DEEP LEARNING: CIFAR10

CHABCHOUB aymen - LAMA Graih Jules - MALONGA-NKOUNKOU Gautier - OUMESSAOUD Rabah

Presentation of Cifar 10

- CIFAR stands for: Canadian Institutes For Advanced Research
- collection of images that are commonly used to train machine learning and computer vision algorithms.
- The dataset is composed of **10 classes**:
  - cars, trucks, airplanes, ships
  - deer, dogs, cats, birds, horses, frogs

The dataset in theory

- The CIFAR-10 dataset consists of 60000
- 32x32 colour images
- 6000 images per class
- 50000 training images
- 10000 test images
- The classes are completely mutually exclusive

The dataset in practice

- ***5 pickled files***
  - data
  - labels
  - label_names
- ***data***:
  - 10000 x 3072 numpy array
  - 10000 is the number of images
  - 3072 = 32*32*3(channels)
- ***Labels***:
  - list of 10000 numbers between 0-9.
- ***label_names***:
  - string list with label names (size 10)

Dataset: conclusion

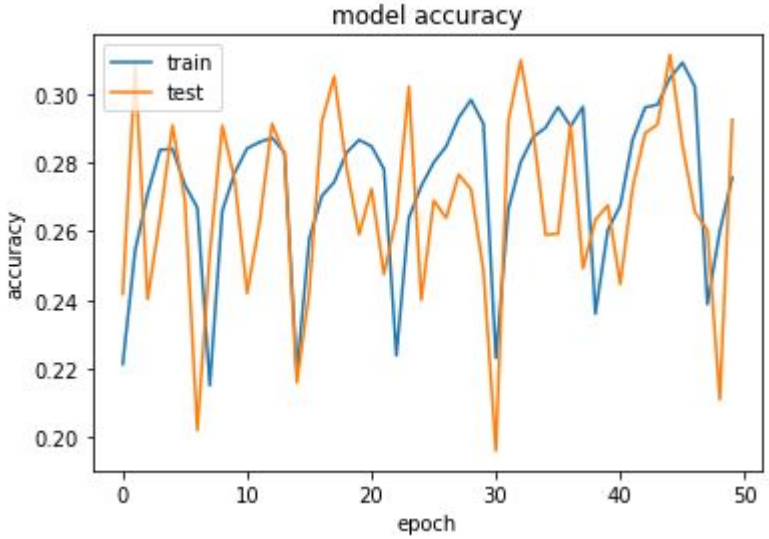
- Small dataset: fits in the ram!
- Efficient for quick learning and fine tuning
- Classes are not skewed

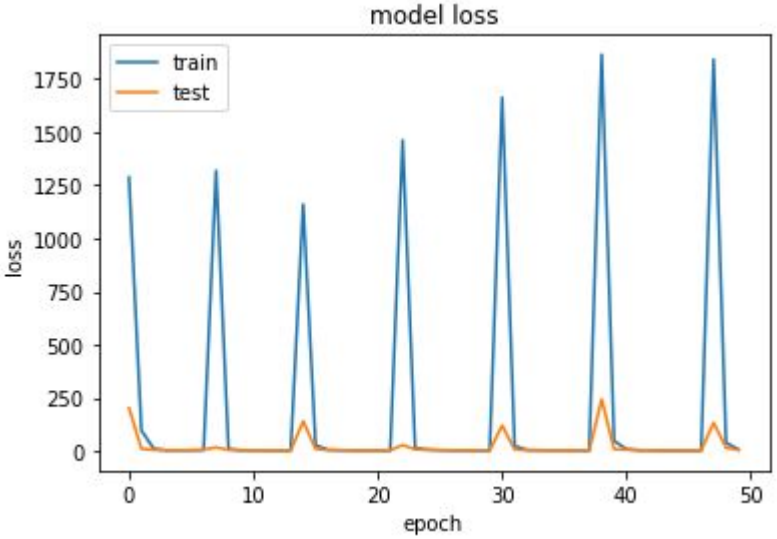
Linear models

- 3072 pixels for the input
  - activation function: linear
- 10 classes for the input with Softmax activation function
- optimizer: Adam

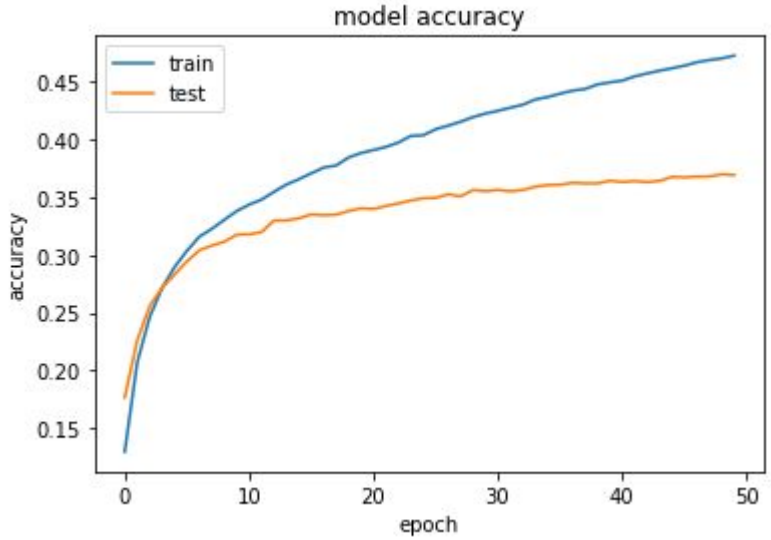
- 3072 pixels for the input
  - activation function: TanH
- 10 classes for the input with Softmax activation function
- optimizer: Adam

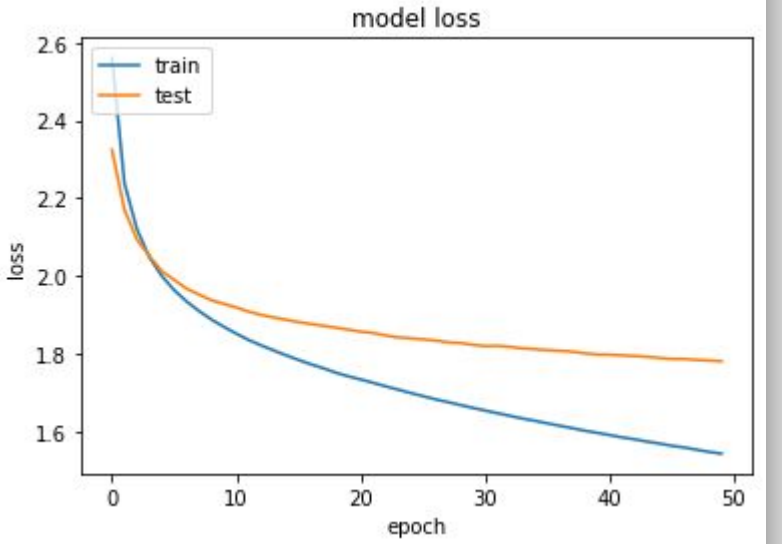
Linear model v1





Linear model v2





Linear models: conclusion

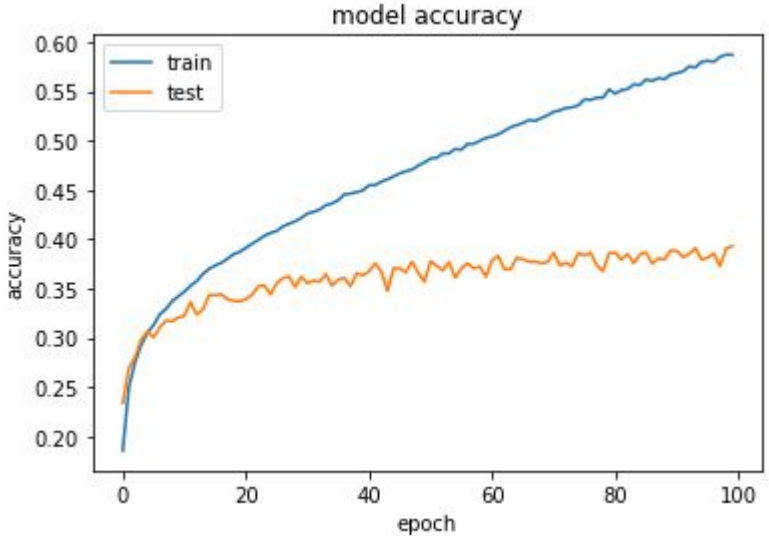
- The model's predictions won't improve by just tuning the hyperparameters
- Adding layers allows predicting more complex functions
- Different optimizer allow faster model converging

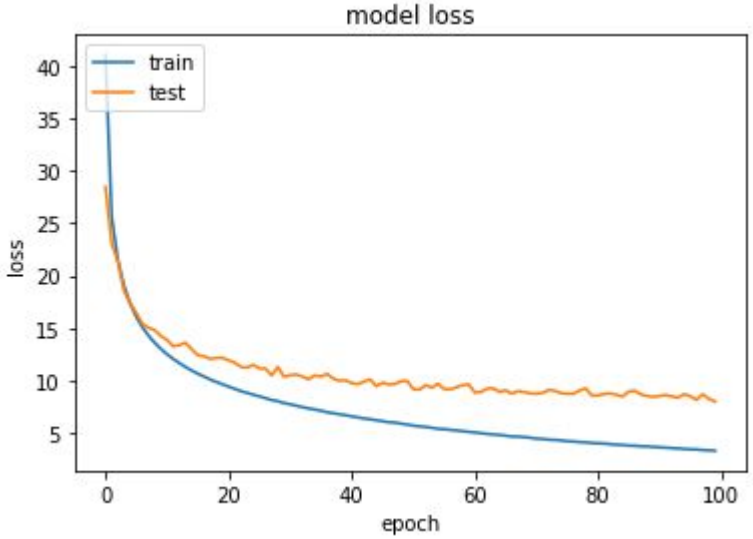
Feed forward models

- 3072 pixels for the input
  - activation function: reLu
  - 2 hidden layers (1040)
- 10 classes for the input with Softmax activation function
- optimizer: Adadelta

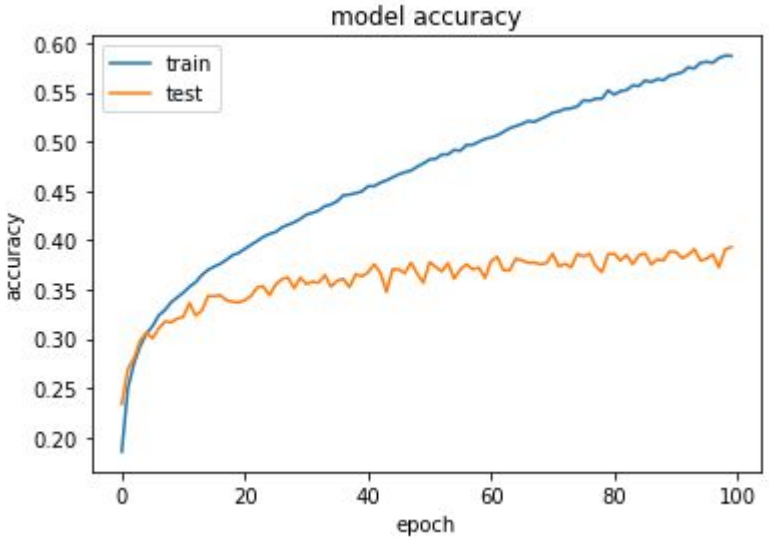
This model is close to our Linear model V2

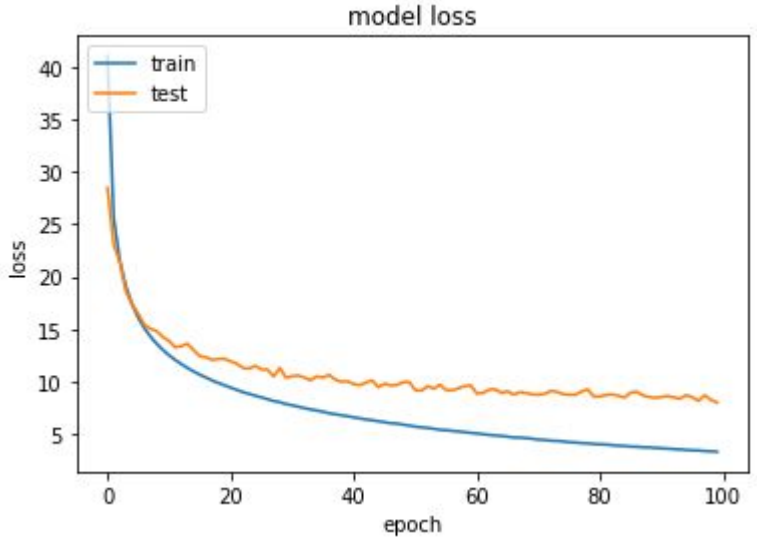
Feed forward v1



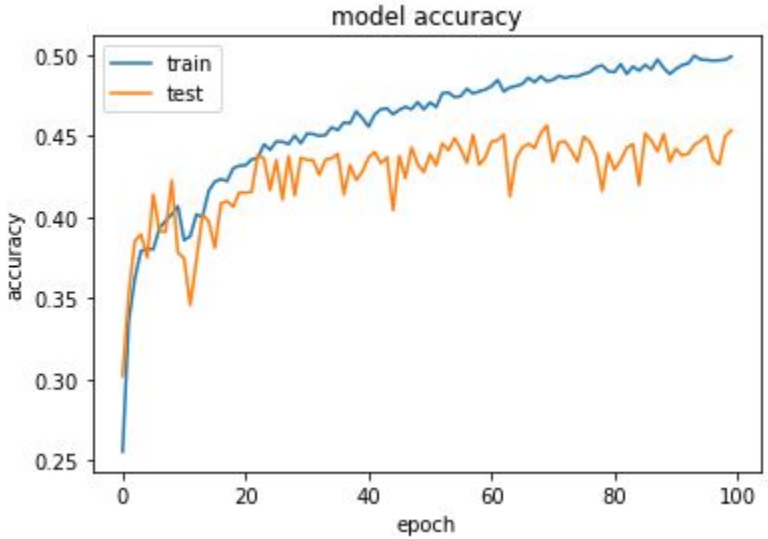


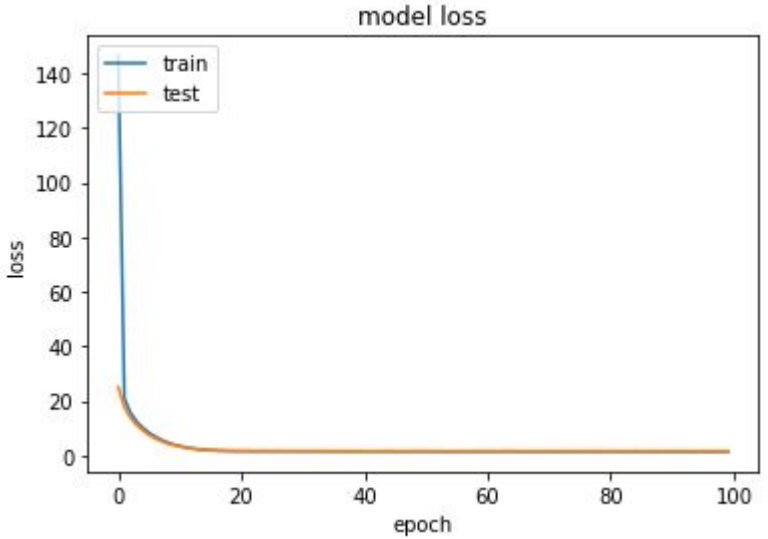
Feed forward: regularizer 0.01



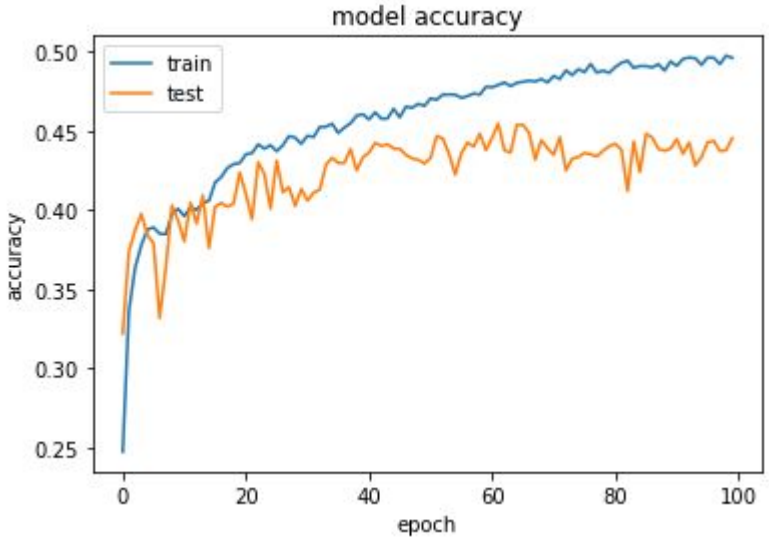


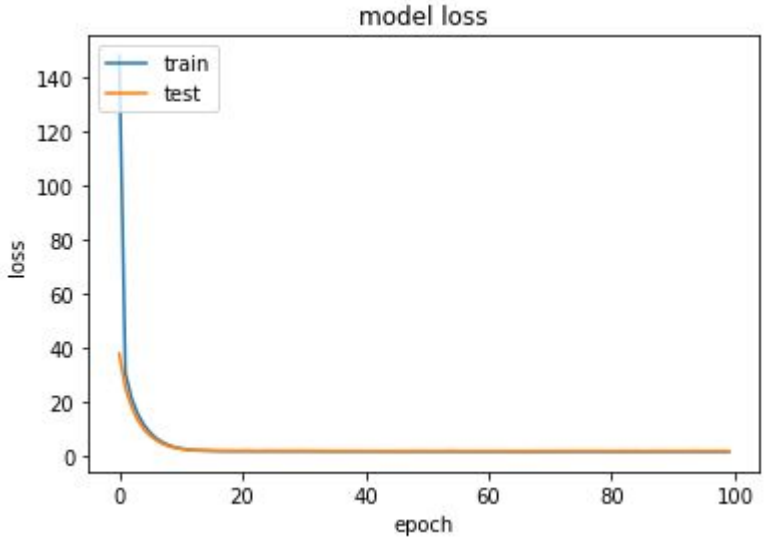
Feed forward: regularizer 0.05



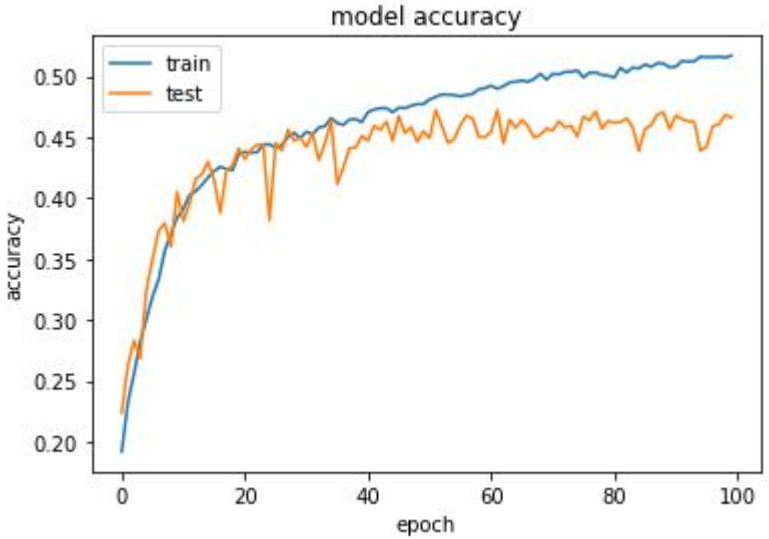


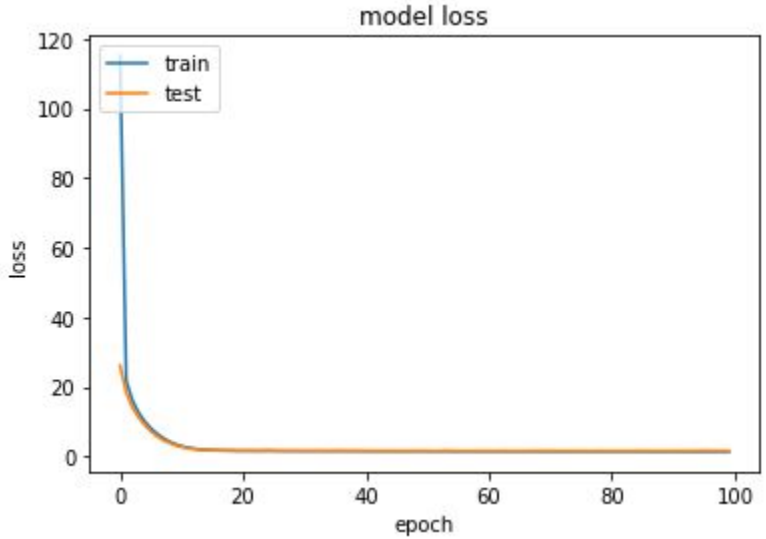
Feed forward: regularizer 0.1



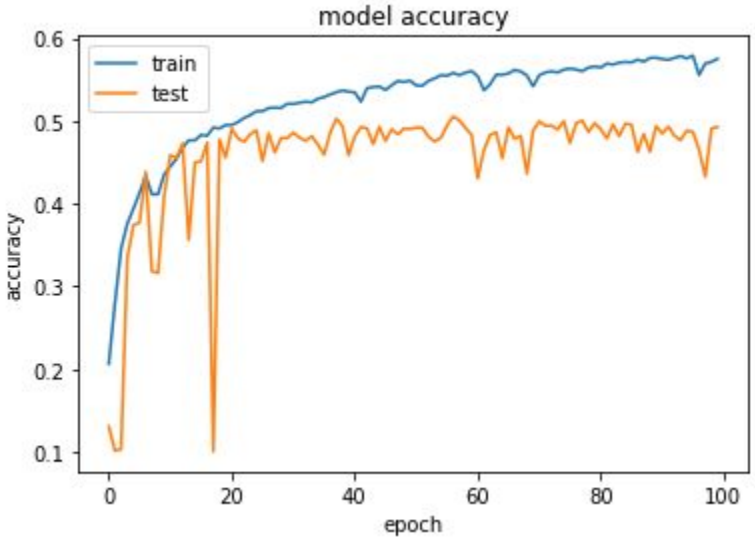


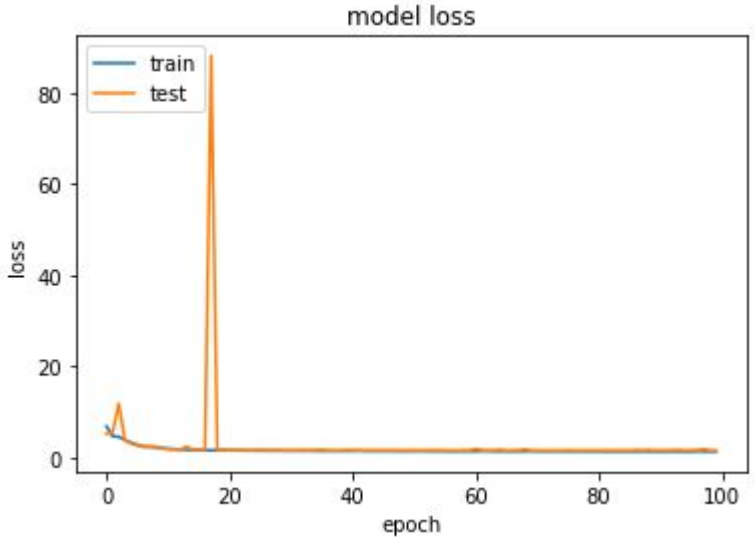
Feed forward: regularizer 0.05 + dropout 0.1





