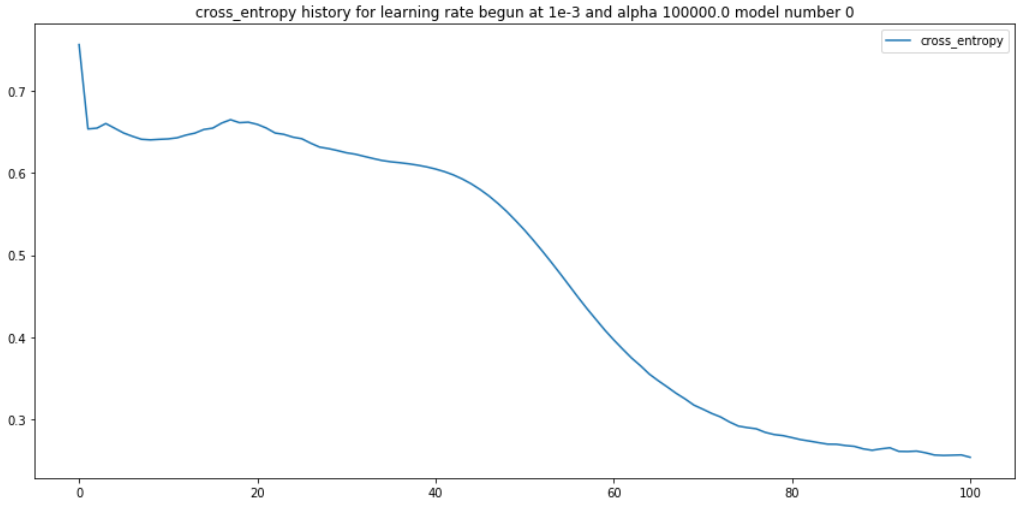

Figure E.2: Cross entropy for $\alpha = 10^6$ with Custom Loss.

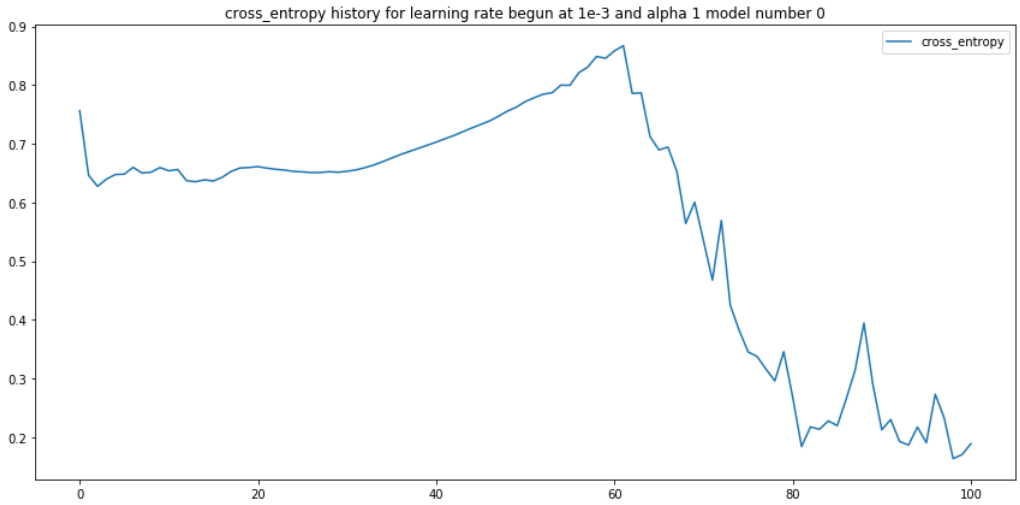

Figure E.3: Cross entropy for $\alpha = 1$ with l_2-norm.

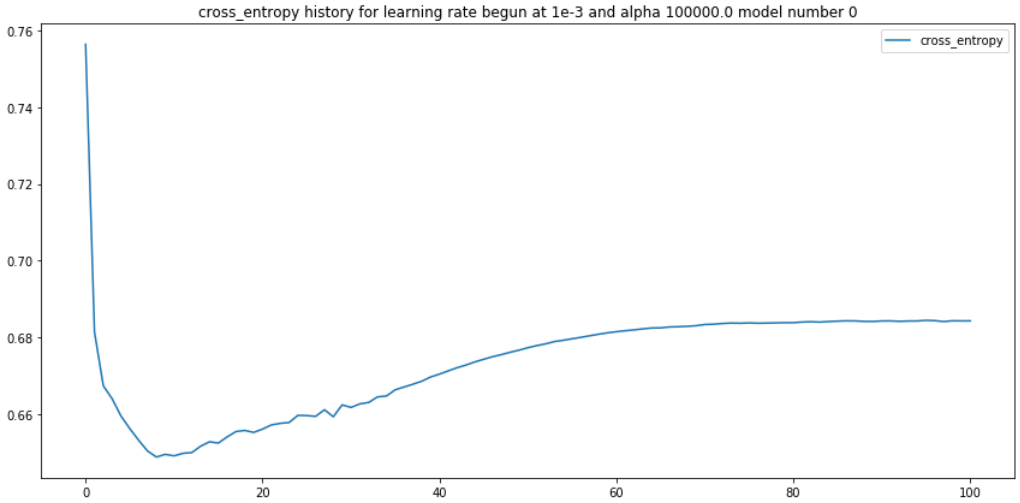

Figure E.4: Cross entropy for $\alpha = 10^6$ with l_2-norm.

