

Frage 1

Antwort
gespeichert

Erreichbare
Punkte: 1,00

Frage
markieren

Assume you have a number of different deep-learning classifiers. Which of the following metrics should be used to decide which model is best?

- ☐ a. Test loss
- ☐ b. Training loss
- ☐ c. Training accuracy
- ☒ d. Test accuracy

[Meine Auswahl widerrufen](#)

Frage 2

Antwort
gespeichert

Erreichbare
Punkte: 4,00

Frage
markieren

In task d) of the exercise, we visualize the weights of a very small neural network. For each of the following descriptions, select the class whose visualized weight matches best.

brown area in the center surrounded by green spots

deer

big green spot in the center

frog

blue in the bottom corners

ship

big reddish area in the upper half with a green spot below

horse



Frage 3

Antwort
gespeichert

Erreichbare
Punkte: 1,00

Frage
markieren

We are training a classifier using the cross-entropy loss. We observe, that the training loss decreases the whole time during training. Which of the following statements regarding the test loss is true? Only consider the overall trend, not random fluctuations.

- ☒ a. The test loss can get worse than the average loss of the first epoch.
- ☐ b. It might happen, that the test loss starts to increase after some time but it will stay below the average loss of the first epoch.
- ☐ c. Independent of the model and the data, the test loss will always decrease.

[Meine Auswahl widerrufen](#)

Frage 4

Antwort
gespeichert

Erreichbare
Punkte: 2,00

Frage
markieren

How many trainable parameters has the MLP in exercise task g)?

Antwort: 1578506

```
relu_mlp = nn.Sequential(  
    # YOUR CODE HERE  
    nn.Flatten(),  
    nn.Linear(in_features=3072, out_features=512),  
    nn.ReLU(),  
    nn.Linear(in_features=512, out_features=10)  
)
```

$(3072+1)*512$

$(512+1)*10$

Total: 1578506

Or use

```
n_params = sum(p.numel() for p in model.parameters() if p.requires_grad)  
print(f'Number of trainable parameters: {n_params}')
```

Frage **5**

Antwort
gespeichert

Erreichbare
Punkte: 3,00

🚩 Frage
markieren

How many trainable parameters has the CNN in exercise task h)?

Antwort:

```
cnn = nn.Sequential(  
    nn.Conv2d(in_channels=3, out_channels=64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2, stride=2),  
    nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1),  
    nn.ReLU(),  
    nn.MaxPool2d(kernel_size=2, stride=2),  
    nn.Flatten(),  
    nn.Linear(in_features=4096, out_features=10)  
)
```

$(3*3*3+1)*64$

$(64*3*3+1)*64$

$(4096+1)*10$

Total: 79690

Frage 6

Antwort
gespeichert

Erreichbare
Punkte: 2,00

🚩 Frage
markieren

What is the receptive field of the CNN in exercise task I)? Only enter n if the size is $n \times n$. Hint: draw a 1D visualization.

Antwort:

```
cnn_global_pool = nn.Sequential(  
    # YOUR CODE HERE  
    nn.Conv2d(in_channels=3, out_channels=64, kernel_size=3, padding=1, stride=2, bias=False), # stride=2  
    nn.BatchNorm2d(64), #Batch normalization  
    nn.ReLU(),  
  
    nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1, stride=2, bias=False), # stride = 2  
    nn.BatchNorm2d(64), #Batch normalization  
    nn.ReLU(),  
  
    nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1, stride=2, bias=False), # stride = 2  
    nn.BatchNorm2d(64), #Batch normalization  
    nn.ReLU(),  
  
    GlobalAvgPool2d(), # input: [64, 4, 4] -> output: [64], for each batch  
    nn.Linear(in_features=64, out_features=10)  
)
```

1x1 -> 3x3 -> 7x7 -> 15x15

Frage 7

Antwort
gespeichert

Erreichbare
Punkte: 2,00

🚩 Frage
markieren

What is the final test accuracy your ResNet achieved after 50 epochs *without* learning rate decay?

Antwort: