Dog Breed Classifier Project by Neil Simon

Image Classification using AWS SageMaker

This project takes a pretrained CNN model (Resnet18) and finetunes it for use in classifying dog breeds into 133 classifications based on a Dog Breeds dataset.

The steps that the notebook goes through are:

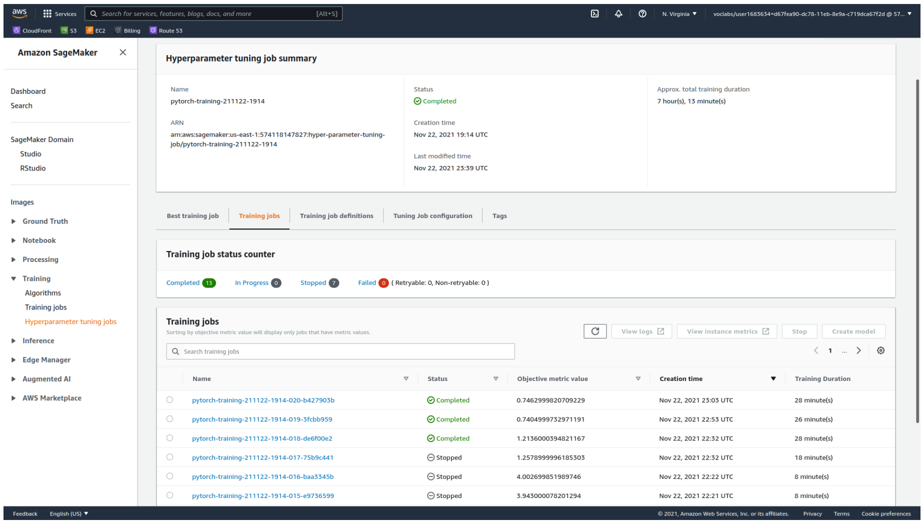
1. Retrieving a dataset
2. Uncompressing that dataset.
3. Uploading that dataset to an S3 bucket.
4. Setting up hyperparameter tuning using learning rate, weight decay, eps and batch size using the AdamW optimizer on ResNet18.
5. Starting a hyperparameter tuning job using 20 training jobs (2 at a time) with autoshutdown on the AWS SageMaker Hyperparameter Tuning system.
6. Record the best hyperparameters as discovered from the above.
7. Train a fine-tuned model using ResNet18 and the best hyperparameters over a larger number of epochs, recording profiling and debug data.
8. Examine the output from the profiling and debug of the above.
9. Re-run the training with profiling and debug turned off (due to issues deploying model trained with the above).
10. Deploy that model as an endpoint on AWS.
11. Test that endpoint with new images.

Hyperparameter Tuning

I chose to use the ResNet18 model with a single layer Linear NN. ResNet18 is a well proven pretrained CNN and the results of a single layer NN was enough to get good results from training. The hyperparameters I chose to tune were: 1. Learning rate 1. eps 1. Weight decay 1. Batch size These hyperparameters gave me broad coverage of the tunables for the system. I chose to use values between 0.1x and 10x the default as the default values tend to give good results for the first 3, and for batch size I stuck with the options of 32 and 64.

Results

I found debugging the system was painful. The long turnaround between making a change/fix, and it being tested was particularly bad. Additionally, the documentation for much of these systems is rather limited. For instance working out how to create the DataLoaders was far more difficult than it should have been based on what was ultimately required.



Hyperparameter tuning job summary

Name
pytorch-training-211122-1914

ARN
arn:aws:sagemaker:us-east-1:574118147827:hyper-parameter-tuning-job/pytorch-training-211122-1914

Status
Completed

Creation time
Nov 22, 2021 19:14 UTC

Last modified time
Nov 22, 2021 23:39 UTC

Approx. total training duration
7 hour(s), 13 minute(s)