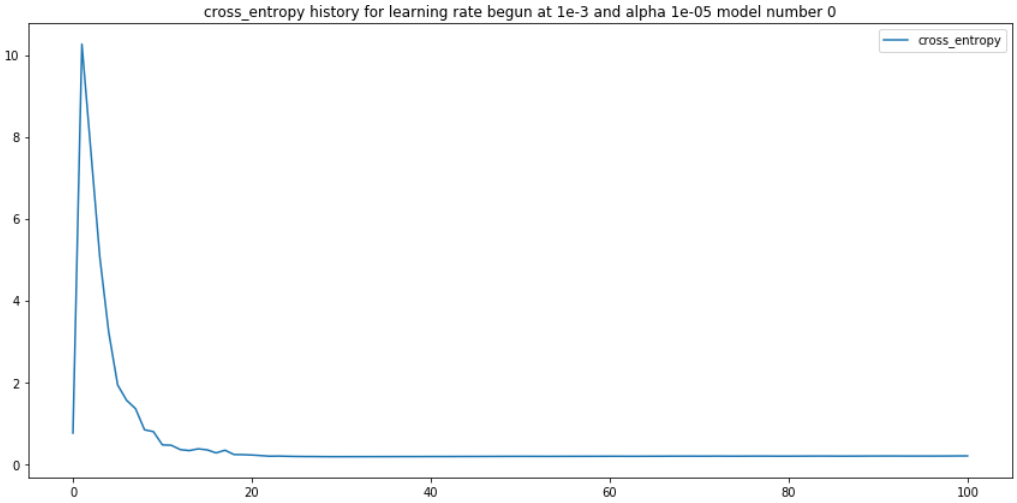

Figure E.5: Cross entropy for $\alpha = 10^{-5}$ with Custom Loss.

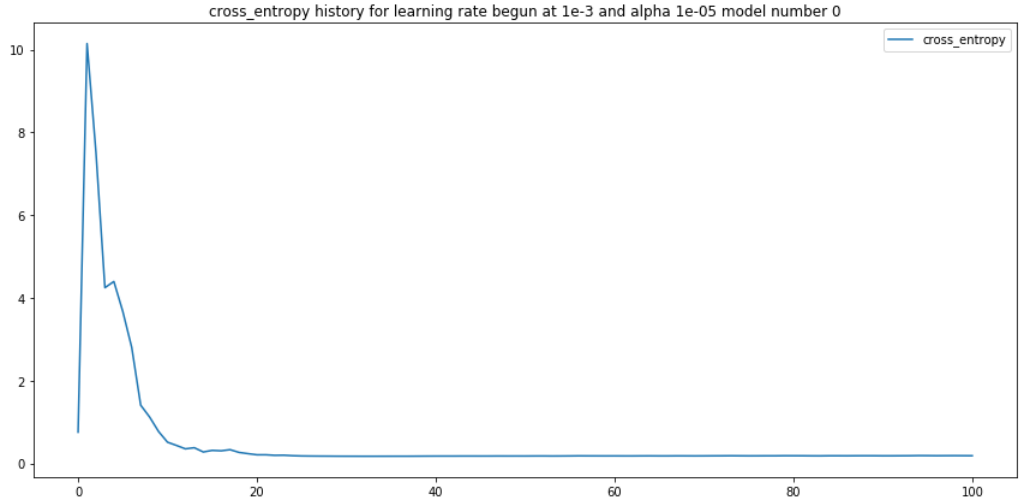

Figure E.6: Cross entropy for $\alpha = 10^{-5}$ with l_2-norm.

Appendix F

l_2 Penality GO

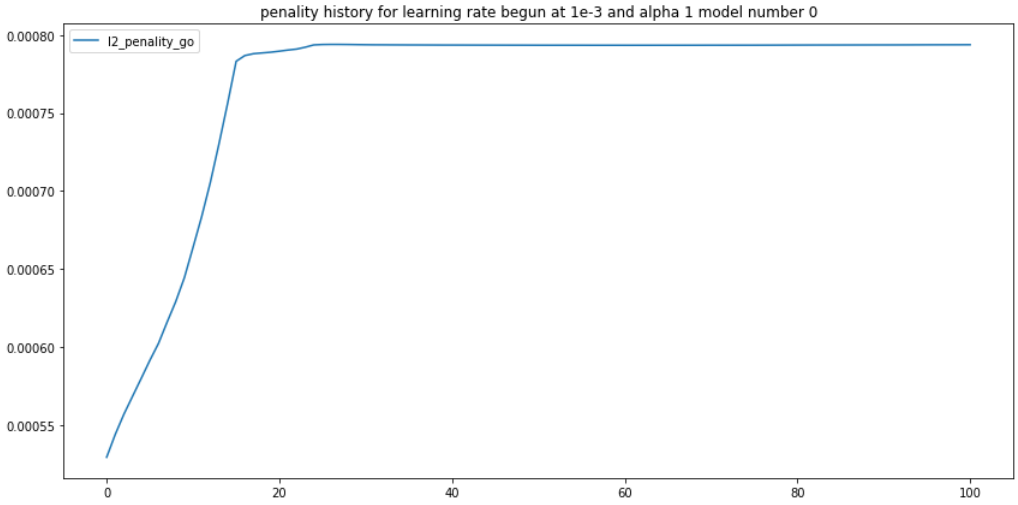

Figure F.1: l_2 Penality GO when $\alpha = 1$ with Custom Loss.

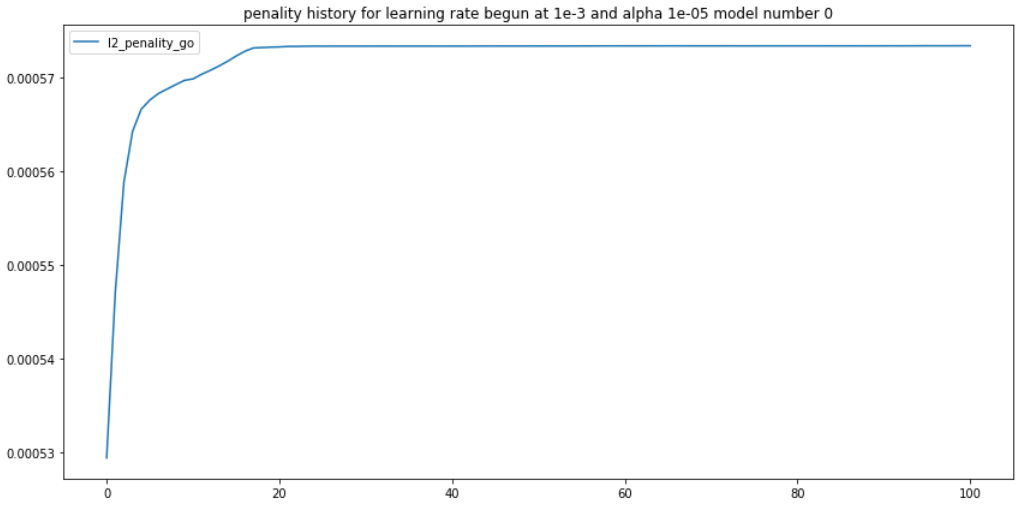

Figure F.2: l_2 Penality GO when $\alpha = 10^{-5}$ with Custom Loss.

