

KTH Royal Institute of Technology  
DD2424-VT19-1 Deep Learning in Data Science  
Assignment 1 Bonus Points

Marina Herrera Sarrias

April 12, 2019

## 1 Exercise 2.1

Reports on your trained network with the best test accuracy, what improvements you made and which ones brought the largest gains.

The optimization performed on the network consisted on:

- Shuffle the order of the training examples at the beginning of each epoch.
- Decaying the learning rate by a factor of 2 after every 5 epochs.
- A grid search for finding *good* values of the hyper-parameters: `eta`, `lambda`, `n_batch`, and `n_epoch`. I tested all possible combination of all the following parameter values:
  - `eta` = [1e-4, 1e-3, 1e-2]
  - `lambda` = [1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 0]
  - `n_batch` and `n_epochs` = [(10, 200), (50, 200)]
- The grid search was also applied to the ordering shuffle and the decaying learning rate cases.

From all the cases above mentioned the parameter setting that maximized the accuracy of the model was under the data shuffling scenario, with a 39.07% of the test accuracy, using the following parameters: `eta` = 0.001, `lambda` = 0.01, `n_batch` and `n_epochs` = [(10, 200)]. Which compared to the test accuracy results obtained using the original model, under the same parameter setting, there is a 0.81% improvement.

This, as the first epochs are usually the ones that bring more gain, and when no shuffling is performed, the same data will always be at the beginning of the training.

The second best model was obtained applying `eta` decay, with a test accuracy of 38.87% for the parameter setting: `eta = 0.001`, `lambda = 0.0001`, `n_batch` and `n_epochs` = [(50, 200)], which has only show a small (0.26%) improvement with respect to the non-optimized model, using the same parameter setting.

In general, the results obtained after applying the different optimization tricks did not result in big improvements. Most of the results obtained for the different parameter settings were very similar, among the three scenarios.

