Notes on running the code:

The code has been tested with Python 3.8.13, the packages required to run can be installed from the `requirements.txt` file, or manually: `autograd`, `matplotlib`, `imageio`. Running `python sgd.py` will train classifiers using a number of different step sizes (described in more detail below) and save figures and gifs to the directory `./figs`. To speed up the training the viz variable on line 191 can be set to `False` to prevent the visualization of the decision boundary.

1. Gradient

The gradient of

$$\begin{aligned} F_i(\theta) &= \log(1 + e^{-y_i(\langle x_i, w\rangle + b)}) \\ &= \log(1 + e^{-y_i \langle \tilde{x}_i, \theta\rangle}), \\ \text{where} \quad \tilde{x} &= \begin{bmatrix} x_1, x_2, 1 \end{bmatrix}^\top, \quad \theta = \begin{bmatrix} w_1, w_2, b \end{bmatrix}^\top \end{aligned} \tag{1}$$

is analytically computed as

$$\nabla F_i(\theta) = \frac{-y_i \tilde{x}_i e^{-y_i \langle \tilde{x}_i, \theta\rangle}}{1 + e^{-y_i \langle \tilde{x}_i, \theta\rangle}}. \tag{2}$$

This is the same as the autograd[1] automatic differentiation of the funciton. The difference between the analytical and differentiated gradients are computed when running the `sgd.py` file.

2 & 3. SGD with different step sizes

The training was done using step sizes, $\alpha = \begin{bmatrix} 0.001, 0.01, 0.05, 0.1, 1.0 \end{bmatrix}$, as well as with α decreasing according to $\alpha_k = \frac{\alpha}{\sqrt{k+1}}$ with initial value of 0.05. The training and test errors along with the difference between training error at epoch k and $k-1$ for the different step sizes are shown below. The training terminates after 250 epochs.

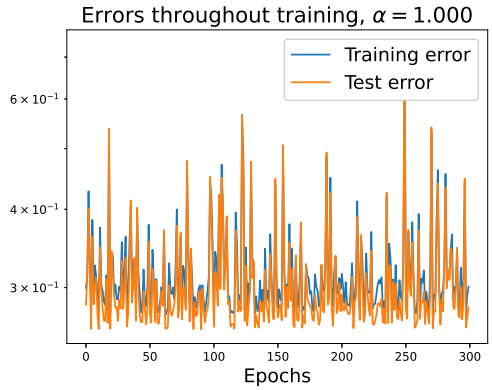
As is expected training with a larger step size quickly converges to a decision boundary around which it oscillates, the bigger the step size the bigger the oscillations. Among the evaluated step sizes $\alpha = 0.001$ is the only one for which the training does not converge in 250 epochs. Looking at Figure 7, showing the final decision boundaries, it is clear that basically the same boundary is found for all α's except 1.0 and 0.001. For the large α this is due to large oscillations and for the small α this is due to lack of convergence. Animations of how the boundary changes throughout training can be seen in the attached `.gif` files.

4. Final remarks

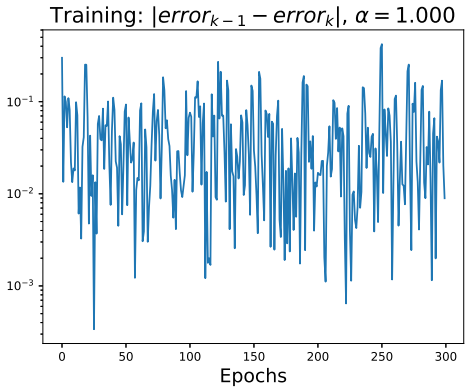
Notable for the experiments presented here is that the test error in general is slightly lower than the training error. As the two classes clealy are not linearly separable the classifier will not fit to noisy edge cases in the training data to a great extent. Hence, the smaller test error is probably explained by a simpler dataset, i.e. less data points in the misclassified regions, due to the stochastic data generation.

I ran the training for a set number of epochs, mainly for the purpose of visualisation and comparison between the different learning rates. In this setup it obviously does not make any difference to keep training after the training error has converged. The training could be stopped when the difference $\text{error}_{k-1} - \text{error}_k$ is small enough or negative which would suggest that the gradient step passed the minima.

