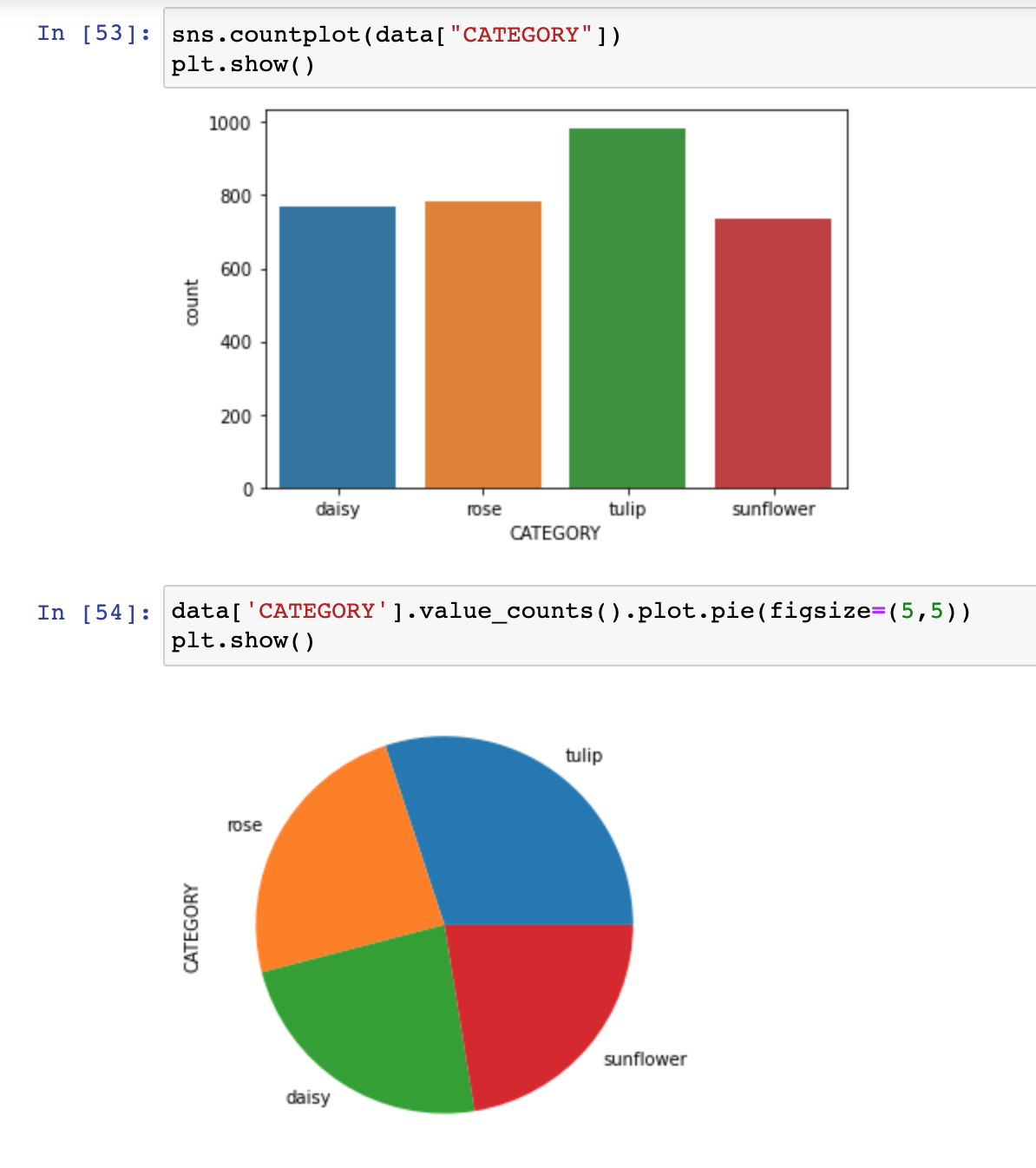
Deep Learning project documentation

For this project I compared these two architectures: Convolutional Neural Networks and Recurrent Neural Networks to observe the difference between them on classifying a set of pictures.

So, in the first part i checked some pictures from the set and plotted some exploratory plots to get a hold of the set. As I expect the pictures in the set are not standardized on the same size and some of them are not even representing what they are supposed to.

The set contains images of flowers and provides us with 4 classes: daisy, rose, sunflower and tulip. So as I said the first part of the project is about loading, exploring and building up our data for the models.

