Explaining autoencoders with local impact scores

Clément Picard, Hoel Le Capitaine

Explaining autoencoders with local impact scores

Clément Picard, Hoel Le Capitaine

Nantes Université, LS2N, France

Abstract. The growth of deep learning methods underlines our need for explicability, to clarify and control the choices made by artificial intelligence programs. This paper summarizes our proposal to explain the representations created by autoencoders. The pattern recognized by each neuron of the network is defined in a few words by combining the weighted definitions of the neurons from the previous layer. This process explains the network as a network of patterns and allows to understand both the organization and the output of the autoencoder. Thanks to its design made for neural networks, the method can also be generalized to many problems such as binary classification, multi-class classification or regression.

Keywords: XAI, Explainability, Neural network, Autoencoder, Representation learning.

1 Introduction

As deep learning grows, its methods are being applied to an increasing variety of tasks. However, the validation of these approaches often comes up against the "black-box" behavior of their models, unable to justify their decisions [22]. The explainability of deep learning models and/or of their results is therefore a key element in the development of this field [21]. As outlined by Adadi et al, explainable AI (XAI) "*tends to refer to the movement, initiatives, and efforts made in response to AI transparency and trust concerns, more than to a formal technical concept*" [1]. Thus, XAI aims to produce more explainable models, while maintaining a high level of learning performance [12].

We suppose that the reader is familiar with neural networks approaches, and thus, we briefly introduce the necessary concepts. The interested reader can refer to [13] for more details. A neural network consists of a set of layers of neurons connected one after the other. Each neuron receives as input the activation of the neurons from the previous layer, and returns as output an activation value that corresponds to a non-linear function of the sum of its inputs multiplied by weights. Training a neural network from a given example is done by determining the difference between the network's processed output and the target output. The network then adjusts its weighted associations using this error value, usually with backpropagation. The output of the network is materialized by the activation of the neuron(s) of the last layer, and is therefore abstract [21]. It is in this abstraction that lies the difficulty to interpret it.

2 Related work

General methods have already been proposed to explain the results of a model, independently of the model used: LIME [23], FairML [2], Sensitivity analysis [7], Auditing [3], Counterfactuals [26]... These approaches usually work in two ways. Some make a direct link between input data and output data, highlighting relevant inputs to the network's decision. Others focus on selecting representative examples of specific patterns recognized by the network.

More recently, given the wide adoption of neural networks and its black-box problem, a number of methods designed for neural networks have been proposed, e.g. Layer-wise relevance propagation [6] and DeepLIFT [25]. These strategies take into account the architecture of the neural network, explain how it works, and use this explanation to clarify the results obtained.

As shown in the Figure 1, these methods generally quantify the impact of each neuron from the previous layer on the activation of the neurons of the next layer, by looking back in the network. The equation used to calculate the impacts is called the propagation of the relevance score, both for the deep Taylor decomposition [20] and for the method called excitation backprop [27]. This equation is defined as follows:

$$R_{j \leftarrow k} = \frac{a_j w_{jk}^+}{\sum_i a_i w_{ik}^+} R_k \tag{1}$$

with a_j the activation of one neuron j, and w_{jk} the weight associated to neuron j in the calculation of neuron k activation (the + denotes the fact that, in this equation, only positive weights are considered). i denotes each neuron of the previous layer. Finally, R_k defines the relevance score contained into neuron k. This calculation allows to measure the normalized impact of a neuron j of the previous layer on the activation of a neuron k of the next layer.

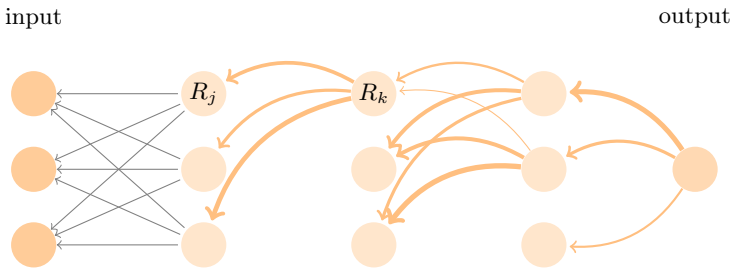

Fig. 1. Schema illustrating the method of Layer-wise Relevance Propagation [21].

Going back to the input layer, we can measure the impact of each input element on the output, as shown in Figure 2.

