Deep Learning Lab
Università della Svizzera Italiana
Faculty of Informatics
November 7, 2019

Assignment 2: Convolutional Neural Network

Giorgia Adorni (giorgia.adorni@usi.ch)

1 Introduction

The scope of this project is to implement a convolutional neural network to classify the images in the CIFAR-10 dataset.

First of all, the original training set has been shuffled and divided into train and validation sets, with 49000 and 1000 images respectively. A seed has been used to reproduce the same sample split and use them in the different models. Instead, the test set provided contains 10000 images.

A certain preprocessing has been applied to the data. The pixel values of each sample, initially comprised between 0 and 255, have been rescaled between 0 and 1. To represent the class assignments, which were integers between 0 and 9, three binary assignment matrices have been created, one for each set of data.

The architecture of the convolutional neural network follows the instructions provided, as well as the hyper-parameter values for the models presented in Sections 2 and 3. In the training phase, mini-es were used. In particular, each epoch splits the training set in different samples of data.

All the models were implemented using `TensorFlow` and trained on an NVIDIA Tesla V100-PCIE-16GB GPU.

Note: Since mini-batches are used during the training of the models, the loss and the accuracy of all samples are averaged, obtaining a less noisy estimate.

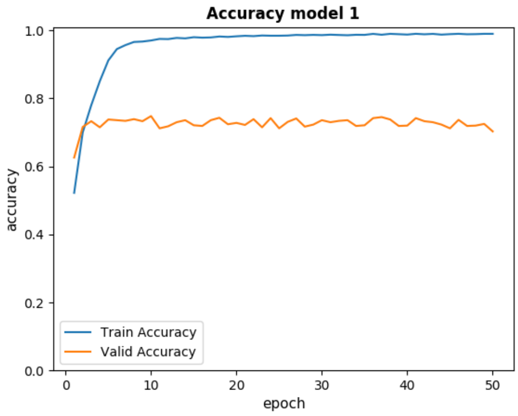
2 Performance of the initial model

In Table 1 is summarised the architecture of the network used in the first experiment.

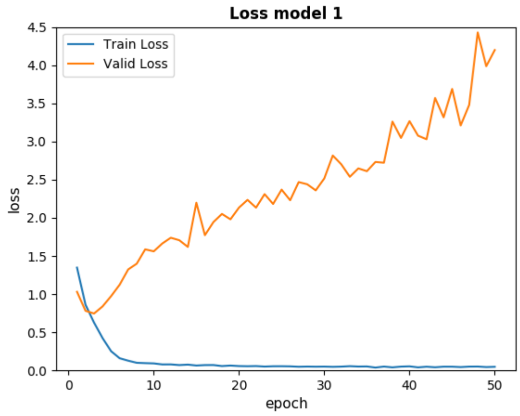
conv1	**conv2**	**mpool1**	**conv3**	**conv4**	**mpool1**	**fc**	**softmax**
3×3, 32 s. 2×2 p. same	3×3, 32 s. 2×2 p. same	2×2 s. 1×1 p. same	3×3, 64 s. 1×1 p. same	3×3, 64 s. 2×2 p. same	2×2 s. 2×2 p. same	512	10

Table 1: Network architecture

The model is trained for 50 epochs and `Adam` is used as optimiser with learning rate $0,001$. As loss function, the Softmax Cross Entropy with Logits is used since the model is a multi-class classifier. Moreover, once per epoch, is documented the classification accuracy on both the train and validation set. The performance is shown in Figure 1.



(a)



(b)

Figure 1: Training and validation curves in the initial model

As can be seen in Figure (a), the training accuracy rapidly grows up to 100%, while the validation accuracy remains stable at 70%. The final accuracy on the valid set is 70.30%. For what concerns the loss, it is clear that the model overfits the data. The training loss is close to 0 while the validation one diverges.

	Accuracy		**Loss**		**Train Time**
	Train	Validation	Train	Validation	
Model 1	98.99%	70.30%	0.05	4.20	406 sec

Table 2: Model performances

For this reason, in Section 3 is presented a new model that has the aim of improving these results.

3 Regularisation of the model with dropout

The model proposed in this section involve the use of a model regularisation technique, that is the addition of a dropout layer after each max-pooling and fully-connected layer. In particular, during the training phase, the probability to keep each neuron is set to 0.5, while in the validation set should be 1.

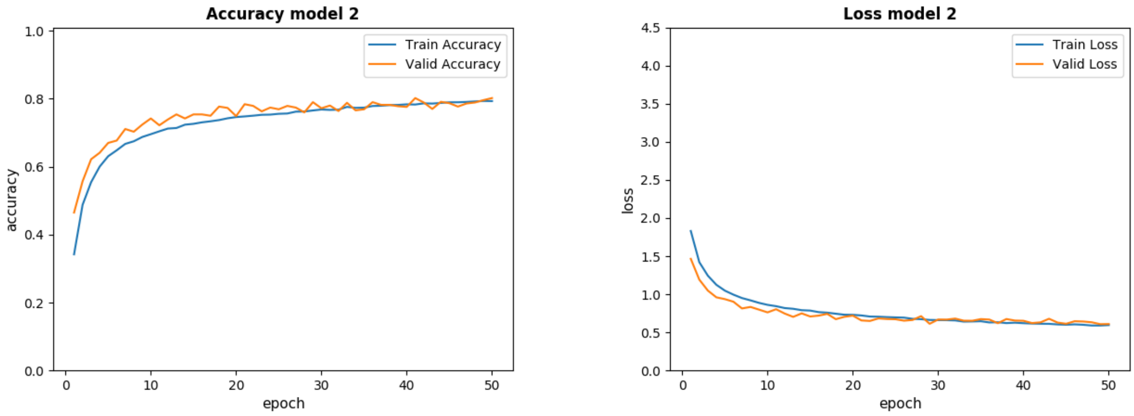
The architecture of the new network is presented in Table 3.

