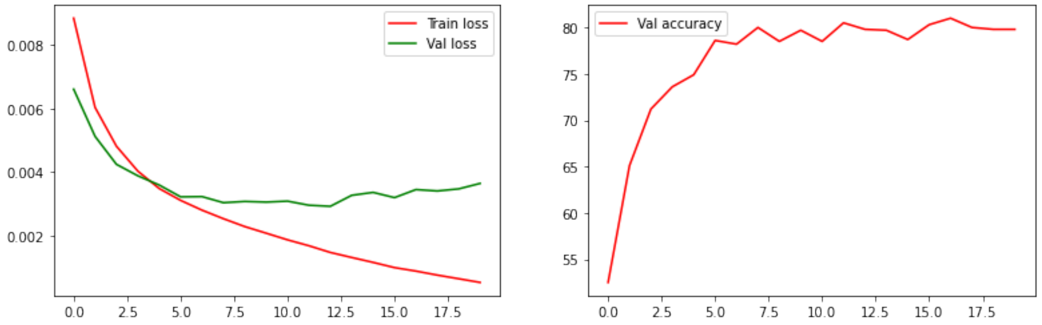
Advanced Machine Learning
Exercise 3
Convolutional Networks

Abbonato Diletta, Pezone Francesco, Testa Lucia

6 May 2020

1 Question 1

1.a



(a) Loss on train and validation

(b) Validation accuracy

In this picture we can observe the behaviour of train and validation losses and the validation accuracy for the first training session.

In this session we implemented the network using Conv2d, MaxPooling and Relu, constructing 5 final blocks and at the end a sequential layer to have the final output with probabilities.

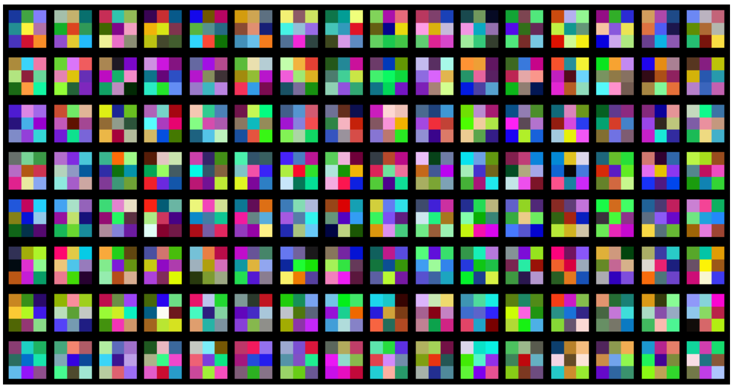
In this frame the total epochs have been 20, for a final result of 79.8 for the validation accuracy.

1.b

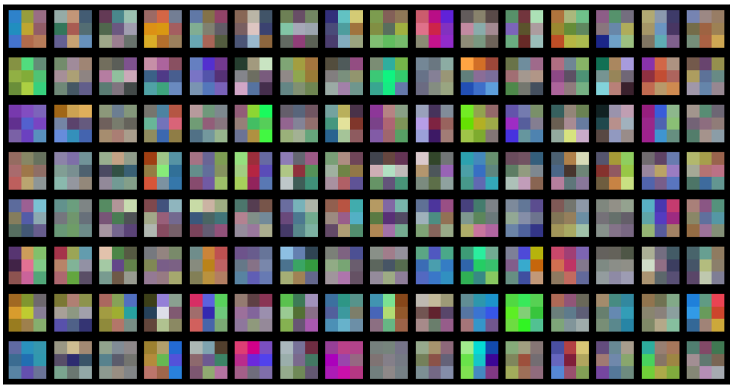
When we talk about model capacity we consider the amout of weights and bias for our model. In this case the value of trainable parameters is 7678474. This huge number is due to the fact that we have 128-512 for the size of in-channels of layers.

1.c

As we can see in the two images, at the beginning the filters have no pattern, they are random RGB values; after 22 epochs we start to see many filters with verticals lines or orizontal lines. Moreover now the green and blue colors tends to be aggregated, not dispersed as before training, this is due to the fact that grass and sky/water are present in many images.



(c) Filters at the beginning



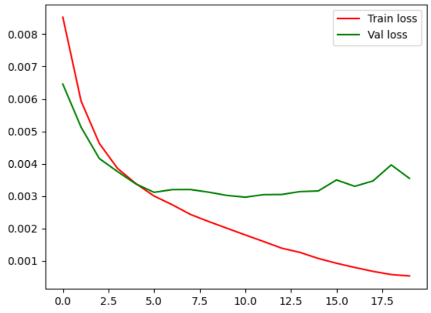
(d) Filters after 22 epochs

2 Question 2

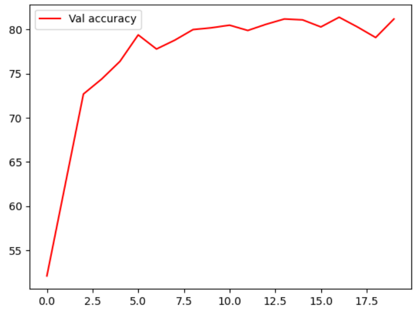
2.a

Batch Normalization was introduced to distribute the data uniformly across a mean that the network sees best, before squashing it by the activation function.