Feed forward: regularizer 0.05 + dropout 0.2 + BatchNormalization

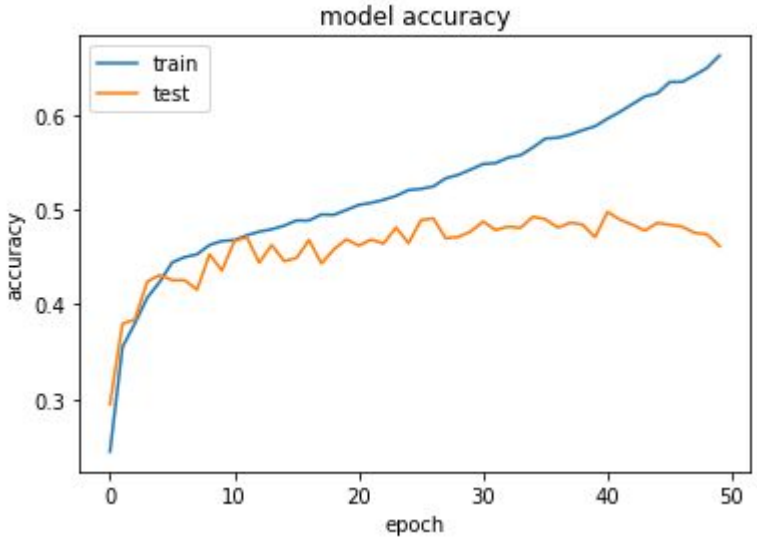


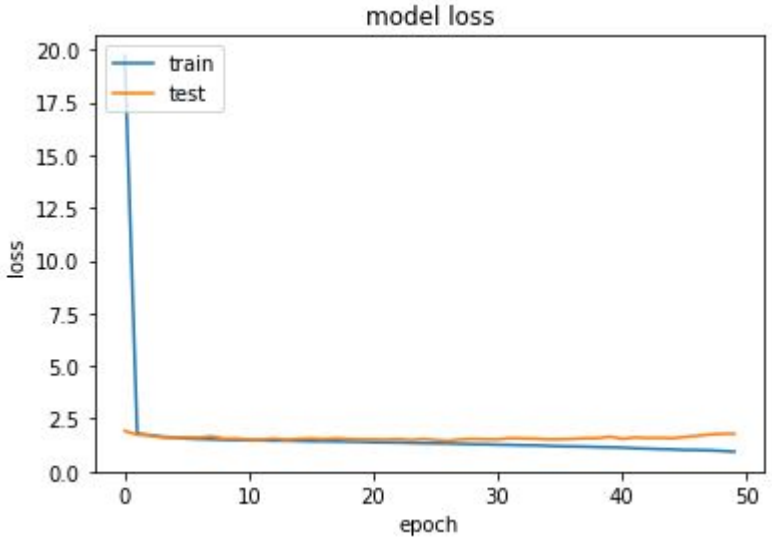


Feed forward models

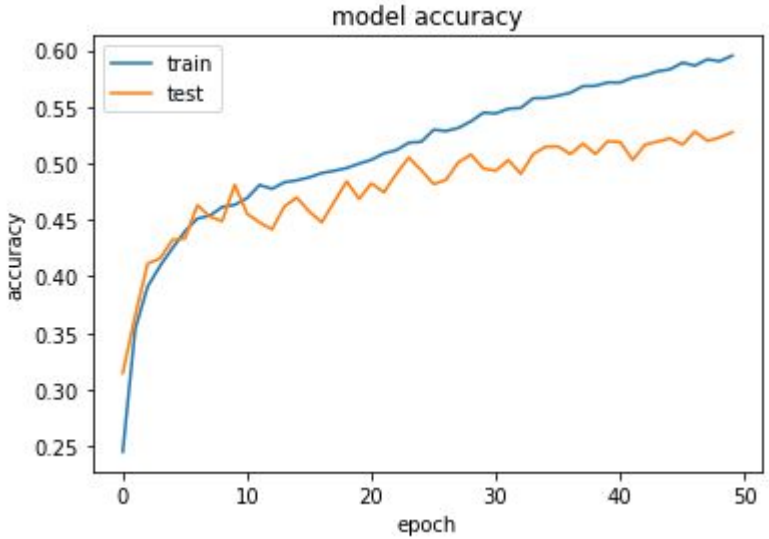
- 3072 pixels for the input
  - activation function: reLu
  - 5 hidden layers
- 10 classes for the input with Softmax activation function
- optimizer: Adam