## 2 Exercise 2.2

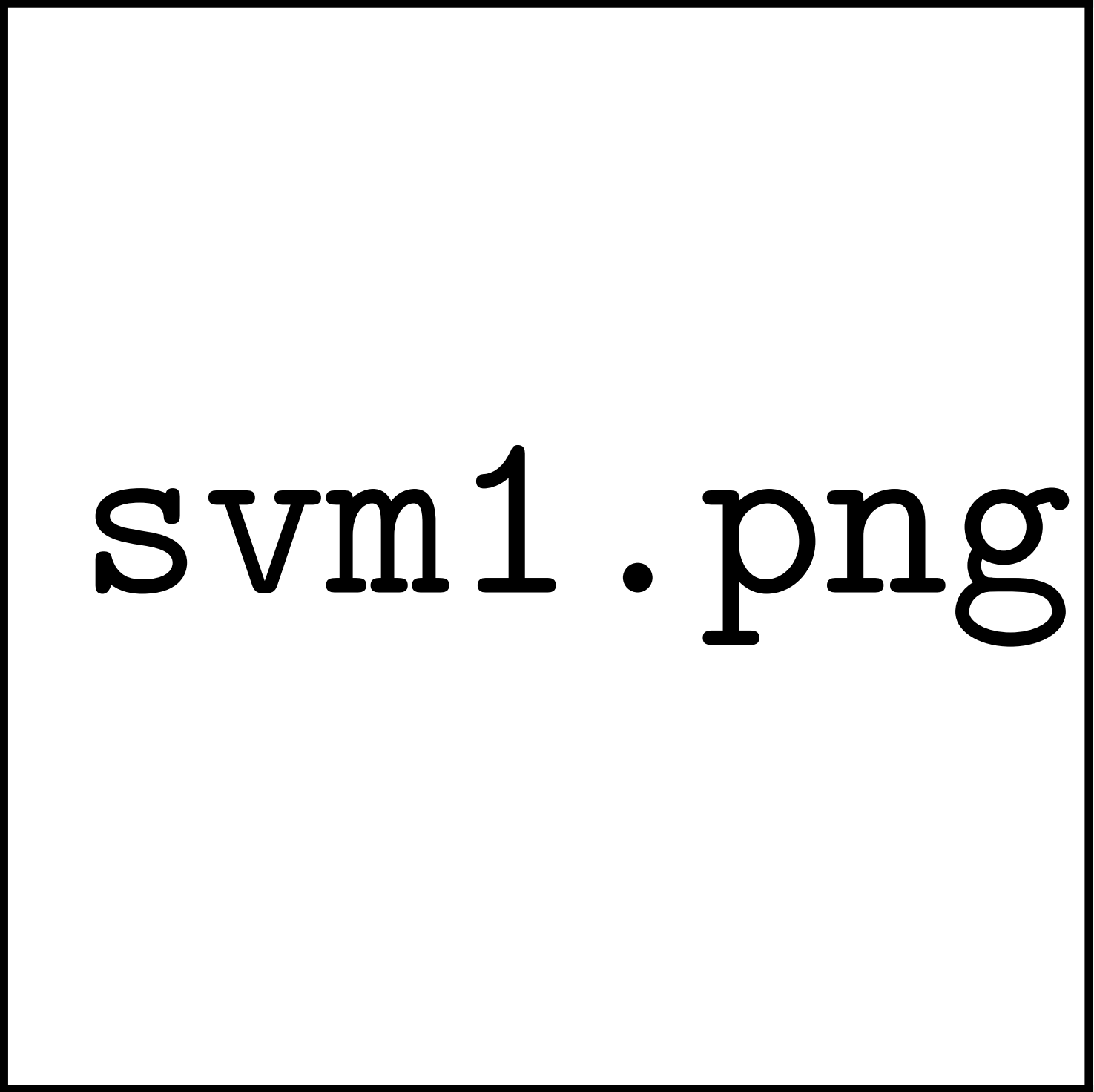
Compare the test accuracy of the network trained with the SVM loss compared to the cross-entropy loss (for several sensible training parameter settings).

After applying the same grid search proposed in the previous section the test accuracy obtained with the Network trained using the cross-entropy loss had the highest values most of the time (30/36) in comparison with the ones obtained on the Network trained with the SVM multi-class loss(6/36).

This, in addition to the fact that training the Network using the SVM multi-class, was considerably more computationally expensive, compared to the Network trained using cross-entropy loss.

