376.054 Machine Vision and Cognitive Robotics
Exercise 4: Deep Learning

Severin Jäger, 01613004

December 7, 2020

Contents

1 Experiment 1 **2**
1.1 Training histories of different approaches 2

2 Experiment 2 **4**
2.1 Stochastic gradient optimiser with learning rate 0.1 4
2.2 Stochastic gradient optimiser with learning rate 0.001 4
2.3 Adam optimiser 4

3 Experiment 3 **5**
3.1 Influence of the test image 5
3.2 Linear classifier and its limitations 6

4 Experiment 4 **7**
4.1 Receptive field 7

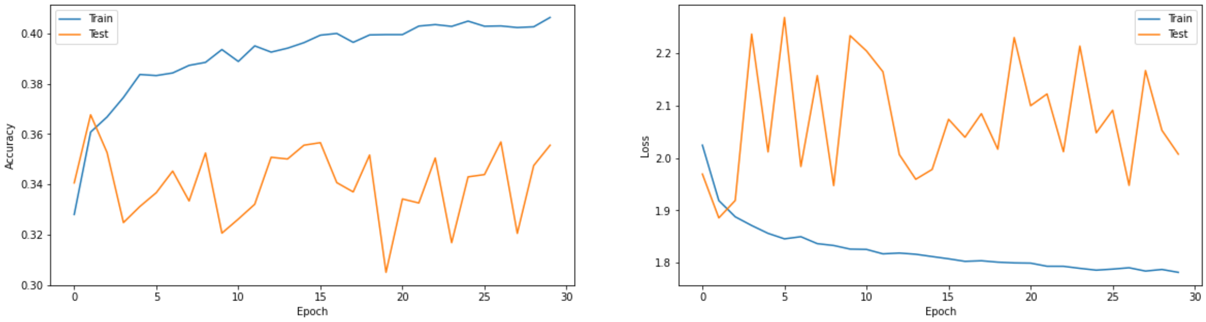


Figure 1: Training history of the linear classifier

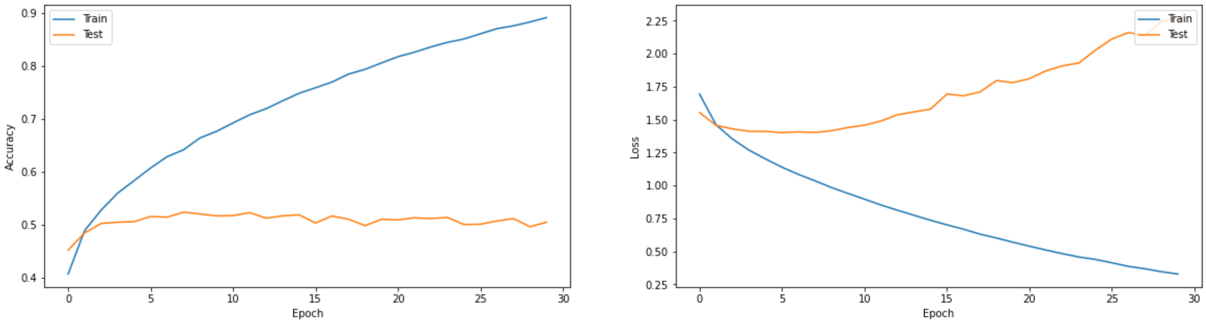


Figure 2: Training history of the multilayer perceptron

1 Experiment 1

1.1 Training histories of different approaches

Figures 1 to 5 compare the training histories (composed of accuracy and loss) for the different implemented machine learning approaches.

The linear classifier (s. Figure 1) has the worst performance in terms of accuracy and hardly improves over the learning epochs. This is reasonable, as its simple structure is not suitable for the complex task of image classification.

The multilayer perceptron (s. Figure 2) contains two layers (plus one output layer) and shows already a better accuracy. However, over the training epochs the training accuracy diverges from the test accuracy (which hardly improves). The loss function divergence supports the observation of significant overfitting. This can be mitigated with regularisation measures (s. Figure 3). In this case, dropout (with $d = 0.2$) and L2 regularisation (with $\lambda = 0.003$) were applied. As a result, on not only overfitting is mitigated, but also the peak test accuracy is increased (from 52 % to 56 %).

A major step in performance comes with the use of convolutional neural networks (CNNs). Their structure is advantageous for image classification, as they detect features, shapes or objects independent of their exact position in the image. This results in significantly improved accuracy and an overfitting-free learning process if dropout (with $d = 0.2$) is applied (s. Figure 4).

