UDACITY

Return to Classroom

DISCUSS ON STUDENT HUB

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Excellent work with the project! Yours is probably one of the best submissions I have come across till now.

Here are some resources if you wish to explore CNN and its application:

- My post about School of AI NDs
- 9 Deep Learning papers
- MIT AGI: Building machines that see, learn, and think like people
- YOLO Object Detection
- The AlphaGo Movie
- Inception Network Overview
- Research Paper Xception network
- Weight Initialization Techniques in Neural Networks

Congratulations on Finishing the Project !!! 👍

Files Submitted

The submission includes all required, complete notebook files.
Thank you for submitting all the require files for submission.

Step 1: Detect Humans

The submission returns the percentage of the first 100 images in the dog and human face datasets that include a detected, human face.

Excellent job on implementing a function which detect the human faces in an images and achieving a 98% accuracy on human images

```
There are 98.0% human faces detected in human_files.
There are 17.0% human faces detected in dog_files.
```

Step 2: Detect Dogs

Use a pre-trained VGG16 Net to find the predicted class for a given image. Use this to complete a `dog_detector` function below that returns True if a dog is detected in an image (and False if not).

Excellent work on utilizing VGG16 for detecting dog in an images.
Try some experiment by using Resnet, Inception and Xception net and check how accuracy will improve

The submission returns the percentage of the first 100 images in the dog and human face datasets that include a detected dog.

Excellent job on implementing a function which detect the dogs in an images and achieving a 100% accuracy on dog images and no false positive on human images

```
There are 0.0% dog images detected in human_files.
There are 100.0% dog images detected in dog_files.
```

Step 3: Create a CNN to Classify Dog Breeds (from Scratch)

Write three separate data loaders for the training, validation, and test datasets of dog images. These images should be pre-processed to be of the correct size.

You have correctly written the three separate data loaders for the training, validation, and test datasets of dog images.

Read the blog by Sourav Kumar on Data Augmentation Increases Accuracy of your model — But how

Answer describes how the images were pre-processed and/or augmented.

You have correctly provided the answer

The submission specifies a CNN architecture.

Nice work, your model comprises of convolution layer, pooling layer, fully connected layer, dropout regularization layer, Batch Normalization along with relu activation

Couple of Suggestions :

- Consider using ELU or LEAKY_RELU activation function
- Consider adding weight initialization parameter in conv2d operation
- Set dropout rate between 0.4-0.5

Answer describes the reasoning behind the selection of layer types.

You have correctly provided the answer, here is the summary and more clarification of your answer :

- Convolution layer which is used to extract useful features from images like edges, corners etc.,
- Dropout is used as regularization to avoid overfitting
- Batch Normalization perform scaling/shifting of normalized mean and variance of mini batch
- Relu activation function convert your linear data to non linear form
- Max Pooling layer is used to downsample the images

Choose appropriate loss and optimization functions for this classification task. Train the model for a number of epochs and save the "best" result.

Nice job on using CrossEntropy loss function and SGDOptmizer

Guideline for selecting an optimizer for training neural network:

1. AdaGrad penalizes the learning rate too harshly for parameters which are frequently updated and gives more learning rate to sparse * parameters,parameters that are not updated as frequently. In several problems often the most critical information is present in the data that is not as frequent but sparse. So if the problem you are working on deals with sparse data such as tf-idf,etc. Adagrad can be useful.
2. AdaDelta,RMSProp almost works on similar lines with the only difference in Adadelta you don't require an initial learning rate constant to start with.
3. Adam combines the good properties of Adadelta and RMSprop and hence tend to do better for most of the problems.

4. Stochastic gradient descent is very basic and is seldom used now. One problem is with the global learning rate associated with the same. Hence it doesn't work well when the parameters are in different scales since a low learning rate will make the learning slow while a large learning rate might lead to oscillations. Also Stochastic gradient descent generally has a hard time escaping the saddle points. Adagrad,Adadelta,RMSprop and ADAM generally handle saddle points better. SGD with momentum renders some speed to the optimization and also helps escape local minima better.

The trained model attains at least 10% accuracy on the test set.

Excellent work your model attain the accuracy of 11% on test set.

```
Test Loss: 3.925461

Test Accuracy: 11% (92/836)
```

Step 4: Create a CNN Using Transfer Learning

The submission specifies a model architecture that uses part of a pre-trained model.

Excellent choice of selecting Resnet50 network for transfer learning

The submission details why the chosen architecture is suitable for this classification task.

You have correctly provided the answer of Question-5

Train your model for a number of epochs and save the result wth the lowest validation loss.

Excellent job on training the model for 30 epochs

Accuracy on the test set is 60% or greater.

Excellent work your transfer learning model achieve a accuracy of 76% on test dataset.

```
Test Loss: 1.391905

Test Accuracy: 76% (641/836)
```

The submission includes a function that takes a file path to an image as input and returns the dog breed that is predicted by the CNN.

Excellent work on implementing predict_breed_transfer() function which return predicted breed.

Step 5: Write Your Algorithm

The submission uses the CNN from the previous step to detect dog breed. The submission has different output for each detected image type (dog, human, other) and provides either predicted actual (or resembling) dog breed.

Step 6: Test Your Algorithm

The submission tests at least 6 images, including at least two human and two dog images.

Excellent work on testing the dog app on different types of images

Submission provides at least three possible points of improvement for the classification algorithm.

You have correctly provided the answer of Question-6
For more information read the blog of How To Improve Deep Learning Performance
https://machinelearningmastery.com/improve-deep-learning-performance/

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START