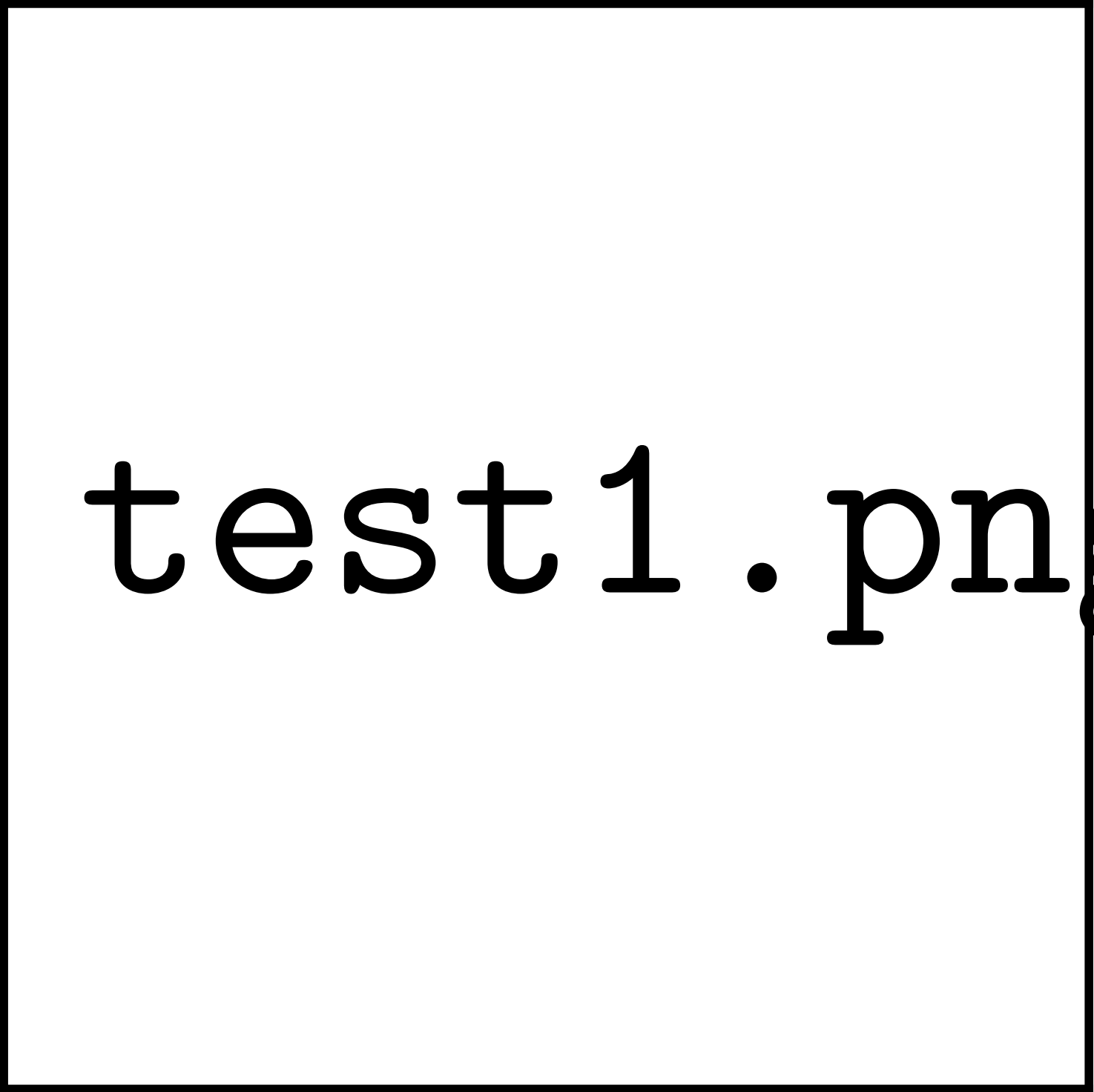
Some of the tested parameter settings:

