UDACITY

< Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW

HISTORY

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Nice submission. One change and the project will be complete.
Hope you found the material interesting and enjoyable.
Great job and good luck going forward.

Files Submitted

The submission includes all required, complete notebook files.

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.

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).

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

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.

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

Yes, good points.

The submission specifies a CNN architecture.

Great you experimented with dropout.

Answer describes the reasoning behind the selection of layer types.

When you next have the opportunity, if you like, have a look at batch normalization for improving training speed, as well as other aspects.
Also, ELU activation is gaining in popularity vs Relu though they are similar.

https://pytorch.org/docs/master/nn.html#batchnorm1d

https://pytorch.org/docs/master/nn.html#torch.nn.ELU

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

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

All good here with this %.

Step 4: Create a CNN Using Transfer Learning

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

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

Point about dataset size is key.

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

Accuracy on the test set is 60% or greater.

Inception does the job.

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.

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.

Output format looks correct.

Appears 2 of the dog images tested are not from the training set(the first and large dog+man) so this is marked as completed. Not sure as there is no list of images. However, many of the dog images tested are from the training data. Please, in the future do not test any algorithm/model with training data.
Other images are well selected for testing.

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

The 3 points of improvement are more about specific recommendations to improve the algorithm, not particular points of failure as listed.
For an idea of how to improve the accuracy of a neural network here is a resource:
https://machinelearningmastery.com/improve-deep-learning-performance/

Please, provide 3 specific recommendations. They may be about the neural network or other aspects of the overall algorithm.

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH