Docker for Science

Outline

- What is Docker?
- Workflow
- Use cases
  - Portable environments
  - Reproducibility
- Demo
- Related technologies

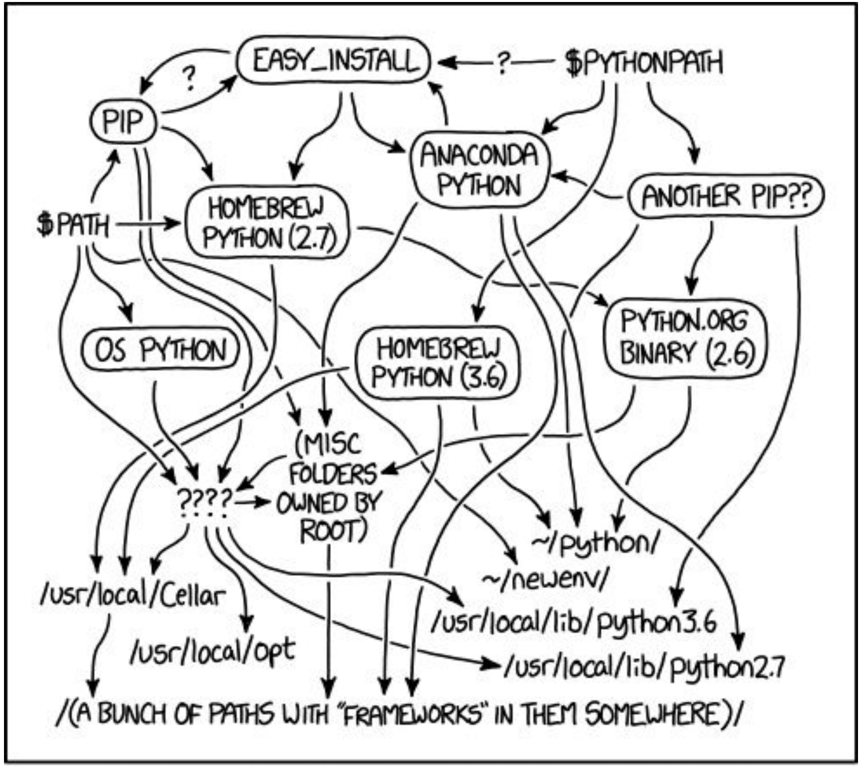
What is Docker?

- An operating system level virtualization platform based on Linux

Workflow

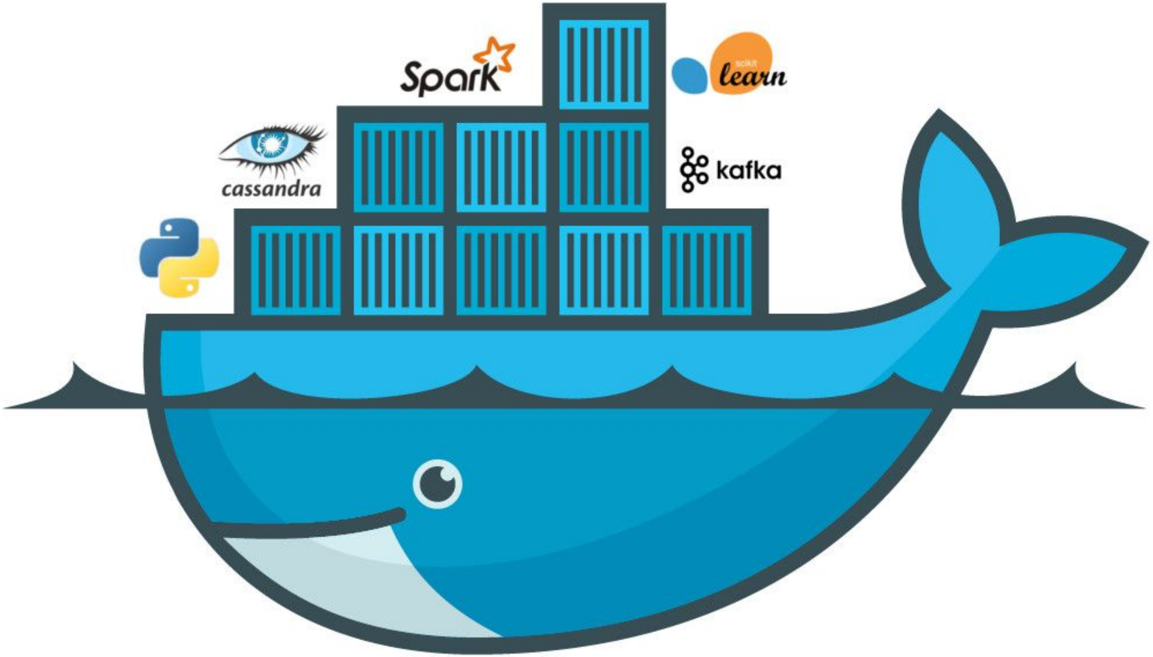
- Build Docker **images**
- **Push** images to Docker Hub
- Run Docker **containers**
- Deploy and scale applications

Use Case: Portable Environments



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Use Case: Portable Environments



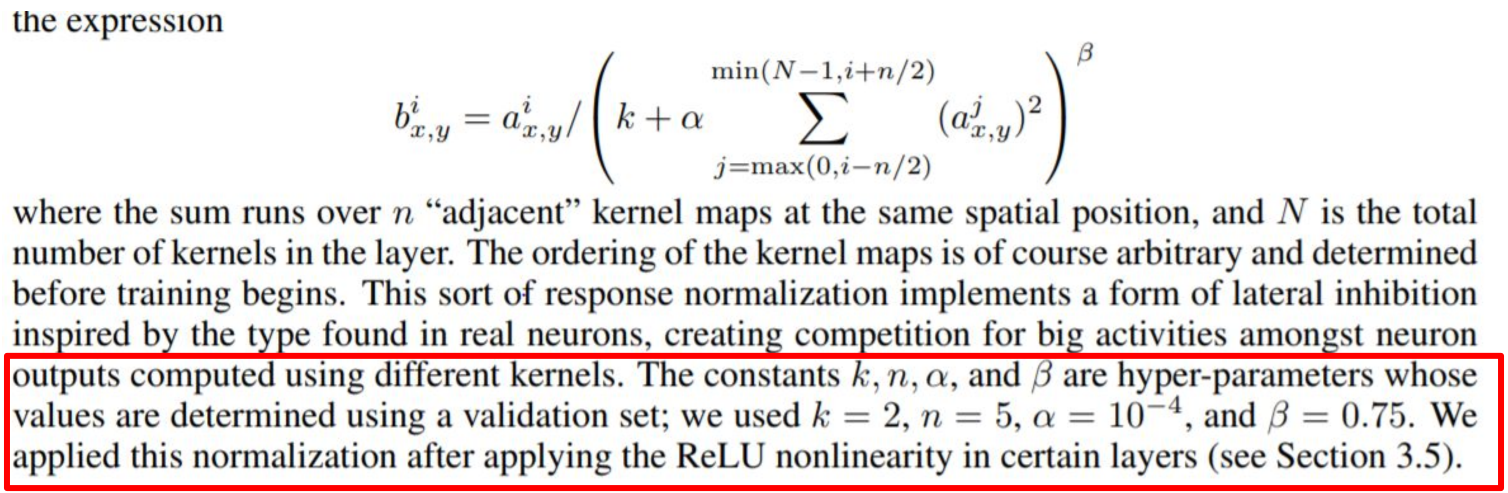
Use Case: Reproducibility

Use Case: Reproducibility

Lipton and Steinhardt: “Troubling trends of machine learning scholarship”

- Failure to distinguish between explanation and speculation
- **Failure to identify sources of empirical gains**
- Mathiness
- Misuse of language

Use Case: Reproducibility



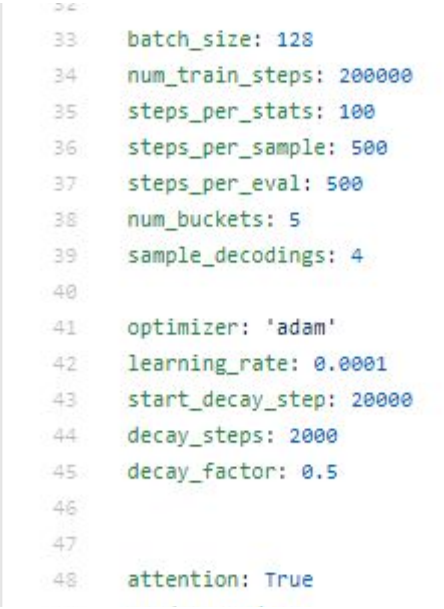
the expression

$$b^i_{x,y} = a^i_{x,y} \Big/ \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

where the sum runs over n “adjacent” kernel maps at the same spatial position, and N is the total number of kernels in the layer. The ordering of the kernel maps is of course arbitrary and determined before training begins. This sort of response normalization implements a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuron outputs computed using different kernels. The constants k, n, α, and β are hyper-parameters whose values are determined using a validation set; we used $k = 2$, $n = 5$, $\alpha = 10^{-4}$, and $\beta = 0.75$. We applied this normalization after applying the ReLU nonlinearity in certain layers (see Section 3.5).