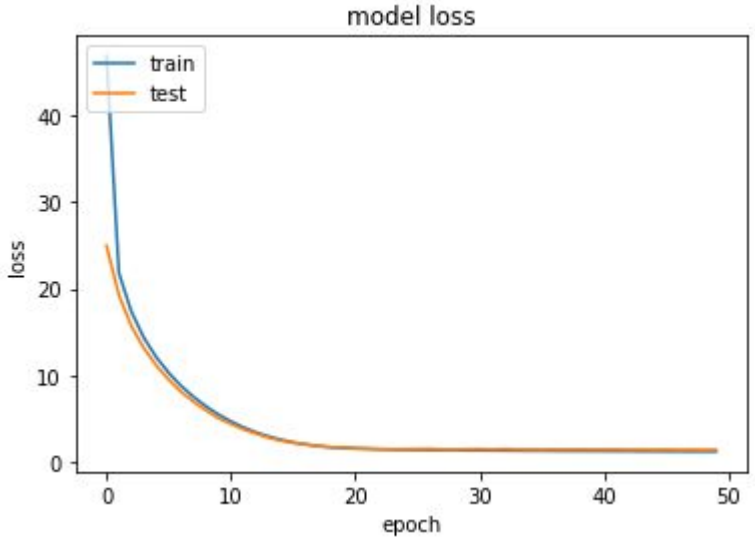
Feed forward: no regularization



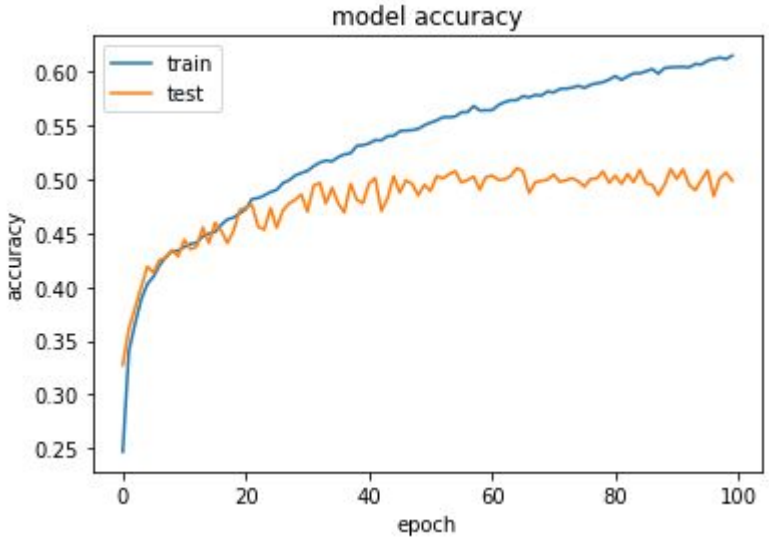


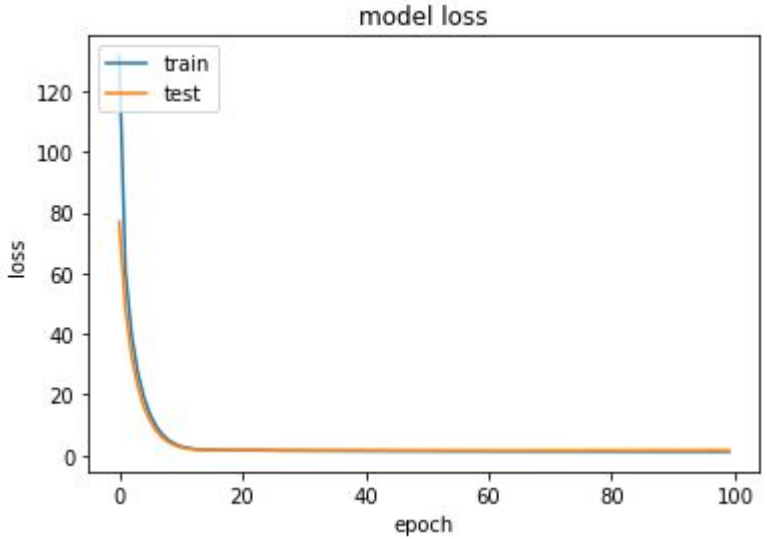
Feed forward: regularization 0.1



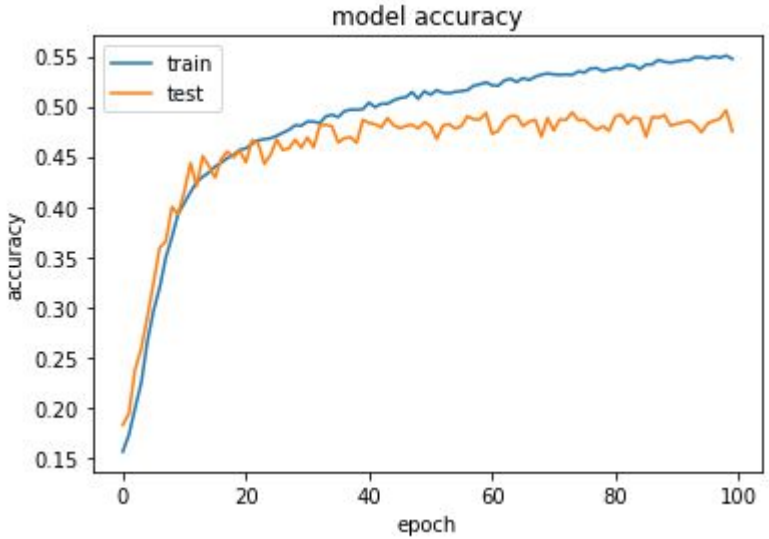


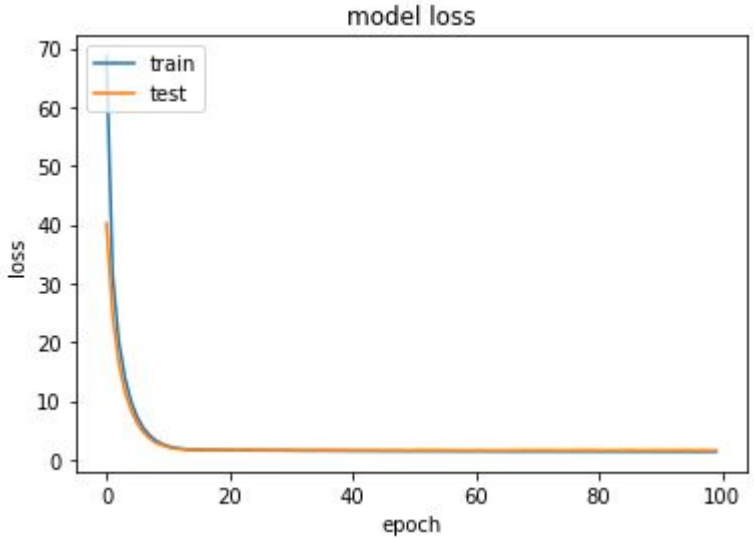
Feed forward: regularization 0.05



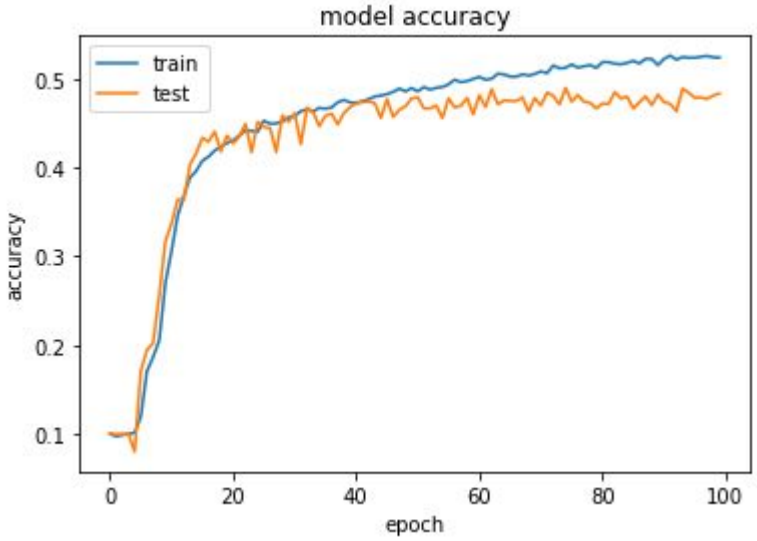


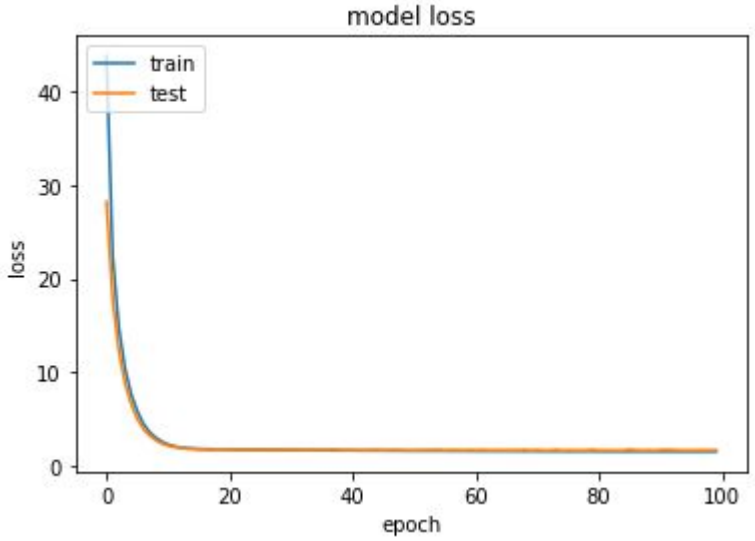
Feed forward: regularization 0.1 + dropout 0.1



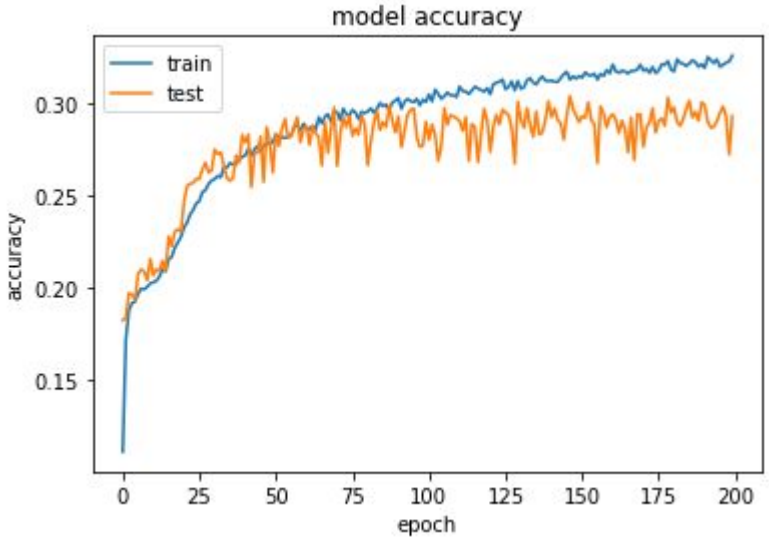


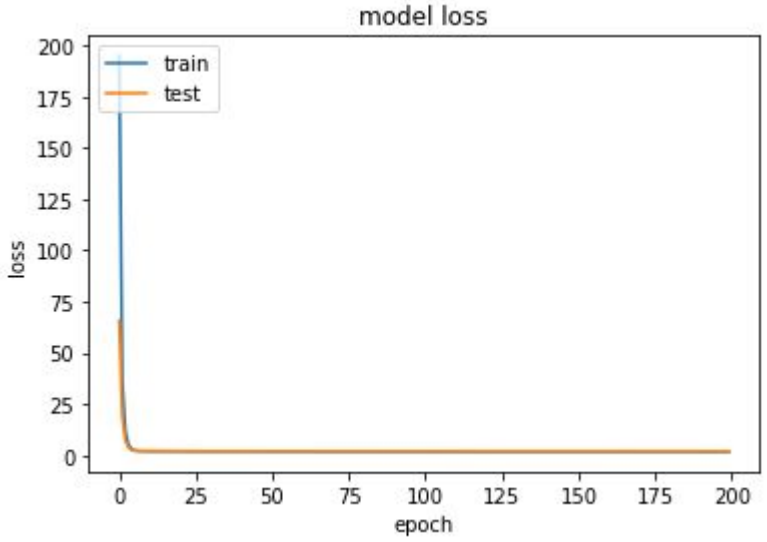
Feed forward: regularization 0.1 + dropout 0.2



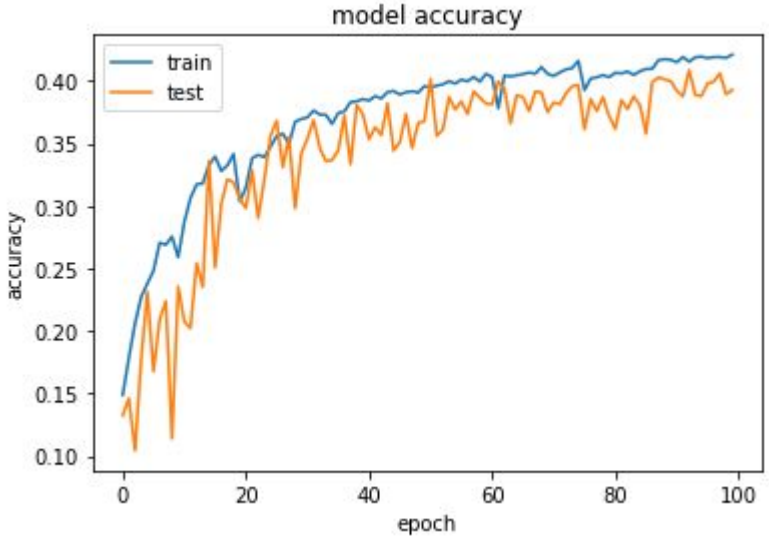


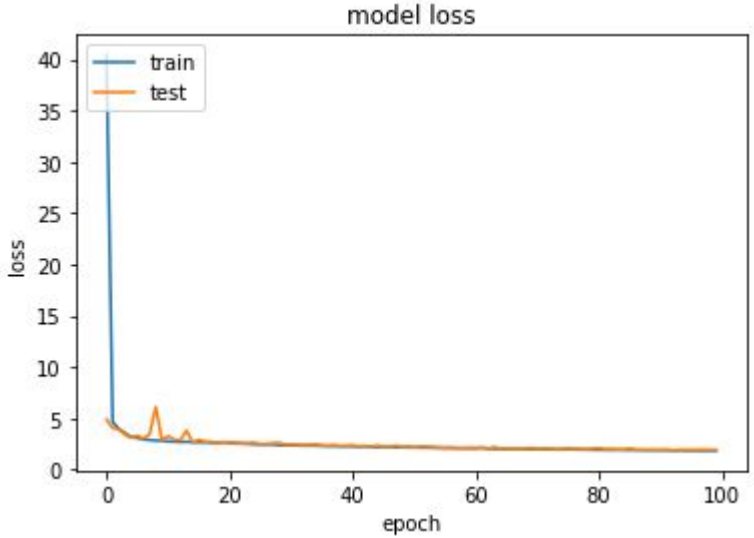
Feed forward: regularization 0.1 + dropout 0.2 + lr 0.001



