Deep Learning Assignment 5 in FS 2022

Binary and Categorical Classification

Microsoft Forms Document:
https://forms.office.com/r/4PAnYRs2Bf

Manuel Günther

Distributed: Friday, March 25, 2022
Discussed: Friday, April 1, 2022

For this exercise, we will switch to an implementation in PyTorch. The goal of this exercise is to get used to some of the concepts in PyTorch, such as implementing the network, the loss functions, the training loop and accuracy computation.

1 Dataset

For the exercise, we will use two different datasets, one for binary classification and one for categorical classification. The binary classification dataset[1] contains features extracted from emails, which are classified as either spam or not. The categorical classification dataset[2] contains some manually selected features for three different types of wines. For the former, the class is provided in the last column of the data file, whereas for the latter, the first index provides class information.

Task 1: Dataset Loading The first task deals with the loading of the datasets. When training networks with PyTorch, all data needs to be stored as datatype `torch.tensor`. The data should be split between input sets $\mathbf{X} = [\vec{x}^{[1]}, \ldots, \vec{x}^{[N]}]^T \in \mathbb{R}^{N \times D}$ and targets. There is **no need to add a bias neuron to the input**, and the transposition of the data matrix is different from what we have seen before. For the targets, we have to be more careful as there are differences w.r.t. the applied loss function. For binary classification, we need $\mathbf{T} = [[t^{[1]}, \ldots, t^{[N]}]]$ to be in dimension $\mathbb{R}^{N \times 1}$ and of type `torch.float`. For categorical classification, we only need the class indexes $\vec{t} = [t^{[1]} \ldots, t^{[N]}]$ to be in dimension $\mathbb{N}^N$ and of type `torch.long`. Return both the input and the target data for a given dataset.

Test 1: Assert Valid Data Call the functions to extract the input and targets for both datasets. Assure that the input and target tensor shapes are appropriate. Assure that all target values are in the appropriate ranges, and that the data types are correct.

Task 2: Training and Validation Split The data should be split into 80 % for training and 20 % for validation. Implement a function that takes the full dataset (X, T) and returns (X_t, T_t, X_v, T_v) accordingly. What do we need to take care of before splitting the data?

Task 3: Input Data Standardization As we have seen last week, the standardization of the data provides many advantages. Hence, in this task you should write a function that takes (X_t, X_v) and standardizes them by subtracting the mean and dividing by the standard deviation of X_t, and returning the standardized versions of both. Assure that each input dimension is standardized individually.

[1] https://archive.ics.uci.edu/ml/datasets/spambase
[2] https://archive.ics.uci.edu/ml/datasets/wine

2 Network Training

We will use a two-layer fully-connected network with D input neurons, K hidden neurons and O output neurons. Depending on the task, D and O need to be selected appropriately, while K is a parameter to play around with.

In PyTorch, the easiest way to implement a network is by providing the requested sequence of layers to `torch.nn.Sequential`, which will build a network containing the given layers. We will use two `torch.nn.Linear` layers and one `torch.nn.Tanh` activation function in between. The network will return the logits $\vec{z}$ for a given input $\vec{x}$.

Task 4: Implement Network Implement a function that generates a network with D input neurons, K hidden neurons and O output neurons in PyTorch.

Task 5: Accuracy Computation To monitor the training process, we want to compute the accuracy. The function will obtain the logits $\vec{z}$ extracted from the network and the according target t. Assure that this function works both for binary and categorical classification. How can we identify, which of the two variants is currently required?

Test 2: Accuracy Function Design some test data such that you can test the accuracy computation. Make sure that you test both cases, i.e., binary and categorical classification.

Task 6: Training Loop Implement a function that takes all necessary parameters to run a network training on a given dataset. This week, we will run gradient descent, i.e., we will train on the whole training set in each epoch. Select the optimizer to be `torch.optim.SGD`. Implement a training loop over 10'000 epochs with a learning rate of 0.1. In each loop, compute and store the training loss, training accuracy, validation loss and validation accuracy. Make sure that you train on the training data only. At the end, return the lists of these values.