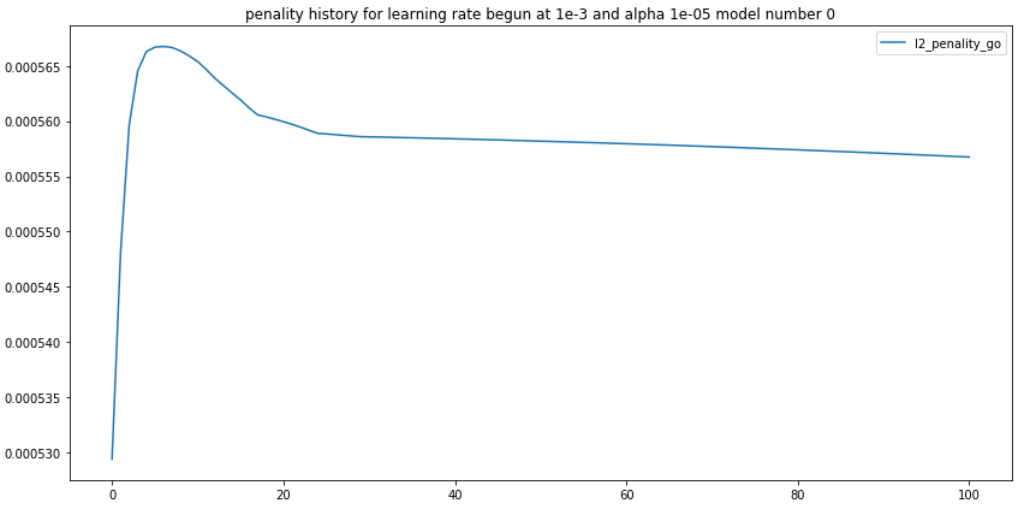

Figure F.3: l_2 Penality GO when $\alpha = 10^{-5}$ with l_2-norm.

Appendix G

l_2 Penality NO GO

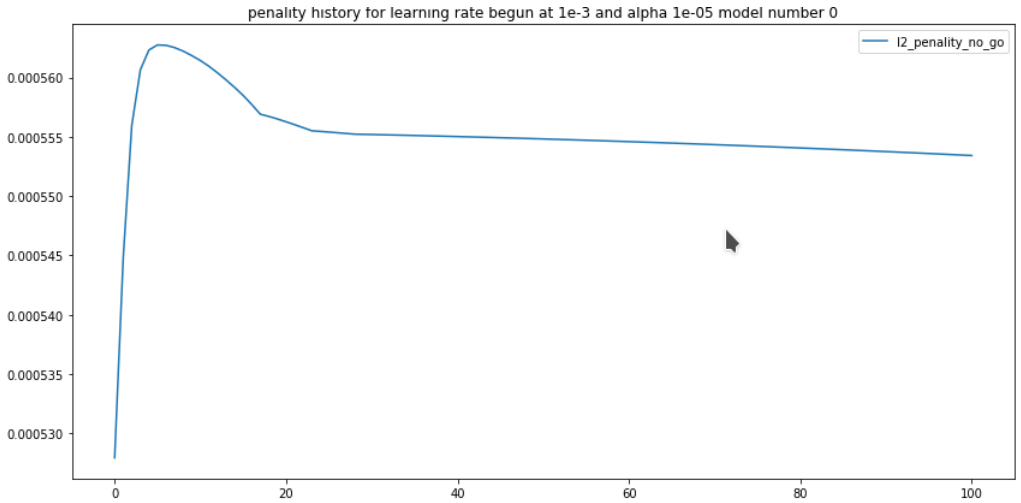

Figure G.1: l_2 Penality NO GO when $\alpha = 10^{-5}$ with Custom Loss.

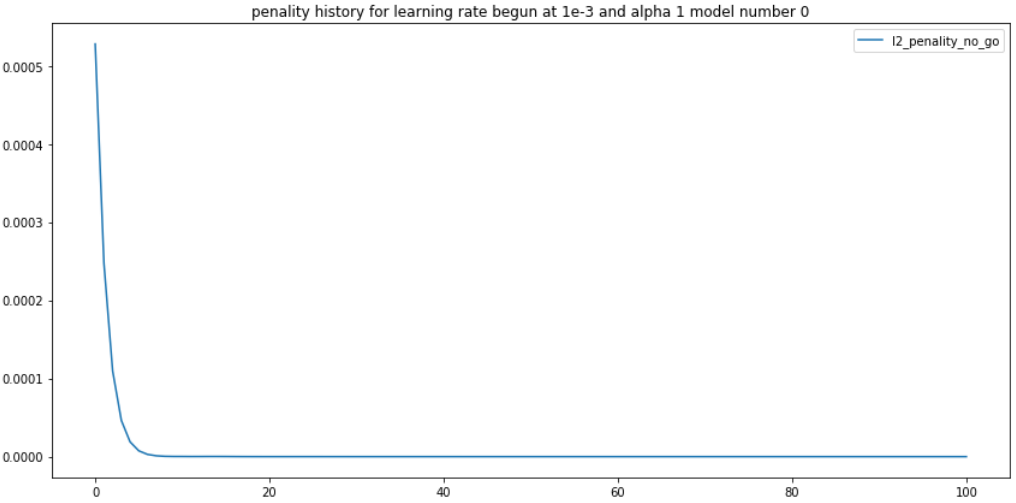

Figure G.2: l_2 Penality NO GO when $\alpha = 1$ with Custom Loss.