conv1	conv2	mpool1	conv3	conv4	mpool2	fc	softmax
3×3, 32 s. 2×2 p. same	3×3, 32 s. 2×2 p. same	2×2 s. 1×1 p. same dropout	3×3, 64 s. 1×1 p. same	3×3, 64 s. 2×2 p. same	2×2 s. 2×2 p. same dropout	512 dropout	10

Table 3: Network architecture

The actual performance is shown in Figure 2. The performances legitimately improved as expected after the application of the dropout.

Even if the performance on the training worsen respect to the previous model, since the regularisation is applied only on this set, the performance on the validation increased by 9.90%. Now its value is 80.20%, significantly better compared to the previous model.



(a) (b)

Figure 2: Training and validation curves in the regularised model

Instead, observing the loss curves, after the 30th epochs it is possible to see some signs of overfitting.

	Accuracy		Loss		Train Time
	Train	Validation	Train	Validation	
Model 2	79.32%	80.20%	0.60	0.61	415 sec

Table 4: Model performances

In the following section will be attempted some additional experiments by modifying the model's hyperparameters.

4 Hyperparameter settings

In this section will be discussed 6 different additional configurations for the hyperparameters of the network, to improve the validation accuracy. In particular, will be documented the performances according to the modification of the following hyperparameters: learning rate, mini-batch size, dropout and number of epochs. In this analysis, only the models with the best performances will be included. In Table 5 are documented the different hyperparameter set for the models.

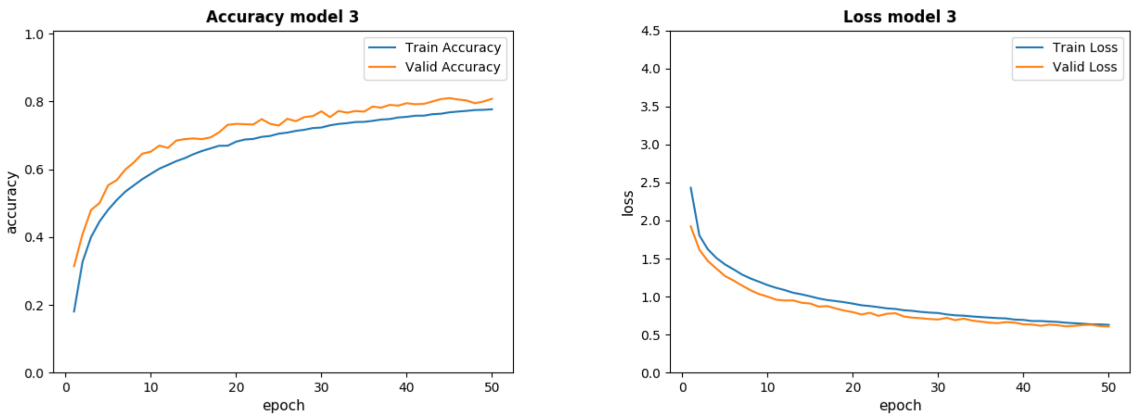
	learning rate	batch size	dropout
Model 1	1e-3	32	-
Model 2	1e-3	32	0.5
Model 3	1e-4	32	0.5
Model 4	1e-3	128	0.6
Model 5	1e-3	128	0.5
Model 6	1e-4	128	0.5
Model 9	1e-3	128	0.25 - 0.5

Table 5: Model hyperparameters

All the models are trained for 50 epochs. The other configurations tested, also those that include the modification of other hyperparameters, for example using the Gradient Descent as optimiser or adding the decay rate and other parameters to Adam optimizer will not be presented in this report due to their mediocre results.

The first experiment performed simply consists in the reduction of the learning rate from 0.001 down to 0.0001.

In Figure 3 are visualised the performances of the model. The accuracy, which now measures 80.80%, increased only by 0.60% compared to the previous model. Furthermore, it is visible a distance between the training and validation curves. The loss curve rapidly decreases towards 0, and there are no signs of overfitting.

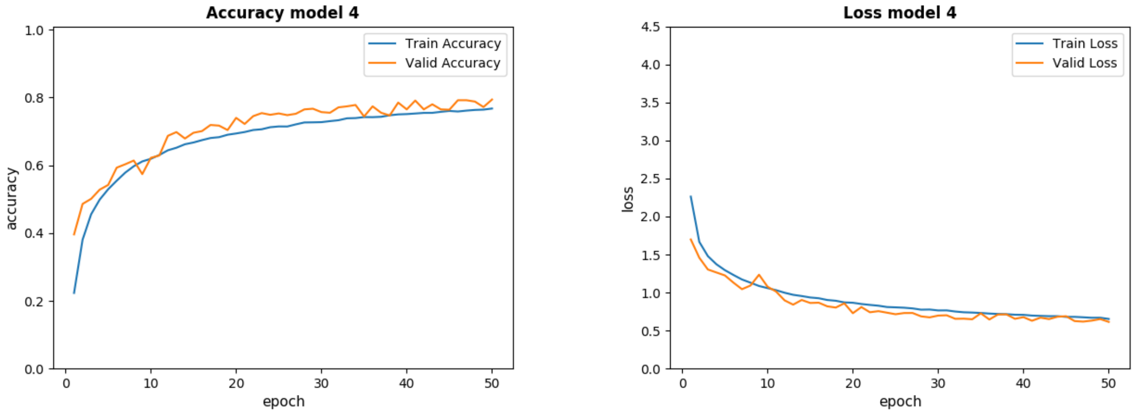


(a) (b)

Figure 3: Training and validation curves in model 3

In Section 4.2 will be discussed a further modification to this model that consists of increasing the number of epochs, since the trend of the curve seems to be growing.

