DOMAIN KNOWLEDGE IN EXPLORATION NOISE IN ALPHAZERO

Anonymous authors
Paper under double-blind review

ABSTRACT

The AlphaZero algorithm has achieved remarkable success in a variety of sequential, perfect information games including Go, Shogi and chess. In the original paper the only hyperparameter that is changed from game to game is the alpha parameter governing a search prior. In this paper we investigate the properties of this hyperparameter. First, we build a formal intuition for its behavior on a toy example meant to isolate the influence of alpha. Then, by comparing performance of AlphaZero agents with different alpha values on Connect 4, we show that the performance of AlphaZero improves considerably with a good choice of alpha. This all highlights the importance of alpha as an interpretable hyperparameter which allows for cross-game tuning that more opaque hyperparameters like model architecture may not.

1 INTRODUCTION

AlphaZero is a general reinforcement learning algorithm designed to play games like Go and chess (Wang et al., 2019)(Silver et al., 2017b; 2018; 2017a). It has achieved superhuman performance on such games, without explicitly learning from any human gameplay (Ryan, 2018; Silver et al., 2017b). That said, AlphaZero certainly benefits from human insight for each of those games, because knowledge of the properties of good search coupled with an interpretable hyperparameter allows AlphaZero to be tuned in such a way as to play many different games well. Often hyperparameters come in the form of model architecture choices that can be opaque and challenging to optimize for a given problem. There is literature, including the work of (Wang et al., 2019), dedicated entirely to finding the correct settings of hyperparameters of AlphaZero. There are even entire algorithms developed to tune the hyperparameters of AlphaZero and similar algorithms (Wu et al., 2020). This kind of tuning allows for improved performance, but comes at a massive computational cost. Hyperparameters must be tuned from problem to problem, and as a result this process becomes expensive going from problem to problem. As a result, as they do in Silver et al. (2017b), many times researchers reuse hyperparameters hoping that the problems are "close enough" that changes are not necessary. The one hyperparameter in AlphaZero that is changed game to game is the α value, because it can easily be adjusted to account for differences in game branching factors. With this parameter, the authors are specifying a certain trade-off between exploration and exploitation that they believe requires tuning for optimal performance.

This ability to tune based on intuition is powerful. It is likely that other hyperparameters, including model architecture, are not optimal for the three different games, but because of an inability to build intution around these hyperparameters, they are left alone. In this paper we explore the intuition around the choice of the search parameter, examine the empirical differences in search with a dirichlet prior, and finally do a more formal hyperparameter sweep that demonstrates that sufficient intuition allows for a strong choice of hyperparameter. We show that for searches with large enough branching factors, a Dirichlet prior that is appropriately aligned with information about the Monte Carlo Tree Search (MCTS) task could significantly improve performance over a uniform prior. We demonstrate the strong flexibility afforded by introducing α as a hyperparameter, showing how its helps the search in toy problems. However, this flexibility comes at the cost of needing correct bias to work well, as deeper search becomes more sensitive to choice of α.

We are able to make an informed choice of α with intuition about what kind of search behavior is likely beneficial for an agent for different types of games. This implies that this intuition and helpful

choice of α is among the many choices that make up the implicit inductive assumptions that allow for AlphaZero's success, as is the case for all successful learning and search systems (Schaffer, 1994; Montañez, 2017; Montañez et al., 2019). That is, AlphaZero may not have had its parameters tuned as a result of human gameplay, but there are aspects of AlphaZero that benefit greatly from human experience.

1.1 ALPHAZERO OVERVIEW

AlphaZero utilizes "self-play" and a variant on the MCTS to play discrete, symmetric, two-player games. AlphaZero's MCTS utilizes a multi-headed neural netork to predict values and future actions for different hypothetical states in a game tree, and exploration dictates the final action distribution. In its search, AlphaZero stores and updates exploration probabilities at each node depending on a combination of several factors including the neural network policy output, results of simulated games, the number of visits to a given node, and the root has an additional prior distribution controlled by the hyperparameter α. The technique is more likely to explore a less visited or higher reward state as determined by experience and the policy network (Silver et al., 2017a). The hyperparameter α influences the balance between trusting the networks predicted policy or the experience and exploring the problem space.

