Project 4 for PSYC 489

Bingjun Guo (bingjun3)

March 28, 2024

Part 1

Epochs needed for the nets to succeed on classification are all plotted in this part. The criterion for all sets is to classify all samples in each set correctly. The maximum number of epochs is set to be 500. Net2 is in shape 10-6-4. All nets' parameters are randomized initially.

Set 1: Non-Overlapping Categories

Both networks succeed in classifying, since this set is non-overlapping, which indicates perpendicular, and thus makes this set specifically easy to classify. It is found that learning rate and momentum are "the larger, the better" for net1. For net2, the best learning rate lies around 2, and the best momentum lies between 0 and 0.3. Under the best settings, it takes only 1 epoch for net1 and 3 or 4 epochs for net2 to reach the criterion.
Notes that for Part 1, all experiments on momentum are run under the best learning rate without momentum, with is 8 for net1 and 2.2 for net2. Moreover, momentum are limited between 0 and 1 to avoid explosion. For this set, parameters in net1 actually never explode.

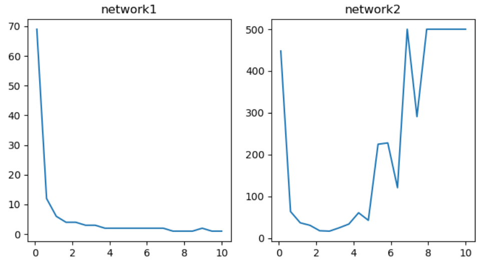

Figure 1: epochs v.s. learning rate (set 1)

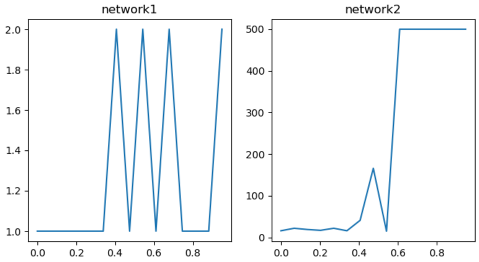

Figure 2: epochs v.s. momentum (set 1)

Both nets perfectly classifies the original training patterns. With noises, as is plotted below, the error rate for both nets will increase as the noise level increases. It can be indicated that net2 has stronger generalization capacity.

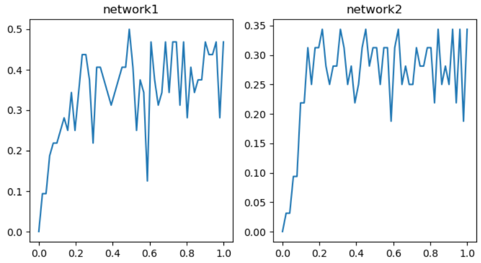

Figure 3: error rate v.s. noise level (set 1)

Set 2: Linearly Separable Categories

It seems that net1, which is simpler, fails to classify this set with larger learning rates, while net2 seems to occasionally fail on learning rates less than 20, and constantly fail on learning rates larger than 25. These can be caused by the dataset being more complex. Yet it is still linearly separable, it would be impossible to classify based on merely basic hyperplanes that are perpendicular to axis. For both nets, large momentum will lead to failure, and best momentum under the best learning rate is rather small for both nets. Under the best settings, it takes 1 or 2 epochs for net1 and 18 to 25 epochs for net2 to reach the criterion.

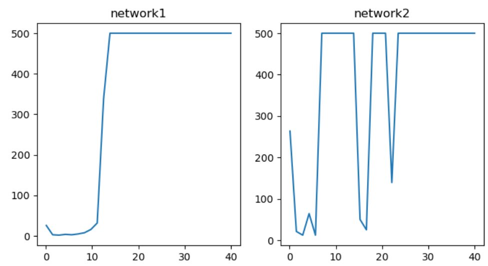

Figure 4: epochs v.s. learning rate (set 2)

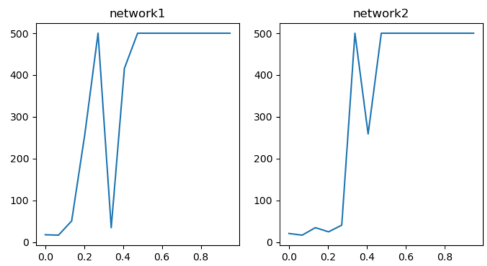

Figure 5: epochs v.s. momentum (set 2)

Still, both nets perfectly classifies the original training patterns. With noises, still, the error rate increases as noise level increases for both nets. However, the generalization capacities for both nets seem the same now.

