NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING: ANALYSIS

In this paper, authors use a recurrent network to generate the model descriptions of neural networks and train this RNN with reinforcement learning to maximize the accuracy of the generated architectures on a validation set.

This paper presents Neural Architecture Search, a gradient-based method for finding good architectures. The work is based on the observation that the structure and connectivity of a neural network can be typically specified by a variable-length string.

Proposed Approach:

1. Generate model descriptions with a controller recurrent neural network
2. Training with reinforce
3. Increase architecture complexity with skip connections and others layer types
4. Generate recurrent cell architecture

The methodology followed was to generate the layer hyperparameters (for example, a convolutional layer would require stride, kernel size, number of kernels, whether it has a skip connection, etc.) sequentially. These hyperparameters are what we call a search space. The generated architectures were trained on the CIFAR-10 dataset for a certain number of epochs. The reward function used was the maximum validation accuracy found in the last 5 epochs cubed, and the controller was updated using the reinforce policy gradient.

Experiment and Results

A total of 12800 child models were trained over 21–28 days of GPU time , the best architectures that this controller found were a DenseNet-BC (L=100 , K=40) with 190 layers depth, and 25.6M trainable parameters, this model achieved an error percentage of 3.46% which was state-of-art model on CIFAR-10. Another model would be considered regarding its lighter depth was NAS v3 max pooling and more filters with 39-layer depth, 37.4M trainable parameters and 3.65% error percentage.

Likes:

1.The ambiguity on the process to design the best NNs was resolved.

2.The Related Work section in the paper is insightful, the authors not only mention the related works done but also talk about the success and limitations of those works and why their approach was better than the methods used in related work.

3. I read that 800 GPUs were used for 28 days and I started exploring for improvements in this, I read more about efficient neural architecture search(enas) and search space strategies macro and micro.

4. There were a lot of things I understood from this paper: how the search space is designed, how the search strategy is designed and different ways to speed up architecture performance estimation.

Dislikes:

The only drawback of their methods was the time it took for the search space before coming up with a definite solution. They used 800 GPUs for 28 days to go through the entire search space before coming up with the best architecture. This was a computational complexity and a way to design controllers that could navigate the search space more intelligently was needed. They mention the ways to overcome these in the paper.

Inspirations:

Deep learning experts are always expected to know about what architecture might work best. One can implement endless number of architectures. The neural architecture search aim to automate finding the best model architecture, given a dataset.

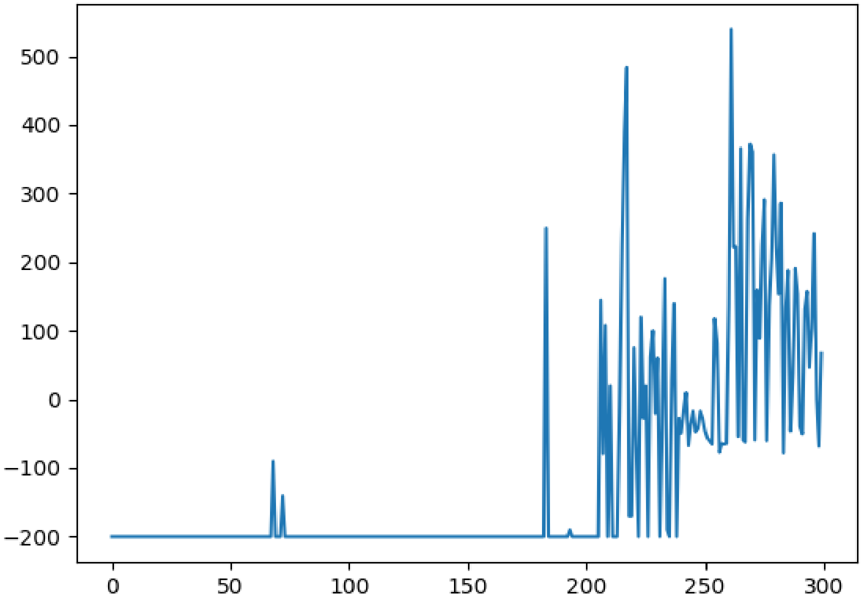
Report:

On increasing the number of layers, the car learns faster to reach the flag and reaches the flag more often.