Not ideal...

Use Case: Reproducibility



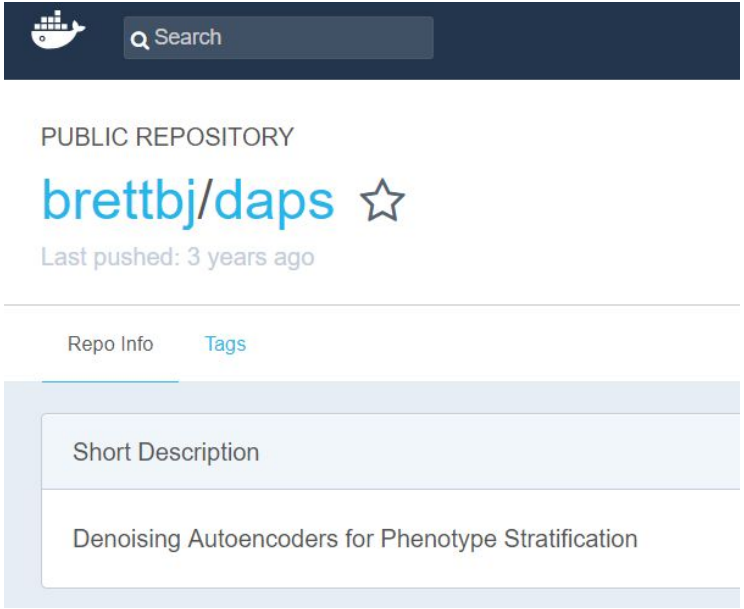
```
batch_size: 128
num_train_steps: 200000
steps_per_stats: 100
steps_per_sample: 500
steps_per_eval: 500
num_buckets: 5
sample_decodings: 4

optimizer: 'adam'
learning_rate: 0.0001
start_decay_step: 20000
decay_steps: 2000
decay_factor: 0.5


attention: True
```

Better...

Use Case: Reproducibility



PUBLIC REPOSITORY

brettbj/daps ☆

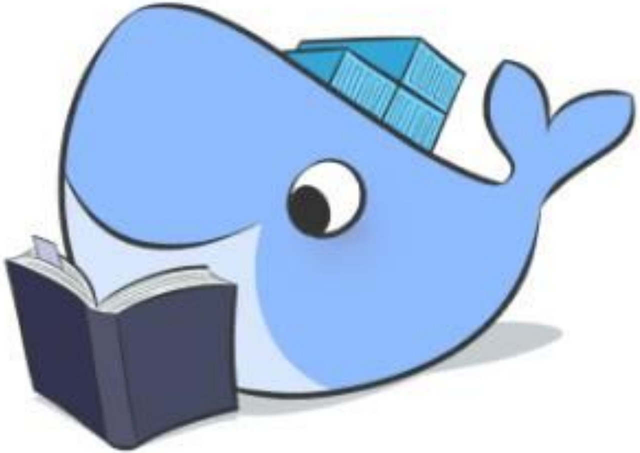
Last pushed: 3 years ago

Short Description

Denoising Autoencoders for Phenotype Stratification

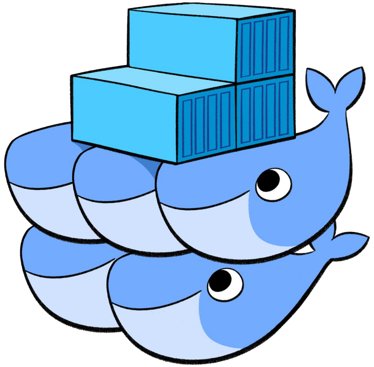
Great!

Demo



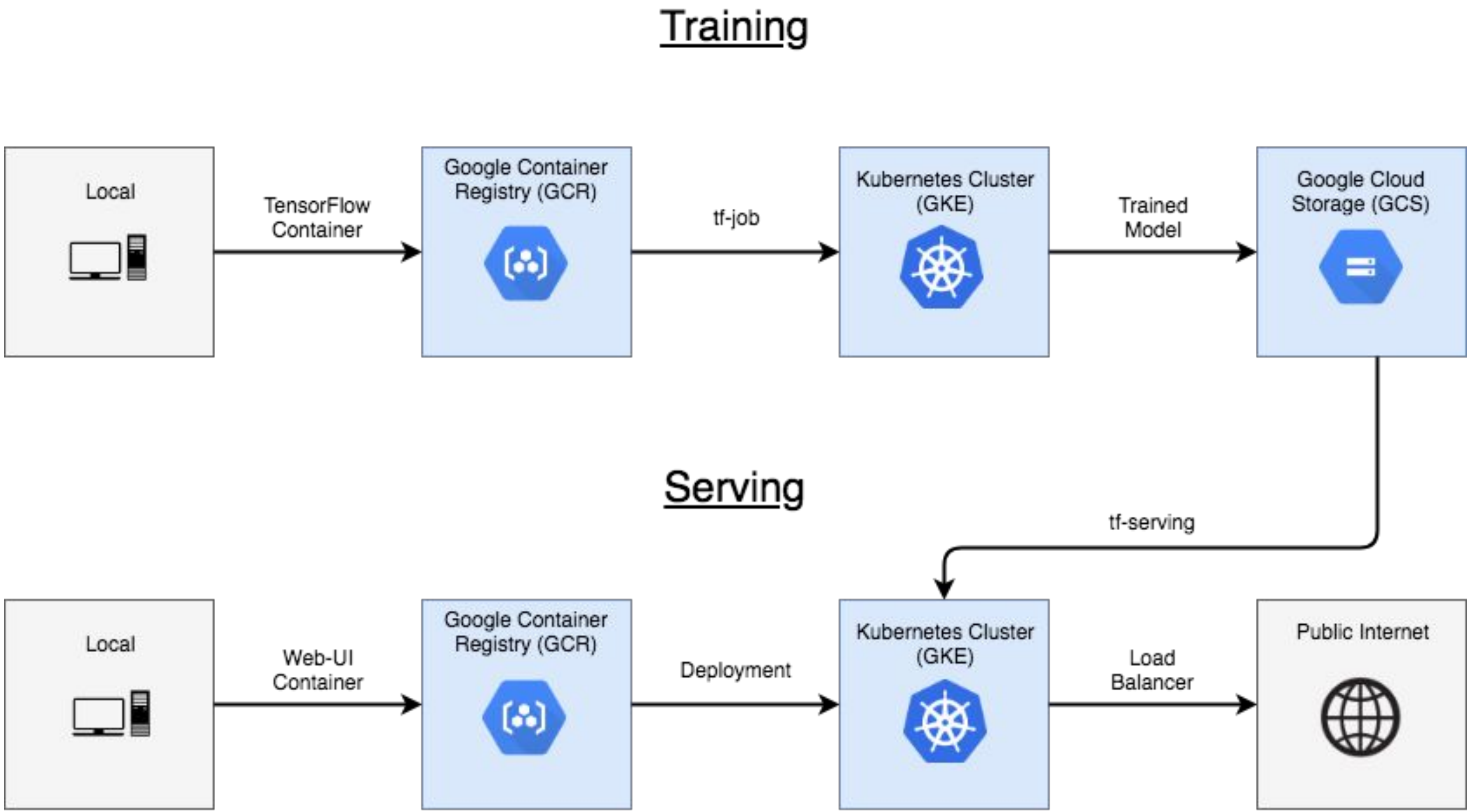
Related Technologies

- Docker Swarm + Compose
- Kubernetes



Kubernetes

Kubeflow


Training
Local
TensorFlow Container
Google Container Registry (GCR)
tf-job
Kubernetes Cluster (GKE)
Trained Model
Google Cloud Storage (GCS)
tf-serving
Serving
Local
Web-UI Container
Google Container Registry (GCR)
Deployment
Kubernetes Cluster (GKE)
Load Balancer
Public Internet

Links of interest

- Deep learning container: https://github.com/floydhub/dl-docker
- Troubling Trends in ML: https://arxiv.org/pdf/1807.03341.pdf
- Jupyter Docker stacks: https://github.com/jupyter/docker-stacks
- Auto-sklearn: https://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning
- Kubeflow: https://www.kubeflow.org/