Dutse Journal of Pure and Applied Sciences (DUJOPAS) Vol. 3 No. 1 June 2017

Hardware Architectures and Neural Computing

A Review of the AlphaGo

.

Aminu A. Abdullahi Department of Computer Science
Federal University Dutse

Farouk Lawan Gambo Department of Computer Science
Federal University Dutse

Jibril Muhammad Adam Department of Computer Science
Federal University Dutse

Abstract

Advances in deep neural networks have opened up immense possibilities for Artificial Intelligence, the success of DeepMind in attaining professional competency in computationally taxing games like Go could result in networks stochastic networks that could challenge human superiority in other aspects. But these deep neural networks usually require immense computational resources in order to perform within satisfactory limits. This paper reviews the rudumentary architectures that have been developed and tailored to withstand the computational requirements of Deep Neural Networks like the Alpha Go and examines to which extent hardware conigurations affect the performances of these algorithms

Keywords- *Deep Neural Networks; Computer Architectures; Google; Deep Mind; Alpha Go*

INTRODUCTION

In October 2015 Alpha Go by Google's deep mind succeeded in defeating a human go player, this was followed by similar successes against professional players in 2016 (Schraudolph *et al*. 2016). these episodes were the first instances in which an artificial intelligence successfully confronted a human player in games with a high branching actor; such as go, which makes deterministic or brute force methods computationally unfeasible.

This paper attempts to look at the the vestigial computing architecture that made the tremendous achievements of the AlphaGo possible. The paper will attempt to compare the deferent renditions of the AlphaGo in terms of hardware configuration and examine to which extent these configurations affect the performance of the Alpha Go. It will also compare these effects to those already witnessed on other Go-based artificial neural networks.

GO AND DEEP NEURAL NETWORKS

Go

Go is an ancient board game with a relatively simple rule set but a very complex model (Cobb, 2002). According to Cobb (2002) a game of Go has more possible configurations than there are atoms in the observable universe.
According Silver *et al*. (2016), this complexity is a side effect of simple rule set of the game, these relatively simple rules allow for multiple configurations which in turn give the game a very high branching factor (Silver, 2016). This high branching action makes traditional AI methods like heuristic searches, Tree methodologies etc computational expensive and even unfeasible (Silver and Hassabis, 2016).

Deep and Convolutional Neural Networks

The Alpha Go uses a combination Monte Carlo Search tree methods and Deep Learning (Silver *et al*, 2016). Deep Learning involves layering multiple layers of artificial neural networks in combination with a learning regime in order to tailor the networks with an intended predictive capability (LeCun *et al*. 2015).

Deep Learning uses these multiple stack of neural layers to create better representations and in some cases, to formulate probabilistic representations of real life problems (LeCun, 2015). It is these probabilistic representations, coupled with the search capabilities of Monte Carlo Tree chains that form the intelligence behind the Alpha Go (Schraudolph *et al*. 2016).
Conventional Neural networks usually harness some sort of pre-processing in the form of feature extraction or dimensionality reduction (Hinton and Salakhutdinov, 2002), however for the Alpha go, very limited pre processing was employed (silver *et al*, 2016). This is also a feature common with deep convolutional neural netwroks where the black box convolutional stages are trusted to perform feature extraction and selection thereby reducing the computational load on the system (Schmidhuber, 2015).

ARCHITECTURE OF THE ALPHA GO

The first rendition of the AlphaGo to be deployed against professional Go opponents ran on a on a 48 CPU, 1 GPU machine. It had a total of 40 threads running computations along the different central and graphics processors (Silver *et al*. 2016). Figure 1 shows the relative performance of the AlphaGo on various non-distributed configurations. In such an architecture, Alpha Go would have to be implementing Task Level Parallelism in order to accommodate the multiple threads that run simultaneously (Patterson and Hennessy, 2013). The integration of Graphic processing Units also brings about the possibility of multicore processing, as according to Owens *et al*. (2008) most modern GPUs use multicore processing for more efficient computations.

For extremely complex processing tasks such as running deep neural networks, an interleaved parallel architecture would be more prudent, as such a design would remove the possibility of data dependency along the various processing pipelines (Owens *et al*, 2008).

Vector processing has also been employed in the hardware this could be MIMD (Marinheiro and Domingues, 2016) in the CPU to allow for the convolutional processing along the various layers of the Deep Neural Network and SIMD in the GPU which will allow for more efficient processing of the graphics-like data which serves as input for Alpha Go (Silver *et al*, 2016).

