POLITECNICO
MILANO 1863

Artificial Neural Networks and Deep Learning
Homework 1: Leaf Images Classification

Team: Students:
Hessian Learners Lodari G., Hashemian A., Scaramuzza F.

A.Y. 2021/2022

To better organize this report, we decided to split our work in two phases. In the first one (Section 1), we put our effort in improving the accuracy of a model designed by scratch, adjusting its hyperparameters. In a second phase (Section 2), we focused on a Transfer Learning model, adjusting the results with fine tuning.

Before the actual implementation of the model, we took a look at the dataset. We noticed that this was extremely unbalanced and that the model would not be able to generalize adequately. To solve this first issue we adopted the "stratification" technique for the train and validation set splitting.

1 First Phase

1.1 First Set-Up

Our first attempt, and first submission, was a model designed by scratch made of 5 **Convolutional Layers**, interspersed with **Max Pooling** layers. The last 3 layers were a **Flattening Layer**, and two **Dense Layer**. The latter two were composed of 512 and 14 neurons respectively and where interspersed with **Droupout layers**.

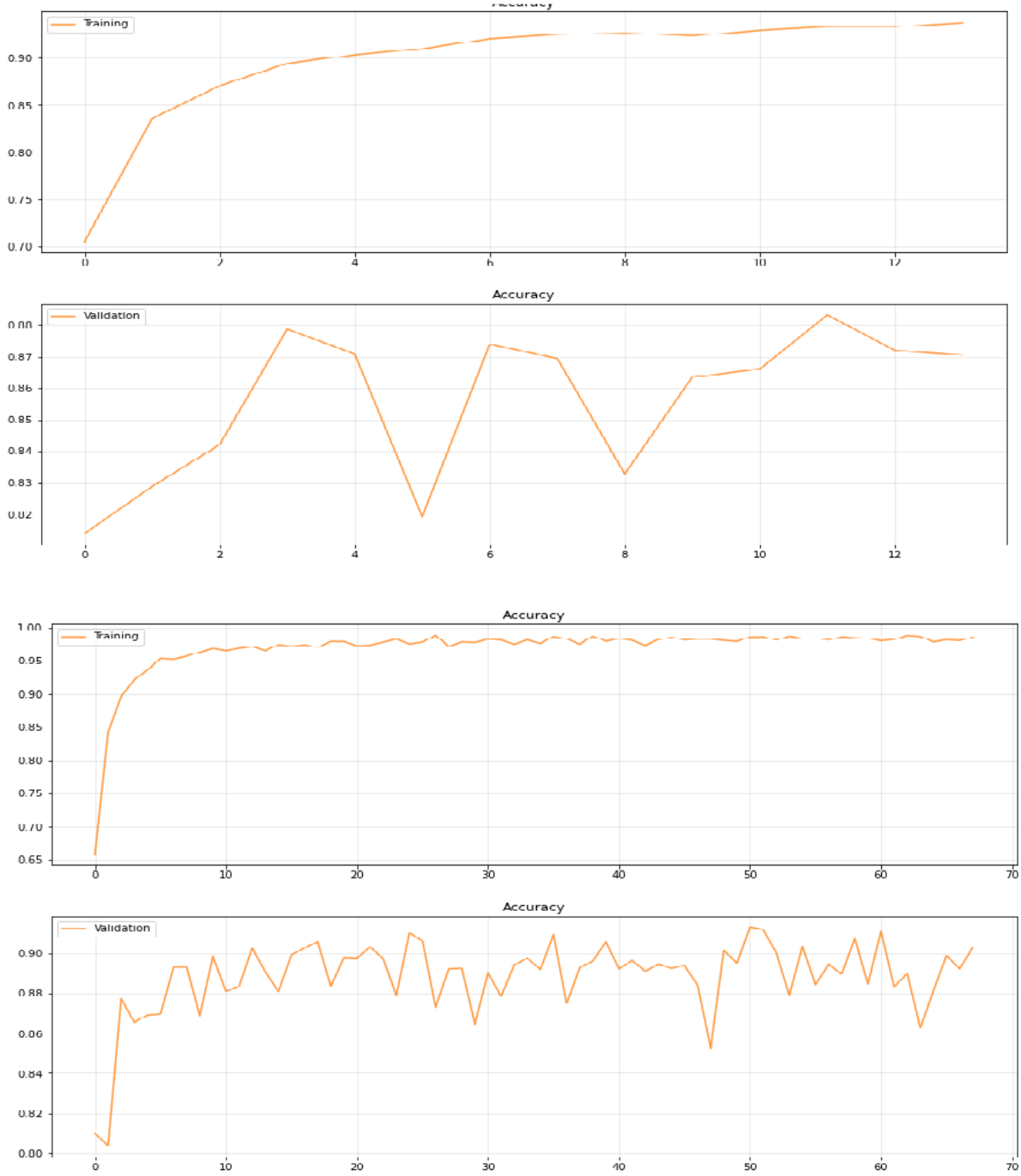
To better generalize, we have also adopted the technique of "Data Augmentation" with classic transformations such as zoom, rotation and so on.

In every layer we used the ReLu activation function, while in the last one we used SoftMax.

