Clinical Brain Computer Interfaces

Pintilie Andrei, Tabarcea Augustus

March 22, 2020

1 Introduction

1.1 Motivation

2 Method

2.1 1D Convolutional

As we saw in[1] a very successful approach to time series is to use a *Convolutional neural network*[2]. We used a popular architectureFigure 1 to create a goal for our NeuroEvolution algorithms.

2.2 NEAT

Following the paper[3] we applied topological evolution of the artificial neural network used. NEAT recommendation is to start with a small network and evolve from there, so we started with just the input and output layers. The mutation is made by changing a property (connections,weights,bias) of a chosen node to mutate. NEAT handles crossover this by keeping track of the origins of the nodes, with an identifying number (new, higher numbers are generated for each additional node). Those derived from a common ancestor (that are homologous) are matched up for crossover, and connections are matched if the nodes they connect have common ancestry. The probabilities for mutation and crossover are underlying properties of the package that we use, but they let us choose the probability for a change in the network as follows:

- A node can either be disable,deleted or enabled,added. In our case this probability is $\mathcal{P}(N_+) = 0.2$ and a $\mathcal{P}(N_-) = 0.2$, where $N_+ =$ adding a new node and $N_- =$ removing a node.
- A node can either gain a new connection or lose it. In this case we use $\mathcal{P}(C_+) = 0.5$ and a $\mathcal{P}(C_-) = 0.4$, where $C_+ =$ adding a new connection and $C_- =$ removing a connection.
- A node can modify its weights. We use for that a $\mathcal{P}(W_C) = 0.8$ and a $\mathcal{P}(W_R) = 0.1$, where $W_C =$ change the weight by adding or subtracting

with value from a gaussian distribution centered on 0
and W_R = replacing the weight value.

3 Experiment

4 Results

4.1 Interpretation

5 Conclusions

Appendices

References

[1] Deep learning for time series classification by Hassan Ismail Fawaz, Germain Forestier,Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller [2019] `https://arxiv.org/pdf/1809.04356.pdf`

[2] 1D Convolutional Neural Networks and Applications by Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman [2019] `https://arxiv.org/ftp/arxiv/papers/1905/1905.03554.pdf`

[3] Efficient Reinforcement Learning through Evolving Neural Network Topologies by Kenneth O. Stanley and Risto Miikkulainen [2002] `http://nn.cs.utexas.edu/downloads/papers/stanley.gecco02_1.pdf`

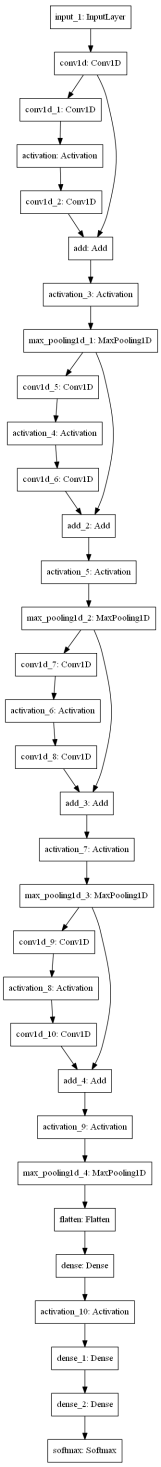

Figure 1: Conv1D Arhitecture.