Training job status counter

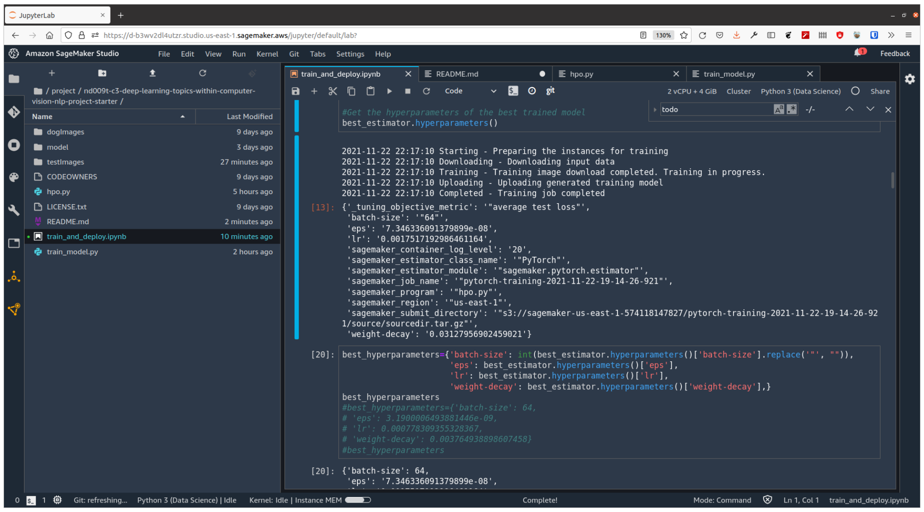
Completed 13 In Progress 0 Stopped 7 Failed 0 (Retryable: 0, Non-retryable: 0)

Training jobs

Sorting by objective metric value will display only jobs that have metric values.

	Name	Status	Objective metric value	Creation time	Training Duration
○	pytorch-training-211122-1914-020-b427903b	Completed	0.7462999820709229	Nov 22, 2021 23:03 UTC	28 minute(s)
○	pytorch-training-211122-1914-019-3fcbb959	Completed	0.7404999732971191	Nov 22, 2021 22:53 UTC	26 minute(s)
○	pytorch-training-211122-1914-018-de6f00e2	Completed	1.2136000394821167	Nov 22, 2021 22:32 UTC	28 minute(s)
○	pytorch-training-211122-1914-017-75b9c441	Stopped	1.2578999996185303	Nov 22, 2021 22:32 UTC	18 minute(s)
○	pytorch-training-211122-1914-016-baa3345b	Stopped	4.002699851989746	Nov 22, 2021 22:22 UTC	8 minute(s)
○	pytorch-training-211122-1914-015-e9736599	Stopped	3.943000078201294	Nov 22, 2021 22:21 UTC	8 minute(s)

Figure 1: Hyperparameter Training Job

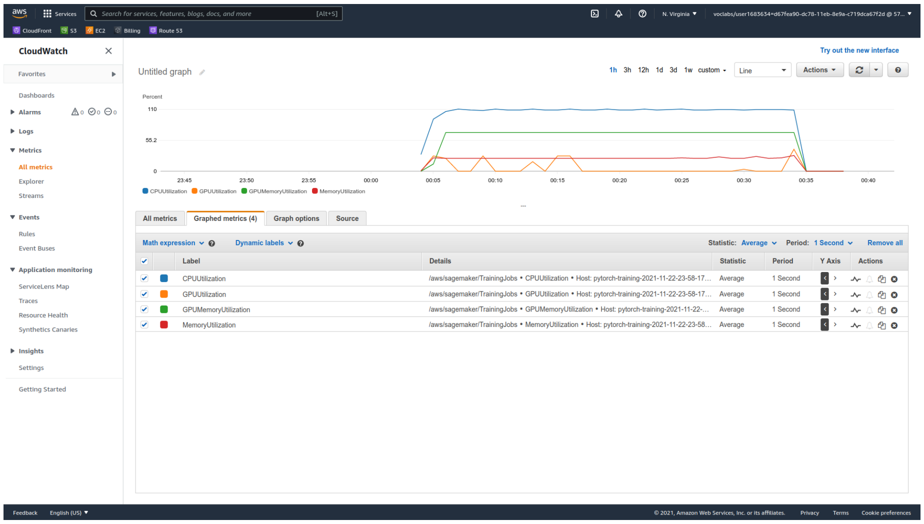


/ project / nd009t-c3-deep-learning-topics-within-computer-vision-nlp-project-starter /

Name	Last Modified
dogImages	9 days ago
model	3 days ago
testImages	27 minutes ago
CODEOWNERS	9 days ago
hpo.py	5 hours ago
LICENSE.txt	9 days ago
README.md	2 minutes ago
train_and_deploy.ipynb	10 minutes ago
train_model.py	2 hours ago