- `lambda=0.1, n_epochs=200, n_batch=10, eta=.01`



svm1.png

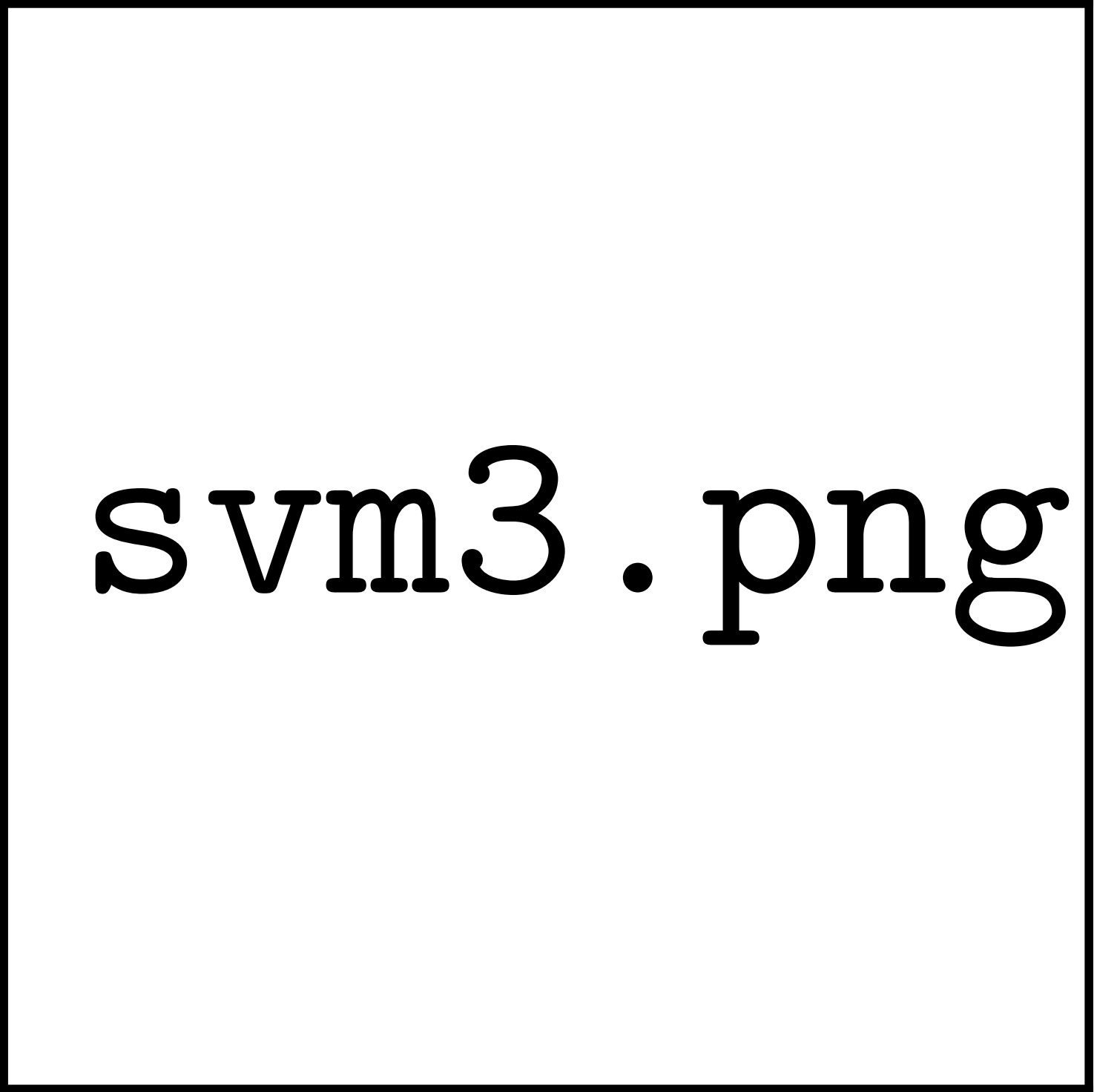
Figure 1: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.01`, `n_epochs= 10` and `lambda= 0.1`. Using the SVM multi-class loss function



test1.png

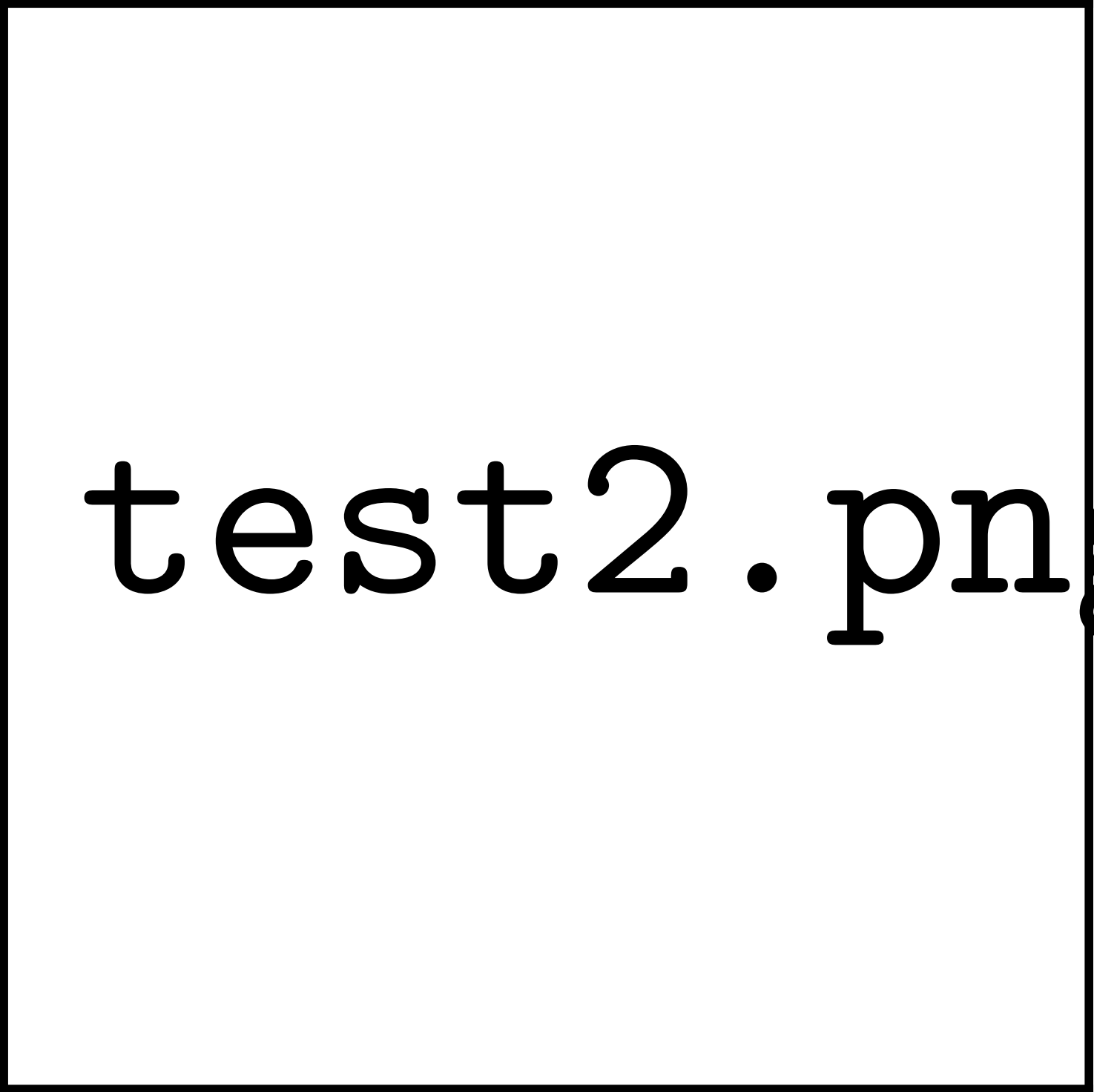
Figure 2: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.01`, `n_epochs= 10` and `lambda= 0.1` Using the cross-entropy loss.

- `lambda=1e-5, n_epochs=200, n_batch=50, eta=.001`



svm3 . png

Figure 3: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.01`, `n_epochs= 10` and `lambda= 0.1`. Using the SVM multi-class loss function