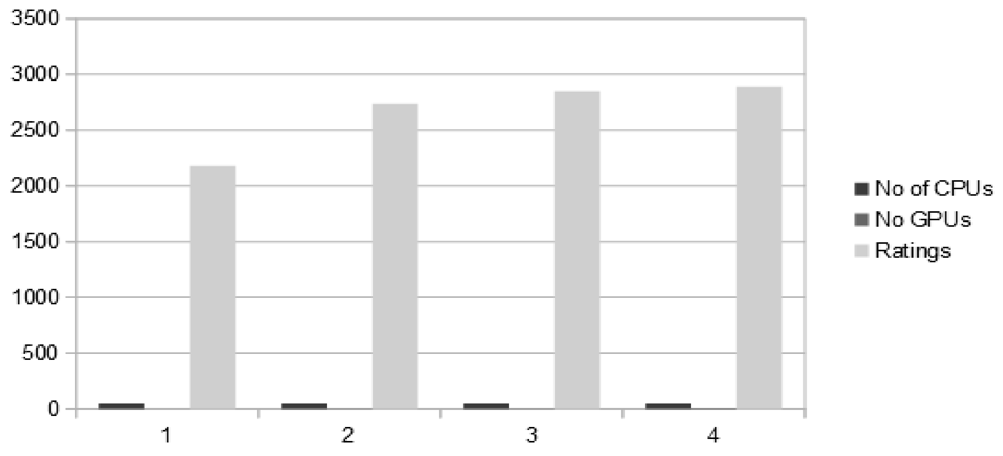

Figure 1. Improvements in performance ratings of Alpha Go on Different GPU configurations (48 CPUs) (Patterson and Hennessy, 2013)

When the number of GPUs were incresed, the Elo Ratings of AlphaGo steadily increased reaching 2,390 on 8GPUs (Silver *et al*, 2016).

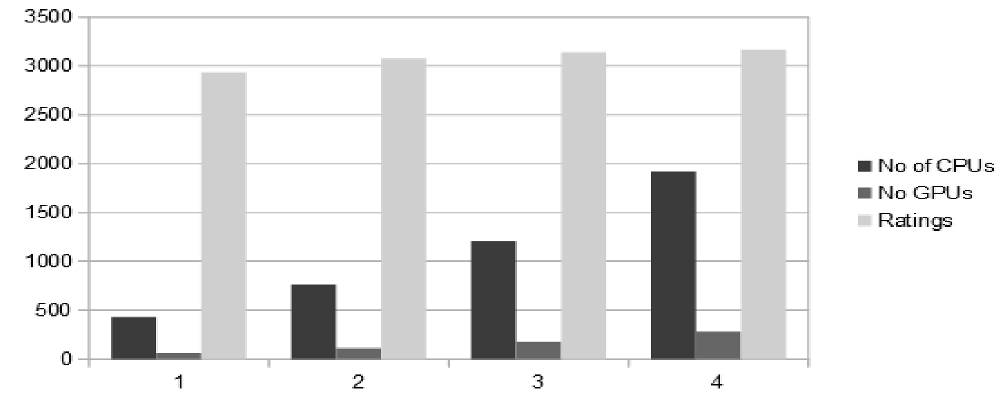

Figure 2: Performance of Distributed Alpha Go on Differnent CPU/ GPU configuration (Silver et al. 2016)

When the Alpha was deployed on distributed architectures, performance steadily increased as the no of CPUs and GPUs increased. As shown in igure 2, the rate of improvement gradually leveled out at

1202 CPUs and 176 GPUs respectively. According to Silver *et al* (2016), the intrinsic properties of the Monte Carlo tree search algorithm might be responsible.

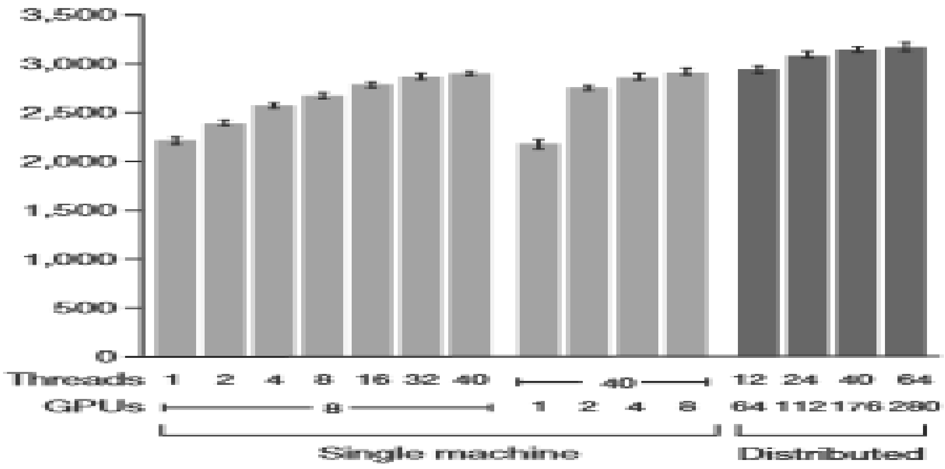

Figure 3: Performance of the Alpha Go on Different Configurations of Computing Hardware (Silver et al. 2016)

The Alpha Go and Other Go algorithms

A comparison of the performances and the architectiures o the the AlphaGo and other Go artificial intelligence algorithms, specifically Zen Go and Crazy Stone, also shows the effect of the hardware architecture on performance.

