Neural Networks and Deep Learning
Homework 1

Amedeo Giuliani - 2005797

January 11, 2022

1 Regression

The implemented neural network is a *Feed-forward Neural Network* (FFN) with two fully-connected hidden layers, as depicted in Fig 1.

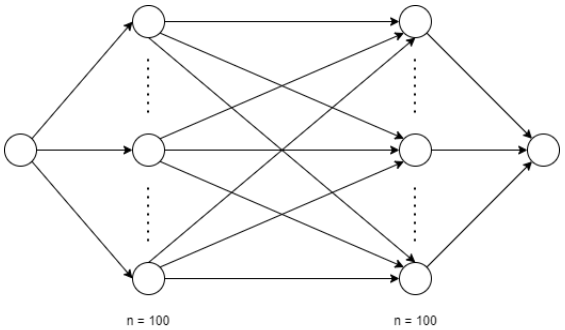
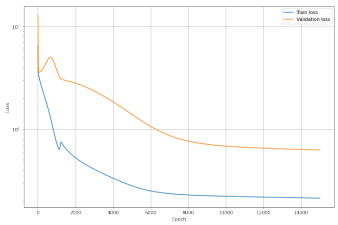


Figure 1: Employed Feed-forward neural network model.

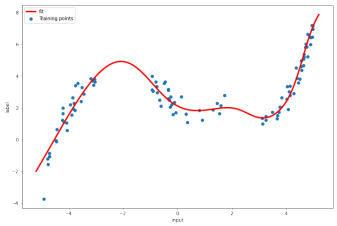
The activation function of hidden layers is the Sigmoid function, while the output one has linear activation. We start with the *Stochastic Gradient Descent* (SGD) optimizer with no momentum, a learning rate of 0.01, and a batch size of 25. For the loss function we use the *Mean Square Error* (MSE). The FFN has been trained for $15 \cdot 10^3$ epochs. The results are shown in Fig. 2a, Fig. 2b, and Fig. 2c. It can be seen that while the FFN can follow the first peak sufficiently good, it is not capable of approximating well the second peak. If we set a momentum of 0.9 things get better, and the number of epochs can be reduced to $8 \cdot 10^3$ since the algorithm converges faster. The plots are depicted in Fig. 3a, Fig. 3b, and Fig. 3c. Advanced optimizer methods can be employed, such as Adam. The learning rate has been set to 10^{-3} and the number of epochs was further reduced to $5 \cdot 10^3$. The results are shown in Fig. 4a, Fig. 4b, and Fig. 4c. In both cases the FFN does not "overshoot" the first peak and now it can follow nicely the second peak too.
By feeding to the FFN the test set, and comparing the predictions the model makes with the ground truth, we can evaluate the test loss. With SGD, the true risk is 0.2739, with SGD and momentum it is 0.1151, while with Adam it is 0.1079.

1.1 Regularization

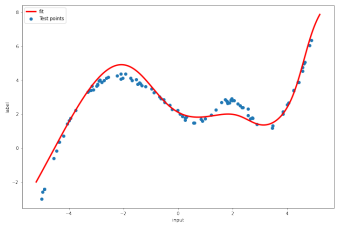
As a further step, regularization methods can be applied, such as dropout and weight decay (L2 regularization). For example, we try with dropout equal to 0.25 and weight decay set to 10^{-4}.



(a) Training vs validation loss with SGD.

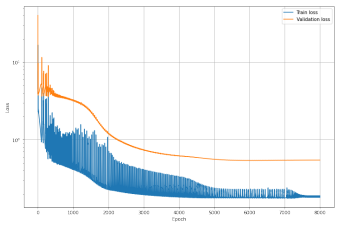


(b) Fitted line vs training points with SGD.

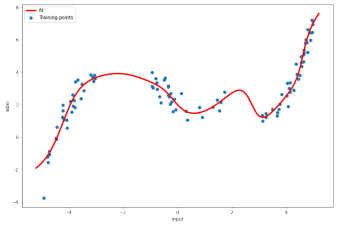


(c) Fitted line vs real (noisy) model with SGD.

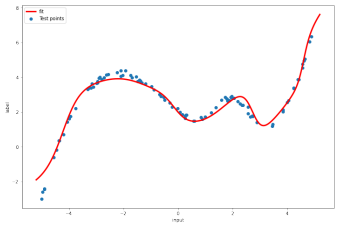
Figure 2



(a) Training vs validation loss with SGD and momentum.

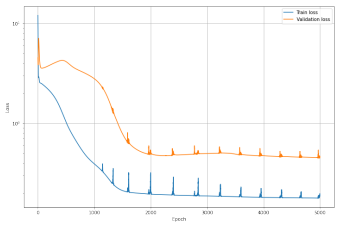


(b) Fitted line vs training points with SGD and momentum.

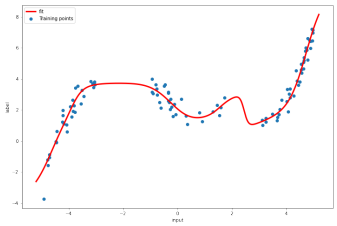


(c) Fitted line vs test points with SGD and momentum.

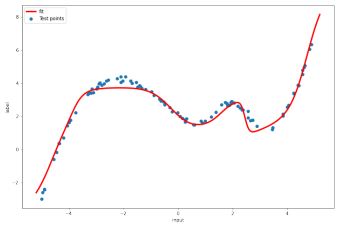
Figure 3



(a) Training vs validation loss with Adam.



(b) Fitted line vs training points with Adam.



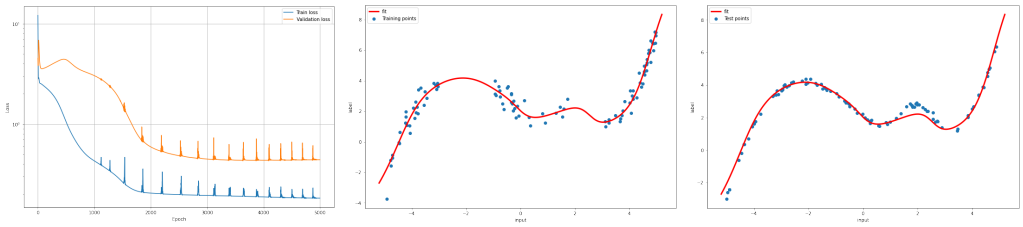
(c) Fitted line vs test points with Adam.

Figure 4

In the first case the results are terrible, while in the second case the FFN follows almost perfectly the first peak but underestimate the second one, as shown in Fig. 5a, Fig. 5b, and Fig. 5c. The test loss is 0.0976.

1.2 Hyperparameters tuning

A grid search with k-fold cross validation – with k = 3 – has been implemented with the dictionary of parameters shown in Tab. 1:
The optimal set of parameters turned out to be: [Batch size: 10, Hidden layer 1 neurons: 100, Hidden layer 2 neurons: 80, Dropout: 0, Weight decay: 0]. The results, which are quite similar with respect to the previous configuration, are shown in Fig. 6. Now the test loss is 0.0849.



(a) Training vs validation loss with Adam and L2 regularization.

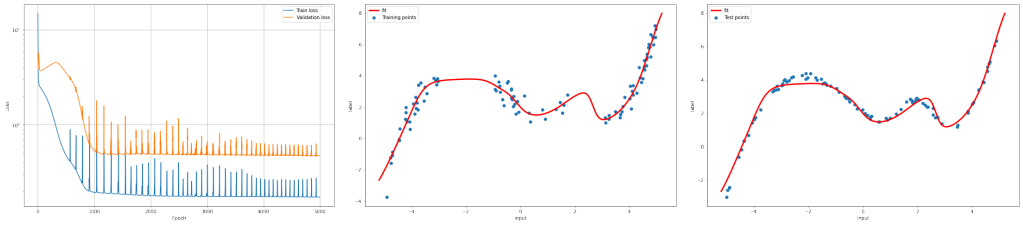
(b) Fitted line vs training points with Adam and L2 regularization.

(c) Fitted line vs test points with Adam and L2 regularization.

