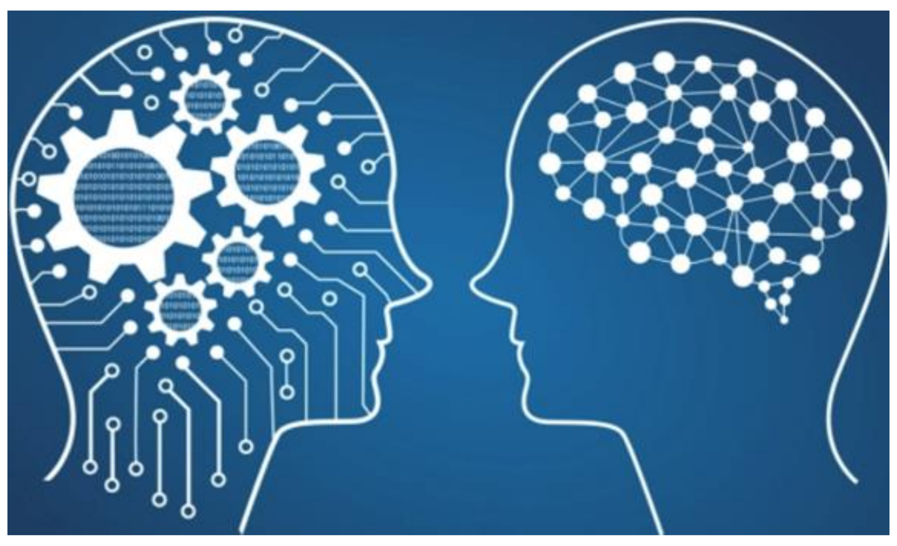
Validation of stroke-risk prognosticator algorithms using CNN and Microsoft Azure

The Project

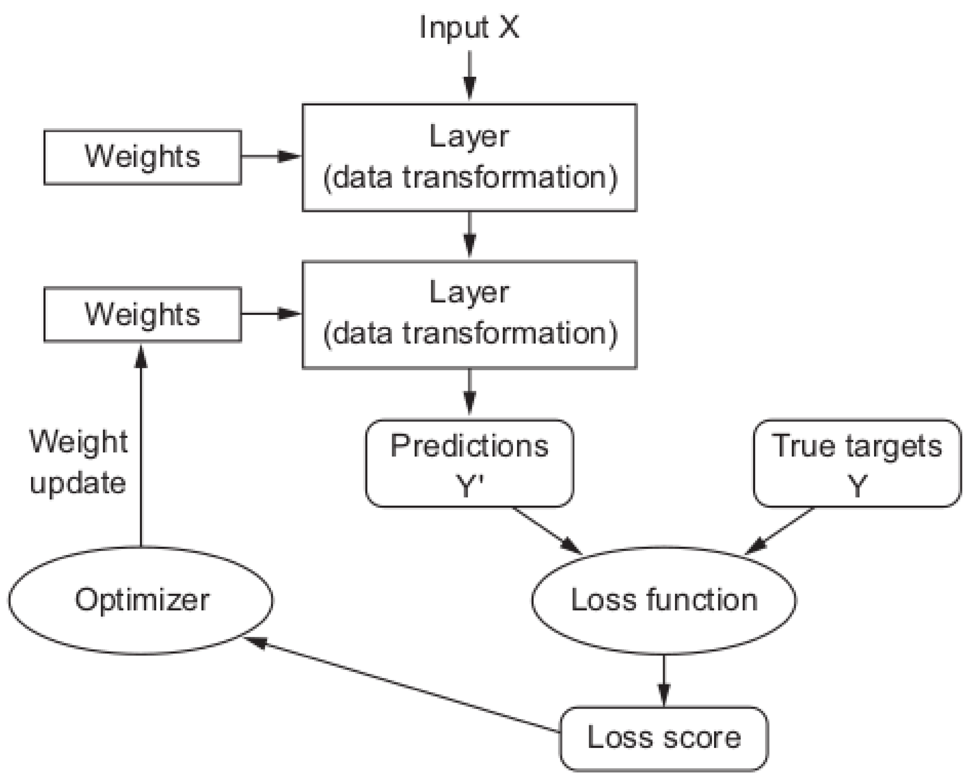


Foundations

Nowadays more and more people talk about **Artificial Intelligence**. The **AI** is a very useful thing, it can help a lot. But lots of article light it grim. One of the most useful ways of use it making diagnosis. And by learning about it and working with it, we – as second year BSC students – tried to make good use of it in our project. For it we used ***Deep Learning*** to do it.

The **Artificial Intelligence** is very wide definition. It contains ***Deep Learning***, **Machine Learning** and every learning and hard coded approach. A good example for the hardcoded way is the first chess programs. A lot of rules and strategies was burn to the code. Some group of programmers believe in **the human level AI**. They wrote programs with manipulating rules. This way is called **symbolic AI.**

In **Machine Learning**, we determine the rules from the input and output data, rather than using rules to determine some input data. It's trained instead of explicitly programmed. The most popular and the most interesting way of **AI** is the ***Deep Learning***. This is a special approach of **Machine Learning**. We used this to build a ***Neural Network*** to try to make stroke-risk prognosticator algorithms and validate them and other algorithms by their help.