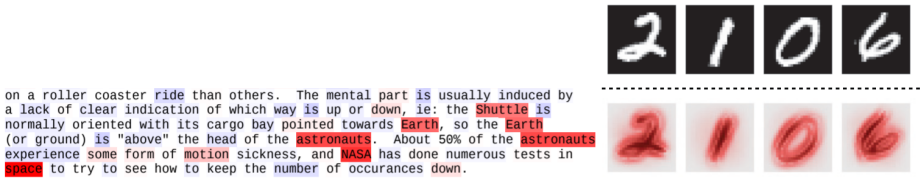

Fig. 2. Left: Relevant words on the theme of space [5]. Positive relevance is mapped to red, negative to blue. Right: Relevant pixels in the recognition of each number [21]

The work of Bau and his colleagues [8] takes this reasoning a step further, by placing a concept (i.e. a word) on the pattern recognized by each hidden unit in a convolutional neural network (CNN). For instance, a neuron (i.e. hidden unit) could recognize the color pink, trees, cats, etc. This process is automatically performed by gathering hidden units' responses to known concepts and quantifying the alignment between hidden units and concepts, and then by selecting the top ranked label(s) (cf Figure 3). The network structure can then be interpreted as a set of recognized patterns.

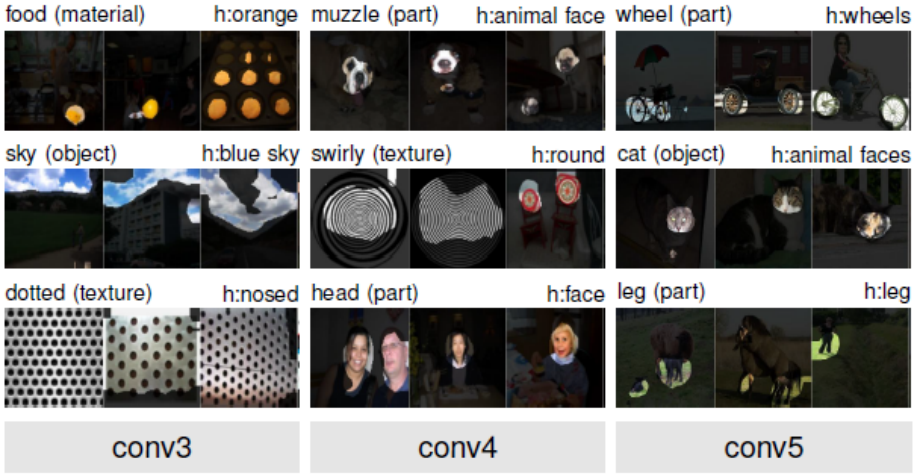

Fig. 3. Patterns identified by some units of a CNN (top left) and human annotated labels (top right) [8]

However, such experiments have only been observed on pictures. An image has the advantage of being directly interpretable by one of the human senses, which is not the case for most feature sets given as input to a neural network, and in particular tabular datasets.

Also, all the previous approaches (including Bau's work [8]) focus on the explainability of neural networks dealing with non-tabular data: images, texts, etc. In this kind of networks, the definition of the input features is unstable over the examples. For instance, a first picture may show a dog's head on the bottom left, and a second a dog's head on the top right. While in both cases a dog's head is present, the pixels in the bottom left corner will define a dog's head only for the first example. Therefore, we cannot give the same meaning to each pixel over the entire dataset, but we can do so with each cell of a tabular dataset.

To address these limitations, we will provide in this paper a new approach to explain networks dealing with tabular data. We will focus specifically on autoencoders. This neural network architecture offers the advantage of doing unsupervised learning, as opposed to other neural approaches. In this case, it is essential to understand how the network works to understand its output.

An autoencoder (AE) is a specific neural network architecture (cf Figure 4). Its aim is to obtain an output as similar as possible to the input. However, its architecture forces the AE to summarize the input information into a limited number of central variables. Then, the network is forced to decode this representation to create an output as close as possible to the input.

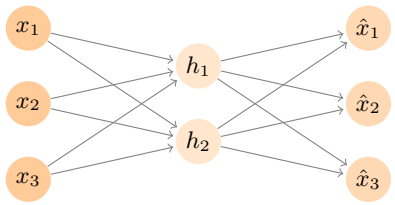

Fig. 4. Structure of an autoencoder: $\hat{\mathbf{x}}$ is an approximation of $\mathbf{x}$ through a learned representation $\mathbf{h}$.

To explain the middle layer encoding, many research works have tried to define each central neuron with the concept that it recognizes [9]. Thus, a model trained on faces could recognize the size of the nose, the length of the hair, the color of the eyes, etc. Some AEs have been developed recently to achieve this task known as feature disentanglement: Total Correlation VAE [10], Wassertein AE [24], InfoGAN [11]. Other approaches, e.g. [15], try to explain the output rather than intermediate layers explaining the output.

