ML2_assignment2

March 22, 2024

1 Assignment 2

1.1 General information

You are required to submit two files to Moodle: an `.ipynb` file and the rendered `.pdf` file with your solutions. Do not zip them together so I will be able to annotate the `.pdf` directly.

Please give short (2-3 sentences) interpretations / explanations to your answers, not only the program code and outputs. Be concise and focused (less could be more ;)).

Grades will be distributed with the following rule: from the points you earn, you get 100% if you submit until the due date (`2024-04-05 20:00`), 50% within 24 hours past due date, and 0% after that.

1.2 Classify fashion images on the MNIST data (20 points)

Take the alternative version of the famous "MNIST dataset", which consists of images of Zalando's articles. Your task is to correctly classify the images into one of the ten categories, such as dress or shirt. The images are in exactly the same format as we saw for the handwritten digits: 28x28 pixel grayscale images. The task is to build deep neural network models to predict the items. You can use either `sklearn` or `keras`; to get the data, go to the corresponding Kaggle page or use the `fashion_mnist.load_data()` function from the `keras.datasets` module. Make sure you split the training set into two sets: one for training your models on and one for validation and model selection. You can work with a relatively small train set if you have computational problems.

- (2 points) What would be an appropriate metric to evaluate your models? Why? (*Hint:* No code required.)
- (2 points) Get the data and show some example images from the data.
- (2 points) Train a simple fully connected single hidden layer network to predict the items. Remember to normalize the data similar to what we did in class. Make sure that you use enough epochs so that the validation error begins to level off - provide a plot of the training history.
- (6 points) Experiment with different network architectures and settings (number of hidden layers, number of nodes, regularization, etc.). Train at least 3 models. Explain what you have tried and how it worked.
- (6 points) Try to improve the accuracy of your model by using convolution. Train at least two different models (you can vary the number of convolutional and pooling layers or whether you include a fully connected layer before the output, etc.).

- *(Optional for 5 points) Try to use a pre-trained network to improve accuracy.*
- (2 points) Select a final model and evaluate it on the test set. How does the test error compare to the validation error?