In Model 4, the training has been carried out using 128 samples for batch instead of 32 and the learning rate has been restored to its original value of 0.001. Furthermore, the dropout value is updated: the probability to keep each neuron is increased to 0.6.

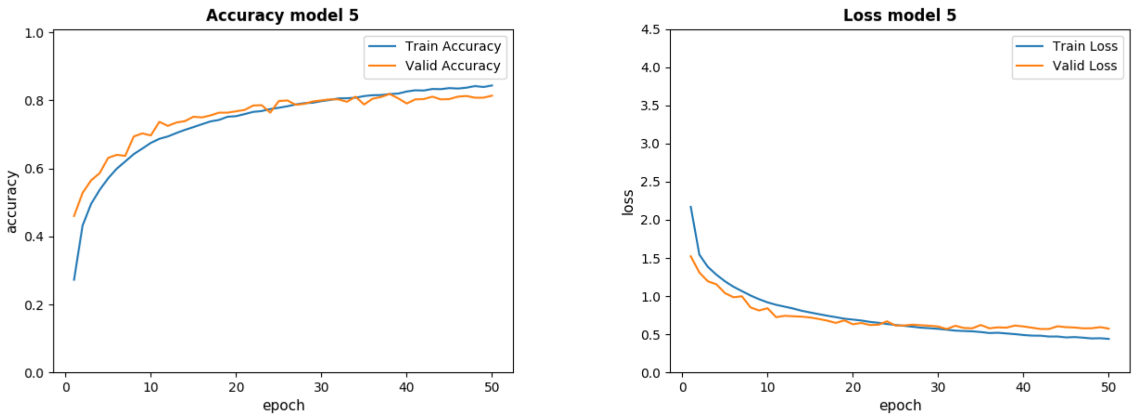


(a) (b)

Figure 4: Training and validation curves in model 4

In this case, the performance is very similar to the previous experiment, even if slightly worse. The validation accuracy is reduced to 79.4% despite the validation loss is still the same, as can be seen in Figure 5.

In Model 5, the number of samples for batches and the learning rate are kept equal to the previous experiment, while the dropout has been restored to its original value. The current validation accuracy has begun to rise again, reaching 81.4% and improving the performance of model 3. However, as it can be seen in Figure 4, the validation curve, initially promising, after the 30th epoch it falls below the training curve.



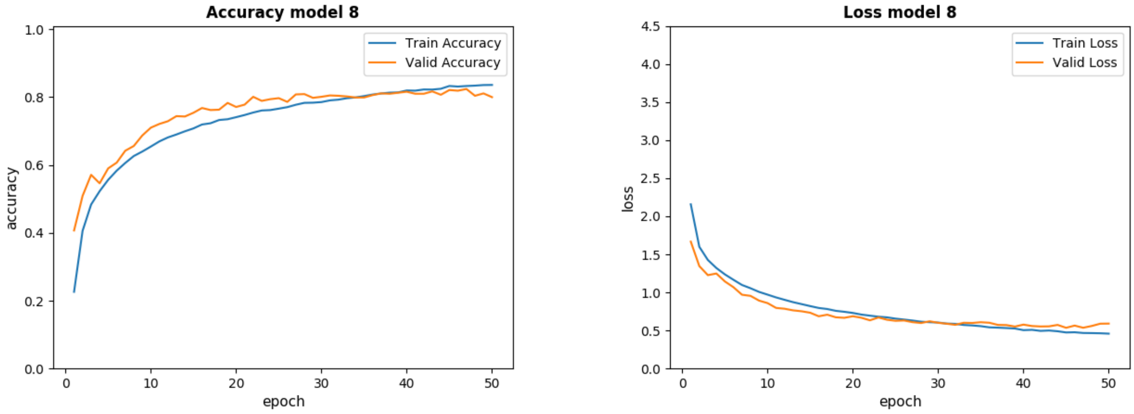
(a) (b)

Figure 5: Training and validation curves in model 5

For the next two experiments, it was decided to set the learning rate to the previous value of 0.0001 and restore the dropout to his original value of 0.5. Instead, the number of examples in the batch for model 6 has been kept equal to the previous experiments, that is 128 and increase up to 256 for model 7.

Unfortunately, in both cases, the performance has deteriorated compared to all the previous performances. The first model has a validation accuracy of 73.50%, while the second fell to 68.40%, that is the worst result measured so far.

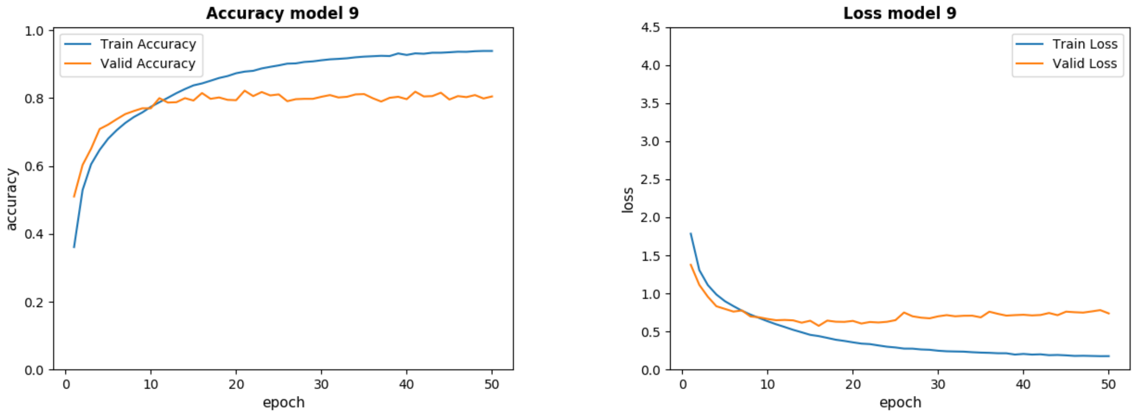
For this reason, it was decided to retrain the model keeping the learning rate value at 0.001 and the number of examples per batch at 256. As shown in Figure 6, the final validation accuracy is 80.00%, which is a nice result.



(a) (b)

Figure 6: Training and validation curves in model 8

In the last experiment, a further attempted modification foresees in the use of different dropouts based on its application after a max-pooling or a fully connected layer. In particular, the rate of the dropout is set to 0.25 after the max-pooling layers and kept to 0.5 after the fully connected layer.



(a) (b)

Figure 7: Training and validation curves in model 9

The performances of the model are shown in Figure 7. The validation accuracy obtained, that is 80.5%, is among the best three.

In Table 6 are summarised the performance of the presented models. Among the documented model, the best achieved are model 5, model 3 and model 2. In the following section will be presented the performances of some models after the increase of training epochs.

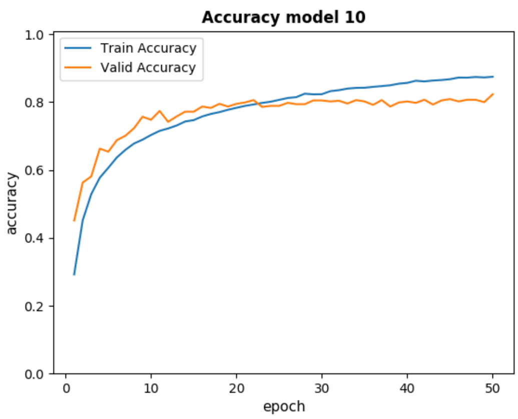
	Accuracy		Loss		Train Time
	Train	Validation	Train	Validation	
Model 1	98.99%	70.30%	0.05	4.20	406 sec
Model 2	79.32%	80.20%	0.60	0.61	415 sec
Model 3	77.70%	**80.80%**	0.63	0.61	445 sec
Model 4	76.75%	79.40%	0.65	0.61	148 sec
Model 5	84.39%	**81.40%**	0.44	0.57	185 sec
Model 6	71.03%	73.50%	0.82	0.77	149 sec
Model 7	65.37%	68.40%	0.98	0.88	119 sec
Model 8	83.60%	80.00%	0.46	0.59	120 sec
Model 9	93.91%	**80.50%**	0.18	0.73	155 sec

Table 6: Model performances

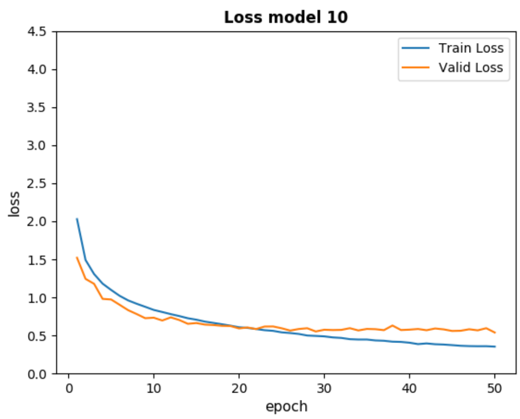
4.1 Modification of the number of hidden units

In this section will be discussed modification to the current best model, which is the model 5, that consists in an increase of the number of hidden units of the fully connected layer from 512 up to 1024.

Since the learning curve of the network, shown in the previous section, seems to be underfitting, increasing the complexity of the network should be a good idea. The new performances are shown in Figure 8.



(a)



(b)

Figure 8: Training and validation curves in model 10

The previous validation accuracy, that was 81.40%, increases by 0.90%, reaching 82.3%.

Further modifications to this model will be discussed in Section 4.3.

	Accuracy		Loss		Train Time
	Train	Validation	Train	Validation	
Model 10	87.47%	**82.30%**	0.35	0.54	161 sec