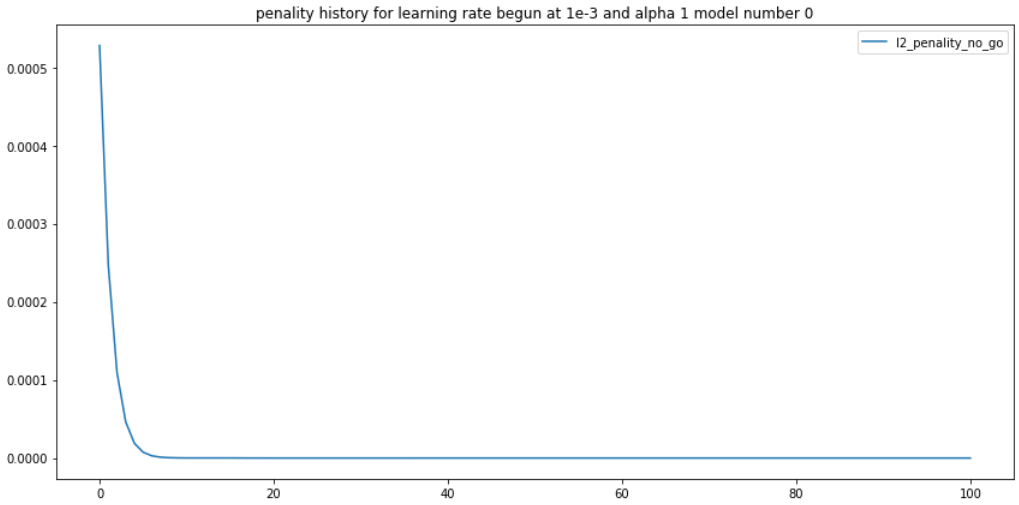

Figure G.3: l_2 Penality NO GO when $\alpha = 1$ with l_2-norm.

Figure G.4: l_2 Penality NO GO when $\alpha = 10^6$ with l_2-norm.

Appendix H

Loss

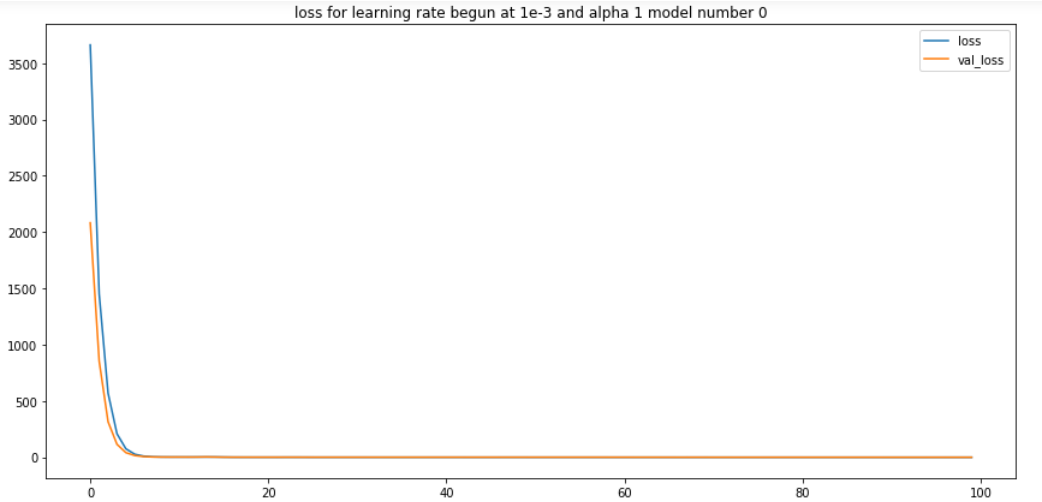

Figure H.1: l_2 Penality NO GO when $\alpha = 1$ with Custom Loss.

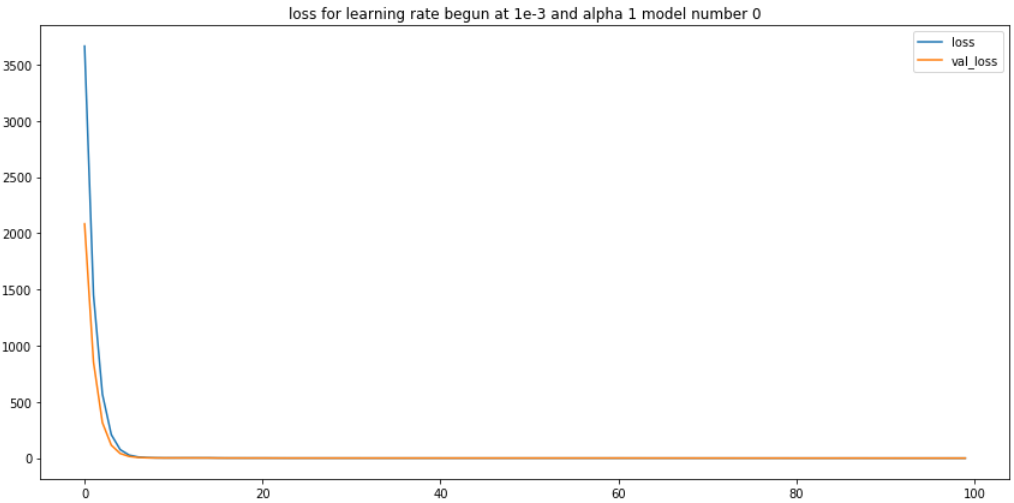

Figure H.2: l_2 Penality NO GO when $\alpha = 1$ with l_2-norm.