Figure 5

Table 1: Hyperparameter dictionary

Hyperparameter	**Values**
Hidden layer 1 neurons	$[80, 100]$
Hidden layer 2 neurons	$[80, 100]$
Dropout	$[0, 0.25, 0.5]$
Weight decay	$[0, 10^{-5}, 10^{-4}]$
Batch size	$[10, 25]$



(a) Training vs validation loss after grid search.

(b) Fitted line vs training points after grid search.

(c) Fitted line vs test points after grid search.

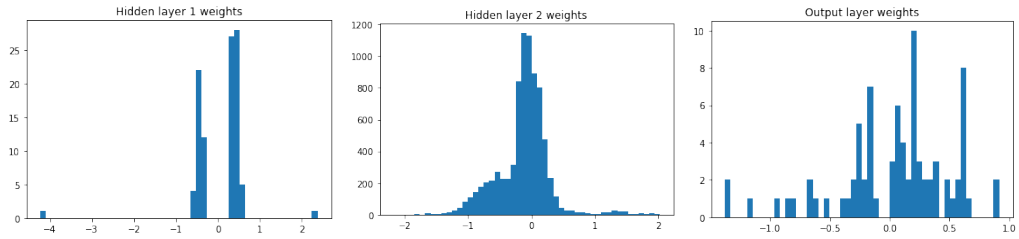
Figure 6

1.3 Weights histograms

The weights histograms for the two hidden layers and the output layer are depicted in Fig. 7.

2 Classification

The implemented neural network is a *Convolutional Neural Network* (CNN) with three convolutional layers and two fully connected layers, as depicted in Fig. 8.
In particular, the first layer takes in input one channel (since images are in gray-scale), while the output layer has 10 neurons, as much as the number of classes of the data set, and is followed by a SoftMax transformation, to scale values between 0 and 1. In this way, for a certain input, the outputs will be the probabilities for each class. The chosen loss function is the cross entropy loss. The above described model has been trained for 50 epochs, with the Adam optimizer, a learning rate of 10^{-3}, and a batch size of 600 samples. In Fig. 9 are shown the training and



(a) Weight histogram for the first hidden layer.

(b) Weight histogram for the second hidden layer.

(c) Weight histogram for the output layer.

Figure 7

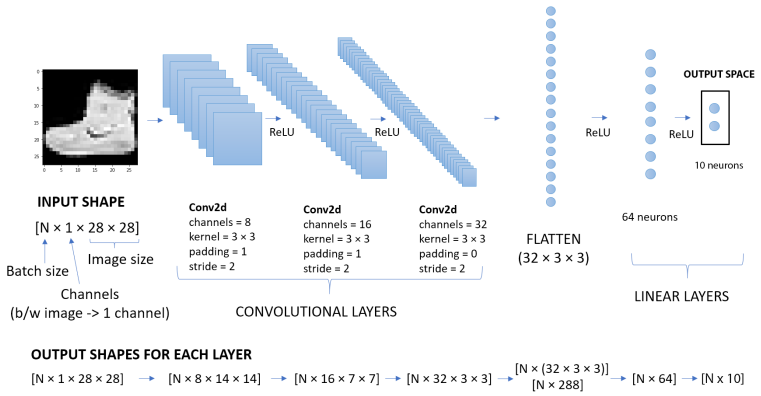


Figure 8: Employed CNN model.

validation losses versus the number of epochs. As said before, after training, if we give in input

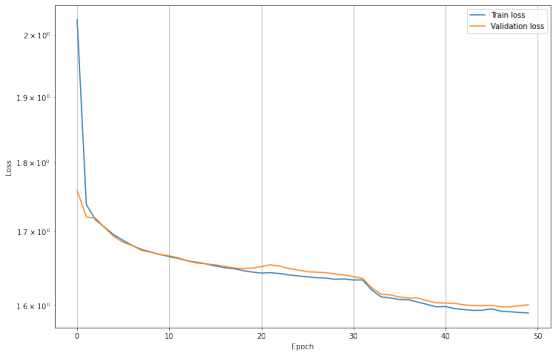


Figure 9: Training vs validation loss with Adam.

a test image to the CNN, the values on the output layer neurons are just the energies for the ten classes. The higher the energy for a class, the more the network "thinks" the input image is of that class. So, to obtain the network most likely guess, we just have to select the neuron with the highest energy. By doing so and comparing with the true label for each image in the test set, the model returns an accuracy of $\sim 85\%$.

