UDACITY

Back to Deep Learning

Dog Breed Classifier

REVIEW

HISTORY

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Files Submitted

The submission includes all required, complete notebook files.

The notebook files are required~ 😉

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.

Nice work!

You can use this to simplify your code:

```
np.mean([face_detector(human) for human in human_files_short])
```

or this:

```
np.mean(list(map(face_detector, human_files_short)))
```

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).

Nice implementation~ 👍

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

Nice work!

You can use this to simplify your code:

```
np.mean([dog_detector(human) for human in human_files_short])
```

or this:

```
np.mean(list(map(dog_detector, human_files_short)))
```

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.

Well done!

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

Good job!

The submission specifies a CNN architecture.

Great model architecture!

Answer describes the reasoning behind the selection of layer types.

1. convolutional layers with more and more filters, so we can extract more and more advanced features
2. Max pooling and Global Average Pooling are used to preserve only the strongest of these features, to ensure the position and rotation invariance of features, reduce the number of model parameters, and reduce over-fitting problems.
3. Dropout layer is used to reduce the complexity of the model, enhance the generalization ability of the model, prevent over-fitting, and reduce computation complexity.

This is a great resource for discussing the principles of different basic CNN structures:
http://cs231n.github.io/convolutional-networks/#architectures

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

well done!

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

Test Accuracy: 23% (219/928)

There are many tricks you can use in ConvNets. A good advice is to increase the network size until it overfits (on the validation set) and then add regularizers and data augmentation and if it doesn't overfit any more, again try to increase the model size. Generally, the model size depends on the amount of data and its complexity, but if you use max-pooling, you want to increase the number of neurons towards the top (e.g. double for each layer). Usually it will be 2-5 layers before the dense layer and for the kernel sizes its trial and error (somewhere between 3 and 5). To get a feeling for what works best, you can do a grid search over the parameters.

Some regularizers:

1. Batch normalization can help if you have more layers (>2). It prevents diminishing gradients (all layers receive the gradient to a similar extend). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
2. Max-Norm regularization & Dropout: Dropout: A Simple Way to Prevent Neural Networks from Overfitting
3. L1 / L2 weight regularization
4. Sparsity regularization (e.g. [Sparse deep belief net model for visual area V2] (http://web.eecs.umich.edu/~honglak/nips07-sparseDBN.pdf))
5. Gradient clipping (more thorough search in the cost landscape)
6. Data augmentation increases your dataset and thus prevents overfitting (and you can inject prior knowledge) and max-out units were successful in recent image-classification contests: Galaxy Zoo challenge on Kaggle 和 Classifying plankton with deep neural networks

(from: some advices about how to improve the performance of Convolutional Neural Networks)

For more information:

1. What is maxout in neural network?
2. What is the difference between max pooling and max out?
3. Maxout Networks

Step 4: Create a CNN Using Transfer Learning

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

Great choice~ You can also try the other three model architectures~ All you have to do is to download the bottleneck features, and then retrain, you will see different performances of different architectures ~

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

Great answer!

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

well done!

Accuracy on the test set is 60% or greater.

Test Accuracy: 75% (703/928)

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 implementation of the algorithm! 👍

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.

Excellent implementation of the algorithm! 👍
I recommend you to place the dog_detector before the face_detector because the accuracy of the former is higher.

Step 6: Test Your Algorithm

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

As requested, you should test pictures of you own (download online or your own pictures).

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

provided good suggestions for model improvement ~

Here are some suggestions for model improvement, I hope it can help you:

The suggestions have been divided into 4 sub-topics:

1. Improve Performance With Data.
2. Improve Performance With Algorithms.
3. Improve Performance With Algorithm Tuning.
4. Improve Performance With Ensembles.

For more information, please refer: How To Improve Deep Learning Performance

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

Rate this review