MNIST & CIFAR-10

David Barbas Rebollo

Artificial Neural Networks
January, 2021

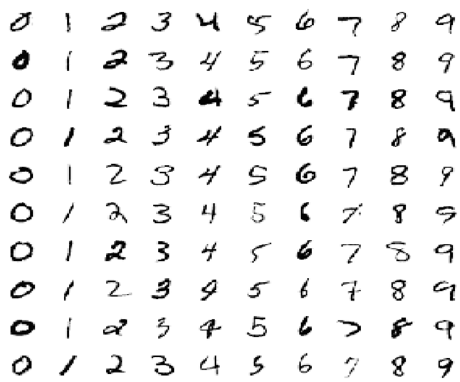
1 Introduction

The objective of this assigment is to solve 2 different classification problems using artificial neural networks, minimizing as much as possible the error obtained by the designed models.

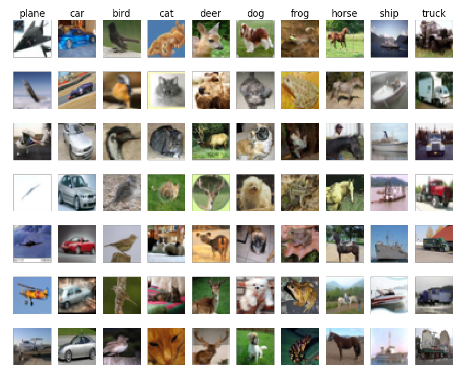
The first task is the recognition of digits between 0 and 9. The dataset used for this problem is MNIST, which consists of 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images.

The second task is the recognition of 10 different classes of images. The dataset used for this problem is CIFAR-10, which consists of 50,000 32x32 color training images and 10,000 test images, labeled over 10 categories.

The implementation of both solutions is done using *Jupyter Notebooks* and *Google Colaboratory*. This enabled to benefit from the use of GPUs and speed up the training of the classifiers.



(a) Example of the samples in the MNIST dataset.



(b) Example of the samples in the CIFAR-10 dataset.

2 MNIST and Multilayer Perceptron

In the first problem the images of the different digits a linearzed from a 28x28 2D image to a 784 value vector. The experiments were conducted with the model given as example, a batch size of 64 and the training of the network lasted 200 epochs.

The tuning only involved 2 different parameters. The first, is how the learning rate of the network varies during training. An alternative to the default version was tested, were the learning rate was scheduled to decay later from 0.1 to 0.01 on epoch 75 and to 0.01 on epoch 125, instead of epoch 25 and 50. Also, as a different callback was used, *ReduceLROnPlateau*. With this callback the learning rate reduces by half from 0.1 to 0.001 if there is no improvement in the loss value after 10 epochs. The second variation involves the data augmentation parameters. Apart from the default configuration, rotation of 10 degrees and image scaling of $[1.0 - 1.1]$ was introduced to have more alterations of the images.

The results obtained are shown in the table below:

Table 1: Results of experimentation for the MNIST task.

Model	LR Tuning	Data Augmentation	Accuracy (%)
Default	lr_scheduler	Default	99.35
Default	lr_reducer	Customized	99.36
Default	late lr_scheduler	Default	99.42
Default	late lr_scheduler	Customized	99.42

From the results obtained we can conclude that the complexity of the default network is complex enough to learn the features of each class. Additionally, the tuning the learning rate decay and data augmentation affects very little the accuracy in test. In fact, the gaussian noise, batch normalization layers and simple data augmentation are enough to avoid the overfitting of the network.

3 CIFAR-10 and Convolutional Networks

In the second problem the experiments again begin from the CNN given as example with training over 200 epochs and batch size of 128.

Give that this task was a harder problem, the learning rate scheduler and data augmentation were more refined. The learning rate was reduced more smoothly over training begging at 0.1 and reducing at the epoch 50, 75 and 150. The data augmentation considered not only the default setting (vertical shift, horizontal shift and horizontal flip), but zoom range, rotation and shear range. This is done to reduce over fitting and train a more robust network. The optimizer used is again SGD.

As shown in the example for this laboratory assigment, a auxiliary function was created to ease the creation of models without the max pooling. As described below the models stack convolutional layers.

The first experiment was inspired by the default model. The difference between them was using more filters in each convolutional layer, chaging the flatten of the last convolutional layer by a global average pooling layer and using less noise on the initial layers.

The second model evolved from the previos one. As its complexity increased while maintaining the changes commented earlier. The network increased its complexity by stacking 3 convolutional layers before each pooling layer.

The results obtained are shown in the table below:

Table 2: Results of experimentation for the CIFAR-10 task.

Model	Accuracy (%)
Default	87.64
Exp1	90.24
Exp2	92.63

As expected each subsequent experiment increases the model's performance. Initially, the network was not complex enough to learn correctly the different features of each class. The number of parameters reflect this fact as the second model created has 7 times more parameters than the default model.

4 Conclusion

This lab assignment has exposed the importance not only of the model but of other features. Such as, the evolution of the learning rate through training, the use of data augmentation, the use of gaussian noise and the use of batch normalization to reduce overfitting. Therefore, it is possible to conclude that without this techniques it would have not been possible to obtained these high accuracy values for these tasks.