However, two problems arise from these proposals.

First, none of them seems to rely on the structure of the AE. The methods make a direct link between inputs and outputs by empirically placing a concept on the pattern recognized by each central neuron.

Second, as we can see in a more general way for neural networks, the current research works mainly deal with non-tabular data, which is directly interpretable by humans (e.g. images, text). Yet, in many cases, the input is just tabular data,

a collection of variables that humans do not know how to organize as a whole. It is important to provide explainability for these situations.

This paper features two main contributions:

- A method to define the pattern recognized by each neuron of the latent representations generated with autoencoders, but also to define the pattern of each neuron of the network.
- This method will be generalized to several kinds of problems (binary classification, multi-class classification, regression) to bring clear and compact explanations of the results.

3 Propositions

3.1 Overview

Our proposal is based on 2 ideas of the literature. These proposals were used on non-tabular data (pictures, text, etc.), and we adapt and combine them on tabular data:

1. Compute the impact of the neurons from the previous layer on the neurons of the next layer.
2. With these impact scores, define layer after layer the pattern recognized by each neuron. Doing so, explain the network as a set of patterns combined together to allow the recognition of more and more complex patterns.

In practice, we first compute the impacts of the input features on the first layer. Using these impacts, we define in a few words the concepts recognized by each neuron of the first layer. This technique is repeated, progressing by one layer at a time, until the central layer of the AE is reached and defined.

3.2 General impact scores

We calculate general impacts received by each neuron. These impacts must be understood as the mean impact of the neurons from the previous layer on the activation of the concerned neuron, and this over the whole dataset used to train the network. The goal of this step is to understand the pattern recognized by the neuron.

Our impact calculation is based on the equation 1. However, this equation was made for non-tabular data, which means that each neuron didn't define the same concept over all a dataset. In our case, we take advantage of the consistency of variables by using covariance. Doing so, we measure the link between each neuron of consecutive layers for the whole dataset, and not only at the level of an example:

$$I_{j \leftarrow k} = \frac{cov(a_j, a_k) w_{jk}}{\sum_i |cov(a_i, a_k) w_{ik}|} I_k \tag{2}$$

Here, I_k defines the total amount of impact contained into neuron k, and other annotations are identical to equation 1. It is worth noting that we chose the covariance over the correlation as covariance preserves the impacts of standard deviations whereas correlation normalizes it.

3.3 Patterns definition

The central idea of our work is, for each neuron, to combine the definitions of the most impactful inputs into a new definition, which would define the pattern/concept recognized by the concerned neuron. The definition of the neuron is, in the current state of the method, created manually by the user by combining the impacts automatically calculated by the program.

To understand the definition of a concept, let's consider the example of a neural network that estimates house prices. Table 1 (left) hypothetically describes the impacts acting on a neuron. Here, we could define our neuron as "size of the living areas". The activation of the neuron will be high if the size of the kitchen, the living room and the veranda are high, and conversely if they are low.

In some cases, as for table 1 (right), it is complicated to find a definition that gathers all the main impacts. In this example, we can define our neuron as "the size of the garage and the bathroom". This neuron will be strongly activated if the size of the garage and/or the bathroom is high, and vice versa. We have to keep in mind that the neural network does not necessarily detect patterns that make sense for the human cognition.

Neuron definition	Impact
Size of the kitchen	0.41
Size of the living room	0.25
Size of the veranda	0.14
Size of the toilet	0.09
...	...

Neuron definition	Impact
Size of the garage	0.55
Size of the bathroom	0.29
Size of the office	0.12
Size of the toilet	0.07
...	...

Table 1. Housing price example - impacts on a neuron

3.4 Specific impact scores

To explain the activation of a neuron for a specific example, the specific impacts of that example are calculated. Let's take a look at table 1 to illustrate this need. Sometimes the neuron is activated because the kitchen is very large, while the living room and the veranda are of normal size. In other cases, the neuron can be activated while the kitchen is normal size, but this time the living room and the veranda are above average size. Thus, in both cases, the pattern "size of the living areas" is well recognized, but a more specific explanation allows to shed light on the reasons for this activation. Also, in some rarer cases, the neuron is

activated while the kitchen, the living room and the veranda are of average size, but the toilet is exceptionally large. These specific impacts can also help us to understand outliers.