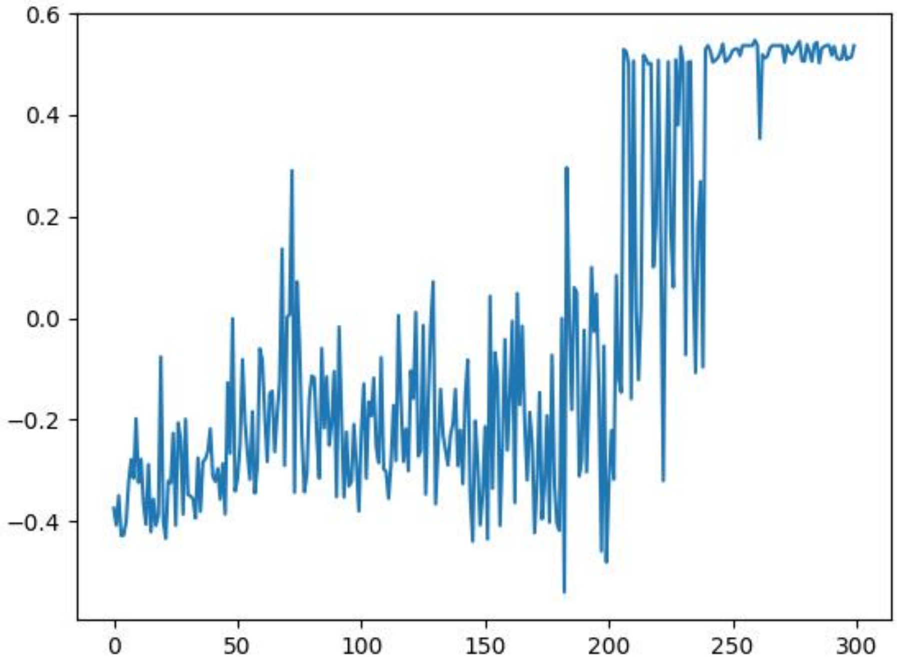
2 Layer: (50x50)

