TBMI26 – Computer Assignment Reports
Deep Learning

Deadline – March 14 2021

Author/-s: Matilda Granqvist, Andreas Hertin
matgr197 and andhe794

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. Please upload the document in PDF format. **You will also need to upload the Jupyter notebook as an HTML-file (using the notebook menu: File -> Export Notebook As…)**. We will correct the reports continuously so feel free to send them as soon as possible. If you meet the deadline you will have the lab part of the course reported in LADOK together with the exam. If not, you'll get the lab part reported during the re-exam period.

1. **The shape of X_train and X_test has 4 values. What do each of these represent?**
   They represent a number of images, x-resolution, y-resolution and color channels (RGB).

2. **Train a Fully Connected model that achieves above 45% accuracy on the test data. Provide a short description of your model and show the evaluation image.**
   x = Dense(64, activation="softplus")(x)
   x = Dense(128, activation="softplus")(x)
   x = Dense(32, activation="softplus")(x)
   x = Dense(10, activation="softmax")(x)

   The choice of softplus as an activation was made after some trial and error.

   Three dense layers with higher followed by a dense layer of 10 outputs corresponding to the 10 the 10 label. In figure 1 the result is presented, the accuracy was 50 %.

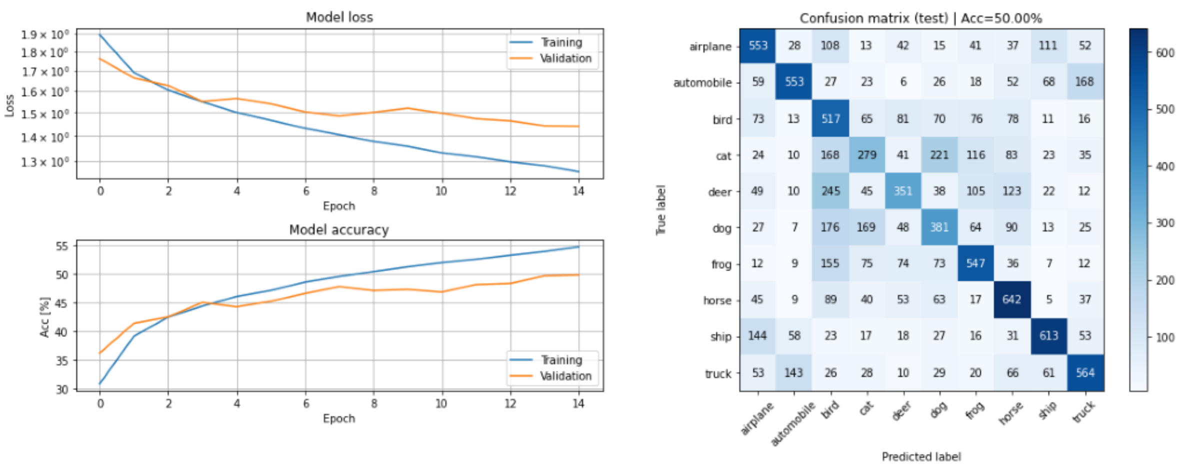

Figure 1. The accuracy and model loss and the confusion matrix.

3. **Compare the model from Q2 to the one you used for the MNIST dataset in the first assignment, in terms of size and test accuracy. Why do you think this dataset is much harder to classify than the MNIST handwritten digits?**

   A lot more images, and all images are 32x32 RGB compared to 28x28 BW which means the data is a lot bigger. It has a lot more features that are all harder to discern from each other.

4. **Train a CNN model that achieves at least 62% test accuracy. Provide a short description of your model and show the evaluation image.**

   The model consists of three Conv2d layers with pooling in between and two Dense layers before the final output Dense layer.

