Peer Review of CA 3 – Assignment by Group 2, Review by Group 4 (Henrik)

Intro

Apologies for the tardiness in writing this peer review! We completely missed that the peer reviews were supposed to be written continuously as the course progressed until we got the email for CA7. Anyways, we are writing all our peer reviews now, so expect to receive a bunch.

Backpropagation Implementation

Cool to see that you did a full backprop implementation! I did the same and found it extremely time consuming, but still cool to learn how the backend of DNN training works.

GD vs PGD

Is N_train = 1000 here like for SGD? I assume so.

"The GD consistently gets higher final cost than SGD. Does it get stuck in local minima?"

I have two comments on this statement:

- Maybe it's getting stuck on a saddle point? I believe the main point of introducing PGD is to avoid converging to saddle points.
- It's not getting a significantly higher cost than SGD, you got 0.86 for GD and 0.84 for SGD. Maybe you meant higher than PGD?

My guess would be that GD consistently gets stuck in saddle points. I remember hearing in a lecture that for deep neural networks there are way more saddle points than there are local minimas, so chances are that's where you will end up. Then PGD is better because it manages to find a minimum.

SGD

As you mention, the final cost of using mini-batch SGD and pure SGD is comparable. It could be that this is happening because you're covering the same number of datasamples (5000 in both cases, as you mention). But just as you seem to ponder yourself, I'm not sure about this. I haven't read anything about this comparison. However, for sure I think we can say that it would be a bigger difference if you covered a larger proportion of the dataset with one algorithm over the other.

Interesting to see that the cost is consistently higher than PGD. Maybe this is because you ran 20 epochs with PGD, but only 5 epochs with SGD. (1 epoch being one run through the entire dataset)

SVRG

I'm a bit confused about the numbers here. You have 500 iterations and 100 samples per minibatch. So you consider 50 000 datasamples in one epoch? I thought it was only 1 000 samples per epoch. Maybe there's something I'm misunderstanding here.

Summary

The results look pretty good and all solvers are working! However, I saw very little discussion and text in general so sometimes it's a bit hard to understand exactly what you're doing. (How many datasamples for each algorithm, how many epochs, selection of step size, etc…) Either way, the backprop implementation was tough, so I can understand if you simply ran out of time.