Total award: -36102.0

Average x-position: -0.020113933301224934



Layer 2 award

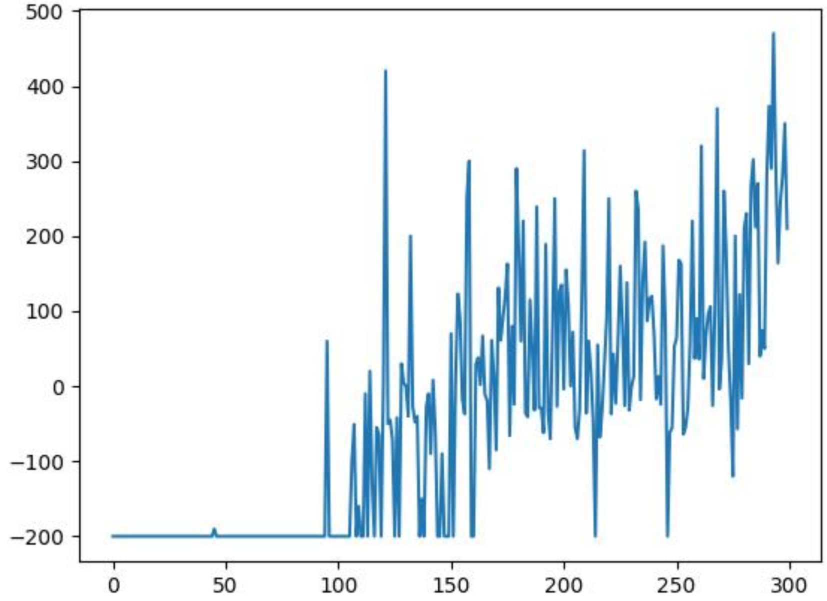


Layer 2 x- position

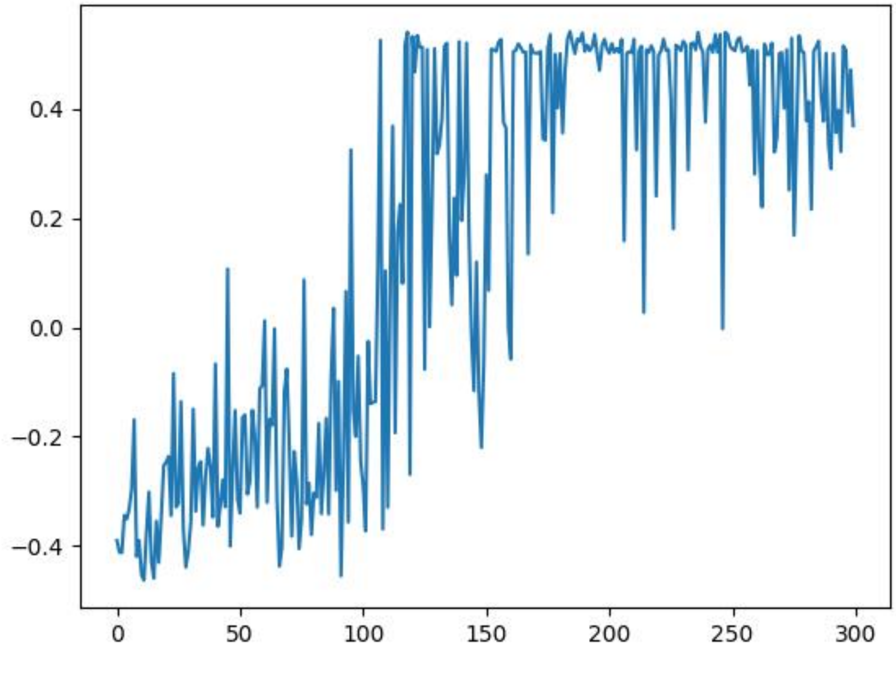
3 Layer: (50x50x65)

Total award: -13012.0

Average x-position: 0.16360094901965155



Layer 3 award

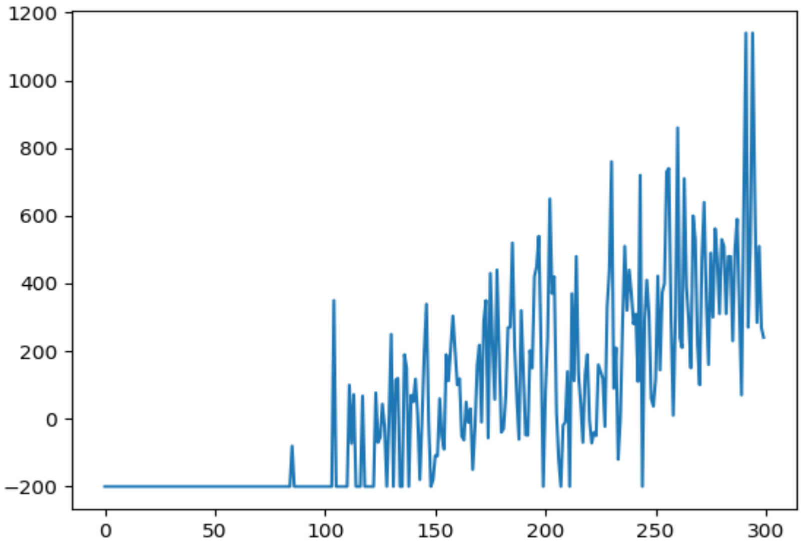


Layer 3 x position

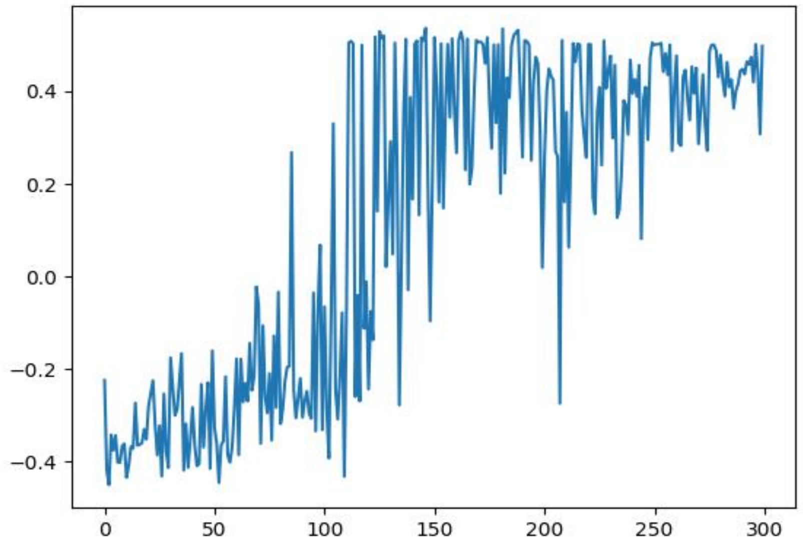
4 Layer: (50x50x70x90)