[1] `https://github.com/HIPS/autograd`

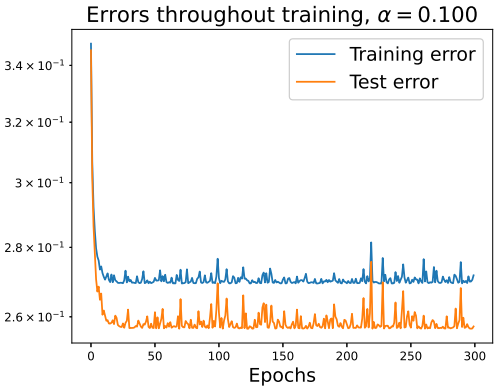


(a)

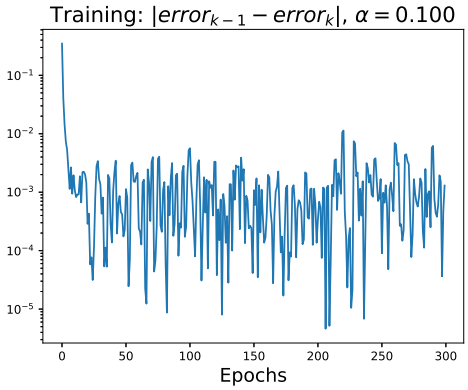


(b)

Figure 1

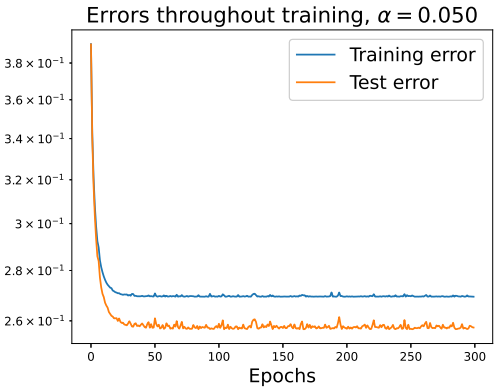


(a)

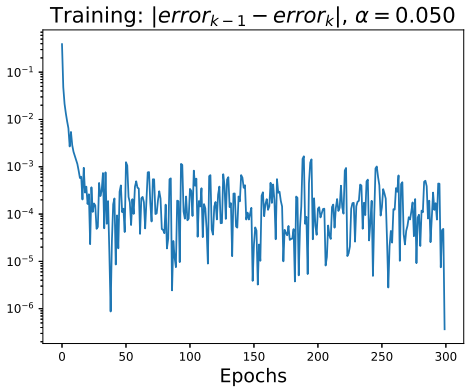


(b)

Figure 2

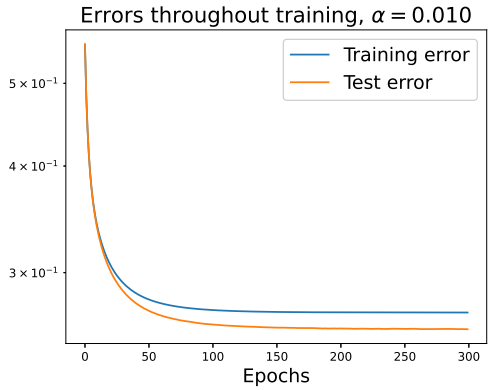


(a)

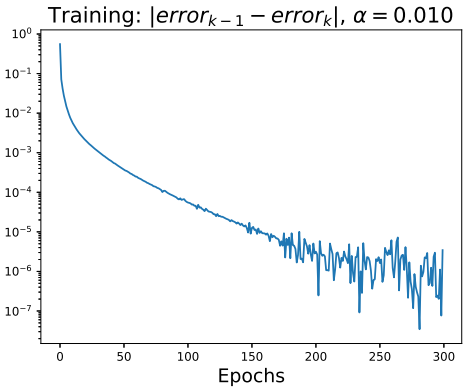


(b)

Figure 3

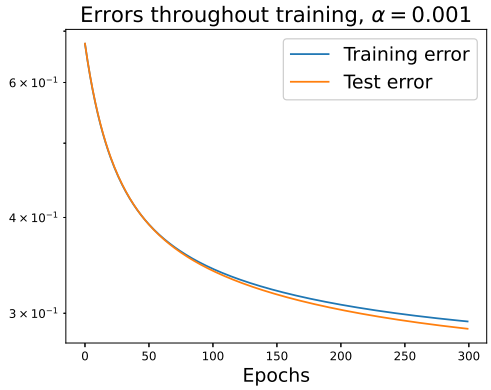


(a)

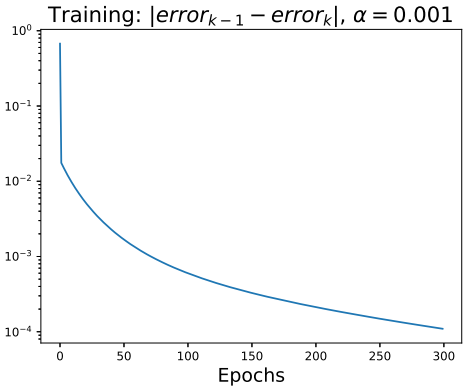


(b)