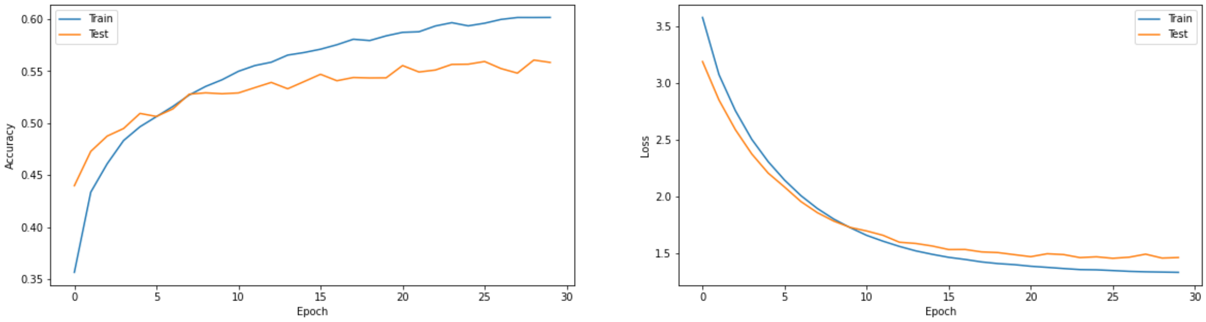


Figure 3: Training history of the multilayer perceptron with regularisations measures

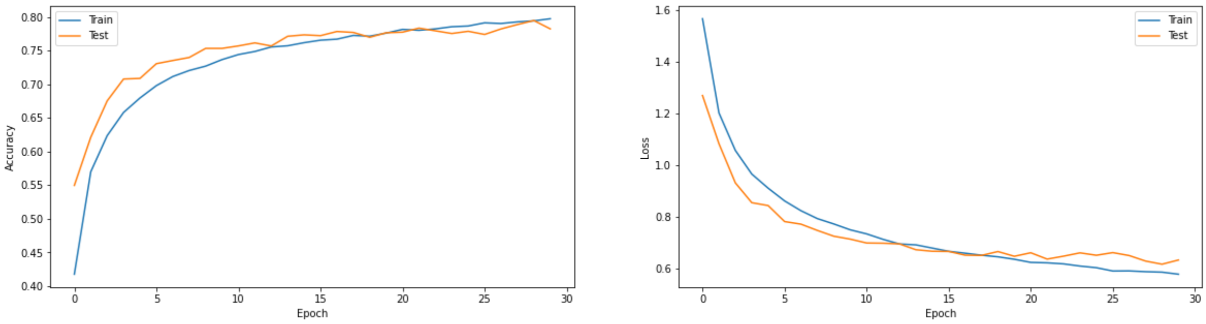


Figure 4: Training history of the convolutional neural network

Another approach is using a pre-trained CNN, in this case ResNet50 was used. As this network was trained with 1000 classes, the last layer was removed and replaced by a dense layer with 1000 nodes and an output layer mapping to the 10 relevant classes. This approach works well (s. Figure 5), even after few training cycles. However, the training is computationally expensive and the accuracy hardly increases with further training epochs.