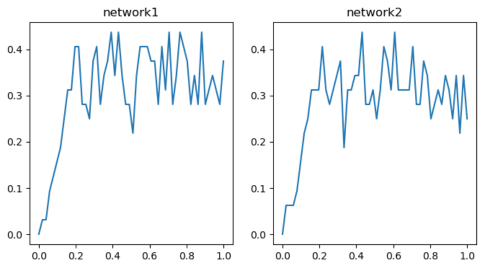

Figure 6: error rate v.s. noise level (set 2)

Set 3: Non-Linearly Separable Categories

Net1 fails to classify this set under any conditions, since single layer nets can only classify linearly separable data. Net2, as a two-layer net, succeed when learning rate is approximately between 0.5 and 19, under most of the momentum settings, taking ≈ 50 epochs.

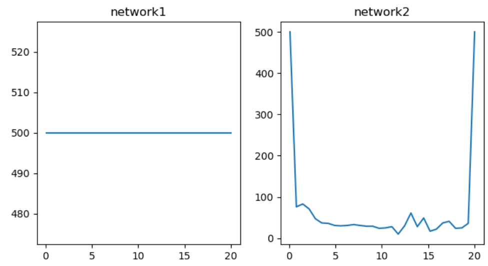

Figure 7: epochs v.s. learning rate (set 3)

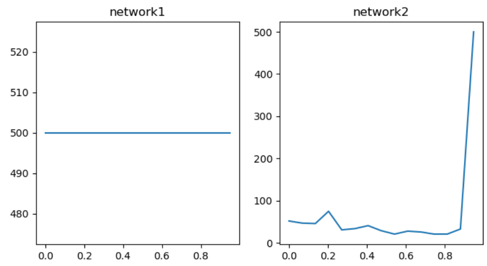

Figure 8: epochs v.s. momentum (set 3)

Net1 fails on classifying this set, but its overall error rate somehow seems the same as that of net2 as noise gets larger. Moreover, the generalization capacity of net2 seems improved for a bit.

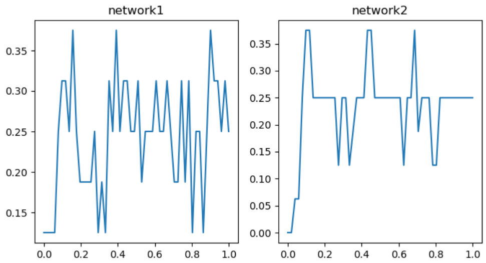

Figure 9: error rate v.s. noise level (set 3)

Part 2

Part 2A

Yes. It seems the training is fastest when learning rate is set to 6, which takes 7 to 15 epochs.

Part 2B

The network gives [0, 0, 0, 0, 0, 0, 0, 0]. No. The network is still not able to generalize on this dataset, which can due to the limited training data. The network doesn't know the rule yet.

Part 2C

Still with learning rate 6, it takes 6 to 12 epochs this time, which is a bit faster. The network was tested on patterns [0, 0, 1, 1, 0, 0, 0, 0], [1, 1, 0, 0, 0, 0, 1, 0], and [0, 0, 1, 0, 0, 0, 0, 0], then produced [0, 1, 0, 0, 0, 0, 0, 0], [1, 1, 1, 0, 0, 0, 0, 0], and [1, 1, 1, 0, 0, 0, 0, 0] as outputs. None of them is correct. The network still fails to generalize to untrained patterns, and it still doesn't know the rule. This may indicate the limited learning capacity of the network.

Part3, Rules

The output is [0, 0, 0, 0, 0, 0, 0, 0] in most cases. The network seems always fail to generalize. The additional training seems not helping. It even forgets the training patterns in Part 2A. By the way, I tried adding the additional patterns to the training patterns and training all the 5 patterns at once. The training was still successful, but the network still failed on generalize to all the untrained patterns. I think this can be an issue of initialization. Different initialization has great influence on how the network will be understanding the training patterns.