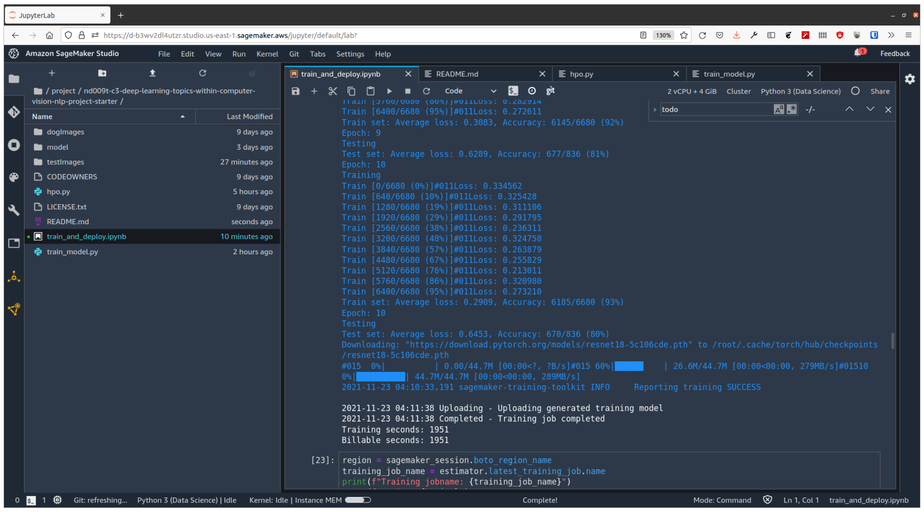
```
#Get the hyperparameters of the best trained model
best_estimator.hyperparameters()

2021-11-22 22:17:10 Starting - Preparing the instances for training
2021-11-22 22:17:10 Downloading - Downloading input data
2021-11-22 22:17:10 Training - Training image download completed. Training in progress.
2021-11-22 22:17:10 Uploading - Uploading generated training model
2021-11-22 22:17:10 Completed - Training job completed
[13]: {'_tuning_objective_metric': '"average test loss"',
 'batch-size': '"64"',
 'eps': '7.346336091379899e-08',
 'lr': '0.0017517192986461164',
 'sagemaker_container_log_level': '20',
 'sagemaker_estimator_class_name': '"PyTorch"',
 'sagemaker_estimator_module': '"sagemaker.pytorch.estimator"',
 'sagemaker_job_name': '"pytorch-training-2021-11-22-19-14-26-921"',
 'sagemaker_program': '"hpo.py"',
 'sagemaker_region': '"us-east-1"',
 'sagemaker_submit_directory': '"s3://sagemaker-us-east-1-574118147827/pytorch-training-2021-11-22-19-14-26-92
1/source/sourcedir.tar.gz"',
 'weight-decay': '0.03127956902459021'}

[20]: best_hyperparameters={'batch-size': int(best_estimator.hyperparameters()['batch-size'].replace('"', "")),
                       'eps': best_estimator.hyperparameters()['eps'],
                       'lr': best_estimator.hyperparameters()['lr'],
                       'weight-decay': best_estimator.hyperparameters()['weight-decay'],}
best_hyperparameters
#best_hyperparameters={'batch-size': 64,
# 'eps': 3.1900006493881446e-09,
# 'lr': 0.000778309355328367,
# 'weight-decay': 0.003764938898607458}
#best_hyperparameters

[20]: {'batch-size': 64,
 'eps': '7.346336091379899e-08',
```

Figure 2: Hyperparameters best results



☑		Label	Details	Statistic	Period	Y Axis	Actions
☑	🟦	CPUUtilization	/aws/sagemaker/TrainingJobs • CPUUtilization • Host: pytorch-training-2021-11-22-23-58-17...	Average	1 Second	< >	
☑	🟧	GPUUtilization	/aws/sagemaker/TrainingJobs • GPUUtilization • Host: pytorch-training-2021-11-22-23-58-17...	Average	1 Second	< >	
☑	🟩	GPUMemoryUtilization	/aws/sagemaker/TrainingJobs • GPUMemoryUtilization • Host: pytorch-training-2021-11-22-...	Average	1 Second	< >	
☑	🟥	MemoryUtilization	/aws/sagemaker/TrainingJobs • MemoryUtilization • Host: pytorch-training-2021-11-22-23-58...	Average	1 Second	< >	

Figure 3: Training job resource use



/ project / nd009t-c3-deep-learning-topics-within-computer-vision-nlp-project-starter /

Name	Last Modified
dogImages	9 days ago
model	3 days ago
testImages	27 minutes ago
CODEOWNERS	9 days ago
hpo.py	5 hours ago
LICENSE.txt	9 days ago
README.md	seconds ago
train_and_deploy.ipynb	10 minutes ago
train_model.py	2 hours ago