Table 7: Model 10 performances

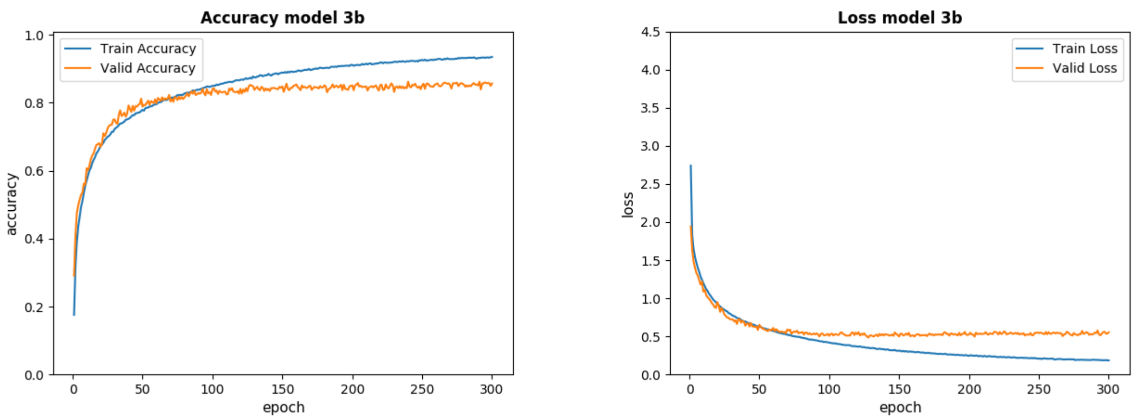
4.2 Increasing the number of epochs

In this section are analysed the performances of the previous models trained for a greater number of epochs.

These experiments have been carried out only on models 3, 4, 5, 8 and 9 since they shown better accuracy values.

The models are trained on GPU which allows training without worsening the time too much.

The first experiment were performed on model 3, which curves are shown in Figure 9.

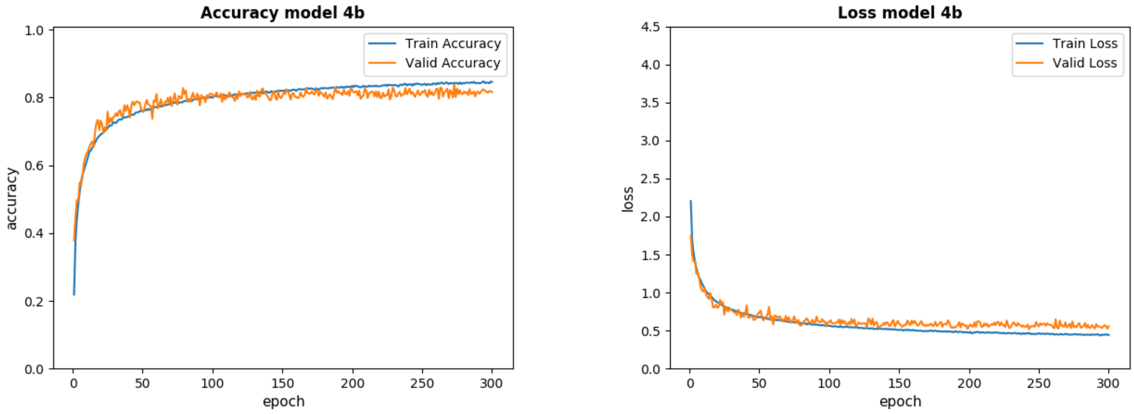


(a) (b)

Figure 9: Training and validation curves in model 3b

Compared to the previous, this model has improved the validation accuracy of 4.9%, reaching 85.7%, which is the highest result achieved so far.

The following experiments was carried out on model 4.

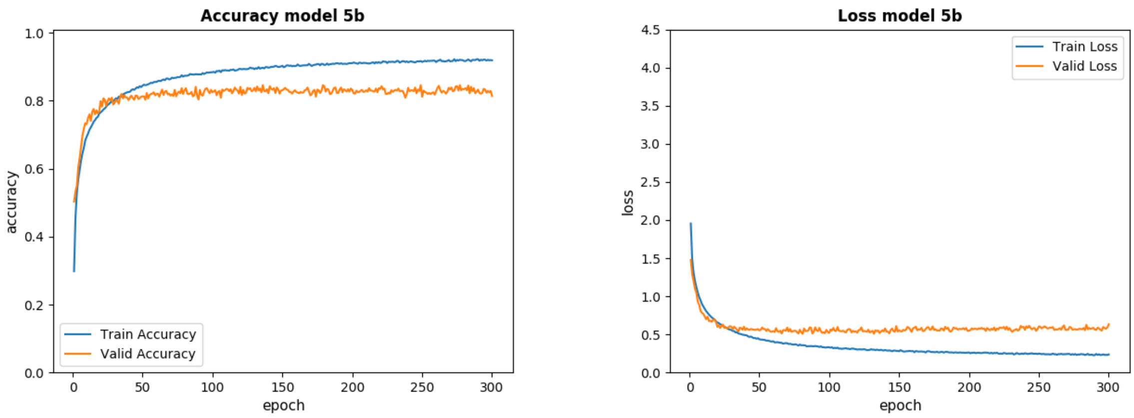


(a) (b)

Figure 10: Training and validation curves in model 4b

The results obtained, shown in Figure 10, are worse than those obtained from the previous model. The validation accuracy is fallen to 81.60%. In this case, the validation accuracy obtained is more or less the same as the one achieved with only 50 epochs.

Continuing, the results for the model 5b are shown in Figure 11. Despite the performances of the model on 50 epochs were the best, this model obtains exactly the same results, hence it is no longer the best among the proposed models.



