Assignment 2
Advanced Machine Learning
University of Milano-Bicocca
M.Sc. Data Science

Federico Signoretta
ID number: 847343
f.signoretta@campus.unimib.it
October 28, 2019

Prediction of gray-scale images

The assignment consists on the prediction of gray-scale images of letters *P-Z* through different implementation of ANN.

1 Supervised classification with a traditional neural network

In order to solve this supervised classification task, it was split the training set into training (70%) and validation set (30%) and it was implemented a neural network composed as follow:

- an input layer composed by 784 neurons (the images was 28x28);
- three hidden layer with activation function "relu": the first one composed by 256 neurons, the second one composed by 512 neurons and the last one with composed by 256 neurons. It was noted that symmetry of hidden layers improves the performance of the neural network;
- after each hidden layer, dropouts were inserted in order to reduce the overfitting of the NN;
- an output layer with 11 neurons (equal to the number of classes) and activation function "softmax".

Before training the neural network, it was needed to configure two important parameters: it was chosen "adadelta" as optimizer and "categorical cross-entropy" as loss function.
For training the model, it was set a batch size equal to 32 and epochs equal to 75: after several tests, it was observed that the model overfitted in any combinations of the above parameters. So, the necessity to insert some dropouts (as mentioned above) and an additional parameter called "EarlyStopping". The function "callback" with "EarlyStopping" as parameter, stops training when a monitored quantity (loss function on the validation set in this case) has stopped improving.

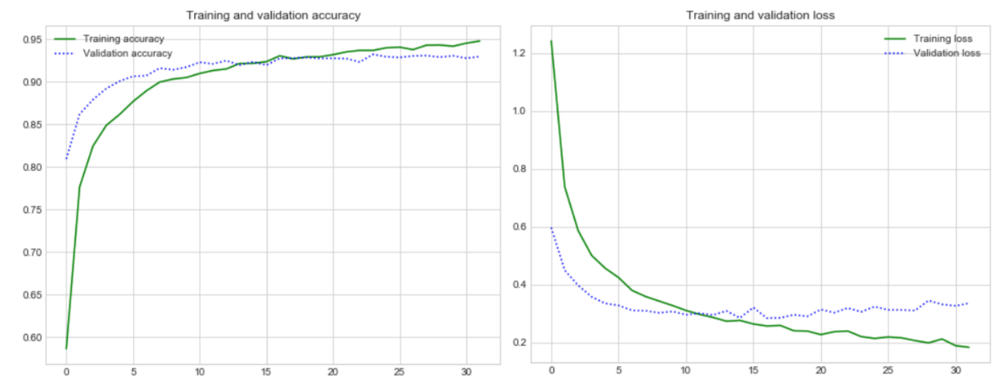

Figure 1: Accuracy and Loss

Finally, the above artificial neural network has produced the result on figure 1. In particular, there were obtained satisfactory results: the values of accuracy converge to about 0.95 and the values of loss function converge to values close to zero, both training and validation set.

2 Visual investigation of the reconstruction abilities of auto-encoder

In order to solve the visual investigation by using auto-encoder, it was implemented a neural network with three different parts: the encoder, the code and the decoder.

1. **encoder**: an input layer composed by 784 neurons (the images were 28x28) and two hidden layers with activation function "relu": the first one composed by 512 neurons and the second one composed by 256 neurons;
2. **code**: layer with activation function "relu" and number of neurons equal to 128;
3. **decoder**: two hidden layers with activation function "relu" the first one composed by 256 neurons and the second one composed by 512 neurons; an output layer with 11 neurons (equal to the number of classes) and activation function "sigmoid".