Selection of α is an exploration vs. exploitation trade-off, and empirical evidence suggests that the optimal exploration-exploitation trade-off varies across different games (Silver et al., 2018). This suggests that changing α can contribute significantly to success in AlphaZero, and we will explore this further in Section 2.

2 FAVORABLE BIAS THROUGH HYPERPARAMETER TUNING

2.1 DIRICHLET DISTRIBUTIONS

A Dirichlet distribution is a continuous probability distribution over the space of possible categorical distributions (Lin, 2016). Generally, the Dirichlet distribution over β categorical variables is parameterized by a β element vector of positive entries, $\boldsymbol{\alpha}$. In our case, the relevant family of Dirichlet distributions are centered about the uniform distribution, and thus every entry of $\boldsymbol{\alpha}$ is equal. Thus, we will use α to denote this scalar value in each entry. For this special case, the probability density function is given by

$$f(x_1, x_2, \ldots, x_\beta; \alpha) = \frac{\Gamma(\beta\alpha)}{\Gamma(\alpha)^\beta} \prod_{i=1}^{\beta} x_i^{\alpha-1}.$$

Intuitively, a large positive value of α indicates a high probability of selecting the uniform categorical distribution, $\alpha = 1$ indicates a uniform probability of selecting any categorical distribution, and a very small value of α indicates a high probability of selecting a categorical distribution that heavily favors a particular label.

2.2 EXPLORATION IN ALPHAZERO

Exploration in AlphaZero is partially driven by a Dirichlet distribution. AlphaZero's search is similar to that of Polynomial Upper Confidence Tree (PUCT) (Rosin, 2011). This strategy initially favors nodes with high prior probability and low visit count and asymptotically prefers high scoring moves. In this algorithm, the action taken from each node is chosen by the formula: $a_t = \text{argmax}_a(Q(s_t, a) + U(s_t, a))$, with

$$U(s, a) = c_{puct} P(s, a) \frac{\sqrt{\Sigma_b N(s, b)}}{1 + N(s, a)}.$$

In the above formula c_{puct} is a hyperparameter to weight the trade-off between recent simulated experience and the neural network trained policy, $N(s, a)$ is a term that counts the number of times an action has been taken from a given state, and

$$P(s, a) = (1 - \epsilon)p_a + \epsilon\eta_a,$$

where $\epsilon = 0.25$ at the root, and 0 otherwise. We see ϵ is a hyperparameter that tunes the weight given to the search prior, and in AlphaZero the search prior was only used at the node. We see p_a is the output of AlphaZero's policy network and η_a is sampled from a symmetric Dirichlet distribution parameterized by α. In testing AlphaZero across different games, the authors changed the parameter α of the symmetric Dirichlet distribution as in Table 2.2. In doing so the architects of AlphaZero were

Table 1: Chosen α Values Compared Against the β Values of Relevant Games (Silver et al., 2017a; Matsubara et al., 1996).

Game	Chess	Shogi	Go
α	0.3	0.15	0.03
Approx. β	35	80	250

able to successfully inject a significant amount of inductive bias into their program based on an intuitive understanding of the parameter, and this inductive bias has been shown by Mitchell (Mitchell, 1980) and Montañez et al. (Montañez et al., 2019) to be a necessary precondition for successful learning. We will explore exactly what this means intuitively further in discussion of the toy game.

