

EAI 6010: Applications of Artificial Intelligence

Module 5 Assignment

Jeseeka Shah

Prof. Abhijit

Sanyal

30th June 2022

Introduction :-

Creating technology from the bottom up is getting increasingly rare these days. Most new products need a service at some point. Almost every new initiative, regardless of industry, uses existing solutions to some level. When developing a machine learning model and looking for existing expertise and pre-designed (pre trained) code to use in your solution, the process isn't all that different. All you must do now is modify the code to suit your own company requirements. Machine learning algorithms have several challenges, the most serious of which is a shortage of high-quality data. A project may be halted before it even begins due to a lack of training data. Data gathering, labeling, cleaning, and preprocessing can all add up to a significant financial outlay for a project. As a result, many machine learning initiatives fail due to a lack of model training and validation resources. Using off the-shelf technologies reduce preprocessing costs. They are intended to do the most common operations with little setup. The best thing is that they are available as ready-to-use solutions for all phases of machine learning workloads.

Cloud computing with Google: From healthcare to finance to robotics, there is no industry where artificial intelligence (AI) and machine learning (ML) are not transforming the world and simplifying everyone's lives. We cannot, however, pretend that cloud computing and artificial intelligence are unrelated. The administration of data lakes, big data analysis, streaming data, and enormous data storage are all crucial parts of any machine learning model, and more than 90% of enterprises are using cloud services for these purposes. I used the input data from the news category classifier. Data scientists frequently have to explain their findings to others.

```
news-classifier --zsh -- 165x46
Allow unauthenticated invocations to [news-classifier] (y/N)? y

Building using Dockerfile and deploying container to Cloud Run service [news-classifier] in project [my-project-1108-341900] region [us-central1]
✓ Building and deploying new service... Done.
  ✓ Uploading sources...
  ✓ Building Container... Logs are available at [https://console.cloud.google.com/cloud-build/builds/4f4ea2a8-fa36-41d4-ad70-c8a759933614?project=221286998744].
  ✓ Creating Revision...
  ✓ Routing traffic...
  ✓ Setting IAM Policy...
Done.
Service [news-classifier] revision [news-classifier-00001-nof] has been deployed and is serving 100 percent of traffic.
Service URL: https://news-classifier-q7uu4hjoea-uc.a.run.app
(base) ashutoshsingh@Ashutoshs-MacBook-Pro news-classifier % curl -d '{"headline": "A Swirling Vortex Is No Match for This Deep-Sea Sponge","shortDescription": "A newspaper finds a glass sponge has the power to eliminate destructive vortices that are created when fluid moves around a blunt object."}' -H "Content-Type: application/json" -X POST https://news-classifier-q7uu4hjoea-uc.a.run.app/classify
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.</p>
(base) ashutoshsingh@Ashutoshs-MacBook-Pro news-classifier % curl -d '{"headline": "A Swirling Vortex Is No Match for This Deep-Sea Sponge","shortDescription": "A newspaper finds a glass sponge has the power to eliminate destructive vortices that are created when fluid moves around a blunt object."}' -H "Content-Type: application/json" -X POST https://news-classifier-q7uu4hjoea-uc.a.run.app/classify
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.</p>
(base) ashutoshsingh@Ashutoshs-MacBook-Pro news-classifier % gcloud run deploy news-classifier --project my-project-1108-341900 --source . --region us-central1 --allow-unauthenticated --memory=1024Mi
This command is equivalent to running `gcloud builds submit --tag [IMAGE] .` and `gcloud run deploy news-classifier --image [IMAGE]`

Building using Dockerfile and deploying container to Cloud Run service [news-classifier] in project [my-project-1108-341900] region [us-central1]
✓ Building and deploying new service... Done.
  ✓ Uploading sources...
  ✓ Building Container... Logs are available at [https://console.cloud.google.com/cloud-build/builds/d41d5ce3-6396-43d8-8fc7-9dea9126d7fc?project=221286998744].
  ✓ Creating Revision...
  ✓ Routing traffic...
  ✓ Setting IAM Policy...
Done.
Service [news-classifier] revision [news-classifier-00001-xin] has been deployed and is serving 100 percent of traffic.
Service URL: https://news-classifier-q7uu4hjoea-uc.a.run.app
(base) ashutoshsingh@Ashutoshs-MacBook-Pro news-classifier % curl -d '{"headline": "A Swirling Vortex Is No Match for This Deep-Sea Sponge","shortDescription": "A newspaper finds a glass sponge has the power to eliminate destructive vortices that are created when fluid moves around a blunt object."}' -H "Content-Type: application/json" -X POST https://news-classifier-q7uu4hjoea-uc.a.run.app/classify
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
```

I've made a microservice available on the cloud that contains the News Category Classifier model. Run a Google Cloud Platform (GCP) instance with a backend URL created so that it may be used as an API by other suppliers to accept data and provide results.