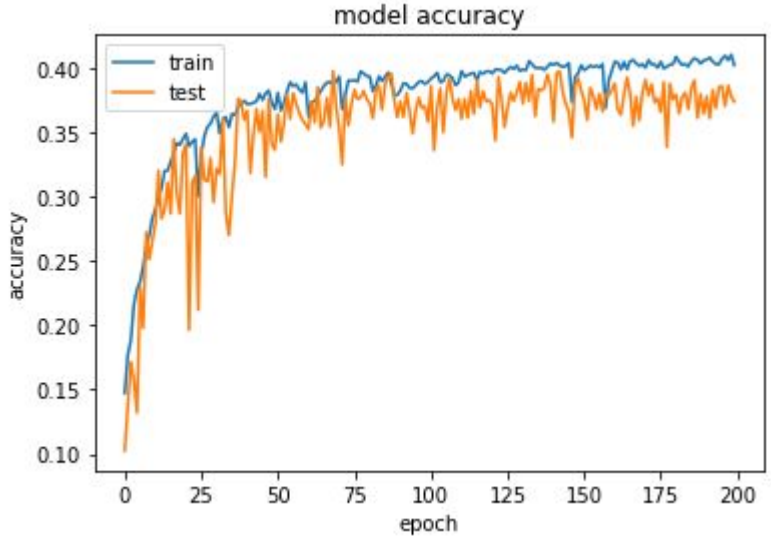


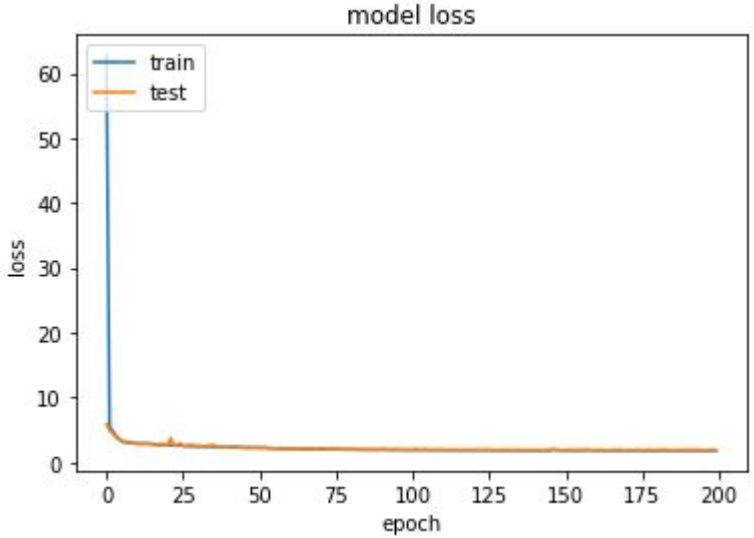
Feed forward: regularization 0.1 + dropout 0.2 + 1 layer (6 dense)





Feed forward: regularization 0.1 + dropout 0.2 + 1 layer + lr 0.001

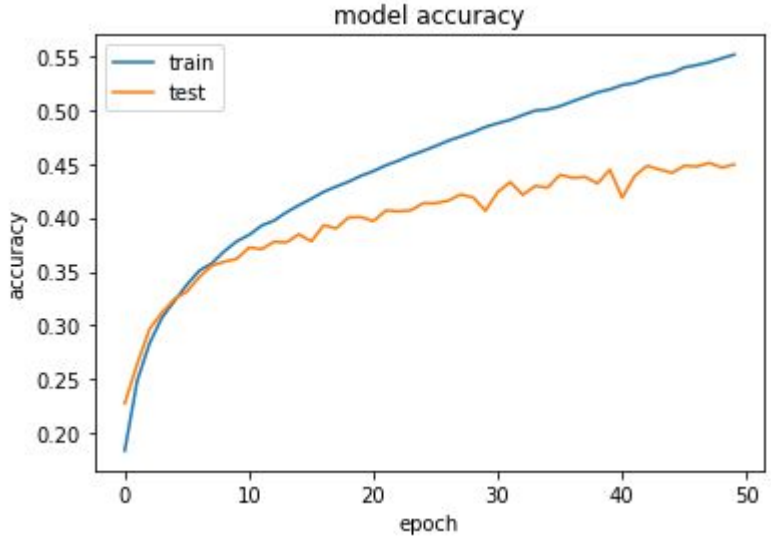


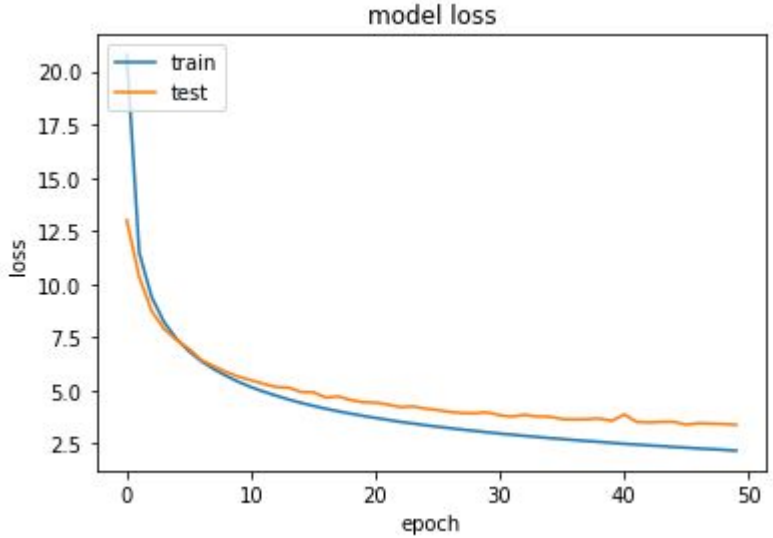


Convolutional neural network

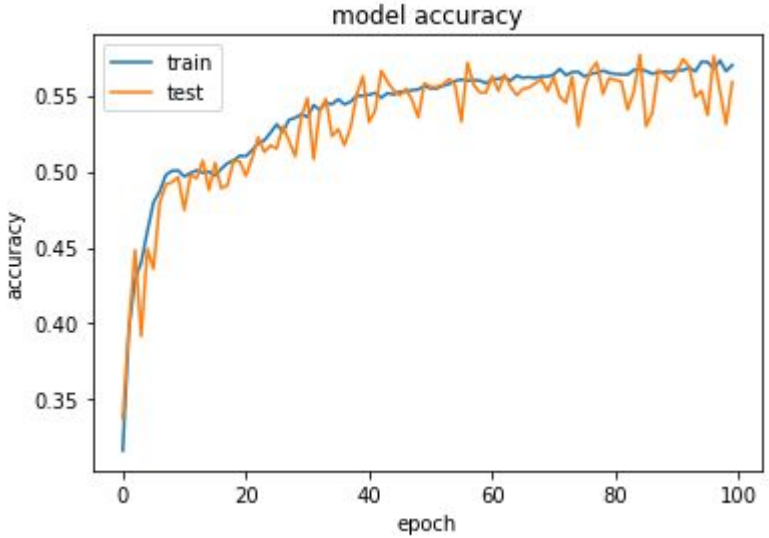
- 32,32,3 input shape
  - activation function: reLu
  - kernel_size: 3,3
  - filters: 32
- 2 hidden layers: 1040 neurons
- 10 classes for the input with Softmax activation function
- optimizer: Adam

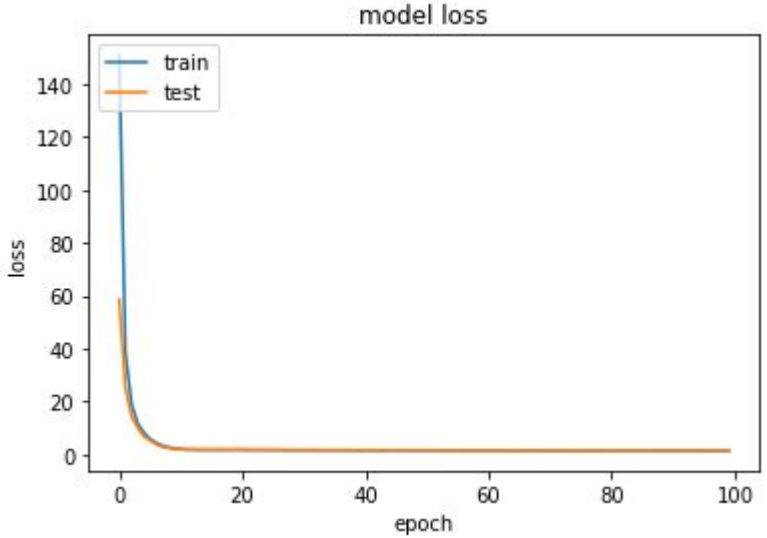
CNN


model accuracy
train
test
accuracy
epoch
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
10
20
30
40
50

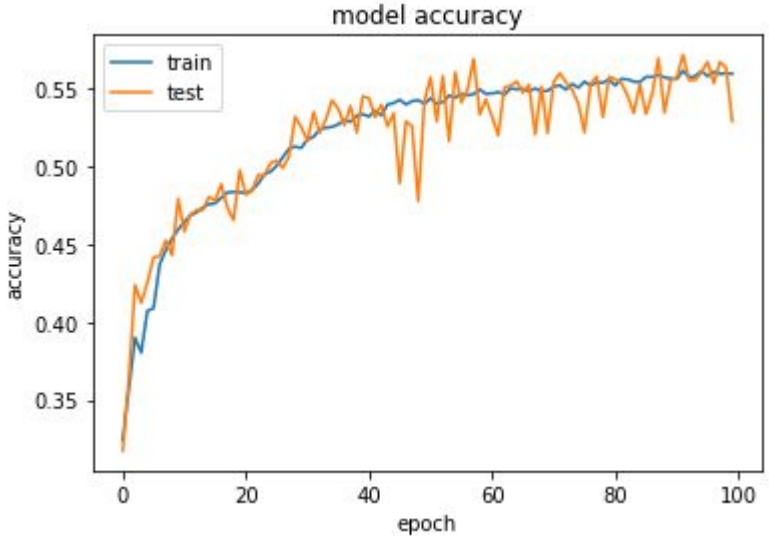

model loss
train
test
loss
epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0
10
20
30
40
50

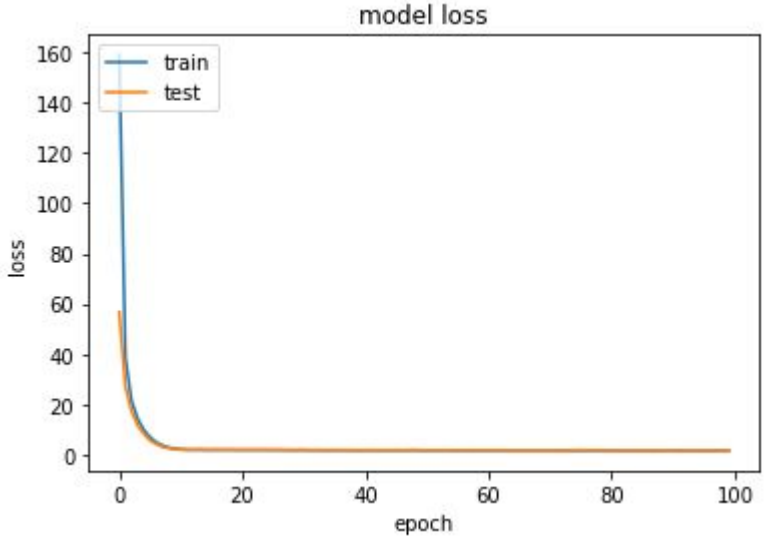
CNN + regularization: 0.1



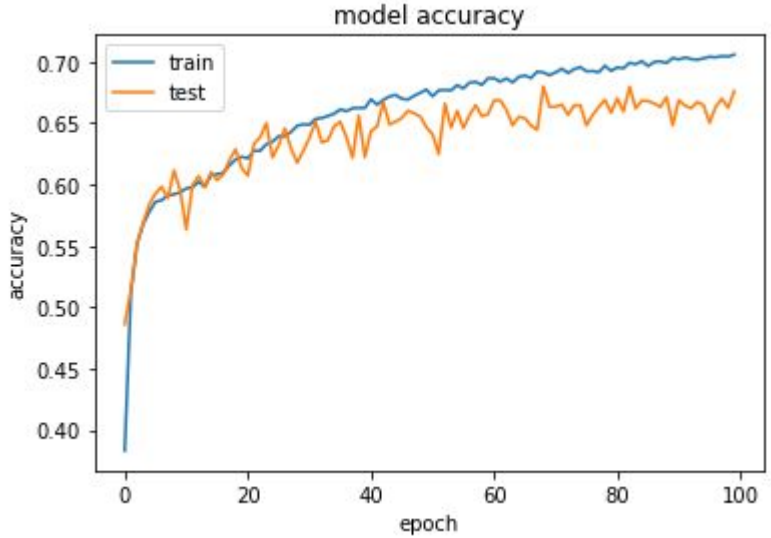


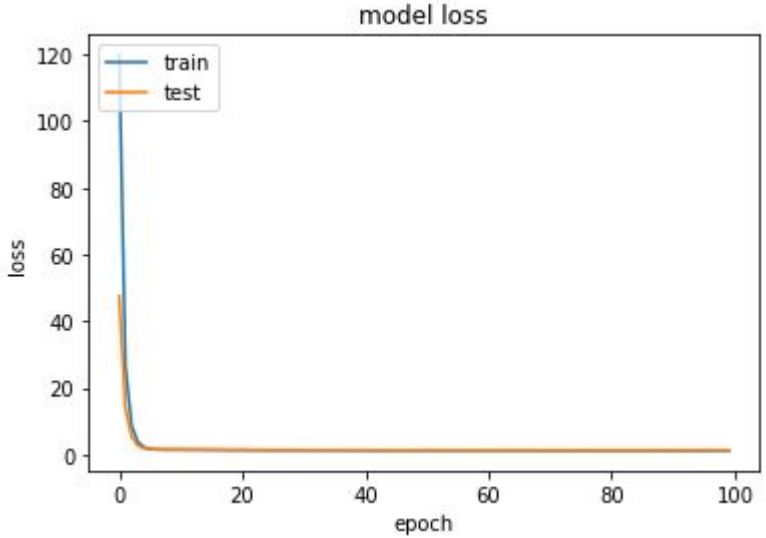
CNN + regularization: 0.1 + filters: 64



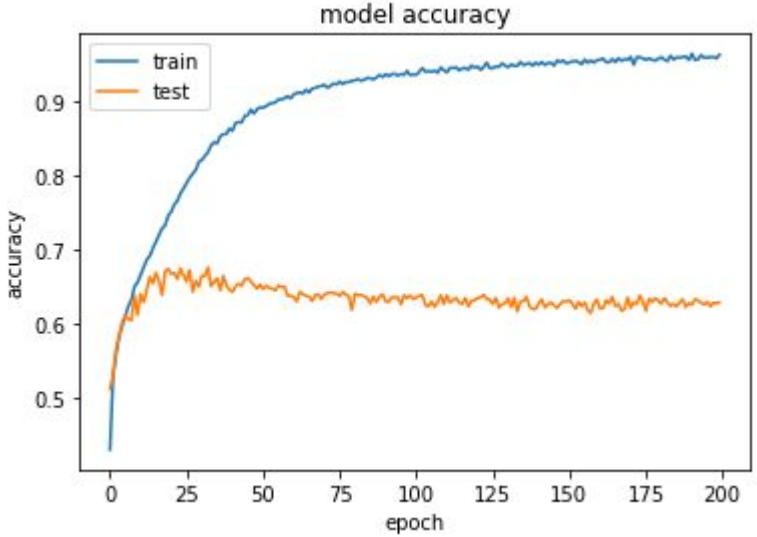


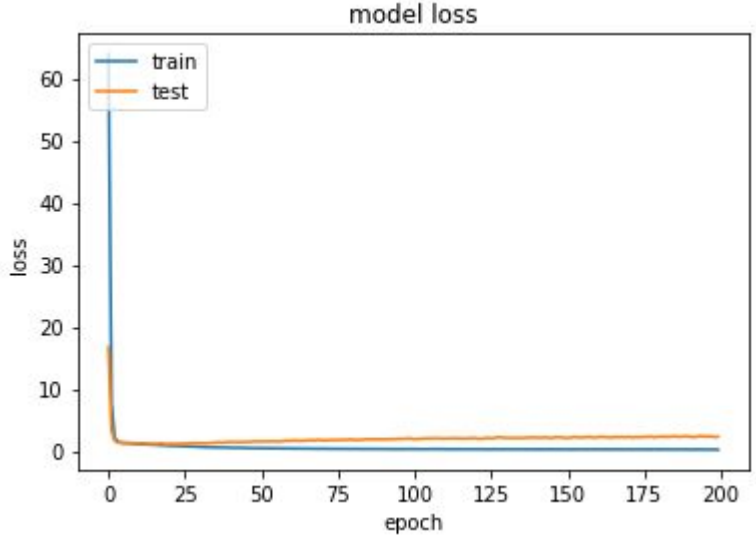
CNN(2conv) - regularization: 0.1 - filters: 32/64



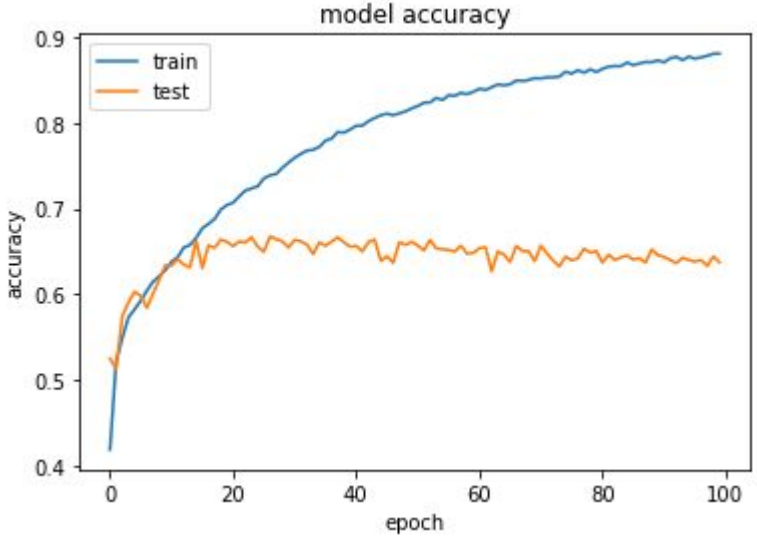


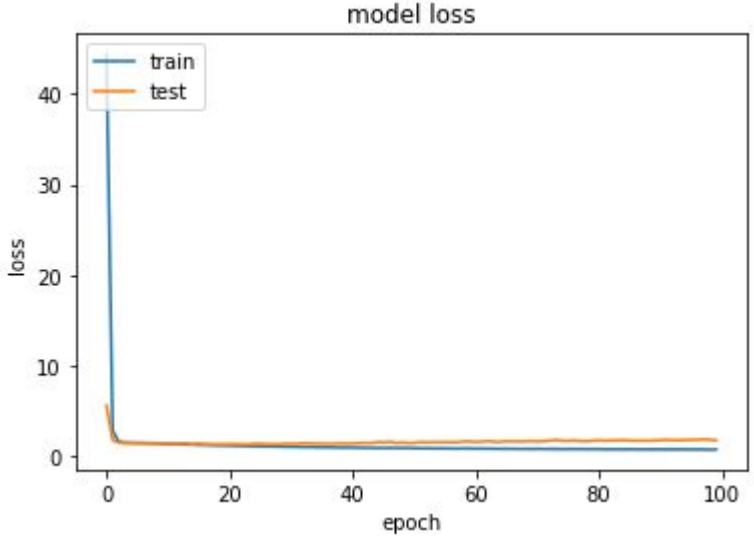
CNN(3conv) - regularization: 0.1 - filters: 32/64/128



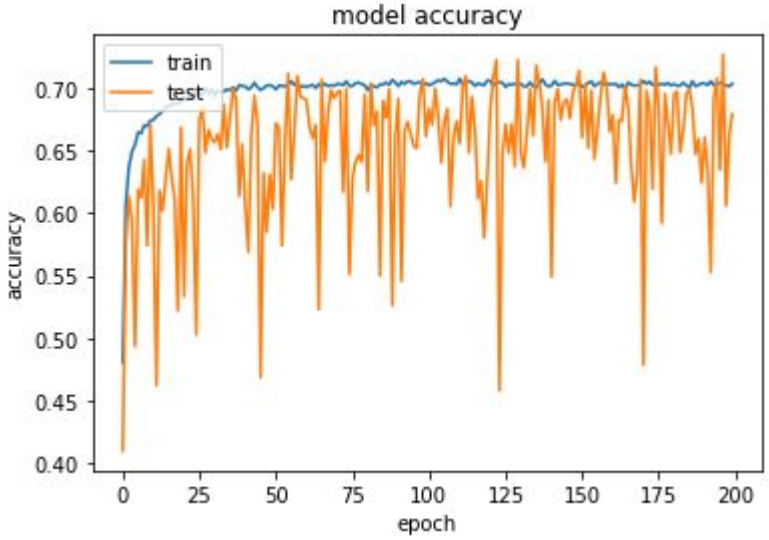


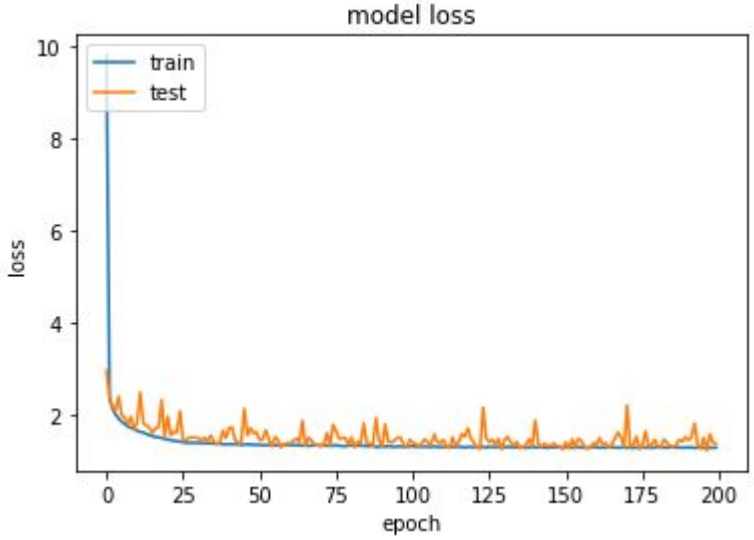
CNN(3conv) - regularization: 0.1 - filters: 32/64/128 - dropout 0.2



