**Status Update – Week 5**

**Research Question**: Under what assumptions can we use a combination of global and local rewards in order to speed up training whilst ensuring cooperation? Specifically, can this be done under locality assumptions?

For example, in the multi cart pole scenario, it is obvious that the n’th cartpole has no effect on the first cartpole, and therefore it should not learn from it. This is clear from observation of the local rewards.

**Middle Ground:**

In the last update, I talked about the 2 extremes of losses that we can define:

Global Loss (Correct and Cooperative, Slow Learning):

Local Loss (Non-cooperative, Fast Learning):

And I talked about trying to learn from both. Note that these 2 are extremes on a scale between global correctness and learning speed.

I define a middle ground, that similar references to it can be found in the Locality Paper, called :

* – current agent we’re looking at
* – the size of the neighborhood of individual we’re looking at

Note that for , we get exactly , and for for sure we will get . So the parameter gives us a more correct middle ground between the 2. Notice that for any that isn’t the entire network we get n individual losses but for an which is the entire network we get n losses which are exactly the same so we can just add them up into one single loss.

Notice how the speed of convergence rises as shrinks – for small ’s we can see that credit assignment is much easier since the sum of rewards is smaller.

Also notice the distinction between the 2 types of parameters – and . is the parameter from the locality paper which defines the individual truncated , and tells us how many of them we should add, and only affects the loss, not the network architecture. **I have a feeling there is some more theoretical work to be done around – this will occupy me for the next few weeks.**

**Coding:**

I have started coding the following scheme with slowly increasing . There are a lot of ideas regarding how to do this incrementally. This is also related in some way to the world of curriculum learning and warm starts. We start off with easy examples, since we only expect the network to learn individual effects. Only later on as we increase do we expect the network to learn the interactions from the other agents.

* We can have increasing in additive or multiplicative steps. The exponentiality of the multiplication is somewhat interesting since we do expect to have some decaying factor from the locality of the problem. We call this .
* We can do both hard and soft updates of the loss as a function of . Soft updated means we slowly increase from 0 to 1:
* We can also consider not a strict curriculum, but a gradual one of both hard and soft examples.

**Bounded Cooperation:**

My guess is that after some , we will stop seeing improvement and reach the optimal solution for a lot of problems. These problems are problems where cooperation is somewhat bounded and there is no need for larger . This are very interesting in my opinion, and relate closely to the theoretical work I wish to do regarding the meaning of that I mentioned earlier.

**Stairs of :**

I would really like to see the effect of increased cooperation as a function of . I plan on trying to show this by doing a hard update after long intervals, and plotting the x timestep of each ard update. I hypothesize that we will see stairs for each new larger value of l. These should be compared to just starting off without gradually increasing l, as a proof of the effectiveness of curriculum learning.

I also need to start checking the other directions I has for the start of the mixing layer – GCN with positive weights? Regular convolution with gaussian kernel?

Maybe also try to find the connection between k and

The concept of **Curriculum Learning for RL is SUPER relevant: https://arxiv.org/pdf/2003.04960.pdf**