Total award: 13708.0

Average x-position: 0.12296246465138845



Layer 4 award

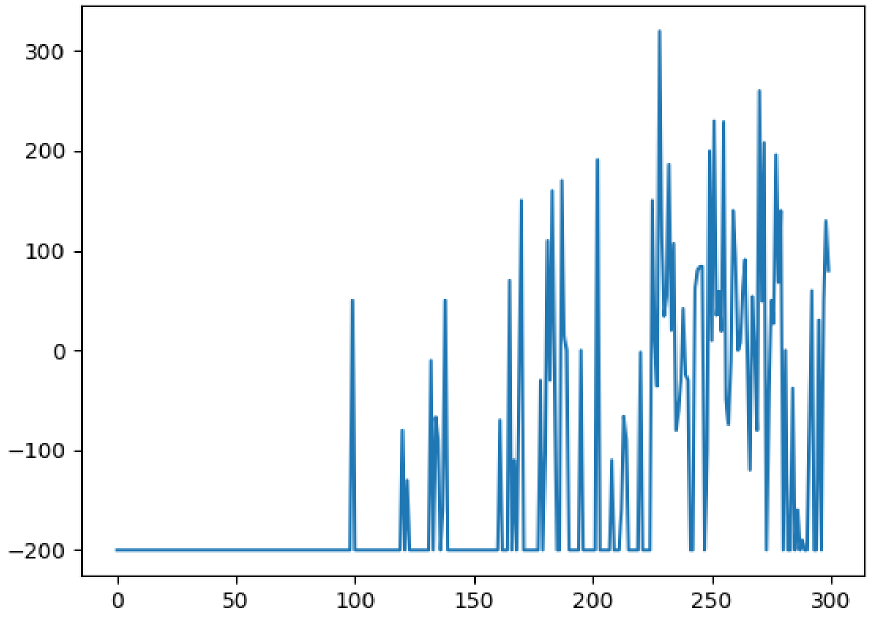


Layer 4 x position

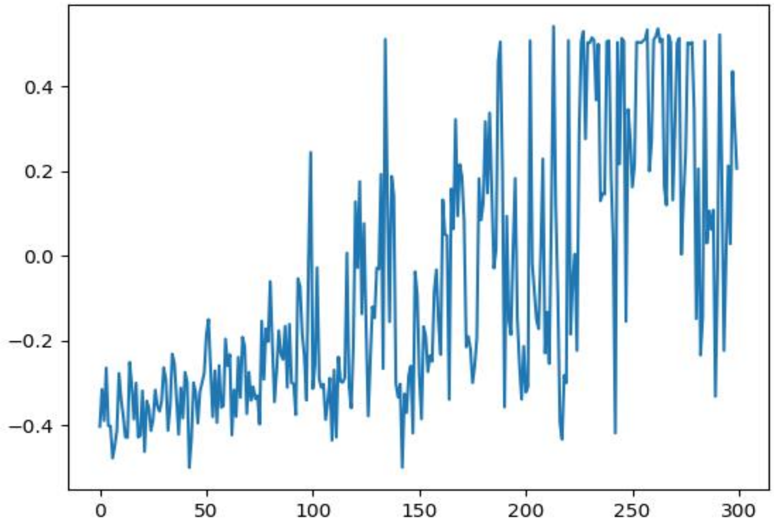
5 Layer: (50x50x70x85x70)

Total award: -39673.0

Average x-position: -0.07034538744267457



Layer 5 award



Layer 5 x position

I got the highest total award for 4 layer NN this surprised me a bit as it is odd one amongst all the other awards. I ran this again and I got the total award – 8331 this is still the highest but I did not understand what happened.

The greatest average x position is for 2 Layer NN.