Figure 3 below shows the relative performance of Zen Go and Crazy Stone to AlphaGo. Crazy stone uses no GPUs and slightly less number of CPUs in a non distributed configuration. ZenGo by Yoji Ojo uses a scalable configuration but had achieved near similar performance levels with considerably more CPUs (Brudge, 2016).

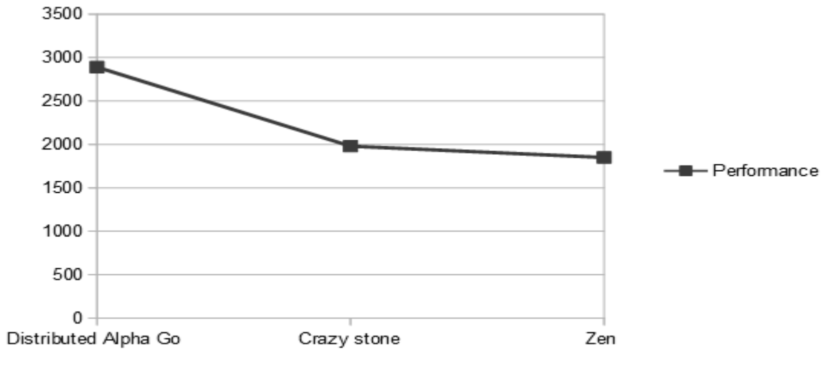

Figure 4: Performance of the AlphaGo compared to other Go AI Algorithms(Brudge, 2016) (Yoji, 2016) (Siver et al. 2016)

The results show that distributed Alpha go greatly out performs both ZenGo and Crazy Stone.

CONCLUSION AND FUTURE WORK

Results from the various researches show distribution and parallel computing is positively correlated with the performance of the AlphaGo. The steady increase in performance however evens out at 176 GPUs on distributed systems and 8 GPUs on non distributed systems. This diminishing in marginal performance might be a side effect of some structure in the AlphaGo algorithm.

It is however of immense implication to note that changes in hardware configuration alone where able to give the AlphaGo a near two fold increase in performance. This shows that the algorithmic efficiency alone might not be sufficient to give Artificial intelligence the near human performances researchers strive for. It would be interesting to examine the effect of hardware configuration on other types of neural networks and machine learning algorithms.

According to Jouppi (2016), DeepMind is deploying the Alpha Go on an Application Specific Integrated Circuit, the Tensor Processing Unit, which has proved more efficient in tensor flow calculations and the low precision probabilistic requirements of the AlphaGo (Jouppi, 2016). The explorations of TPU efficiency while running the AlphaGo might provide for interesting future works.

Also, with sufficient analysis of complexity of the Algorithm at its various neural states, one can extract sufficient information to allow for a quantitative comparison of the AlphaGo and other neural network based monte carlo search tree algorithms, this could theoretically unravel structural flaws in the the AlphaGo algorthim.

REFERENCES

Cobb, W.S. 2002, The book of GO, Sterling Publishing Company, Inc.

da Rocha Marinheiro, João Pedro Domingues 2016, "A Generic Agent Architecture for Cooperative Multi-Agent Games", .

Hinton, G.E. and Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. *science*, *313*(5786), pp.504-507.

LeCun, Y., Bengio, Y. & Hinton, G. 2015, "Deep learning", Nature, vol. 521, no. 7553, pp. 436-444.

Miles Brundage. 2016. *AlphaGo and AI Progress* . [ONLINE] Available at: http://www.milesbrundage.com/blog-posts/alphago-and-ai-progress. [Accessed 24 March 2017]

Owens, J.D., Houston, M., Luebke, D., Green, S., Stone, J.E. & Phillips, J.C. 2008, "GPU computing", Proceedings of the IEEE, vol. 96, no. 5, pp. 879-899.

Patterson, D.A. & Hennessy, J.L. 2013, Computer organization and design: the hardware/software interface, Newnes.

Schraudolph, N., Dayan, P. & Sejnowski, T. 2016, "Temporal Difference Learning of Position Evaluation in the Game of Go", .

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, *61*, pp.85-117.

Silver, D. & Hassabis, D. AlphaGo: Mastering the Ancient Game of Go with Machine Learning, Google Research Blog, 27 January 2016, .

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V. & Lanctot, M. 2016, "Mastering the game of Go with deep neural networks and tree search", Nature, vol. 529, no. 7587, pp. 484-489.

Tian, Y. and Zhu, Y., 2015. Better computer go player with neural network and long-term prediction. *arXiv preprint arXiv:1511.06410*.

Jouppi, N., 2016. Google supercharges machine learning tasks with TPU custom chip. *Google Blog, May*, *18*.

Yoji Ojima. 2016. *Zen Go Prgram*. [ONLINE] Available at: http://senseis.xmp.net/?ZenGoProgram. [Accessed 24 March 2017]