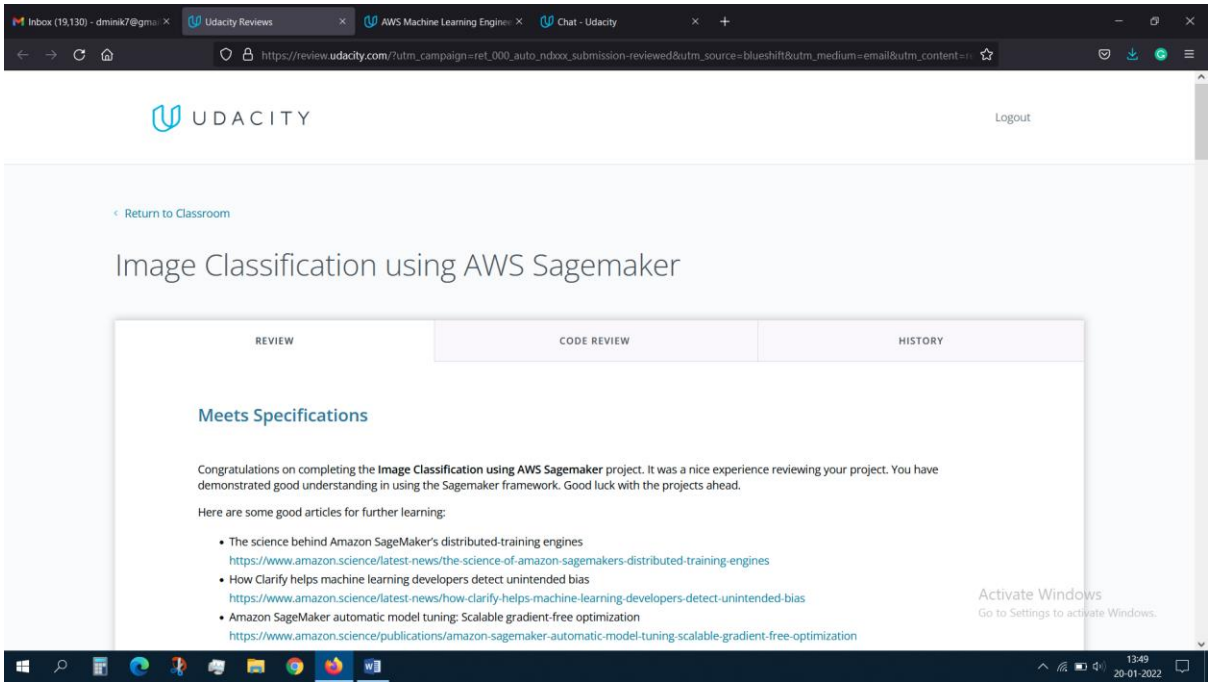
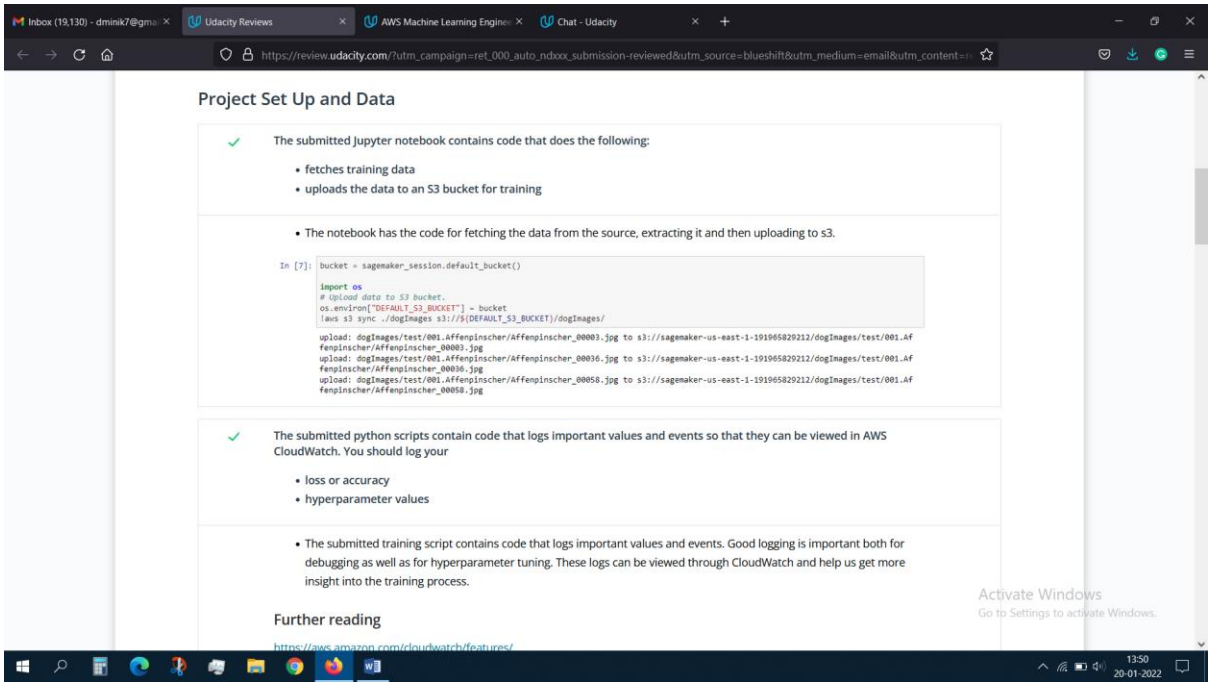
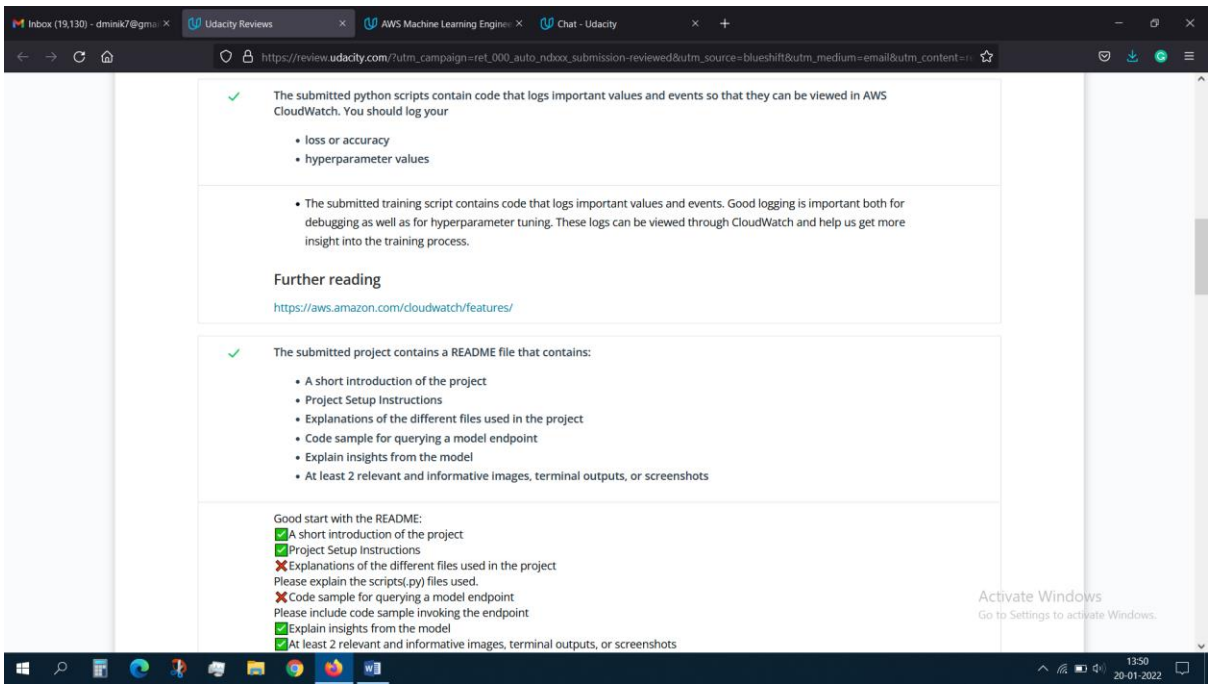
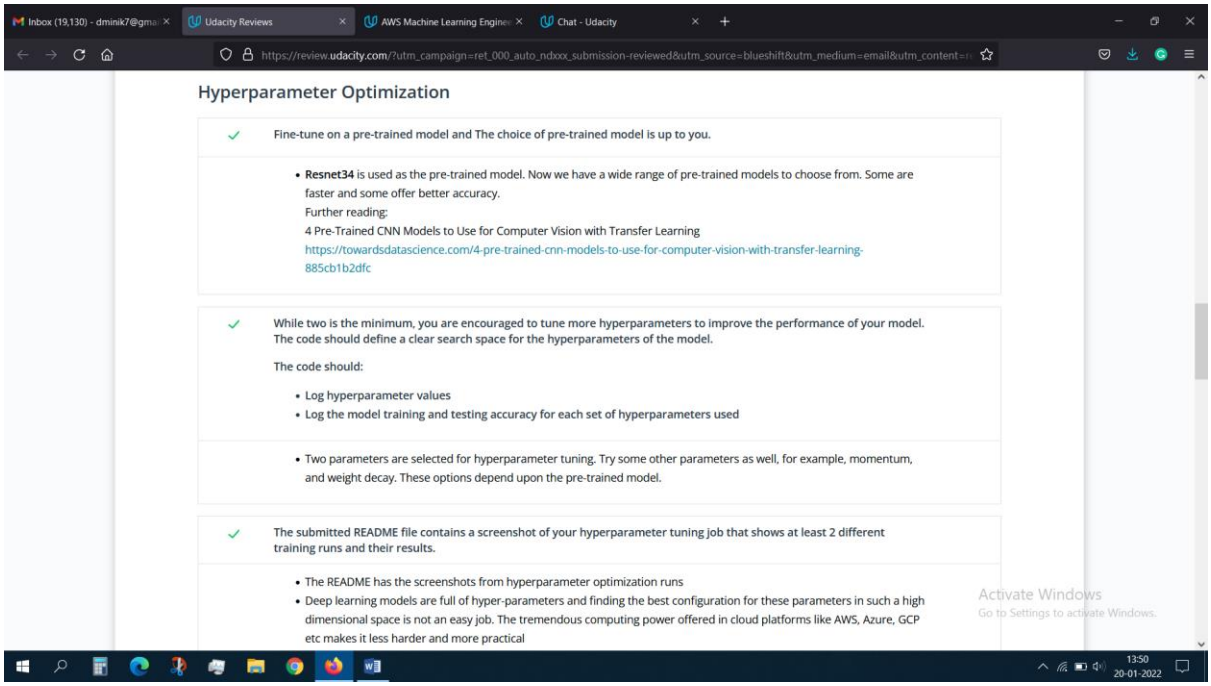


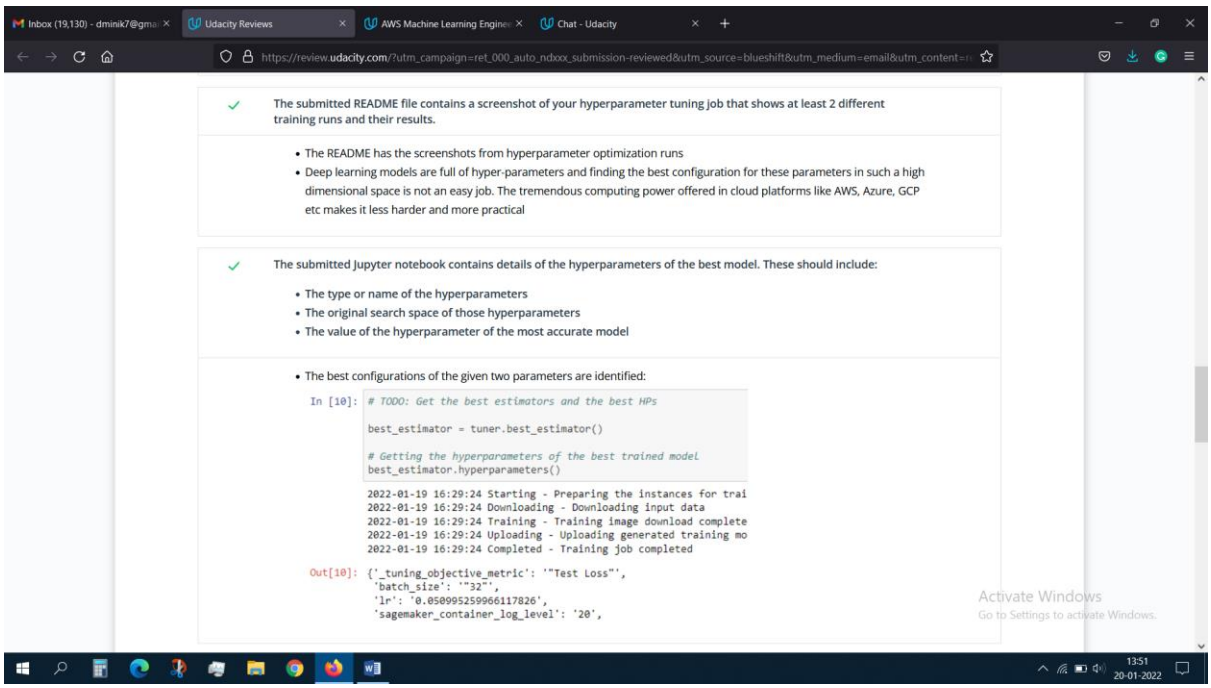
Here are some good articles for further learning:

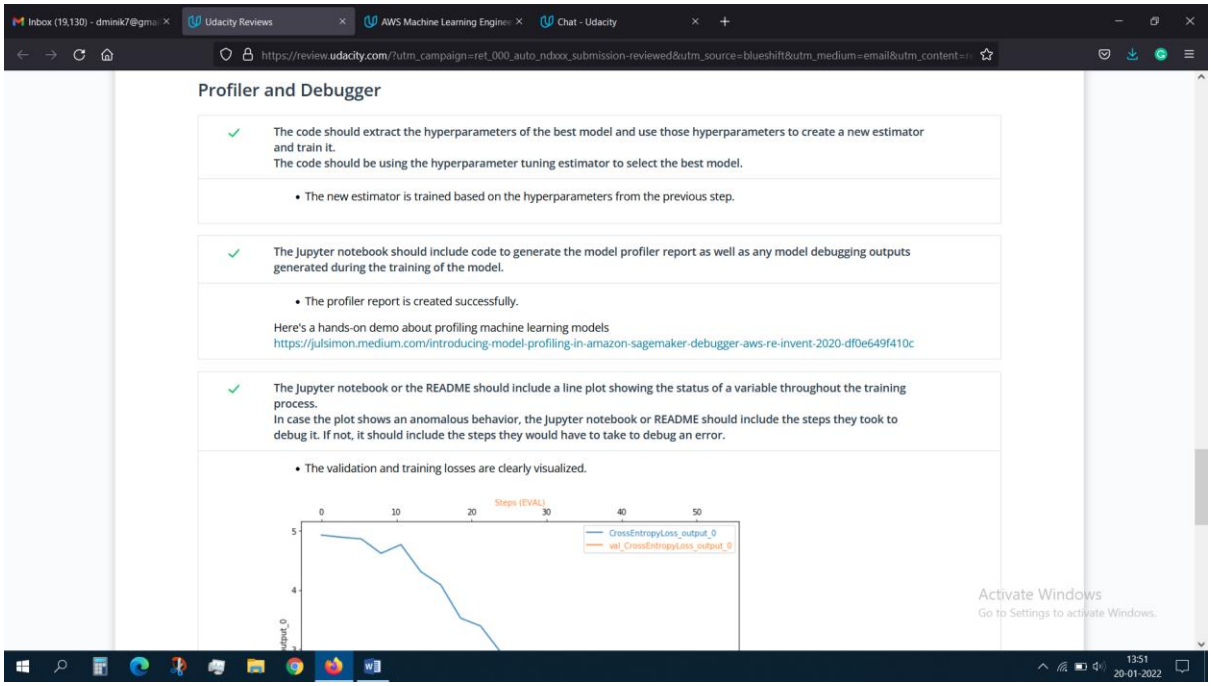
- The science behind Amazon SageMaker's distributed-training engines https://www.amazon.science/latest-news/the-science-of-amazon-sagemakers-distributed-training-engines
