UDACITY

‹ Return to "Deep Learning" in the 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 required 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.

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

```
The percentage of the first 100 images in human_files have a detected human face is: 99%
The percentage of the first 100 images in dog_files have a detected human face is: 17%
```

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 (Recommended) and Renset50 for detecting dog in an images.

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

Excellent work on achieving the accuracy of 80% on dog images and no false detection on human images.

```
The percentage of the images in human_files_short have a detected dog is 0%
The percentage of the images in dog_files_short have a detected dog is 80%
```

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.

Nice work for writing three separate data loaders for the training, validation, and test datasets of dog images and augmenting training data.

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, and dropout regularization along with relu activation.

Couple of Suggestions

- Consider using ELU or LEAKY_RELU activation function
- Consider adding Batch Normalization after each convolution layer
- Consider adding weight initialization parameter in conv2d operation
- Consider adding at least 5 convolution layer to improve accuracy
- Set Dropout rate to 0.5

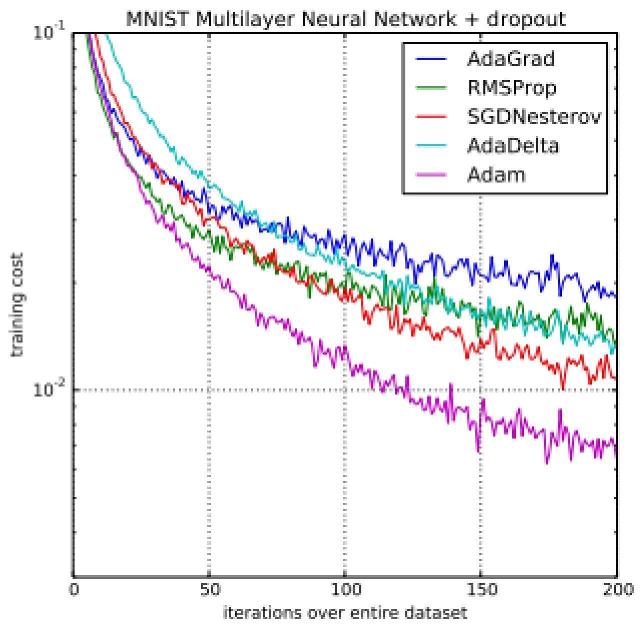
Answer describes the reasoning behind the selection of layer types.

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 SGD Optimizer.
One small suggestion, consider using Adam Optimizer which perform well among all other optimizer.
below figure depicts the same.



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

Excellent work on achieving the test accuracy of 12% on test dataset.

```
Test Loss: 3.887266


Test Accuracy: 12% (108/836)
```

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 Densnet161 network for transfer learning.
Try to explore some more architectures like Inception net and Xception net.
(link for both the net provided in summary)

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

You have correctly provided the answer.

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

Nice job on training the model for 30 epochs and saving the best weight.

Accuracy on the test set is 60% or greater.

Excellent on achieving the accuracy of 75% on test dataset.

```
Test Loss: 0.888454


Test Accuracy: 75% (629/836)
```

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 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

DOWNLOAD PROJECT

RETURN TO PATH

Rate this project