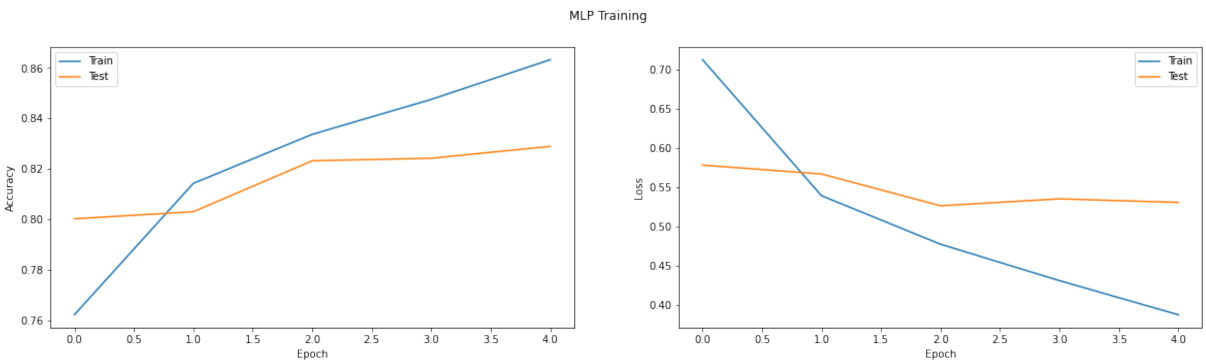


Figure 5: Training history of the ResNet50 network

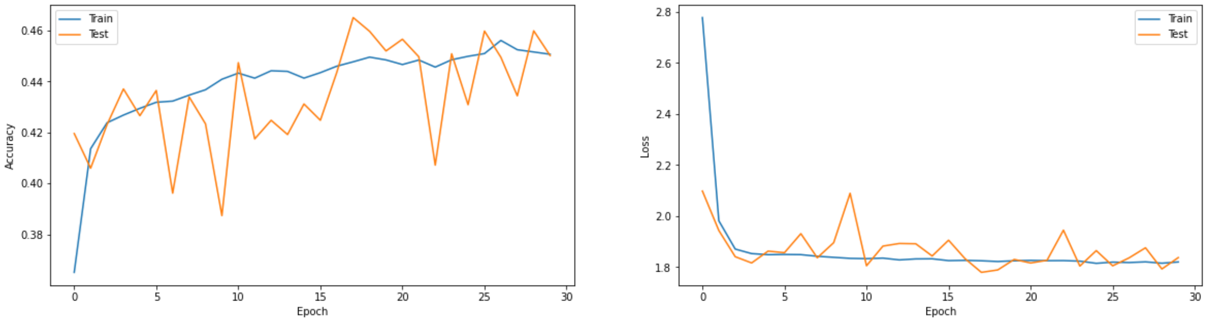


Figure 6: Training history with the SGD optimiser with a learning rate of 0.1

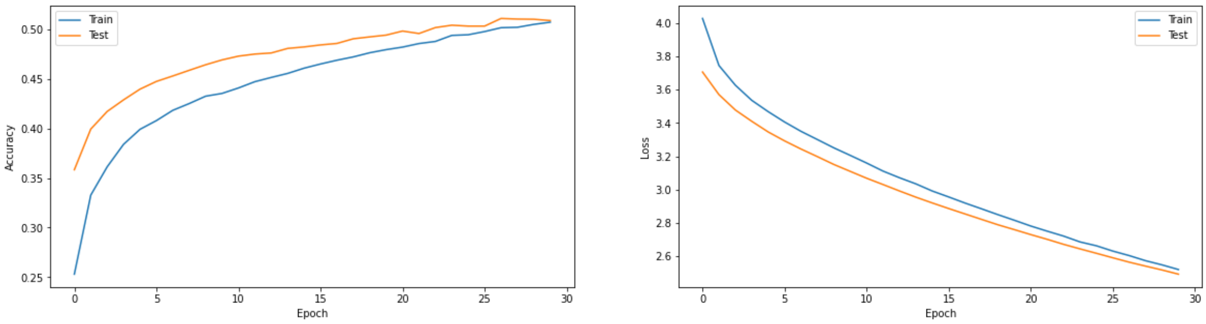


Figure 7: Training history with the SGD optimiser with a learning rate of 0.001

2 Experiment 2

In this section, the optimiser of the previously described multilayer perceptron with regularisation measures is altered in order to investigate its effect.

2.1 Stochastic gradient optimiser with learning rate 0.1

This optimiser features a very high learning rate and is therefore able to minimize the loss function rather quickly. However, this does not lead to a high test accuracy as the learning results from earlier cycles are replaced rather quickly (s. Figure 6). So the model adjusts too quickly and is therefore only suitable for the last data used for training.

2.2 Stochastic gradient optimiser with learning rate 0.001

In case the learning rate is reduced, the previously described disadvantage is mitigated (s. Figure 7). With a learning rate of 0.001 the loss test as well as the training accuracy increases monotonously. However, the training process takes very long, here it does not converge within 30 cycles.

2.3 Adam optimiser

The Adam optimiser features an adaptive learning rate. In theory, this should combine the advantages of the two previously described approaches: It offers a high learning rate in the beginning (and therefore fosters convergence) and decreases it steadily in order to

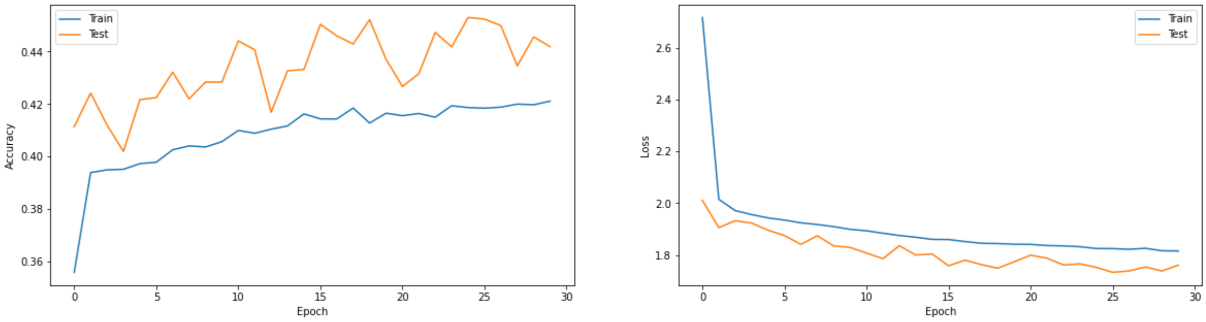
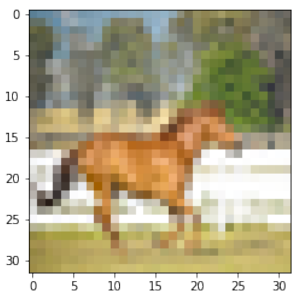