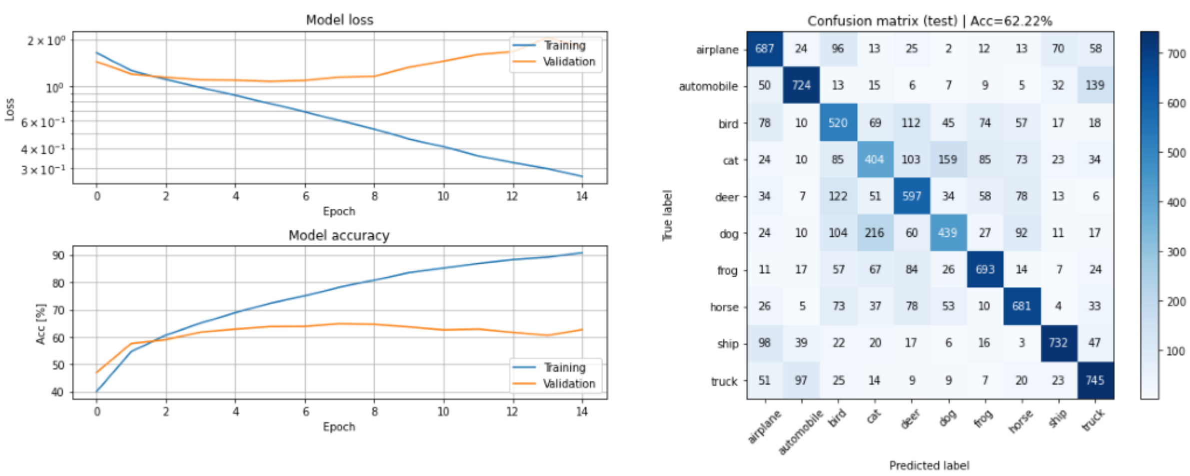

Figure 2. Accuracy, model loss and the confusion matrix

5. **Compare the CNN model with the previous Fully Connected model. You should find that the CNN is much more efficient, i.e. achieves higher accuracy with fewer parameters. Explain in your own words how this is possible.**

Fully Connected has 209 450 parameters and the CNN 128 490. The dense-layers have more parameters than conv-layers. This is possible due to the conv-layers doesn't needing to train all individual pixels as nodes. It only needs to train the ones in the kernel plus bias..

6. **Train the CNN-model with added Dropout layers. Describe your changes and show the evaluation image.**

   We added a Dropout(0.2) - layer after the last pooling layer in the convolution part of the network. This made the validation loss/accuracy reach better values because the dropping of random nodes help with overfitting.

The testing accuracy now increases for all epochs and overall gets a better result.

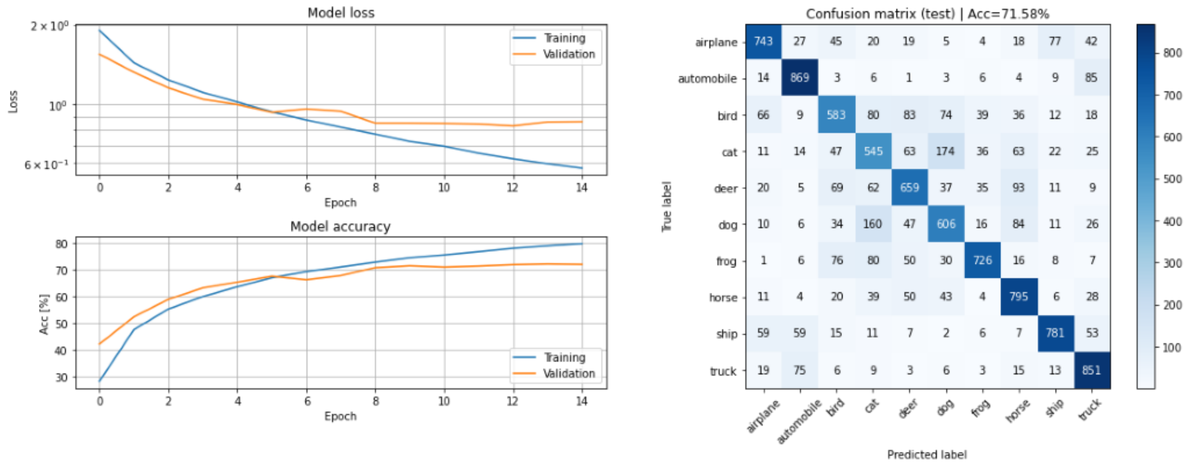

Figure 3.

7. **Compare the models from Q4 and Q6 in terms of the training accuracy, validation accuracy, and test accuracy. Explain the similarities and differences (remember that the only difference between the models should be the addition of Dropout layers).**
   Hint: what does the dropout layer do at test time?
   Dropout removes unnecessary calculations in the test time leading to that the models get less overfitted. Since one drop-layer was added to the Q6 model, it is possible to see that that training accuracy was better but the validation accuracy was worse. The curve looks therefore better with a lot less overfitting. The test accuracy went from 62,22% in Q4 to 71,58% in Q6 which is an improvement.