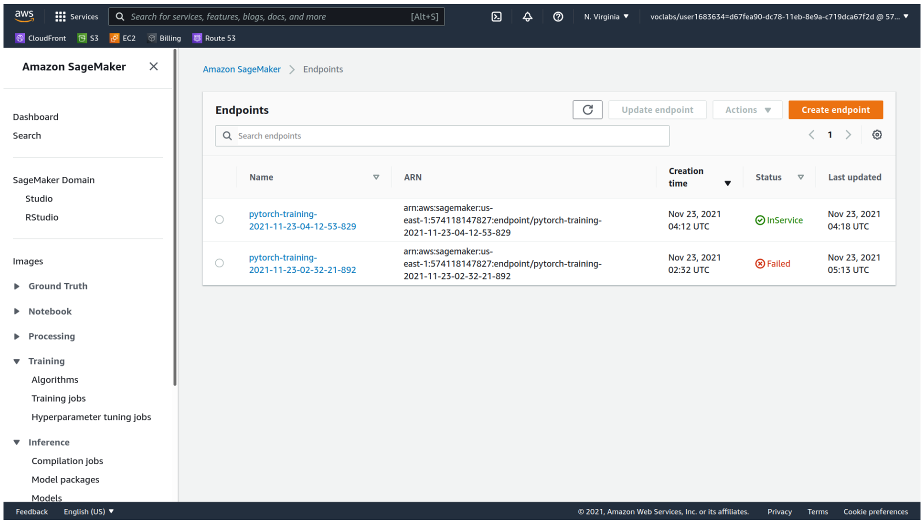
```
Train [5760/6680 (86%)]#011Loss: 0.282914
Train [6400/6680 (95%)]#011Loss: 0.272611
Train set: Average loss: 0.3083, Accuracy: 6145/6680 (92%)
Epoch: 9
Testing
Test set: Average loss: 0.6289, Accuracy: 677/836 (81%)
Epoch: 10
Training
Train [0/6680 (0%)]#011Loss: 0.334562
Train [640/6680 (10%)]#011Loss: 0.325428
Train [1280/6680 (19%)]#011Loss: 0.311106
Train [1920/6680 (29%)]#011Loss: 0.291795
Train [2560/6680 (38%)]#011Loss: 0.236311
Train [3200/6680 (48%)]#011Loss: 0.324750
Train [3840/6680 (57%)]#011Loss: 0.263879
Train [4480/6680 (67%)]#011Loss: 0.255829
Train [5120/6680 (76%)]#011Loss: 0.213011
Train [5760/6680 (86%)]#011Loss: 0.320980
Train [6400/6680 (95%)]#011Loss: 0.273210
Train set: Average loss: 0.2909, Accuracy: 6185/6680 (93%)
Epoch: 10
Testing
Test set: Average loss: 0.6453, Accuracy: 670/836 (80%)
Downloading: "https://download.pytorch.org/models/resnet18-5c106cde.pth" to /root/.cache/torch/hub/checkpoints
/resnet18-5c106cde.pth
#015  0%|          | 0.00/44.7M [00:00<?, ?B/s]#015 60%|██████    | 26.6M/44.7M [00:00<00:00, 279MB/s]#01510
0%|██████████| 44.7M/44.7M [00:00<00:00, 289MB/s]
2021-11-23 04:10:33,191 sagemaker-training-toolkit INFO     Reporting training SUCCESS

2021-11-23 04:11:38 Uploading - Uploading generated training model
2021-11-23 04:11:38 Completed - Training job completed
Training seconds: 1951
Billable seconds: 1951
```

```
[23]: region = sagemaker_session.boto_region_name
      training_job_name = estimator.latest_training_job.name
      print(f"Training jobname: {training_job_name}")
```

Figure 4: Training job output

Model Deployment

The deployed model runs a tuned version of ResNet18 and accepts data as preprocessed in the notebook.