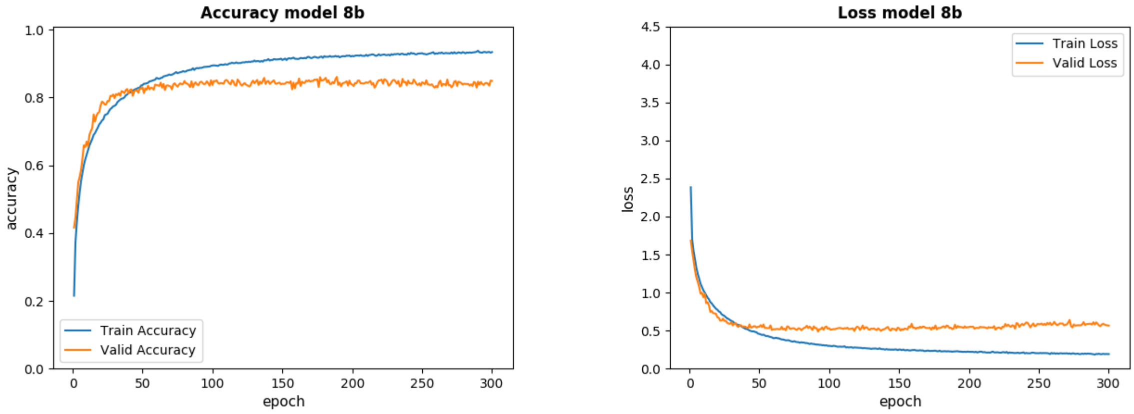
(a) (b)

Figure 11: Training and validation curves in model 5b

In fact, looking at the two curves, it is clearly visible that after few epochs the model stabilizes and stops improving its performance. This result was predictable since the learning curve seemed to have already stabilized before 50 epochs.

The next experiment was carried out on model 8, which presented a further increasing equal to 0.40% of the performance respect to the last experiment.

The situation presented, in particular, the trend of the train and validation accuracy curve, shown in Figure 12, is very similar to the one which occurred with model 5. This is because, after a certain epoch, both the model have stopped or significantly slowed down

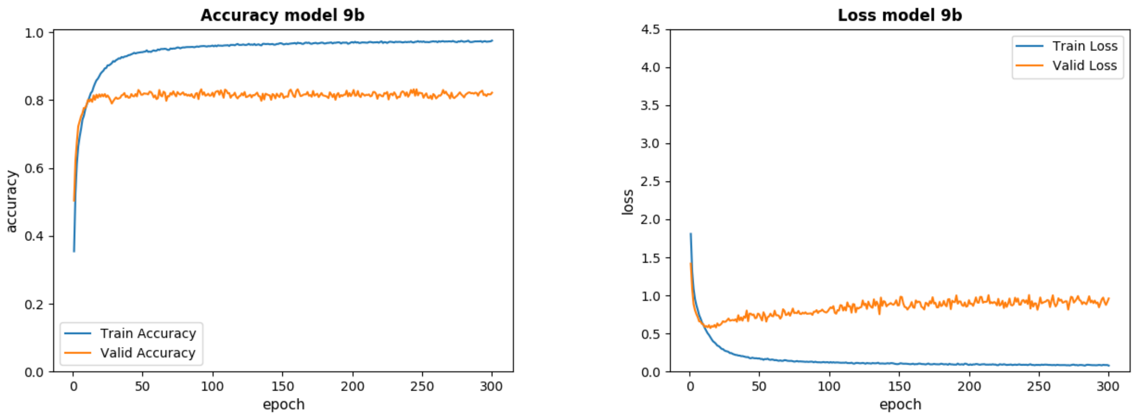


(a) (b)

Figure 12: Training and validation curves in model 8b

the learning.

The last experiment was performed on model 9. Despite the performance is better than the latest, it does not reveal a significant increase in performance compared to the latest and, above all, respect to model 3b.



(a) (b)

Figure 13: Training and validation curves in model 9b

The current validation accuracy is equal to 82.20%. The curves are displayed in Figure 13.

In Table 8 are summarised the performance of the presented models. Among these, the bests are the model 3b, model 8b and model 9b.

	Accuracy		Loss		Train Time
	Train	Validation	Train	Validation	
Model 3	77.70%	80.80%	0.63	0.61	445 sec
Model 3b	93.47%	**85.70%**	0.18	0.55	2974 sec
Model 4	76.75%	79.40%	0.65	0.61	148 sec
Model 4b	84.60%	**81.60%**	0.44	0.56	1040 sec
Model 5	84.39%	81.40%	0.44	0.57	185 sec
Model 5b	91.91%	**81.40%**	0.24	0.63	893 sec
Model 8	83.60%	80.00%	0.46	0.59	120 sec
Model 8b	93.40%	**84.90%**	0.19	0.57	714 sec
Model 9	93.91%	80.50%	0.18	0.73	155 sec
Model 9b	97.55%	**82.20%**	0.08	0.96	931 sec

Table 8: Model performances after increasing the number of epochs

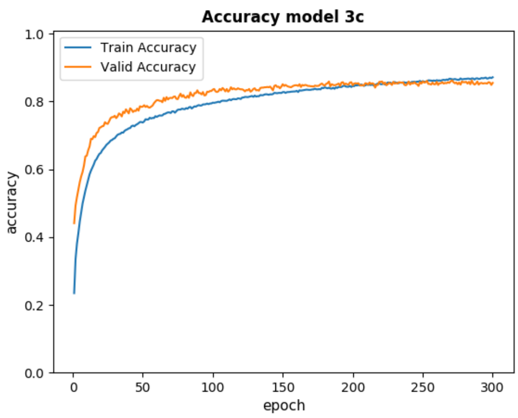
The following section will present a further attempt to improve performances that consists in the addition of batch normalisation to the best models.