Figure 8: Training history with the Adam optimiser

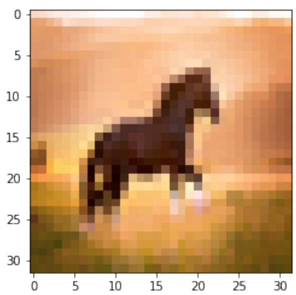
maintain a smooth improvement process without overruling the weights from the early training completely.

Figure 8 shows this exemplarily: Initially, the loss drops quickly, in later epochs only slight changes emerge. This matches the expectations. However, the overall accuracy is below the ones reached with the SDG optimiser. One reason might be that the changes with the high learning rate in the beginning were only suitable for a small subset of the training data (so overfitted for these images) and later steps cannot overrule these weights as their learning rates are very small.

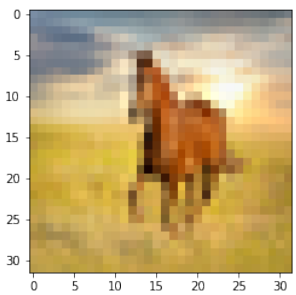
3 Experiment 3



(a) Image 1



(b) Image 2



(c) Image 3

Figure 9: Test images

3.1 Influence of the test image

Table 1 compares the detection results of the multilayer perceptron with regularisation as described above with the ones obtained by the aforementioned CNN. The respective test images are shown in Figure 9.

Image 1 is the greatest challenge for both classifiers: Both clearly detect a deer (the CNN even clearer), however their second choice is a horse. This is likely to be due to the colors: A deer might be brown as well and the shapes might resemble, too.

Image 2 clearly separates the two classifiers: The CNN unambiguously detects a horse, while the MLP is unable to classify the image properly. One potential reason is that the

Classifier	$p_{horse,1}$	$p_{deer,1}$	$p_{horse,2}$	$p_{deer,2}$	$p_{horse,3}$	$p_{deer,3}$
MLP	27.7 %	42.1 %	26.2 %	16.6 %	49.1 %	19.2 %
CNN	24.5 %	74.6 %	96.4 %	3.1 %	97.6 %	2.1 %

Table 1: Detection results of the MLP with regularisation and the CNN

Figure 10: Classification results of the linear classifier

CNN is able to detect the shape, while the MLP is mainly relying on the colours and cannot classify the image as typical horse images (in the training set) have a different background.

However, image 3 shows that both detectors are able to classify image correctly: Here the MLP also detects a horse rather clearly. As mentioned before, the background colour might be a reason for this behaviour.

3.2 Linear classifier and its limitations

The linear classifier is a very simple machine learning system. However, as shown in Figure 10 it is able to detect some images correctly. This is mainly due to a similar pattern of colors within the image. So brown or white horses in front of a green background and filling most of the image are detected rather well, while different settings like different background colours or portraits are not detected. Additionally, images with a similar colour distribution tend to be classified as horses as well. CNNs overcome some of these disadvantages, especially by providing translation invariance due to the kernel moving over the image. Additionally, it preserves neighbourhood relations (as the image is not vectorised) and can therefore utilise features like edges.

4 Experiment 4

4.1 Receptive field

Trivially, the input layer has a receptive field of $l_0 = 1$. The receptive field of the first convolutional layer can be calculated using the kernel size $w_{kernel} = 3$ and the equation from slide 27 in lecture 7 as

$$l_1 = l_0 + (w_{kernel} - 1) = 1 + 2 = 3. \tag{1}$$

The receptive field of the first MaxPool layer is (considering the kernel width of the pooling operator $w_{pool} = 2$)

$$l_2 = l_1 + (w_{pool} - 1) = 3 + 1 = 4. \tag{2}$$

In the following, the receptive field of the second convolutional layer is under consideration of stride the pooling layer $strides = 2$ given as

$$l_3 = l_2 + (w_{kernel} - 1) \cdot s = 4 + 2 \cdot 2 = 8. \tag{3}$$

The receptive field of the second MaxPool layer calculates to

$$l_4 = l_3 + (w_{pool} - 1) \cdot s = 8 + 2 = 10. \tag{4}$$

Finally, the receptive field of the third convolutional layer is given as

$$l_5 = l_4 + (w_{kernel} - 1) \cdot s^2 = 10 + 2 \cdot 2^2 = 18. \tag{5}$$