

Wojciech Oziębły
Wojciech Pratkowiecki
[Repository](#)

Project report

Project for the Neural Networks and Deep Learning 2017 class taught at the University of Wrocław.

This project is a reproduction of **Code Synthesis with Priority Queue Training** by Daniel A. Abolafia, Quoc V. Le, Mohammad Norouzi: <https://openreview.net/pdf?id=r1AoGNIC->

Introduction:

Based on paper results we expected our net to cope with some simple tasks programs. Due to a limitation of resources we tested our code generator on 6 tasks:

- reverse
- shiftl
- shiftr
- echo half
- hello
- len

with number of executed programs (NPE) = 2.5M

Results:

After 2.5M iterations we collected the results of our net. It managed to generate acceptable BF code for **reverse** - passed 6 tests from 7 taken, and **shiftl** (4/7). Unfortunately the results for **shiftr**, **echo half** and **len** are bad, the net wasn't able to find proper output. For **hello** task the problem is lack of tests, there is just one test case - empty input and "hello" as output. As the reward function is based on Hamming distance for sequences of different length, the net found some BF codes that generate strings of length 5 (equal with "hello") and it was given some satisfying reward. Therefore our generator didn't try to find some better solution and all we were given were programs that yields "_____".

Comparison with the paper:

We expected our results to be a bit worse than those achieved by the paper authors, because of the lower NPE during training. You can learn from the paper that the net should deal with all of tested tasks but **shiftr** and **echo half**, generated programs should be able to produce correct output in most cases. The authors trained their net with NPE = 20M, so the net generated almost 10 times more programs to execute and learn on. However 2.5M iterations turned out to be enough to find solutions for **reverse** and **shiftl**. It seems logical as those tasks require only simple loop. Unfortunately the net failed to find solutions for rest of tested tasks. We believe that increasing the number of iteration would significantly help to

find correct BF codes and our results would be even more similar to those collected by the authors.

Summary:

The day that neural networks will replace programmers is still far away.

Acknowledgments

The authors thank Google for GCE Credits awarded through Google Cloud Platform Education Grants to the Neural Networks and Deep Learning course and to this project.