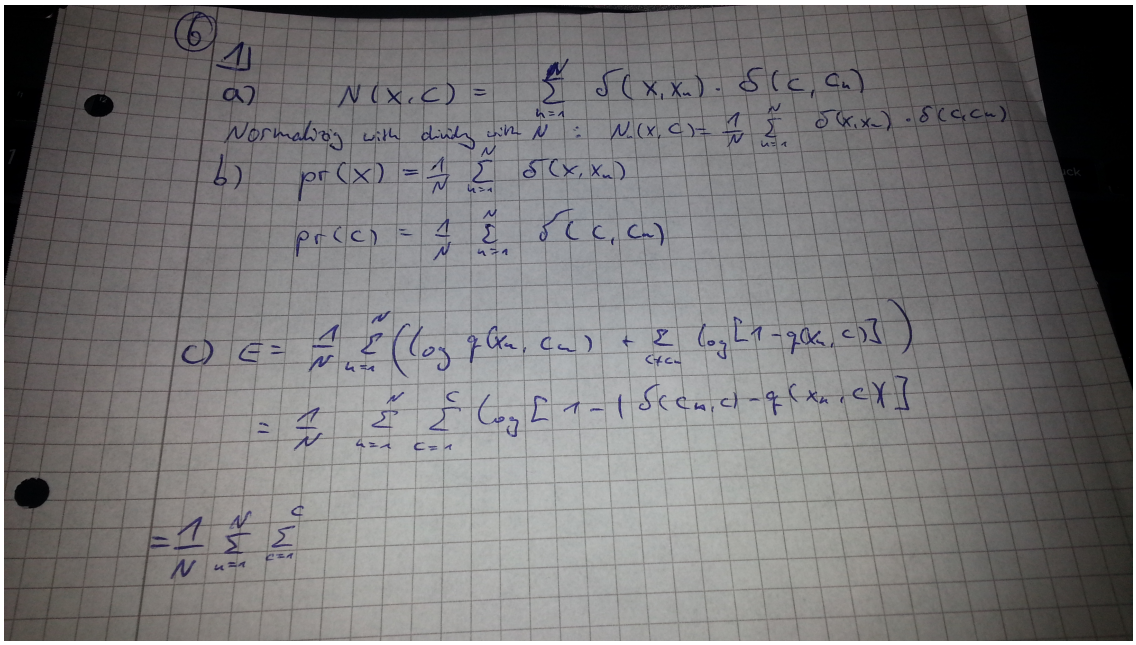
1



2

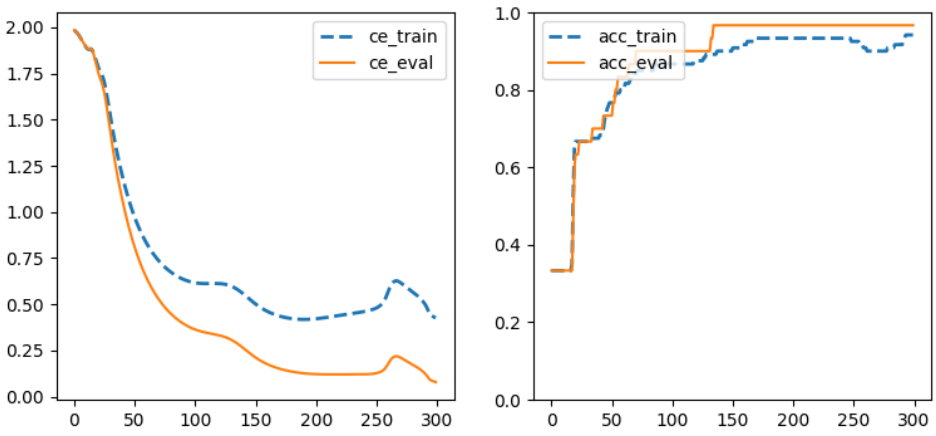
a-d)

e)

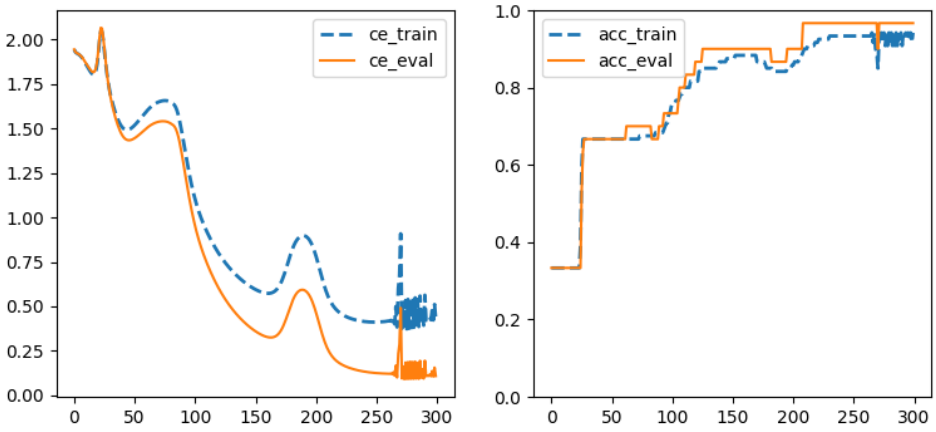
i. Works

ii.

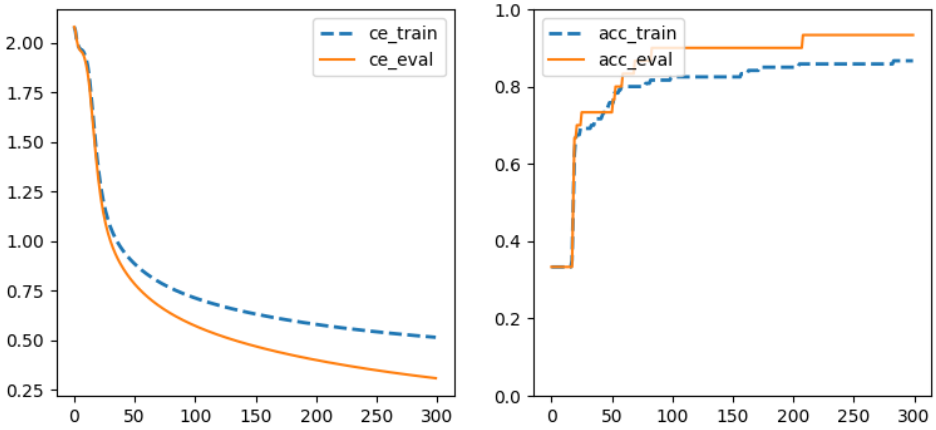
training:online layer:354 lr:0,1 epochs:300.png



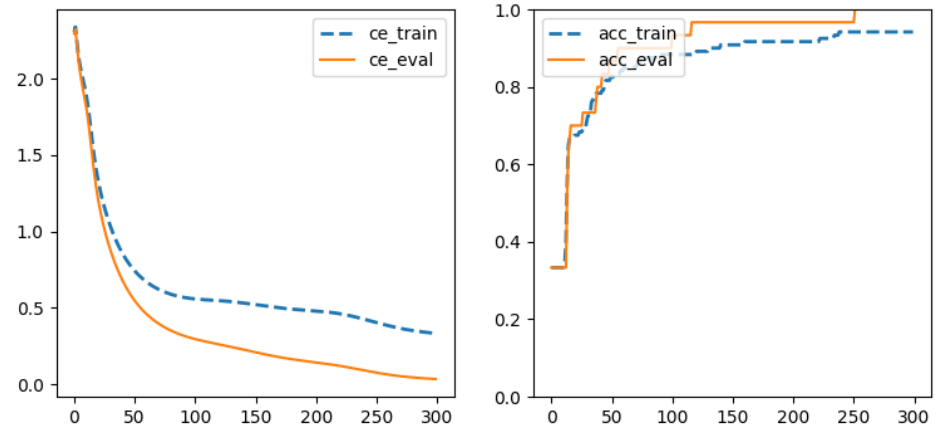
training:online layer:364 lr:0,1 epochs:300.png