To compute the specific impacts, we first calculate, for the specific example, how much the activation of a previous neuron differs from its mean activation (computed over all the dataset):

$$\delta(a_{j,z}) = a_{j,z} - \overline{a_j}, \text{ where } \overline{a_j} = 1/n \sum_{k=1}^{n} (a_{j,k})$$

with j the index of the neuron, $\overline{a_j}$ the mean activation of the neuron over all the dataset, z the index of our given example, $a_{j,z}$ the activation of the neuron on our example.

Then, our goal is to measure how much this deviation impacts the activation of the neuron k of the next layer. By transposing equation 2 to this application, each specific impact $\mathcal{S}$ is calculated as follows:

$$\mathcal{S}_{j,z} = \frac{\delta(a_{j,z}) w_{jk}}{\sum_i |\delta(a_{i,z}) w_{ik}|}$$

with k the index of the neuron of next layer, w_{jk} the weight linking neuron j and neuron k, and $\sum_i \delta(a_{i,z}) w_{ik}$ the sum of deviations on the whole layer.

3.5 Network requirements

The next points are not mandatory to use the method on an AE, for example if one wants to keep a high complexity of the network. However, meeting these requirements will lead to a highly explainable result.

Network architecture The architecture of the neural network must follow some rules to efficiently define the pattern recognized by each neuron.

Pruning: Our human cognition does not allow to group a large amount of parameters into a single pattern. So, to be able to define the pattern of a neuron, we limit the number of links between each layer. This is called pruning [16]. Each neuron is connected to a number n of neurons of the previous layer. It constrains the neuron to determine a pattern from this limited set of inputs.

Number of layers: A balance must be found between:
- A neural network deep enough to be able to cross patterns and obtain complex results.
- A neural network sufficiently shallow to be conceptualized as a network of patterns by the human cognition.

Considering this, we limit the encoder part of the AE to 3 layers: 1 for inputs, 1 hidden layer, and 1 for the central layer. Later, when generalizing to other types of architectures, the network is made of 4 layers: 1 for inputs, 2 hidden layers, and 1 for outputs.

Network selection To define interesting and diversified patterns, we want the network to restrict its use of the same input neurons. Let's take again the example of estimating the price of a house. For a first network, the neuron defined as "size of the land" has a big impact (more than 0.1) on 6 neurons of the next layer. For a second network, this same variable impacts only 2 neurons of the following layer. We realize that, for the first network, the definitions are likely to be more redundant than those of the second network, because they will often focus on the size of the terrain. Therefore, we want to limit the use of the same neurons in the definitions to obtain a more diversified vocabulary.

To this end, we build several networks. A score is calculated for each of them, and the network with the lowest result is chosen. To compute the score of a network, we count for each neuron the number of times that its impact on a neuron of the next layer is greater than 0.1. Then, we select in these numbers of appearances those that are greater than or equal to 3, and we sum them squared. The result of this sum is our score. By lowering this score, the number of too frequent appearances of the same variables in our definitions is reduced. In some cases, some variables will keep a high number of appearances, which is expected when these variables explain a large part of the result we are looking for.

Note: we do not use entropy as we only care about avoiding neurons being too frequently present in the explanations, but we do not want to force the network to use neurons that would not be useful in the calculation of the result. This is important especially for generalizing the method to classification and regression.

This method can be used to accomplish various tasks. It first offers the possibility to observe the general functioning of a network, to understand the patterns recognized within the different layers. It also allows to explain the result obtained for a specific example. The code of our work can be found at: link.

4 Results and Discussion

4.1 Autoencoder explanation

To study the explainability of AEs, we used the dataset created by the research team of Robert Joel Lewis [19]. This dataset deals with films and indicates for each production the number and ratio of spoken words that belong to several categories. For example, it includes for each movie the number of words referring to work, money, sadness, friendship, etc. We selected 25 of these categories and created for each movie a representation made of 13 features.

Performance with pruning The balance between performance and explainability is one of the main concerns of XAI. Often, the highest performing methods are the least explainable, and the most explainable are the least accurate [14].

Figure 5 depicts this dilemma between performance and explainability, according to the pruning performed. For each neuron of the model, the more the 3

most impactful neurons of the previous layer contain a high quantity of impact, the more it will be likely to give a relevant definition to the pattern recognized.

Looking at figure 5, the loss is higher for the most limiting prunings. But on the other hand, the more limiting the pruning is, the higher the impact contained in the 3 most impactful neurons. In our case, we choose to continue with a pruning restricting to 5 back connections per neuron, which allows on average to obtain an summed impact of about 0.95 in the 3 most impactful neurons. However, we could also choose to work with a larger number of connections to increase the performance of the network, but we have to keep in mind that this increase would lead to less pattern-matching definitions.

