Mask CNN

Datasets

The first task we had to face was splitting the data into train and validation sets. To do this we created our own script that makes the split based on a desired ratio (80% to 20%) to preserve the proportion of the samples for every class in both the datasets.

Custom Model

Our first block of tests was focused on finding a full custom network with standard convolutions and dense layers at the top for the classification. With this approach we actually made good results (like 0.87 in the test set) but we couldn't get any better result. To achieve this first result we used a lot of data augmentation and regularization and tried out a really huge number of models with different parameters and order of layers. After trying different depths, number of filters for each layer and kernel size we remained stuck in overfitting at a maximum of 0.88%-0-89% of validation accuracy.

We spent a lot of time tuning the hyperparameters of our custom model but breaking the 90% wall was a really hard objective to reach. We also noticed that it was hard to replicate the same results over different "train runs". We had to search for another way of solving this problem.

We spent 8-9 days on this kind of models and we messed up almost anything, we faced a lot of errors in setting parameters and thus we saw a lot of underfitting and overfitting.

Transfer Learning

After we failed with custom models we explored the idea to use transfer learning to improve our results. We tried out almost any pre-trained network that keras offers.

At first we tried to freeze the pre-trained weights and attach our dense layers with Flatten layers between them. This approach gave us average results that required a lot of regularization with dropout layers. In this case we also noticed that some networks (eg: Xception) tended to learn the dataset so well that it achieved 90% accuracy on the train set in just a few epochs but they did not generalize well on validation data.

To have better performance we tried to fine tune the classifier part. Even in this case the network was really good in overfitting the training set with really poor performances in generalization.

Searching in different papers we noticed that a lot of powerful models (Xception, Inception, ResNet) used Global Average Pooling instead of a Flatten layers and had really good results on ImageNet so we decided to try to use the same approach and this led us to our final version.

Current Model

Our final model reached 89% in accuracy in the test set (~91% in validation), this is a better result but there is room for improvement. The actual model can easily reach 99% accuracy on the train set with a lot of data augmentation. This means that the model could learn a lot more but we did not manage to find a good regularization balance to make it generalize well.

This model is a partially re-trained (last 20% of layers) Xception classifier (loading imageNet weights) without any dense layers. We found out that Xception is so powerful for our problem that a Global Average Pooling layer followed by the output neurons was enough to have really good results.
By removing the dense layers we obtained a classifier with less hyper-parameters to tune and also less parameters to train, all we had to do was fine-tuning only a part of Xception.
To make the network generalize we played a lot with data augmentation to confuse our classifier and force it to generalize.
With this in mind we also discovered that by reducing the batch size we could increase the gradient noise. This means that by having little batch size, the model could generalize more.
One last parameter we've tuned a lot was the learning rate, we've tried a lot of combinations with higher or lower rates but in our case we found out that setting the rate to 1e-4 worked best because it gave us good results with decent training times.

We tried out some combinations of data augmentation, fine-tuning layers, learning rate but we ran out of time and could not find a good balance in hyper-parameters to force the network to generalize even further.

Consulted Papers

A Survey of the Recent Architectures of Deep Convolutional Neural Networks
Asifullah Khan1, Anabia Sohail1, Umme Zahoora1 and Aqsa Saeed Qureshi.

Xception: Deep Learning with Depthwise Separable Convolutions
François Chollet