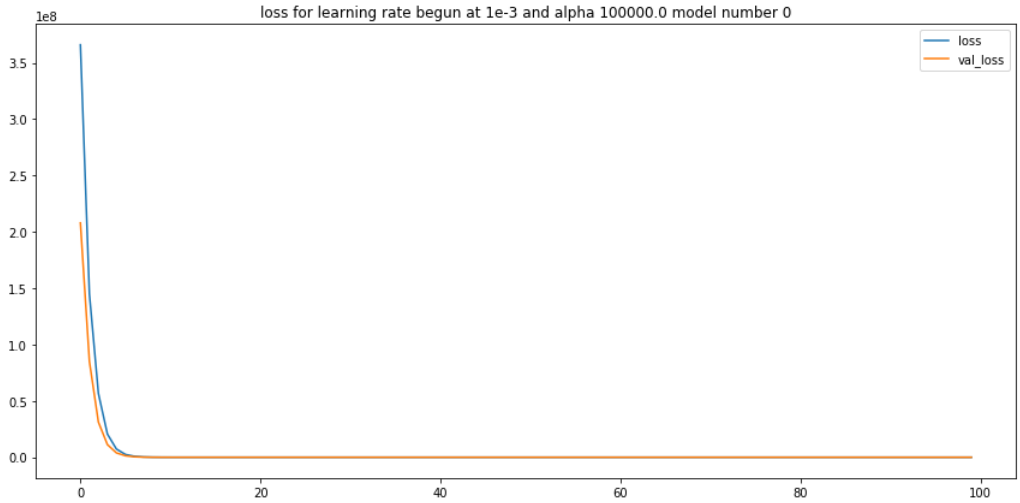

Figure H.3: l_2 Penality NO GO when $\alpha = 10^6$ with l_2-norm.

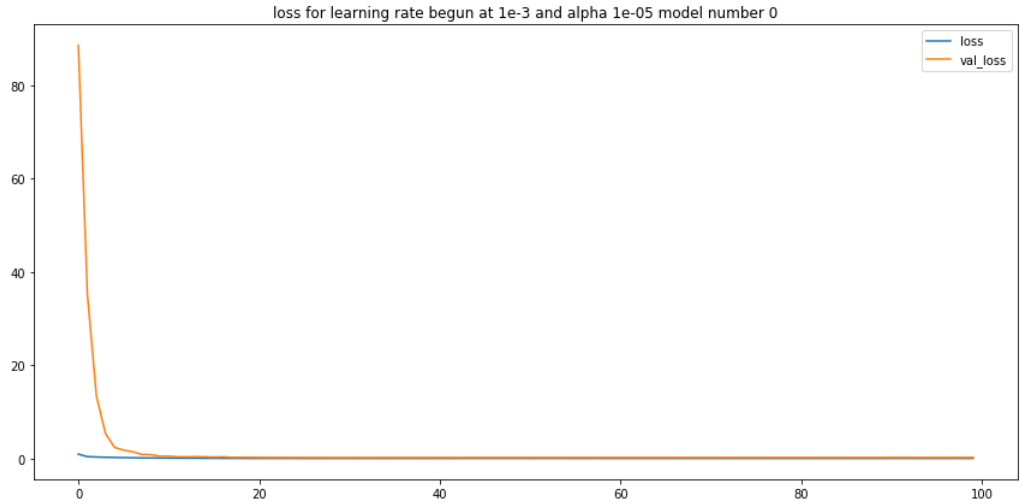

Figure H.4: l_2 Penality NO GO when $\alpha = 10^{-5}$ with Custom Loss.

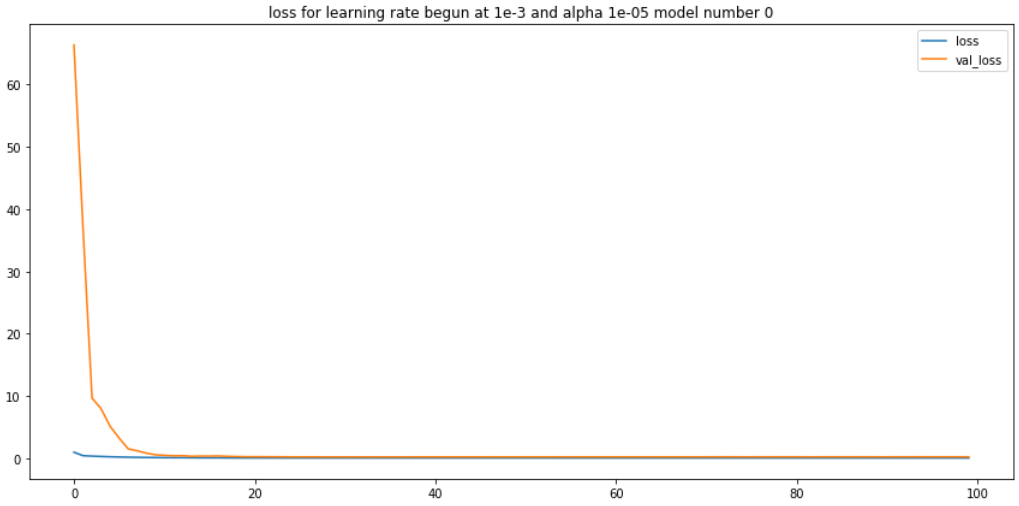

Figure H.5: l_2 Penality NO GO when $\alpha = 10^{-5}$ with l_2-norm.

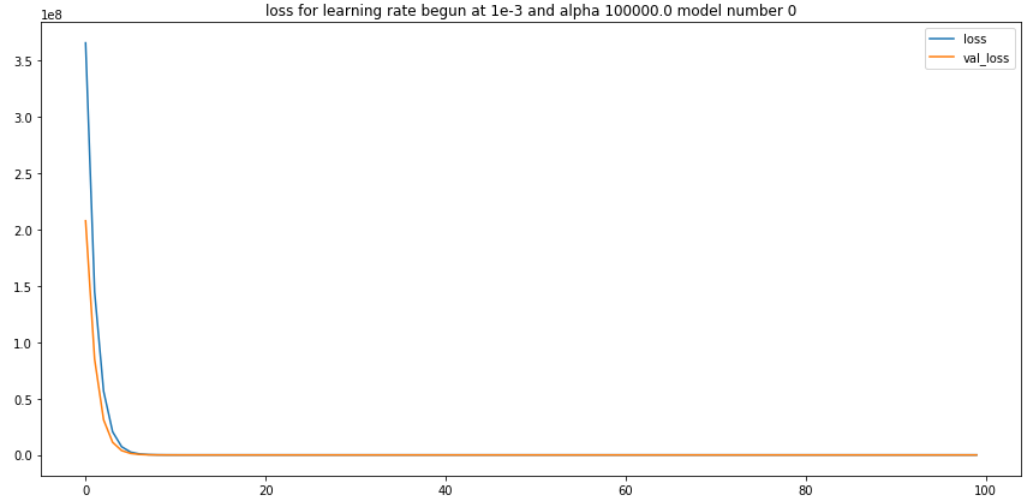

Figure H.6: l_2 Penality NO GO when $\alpha = 10^{6}$ with Custom Loss.