About the Neural Network

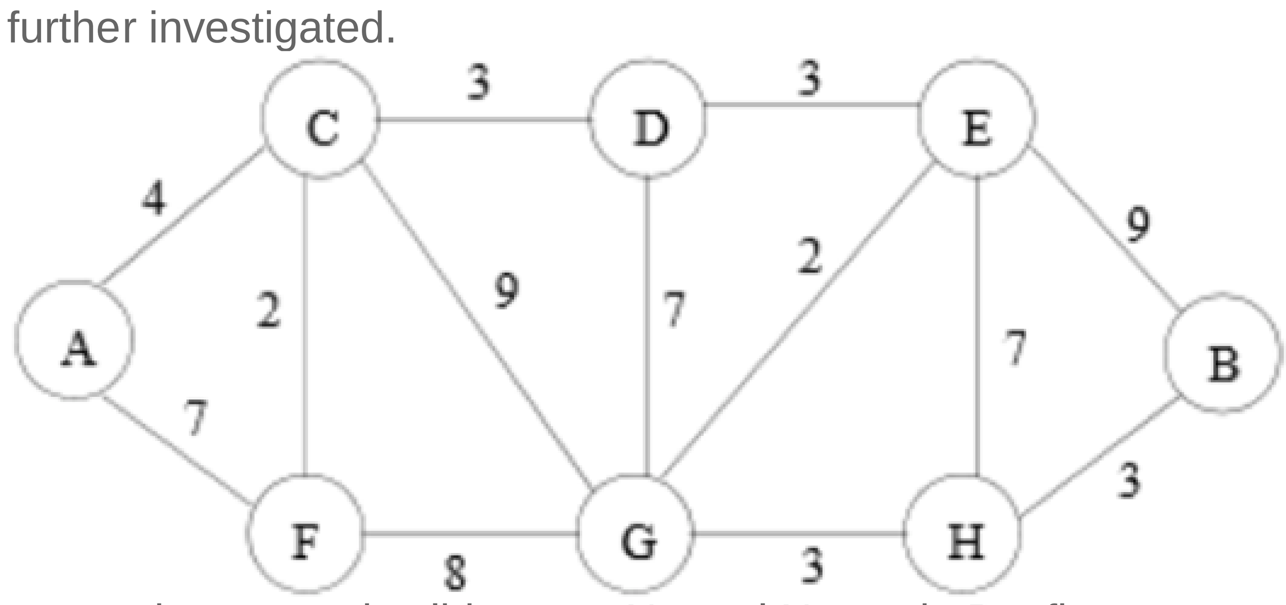
In this picture you can see how the ***Neural Network*** works. There is the X. The X is the input data, which is filtered by the **Layers**. And we get predictions about what the output will be, this is the Y'. We now the true target, the Y (usually the user gives it). There can be differences between the two data. The rule of the loss function is calculating the difference, which is called loss score and give it to the optimizer. The optimizer sets the weights of the layers to become more accurate.

When we speak about ***Neural Network***s, there are three types of data. **Train, test** and **validation data**. With the **train data**, we can teach the network, optimize for the work.

Maybe, in some cases the **train data** can be not very representative. The **validation data** is used to **validate the network**. For testing the network, we use the **test data**. In this project, we used a ***CNN*** (it is a special form of ***Neural Network***s), which try to make stroke diagnosis. To do this we feed graphs into the ***CNN*** in the form of adjacency matrices. The matrices represent the links between a people's illnesses. The program handles it like a picture. **CNN** see the data as a black picture with white dots. White dots are the links.

