Assignment topic: Neural Network Models for Object Recognition

Master in Artificial Intelligence

Module: Machine Learning

Table of content

1. Introduction

- AI technologies are more and more integrated in daily life in several areas
- Computer vision is considered to be the biggest success story of deep learning
- Deep vision models are part of our life and will be part of our future; e.g. YouTube, PCR software, Google image search, robotics, autonomous driving

2. Data loading and preprocessing

- CIFAR-10 image dataset
- Data normalization
  - Adjust pixel values of images
  - Ensure all input features are on similar scale
  - Makes the training process more stable and efficient
- One-hot encoding
  - Transform the categorical class labels to a binary matrix representation
  - Ensure that labels have been converted into a suitable format for the classification task
  - Ensure that the training and evaluation metrics, such as accuracy, are more interpretable and understandable
- Data split
  - Train – evaluate – test

3. CNN Architecture

- Extracts features from images, reduces their dimensionality, and performs classification using fully connected layers.
- 2 Convolutional Layers
  - First one with 32 filters, size 3x3, applied to input image of shape
  - Second one with 64 filters, size 3x3, applied to the output of the first convolutional layer
- Activation Function (ReLU)
  - Applied after each convolutional layer to introduce non-linearity
- Max Pooling Layers
  - 2 max-pooling layers, both with a pool size of 2x2

3. CNN Architecture

- Flattening
- 2 Fully Connected Layers (Dense)
  - First one with 64 neurons uses ReLU activation
  - Second one with 10 neurons uses softmax activation
- Activation Function (Softmax)
  - Applied in the output layer for multi-class classification

```python
#CNN Architecture
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model= Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(32,32,3)),
    MaxPooling2D((2,2)),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])
```