Appendix I

Synthesis

I.1 Context

The subject of this internship is placed in the context of precision medicine. In our case, we have transcriptomic data (gene expressions). Deep learning having greatly successful in the problems of image classification and recognition of natural language [59], it is perfectly legitimate to ask whether this could not be the case in precision medicine. This is the question that arise [60] about biology. Indeed, there is a very large amount of data which are complex and poorly understood. Deep learning could therefore give them sense.

It is important to have transparent systems since when any incident occurs, no one will be able to provide a reasoned explanation to justify the underlying decision. How can one defend oneself against a computer having declared you "terrorist" based on of its own judgment? What about misdiagnosis for a serious illness? Who will be held responsible in case of an accident of an autonomous car? The notion of "foreseeable harm", to which many legal texts refer, will be impossible to define since the behaviour of an AI is inherently unpredictable. For example, the European Commission accuses Google of favouring the results of its own shopping comparison in its search results. A charge that will tomorrow be impossible to demonstrate if it is a self-learning algorithm and not a human that feeds the search engine.

I.2 Motivations

As noted by [61], there is a problem with the very large size of the gene expression data and the small number of examples. Thus, it can be difficult to be precise in the predictions, it must be kept in mind that this is not the subject of this memoir. The second problem noted concerns the "interpretability" of the neural network and its predictions. This is what [62] are interested in. Indeed, it is clearly emphasized that the interpretation aspect would make it possible to understand this multitude of data. In addition, this would be an opportunity to give confidence in the predictions. It is this second problem which constitutes the subject of this memoir.

I.3 State of the art

As the subject of this dissertation relates to the fields of precision medicine and deep learning, two things must be distinguished. On the one hand, for the biology aspect, we have some articles that begin to deal with the use of learning deep ([62]). We note that the interest of deep learning has grown. However, we find the two problems seen previously. In terms of interpretation, most papers looked only at the value of weights and activations to try to interpret their models. However, no

method specific to a biological interpretation has been put forward.
On the other hand, we have the interpretation of neural networks that has grown strongly and gained popularity, especially in image processing. This can be seen in the [4] survey.
Among all the methods presented, we can discover what are called the methods of attribution of relevance scores. We can assign a relevance score for each neuron. As a result, we will be able to know which variable or which neuron has a determining role in the decision-making. We will know what our model is based on. The main methods are Simple Gradient Method ([26]), LRP ([2]), DeepLIFT ([26]), Integrated Gradients ([27]) and Guided Backpropagation ([28]).

I.4 Personal contributions

To start I had to extract the biological informations associated with the probes on input in the future neural network. A statistical analysis concerning the loss of informations such as probes, genes and GO functions was performed. It turns out that there is 33% loss of probes and after building the direct acyclic graph from the GO function of our dataset, I have taken all 5 levels having the most GO functions continuously. An adjacency matrix was constructed from the direct acyclic graph previously contrasts to add this information to a custom cost function during learning.

An implementation of the InOmicNet architecture was carried out on Tensorflow 2.0 with layers of the custom neuron network as well as a very pushing monitoring system considering the precision, the loss according to different norms, and the penalties. An in-depth research on the best coefficiant regularizer was performed and a verification regarding of the different penalties GO and NO GO too. Then retrieve the neurons associated with the best average score of relevance from the LRP done but also an enrichment test on the return of 5000 probes and their associated biological information compared to neurons taken previously. Finally, research on these biological functions in return and their association on current research around cancer.

I.5 Potential use of work done

After this internship, the IBISC laboratory will have a documented source code that can be reused. So, we could create a more comprehensive tool that would automate what was done experimentally. This tool would not only give confidence to neural networks, but could also become a tool for data analysis and contribute to medical research. In addition, the results obtained should be used to write the thesis of Victoria Bourgeais and also future publications. In general, the interpretation presented throughout this paper is valid for the entire field of learning. Thus, all areas using learning would be impacted by this research topic. One can even imagine that this will become a necessity since for several domains, the legislation requires a certain transparency (social networks, public services, ...).

I.6 Prospects for work

There are many points to raise in the area of interpretation. First, terminology, interpretation methods and validation methods should be standardized. In addition, we should define what is a good interpretation. What are the criteria to respect? Since the interpretation is directly related to human understanding, its quality becomes subjective.

For our case, the goal was to discriminate as many neurons as possible. However, this may not be the case for all domains. Moreover, how to judge the interpretation of a neural network on a domain in which our knowledge is still limited. For the future, it becomes an analysis tool and

gives research leads, but it does not make it possible to judge the model.

The field of learning and the subject of interpretation being trending upwards, I think all of this should happen in 5 or 10 years. The interpretation could give confidence to the neural networks and participate in their adoption. However, it will be interesting to see if this gain in transparency will be favourable at all levels for deep learning.

As for my future on this subject. I intend to continue in the field of deep learning. I do not know if I will work directly about interpretation, this internship still allowed me to have advanced knowledge on a key point of learning, which could work in my favour in the future.

Bibliography

[1] Ribana Roscher, Bastian Bohn, Marco F. Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries, 05 2019.

[2] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *CoRR*, abs/1706.07979, 2017.

[3] Zachary Chase Lipton. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016.

[4] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani B. Srivastava, Alun D. Preece, Simon J. Julier, Raghuveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. Interpretability of deep learning models: A survey of results. *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6, 2017.

[5] Bryce Goodman and Seth Flaxman. Eu regulations on algorithmic decision-making and a "right to explanation", 2016. cite arxiv:1606.08813Comment: presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY.

[6] Frank C. Keil and Robert A. Wilson. *Explanation and Cognition*. MIT Press, 2000.