8. **Train the CNN model with added BatchNorm layers and show the evaluation image.**

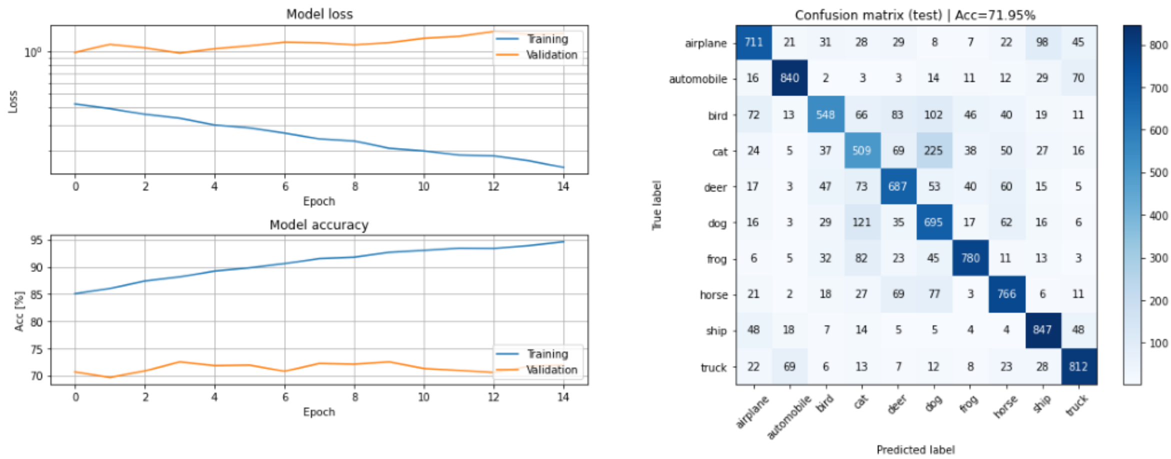

Figure 4. Showing the model loss, model accuracy and the confusion matrix after the BatchNorm was added. The accuracy was 71.95%.

9. **When using BatchNorm one must take care to select a good minibatch size. Describe what problems might arise if the wrong minibatch size is used.**

You can reason about this given the description of BatchNorm in the Notebook, or you can search for the information in other sources. Do not forget to provide links to the sources if you do!

The mini-batch is a part of the data that is being used, and can be normalized if preprocessed. It’s good to use a bigger part of the data. If using a small mini-batch size the part could be unrepresentative of the batches. And if the mini-batch is normalized the normalizations could be off from reality. But too large mini-batch could lead to long calculations.

10. **Design and train a model that achieves at least 75% test accuracy in at most 25 epochs. Explain your model and motivate the design choices you have made and show the evaluation image.**

    Our model consists of double parts with two conv-layers with Batch Norm and ReLU-activation followed by a MaxPooling layer. It was then flattened and fed through a dense layer with BatchNorm and ReLU-activation before finally being classified with a dense layer with softmax-activation.
    The batch_size was set to 64, which is a number that evenly could be divided into the total dataset. It is also less than total dataset size. By using a small batch_size it takes longer to get up the training data accuracy. We used 25 epochs.

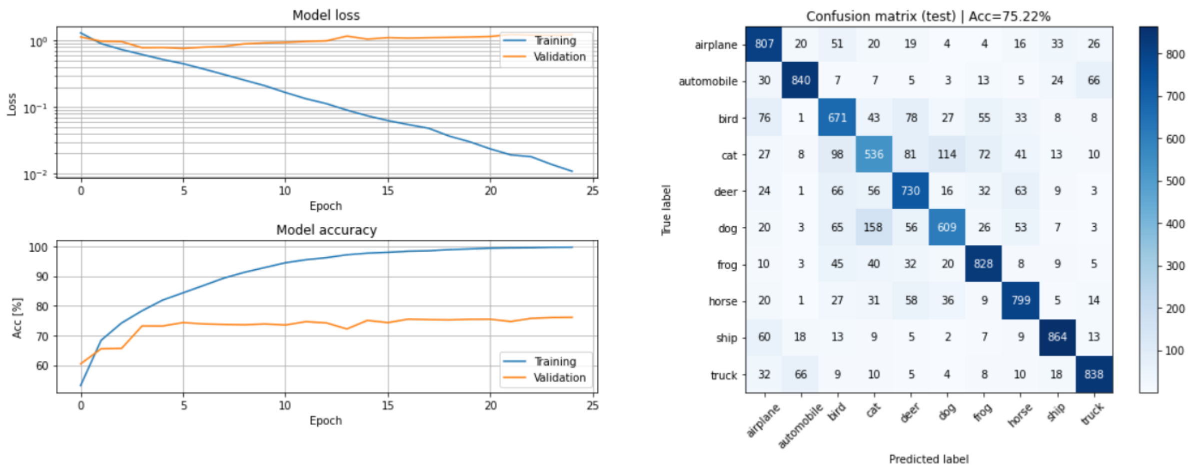

Figure: Showing the model loss, model accuracy and the confusion matrix. The accuracy was 75.22% and the batch_size was 64.