- How Clarify helps machine learning developers detect unintended bias https://www.amazon.science/latest-news/how-clarify-helps-machine-learning-developers-detect-unintended-bias
- Amazon SageMaker automatic model tuning: Scalable gradient-free optimization https://www.amazon.science/publications/amazon-sagemaker-automatic-model-tuning-scalable-gradient-free-optimization


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
Project Set Up and Data
The submitted Jupyter notebook contains code that does the following:
fetches training data
uploads the data to an S3 bucket for training
The notebook has the code for fetching the data from the source, extracting it and then uploading to s3.
In [7]:
bucket = sagemaker_session.default_bucket()
import os
# Upload data to S3 bucket.
os.environ["DEFAULT_S3_BUCKET"] = bucket
!aws s3 sync ./dogImages s3://${DEFAULT_S3_BUCKET}/dogImages/
upload: dogImages/test/001.Affenpinscher/Affenpinscher_00003.jpg to s3://sagemaker-us-east-1-191965829212/dogImages/test/001.Affenpinscher/Affenpinscher_00003.jpg
upload: dogImages/test/001.Affenpinscher/Affenpinscher_00036.jpg to s3://sagemaker-us-east-1-191965829212/dogImages/test/001.Affenpinscher/Affenpinscher_00036.jpg
upload: dogImages/test/001.Affenpinscher/Affenpinscher_00058.jpg to s3://sagemaker-us-east-1-191965829212/dogImages/test/001.Affenpinscher/Affenpinscher_00058.jpg
The submitted python scripts contain code that logs important values and events so that they can be viewed in AWS CloudWatch. You should log your
loss or accuracy
hyperparameter values
The submitted training script contains code that logs important values and events. Good logging is important both for debugging as well as for hyperparameter tuning. These logs can be viewed through CloudWatch and help us get more insight into the training process.
Further reading
https://aws.amazon.com/cloudwatch/features/
Activate Windows
Go to Settings to activate Windows.