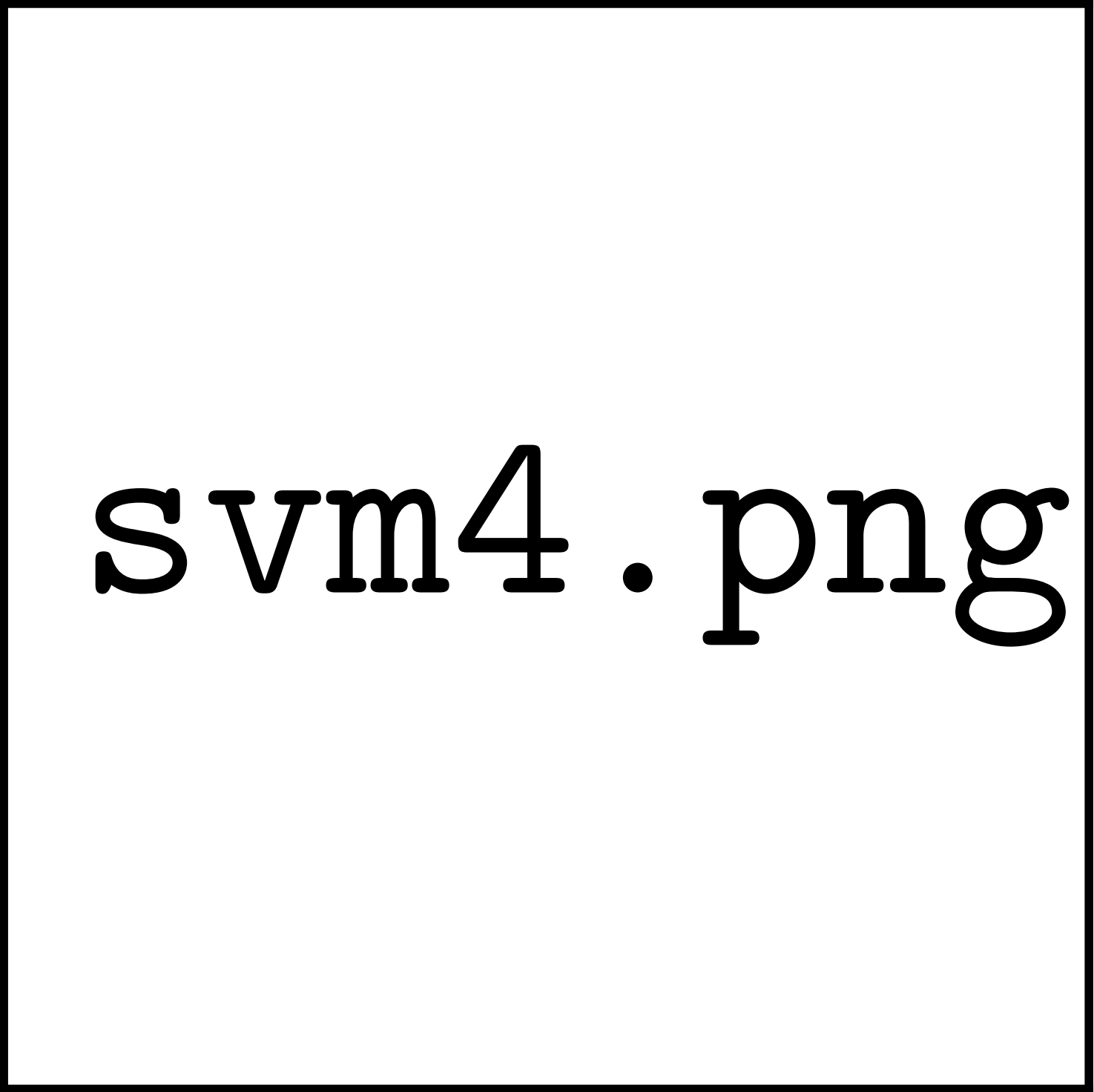


test2.png

Figure 4: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.01`, `n_epochs= 10` and `lambda= 0.1` Using the cross-entropy loss.