4.3 Batch normalisation

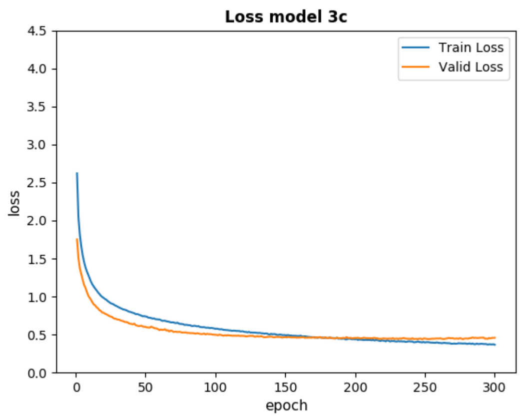
In this section are analysed the performances of the models 3c, 5c, 8c, 9c and 10c, after the addition of batch normalisation. The main purpose of this change is to improve the performance and the stability of the network. The transform is introduced immediately before each ReLU non-linearity, in order to normalise the layer inputs [1].

This experiment has been carried out only on five models, which are the ones that have obtained better performances, and for 300 epochs.

The first experiment is carried out on model 3b, that is the one with the best performances so far.



(a)



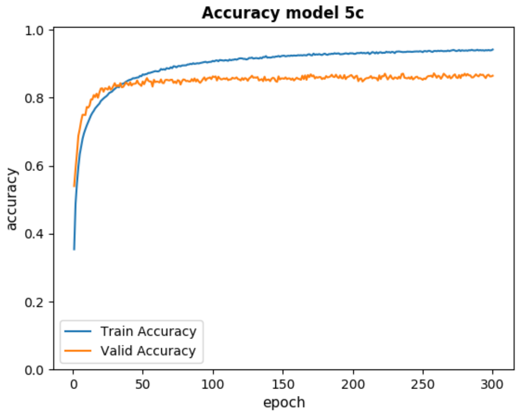
(b)

Figure 14: Training and validation curves in model 3c (with batch normalisation)

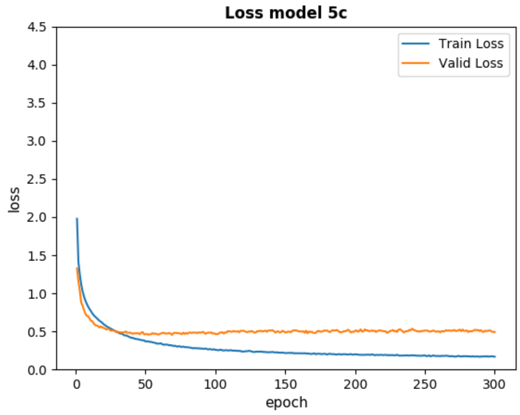
[1] Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

The learning curves are displayed in Figure 14. The validation accuracy is slightly worst compared to that of the model without batch normalisation. In fact, the current one is decreased by 0.20%.

The next experiment is performed on model 5b. In this case, the batch normalisation increases the validation accuracy of the previous model of 5.3%, reaching the new best results of 86.5%, as shown in Figure 15.



(a)

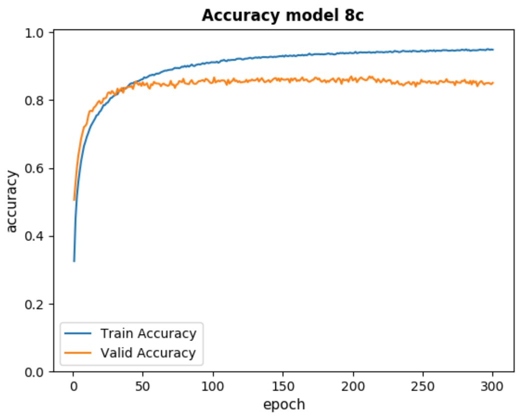


(b)

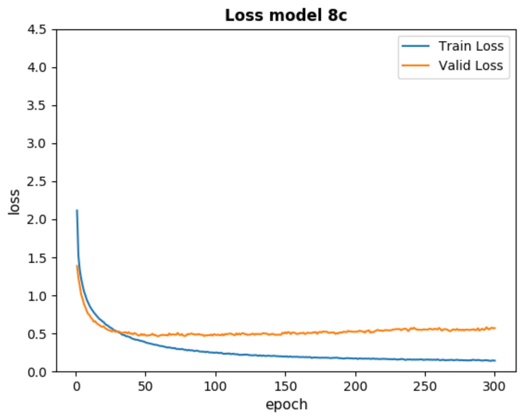
Figure 15: Training and validation curves in model 5c (with batch normalisation)

Probably, the reason why model 5c has better performance than 3c is that batch normalisation works better with a higher learning rate.

The experiment performed on model 8c has achieved lower result respect to model 5c even if further increasing the validation accuracy of 0.20% of the model without batch normalisation. The performances are shown in Figure 16.



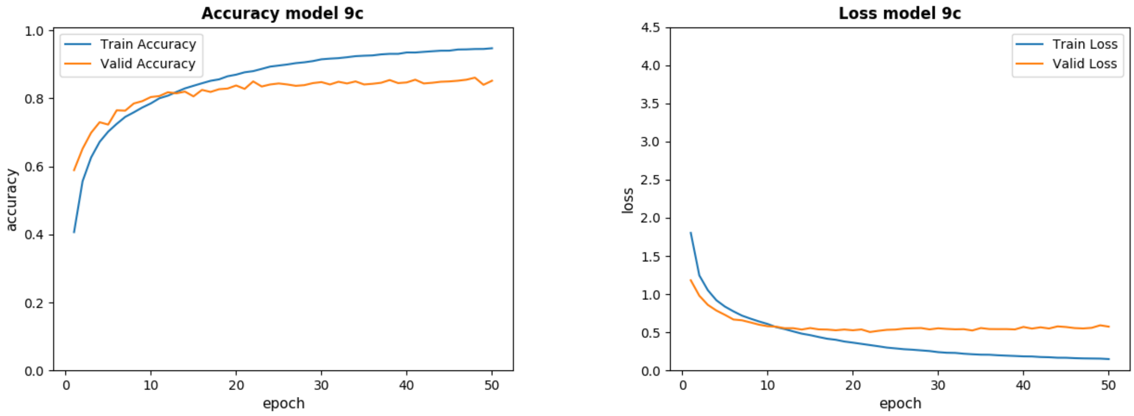
(a)



