COMP6248 REPRODUCIBILITY CHALLENGE WEIGHT AGNOSTIC NEURAL NETWORKS

Andrei Carpiuc, Domas Navikas, Qi Zhang
University of Southampton

1 INTRODUCTION

This reproducibility challenge paper is made as a project (on GitHub) for module COMP6248. We aim to reproduce the paper chosen, perform additional experiments and discuss the results obtained.

We have chosen to replicate and analyze the Weight Agnostic Neural Networks paper first presented in NeurIPS Proceedings 2019 (Gaier & Ha, 2019). This paper focuses on an approach of finding artificial neural network solutions not by training the networks explicitly with gradient methods, but by rather encoding the solution in the topology of the network, thus making the network less dependable on the weight values themselves. Networks found in such a way are called Weight Agnostic Neural Networks (WANNs) described in more depth in the next section. This paper references a GitHub repository containing the code needed to replicate the results. However, to run all of our experiments we had to modify some parts of the code.

In the following sections we will first expand more on the topic of the paper we were trying to reproduce and give essential background information. Next, we will present the performed experiments, methodology, results and their significance. Finally, we will give a critical review of the paper, explain its achievements, problems, and end with a conclusion.

2 BACKGROUND

The aim outlined in the Weight Agnostic Neural Networks (Gaier & Ha, 2019) paper is to create a sparse neural network capable of performing a task on a satisfactory level with any random neural network weight parameter.

The intuition behind finding a network topology capable of encoding a certain behaviour comes from biology. Newborn animals usually are capable of performing certain actions without any training at all, which points to the fact that information is already encoded in their brains. The few examples mentioned in the paper are ducks (Starck & Ricklefs, 1998), which are able to swim on their own straight after hatching, as well as turkeys, lizards, and snakes (Goth, 2001; Miles et al., 1995; Burger, 1998), which have the ability to recognize and escape predators from the moment of birth. So it seems nature is capable of encoding certain behaviours without explicit teaching. The question arises should we be doing the same with neural networks.

There has been a lot of related work done, which is covered in the original paper in sufficient detail so we won't be expanding here. Most importantly the work in architecture search algorithms such as NEAT (Stanley & Miikkulainen, 2002) and network pruning approaches Zhou et al. (2019). The neural network search proposal in the WANN paper is to de-emphasize the weights as much as possible by evaluating the network on its performance over a range of weights from a uniform distribution.

The WANN topology search is as follows:

1. An initial population of minimal neural networks is created.
2. Each network is evaluated with different shared weight values multiple times producing a fitness score.
3. The best performing networks are selected for new population and mutated by inserting a node, changing an activation function in a node, or making a new connection.

3 EXPERIMENTS

In this section we will present the experiments done precisely as described in the paper (Base Experiments) to reproduce the results and an extra set of experiments (Additional Experiments) in order to gain a more in-depth understanding of the Weight Agnostic Neural Networks and accurately judge its achievements.

3.1 BASE EXPERIMENTS

We have successfully managed to replicate the results from Weight Agnostic Neural Networks paper. We have evolved the weight agnostic topology for the classic BipedalWalker, CartPoleSwingUp, and MNIST tasks. The experiments were performed using the code provided by the paper's authors, however we had to make some modifications to make it actually run. Fortunately, we had the hyperparameters of the algorithm such as population size, elitism, mutation probabilities, etc. already preset for all of the runs. They should have mentioned in the paper how much effort it took to set them right. Based on our experience with defining hyperparameters ourselves for the additional experiments and the great variance between hyperparameters provided for different environments, we expect the hyperparameter tuning to be a quite significant topic worth mentioning. An example of reproduced fitness graph for CartPole can be seen in Figure 1.