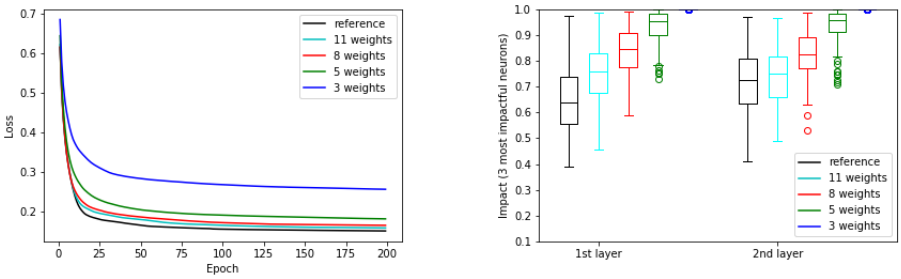

Fig. 5. Learning curves for various pruning limitations (left) and sum of the impacts contained in the 3 most impactful neurons (right). The number of weights must be understood as the number of connections between a neuron and the previous layer.

Definition process The definition process goes layer by layer. We first compute the impacts of the input data on the first layer, and use these impacts to create a definition of the neurons. The same process is repeated for the next layers. As a reminder, in the current state of the method, definitions are created manually by the user by combining the impacts calculated by the program.

For example, in our case, we obtain for the first layer some definitions like "No achievement or reward", "Affiliation, positive emotions, men", "Death and anger, no positive emotions", "No negative emotions or risk", "Anger/swear/bad health", "No leisure or positive emotions, but power". These definitions are meaningful in terms of movies, we can easily envision the types of films that do or do not fit. However, some definitions may be less explicit to our cognition. For instance, "Affiliation with risk but no reward" may fit the movies that approach a social relationship in a risky context, but without a link with reward/benefit/prize.

After having carried out this stage of definition, we end up with 13 descriptions, which describe the neurons of the central layer. The next part focuses on two meaningful examples to make these results and their applications explicit.

Central representations: examples The first example is the film The Game of Their Lives (2005). The movie is based on the true story of the U.S. soccer team which, against all odds, beat England during the 1950 FIFA World Cup.

We represent this film in the table 2, making the link between the definitions created through our method and the standardized activations observed on the central layer. The table shows that several values deviate from the mean. We first observe neurons 1 and 5, where low values are explained by the abundance of leisure and positive emotions in the movie. Neuron 4 is pulled up, once again thanks to the importance given to leisure and positive emotions, but also thanks to the fact that many men are represented, with the notion of affiliation. Neuron 6 has a value well below the average, this result is due to the great importance of achievement and reward in this movie, with the opposition between two teams. Finally, neuron 11 is strongly activated mainly because affiliation and risk are promoted.

Index	Neuron definition	Activ.
1	No leisure/positive emotions, but power/sex/work	-2.58
2	Bad health and work	-0.54
3	No men and movie focused on present or future	-0.2
4	Leisure, positive emotions, men, affiliation, with possibly anger/swear/bad health	4.31
5	Power, no leisure/positive emotions/affiliation, also no anger/swear/bad health	-2.48
6	No achievement or reward	-4.62
7	Death/anger/power, no positive emotions or leisure	-1.02
8	Bad health/death/anger/work	-0.68
9	Friends, affiliation, positive emotions, men	0.63
10	Friends, affiliation, positive emotions, men, without negative emotions or risk	0.5
11	Affiliation with risk and negative emotions	2.12
12	No women or family/affiliation, no negative emotions/sadness	-0.1
13	Friends, affiliation, positive emotions, men, with a bit of religion	0.7

Table 2. The Game of Their Lives - Encoded representation

A second example is the movie Go, Go, Second Time Virgin (Yuke Yuke Nidome No Shojo, 1969). This very dark film tells the story of Poppo, a teenage girl, who was raped by four boys. Tsukio, a teenage boy, has been watching the rape passively. Over the course of a day and a night, Poppo and Tsukio begin a relationship, telling each other of their troubled past and philosophizing about their fate. When the gang returns and again rapes Poppo, Tsukio kills each of them and their three girlfriends. The story ends with Poppo and Tsukio both jumping off the apartment roof to their deaths.

We show this movie in the table 3, again standardizing each of the central activations. First, neuron 3 indicates a rather high value compared to its average. This is mainly due to the fact that the movie deals very little with men, focusing on the two children. Then, neuron 5 is very negative because the topics of anger, swears and bad health are strongly present in the film. The same applies to neuron 8, adding the subject of death. Neuron 6 is activated because of the absence of achievement or reward. Neuron 7 is very strongly activated because death, anger and the absence of positive emotions are predominant. Neuron 10 is

weakly activated due to the absence of friendship, affiliation, positive emotions, men, and the presence of negative emotions and risk. Neuron 12 is negative mainly because the film contains a lot of negative emotions and sadness. Finally, neuron 13 is very negative because friendship and affiliation are not developed, as well as positive emotions, men and religion.