13:50
20-01-2022

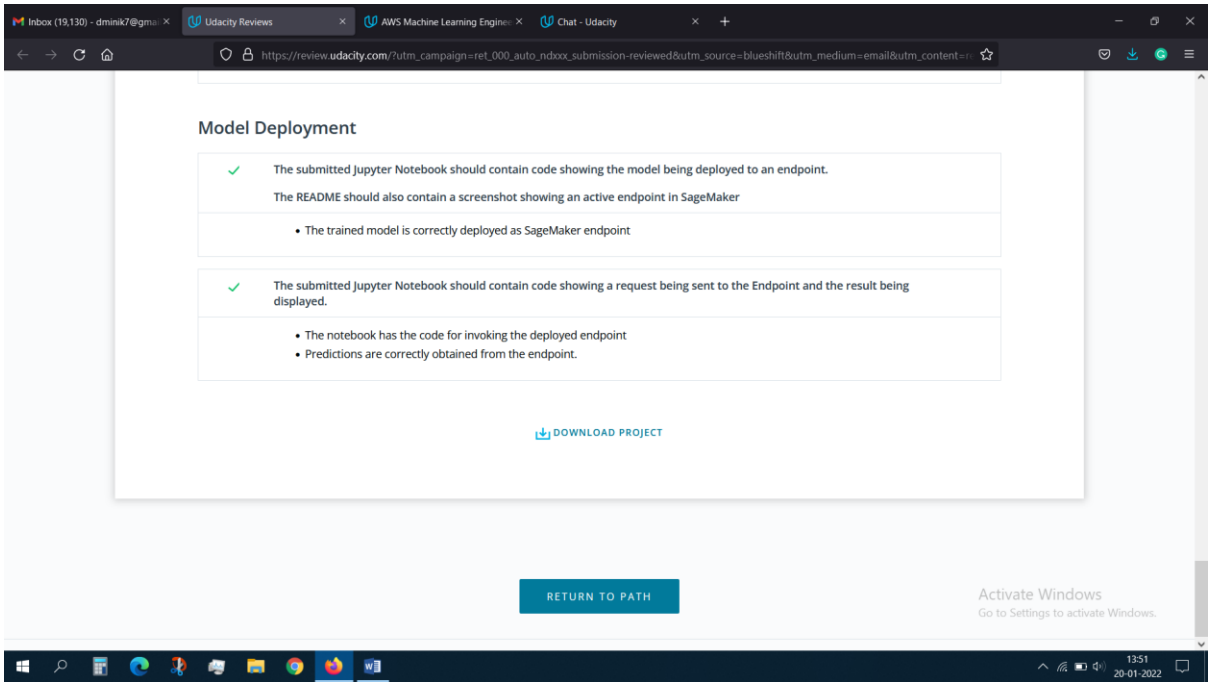

Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
The submitted python scripts contain code that logs important values and events so that they can be viewed in AWS CloudWatch. You should log your
loss or accuracy
hyperparameter values
The submitted training script contains code that logs important values and events. Good logging is important both for debugging as well as for hyperparameter tuning. These logs can be viewed through CloudWatch and help us get more insight into the training process.
Further reading
https://aws.amazon.com/cloudwatch/features/
The submitted project contains a README file that contains:
A short introduction of the project
Project Setup Instructions
Explanations of the different files used in the project
Code sample for querying a model endpoint
Explain insights from the model
At least 2 relevant and informative images, terminal outputs, or screenshots
Good start with the README:
✅A short introduction of the project
✅Project Setup Instructions
❌Explanations of the different files used in the project
Please explain the scripts(.py) files used.
❌Code sample for querying a model endpoint
Please include code sample invoking the endpoint
✅Explain insights from the model
✅At least 2 relevant and informative images, terminal outputs, or screenshots
Activate Windows
Go to Settings to activate Windows.
13:50
20-01-2022


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
Hyperparameter Optimization
Fine-tune on a pre-trained model and The choice of pre-trained model is up to you.
Resnet34 is used as the pre-trained model. Now we have a wide range of pre-trained models to choose from. Some are faster and some offer better accuracy.
Further reading:
4 Pre-Trained CNN Models to Use for Computer Vision with Transfer Learning
https://towardsdatascience.com/4-pre-trained-cnn-models-to-use-for-computer-vision-with-transfer-learning-885cb1b2dfc
While two is the minimum, you are encouraged to tune more hyperparameters to improve the performance of your model. The code should define a clear search space for the hyperparameters of the model.
The code should:
Log hyperparameter values
Log the model training and testing accuracy for each set of hyperparameters used
Two parameters are selected for hyperparameter tuning. Try some other parameters as well, for example, momentum, and weight decay. These options depend upon the pre-trained model.
The submitted README file contains a screenshot of your hyperparameter tuning job that shows at least 2 different training runs and their results.
The README has the screenshots from hyperparameter optimization runs
Deep learning models are full of hyper-parameters and finding the best configuration for these parameters in such a high dimensional space is not an easy job. The tremendous computing power offered in cloud platforms like AWS, Azure, GCP etc makes it less harder and more practical
Activate Windows
Go to Settings to activate Windows.
