Name: Aaloke Mozumdar **Roll No: 2019004**

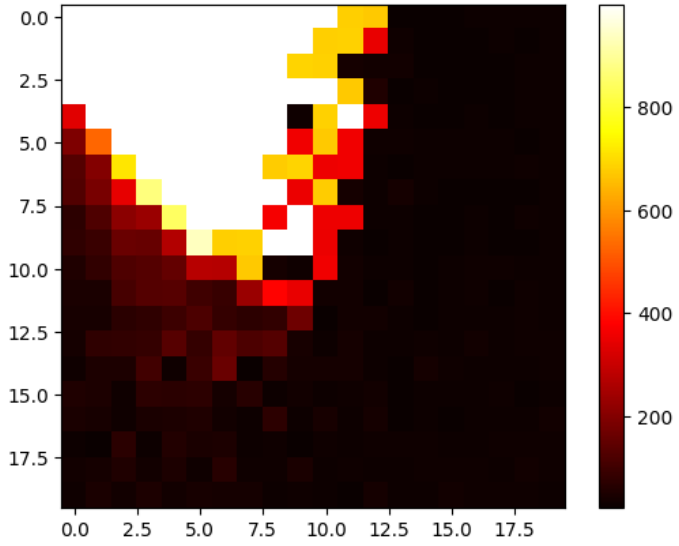
Meta Learning Project

Objective : We intend to train a series of online learning algorithms **(Forward-Forward and GLN)** to perform well on a **Meta-RL cartpole environment** (cartpole with changing lengths and gravities), by doing **rapid behavioural cloning** on an expert. During meta testing time, we hope that our model can **adapt** in as little as **1 episode.**

Methodology : We train an expert PPO policy to do cartpole over various environments. We simultaneously train our online learning algorithm by doing behavioural cloning on the rollouts of the expert. While training we use several episodes long rollouts (5000-10000 timesteps) to clone our supervised learning algorithms.

Finally during test time, we iterate through other unseen environments. We generate a single episode rollout of the expert on the environment. We then rapidly adapt our supervised learning algorithm on this single episode data.

Baseline : Our baseline is a PPO policy expert that is trained on a single environment, and is then tested on all the remaining environments.



Results :

During training time, FF is able to reach a reward closer to the expert as compared to GLN. Which means that FF, when trained over a large number of timesteps (During training ~5000), is able to clone better as compared to GLN.

However during testing time, GLN adapts much more rapidly as compared to FF. FF is unable to adapt with a single episode and thus we mostly get a very low reward for most environments. GLN, even though it adapts faster, is quite often inaccurate. This might be due to the small network size, which is not allowing the network to store much of the past information.

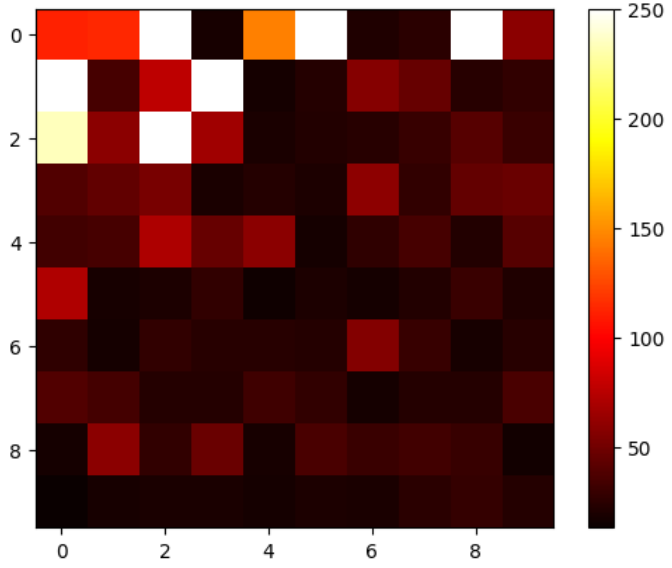


Fig: FF_perf matrix

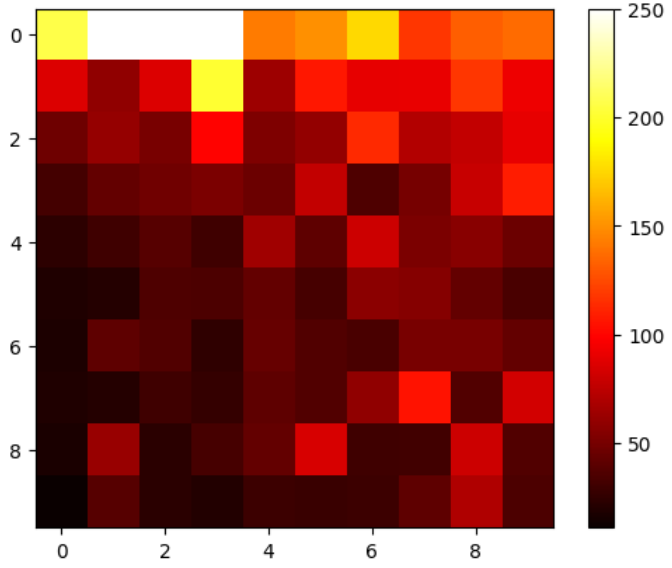


Fig: GLN_perf matrix

Drawbacks :
Since we are performing behavioural cloning on an expert, the overall performance of our model is dependent on the performance of the expert. Therefore, in environments in which the expert is poor, our model also gives poor results.

In the forward forward algorithm paper, Hinton talks about **"Learning Fast and Slow"** where he indicates that with a custom learning rate for a given sample, each layer of the FF network can achieve a required goodness value in just a single epoch. This, although highly overfits the data, is crucial in making the FF network adapt rapidly to new data, enabling it to learn a lot of information from a single sample.

I have tried to implement a version of this **(in FF_fastslow.ipynb)**. In my implementation, even though the goodness rapidly approaches the required goodness in just a few epochs, it stagnates somewhere in between. Further, the negative sample is unable to go to the required goodness, since the network is heavily biassed towards the high goodness of the positive sample. Further, for some samples, the network highly overfits, and thereafter it only predicts the label corresponding to that sample, for all other samples. This might be caused due to implementation error, or a misunderstanding on my part to the implication of the paper in this section.

Future Work :
As discussed in the Drawbacks, **FF_fastslow** does not give the expected results. If it works properly, we can use this for adaptation on the single episode data of the expert. Hopefully, this should make the network adapt more rapidly.

Further, we may try alternate architectures where we train an FF network as an outer network which is then connected to an inner slow learning RL policy network. The outer network adapts fast to changing environment, while the inner policy learns slowly over various environments.