Index	Neuron definition	Activ.
1	No leisure/positive emotions, but power/sex/work	-0.52
2	Bad health and work	0.64
3	No men and movie focused on present or future	1.19
4	Leisure, positive emotions, men, affiliation, with possibly anger/swear/bad health	-0.2
5	Power, no leisure/positive emotions/affiliation, also no anger/swear/bad health	-2.64
6	No achievement or reward	1.28
7	Death/anger/power, no positive emotions or leisure	7.75
8	Bad health/death/anger/work	2.52
9	Friends, affiliation, positive emotions, men	-0.26
10	Friends, affiliation, positive emotions, men, without negative emotions or risk	-3.57
11	Affiliation with risk and negative emotions	-0.33
12	No women or family/affiliation, no negative emotions/sadness	-1.93
13	Friends, affiliation, positive emotions, men, with a bit of religion	-4.6

Table 3. Go, Go, Second Time Virgin (Yuke yuke nidome no shojo) - Encoded representation

Specific impacts: example We now focus on explaining the activation of a specific neuron. Looking again at table 2, we observe that the activation of neuron 10 ("Friends, affiliation, positive emotions, men, without negative emotions or risk") is positive, but not that high for a movie focused on a male football team that wins a match. We would like to understand this activation. To do so, we compute the specific impacts of the neurons of the previous layer on the activation of neuron 10 (table 4).

In this table, we can observe the definition of each neuron of the previous layer, their general impacts on the neuron 10 (central layer), and their specific impacts for the movie "The Game of Their Lives". For instance, the neuron 2 is activated when the movie refers to friendship. Its general impact on the central neuron 10 is equal to 0.59, which is very high. It means that, in the data, when the references to friendship are above the average, then neuron 2 strongly positively impacts the activation of the central neuron 10. In parallel, it also means that when friendship is below the average, neuron 2 strongly negatively impacts the activation of the central neuron 10. On the side of specific impact, we can see that this neuron has a value of 0.15 for the movie The Game of Their Lives, which means that neuron 2 was activated above average and positively impacted the activation of the central neuron 10.

So, to finally explain why the activation of central neuron 10 is only of 0.5, we observe that neuron 15 has a strong negative specific impact. This is due to the fact that the movie contains a lot of negative emotions and risk, which leads neuron 15 to an activation well below its mean. However, central neuron

Index	Neuron definition	General impact	Specific impact
1	No achievement or reward	0	0
2	Friends ++	0.59	0.15
3	Friends ++	0	0
4	Affiliation with risk but no reward	0	0
5	Not sad, + money	0.05	0.05
6	Male ++	0	0
7	Death and anger, no positive emotions	0	0
8	Leisure but no affiliation, no focus past	0	0
9	Bad health, a bit of work	0	0
10	Focus on past	0	0
11	No women or family, no negative emotions	0	0
12	Not sad, bit of religion	0	0
13	Focus on past, no risk	0	0
14	Men without anxiety	0	0
15	No negative emotions or risk	0.17	-0.37
16	Anger/swear/bad health, affiliation	0	0
17	Sex and work, no focus on future	0	0
18	No leisure or positive emotions, but power	-0.04	0.21
19	Affiliation, positive emotions, men	0.14	0.22

Table 4. The Game of Their Lives - Specific impacts on the neuron 10: "Friends, affiliation, positive emotions, men, without negative emotions or risk"

10 has still a positive activation of 0.5 because all the other specific impacts are positive, and this is explained by the fact that the movie treats a lot about friendship, happiness, leisure, positive emotions, affiliation, and men.

4.2 Generalization to other problems

To generalize our approach, we apply it to 3 different types of problems: binary classification, multi-class classification and regression. For comparison purposes, these same problems are also solved using decision trees.

Binary classification For this type of problem, we use a Kaggle dataset on the survivors of the Titanic disaster [18]. The objective is to predict whether a passenger survived the sinking or not.

We create a neural network made of 4 layers: a first one for the input data, then 2 hidden layers and a last one for the output. After defining each pattern recognized by the neurons of the network, we finally obtain the following output definition: "woman, a bit elevated fare, good class, generally young". This pattern is easily understandable by the human cognition and makes a lot of sense in terms of classification. The closer the passenger's profile is to this description, the higher the probability of survival, and vice versa.

