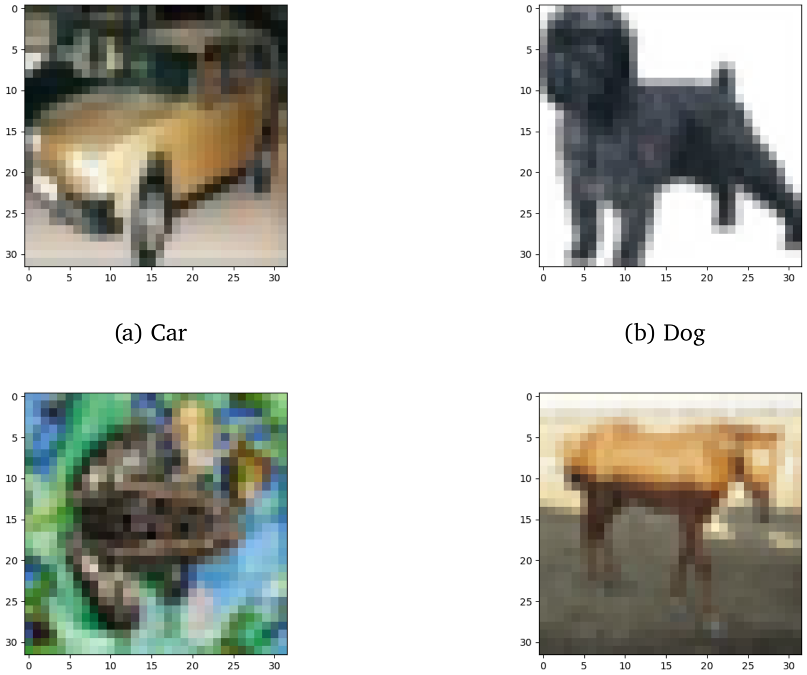
Deep Learning Lab: Report Assignment #2

Marco Latella
marco.latella@usi.ch

November 18, 2021

1 Dataset



(a) Car

(b) Dog

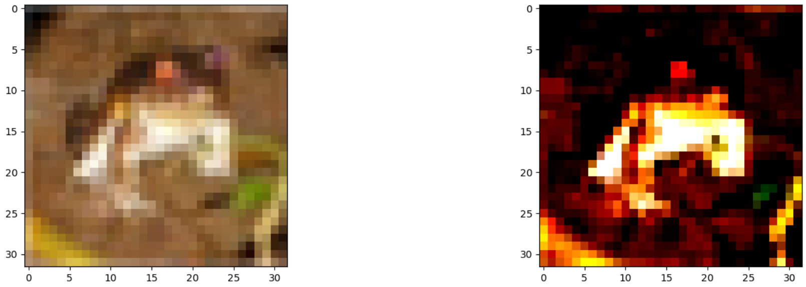
(c) Frog

(d) Horse

Figure 1: Train set images

Task 1 The CIFAR-10 dataset contains a total of 60000 images. About these images, 50000 are dedicated to the Training and 10000 images to the Testing. Above are reported some images taken by the Training set.

Task 2 To normalize the values of each image of the dataset, I first calculate the mean and the standard deviation of the values of the images, and then I passed these values to the Normalize function. The mean and Standard deviation obtained from Training set are: (0.4914, 0.4822, 0.4465), (0.247, 0.243, 0.261). Below is shown an image before and after Normalization.



(a) Before

(b) After

Figure 2: Difference with Normalization

Task 3 The validation set has been created according to the directives given. The last 1000 images of the Training set have been selected to create the Validation dataset. It has been used two lists of indexes passed to the samplers as an argument. Lastly, the two samplers have been given to the Dataloader function.

2 Model

Task 1 The model has been defined using the Sequential Class. In this way, I could "separate" logically the layers of the Convolutional Neural Network.

3 Training

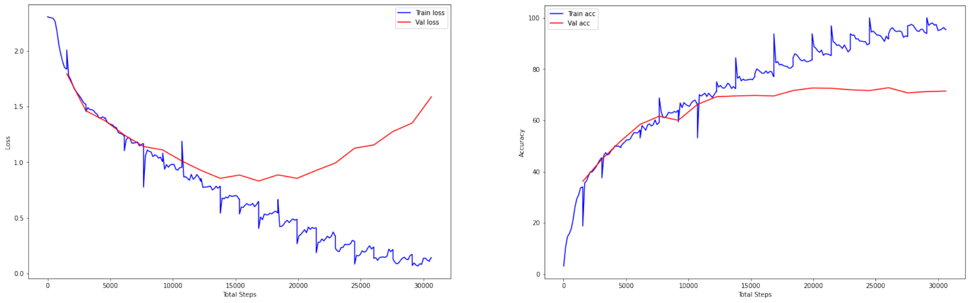
Task 1 The training pipeline has been implemented with the monitoring of the current training loss and accuracy every 150 steps. The validation loss has been monitored every epoch as required. I chose 150 steps to have a fair trade-off between little monitored data and a lot of monitored data.

Task 2 As required I trained the model with the following hyper-parameters:

- SGD optimizer, Learning rate: 0.001, Momentum: 0.9
- Batch size: 32
- Epochs: 20

Task 3 After training the model with the hyper-parameters above specified it reach a final validation accuracy of 72,7% at epoch 17. However we have to consider that training the model more times, with the same hyper-parameters, we obtain different values of accuracy that go, more or less, from 69% to 73%.

Task 4 Looking at the Losses comparison figure you can see that the model is overfitting starting, more or less, from the sixth epoch. On the other hand, the validation accuracy stops to improve meaning that the model doesn't miss the prediction but guesses them with less confidence. A behavior to explain are the peaks in the losses and accuracies curves. This happens because the first step of each epoch is not mediated over 150*32 images but only over 32.

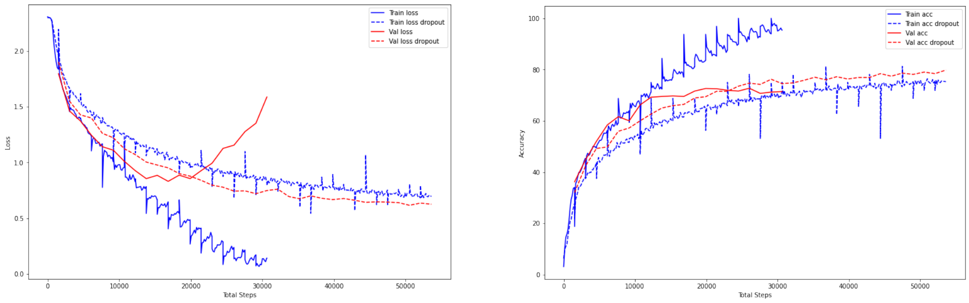


(a) Loss comparison

(b) Acc comparison

Figure 3: Evolution of losses and accuracies

Task 5 Applying the Dropout regularization, the Overfitting problem is resolved. This happens because by adding Dropout you remove, according to a certain probability, some connections among nodes of the Network simplifying the complexity of the model. Respect the previous run I increased the epochs number to allow the model to converge. Another behavior you can notice is that the validation loss is better compared to training one. This could mean that the validation set is easier to predict with respect to the training set probably because is widely represented by the training set.

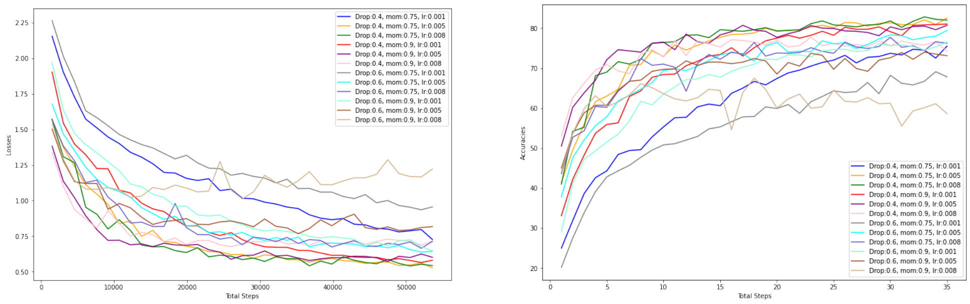


(a) Loss comparison

(b) Acc comparison

Figure 4: Comparison with and without Dropout

Task 6 To find the best Hyper-parameters I launched an execution that try twelve different combinations of Hyper-parameters, changing the Learning rate, the Momentum, and the Dropout probability. All the combinations are considerd on 35 epochs.



(a) Loss comparison

(b) Acc comparison

Figure 5: Combination of Hyper-parameters

As shown in the figures above there are several combinations that make the respective models the bests ones. However there are some model that doesn't reach even the 70% of accuracy. There are two model that seams to be the bests.
Model 2: Learning rate: 0.008, Momentum: 0.75, Dropout: 0.4
Model 3: Learning rate: 0.005, Momentum: 0.75, Dropout: 0.4

Following are shown the bests Accuracies of all models with the corresponding epoch:

Model	Best Accuracy	Epoch	Dropout	Momentum	Learning Rate
1	75.5	35	0.4	0.75	0.001
2	82.5	35	0.4	0.75	0.005
3	82.8	33	0.4	0.75	0.008
4	81.0	35	0.4	0.9	0.001
5	80.7	35	0.4	0.9	0.005
6	77.7	23	0.4	0.9	0.008
7	69.1	34	0.6	0.75	0.001
8	79.4	35	0.6	0.75	0.005
9	77.7	30	0.6	0.75	0.008
10	76.5	35	0.6	0.9	0.001
11	74.0	33	0.6	0.9	0.005
12	67.5	18	0.6	0.9	0.008

Task 7 Using the two bests models of task 6 to predict images on the Test set I obtained the following Accuracies and Error rates:
Model 2: Test accuracy: 80.71 % Test error rate: 19.29 %
Model 3: Test accuracy: 80.76 % Test error rate: 19.24 %

4 Questions

Answer 1 Softmax Layer applies a softmax activation function to the input of the layer. It is used as the last layer to represent a probability distribution over a certain number of classes.

Answer 2 The Momentum parameter allows accelerating learning while using the Stochastic Gradient Descent optimizer considering more the previous gradients. Using a zero value of momentum you are performing a SGD without further benefits.

Answer 3 The main difference is that the CNN layer treats weights as shared thanks the kernel, instead the Fully Connected layer consider a weight for each pixel for each channel. Is a good idea for processing images because in this way can be considered information contained in closer pixels.

Answer 4 Two different type of data that could be used with 1D Convolution are Time Series data and Text data.

Answer 5 Dropout allows skipping a few connections between nodes of the Convolutional Neural Network according to a defined probability. This usually will solve the Overfitting problem simplifying the model complexity.