Before switching to a another model architecture we tried to adjust batch sizes and other hyperparameters like the learning rate. Improvments were not anyway acceptable and our accuracy on the test set on Codalab was extremely low as the training loss remained high.

1.2 First Big Improvments

Our first major improvement came with the adoption of a CNN with a VGG16 architecture with which we obtained an accuracy on the test set of $\approx 50\%$, reaching $\approx 57\%$ thanks to the normalization of the input data and with the use of `class_weights`.



2 Second Phase

After different tests with different configurations we noticed that we couldn't get any improvments. At this point we decided to implement Transfer Learning using VGG19 model trained on `imagenet`.

```
def build_model(input_shape):
  vgg = tf.keras.applications.VGG19( include_top=False,
                    weights="imagenet",
                    input_shape=input_shape
                    )
  for layer in vgg.layers[:-2]:
    layer.trainable = False

  model = tfk.Sequential()
  model.add(vgg)
  model.add(tfkl.BatchNormalization(axis=-1))
  model.add(tfkl.Conv2D(128, (3, 3), padding="same"))
  model.add(tfkl.Activation("relu"))
  model.add(tfkl.MaxPooling2D(pool_size=(2, 2)))
  model.add(tfkl.Dropout(0.2, seed=seed))
  model.add(tfkl.Flatten())
  model.add(tfkl.BatchNormalization())
  model.add(tfkl.Dense(units=256,kernel_initializer = tfk.initializers.GlorotUniform(seed)))
  model.add(tfkl.Activation("relu"))
  model.add(tfkl.Dropout(0.2, seed=seed))
  model.add(tfkl.Dense(units=14, activation="softmax",kernel_initializer = tfk.initializers.
    ↪ GlorotUniform(seed)))

  # Compile the model
  model.compile(loss=tfk.losses.CategoricalCrossentropy(), optimizer=tfk.optimizers.Adam(),
    ↪ metrics='accuracy')

  # Return the model
  return model
```

Listing 1: VGG19 implementation with Transfer Learning

At this point the accuracy of the model on the test set was $\approx 65\%$.

Since the dataset, and specially some classes, where made of very few images, we focused on Fine Tuning the model we got so far instead of introducing too much parameters to the network. With Fine Tuning and normalization, the accuracy we got was $\approx 81\%$.

2.1 Train and Validation Dataset Splitting

The validation set was the 10% of the entire dataset in every model we trained so far. To get an idea of how much it influenced our results, we tried different percentages, like a splitting

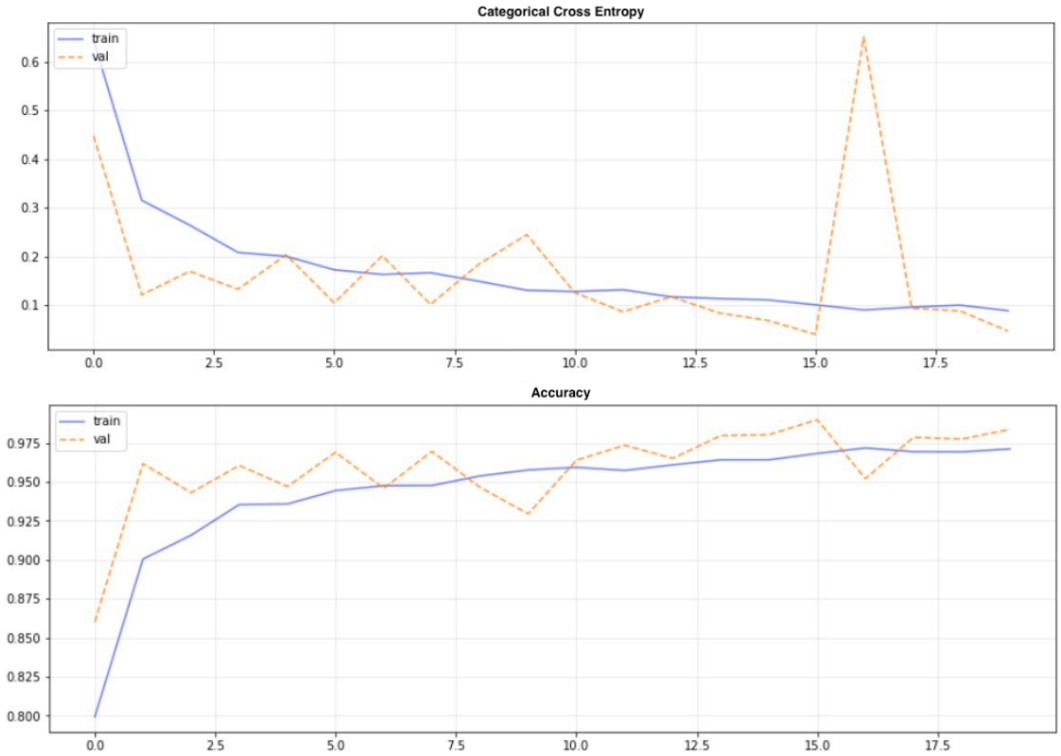


Figura 1: Categorical Cross Entropy and Accuracy on Test and Validation Sets with 20 epochs

of train and test set of 80/20, 60/40 and 50/50. These tests pointed out that our first ratio was the best one.