2.3 Bias of Dirichlet

Research into search problems has shown that achieving better performance than uniform random sampling requires a bias that aligns with the underlying structure of a problem (Montañez, 2017). Without appropriate bias, search can do no better than uniform random sampling (Montañez et al., 2019). Viewing Monte Carlo Tree Search in this light, we can examine the bias in bits of a sampled Dirichlet prior as compared to a uniform prior. This bias caps the improvement a Dirichlet prior driven search can give over uniform search, and determines the proportion of problems this set of biases will do well on. The higher the bias, the more the algorithm is tailored to the specific search problem, and the less the general AlphaZero is with a fixed α. Bias, measured in bits, with respect to the hyperparameter α and the branching factor of the search tree β is given by

$$\text{Bias}(\alpha, \beta) = H(\mathcal{U}) - \mathbb{E}_{X \sim Dir(\alpha)}[H(X)]$$

where $\mathcal{U}$ is the uniform distribution, and $H(\cdot)$ is the differential entropy. The expected entropy of a categorical distribution drawn from a symmetric Dirichlet distribution is known to be $\mathbb{E}[H(X)] = \psi(\beta\alpha + 1) - \psi(\alpha + 1)$, where ψ is the digamma function (Nemenman et al., 2001). Graphed over relevant branching factors and values of α, we see from Figure 1 that as α decreases the bias increases. Asymptotically, as α approaches zero, $\mathbb{E}_{X \sim Dir(\alpha)}[H(X)]$ approaches 0, so when α is sufficiently small, the bias is equal to the entropy of the uniform distribution for a fixed branching factor, or $\log \beta$. This means generally that a smaller α has greater bias, so in some sense AlphaZero only performs well on a relatively smaller set of games with a fixed α.

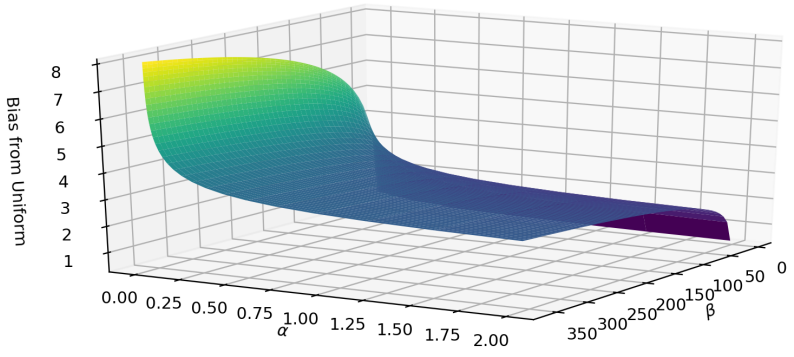

Figure 1: Effect of β and α on Injected Bias.

2.4 ABSTRACT GAME CASE STUDY

Armed with these results about the injected bias, we can explore the how different priors can influence search and improve performance for different games and branching factors.

In order to try to isolate the influence of α on the search of a tree, we created a simple toy simulation that highlights the advantage of using a Dirichlet prior with specific α over a uniform prior in some types of games. We focused on trees that have sparse rewards several steps ahead in tree, which would be analogous to playing a game with sparse positive reward signals available only after several moves, which is representative of the types of games for which AlphaZero has had success. In this simulation, we assumed the game was a full tree of a certain depth, and a player could explore up to 200 nodes. At the specified depth, each node has a 5% chance of having reward 1, and otherwise has reward 0. At each non-leaf node, if the node is unexplored, a categorical distribution is sampled from a symmetric Dirichlet distribution with parameter α, and the next node choice is sampled from that categorical distribution. After exploring a node, the agent starts again from the root and chooses a path based on that nodes categorical distribution, and continues recursively. If the algorithm visits a leaf with reward 1, it is considered successful. If the algorithm repeated a visit to a leaf, that was also considered an expansion in order to penalize revisiting parts of the game tree that do not represent reward. We tested a variety of α values on different branching factors to find which choices of α lent themselves to successful searches for given branching factors and depth of reward.

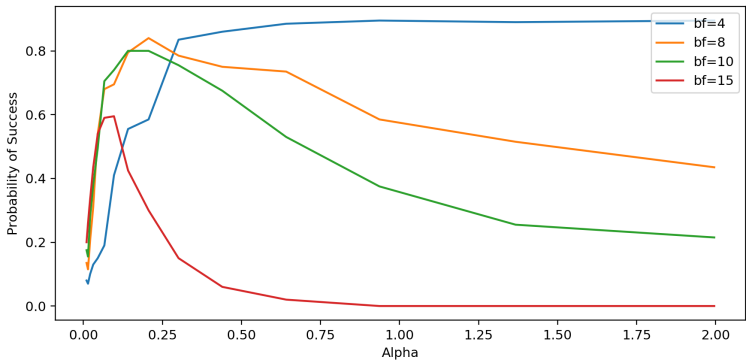

Figure 2: Success Probability with Reward Depth 5.

Figure 2 demonstrates the importance of aligning bias appropriately with the problem. We assert that these results and trends match intuition. For example, at small branching factors, we see that larger alpha values lead to a higher probability of success in this game. High α values correspond to a more uniform distribution sampled from the Dirichlet, and so this means that there isn't a need for large bias to succeed in this type of task for small branching factors. However, as the branching factor increases, we see empirically that the range of α values that achieve good results decrease, and the best results come at smaller and smaller α. This indicates that at high branching factors the problem becomes harder and requires more bias away from a uniform distribution, making the selection of favorable α much more critical.

Repeating this experiment across several different branching factors with rewards at several different depths, we found that in general as the depth of reward increased, the α value that led to the most successful search decreased (Figure 3), and as a result the mode depth explored for the most successful strategies increased. In Figure 3, the shaded regions around the solid-line averages capture the inner 90% of 20 trials, where each trial consists of 50 tree explorations across 30 α values between 0.0025 and 1, with the final score of a trial being the proportion of successful explorations to total explorations.

These results show that for games like this with high branching factor and more sparse, deeper reward, success is more sensitive to choice of the appropriate small α value, and a smaller α value corresponds to higher bias from uniform search.

Furthermore, as the average depth of the reward increased, the mode depth of exploration increased (Figure 4), and the corresponding successful α values were smaller (Figure 3) and more sensitive to changes. Since a lower α means greater bias (cf. Figure 1), higher branching factor problems

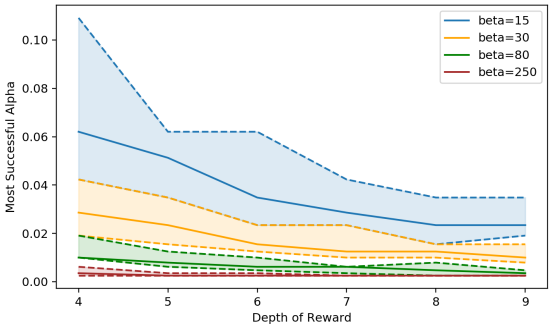

Figure 3: Successful α Across Different Depths.

Figure 4: Mode Depth Explored.

and problems that require a deeper search require more biased search that is more sensitive to α values. That said, while α values were more sensitive, this data suggests the intuition of the AlphaZero architects with respect to changing α to correspond with branching factor was a reasonable approximation. This really represents Silver et al. (2017b)'s belief that occasional exploration to a depth of one or two from an action that may initially have the best prospect from the policy function is good policy. In order to keep this property of AlphaZero across games, they were able to adjust accordingly.

2.5 Connect 4 Results

We have now shown that in theory an appropriate choice of α could significantly improve performance on a toy example, and that it was relatively simple to predict values of α that are close to optimal. This was useful because it removed some of the complexity of AlphaZero and allowed for more controlled experiments, but it does not prove definitively that α played an important role in AlphaZero. To show that these ideas transfer, we performed what amounts to a hyperparameter sweep on AlphaZero trained on Connect 4, and make the case that if performance benefits from parameter tuning on a simple game like Connect 4, it is highly likely from the results of the abstract game that α would be even more important in chess, Shogi and Go. To evaluate the performance of different parameter settings, we sampled several different values of α, trained across ten network initializations per α, and played the agents against each other for four games each in a round robin format. In these models, we explored 40 states in each MCTS, a significant change from the 800 states explored by AlphaZero in the paper, but not far off from the number of states explored in Atari games by MuZero (Silver et al., 2017b; Schrittwieser et al., 2019). The results are included in Figure 5. These two plots represent the same results, but have two different scales for α. On the y-axis, we plot win rate. The alternatives are a draw or loss, but we thought win rate would be most informative, as it represents true superiority over other models. As you can see in Figure 5, we found a roughly 150% improvement in win rate as compared to an effectively uniform baseline ($\alpha = 100$). We also found that the biggest advantage was only in a small window of α values, which

is consistent with the results of the abstract game. These results support the conclusion that even though the root is the only node that directly gets the Dirichlet bias in Monte Carlo Tree Search, the performance of the algorithm changes significantly with different choices of α.

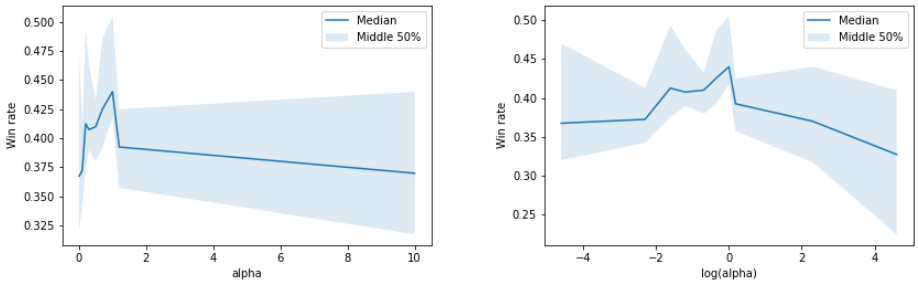

Figure 5: Win rates across different values of α

Beyond reinforcing ideas from the abstract game, the success of $\alpha = 1.0$ demonstrates that this parameter setting benefits from the intuition that a nonuniform exploration term helps AlphaZero find new states that will locally help the algorithm win a game, and at the same time increase training speed with proper exploration. An implementation of AlphaZero with the configuration used to produce these results is available online. ([Link removed for anonymization.])

These networks were not all trained to full convergence due to limits in computational resources. To ensure this did not significantly affect the results, two more tests were conducted. In the first, uninitialized networks were played against each other to ensure the effect of α did not start strong and then taper over time. The results showed no real correlation between α and winning rate, and in fact the highest performing α values were the smallest ones. This is likely because uninitialized policy and value functions perform like a uniform distribution in the toy problem, where there would be in this case benefit to exploring the tree more deeply. In addition to the uninitialized network, ten networks were trained to convergence, five for $\alpha = 0.7$ and five for $\alpha = 100$. These two settings repeated the format of playing all of the trials in the other setting for four games each. The $\alpha = 0.7$ agent beat the $\alpha = 100$ agent in a ratio of 3:1, suggesting the trend identified by the larger study continues and in fact potentially gets stronger as training continues.

2.6 Discussion

Through these experiments we have shown that the the choice of Dirichlet prior can lead to a much greater rate of success both in what we propose is a reasonable, if simple, model of hard games like Go and chess, and in an actual AlphaZero system playing Connect 4. Even if one lacked an intuition for the proper setting of α beforehand, the toy game provides a sense of ranges of α values that would be appropriate to elicit different search behaviors. Even by understanding that since α appears in the exponent of the dirichlet pdf brings intuition that searching on a logarithmic scale will be more informative than a more uniform search of possible α values. Experimentally we see at this scale results are more symmetric, and operating at this scale we understand differences in the magnitude of α as more natural transitions.