 As we can observe, the distribution between classes is pretty much the same so we don't have to worry about an unbalanced set.

So given the fact that the pictures are not the same size and not so clear on our focus, I parsed them through tensorflow keras image generator. By doing this, I also split the train set into train and validation, added some rotation on the pictures, zoom range and enhanced a little bit the training set.



This first picture is the original photo and the second is one of the results.

.

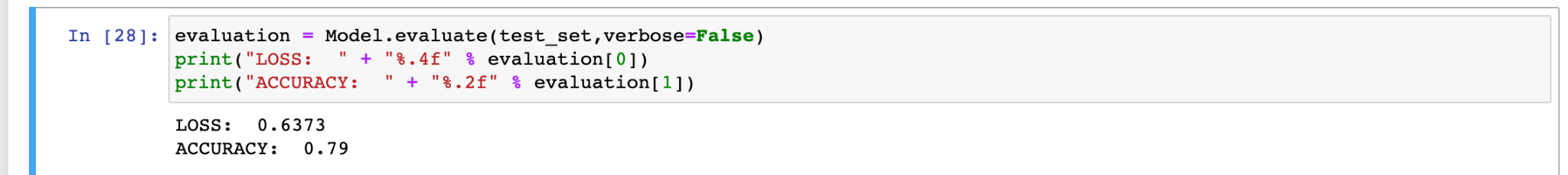
1. **CNN**

The first architecture that i’ve used is a simple convolutional neural network.

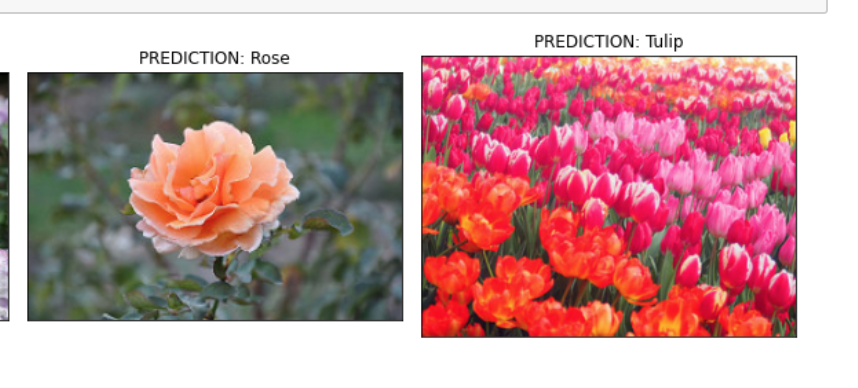


Every convolutional layer is followed by a maxPooling layer. At the end I added flattening since we need the last layer to be fully connected for classification. Also added dropout to try to prevent overfitting and the last layer is based on the number of classes. I used Adam() optimizer and ‘categorial\_crossentropy’ for the loss.

So, I trained the model, using a validation set, for 35 epochs. The values I used are based mainly on try and error, 35 epochs seemed to be the sweet spot. I got 86% accuracy in training with 0.33 loss which is great. I must mention this is the final form of the model. At first I tried different optimizers, no. of epochs, had another layer and I was around 70% accuracy on the test. At 50 epochs the model was overfitting.

After evaluating the model I obtained around 80% accuracy and 0.6 loss which I was satisfied with.

And here we have some results:



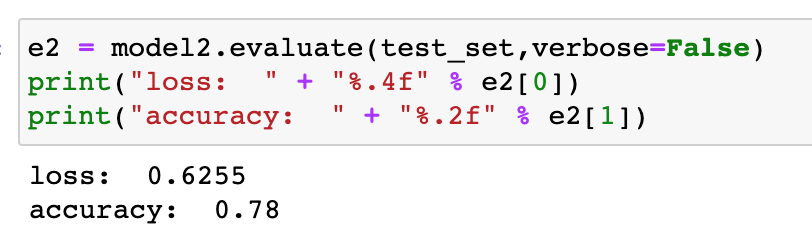
1. **RNN**

In the second part of the project I tried to use a recurrent neural network architecture to try to classify the flowers. I tried to stick to the first architecture as much so the differences would be more notable and valid. I used the same numbers of layers and optimizers. The differences consist of adding batch normalization layers and the most important one: the bidirectional lstm layer before the output layer. This is where the recurrency stands in.



The train, test and validation sets were the same for both models.

So, after training the second model we obtained an accuracy of around 80% which is similar to our first model and it's still ok.



Differences between models:

