Exercise Nr. 1

b) XOR

With the preset parameters, training the network for 10000 epochs, yielded almost the same loss, as training it for 20 epochs, i.e. the loss stayed close to 0.5 during the whole training phase. We hypothesized that this may be because the optimizer was taking too small steps to arrive at a suitable minimum of the loss function within the given number of epochs. Therefore we increased the learning rate from 0.01 to 1.
layers = (2,20,1)
l = 1
m = 0.5
This resulted in a loss of ≈ 0.008

c) Geometry

- layers=(2,3,1); l = 0.01; m = 0.5 (presets):
  Decision boundary looks too simple to correctly classify the points, which we assumed was due to the network being too small, limiting its capacity to model highly non-linear decision boundaries (the boundaries looked rather linear). Resulting Error $\approx$ 0.2

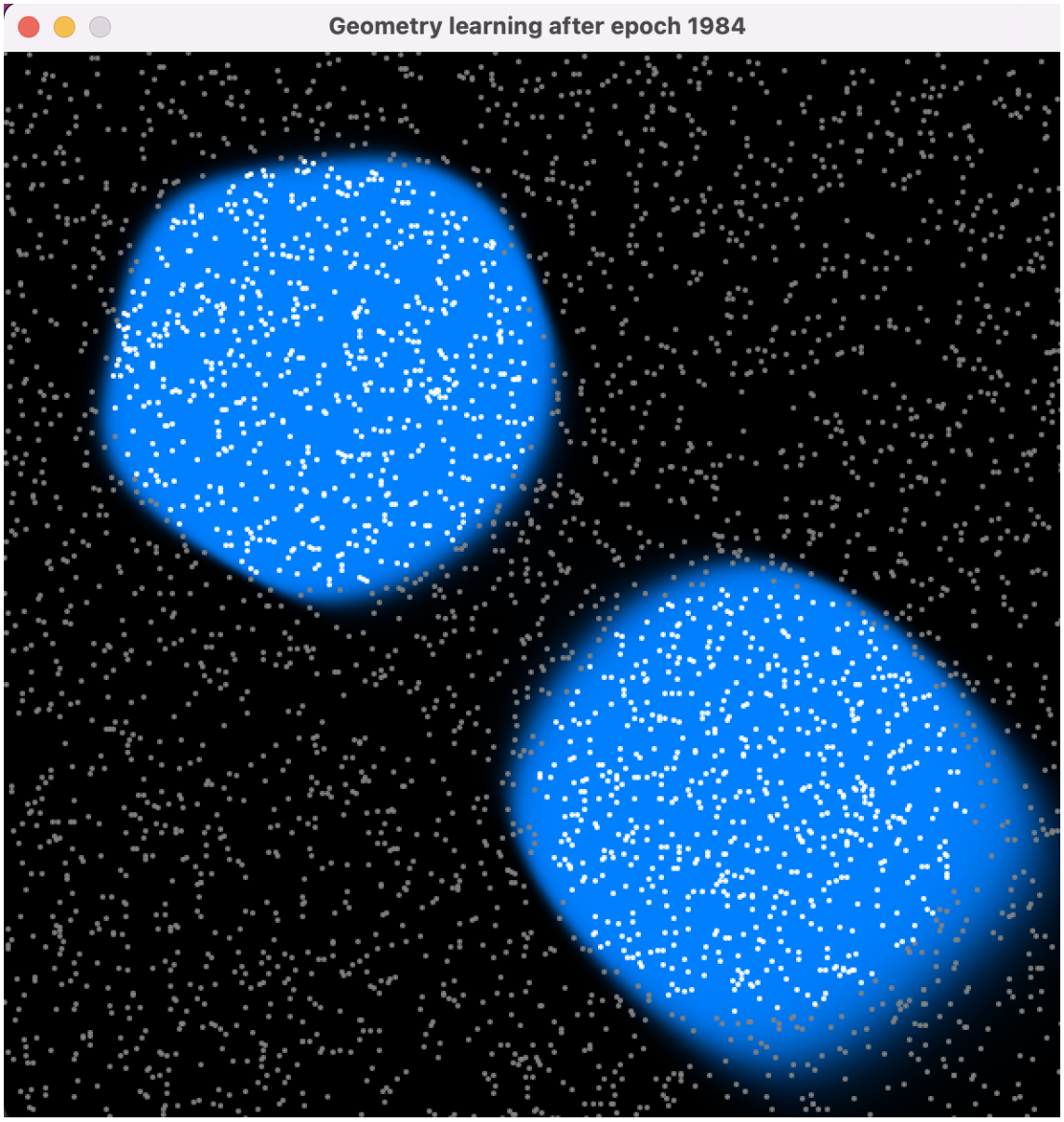
- layers=(2,20,1); l = 0.01; m = 0.5 (layer size increased):
  We increased the number of neurons in the hidden layer in the hope of increasing the capability of the network to model more complex decision boundaries. With this, we obtained a loss of ≈ 0.15, but the two clusters were still not separated (i.e., the blue region extended over the gap between the two disks).

- layers=(2,20,6,1); l = 0.01; m = 0.5 (layer added):
  Looks good in principle, the dots were separated, but not perfect yet. The boundaries were not sharp and produced some false positives around the clusters. This resulted in a loss of ≈ 0.03

- layers=(2,20,6,1); l = 0.1; m = 0.5 (learning rate increased):
  Therefore, we increased the learning rate to try and increase the progress within the same amount of epochs. The error fluctuated, and the blue region "flickered" a bit around the clusters during optimization, so that no real progress was achieved. Resulting loss: ≈ 0.026

- layers=(2,20,6,1); l = 0.008; m = 0.9:
  Instead of taking larger steps, we tried to increase the reliability/robustness of the update steps by using a higher momentum of 0.9 and we finetuned the learning rate from 0.01 to 0.008. This resulted in the best visual separation (see screenshot) and also the lowest loss of our experiments, even thouth it is still not perfect because the boundary is not sharp enough in some regions (especially in the lower righthand corner). Resulting loss: ≈ 0.020

- We tried to expand the network in depth and width, to achieve an even better result, but did not manage to significantly decrease the loss. At the same time this significantly increased the runtime of the training.



Because a one-layered network outputs the result of a sigmoid function applied to a single matrix multiplication, the decision boundary can only be linear (i.e. it can not distinguish points in circles).