2.1 Regularization

As in the regression task, we try first with a dropout rate of 0.25 and then with weight decay set to 10^{-4}. In both cases, the test accuracy is less than a percentage point higher.

2.2 Hyperparameters tuning

A grid search with k-fold cross validation – with k = 3 – has been implemented with the dictionary of parameters shown in Tab. 2:

Table 2: Hyperparameter dictionary

Hyperparameter	**Values**
Dropout	$[0, 0.25, 0.5]$
Weight decay	$[0, 10^{-5}, 10^{-4}]$
Batch size	$[300, 600]$

The optimal set of parameters turned out to be: [Batch size: 300, Dropout: 0, Weight decay: 10^{-4}]. The training and validation trend shown in Fig. 10 are better than before as well as the accuracy of the model, now at $\sim 86\%$.

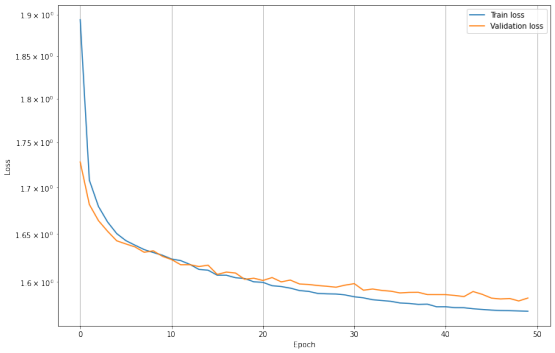
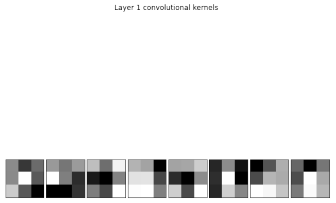


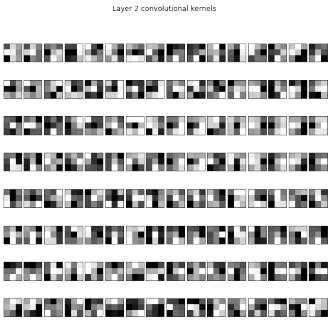
Figure 10: Training vs validation loss after grid search.

2.3 Visualize convolutional kernels

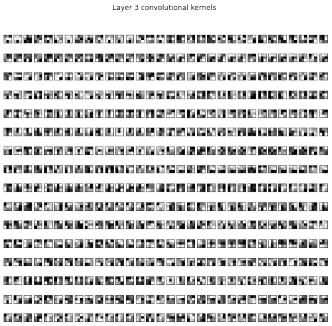
The feature maps for each convolutional layer are shown in Fig. 11.



(a) Convolutional kernels of the first hidden layer.



(b) Convolutional kernels of the second hidden layer.

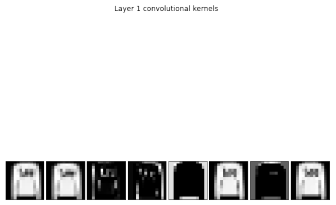


(c) Convolutional kernels of the third hidden layer.

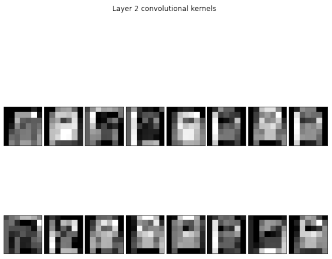
Figure 11

2.4 Visualize intermediate activation profiles

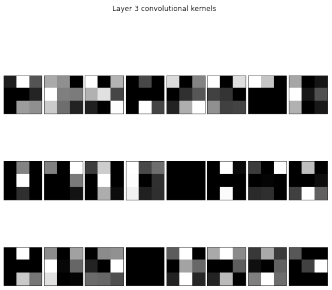
The activation profiles for each convolutional layer are shown in Fig. 12.



(a) Activation profile of the first hidden layer.



(b) Activation profile of the second hidden layer.



(c) Activation profile of the third hidden layer.

Figure 12