4. Model Summary

- Layer-by-layer breakdown

```
#Model Summary
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 64)	147,520
dense_1 (Dense)	(None, 10)	650

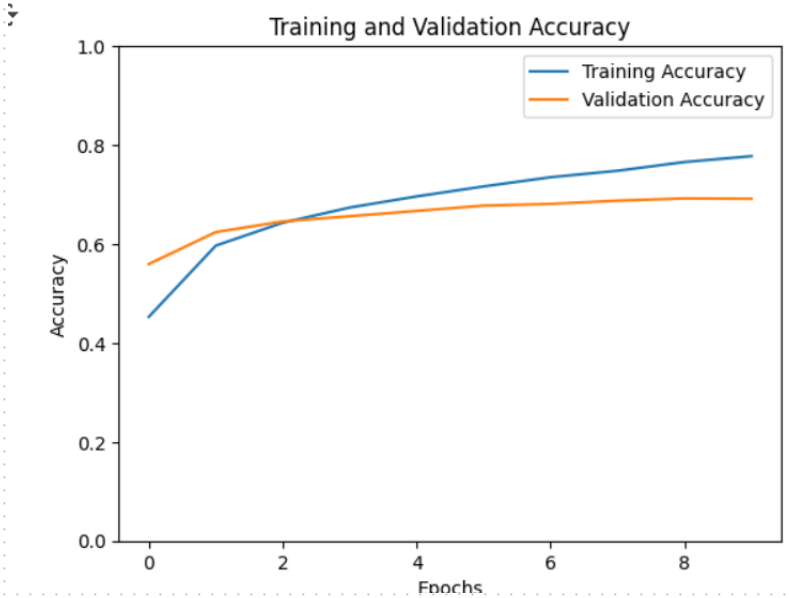
```
Total params: 502,688 (1.92 MB)
Trainable params: 167,562 (654.54 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 335,126 (1.28 MB)
```

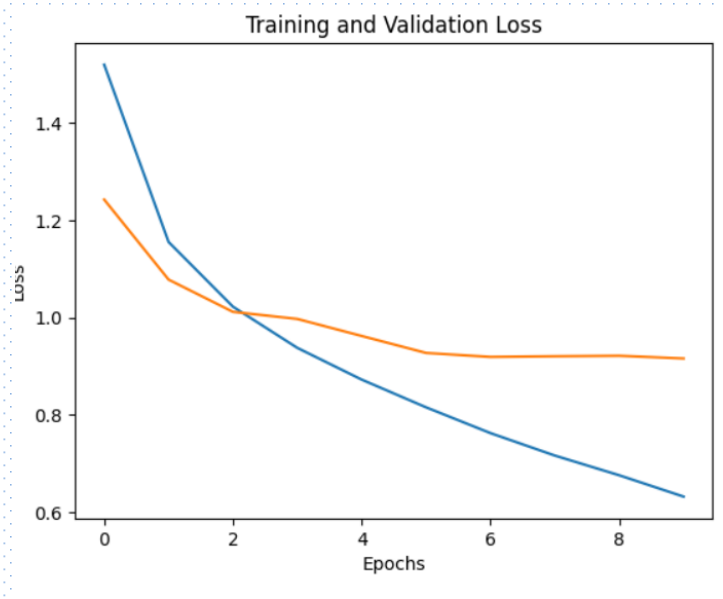
5. Model Compilation

- Configuration of the learning process before training
- Optimiser: Adam
- Loss Function: “categorical_crossentropy”
- Metrics: accuracy

6. Model Training

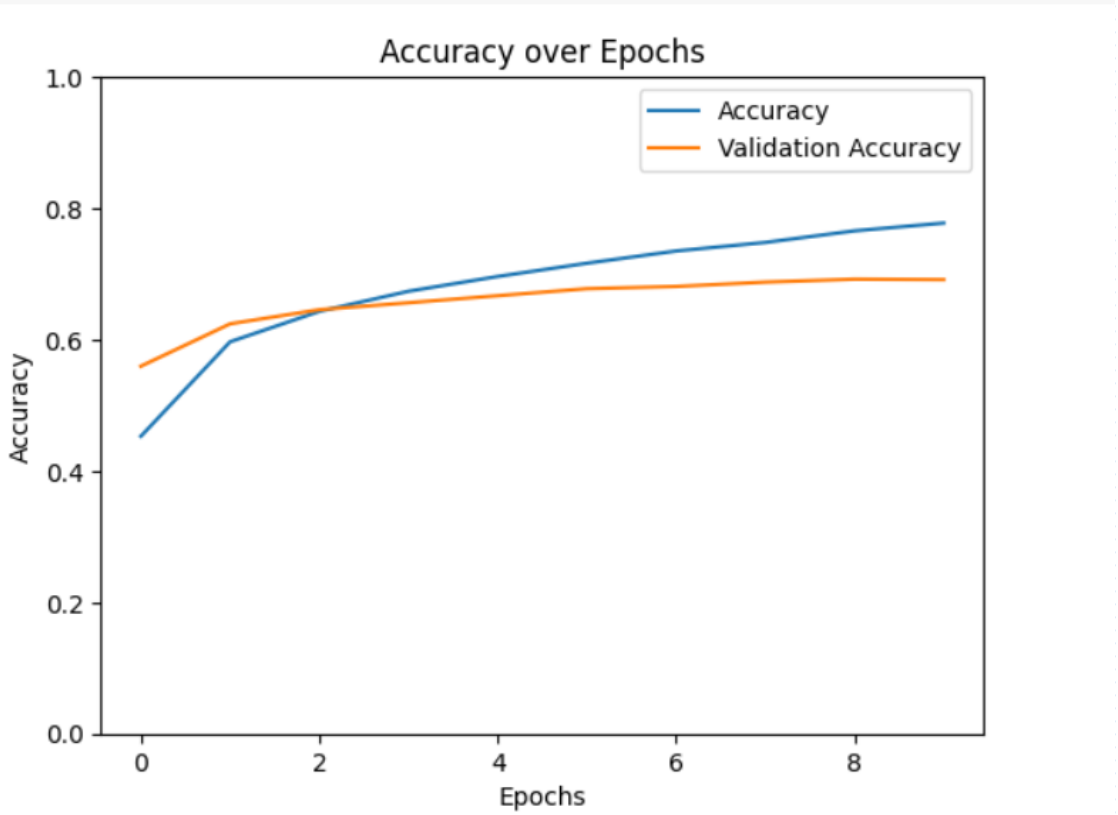
- Process where the model learns from the data to get better
- Epochs: 10
- Batch size: 1250
- Training Accuracy: 78,45%





7. Model Evaluation

- Based on the test dataset
- Accuracy: 69,14%



8. Conclusion

- The developed CNN model learned to classify images from the CIFAR-10 dataset
- Training and validation accuracy provided insights into the model's progress
- Evaluation of the model revealed its generalization capabilities (accuracy: 69%)

9. References

Chollet, F. (2021). Deep Learning with Python . Manning Publications.

Gupta, A. (2023) A comprehensive guide on Optimizers in deep learning, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/ (Accessed: 16 January 2025).

Sharma, S. (2022) Activation functions in neural networks, Medium. Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (Accessed:16 January 2025).

VanderPlas, J. (2022). Python Data Science handbook. O'Reilly Media, Inc.