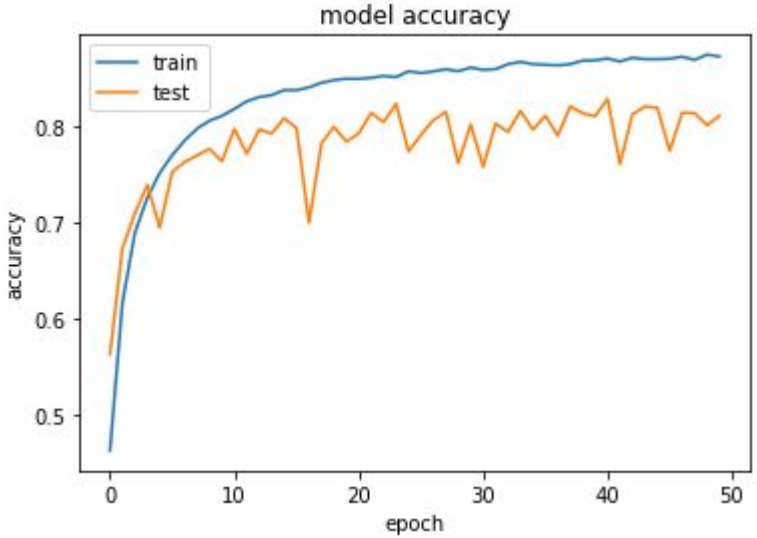


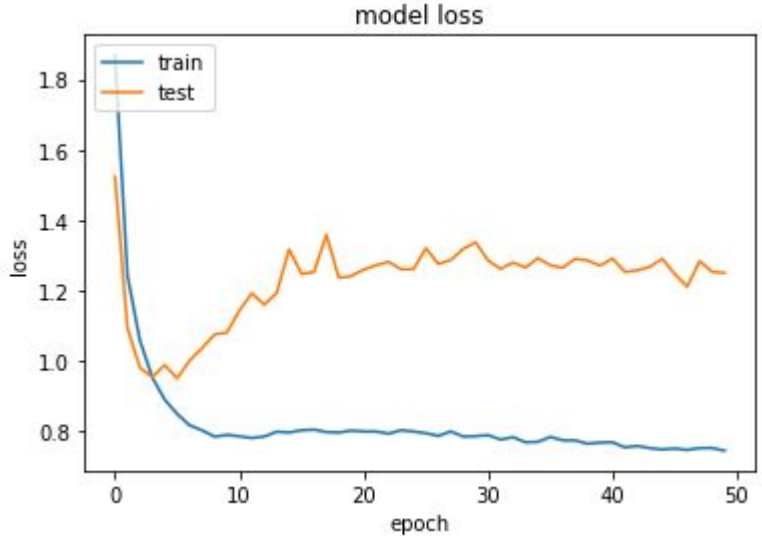
CNN(4conv) - regularization: 0.1 - filters: 32/32/64/64 - dropout 0.2 LR: 0.001



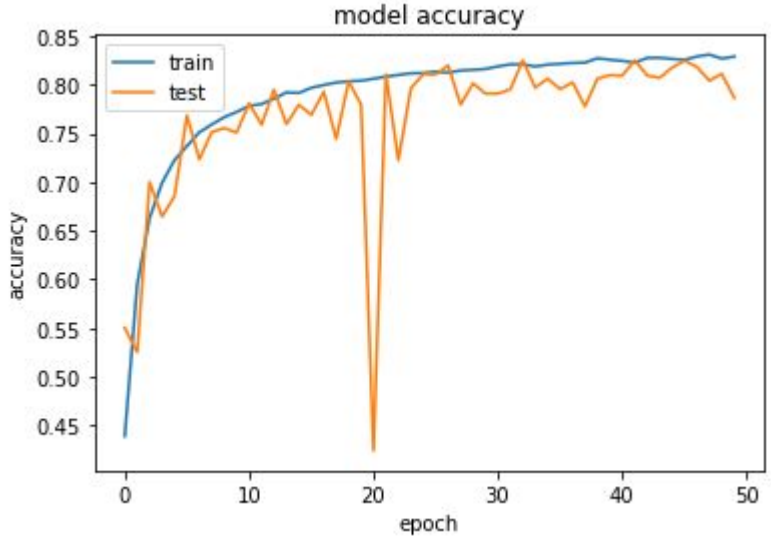


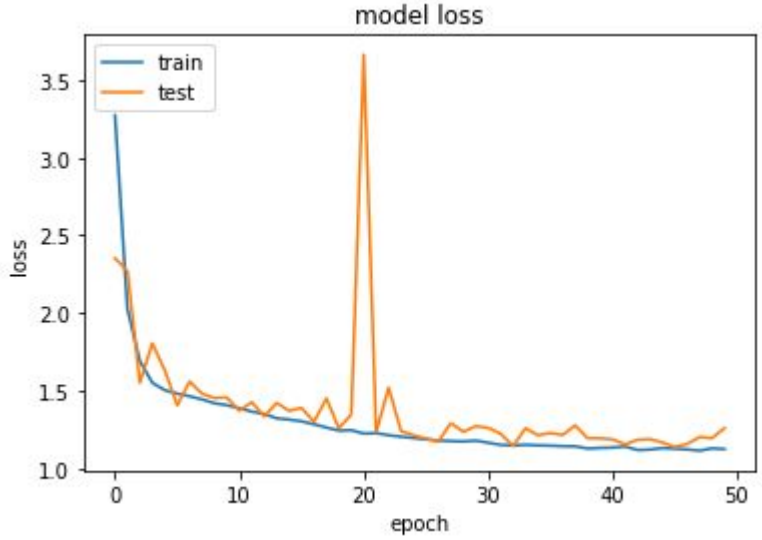
CNN(4conv) - regularization: 0.0001 - filters: 32/32/64/64 - dropout 0.2 LR: 0.001



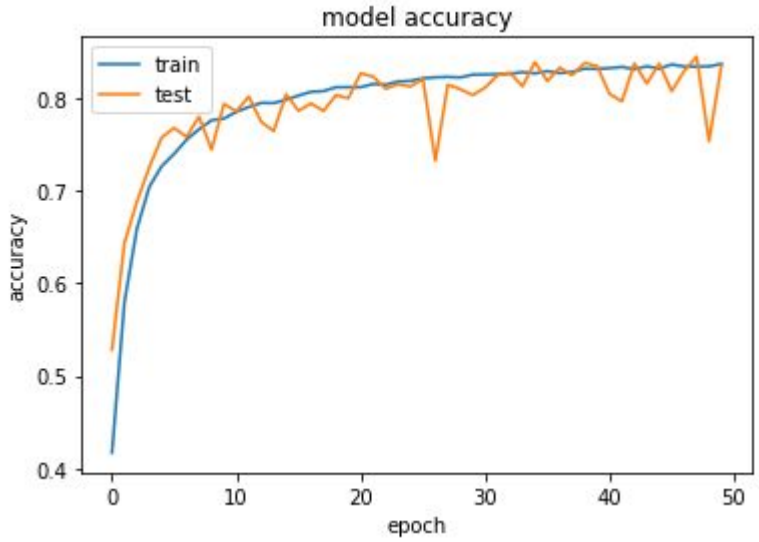


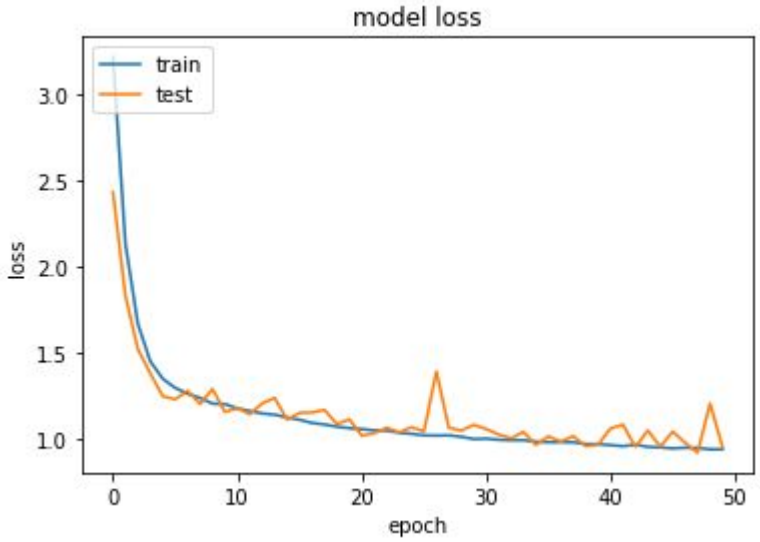
CNN(4conv) - regularization: 0.001 - filters: 32/32/64/64 - dropout 0.4 LR: 0.001



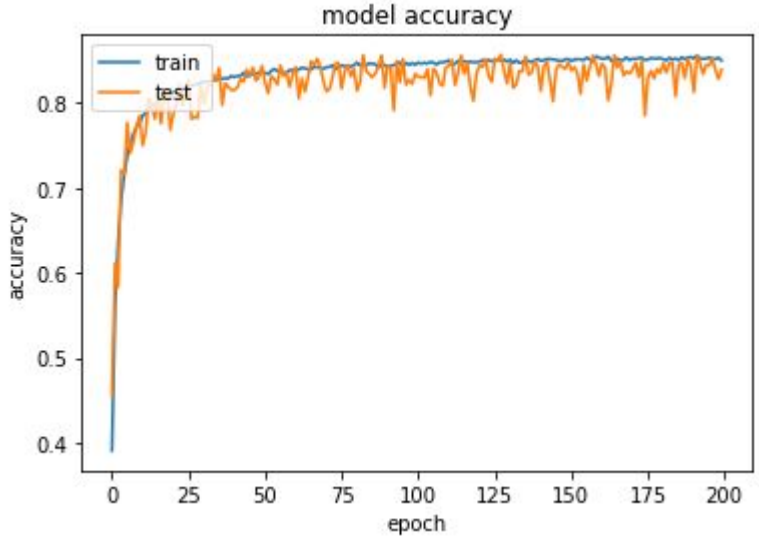


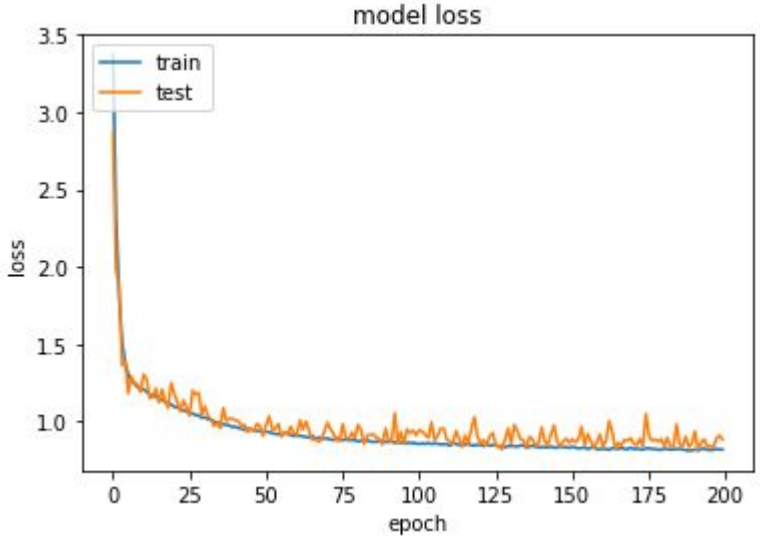
CNN(4conv) - regularization: 0.001 - filters: 32/32/64/64/128/128 - dropout 0.4 LR: 0.001





CNN(4conv) - regularization: 0.001 - filters: 32/32/64/64/128/128/256/256 - dropout 0.4 LR: 0.001