The influence of α in AlphaZero is more complicated than in the toy game. For one, the noise is only applied at the root, and the noise is only part of the exploration signal. We chose to add noise at each node of the toy game because while noise isn't directly added to non-root nodes, exploration choices are made partially based on the policy network, which is trained from the final exploration distribution of the root node. While this isn't an equivalent effect necessarily, choice of α still has a global impact on the tree search.

Our results also show that for games of high branching factor, AlphaZero becomes more sensitive to the choice of α. This highlights the benefit that intuition about its behavior can give, because for more complex games the number of good α values becomes quite small, and hyperparameter search becomes increasingly challenging.

3 CONCLUSION

AlphaZero is an impressive system, capable of learning superhuman strategy across several different games. We have analyzed how the choice of a hyperparameter α contributes to its strength as a system. Through the simple analysis of a synthetic problem, we demonstrate how to build intuition concerning the effect exploration noise parameterized by α has on the search process, and suggest how one can use that intuition to avoid an uninformed hyperparameter sweep. Training AlphaZero systems is computationally costly; therefore, avoiding massively parallel tuning while still being able to tune the system from problem to problem can result in significant time and dollar savings.

Future research directions are many. In this study, we investigate a way to find the effect of α choice on exploration of the toy game in expectation. A full model could allow programmers to increase their intuition and select an expected distribution over different states at each level of search that they believe is targeted to the branching factor and problem attributes they are looking at. This work highlights the benefits of a more fundamental understanding of hyperparameters in general, because the current inability to easily tune systems from problem to problem may cause us to lose out on significant performance improvements. In this vein, a natural extension of this research is to build intuition and theoretical understanding of some of the other hyperparameter choices in AlphaZero, so that more general hyperparameter sweeps can be avoided in favor of targeted tuning.

REFERENCES

Jiayu Lin. On The Dirichlet Distribution. Master's thesis, Queens University, Kingston, Ontario, Canada, 2016.

Hitoshi Matsubara, Hiroyuki Iida, Reijer Grimbergen, and Electrotechnical Laboratory. Chess, Shogi, Go, natural developments in game research. *ICCA*, 19:103–112, 12 1996. doi: 10.3233/ICG-1996-19208.

Tom M. Mitchell. The Need for Biases in Learning Generalizations. In *Rutgers University: CBM-TR-117*, 1980.

George D. Montañez. The Famine of Forte: Few Search Problems Greatly Favor Your Algorithm. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pp. 477–482. IEEE, 2017.

George D. Montañez, Jonathan Hayase, Julius Lauw, Dominique Macias, Akshay Trikha, and Julia Vendemiatti. The Futility of Bias-Free Learning and Search. In *32nd Australasian Joint Conference on Artificial Intelligence*, pp. 277–288. Springer, 2019.

Ilya Nemenman, Fariel Shafee, and William Bialek. Entropy and inference, revisited, 2001.

Christopher D Rosin. Multi-armed Bandits with Episode Context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230, 2011.

Jackson Ryan. Scientists create AI that can crush the world's best AI (at board games, thankfully), Dec 2018. URL `https://cnet.co/2Ge1wgW`.

Cullen Schaffer. A Conservation Law for Generalization Performance. In *Machine Learning Proceedings 1994*, pp. 259–265. Elsevier, 1994.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *arXiv preprint arXiv:1911.08265*, 2019.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR*, abs/1712.01815, 2017a. URL `http://arxiv.org/abs/1712.01815`.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *nature*, 550(7676):354–359, 2017b.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6404. URL https://science.sciencemag.org/content/362/6419/1140.

Hui Wang, Michael Emmerich, Mike Preuss, and Aske Plaat. Hyper-Parameter Sweep on AlphaZero General. *CoRR*, abs/1903.08129, 2019. URL http://arxiv.org/abs/1903.08129.

Ti-Rong Wu, Ting-Han Wei, and I-Chen Wu. Accelerating and Improving AlphaZero Using Population Based Training, 2020.