Figure 4

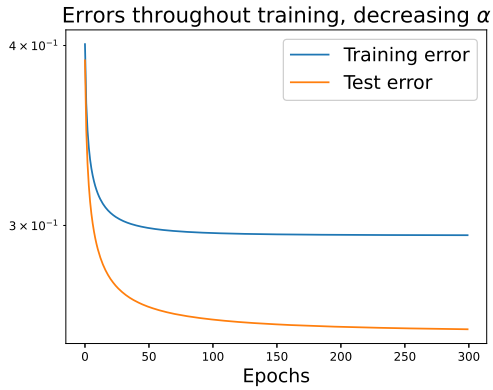


(a)

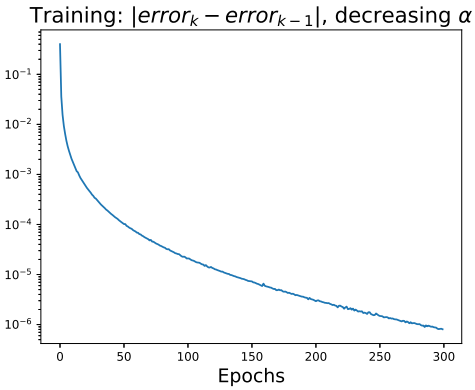


(b)

Figure 5

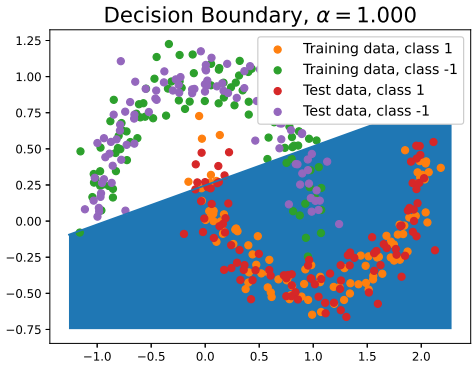


(a)

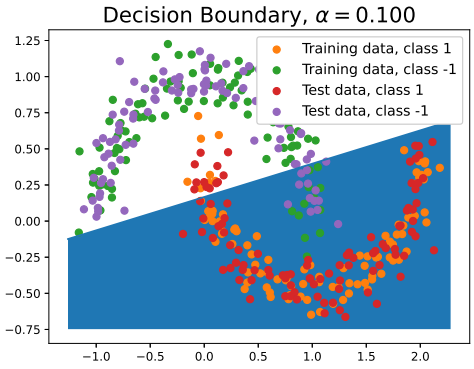


(b)

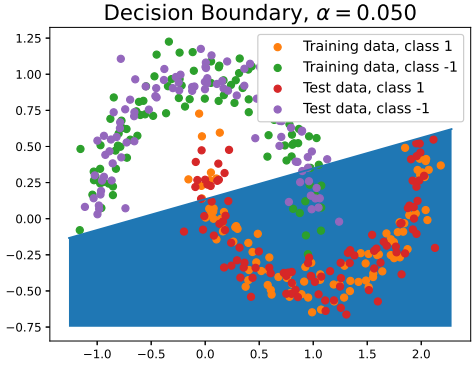
Figure 6



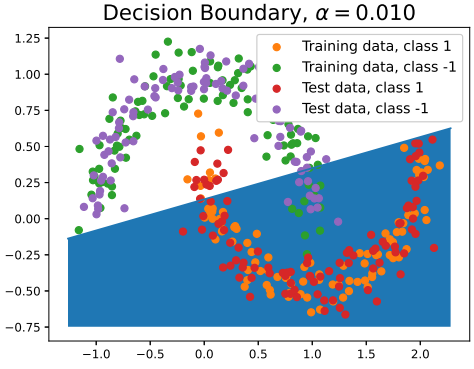
(a)



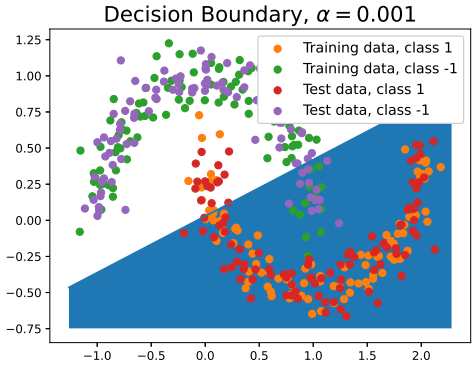
(b)



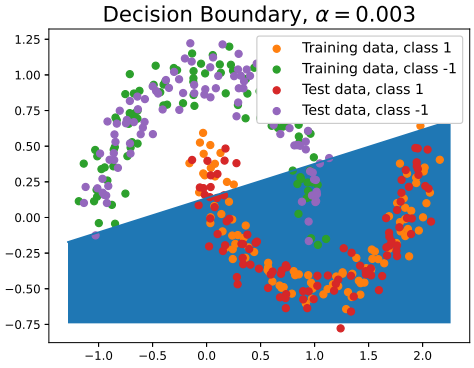
(c)



(d)



(e)



(f) Decreasing α

Figure 7