Deep Learning Report

Classification of dogs and cats using CNNs

Varez Arthur
DIA4
14 May 2021

The dataset

I choose to work with a dataset of images of cats and dog founded in Kaggle (cf : https://www.kaggle.com/kunalgupta2616/dog-vs-cat-images-data). This dataset is composed of 45.5K images : 25000 for the training set, 8000 for the validation set and 12500 for the test set. Test and validation datasets are equally divided of two parts for each label(dog or cat). As the labels for the test set were not given (so i couldn't get the accuracy on this set) I decided to divid the train set into a smaller train set and a validation set. The given validation set will be used as a test set.

The goal is simple, we're trying to classify whether an image represents a dog or a cat. To do so, I'll use CNNs.

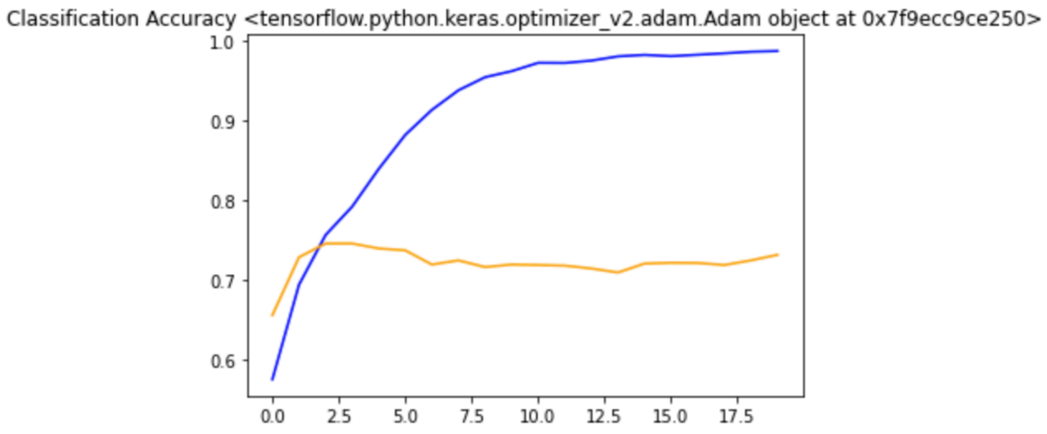
The Neural Networks

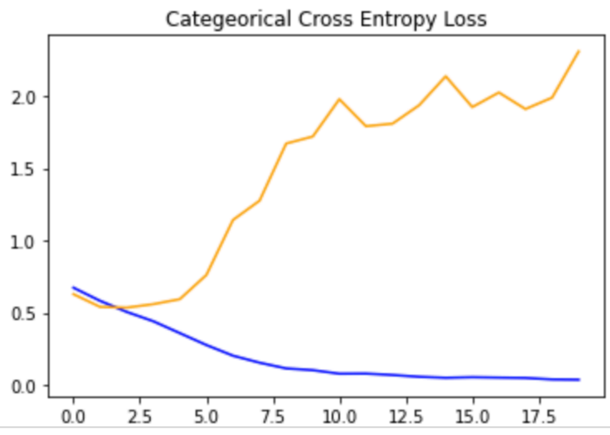
Firstly I decided to go for a simple neural network :

```
model = tf.keras.Sequential([
    layers.experimental.preprocessing.Rescaling(1./255),
    layers.Conv2D(128,4,activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64,4, activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32,4, activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(16,4, activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])

model.compile(optimizer='adam',
              loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
  metrics=['accuracy'])
```

This model is composed of 4 convolutional layers, 4 layers of Maxpooling. I chose to use the Adam optimizer, because it usually gives good results on 20 epochs (which is more than enough as the accuracy of the model on the validation set tends to stay the same). So after the training phase I've been able to have this graph representations :





We can see that our model quickly converges on a validation accuracy of about 75%.

There's also an issue with the validation loss which gets higher and higher. It could be a clue to conclude that our model is overfitting.

