

Analysis of “Neural Architecture Search With Reinforcement Learning”

Preamble

I am really sorry, but this may turn out to be a short paper. After reading and rereading the paper, there are only a few thoughts that I am able to come away with.

1. Designing Architectures requires a lot of experience and a large amount of time.
2. Even if you do everything right, and tune your model with all the best numbers, it is unlikely that you are going to achieve anything that is transcendent or awe inspiring. Best case scenario your model or idea becomes the new “best” by small percentage improvement of the current “best”. Worst case scenario (and far more likely) your model is worse than the current best.
3. It is not possible to stay abreast of all the current knowledge and best practices for model architectures. There is so much happening so fast, and good luck just trying to separate the good from the bad.

A brief statement of the problems addressed in the paper in my own words.

Neural Architecture Search(NAS) is the solution to designing, building and validation models. With the work that they have done, and with supposed future work, anyone could walk up to it and dictate their needs, and provide an accompanying dataset and NAS will spit out a model to suit your needs.

What I agree with/like in the paper and why.

Again, like a previous paper, they appear to have gone to great lengths to ensure that they did not leave any performance on the table. It is impressive to see how many parameters, ideas, and philosophies they were able to cram into NAS

What I disagree with/dislike in the paper and why.

I don't disagree with anything in the paper per se. I just ended up walking away with a feeling of despair that it is not possible to remember “all the things” to do when designing and training a model. Granted, they did have a huge cpu and gpu farm just sitting around, and maybe they thought “hmmm, instead of letting this farm collect dust, let's put it to use!”

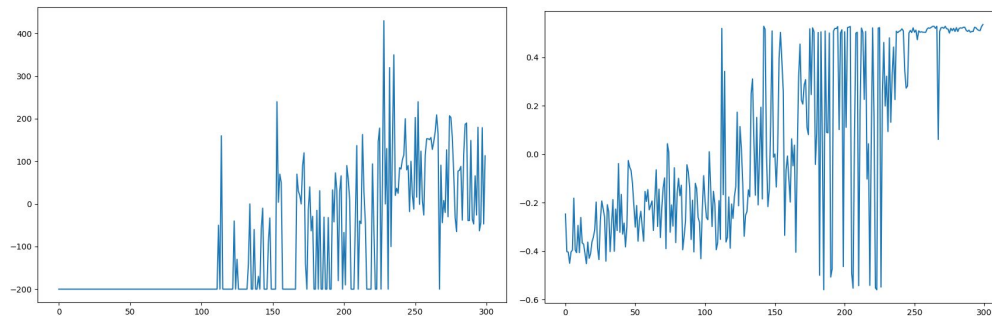
Any inspirations I found in the paper.

Yes, I have some inspirations. Don't go into designing and architecting network models. Leave it to be automated by a machine! There just seems to be too much that has to be tweaked just right, and buttoned up just so that you can achieve anything that is really meaningful. This was a good paper, it was just hard to read because I didn't walk away with thoughts of “Yeah, I can do this.” or “Wow that is something to keep in mind for the next time.”, it was mainly a huge eye-opener. There is no way I can stay abreast or even afloat of all the “best practices”, I can see now why some people just “force” their models to work, regardless of if it could or should be done better.

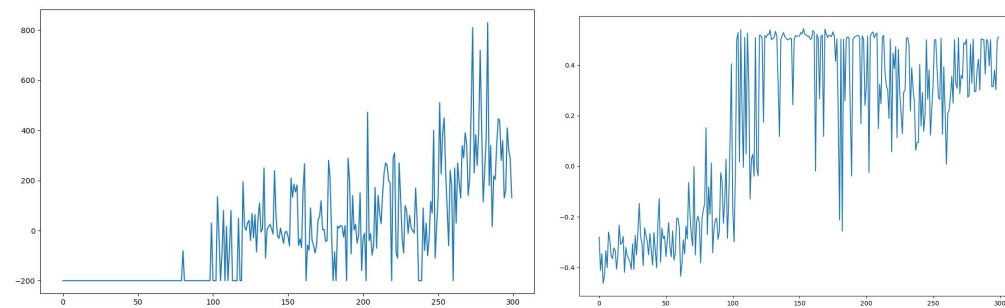
Thanks again for the paper! It was eye opening, and helped me to see that this is not really a thing that you can just practice on the side, if you are going to get into it you need to dive in deep.

Train each model for 300 episodes, generate the reward and max x-position plots for each model, and include your plots on the 2nd page of your paper analysis for Problem 2. Briefly state which NN gave you the highest total award and the greatest average x postion.

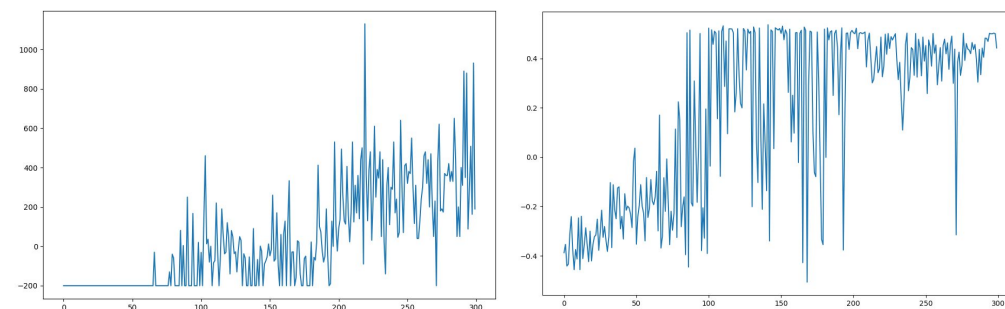
2x50x50x3



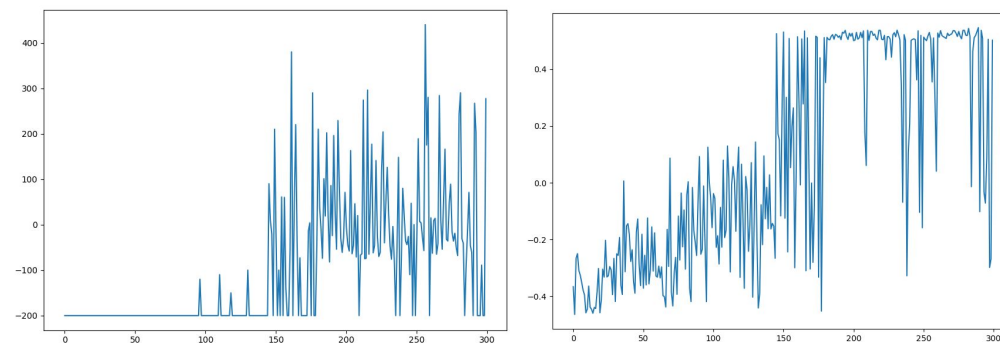
2x100x100x100x3



2x25x25x25x25x3



2x12x12x12x12x12x3



I would have to say that my 4-layer model (2x25x25x25x25x3) gave the highest total award(about 1100) and it also gave the greatest average x position(about .2).