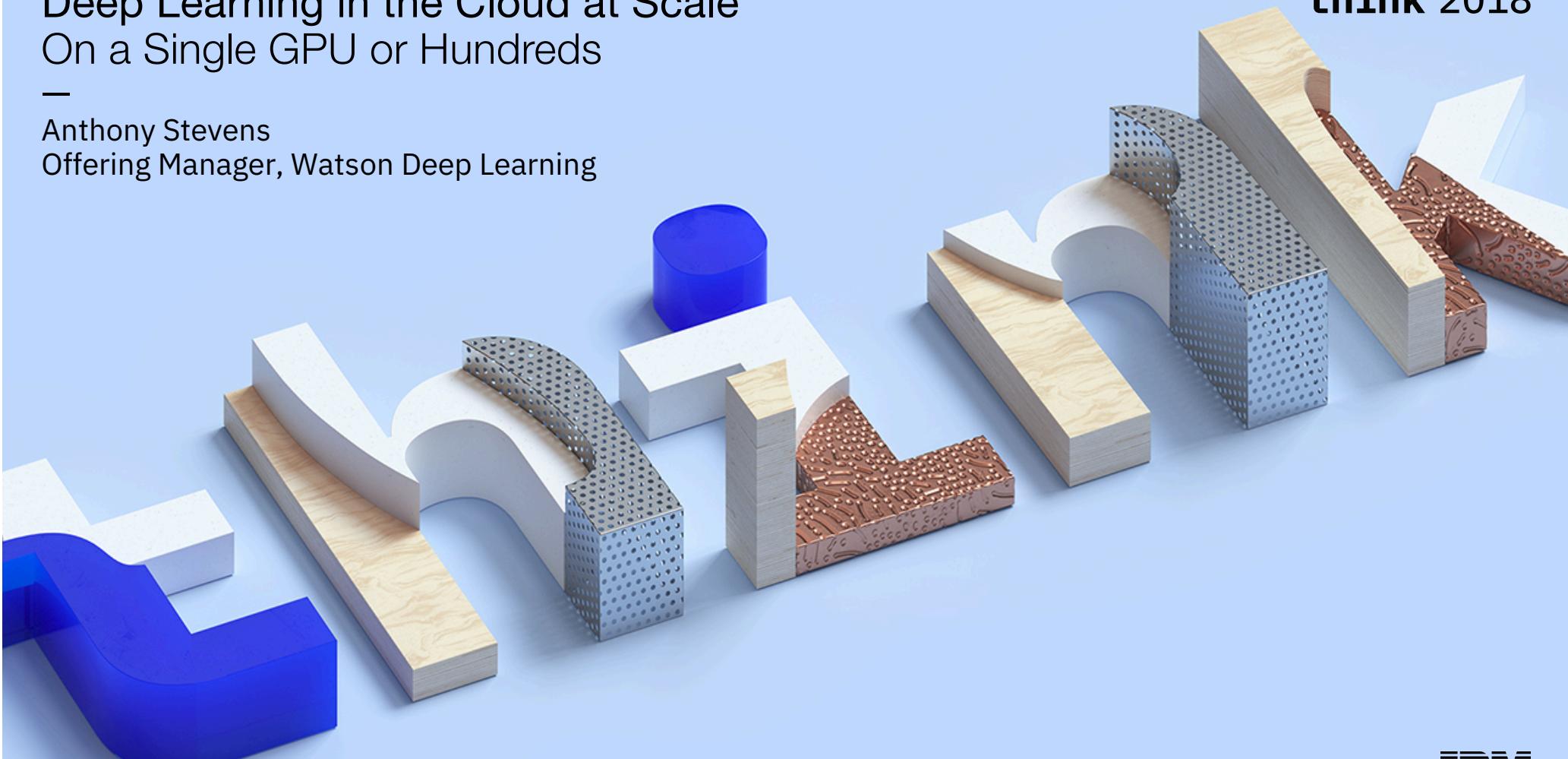


think 2018

Deep Learning in the Cloud at Scale On a Single GPU or Hundreds

—
Anthony Stevens
Offering Manager, Watson Deep Learning



Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

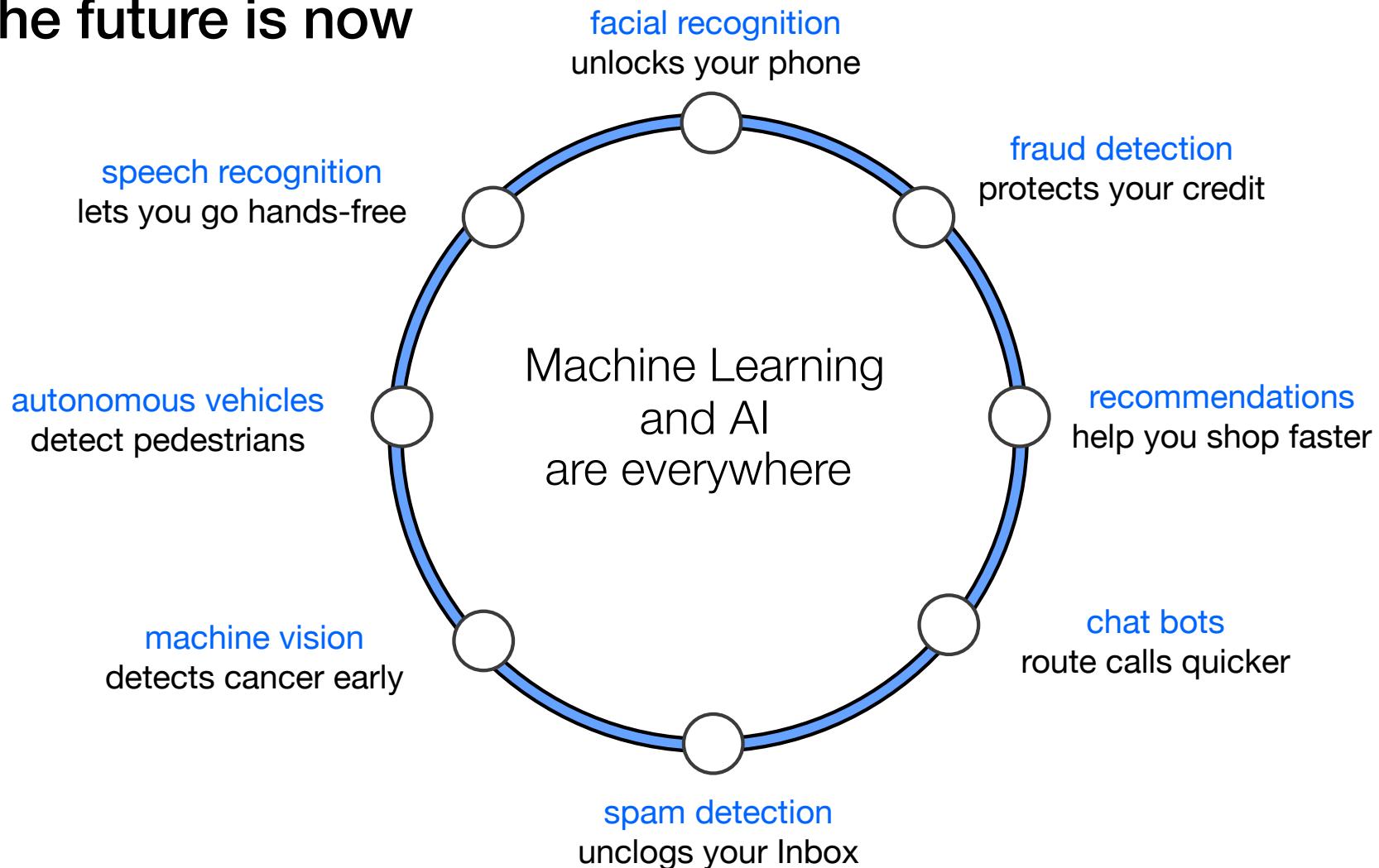
The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Quick Intro to Deep Learning

