Training SAI

BY F. MORANDIN

Abstract

Some considerations, based on AlphaZero paper, on the training hyperparameters in the case of no-gating pipeline.

1 AlphaZero parameters

Careful analysis of the two versions of AlphaZero paper (the one on arxiv and the published Science paper), together with supplementary materials and available pseudocode, allowed to get a good understanding of the project pipeline.

AlphaZero trained for 700k steps, playing 140 million self-play games (21 million for the first run, from the arxiv paper, from now called AlphaZero Symmetries or AZ-S). Learning rate was 0.02 for 300k steps, 0.002 for 200k steps and 0.0002 for 200k steps.

Each step used a minibatch of 4096 positions. Every 1000 steps the training output one network. The window size for training was the last 1 million games, without data augmentation (apart for AZ-S). The positions for the mini-batch were chosen uniformly among all positions of the games in the window.

From this it is easy to see that there were 700 generations and each one played on average 200k games (30k games for AZ-S, with 8-fold data augmentation) and hence each training covered training data from 5 generations (no information for AZ-S). Thus every position had $4096 \times 1000 \times 5$ chances of being chosen among the $1000000 \times L$, where L is the average length of a game. This yields that training used about 20 positions per game.

2 Formalization of constraints

As an abstraction of the setting, consider the *mini-step model*, where at each mini-step the current network generates k new games (corresponding to $k\,L$ new positions), and then trains the network with just 1 position choosen uniformly from those generated in the last M mini-steps.

For AlphaZero, each generation corresponds to perform 4 million mini-steps and to play 0.2 million new games (0.24 million for AZ-S), hence $k = 0.05$ (0.06 for AZ-S), corresponding again to a density of 20 positions per game. The training window covers 5 generations, that is the positions generated by the network in the last 20 million mini-steps, hence $M = 20$ mln.

The parameters k and M should probably be chosen near those of AlphaZero in similar experiments, even if there is no proof that they were actually optimized there. Anyhow with those values it is known that the training progressed quite smoothly.

The main constraint should be on k, as k^{-1} encodes the density of positions used for training. If k is too small, then overfitting could be possible; if k is too large, then resources are wasted.

The other parameter M encodes how far behind training positions should be taken. If it is too large, then the training stays longer than needed on the same set of old games, and so the progress may be slowed without reason. If it is too small, then the training may not be given time and data

enough for learning the subtle features needed to progress.

3 SAI pipeline

In our pipeline, at each generation the network is trained for a variable number n of steps (optimized by promotion matches), with minibatches of size 128. Then the network plays $s = 5120$ self-play games, which are symmetry-augmented 8 times. The training window covers g generations with current typical value $g = 16$.

It is immediate to compute k and M as functions of the above parameters, yielding

$$\frac{8\,s}{128\,n} = k \geqslant 0.05, \qquad 128\,n\,g = M \simeq 20\,\text{mln}$$

This gives the following suggested reference values

$$n \leqslant 1.25\,s, \qquad g \simeq \frac{156250}{n}$$

For $s = 5120$, in particular, one gets $n \leqslant 6400$, yielding that for $n = 6000$ we should use $g \simeq 26$, for $n = 6400$, $g \simeq 24$, for $n = 8000$, $g \simeq 20$, for $n = 10000$, $g \simeq 16$.

Since we moved to the 9×192 network (currently 22 generations ago), we used on average $n \simeq 6300$ (optimally selected by promotion matches), which is surprisingly near to the AlphaZero bound, with an average window size of $g \simeq 15.8$ generations.

We may conclude that we are somewhat low in the number of training generations, meaning that the learning could be impaired a bit since it hasn't enough data to learn subtle features, but on the other hand that if the data is enough, progress should be faster than AZ.

4 Strength progress

For AlphaZero there were two runs. The first one, from the arxiv paper, there declared without the use of symmetries, actually reappears on the Science paper as AlphaZero Symmetries (AZ-S).

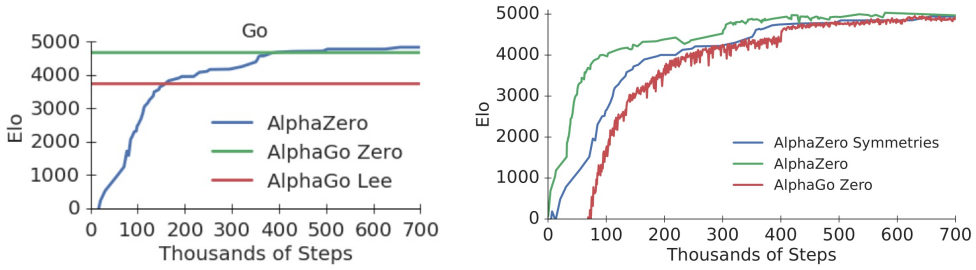

Figure 1. Elo charts from arxiv and Science versions of AlphaZero paper. It is apparent that the blue plots are the same. Also notice that AlphaGo Zero starts from a negative value (the complete plot can be found in the Nature paper, where it starts from about -3500); for AGZ the initial random play level was not anchored to 0 as it appears to be for AlphaZero. The above plots should probably be considered inconsistent for low Elo values, as the peculiar change of concavity between Elo 0 and 1500 suggeests.

