

Final-project

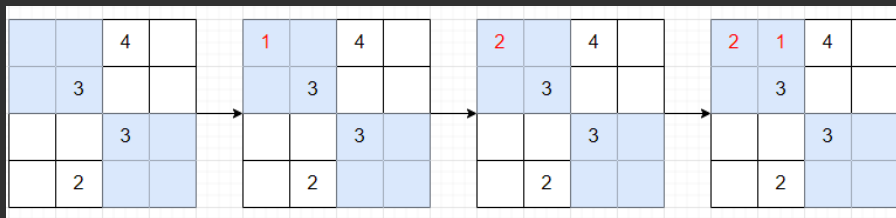
**Different Programming technics to solve
sudokus**

Daniel Lahrmann

March 11, 2022

- 1 Recursive backtracking
- 2 Logical Algorithm
- 3 Using Neural Networks

The algorithm



Results

- relative easy to implement
- can solve every possible puzzle
- very slow (5min 39s for 10 puzzles)
- needs 1000 - 20000 guesses of single cells

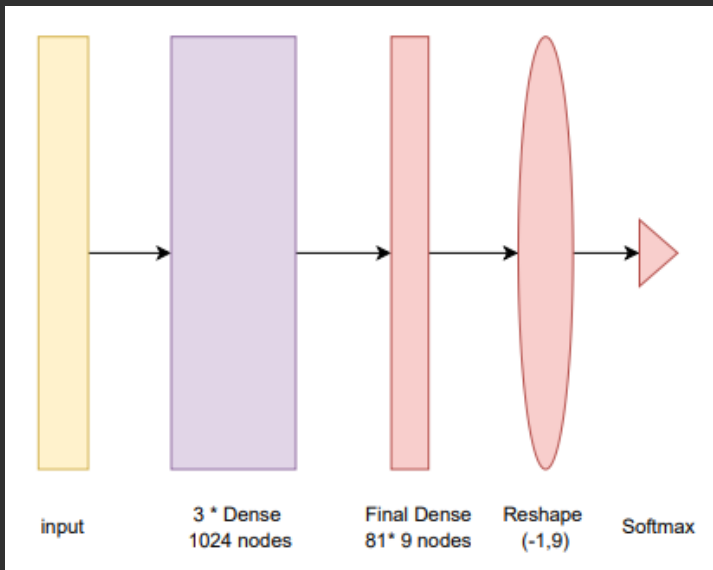
Logic algorithm

- fast solving of the puzzles (2min 36s for 100 puzzles)
- 100% accuracy if algorithm works
- fails if a new trick will be needed
- high effort for implementations
- (<https://www.sudokuwiki.org> listed 38 different tricks and can fail without Recursive Backtracking)

First remarks before modeling

- multi-label classification problem
- puzzle and solution have to be label decoded (solution:0-8)
- works better if input is normalized

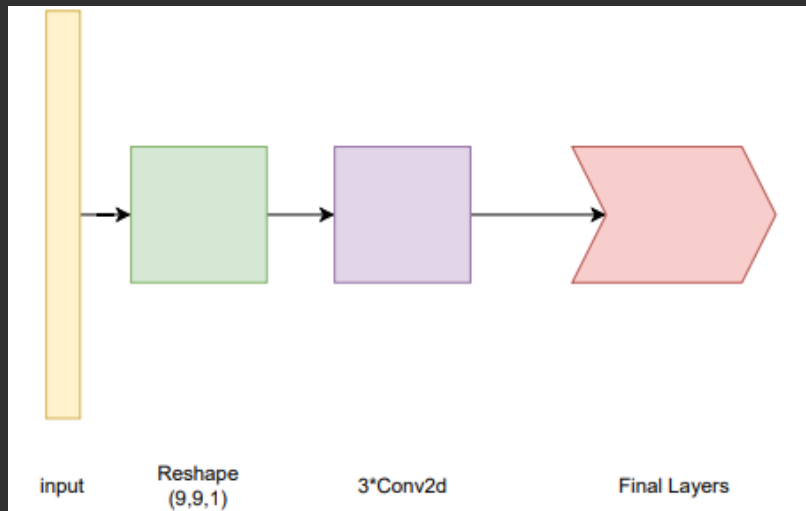
Dense Neural Network



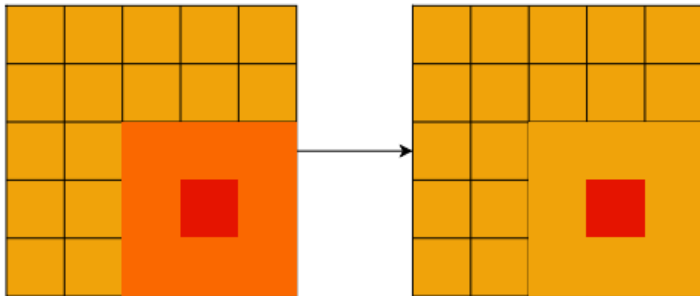
Dense Neural Network:Results

- `model.evaluate` computed accuracy: 45%
- accuracy of solved puzzles: 0

Convolution Neural Network



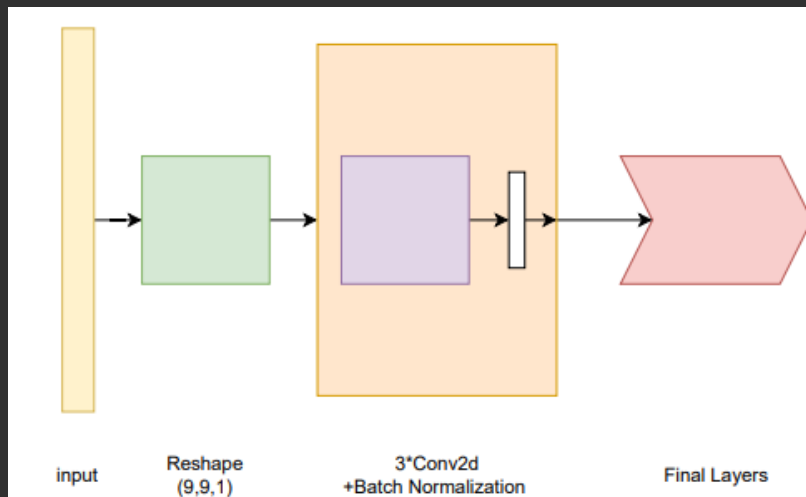
Convolution Layer



Convolution Neural Network: Results

- `model.evaluate` computed accuracy: 82%
- accuracy of solved puzzles: 0
- trick: place empty digits one by one
- accuracy of solved puzzles: 93%
- time: 2 min 51s for 100 puzzles
- adding a new layer results in a worse result

Advanced Convolution Neural Network



Convolution Neural Network: Results

- `model.evaluate` computed accuracy: 84%
- accuracy of solved puzzles: 100%
- time: 2 min 55s for 100 puzzles

Conclusions of Neural Network

- implenting them includes try and error
- slower as the self made algorithm
- needs a lot of data for modeling (1000000 different puzzles)
- but can solve harder puzzles (test:18% accuracy)
- does not need knowledge about solving sudokus
- existing of the possibility of become faster than the algorithm

Conclusion for other problems

- Als: greates tool for solve problems
- its possible to beat the AI by using clever assumptions
- also see: IronKaggle, SpeedRunner against Racing AI
- better use combination of both approaches