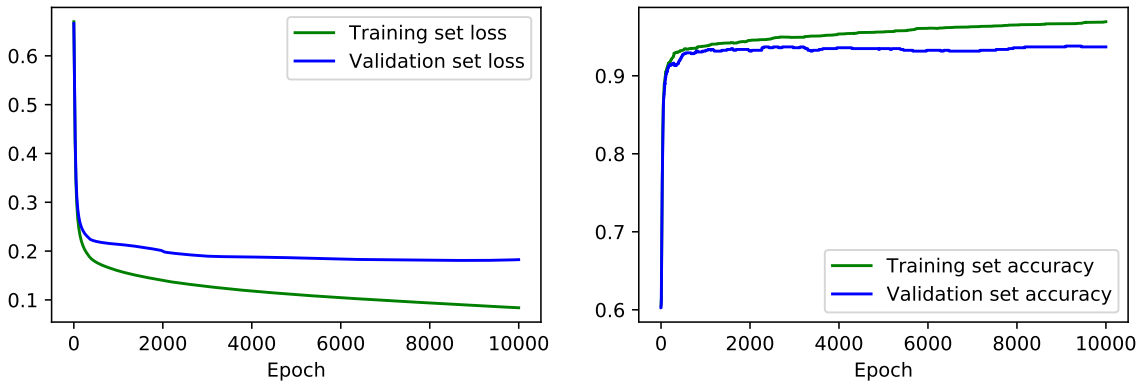
3 Training and Evaluation

Finally, we want to train our network on our data and plot the accuracy and loss values that were obtained through the epochs. Exemplary plots can be found in Fig. 1.

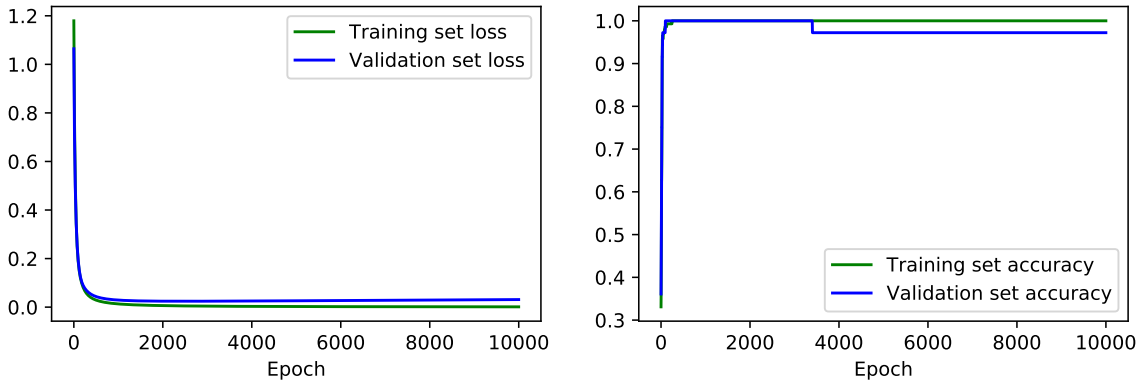
Task 7: Plotting Function Implement a function that takes the four lists of loss and accuracy values and plot them into two plots. The first plot should contain the loss values for both training and validation set. The second plot should contain the according accuracy values.

Task 8: Binary Classification Load the data for binary classification, using the `"spambase.data"` file and split the data into training and validation sets. Standardize both training and validation input data using the function from Task 3. Instantiate a network with the correct number of input neurons, a given number of K hidden neurons and one output neuron. Instantiate the binary cross entropy loss function. Train the network with our data for 10'000 epochs and plot the training and validation accuracies and losses.

Task 9: Categorical Classification Perform the same task for our dataset `"wine.data"` for categorical classification. Change the number of input and output neurons accordingly. Select the loss function for categorical classification. How many hidden neurons do we need to achieve 100 % classification accuracy on the training data?



(a) Binary Classification



(b) Categorical Classification

Figure 1: This figure shows some exemplary progression plots of loss values and accuracies for binary and categorical classification.