UDACITY

Return to "Deep Learning" in the classroom

DISCUSS ON STUDENT HUB

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Hi
Excellent work! You passed the project.

AI is an amazing and huge field.

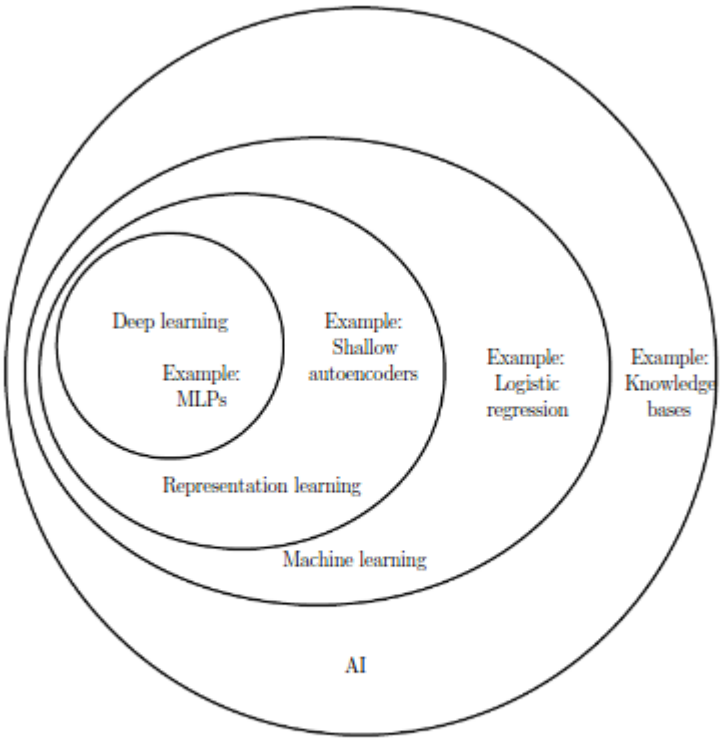

Figure 1.4: A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology.

Check these additional resources:

9 Deep Learning papers
MIT AGI: Building machines that see, learn, and think like people
YOLO Object Detection
⭐New Deep Reinforcement Learning Nanodegree⭐
The AlphaGo Movie

Keep up the good work
Sincerely
Leticia.

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.

```
The percentage of the detected face - Humans:98%

The percentage of the detected face - Dogs: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).

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

Percentage of the human images that have a detected dog: 0%

Percentage of the dog images that have a detected dog: 98%

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 created a single transformation using image augmentation.
IMPORTANT!! ONLY USE IMAGE AUGMENTATION OVER THE TRAINING SET. You can use a different transformations for testing and validation sets.

```
transform = transforms.Compose([transforms.Resize(size=224),
                                transforms.CenterCrop((224,224)),
                                transforms.RandomHorizontalFlip(), # randomly flip a
nd rotate
                                transforms.RandomRotation(10),
                                transforms.ToTensor(),
                                transforms.Normalize(mean=[0.485, 0.456, 0.406], std
=[0.229, 0.224, 0.225])])
```

For testing and validations:

```
transform = transforms.Compose([transforms.Resize(size=224),
                                transforms.CenterCrop((224,224)),
                                transforms.ToTensor(),
                                transforms.Normalize(mean=[0.485, 0.456, 0.406], std
=[0.229, 0.224, 0.225])])
```

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

You answered the question 3.

The submission specifies a CNN architecture.

You wrote a CNN for dog breed recognition.

Using regularization methods, like Dropout, is always helpful when we are training a model that trends to overfit.

Dropout prevents overfitting and provides a way of approximately combining exponentially many different neural network architectures efficiently. The term “dropout” refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections.

The choice of which units to drop is random. In the simplest case, each unit is retained with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks. For the input units, however, the optimal probability of retention is usually closer to 1 than to 0.5.

Related paper: Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Answer describes the reasoning behind the selection of layer types.

You answered the question 4.

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

You used CrossEntropyLoss loss and SGD optimizer.
Pytorch provides great schedulers to improve SGD optimizer by selecting the Learning Rate using different strategies. How to adjust Learning Rate

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

Testing Loss Average: 3.696285

Test Accuracy: 16% (135/836)

Step 4: Create a CNN Using Transfer Learning

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

You used vgg16 pre-trained model.
Consider trying some aof the resnet or densenet models.

Network	Top-1 error	Top-5 error
AlexNet	43.45	20.91
VGG-11	30.98	11.37
VGG-13	30.07	10.75
VGG-16	28.41	9.62
VGG-19	27.62	9.12
VGG-11 with batch normalization	29.62	10.19
VGG-13 with batch normalization	28.45	9.63
VGG-16 with batch normalization	26.63	8.5
VGG-19 with batch normalization	25.76	8.15
ResNet-18	30.24	10.92
ResNet-34	26.7	8.58
ResNet-50	23.85	7.13
ResNet-101	22.63	6.44
ResNet-152	21.69	5.94

Network	Top-1 error	Top-5 error
SqueezeNet 1.0	41.9	19.58
SqueezeNet 1.1	41.81	19.38
Densenet-121	25.35	7.83
Densenet-169	24	7
Densenet-201	22.8	6.43
Densenet-161	22.35	6.2
Inception v3	22.55	6.44
GoogleNet	30.22	10.47
ShuffleNet V2	30.64	11.68
MobileNet V2	28.12	9.71
ResNeXt-50-32x4d	22.38	6.3
ResNeXt-101-32x8d	20.69	5.47

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

You answered the question 5.

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.

Testing Loss Average: 0.637072

Test Accuracy: 81% (678/836)

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.

You wrote predict_breed_transfer for meeting specifications.

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.

You wrote the required algorithm

Step 6: Test Your Algorithm

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

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

You answered the question 6.
More improvement ideas: How To Improve Deep Learning Performance

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review