[7] Ed Southern, Kalim Mir, and Mikhail Shchepinov. Molecular interactions on microarrays. *Nature genetics*, 21:5–9, 02 1999.

[8] Mark Adams, J M Kelley, J D Gocayne, M Dubnick, Mihael Polymeropoulos, Huadong Xiao, Carl Merril, A Wu, Bjorn Olde, and Ruben Moreno. Complementary dna sequencing: Expressed sequence tags and human genome project. *Science (New York, N.Y.)*, 252:1651–6, 07 1991.

[9] Marc Sultan, Marcel Schulz, Hugues Richard, Alon Magen, Andreas Klingenhoff, Matthias Scherf, Martin Seifert, Tatjana Borodina, Aleksey Soldatov, Dmitri Parkhomchuk, Dominic Schmidt, Sean O'Keeffe, Stefan Haas, Martin Vingron, Hans Lehrach, and Marie-Laure Yaspo. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science (New York, N.Y.)*, 321:956–60, 08 2008.

[10] Tahsin Ferdous and Mohammad Ohid Ullah. An overview of rna-seq data analysis. *Journal of Biology and Life Science*, 8:57, 08 2017.

[11] Abby Manthey, Anne Terrell, Salil Lachke, Shawn Polson, and Melinda Duncan. Development of novel filtering criteria to analyze rna-sequencing data obtained from the murine ocular lens during embryogenesis. *Genomics data*, 2:369–374, 12 2014.

[12] Piotr Balwierz, Piero Carninci, Carsten Daub, Jun Kawai, Yoshihide Hayashizaki, Werner Van Belle, Christian Beisel, and Erik Nimwegen. Methods for analyzing deep sequencing expression data: Constructing the human and mouse promoterome with deepcage data. *Genome biology*, 10:R79, 08 2009.

[13] Alicia Oshlack and Matthew Wakefield. Transcript length bias in rna-seq data confounds systems biology. *Biology direct*, 4:14, 05 2009.

[14] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[17] François Chollet et al. Keras. `https://keras.io`, 2015.

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[19] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. 11 2014.

[20] David Burca, Manuel Schüller, and Johannes Zlabinger. Case-based reasoning and machine learning. 05 2018.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[22] Samantha Krening, Brent Harrison, Karen M. Feigh, Charles Lee Isbell, Mark O. Riedl, and Andrea Lockerd Thomaz. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9:44–55, 2017.

[23] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM.

[24] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *preprint*, 12 2013.

[25] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *CoRR*, abs/1704.02685, 2017.

[27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3319–3328. JMLR.org, 2017.

[28] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. 12 2014.

[29] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[30] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *CoRR*, abs/1702.04595, 2017.

[31] Dumitru Erhan, Y Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Univeristé de Montréal*, 01 2009.

[32] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

[33] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3387–3395. Curran Associates, Inc., 2016.

[34] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015.

[35] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. 03 2017.

[36] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, pages 24–30, Cambridge, MA, USA, 1995. MIT Press.

[37] Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *CoRR*, abs/1512.02479, 2015.

[38] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[39] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[40] Stephen Bazen and Xavier Joutard. The taylor decomposition: A unified generalization of the oaxaca method to nonlinear models. 05 2013.

[41] Amirata Ghorbani, Abubakar Abid, and James Zou. INTERPRETATION OF NEURAL NETWORK IS FRAGILE, 2018.

[42] Marco Ancona, Enea Ceolini, A. Cengiz Öztireli, and Markus H. Gross. A unified view of gradient-based attribution methods for deep neural networks. *CoRR*, abs/1711.06104, 2017.

[43] Gordon K Smyth. *Limma: linear models for microarray data*, pages 397–420. Springer, New York, 2005.

[44] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 06 2018.

[45] Aurora Torrente, Margus Lukk, Vincent Xue, Helen Parkinson, Johan Rung, and Alvis Brazma. Identification of cancer related genes using a comprehensive map of human gene expression. *PLOS ONE*, 11:e0157484, 06 2016.

[46] NetworkX developer team. Networkx, 2014.

[47] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. innvestigate neural networks!, 08 2018.

[48] Allen JM Adrian TE, Bloom SR, Ghatei MA, Rossor MN, Roberts GW, Crow TJ, and Polak JM. Tatemoto K. Neuropeptide y distribution in human brain. *Nature*, 306, 12 1983.

[49] Juan Li, Yuchen Tian, and Aiguo Wu. Neuropeptide y receptors: a promising target for cancer imaging and therapy. *Regenerative Biomaterials*, 2, 09 2015.

[50] Bradleigh Whitton, Haruko Okamoto, Graham Packham, and Simon Crabb. Vacuolar atpase as a potential therapeutic target and mediator of treatment resistance in cancer. *Cancer Medicine*, 7, 06 2018.

[51] M Duman-Scheel. Netrin and dcc: Axon guidance regulators at the intersection of nervous system development and cancer. *Current drug targets*, 10:602–10, 08 2009.

[52] Pascal Jézéquel and Mario Campone. Comment on "how the evolution of multicellularity set the stage for cancer". *British Journal of Cancer*, 119, 05 2018.

[53] Francesco Brancati, Bruno Dallapiccola, and Enza Maria Valente. Joubert syndrome and related disorders. *Orphanet journal of rare diseases*, 5:20, 07 2010.

[54] Hui Gao, Bao-Jun Yang, Nan Li, Li-Min Feng, Xiao-Yu Shi, Wei-Hong Zhao, and Si-Jin Liu. Bisphenol a and hormone-associated cancers: Current progress and perspectives. *Medicine*, 94:e211, 01 2015.

[55] Aleksandra Konieczna, Aleksandra Rutkowska, and Dominik Rachoń. Health risk of exposure to bisphenol a (bpa). *Roczniki Państwowego Zakładu Higieny*, 66:5–11, 03 2015.

[56] Rebecca Wong. Apoptosis in cancer: From pathogenesis to treatment. *Journal of experimental clinical cancer research : CR*, 30:87, 09 2011.

[57] Charles J David and James L Manley. Alternative pre-mrna splicing regulation in cancer: Pathways and programs unhinged. *Genes development*, 24:2343–64, 11 2010.