13:50
20-01-2022


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
The submitted README file contains a screenshot of your hyperparameter tuning job that shows at least 2 different training runs and their results.
The README has the screenshots from hyperparameter optimization runs
Deep learning models are full of hyper-parameters and finding the best configuration for these parameters in such a high dimensional space is not an easy job. The tremendous computing power offered in cloud platforms like AWS, Azure, GCP etc makes it less harder and more practical
The submitted Jupyter notebook contains details of the hyperparameters of the best model. These should include:
The type or name of the hyperparameters
The original search space of those hyperparameters
The value of the hyperparameter of the most accurate model
The best configurations of the given two parameters are identified:
In [10]:
# TODO: Get the best estimators and the best HPs
best_estimator = tuner.best_estimator()
# Getting the hyperparameters of the best trained model
best_estimator.hyperparameters()
2022-01-19 16:29:24 Starting - Preparing the instances for trai
2022-01-19 16:29:24 Downloading - Downloading input data
2022-01-19 16:29:24 Training - Training image download complete
2022-01-19 16:29:24 Uploading - Uploading generated training mo
2022-01-19 16:29:24 Completed - Training job completed
Out[10]:
{'_tuning_objective_metric': '"Test Loss"',
'batch_size': '"32"',
'lr': '0.05099525996611782 6',
'sagemaker_container_log_level': '20',
Activate Windows
Go to Settings to activate Windows.
13:51
20-01-2022


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
Profiler and Debugger
The code should extract the hyperparameters of the best model and use those hyperparameters to create a new estimator and train it.
The code should be using the hyperparameter tuning estimator to select the best model.
• The new estimator is trained based on the hyperparameters from the previous step.
The Jupyter notebook should include code to generate the model profiler report as well as any model debugging outputs generated during the training of the model.
• The profiler report is created successfully.
Here's a hands-on demo about profiling machine learning models
https://julsimon.medium.com/introducing-model-profiling-in-amazon-sagemaker-debugger-aws-re-invent-2020-df0e649f410c
The Jupyter notebook or the README should include a line plot showing the status of a variable throughout the training process.
In case the plot shows an anomalous behavior, the Jupyter notebook or README should include the steps they took to debug it. If not, it should include the steps they would have to take to debug an error.
• The validation and training losses are clearly visualized.
Steps (EVAL)
CrossEntropyLoss_output_0
val_CrossEntropyLoss_output_0
Activate Windows
Go to Settings to activate Windows.
13:51
20-01-2022


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
The Jupyter notebook or the README should include a line plot showing the status of a variable throughout the training process.
In case the plot shows an anomalous behavior, the Jupyter notebook or README should include the steps they took to debug it. If not, it should include the steps they would have to take to debug an error.
• The validation and training losses are clearly visualized.
Steps (EVAL)
CrossEntropyLoss_output_0
val_CrossEntropyLoss_output_0
Steps (TRAIN)
Activate Windows
Go to Settings to activate Windows.
Model Deployment
13:51
20-01-2022


Inbox (19,130) - dminik7@gmai
Udacity Reviews
AWS Machine Learning Engine
Chat - Udacity
https://review.udacity.com/?utm_campaign=ret_000_auto_ndxxx_submission-reviewed&utm_source=blueshift&utm_medium=email&utm_content=r
Model Deployment
The submitted Jupyter Notebook should contain code showing the model being deployed to an endpoint.
The README should also contain a screenshot showing an active endpoint in SageMaker
• The trained model is correctly deployed as SageMaker endpoint
The submitted Jupyter Notebook should contain code showing a request being sent to the Endpoint and the result being displayed.
• The notebook has the code for invoking the deployed endpoint
• Predictions are correctly obtained from the endpoint.
DOWNLOAD PROJECT
RETURN TO PATH
Activate Windows
Go to Settings to activate Windows.
13:51
20-01-2022