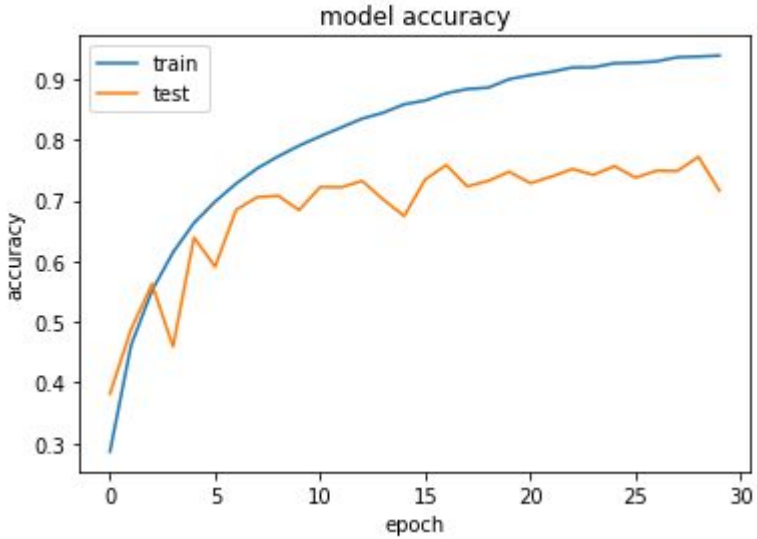


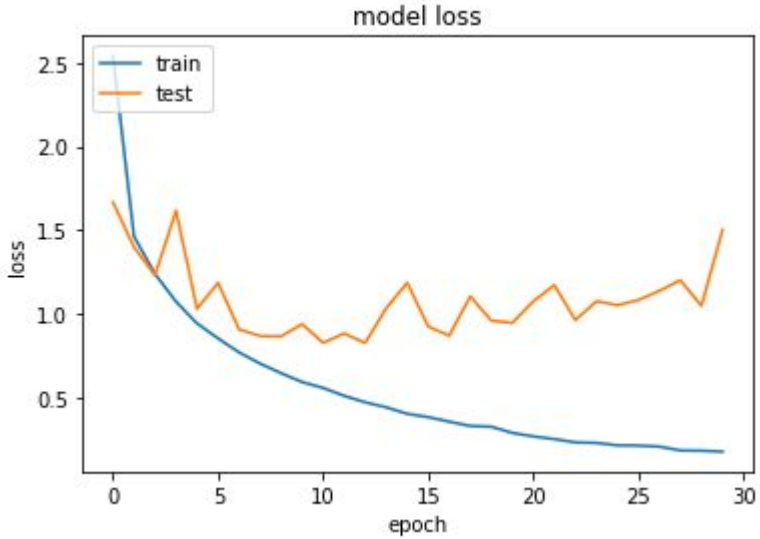


Residual neural network

- Convolution block
- Residual block
- 1 or 2 hidden layers
- 10 classes for the input with Softmax activation function
- optimizer: Adam

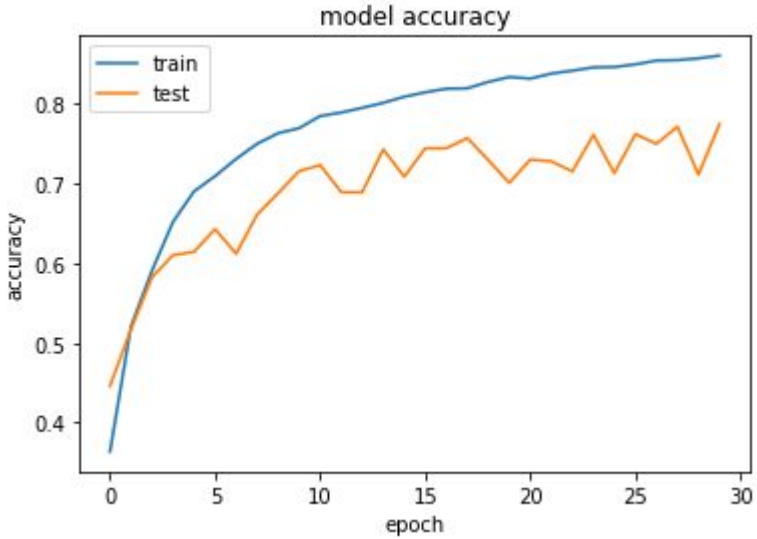
Resnet: 10 residual layers

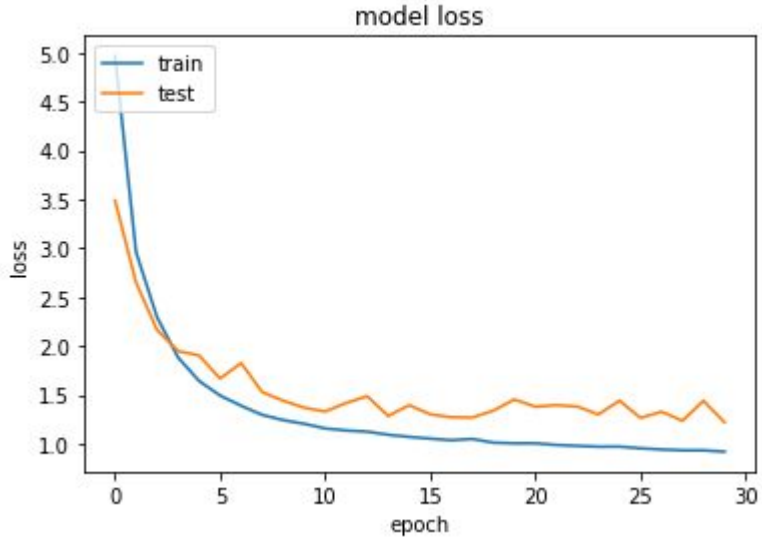




Resnet: 15 residual layers

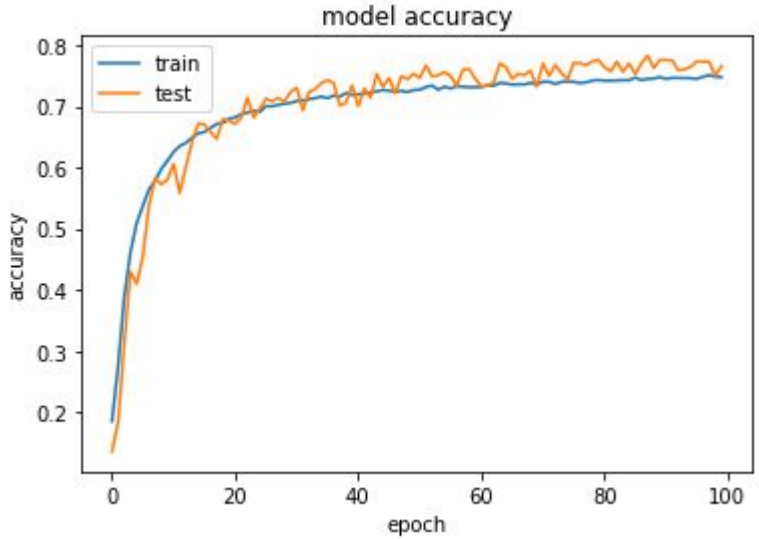
L2 regu: 0.001

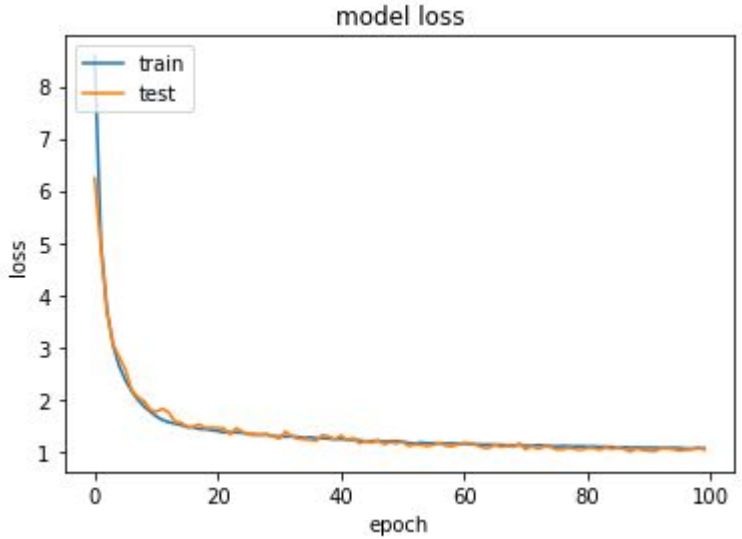




Resnet: 15 residual layers

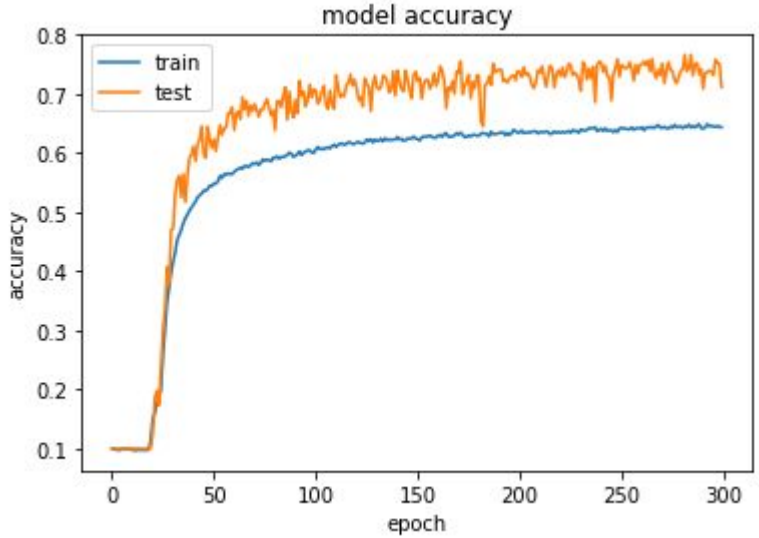
L2 regu: 0.001 + dropout 0.4 + 2 dense

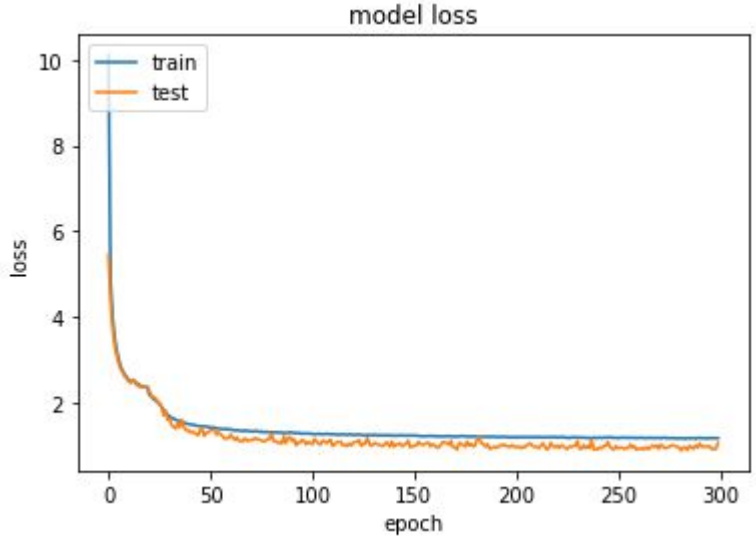




Resnet: 4 conv blocks + 12 residual layers

L2 regu: 0.001 + dropout 0.4 + 1 dense 512





Resnet: 4 conv blocks + 12 residual layers

L2 regu: 0.001 + dropout 0.4 + 1 dense 1024