In parallel, when using a decision tree that achieves a similar level of performance, the node rules first take into account the sex of the person, then the class and fare, and finally the age. Similarly to the neural network, the features used to predict a survivor are female sex, a good class, a fairly high fare and a rather young age.

Multiclass classification For this second type of task, we choose a dataset created by Davide Anguita and his collaborators [4]. This dataset focuses on the recognition of 6 types of physical behaviors: lying, sitting, standing, walking on a flat surface, walking downstairs and walking upstairs. To this end, a smartphone is placed on the subject's waist. 12 features are then selected to describe the movement of the device, using its accelerometer (for the acceleration over 3 axes) and its gyroscope (for the rotation over 3 axes).

Similarly to the binary classification, we create a neural network made of 4 layers: a first one for the input data, then 2 hidden layers and a last one for the output classes. We focus on the definitions of the output classes.

- For the lying position, the main impacts reveal the following definition: "Not standing, back of the waist and/or ribs towards ground, no change in altitude (the phone doesn't move over the vertical axis)".
- For the sitting position, the result is "Back of the waist orientated towards ground, waist doesn't move".
- For standing, "No change in altitude, waist doesn't move, trunk upright".
- For walking on a flat surface, the main impacts lead to "Trunk upwards, hips symmetrically moving back and forth, and up and down".
- For the walking downstairs, "Big changes in altitude (the phone is moving quickly over the vertical axis), not standing still".
- Finally, for walking upstairs, "Trunk upwards, ankles symmetrically moving back and forth, changes in altitude, moving forward".

As a comparison, the decision tree starts by separating walking activities from still activities using the standard deviation of the acceleration along the vertical axis, which describes the changes in altitude during the period. Then, on the stationary activity side, the standing position is separated from the other two using the mean acceleration on the vertical axis. In a second step, the lying position is partly separated from the sitting position by using a threshold for high values of antero-posterior acceleration. In fact, when lying down, gravity no longer acts on the vertical axis but on the antero-posterior axis. On the other side, regarding walking activities, walking downstairs is partly detached from the two other activities by thresholding the very high values of standard deviation on the vertical axis: when walking downstairs, the altitude of the phone changes very quickly. In a second step, a part of walking upstairs is individualized by selecting the high values of forward and slightly upward acceleration. These nodes allow the classification of only about 80% of the data, and a large number of additional nodes is needed to improve performance.

Regression In this last part, we are interested in regression, using a dataset introduced by Kaggle [17]. This dataset is focused on the prediction of the final ranking of a player in a PUGB game. The ranking is materialized by a value between 0 and 1, where 1 corresponds to first place and 0 to last place.

Like previously, we create a neural network made of 4 layers: a first one for the input data, then 2 hidden layers and a last one for the output ranking. As a

result, our method leads to the following definition of the output: "Walks a lot, makes a lot of damages (on opponents and vehicles) and uses health bonuses".

In parallel, a decision tree with a comparable level of performance first bases its decision on the walking distance and then uses many different parameters depending on the case.

5 Conclusion and Perspectives

We have introduced a way to explain autoencoders and, more generally, neural networks. This method can be applied to all kind of tabular data. By putting words on the pattern recognized by each neuron, it allows for a clear and simple explainability of the network and of its results.

However, this approach points out several limitations of the explainability of neural networks, due to the limits of human cognition and human-made concepts. First, we are only able to put words on relatively simple patterns. Although each node of a neural network is able to cross tens, hundreds, thousands of inputs, it is hard for humans to gather more than 4 or 5 features under a single concept. Thus, the networks are created taking this limitation into account and may present a lower performance compared to networks only oriented towards performance. On the other hand, we can also choose to work with less pruned networks that perform well, but we have to keep in mind that this leads to less pattern matching definitions.

Secondly, our culture and experience of the world restricts the diversity of concepts. The network can create links between elements that do not necessary make sense for humans. Our reasoning is impacted by the way we experience the world, but the optimal result of the neural network does not take into account this culture, which would be for him a limit to the performance.

Therefore, the limits of our method, and more broadly of the explainability of tabular data problems, are not caused by a computational problem but rather by the limits of human cognition.

The next step of our work will consist in applying this method to other datasets, but also in comparing the results with other XAI approaches, to prove the relevance of our work. Later, we could also imagine to automate the creation of definitions through natural language processing techniques. One suggestion would be to automatically compute, for each node, the distance between the definitions of the 3 biggest impacts (neurons from the previous layer) and a set of words/concepts, to automatically create a new definition based on the closest words.