[58] Michael Lisanti, Ubaldo Martinez-Outschoorn, Zhao Lin, Stephanos Pavlides, Diana Whitaker-Menezes, Richard Pestell, Anthony Howell, and Federica Sotgia. Hydrogen peroxide fuels aging, inflammation, cancer metabolism and metastasis the seed and soil also needs "fertilizer". *Cell cycle (Georgetown, Tex.)*, 10:2440–9, 08 2011.

[59] Yann LeCun, Y Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[60] Travers Ching, Daniel Himmelstein, Brett K Beaulieu-Jones, Alexandr Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul Agapow, Wei Xie, Gail L Rosen, Benjamin J Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M Cofer, David J Harris, and Casey Greene. Opportunities and obstacles for deep learning in biology and medicine. *bioRxiv*, 05 2017.

[61] PADIDEH DANAEE, Reza Ghaeini, and David Hendrix. A deep learning approach for cancer detection and relevant gene identification. volume 22, pages 219–229, 02 2017.

[62] Robert Haas, Aleksej Zelezniak, Jacopo Iacovacci, Sephan Karmad, StJohn Townsend, and Markus Ralser. Designing and interpreting 'multi-omic' experiments that may change our understanding of biology. *Current Opinion in Systems Biology*, 6, 09 2017.

[63] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *in COMPSTAT*, 2010.

[64] Chensi Cao, Feng Liu, Hai Tan, Deshou Song, Wenjie Shu, Weizhong Li, Yiming Zhou, Xiaochen Bo, and Zhi Xie. Deep learning and its applications in biomedicine. *Genomics, Proteomics Bioinformatics*, 16, 03 2018.

[65] M.M. Ashburner, C.A.C. Ball, Judith Blake, David Botstein, Heather Butler, J.M.J. Cherry, Allan Peter Davis, Kara Dolinski, Selina Dwight, Janan Eppig, Midori Harris, D.P. Hill, Laurie Issel-Tarver, A Kasarskis, Suzanna Lewis, John Matese, J.E. Richardson, M Ringwald, G.M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25:25–29, 05 2000.

[66] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

[67] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1885–1894. JMLR.org, 2017.

[68] Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. *CoRR*, abs/1604.00825, 2016.

[69] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[70] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017.

[71] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[72] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. Published online 2014; based on TR arXiv:1404.7828 [cs.NE].

[73] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermüller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas

Bouchard, Nicolas Boulanger-Lewandowski, Xavier Bouthillier, Alexandre de Brébisson, Olivier Breuleux, Pierre Luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul F. Christiano, Tim Cooijmans, Marc-Alexandre Côté, Myriam Côté, Aaron C. Courville, Yann N. Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Melanie Ducoffe, Vincent Dumoulin, Samira Ebrahimi Kahou, Dumitru Erhan, Ziye Fan, Orhan Firat, Mathieu Germain, Xavier Glorot, Ian J. Goodfellow, Matthew Graham, Çaglar Gülçehre, Philippe Hamel, Iban Harlouchet, Jean-Philippe Heng, Balázs Hidasi, Sina Honari, Arjun Jain, Sébastien Jean, Kai Jia, Mikhail Korobov, Vivek Kulkarni, Alex Lamb, Pascal Lamblin, Eric Larsen, César Laurent, Sean Lee, Simon Lefrançois, Simon Lemieux, Nicholas Léonard, Zhouhan Lin, Jesse A. Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-Antoine Manzagol, Olivier Mastropietro, Robert McGibbon, Roland Memisevic, Bart van Merriënboer, Vincent Michalski, Mehdi Mirza, Alberto Orlandi, Christopher Joseph Pal, Razvan Pascanu, Mohammad Pezeshki, Colin Raffel, Daniel Renshaw, Matthew Rocklin, Adriana Romero, Markus Roth, Peter Sadowski, John Salvatier, François Savard, Jan Schlüter, John Schulman, Gabriel Schwartz, Iulian Vlad Serban, Dmitriy Serdyuk, Samira Shabanian, Étienne Simon, Sigurd Spieckermann, S. Ramana Subramanyam, Jakub Sygnowski, Jérémie Tanguay, Gijs van Tulder, Joseph P. Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm de Vries, David Warde-Farley, Dustin J. Webb, Matthew Willson, Kelvin Xu, Lijun Xue, Li Yao, Saizheng Zhang, and Ying Zhang. Theano: A python framework for fast computation of mathematical expressions. *CoRR*, abs/1605.02688, 2016.

[74] Marco Ancona, Enea Ceolini, A. Cengiz Öztireli, and Markus H. Gross. A unified view of gradient-based attribution methods for deep neural networks. *CoRR*, abs/1711.06104, 2017.

[75] Stephen Bazen and Xavier Joutard. The taylor decomposition: A unified generalization of the oaxaca method to nonlinear models. 05 2013.

[76] Konstantinos Nikolaidis, Stein Kristiansen, Vera Goebel, and Thomas Plagemann. Learning from higher-layer feature visualizations. *CoRR*, abs/1903.02313, 2019.

[77] A. A. Minai and R. D. Williams. Perturbation response in feedforward networks. *Neural Networks*, 7(5):783–796, 1994.

[78] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[79] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[80] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[81] Liangchen Luo, Yuanhao Xiong, and Yan Liu. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019.

[82] William Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. 01 2019.

[83] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An approach to evaluating interpretability of machine learning. *CoRR*, abs/1806.00069, 2018.

[84] Ayad Ghany Ismaeel and Anar Auda Ablahad. Novel method for mutational disease prediction using bioinformatics techniques and backpropagation algorithm. *CoRR*, abs/1303.0539, 2013.

[85] M Ashburner, C A Ball, J A Blake, D Botstein, H Butler, J M Cherry, A P Davis, K Dolinski, S S Dwight, J T Eppig, M A Harris, D P Hill, L Issel-Tarver, A Kasarskis, S Lewis, J C Matese, J E Richardson, M Ringwald, G M Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29, May 2000.

[86] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196. IEEE Computer Society, 2015.