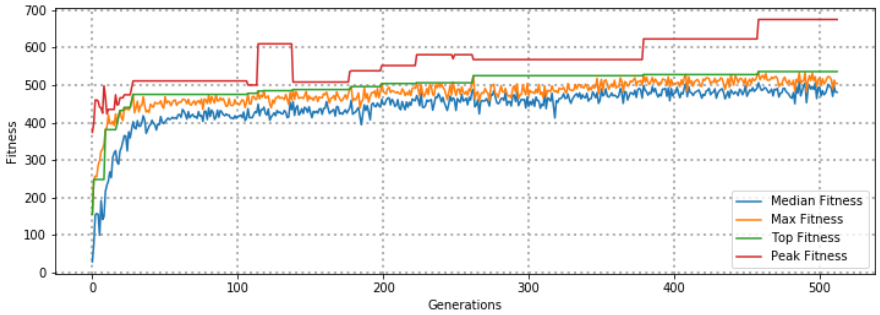
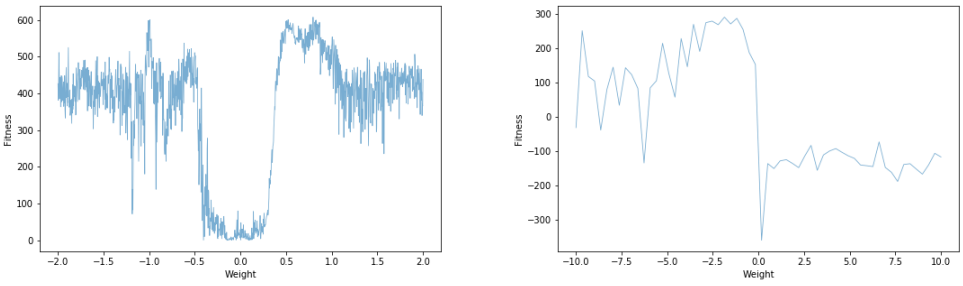


Figure 1: Fitness over generations for CartPoleSwingUp.

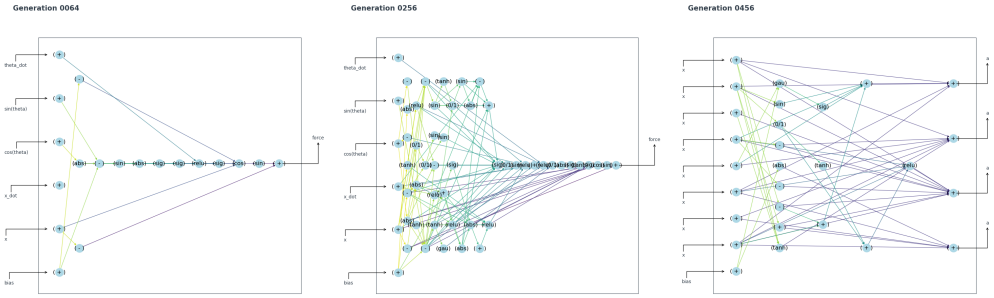
An important observation made during training was that the algorithm does not scale well at all. It took a total of 23 hours to evolve a bit over 500 generations on CartPole. The biggest problem was an increasing amount of time required for each new generation. To perform one population update (make a new generation) it took 0.75, 1.625, 2.25, 5.75 minutes at generation 0, 100, 200, and 500 respectively.



(a) CartPole (b) LunarLander

Figure 2: Performance of the network with a range of uniformly sampled weight values.

Another observation worth noting is that the network is sometimes not fully weight agnostic. Testing the performance over 1200 uniformly sampled weight values uncovers that the network is still somewhat sensitive to the weights, in some cases breaking the performance completely even outside the expected small weight range. See (a) graph in Figure 2 at weight value -1.25.



(a) SwingUp@64 (b) SwingUp@256 (c) LunarLander@456

Figure 3: Architecture snapshots of networks at certain generations.

In general, from the (a) & (b) graphs in Figure 3 we can see that the network almost learned to ignore the inputs completely by repetitive squishing of the input. Of course that makes it not care about the weights that much, but what's the benefit? Now it makes an internal weight representation, where explicit weight values have little effect. This could be used to avoid catastrophic forgetting for multi-task problems, but the original paper does not provide any evidence that the network is capable of adapting to any similar domain. Such an experiment would have improved the value of the paper significantly.

3.2 Additional Experiments

3.2.1 LunarLander-v2

We have additionally evolved a network for a LunarLander task. This experiment uncovered quite a lot of information about the paper. First, the hyperparameter tuning is a real serious problem and takes some time, especially considering that the algorithm is very slow itself. Second, the network is not always fully weight agnostic, for example our LunarLander network worked well with all the negative values, even outside the range used for searching, but failed miserably with all of the positive values (see (b) in Figure 2). On the flip side, the architecture looked quite neat (see (c) in Figure 3). Maybe the network stopped caring about being weight agnostic and focused on the task instead. It could be that with more time it would learn to deal with positive values just as well, but what would be the benefit of that, why use this instead of a classic evolutionary approach? The benefits are not well justified in the original paper.

3.2.2 MNIST

Given the Kolmogorov complexity constraint of the optimisation function of the algorithm, we hypothesised that a WANN, once evolved, should have less variance in its performance on different tasks in the same domain compared to classical neural networks. In essence, because of the reduced search space, the description length constraint and the reduced complexity caused by a single shared weight, those networks can't overfit, thus having less variance at the cost of more bias.

To test this hypothesis, we decided to look at the out of sample classification problem for deep neural nets. Deep neural networks do not cope well with images coming from an out of sample distribution, assigning them to a class with a high degree of confidence (Nguyen et al., 2014).

For our experiment, we started from the normal MNIST data set. Next we decide that the class representing the digit '8' will instead represent unknown images. Ideally, the model would to put any image into this class that is not an image of the other 9 remaining digits. To reinforce this idea, we added 6000 new images (one-tenth of the size of the original data set) of random noise with the label '8' to the data set. For each image, each pixel was sampled from the standard uniform distribution.

For testing, we used two data sets. The first is the MNIST test set and the second is a modified version of the Fashion-MNIST train data, where the label of each image is modified to be '8' (unknown).

Next, we compared the WANN generated after 512 search iterations to a gradient trained CNN with the following architecture:

- 16 channels 5x5 convolution layer with RELU activation
- 8 channels 3x3 convolution layer with RELU activation
- 32 by 10 linear output layer

The CNN achieved an accuracy of 88% on the MNIST set and 71% on the Fashion-MNIST set. The performance of the WANN can be found in Table 1.

Table 1: WANN performance on MNIST and Fashion-MNIST datasets.

Shared Weight	**-2.0**	**-1.33**	**-0.67**	**0.0**	**0.67**	**1.33**	**2.0**
Accuracy (MNIST)	82%	81%	76%	10%	74%	84%	84%
Accuracy (Fashion-MNIST)	75%	76%	75%	0%	65%	74%	74%

4 DISCUSSION

From a purely reproducibility standpoint, the results presented in this paper can be replicated accurately, in a different environment, using different machines, by different researchers. The theoretical overview of the algorithm is clear and straight forward and, coupled with the provided source code, allows for different implementations and extensions of the base work to be carried out. Furthermore, variations to the original experiments performed in the paper, achieve results in accordance the claims of the paper, suggesting that relevant, task specific information can indeed be encoded in the topology of the network, for an arbitrary use case.

However, the paper and the available code are not perfect. Even though the authors provided an implementation of their algorithm (which is more than what most publications do), the documentation was outdated and lacking in explanation, while the code scales poorly and needed debugging in order to run on our machines. Regarding the theoretical contents of the paper, we believe the authors do not fully justify the benefits of this approach. In terms of results accuracy, even the authors note that classical, state of the art methods, outperform WANNs in the given domains. This, coupled with the relative time increase of searching the space, compared to gradient learning, means that WANNs, in their current form, are unsuitable for any practical use cases. The authors have claimed that a weight agnostic network, once evolved, might be able to perform satisfactory on a wider range of tasks, as more general information should be encoded in the structure. Our own observations and experiments seems point to the same conclusion (that WANNs generalise better to different tasks in the same sub-domain), however there is no given evidence to suggest that classical, gradient learning based architectures and methodologies, can not achieve comparable results.

Lastly, the ”weight agnostic” claim can be disputed, as, even though there is no explicit weight training, the performance of the network varies drastically for different values of the shared weight parameter. This effect is not understood and currently there is no theoretical evidence for choosing one value over another, with the only deciding factor being empirical data.

5 CONCLUSION

This paper presents its findings and methodology in a clear manner and, with the addition of source code, the experiments are ”relatively” easy to replicate across different environments. Even though there are critiques that can be leveraged against the usability and usefulness of the algorithm, as well as against aspects of the methodology and breath of the experiments performed, those only highlight a lack of research regarding the topic of exotic network topology and a hyper focus of the literature on gradient based weight optimisation. In conclusion, the Weight Agnostic Neural Networks paper is well produced and serves as a welcomed step in the study of more dynamic adaptable machine learning system.

REFERENCES

Joanna Burger. Antipredator behaviour of hatchling snakes: effects of incubation temperature and simulated predators. *Animal Behaviour*, 56(3):547–553, 1998.

Adam Gaier and David Ha. Weight agnostic neural networks. *CoRR*, abs/1906.04358, 2019. URL http://arxiv.org/abs/1906.04358.

A Goth. Innate predator-recognition in australian brush-turkey (alectura lathami, megapodiidae) hatchlings. *Behaviour*, 138(1):117, 2001.

Donald B Miles, Lee A Fitzgerald, and Howard L Snell. Morphological correlates of locomotor performance in hatchling amblyrhynchus cristatus. *Oecologia*, 103(2):261–264, 1995.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014. URL http://arxiv.org/abs/1412.1897.

Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

J Matthias Starck and Robert E Ricklefs. Patterns of development: the altricial-precocial spectrum. *Oxford Ornithology Series*, 8:3–30, 1998.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, pp. 3592–3602, 2019.