(b)

Figure 16: Training and validation curves in model 8c (with batch normalisation)

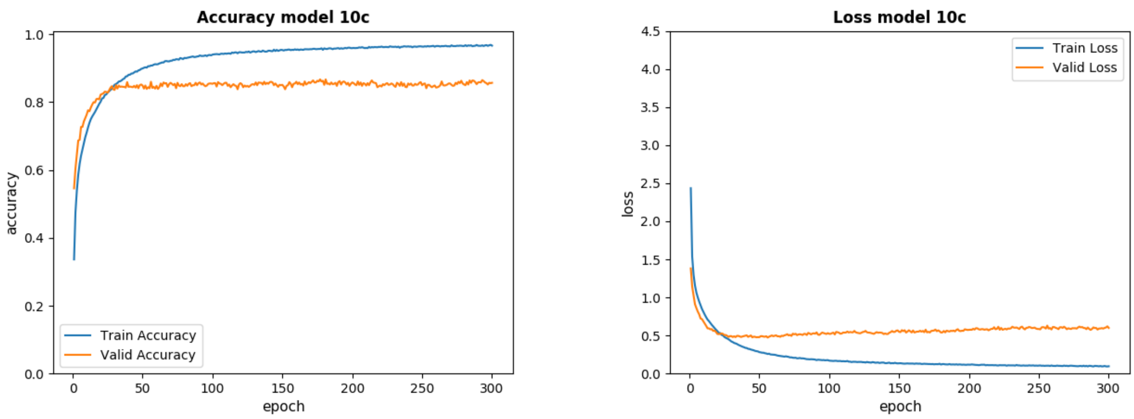
The model 9c, which performances are shown in Figure 17, performs slightly better compared to the previous model. Moreover, the validation accuracy increases by 3.00% respect to the model without batch normalisation.



(a) (b)

Figure 17: Training and validation curves in model 9c (with batch normalisation)

The last experiment performed has achieved a good result, in particular reaching a validation accuracy of 85.70%. Despite these results, the model does not improve the performance of model 5c. The performances are shown in Figure 18.



(a) (b)

Figure 18: Training and validation curves in model 10c (with batch normalisation)

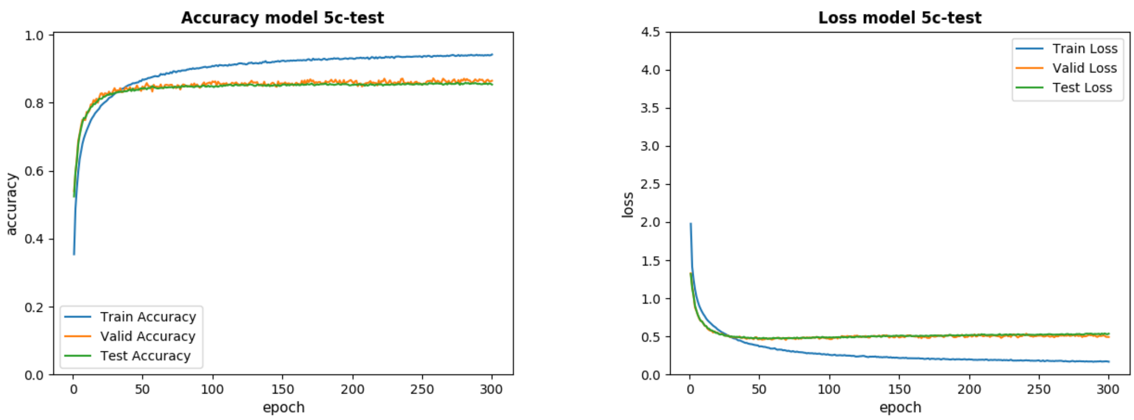
The following table summarises the latest results obtained, highlighting model 5c is the best of all the experiments presented.

	Accuracy		Loss		Train Time
	Train	Validation	Train	Validation	
Model 3c	87.15%	85.50%	0.37	0.46	4254 sec
Model 5c	94.20%	**86.50%**	0.17	0.50	1599 sec
Model 8c	94.87%	85.10%	0.14	0.57	1215 sec
Model 9c	94.74%	85.20%	0.15	0.57	279 sec
Model 10c	96.67%	85.70%	0.10	0.60	1768 sec

Table 9: Model performances after the batch normalisation

5 Test set accuracy

The test set accuracy was measured on the model 5c, that was the one which presents the best performance. As can be seen in Figure 19 and summarised in Table 10, also on the test the performances of the model are high. The test accuracy reached 85.36%, just 1.14% worse than the validation, and the test loss is 0.50.



(a) (b)

Figure 19: Test curves of model 5c

	Accuracy			Loss		
	Train	Validation	Test	Train	Validation	Test
Model 5c	94.20%	86.50%	**85.36%**	1.03	0.47	**0.50**

Table 10: Test set performances of the chosen model