To analyze progress, we zoomed in the picture (which fortunately was vectorial) and took the positions of the edges with some precision.

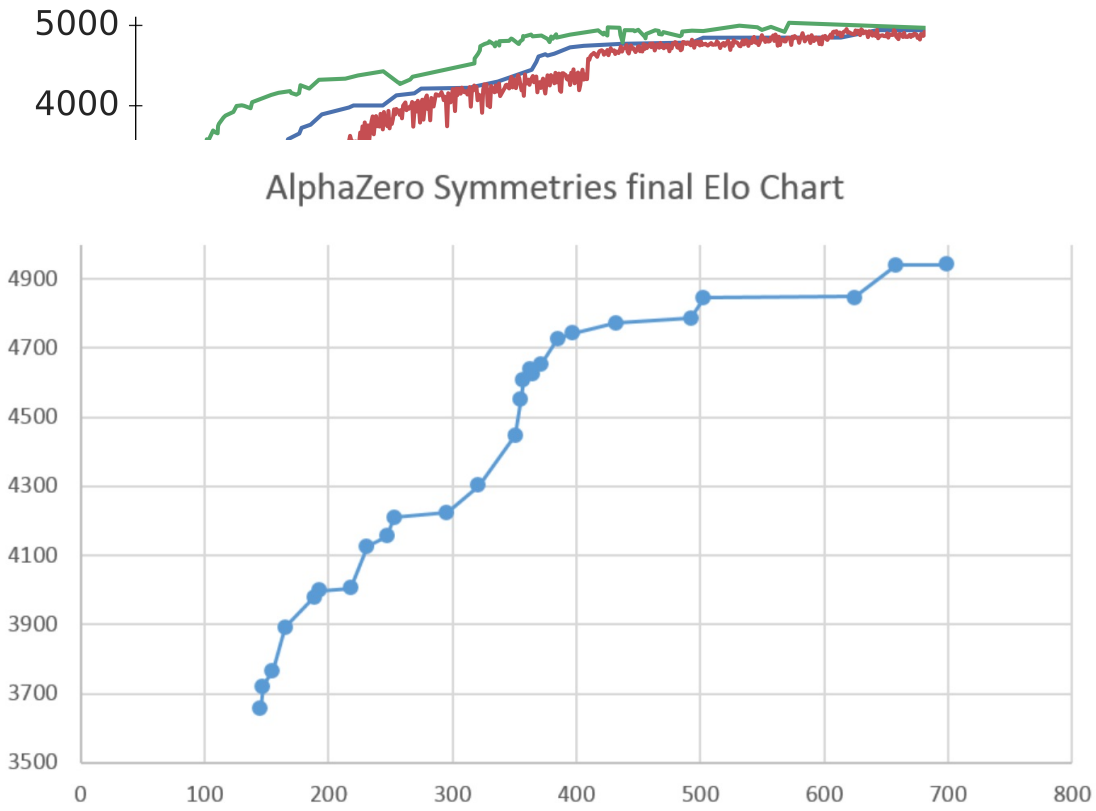

Figure 2. Zoomed progress of AZ-S in its final stage, after the initial fast growth ended. The above plot allows to compute the derivative of Elo score.

Derivative between 150k and 500k steps varies in the range of 2-5 Elo points per generation, corresponding to 0.5–2.5 Elo per million training positions. For SAI pipeline, considering $n = 6400$ training steps per generation, this would corresponds to 0.4–2 Elo per generation.

Our current estimated progress (considering 22 generations of 9×192 networks) is in fact much larger, at about 4 Elo points per generation (4.2 removing outlier SAI157, 3.6 keeping it).

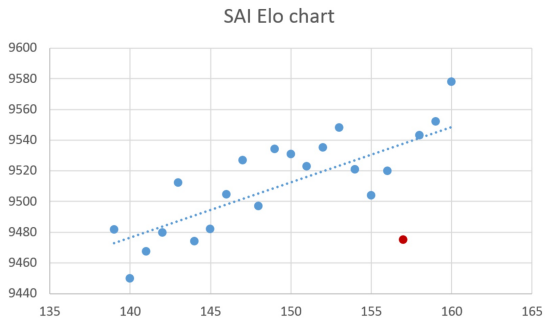

Figure 3. SAI strength progress for the first 22 generations in the 9×192 training. The red dot, SAI157, can probably be considered an outlier.

We may conclude that the current pipeline is in fact quite succesful, with a faster progress than AlphaZero Symmetries though with quite wide strength oscillations. The training window size should be tested at all times, because the AZ recommended value is quite larger.