References

1. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access **6**, 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052

2. Adebayo, J.: FairML: ToolBox for Diagnosing Bias in Predictive Modeling. Ph.D. thesis, MIT (2016)
3. Adler, P., Falk, C., Friedler, S.A., Rybeck, G., Scheidegger, C., Smith, B., Venkatasubramanian, S.: Auditing Black-box Models for Indirect Influence. arXiv:1602.07043 [cs, stat] (Nov 2016), arXiv: 1602.07043
4. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L., others: A public domain dataset for human activity recognition using smartphones. In: Esann. vol. 3, p. 3 (2013)
5. Arras, L., Horn, F., Montavon, G., Müller, K.R., Samek, W.: "What is Relevant in a Text Document?": An Interpretable Machine Learning Approach. PLOS ONE **12**(8), e0181142 (Aug 2017). https://doi.org/10.1371/journal.pone.0181142, arXiv: 1612.07843
6. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE **10**(7), e0130140 (Jul 2015). https://doi.org/10.1371/journal.pone.0130140
7. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Mueller, K.R.: How to Explain Individual Classification Decisions. arXiv:0912.1128 [cs, stat] (Dec 2009), arXiv: 0912.1128
8. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network Dissection: Quantifying Interpretability of Deep Visual Representations. arXiv:1704.05796 [cs] (Apr 2017), arXiv: 1704.05796
9. Charte, D., Charte, F., del Jesus, M.J., Herrera, F.: An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. Neurocomputing **404**, 93–107 (Sep 2020). https://doi.org/10.1016/j.neucom.2020.04.057
10. Chen, R.T.Q., Li, X., Grosse, R., Duvenaud, D.: Isolating Sources of Disentanglement in Variational Autoencoders. arXiv:1802.04942 [cs, stat] (Apr 2019), `http://arxiv.org/abs/1802.04942`, arXiv: 1802.04942
11. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. arXiv:1606.03657 [cs, stat] (Jun 2016), arXiv: 1606.03657
12. DARPA: Explainable Artificial Intelligence (XAI) (2016)
13. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
14. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—Explainable artificial intelligence. Science Robotics **4**(37), eaay7120 (Dec 2019). https://doi.org/10.1126/scirobotics.aay7120
15. Haghighi, P.S., Seton, O., Nasraoui, O.: An explainable autoencoder for collaborative filtering recommendation. arXiv preprint arXiv:2001.04344 (2019)
16. Janowsky, S.A.: Pruning versus clipping in neural networks. Physical Review A **39**(12), 6600–6603 (Jun 1989). https://doi.org/10.1103/PhysRevA.39.6600
17. Kaggle: PUBG Finish Placement Prediction ItemType: dataset
18. Kaggle: Titanic - Machine Learning from Disaster ItemType: dataset
19. Lewis, R.J., Grizzard, M., Lea, S., Ilijev, D., Choi, J.A., Müsse, L., O'Connor, G.: Large-Scale Patterns of Entertainment Gratifications in Linguistic Content of U.S. Films. Communication Studies **68**(4), 422–438 (Aug 2017). https://doi.org/10.1080/10510974.2017.1340903, `https://www.tandfonline.com/doi/full/10.1080/10510974.2017.1340903`
20. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recog-

nition **65**, 211–222 (May 2017). https://doi.org/10.1016/j.patcog.2016.11.008, https://linkinghub.elsevier.com/retrieve/pii/S0031320316303582
21. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (Feb 2018). https://doi.org/10.1016/j.dsp.2017.10.011, https://linkinghub.elsevier.com/retrieve/pii/S1051200417302385
22. Pomerleau, D.A.: Analysis of Network Representations. In: Neural Network Perception for Mobile Robot Guidance, pp. 85–106. Springer US, Boston, MA (1993). https://doi.org/10.1007/978-1-4615-3192-0_6
23. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938 [cs, stat] (Aug 2016), http://arxiv.org/abs/1602.04938, arXiv: 1602.04938
24. Rubenstein, P., Schölkopf, B., Tolstikhin, I.: Learning disentangled representations with Wasserstein Auto-Encoders. Workshop Track Proceedings (2018)
25. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 [cs] (Apr 2014), http://arxiv.org/abs/1312.6034, arXiv: 1312.6034
26. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. arXiv:1711.00399 [cs] (Mar 2018), http://arxiv.org/abs/1711.00399, arXiv: 1711.00399
27. Zhang, J., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down Neural Attention by Excitation Backprop. arXiv:1608.00507 [cs] (Aug 2016), arXiv: 1608.00507