It was chosen "adam" as optimizer and "binary cross-entropy" as loss function. For training the model, it was set a batch size equal to 64, epochs equal to 50 and early stopping as above. The peculiarity of auto-encoders is an efficient data codings in an unsupervised manner. So, it was possible to train the model on the "X_train" and test it on the "X_test", with no target variable. In order to analyze the performance, it was necessary to graphically represent the results (figure 2). From figure 2 it was possible to observe that the model described above was

Figure 2: Original images and images from auto-encoder

able to reach very satisfied results.
After that, it was included some noise on the images in order to test again the performance of the auto-encoder. Setting a noisy coefficient equal to 0.3, the results are shown in figure 3. In

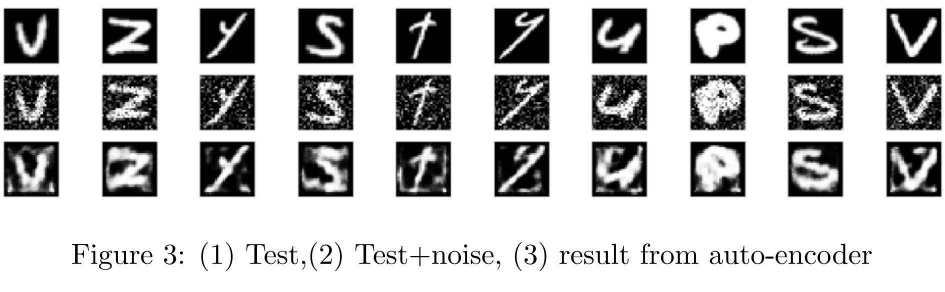

Figure 3: (1) Test,(2) Test+noise, (3) result from auto-encoder

general, it is possible to conclude that this model has good performance because the images created by the auto-encoder are quite similar to the real ones.

3 Use and evaluation of the encoded representation generated by the auto-encoder to solve the problem of supervised classification

In order to use the encoder in a new model, it was created a model with the same charateristics of the encoder of the above auto-encoder (input layer, an hidden layer with 512 neurons, an hidden layer with 256 neurons and a code with 128 neurons); after that, it was set "layer.trainable = False" in order to not train the parameters associated to these neurons. These parameters were set using the "set_weight" function in order to use exactly the same weight of the above auto-encoder. At this point, it was inserted a new hidden layer with 256 neurons with activation function "relu" and the usual output layer with "softmax" as activation function. In particular, on this hidden layer it was added a parameter called "regularizer". Regularizers allow to apply penalties on layer parameters or layer activity during optimization and in this case was set "kernel_regularizer=l2(0.01)".
For training the model, it was chosen "adam" as optimizer and "binary cross-entropy" as loss function. For training the model, it was set a batch size equal to 32, epochs equal to 50 and early stopping as above. The results are shown in figure 4.

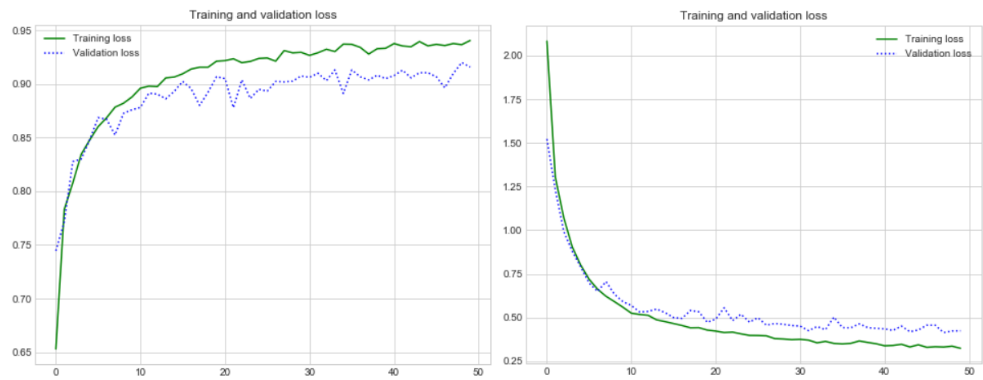

Figure 4: Accuracy and Loss

Concluding, the above artificial neural network with auto-encoder has produced the result showed on figure 4. In particular, there were obtained satisfactory results: the values of accuracy converge to about 0.95 and the values of loss function converge to values close to zero, both training and validation set. As well as in the first model implemented, there are not problems about over-fitting.