Two inputs in JSON format must be provided as arguments to this microservice. These input data's names are: -

Headline: It denotes the news story's headline, whose category must be anticipated.

Short Description - This element denotes a brief description of the news item whose category must be determined. Output - Predicted News Category is the outcome that is returned via API.

The screenshot shows the Google Cloud Build console. On the left, there are tabs for 'Triggers' and 'Settings'. The main area displays a 'Build Summary' for a build named '0: gcr.io/cloud-builders/docker build --network cloudbuild -n...'. The build status is 'Success'. The 'BUILD LOG' tab is selected, showing the following log entries:

```
298 Step 7/7 : CMD exec gunicorn --bind :$PORT --workers 1 --threads 8 --timeout 0 main:app
299 ----> Running in c22ffc2b35e
300 Removing intermediate container c22ffc2b35e
301 ----> 6cbfe79321a5
302 Successfully built 6cbfe79321a5
303 Successfully tagged us-central1-docker.pkg.dev/my-project-1108-341908/cloud-run-source-deploy/news-classifier:latest
304 PUSH
305 Pushing us-central1-docker.pkg.dev/my-project-1108-341908/cloud-run-source-deploy/news-classifier:latest
306 The push refers to repository [us-central1-docker.pkg.dev/my-project-1108-341908/cloud-run-source-deploy/news-classifier]
307 32281cc41ed8: Preparing
308 d8f85dcf0d85: Preparing
309 0b7c3ae4e15d: Preparing
310 1da136635b59: Preparing
311 df1e77b59947: Preparing
312 53dfcb4b8d77: Preparing
313 32034715e5d4: Preparing
314 7d0ebbe3f5d2: Preparing
315 53dfcb4b8d77: Waiting
316 32034715e5d4: Waiting
317 7d0ebbe3f5d2: Waiting
318 1da136635b59: Layer already exists
319 df1e77b59947: Layer already exists
320 53dfcb4b8d77: Layer already exists
321 32034715e5d4: Layer already exists
322 7d0ebbe3f5d2: Layer already exists
323 d8f85dcf0d85: Pushed
324 0b7c3ae4e15d: Pushed
325 32281cc41ed8: Pushed
326 latest: digest: sha256:ef89a24857ebd660ee2a4516f559108053ecf486351630b1b9f9ea7e0f096014 size: 1999
327 DONE
```

The screenshot shows the Google Cloud Storage console. The 'OBJECTS' tab is selected. The bucket 'models-jess-1' is selected. The object 'news_category_classifier' is listed. The object details are as follows:

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Ref
news_category_classifier	158 MB	application/octet-stream	29 Jun 20...	Standard	29 Jun 20...	Not public	-	Google-managed key	-

The screenshot shows a REST client interface. The request is a POST to 'https://classify-article-i742jjsuq-uc.a.run.app/classify'. The request body is a JSON object with the following structure:

```
{
  "headline": "A Swirling Vortex Is No Match for This Deep-Sea Sponge",
  "shortDescription": "A new paper finds a glass sponge has the power to eliminate destructive vortices that are created when fluid moves around a blunt object."
}
```

The response is a JSON object with the following structure:

```
{
  "predictedClass": "HEALTHY LIVING"
}
```

Conclusion

To enable Terraform to use the Docker image, we must construct it and add it to the project's container registry. The procedures below may be executed from the source code's directory after using Cloud Shell to clone it. You have the ability to delete all of the resources you've used in Terraform. Run the following command to accomplish this, being careful to approve it with "yes." So far, we have been able to use Terraform to provide our deployment to Cloud Run as code. You might be able to launch programs considerably more rapidly as a result.

The URL of the service: - <https://news-classifier-q7uu4hjoea-uc.a.run.app>

References

1. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012, September 30). ImageNet Classification With Deep Convolutional Neural Networks. NeurIPS Proceedings. <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
2. Alake, R. (2021, November 3). Implementing AlexNet CNN Architecture Using TensorFlow 2.0+ And Keras. Towards Data Science. <https://towardsdatascience.com/implementing-alexnet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98>.
3. Timothy. (2020, January 23). Deploying Docker Images To Cloud Run Using Terraform. Medium. <https://medium.com/google-cloud/deploying-docker-images-to-cloud-run-using-terraform-ee8ae4ecb72>