Adem Atmaca
Adrian Sauter

Heisenberg-Documentation

I am the one who knocks

Why "Heisenberg"?

The name of our network, "Heisenberg", comes from the show "Breaking Bad". In the dianachess-competition, our chess agent "Jesse" made 3rd place, so we thought of sticking with the series. And since the neural network in this competition has a lot more neurons than our chess-agent had (which had 0 neurons since he was not learning anything), it´s only plausible to call this project Heisenberg, considering he is the brain of their partnership :)

General Architecture

We followed two very different approaches, both of which made use of convolutional layers, since they perform really well when working with pictures.

- First approach: HeisenbergXXL
  Our first network was very large and contained some convolutional layers followed by some fully connected layers. The convolutional layers were connected via maxpooling and ReLu-activation functions. The fully connected layers were then also connected via the ReLu-functions except for the last layer, which was the output layer and used the sigmoid activation function (to generate a value between 0 and 1).

- Second approach: HeisenbergXXS
  Basically following the same CNN-approach, HeisenbergXXS was just much reduced in its number of parameters. We found that combining some convolutional layers with only one fully connected layer performed on the level of HeisenbergXXL with a drastically reduced number of parameters. After trying out a bunch of different combinations (different kernel_size for the Conv-layers, different numbers of channels, different maxpooling sizes etc), we found some good variants.

Activation function

We found the ReLu-functions to be best for learning in this kind of environment.

Maxpooling

We used maxpooling to further reduce the size of the image which goes into the fully connected layer(s).

Optimizer and learning rate

We found the Adam-Optimizer to be performing best in this task. We then tried a couple of different learning rates and found the learning rate 1e-3 to be working best in our networks. Since the Adam-optimizer updates every parameter with an individual learning rate, we didn´t use our own learning rate scheduler.

Data augmentation

After exploring some of the data, we decided to only mirror the images vertically. We even considered rotating the Images, which leads to more data to train on, but boats which are on the side of the image would be cut out, leading to for example empty images of water having a target of 1, which would only cause the model to be confused and not learning properly.

Final validation accuracy

Our big model as well as our small model achieved 99.80% as the final validation accuracy.