Without the BN, the activations could over or undershoot, depending on the squashing function though.

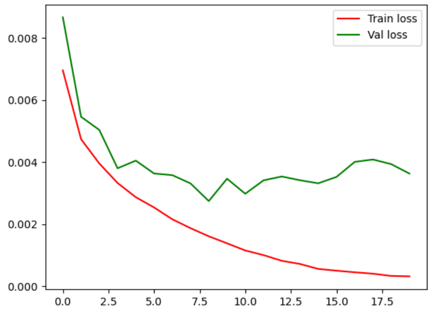


(e) Loss on train and validation without BN

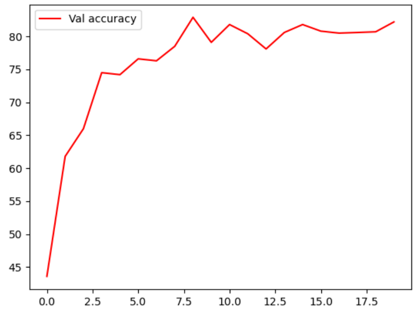


(f) Validation accuracy without BN

The curves of the loss seem to be better with the BN and the same trend can be observed in the validation accuracy where a much higher accuracy is obtained already from the initial epochs. Hence, even in practice, BN gives better performance.



(g) Loss on train and validation with BN



(h) Validation accuracy with BN

2.b

To analyze the early stopping results we executed five times both model with and without batch normalization.

We can notice that for the first case, the standard one we have an early stopping around the 20th epoch.

For the case of the model with batch normalization the early stopping appears later aroud the 30th epoch.

Here the table of results for the execution without batch normalization:

	Epochs	Best loss	Best epoch	Final model accuracy	Best model accuracy	Test accuracy
0	50	0.00312	19	77,50%	81.0%	79,7%
1	50	0.00287	15	82.1%	82.2%	79,7%
2	50	0.00286	20	81.1%	82.0%	79,0%
3	50	0.00327	24	78.3%	79.2%	77.9 %
4	50	0.00311	24	80.1 %	80.5 %	79.8 %

Here the table of results for the execution with batch normalization:

	Epochs	Best loss	Best epoch	Final model accuracy	Best model accuracy	Test accuracy
0	50	0.00303	34	84.1%	84.20%	83.5 %
1	50	0.00261	35	83.6 %	85.1%	84.3 %
2	50	0.00269	34	82.6 %	84.00%	83.8%
3	50	0.00282	39	85.0%	85.4%	83.8%
4	50	0.00302	36	82.8%	84.1%	84.1%

3 Question 3

3.a

As we can see in the following plots the best performances, after 40 epochs when we apply a single transformation refered to the performances with only the batch normalization (the lightgray lines), are given by horizontal flip with probability p=0.5.
Even perspective with distortion_scale=0.5, p=0.5, interpolation=3, fill=0 give, if we allpy the early stopping procedure, at epoch 29 a better result..
Moreover the good result is given even wen we applied both horizontal flip and perspective; after 40 epochs the accuracy is still going up arround 87%.
Two interesting trasforamtions are the gray scale and the colorjitter, both of which works on colors. We can see that after 40 epochs there are no clear signs of improvement in validation loss and accuracy, this could be due to the fact that for the identification colors play an importan role and wen we apply gray scale we are going to add new data without adding new information, the same for the colorjutter with the brightness since our dataset contain clear images.
All the others trasformation, or combination, show clear signs indicating their propensity to increase for accuracy and decrease for validation loss, this propensity could allow, within 30 epochs, to improve the performance.