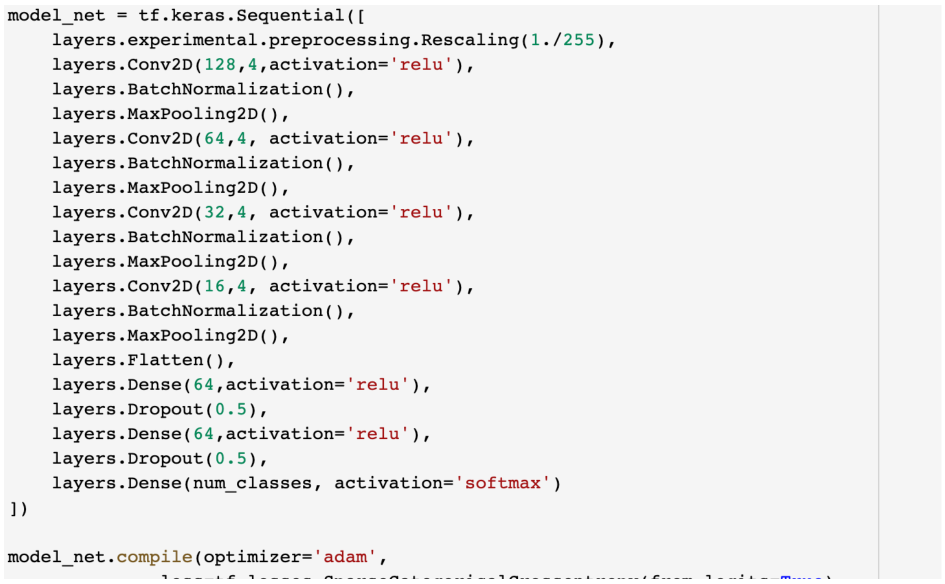
The fact that the validation accuracy does not increase even though the validation loss does increase must come from the fact that when the model has learned the train set enough, the loss will increase for already misclassified samples.

Here are the results on our test set :

```
250/250 [==============================] - 23s 89ms/step - loss: 0.5439 - accuracy: 0.9155
[0.5439342856407166, 0.9154999852180481]
```

That is not so bad but I think we could do better.

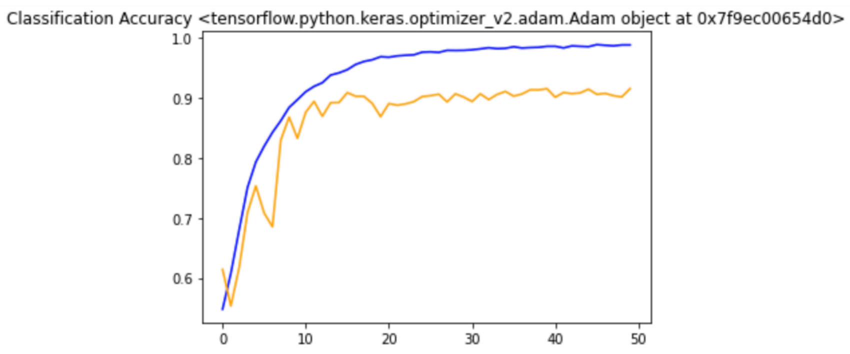
So, I've tried to build another model based on AlexNet architecture which gives could results for image classification. In this model we had layers of BatchNormalization which normalise each batch and also dropout layers to eliminate random neurons.

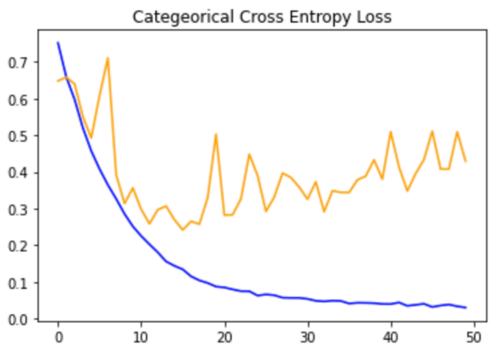


```
model_net = tf.keras.Sequential([
    layers.experimental.preprocessing.Rescaling(1./255),
    layers.Conv2D(128,4,activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling2D(),
    layers.Conv2D(64,4, activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling2D(),
    layers.Conv2D(32,4, activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling2D(),
    layers.Conv2D(16,4, activation='relu'),
    layers.BatchNormalization(),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(64,activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])

model_net.compile(optimizer='adam',
```

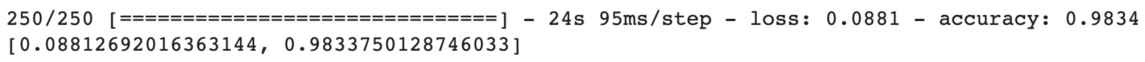
I also chose Adam optimizer to compare my two models on the same optimizer but I tried this model on 50 epochs.





We can see that we have better performances (higher accuracy, lower loss) and we fixed to issue about overfitting we had with our first model.

Here are the metrics on the test set :



```
250/250 [==============================] - 24s 95ms/step - loss: 0.0881 - accuracy: 0.9834
[0.08812692016363144, 0.9833750128746033]
```

It's a significant improvement compared to the first Neural Network.