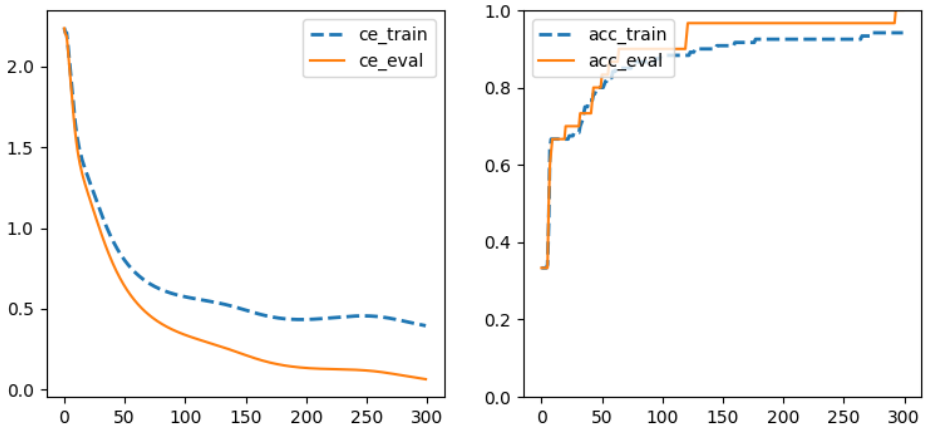
with random weights we sometimes do not learn. This makes the learning process inconsistent (maybe a reason: we are stuck in a local minimum).
training:online layer:354 lr:0,1 epochs:300.png



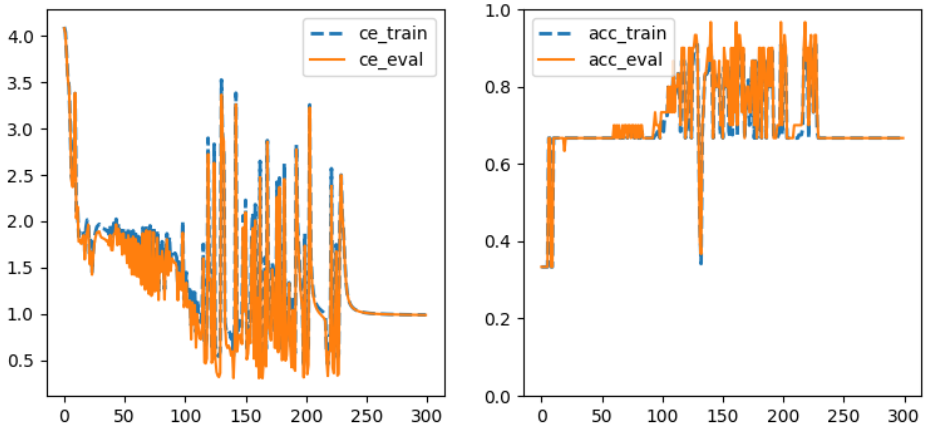
0 works ok.
training:online layer:354 lr:0,1 epochs:300 test:2.png



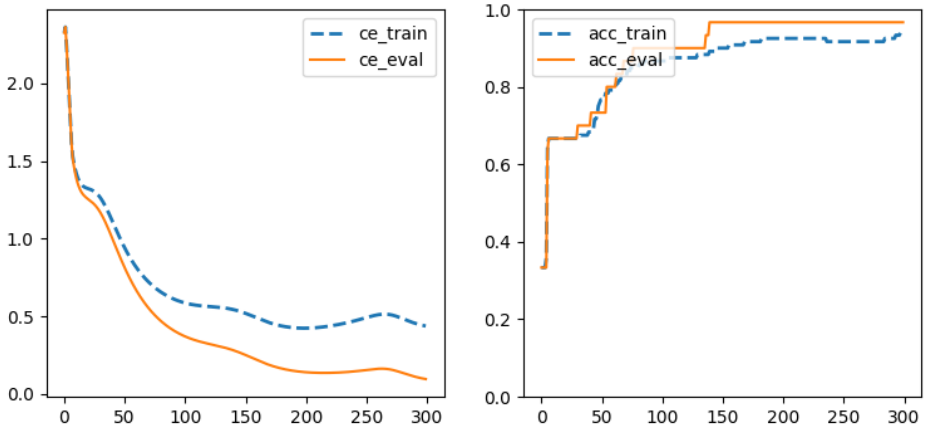
training:online layer:354 lr:0,1 epochs:300 test:1.png



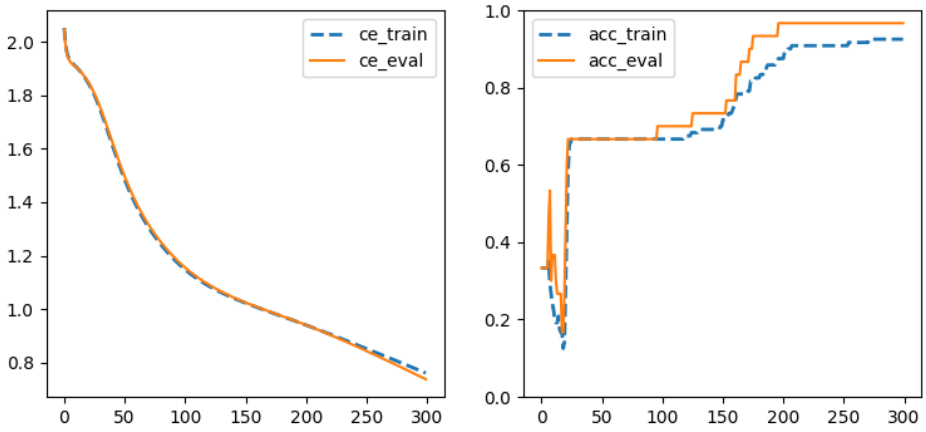
Best results we do have with normalized weights. we get an evaluation accuracy of up to 100% and the network always learns and is consistent. For detailed insight see the graphs
iii.Changing Learning Rate:
training:online layer:354 lr:0,4 epochs:300.png



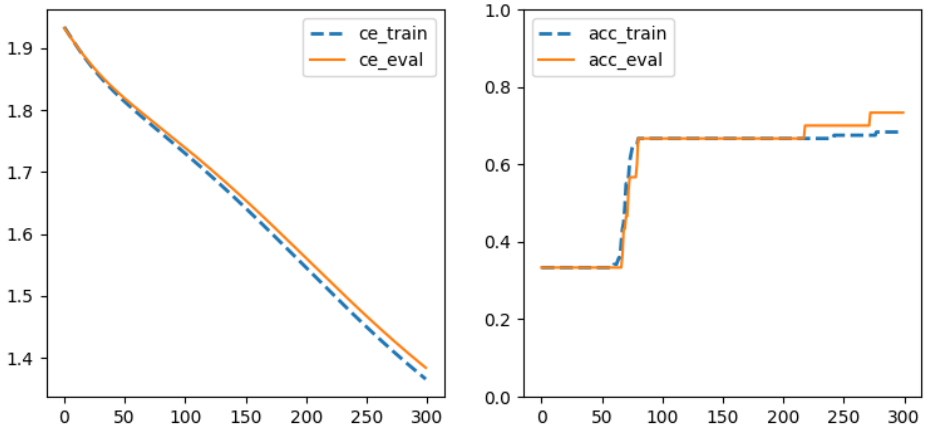
training:online layer:354 lr:0,1 epochs:300.png



training:online layer:354 lr:0,01 epochs:300.png



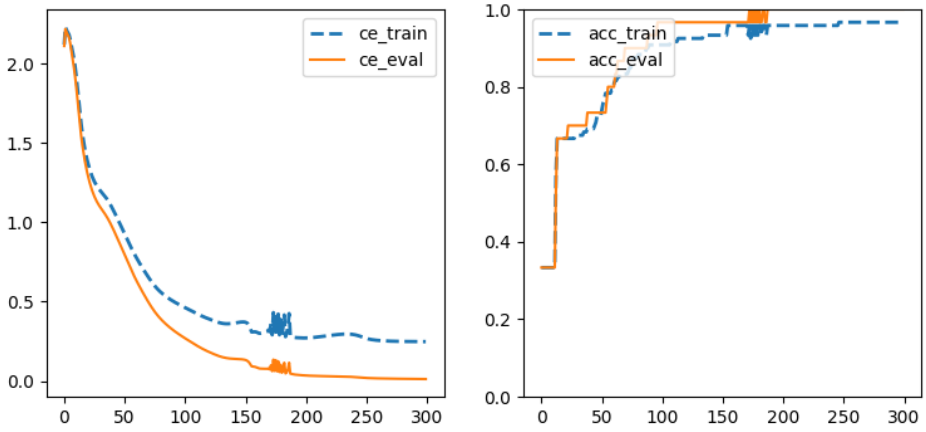
training:online layer:354 lr:0,001 epochs:300.png



