Report - Functional API and Transfer Learning

Olivier D'Ancona & Magali Egger

Exercice 1 - Functional API

Introduction

This report outlines the experiments conducted using the CIFAR10 dataset to compare different architectures implemented in Keras. The focus was on converting a Sequential model to the Functional API, exploring non-sequential architectures, and utilizing model-saving callbacks during training.

Dataset

The CIFAR10 dataset consists of 60,000 32x32 color images in 10 classes, with 50,000 training imagses and 10,000 test images. The dataset was used to train various CNN architectures to classify the images into one of the ten categories.

Methodology

The experiments were conducted using Keras, a high-level neural networks API that allows for easy and fast prototyping. The CIFAR10 dataset was loaded, and the images were preprocessed by normalizing the pixel values. The models were trained using the Adam optimizer and categorical crossentropy loss function. Furthermore, we used data augmentation in each experiment.

The following models were implemented and compared:

1. **Sequential API Model**: A baseline model using Keras' Sequential API.
2. **Functional API Model**: A conversion of the Sequential model to Keras' Functional API.
3. **Non-Sequential Model with Multiple Features**: Utilizing multiple convolutional paths to extract different features.
4. **Non-Sequential Model with Multiple Paths**: Integrating features at different stages of the model.

Experiment 1: Sequential vs Functional API

- **Objective**: Convert a high-performing Sequential model to the Functional API and compare performance.

Experiment 2: Multiple Path

- **Objective**: Explore architectures that use multiple paths or feature extractors to enhance learning capability.
- **Approach**:
  - **Model 2**: Incorporated two different convolutional paths with varying kernel sizes to capture features at different scales.

Experiment 3: Multiple Features

- **Objective**: Mix the features from different stages of the CNN
- **Approach**:
    - **Model 3**: Designed 3 feature extractors at different stages of the model and combined them before the classification layer.

Results

Model Description	Training Accuracy	Validation Accuracy	Test Accuracy	Remarks
exp1_sequential	0.5715	0.5562	0.6071	Sequential model performance.
exp1_functional	0.5696	0.5688	0.6188	Converted to functional API.
exp2_functional	0.4477	0.4680	0.5116	Multiple Path model
exp3_functional	0.5095	0.4883	0.5307	Multiple Features.

Conclusion

The transformation from a Sequential to a Functional API model was successful, with no loss in performance, validating the flexibility of the Functional API. The experiences with multiple paths and features were insightful, but the models did not outperform the Sequential or Functional API models. Further experimentation and tuning are required to optimize these architectures. Furthermore, this is surprising that the scores are so low when we compare with the previous lab. We think that could be due to data augmentation that distorts the images too much. In a future experiment, we could try to run the notebook without data augmentation to see if the scores are better. However, a sure thing is that the data augmentation is useful to avoid overfitting as seen on the graph in the notebook.

Exercice 2 - Transfer Learning

a) Review the lecture slides on using Keras for a transfer learning task.

=> Done b) Chose one of the architecture presented here : https://keras.io/applications/. Be- ware that some architectures are using large memory and lots of cpu (so better move to gpu) !

=> We chose the EfficientNetV2B0 architecture as it is the best ratio between performance and low weight.

c) Download the provided notebook ex2_transfer_learning_stud.ipynb.

=> Done

d) Implement the preprocessing and the training phases.

=> Done following the tutorial

e) Experiment with one or several architectures with different classification heads (hidden layers, ...).

f) Describe the results in your report which architecture and parameters works best

We were unable to test further, due to a lack of time and problem with tensorflow_datasets. However, we did run the notebook in the colaboratory and it worked fine. We could try to download the dataset from another source in a future experiment.