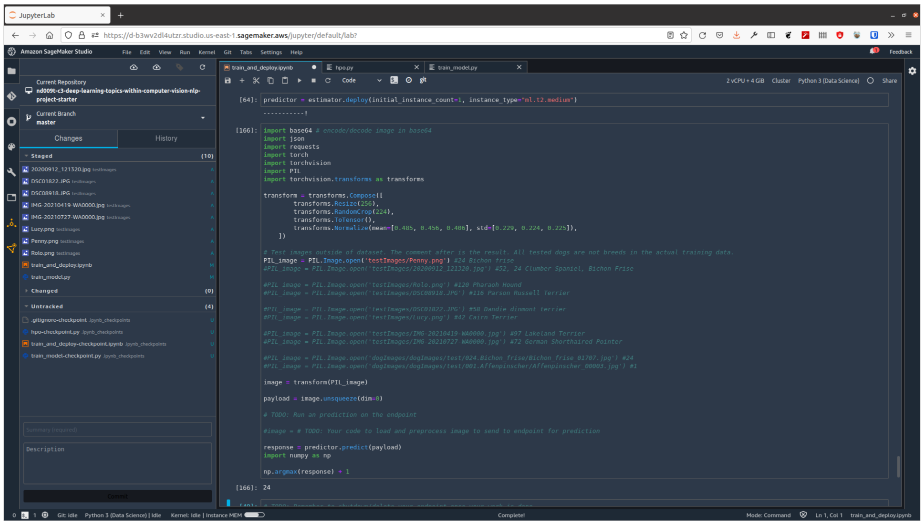


Endpoints

	Name	ARN	Creation time	Status	Last updated
○	pytorch-training-2021-11-23-04-12-53-829	arn:aws:sagemaker:us-east-1:574118147827:endpoint/pytorch-training-2021-11-23-04-12-53-829	Nov 23, 2021 04:12 UTC	✓ InService	Nov 23, 2021 04:18 UTC
○	pytorch-training-2021-11-23-02-32-21-892	arn:aws:sagemaker:us-east-1:574118147827:endpoint/pytorch-training-2021-11-23-02-32-21-892	Nov 23, 2021 02:32 UTC	☒ Failed	Nov 23, 2021 05:13 UTC

Figure 5: Deployed endpoint

Figure 6: Penny for testing



JupyterLab
https://d-b3wv2dl4utzr.studio.us-east-1.sagemaker.aws/jupyter/default/lab?
Amazon SageMaker Studio
File Edit View Run Kernel Git Tabs Settings Help
Current Repository
nd009t-c3-deep-learning-topics-within-computer-vision-nlp-project-starter
Current Branch
master
Changes
History
Staged (10)
20200912_121320.jpg testImages A
DSC01822.JPG testImages A
DSC08918.JPG testImages A
IMG-20210419-WA0000.jpg testImages A
IMG-20210727-WA0000.jpg testImages A
Lucy.png testImages A
Penny.png testImages A
Rolo.png testImages A
train_and_deploy.ipynb M
train_model.py M
Changed (0)
Untracked (4)
.gitignore-checkpoint .ipynb_checkpoints U
hpo-checkpoint.py .ipynb_checkpoints U
train_and_deploy-checkpoint.ipynb .ipynb_checkpoints U
train_model-checkpoint.py .ipynb_checkpoints U
Summary (required)
Description
Commit
train_and_deploy.ipynb
hpo.py
train_model.py
Code
2 vCPU + 4 GiB
Cluster
Python 3 (Data Science)
Share
Feedback
Git: idle
Python 3 (Data Science) | Idle
Kernel: Idle | Instance MEM
Complete!
Mode: Command
Ln 1, Col 1
train_and_deploy.ipynb

```
[64]: predictor = estimator.deploy(initial_instance_count=1, instance_type="ml.t2.medium")
      -----------!

[166]: import base64 # encode/decode image in base64
       import json
       import requests
       import torch
       import torchvision
       import PIL
       import torchvision.transforms as transforms

       transform = transforms.Compose([
               transforms.Resize(256),
               transforms.RandomCrop(224),
               transforms.ToTensor(),
               transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
           ])

       # Test images outside of dataset. The comment after is the result. All tested dogs are not breeds in the actual training data.
       PIL_image = PIL.Image.open('testImages/Penny.png') #24 Bichon frise
       #PIL_image = PIL.Image.open('testImages/20200912_121320.jpg') #52, 24 Clumber Spaniel, Bichon Frise

       #PIL_image = PIL.Image.open('testImages/Rolo.png') #120 Pharaoh Hound
       #PIL_image = PIL.Image.open('testImages/DSC08918.JPG') #116 Parson Russell Terrier

       #PIL_image = PIL.Image.open('testImages/DSC01822.JPG') #58 Dandie dinmont terrier
       #PIL_image = PIL.Image.open('testImages/Lucy.png') #42 Cairn Terrier

       #PIL_image = PIL.Image.open('testImages/IMG-20210419-WA0000.jpg') #97 Lakeland Terrier
       #PIL_image = PIL.Image.open('testImages/IMG-20210727-WA0000.jpg') #72 German Shorthaired Pointer

       #PIL_image = PIL.Image.open('dogImages/dogImages/test/024.Bichon_frise/Bichon_frise_01707.jpg') #24
       #PIL_image = PIL.Image.open('dogImages/dogImages/test/001.Affenpinscher/Affenpinscher_00003.jpg') #1

       image = transform(PIL_image)

       payload = image.unsqueeze(dim=0)

       # TODO: Run an prediction on the endpoint

       #image = # TODO: Your code to load and preprocess image to send to endpoint for prediction

       response = predictor.predict(payload)
       import numpy as np

       np.argmax(response) + 1

[166]: 24
```

Figure 7: Using the endpoint