Brandenburgische
Technische Universität
Cottbus - Senftenberg

Exercise

Visualization CNNs

Exercises

- Use the same train data, test data and the same model structure as in the lecture
- As filter use the array det

```
det = [[[[0]],[[1]],[[0]]],
       [[[0]],[[1]],[[0]]],
       [[[0]],[[1]],[[0]]]]


weights = [asarray(det), asarray([0.0])]
model.set_weights(weights)
```

- Compile the model
- Plot model summary and history
- Plot the output image after the cov2D layer

Exercises

- Read image Mensa.jpg
- Use filters to
  - Smooth
  - Strengthen the contour
  - Strengthen the details
  - Sharpen all edges
  - Sharpen y- or x edges
  - Sharpen the image
  - Have a 3D effect

Exercises

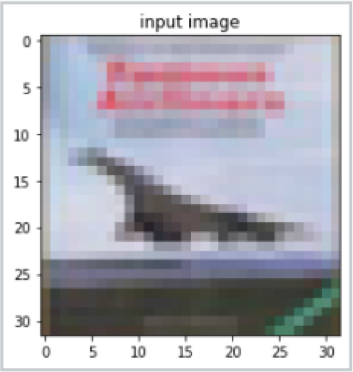
- Read the data from the lecture

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0

class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']
```

- Create the CNN from the lecture
- Add a second convolutional layer with 64 3x3 filters
- Use as last Dense layer a weight matrix 256x10

Exercises

- Do a prediction for the image test_images[3:4]
- Plot the heatmap of the important features



Exercises

- Plot the RGB images and the weights of the first convolutional layer
- Try to interpret the weights
- What for can we use Contrastive Explanation Method (CEM)?

  Get similar images with an alternative prediction

  Get different images with the same prediction
- Train and fit a CEM
- What are the results?