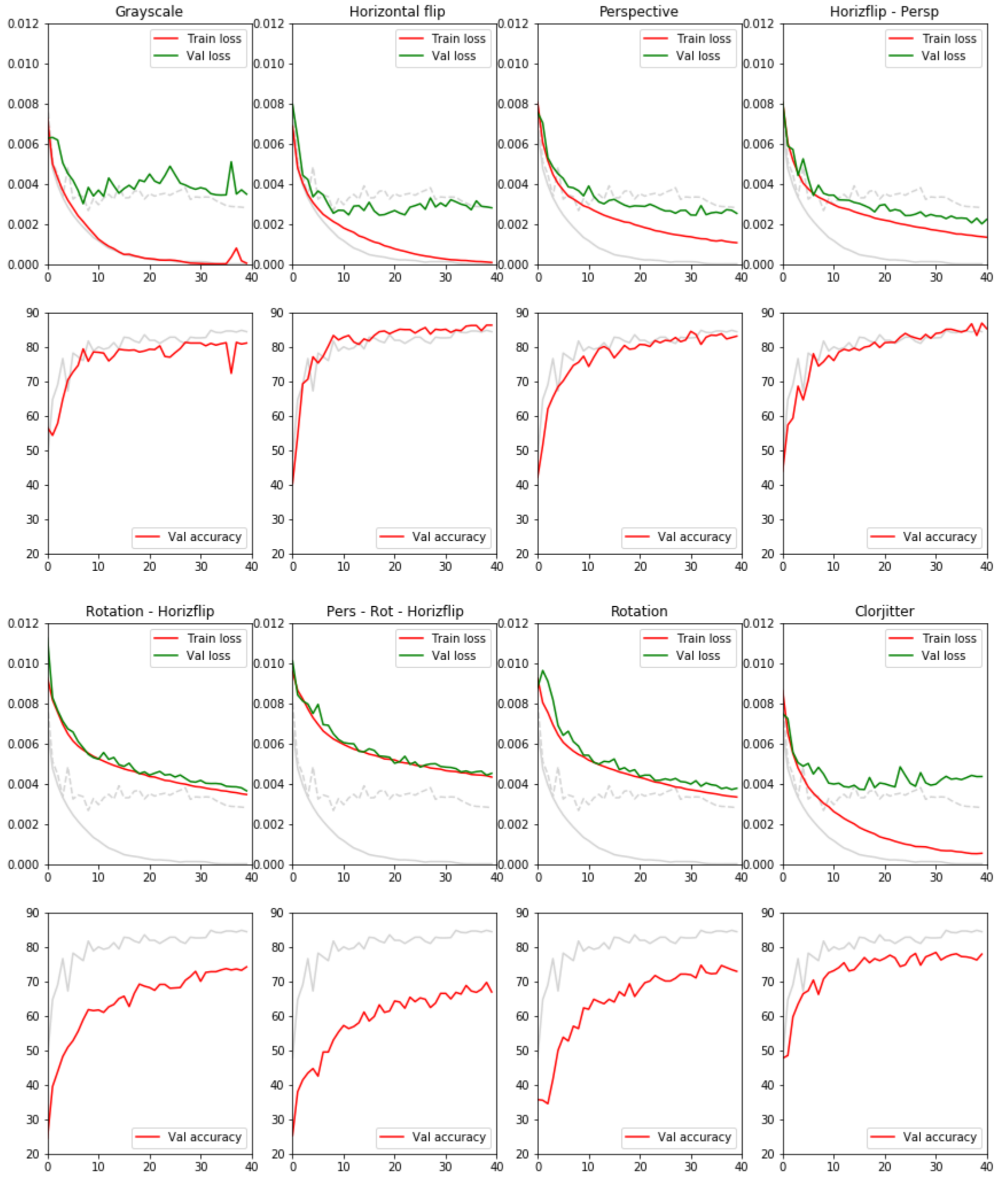
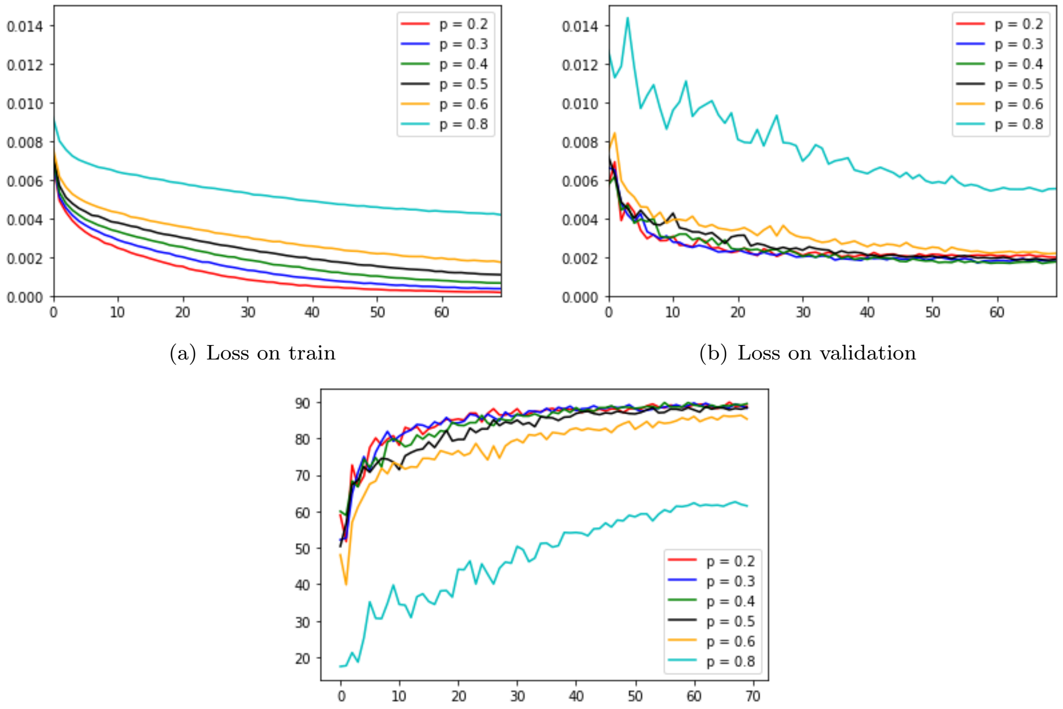


