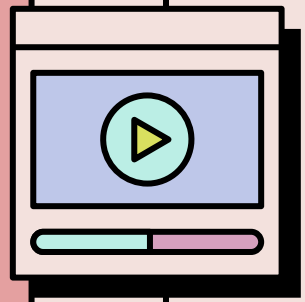
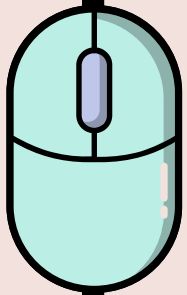
TMI Reddit Content Moderation Week 7





Scoping

1. Debiasing Using Training Data -> what Team 2

2. **Hyperparameter Tuning -> Team 1**
    - Learning rate
    - Number of neurons
    - Momentum
3. Adding Layers and attention heads

Key Updates

Limited Scope for Hyperparameter Tuning:

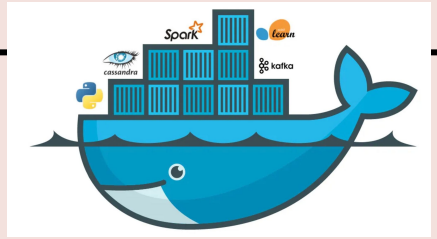
- We believed hyperparameter tuning was restricted to predefined parameters in specific models like **BERT**. (params : learning rate, batch size, number of epochs)

Using Docker allows us to tune custom models and define any number of hyperparameters, including: Number of neurons in layers, Learning rate and momentum, architectural elements like layer sizes or dropout rates

What we can do:

Define Own Model Architecture: We can write any model architecture in our `task.py` file (e.g., CNN, RNN, Transformer).
Use Pretrained Models: Use pretrained models like BERT and fine-tune them
Train Models from Scratch: Build a model completely from scratch

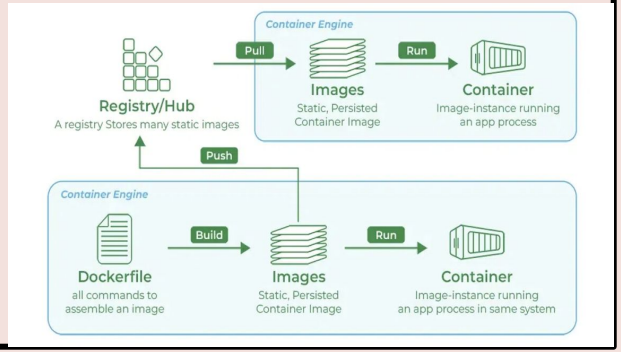
Docker



What is Docker? : Imagine a container that has everything your project needs to run—code, libraries, and dependencies -all packaged together.

Code : The training code (`task.py`).
Libraries: TensorFlow, `tensorflow-datasets`, and `cloudml-hypertune`.(report metrics for hyperparam on Vertex AI)
Base image : `tensorflow/tensorflow:2.5.0-gpu` (provide GPU and support Tensorflow as a foundation)

Create a Dockerfile:
Define the tools, libraries, and dependencies needed
Build a Docker Image:
Package the training script (`task.py`) and
dependencies into a self-contained environment.
Push the Docker Image to Google Cloud:
Upload the image to Google Container Registry for easy access during training.
Run the Training Job on Vertex AI:
Use the Docker image to train models and tune multiple hyperparameters.



```
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ touch Dockerfile
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ mkdir trainer
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ touch trainer/task.py
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ PROJECT_ID='pvergadia-demo'
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ IMAGE_URI="gcr.io/$PROJECT_ID/horse-human:hypertune"
(base) jupyter@hyperparametertuning-demo:~/horses_or_humans$ docker build ./ -t $IMAGE_URI

Sending build context to Docker daemon  11.78kB
Step 1/5 : FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
latest: Pulling from deeplearning-platform-release/tf2-gpu.2-5
25fa05cd42bd: Pulling fs layer
2d6e353a95ec: Pulling fs layer
14d7996407de: Pulling fs layer
0c9c6fc70f16: Waiting
c3c76be11512: Waiting
 Digest: sha256:0cb24474909c8ef0a3772c64a0fd1cf4e5ff2b806d39fd36abf716d6ea7eefb3
 Status: Downloaded newer image for tensorflow/tensorflow:2.5.0-gpu
  ---> 8b9d78381e5d
 Step 2/5 : WORKDIR /
  ---> Running in 46d8b5e660ae
 Removing intermediate container 46d8b5e660ae
  ---> ee44a069b29f
 Step 3/5 : RUN pip install cloudml-hypertune
  ---> Running in 60bc19e55a1b
 Collecting cloudml-hypertune
   Downloading cloudml-hypertune-0.1.0.dev6.tar.gz (3.2 kB)
 Building wheels for collected packages: cloudml-hypertune
   Building wheel for cloudml-hypertune (setup.py): started
   Building wheel for cloudml-hypertune (setup.py): finished with status 'done'
   Created wheel for cloudml-hypertune: filename=cloudml_hypertune-0.1.0.dev6-py2.py3-none-any.whl size=3988 sha256=124548786d47e36a0b14127b89bd7de7abdfb2cfcd8586b47b85309cc2a17bed
   Stored in directory: /root/.cache/pip/wheels/77/12/d3/08eb4af729a5de0297d87666c0bdd98c3689df316e80051c4e
 Successfully built cloudml-hypertune
 Installing collected packages: cloudml-hypertune
 Successfully installed cloudml-hypertune-0.1.0.dev6
 WARNING: You are using pip version 20.2.4; however, version 21.3.1 is available.
 You should consider upgrading via the '/usr/bin/python3 -m pip install --upgrade pip' command.
 Removing intermediate container 60bc19e55a1b
  ---> 9b90f11d3c2c
 Step 4/5 : COPY trainer /trainer
  ---> 06434b3fa0f8
 Step 5/5 : ENTRYPOINT ["python", "-m", "trainer.task"]
  ---> Running in 2cdc693aae65
 Removing intermediate container 2cdc693aae65
  ---> d388f4261fea
 Successfully built d388f4261fea
 Successfully tagged gcr.io/tmi-rcm-new/horse-human:hypertune
 The push refers to repository [gcr.io/tmi-rcm-new/horse-human]
```

Error Last Time

Sources of error I have gone through
1. Base Image not found 2. CPU quota exceeded 3. Missing tensorflow_datasets(docker dependency issue).....and more

Training pipeline failed with error message: Hyperparameter Tuning Trial #1 Failed before any other successful trials were completed. The failed trial had parameters: num_neuron=128, learning_rate=0.10000000000000002, momentum=0.5, . The trial's error message was: The replica workerpool0-0 exited with a non-zero status of 1. Termination reason: Error. To find out more about why your job exited please check the logs: https://console.cloud.google.com/logs/viewer?project=17243723720&resource=ml_job%2Fjob_id%2F2603291931370848256&advancedFilter=resource.type%3D%22ml_job%22%0Aresource.labels.job_id%3D%22260329193137084825 6%22

Status	Failed
Training pipeline ID	3667918254220771328
Created	Dec 1, 2024, 7:17:41 AM
Start time	Dec 1, 2024, 7:17:41 AM
Elapsed time	18 min 36 sec
Region	us-central1
Encryption type	Google-managed
Algorithm	Custom training
Objective	Custom
Metric to optimize	accuracy
Goal	Maximize

Log Explorer tells what's