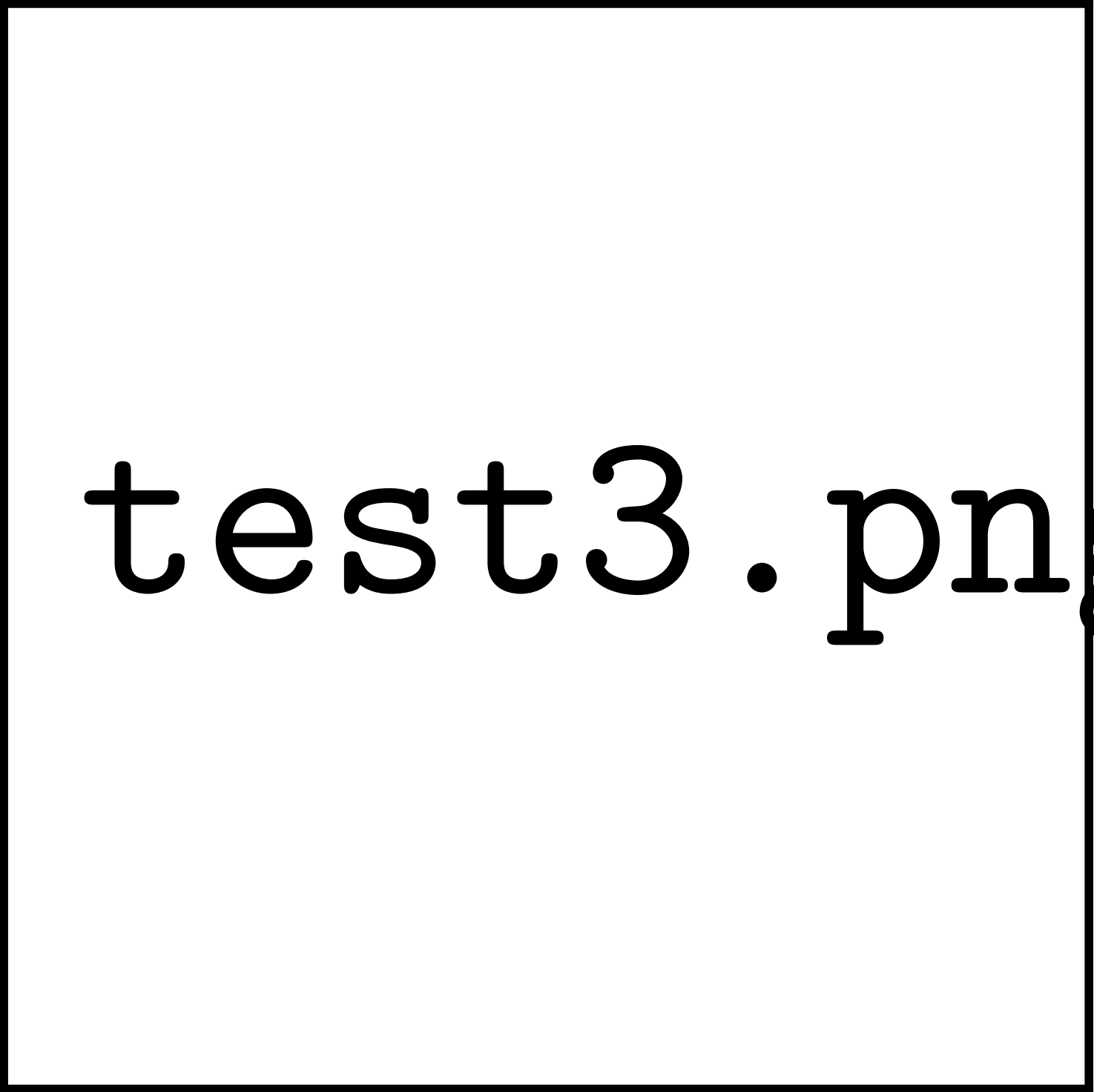


- `lambda=0.001, n_epochs=200, n_batch=50, eta=0.0001`



svm4.png

Figure 5: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.0001`, `n_epochs= 50` and `lambda= 0.001`. Using the SVM multi-class loss function



test3.png

Figure 6: The network was trained with the following parameter settings: `n_batch=200` `eta= 0.0001`, `n_epochs= 50` and `lambda= 0.001` Using the cross-entropy loss.