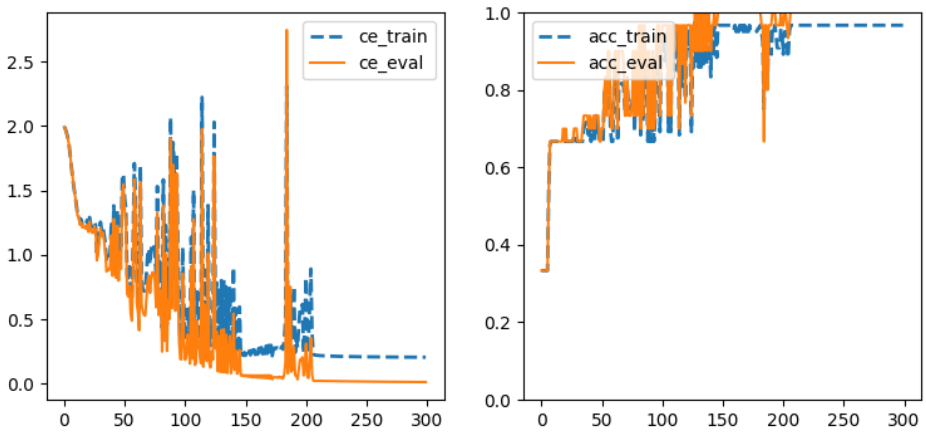
Lower learning rate makes learning curve smoother but we do not reach 100% success rate. With higher learning rate we faster reach the maximum of classification which is not that good (65%). Further the maximum is not stable but flips around a bit.

Changing Mini Batch Size:

training:batch4 layer:354 lr:0,1 epochs:300.png



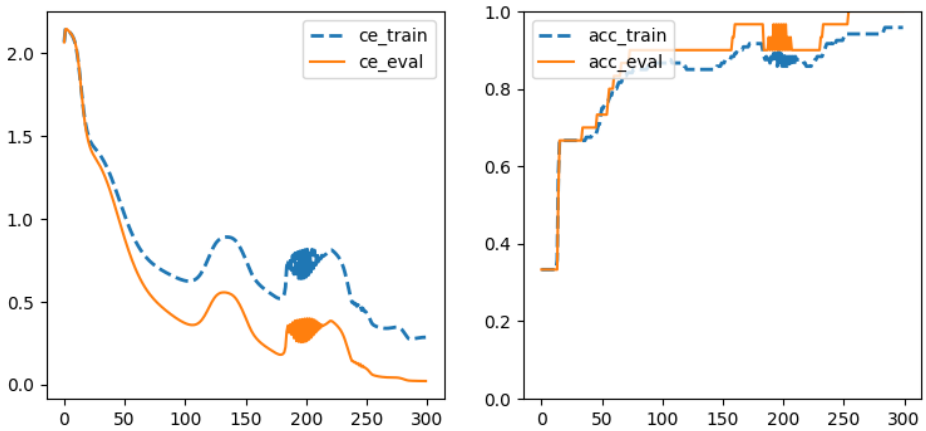
training:batch8 layer:354 lr:0,1 epochs:300.png



Higher batch size (8) improves learning speed. But curve is not as smooth as with only 4 as batch size.

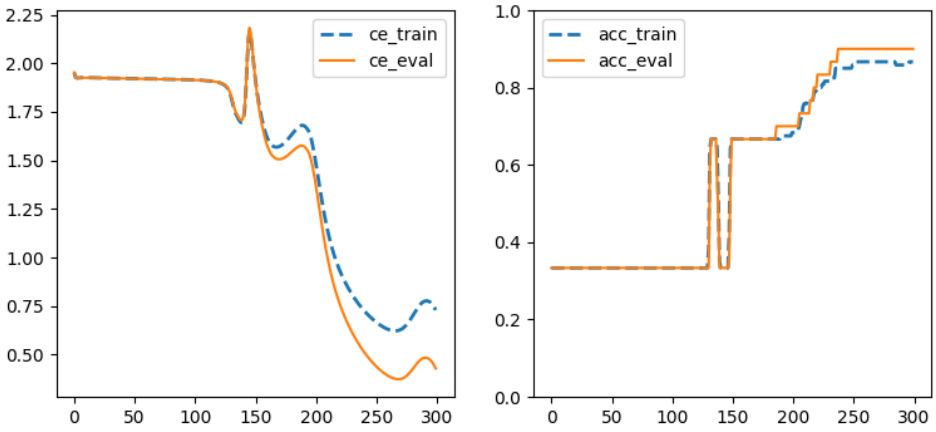
Changing Layer Size:

training:online layer:3554 lr:0,1 epochs:300.png



Second hidden layer makes two peaks in the learning curve.

training:online layer:384 lr:0,1 epochs:300.png



Hidden layer with 8 nodes, classification has only 90% and is slower.