Figure 1: Loss and accuracy for data agumentation

3.b

In this part we will use the CNN with batch normalization and drop out.
As we can see from the graphs, for $p = 0.6$ the performances start to deteriorate a lot, for higher values both validation loss and accuracy are very bad.
The optimal value is $p = 0.4$, the green line, since as the number of epoch increases the validation loss and accuracy are optimal, w.r.t. the other values of p; in fact at the beginning there are good performances with small values of p, but after some epoch the validation loss start to encrease.



(a) Loss on train

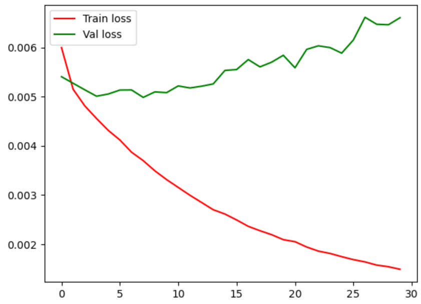
(b) Loss on validation

(c) Validation accuracy

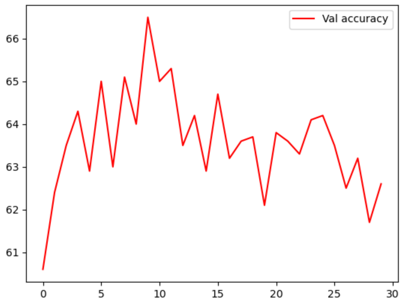
4 Question 4

4.a

After having carried out the train with **VGG_11_bn** on the whole network for the **CIFAR-10** dataset, the results obtained show a validation accuracy on the network of **61.1%** and despite the various changes in the validation accuracy, there do not seem to be variations in the loss.



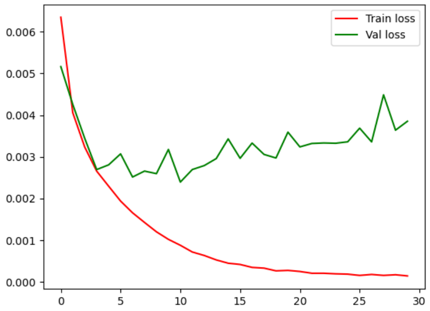
(d) Loss on train and validation



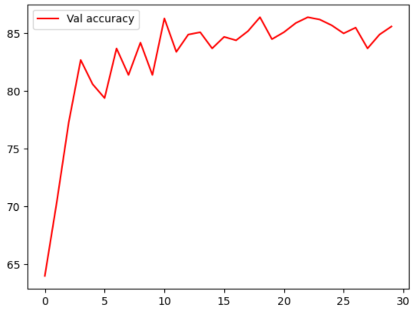
(e) Validation accuracy

4.b

To improve the performance, as requested, we fine-tune the whole network on the **CIFAR-10** dataset, starting from the ImageNet initialization.

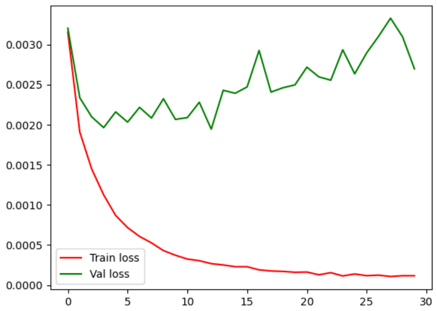


(f) Loss on train and validation

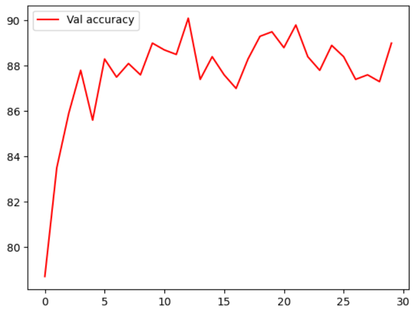


(g) Validation accuracy

The results obtained indicate that the performance of the network has improved to **86.1%** accuracy despite being trained from scratch.



(h) Loss on train and validation



(i) Validation accuracy

Repeating the analysis by setting the pretrained variable to true gives even better results with **89%** accuracy.