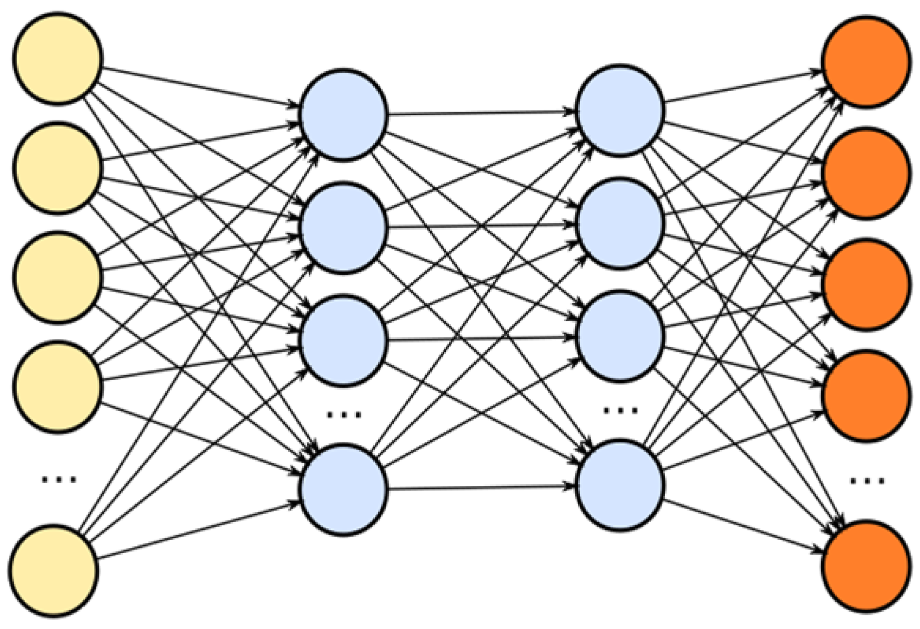
Sequential Networks are used with ***Dense*** (the ***Layers*** relate to each other) ***Layers***, ***Dropout*** (some data aren't in use) and ***Regularization*** (choose the smallest value). The layers use different activation functions to improve the hypothesis space. ***Hypothesis Space*** means the operators which is used by the ***Layer***s and the ***Loss Function***. Whit using a Neural Network we believed we can achieve a fair prognostication, which can help to call attention for potential dangers, so they can be further investigated.

The used data



We use patients' anonymized data. We operate with weighted graphs of patients' illnesses, where the sickness is represented as their BNO-10 codes. These codes in the graph is connected, if they were occurring during the same medical examination. Their connection also has weights, which represents the number of times the 2 symptoms were witnessed during the same time. With these graphs, we try to establish the relations between the illnesses and the disease. The stronger the connection is, the more likely that they're connected somehow, and the fact, if the illness is just barely represented in healthy data can confirm these hypotheses further. For now, these graphs are summed from the last 2 years before the patient had a stroke, in the future we plan to consider the time, as a factor, and try to give more serious warnings for patients whose conditions became more alarming as time goes by.

For our project, we transform our graphs into adjacency matrices with fixed dimensions, and we can use said matrices to train, test and validate our Neutral Network. But first, we preprocess these converted graphs, so it can be processed more easily. First, the small graphs get filtered out, because they would be just noise. After that, the remaining gets sorted, summed and compressed, and then the data gets processed by a ***Convolutional Neural Network***.



The Layers and the specific Network

The ***Neural Networks*** are using many of layers. This is the fundamental data structure. They are data-processing module, you can think of as a filter for data. We train multiple layers in our network to count a probability index from patients' data with these. ***Deep Learning*** and especially ***Convolutional Neural Networks*** uses fully connected layers, so every parameter change also has influence over every output. Every layer is built by ***Neurons***. These each represents a filter, which gets convoluted with the whole input tensor and with this it tries to predict how the searched patterns are can be found in the original image. These also have weights, which gets updated by the backpropagation algorithm. With these, the network tries to find its best form. For our project, we used this mathematical model to try to find connections between them.

For our project this means that every patients data, after conversion and preprocessing, gets convoluted by illnesses and their connections. The network found the most common connections and their weights, so it convolutes them with each other. If it's found enough connection, which it thinks to connect strongly to the decease, then it considers it as a potential threat. It gets reported to the doctor, who can order further inspections and special examinations.

Validation

And what we want to do with it? We want to compare our current algorithm to other ones, so we validate all of them in the same test sets (which are different than the training and validation sets) to see each model's goodness. With this information we can allocate the best algorithm for the job and we can validate their usefulness compared to each other, even if now we only can compare a ***CNN*** to a random probability. "It is better than that?" can be a valid question to ask, because the ***CNN*** tries to find patterns, and with a big matrix like this is not easy. We will continue our research by solving and validating the problem with Graph Neural Networks. Their actual usefulness is unknown for us, but when we finish, we will be able to use the best mathematical model as the solution for our problem. With this we aim to make a real time AI network, what can send warning to doctors about potential threats, so they can investigate it further.

Ambitions

Strokes are treacherous and very dangerous disease. There are symptoms such as vestibular disorder, left or right side of face is paralyzed, not clear speech. If you see these on a person, then don't wait! Call an ambulance! Every minute decrement the chance of becoming healthy again. The stroke can come from nowhere and can cause a great trouble in someone's life. Most people can't get back their health. Usually the people become paralyzed. They cannot work and need special treatment. More and more under 60 years old people get stroke as time passes. A few years ago, the average age of the stroked people was between 70-80, but nowadays this is between 50-60.

With this we want to help reduce completely unforeseen and sudden strokes. Because we believe that this can help doctors to see the risks just in time to make the necessary examinations and actions to prevent tragedies. And even though it's risk can be foreseen from patients' path to the disease at 100%, but it can help to predict it.

References

FRANÇOIS CHOLLET - Deep Learning with Python

Sebastian Raschka & Vahid Mirjalili - Python Machine Learning (second edition)

Acknowledgement

This project was made by Attila Tiba, Ákos Mándi, Matteo Giusti, Máté Vágner, Tamás Bérczes and Tamás Girászi.
This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.