IBM

The future is now



Use Cases for Deep Learning

Cyber Defense

IoT

Earth Monitoring

Advanced Physics Research

Drug Discovery

Medical Decision-Making

Climate Change

Supply Chain Management

Fraud Detection

Smart Cities

Weather Forecasting

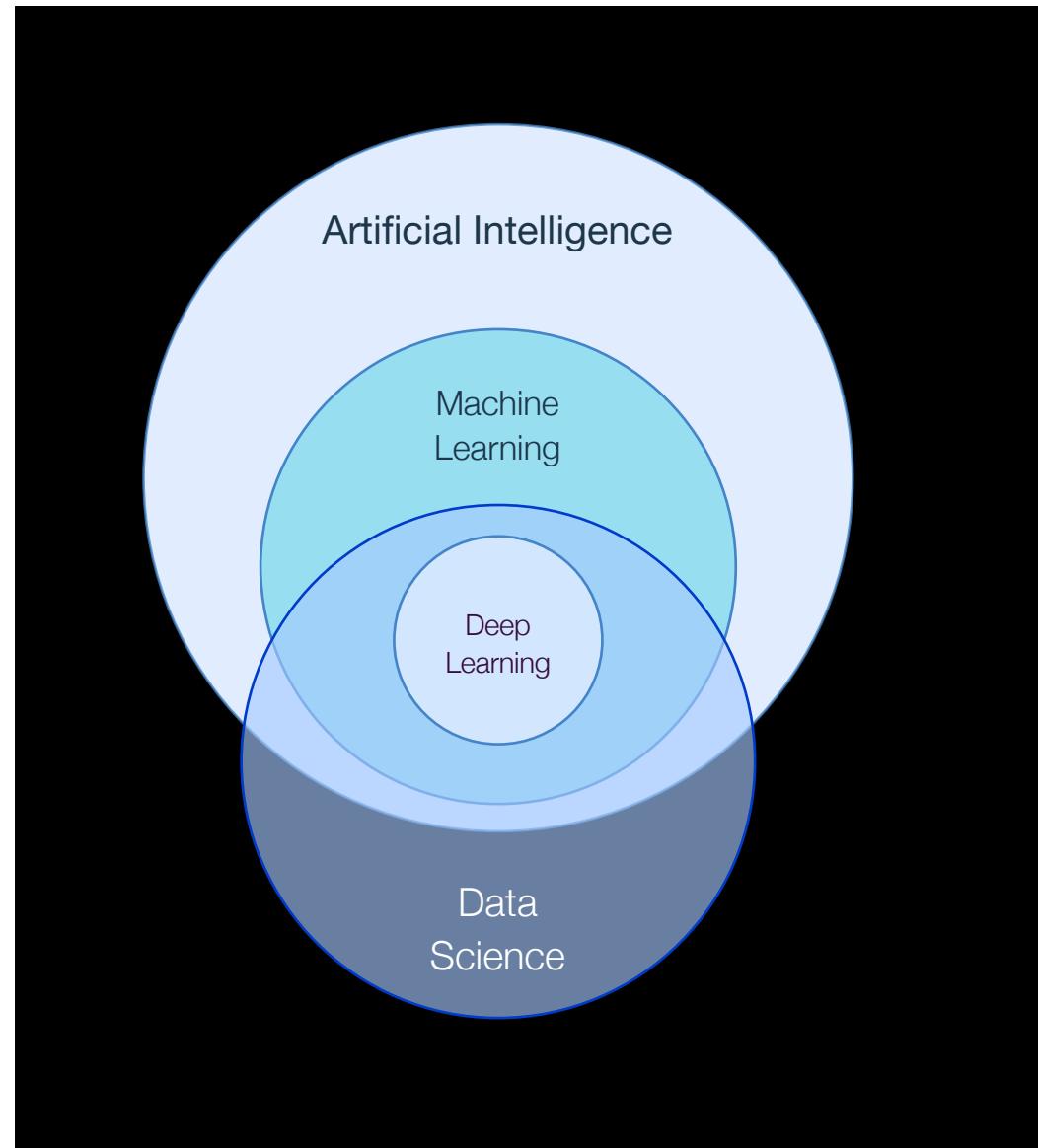
Robotics

Aeronautics

Disease Diagnostics

Media Analytics

How does machine learning
relate to data science and AI?



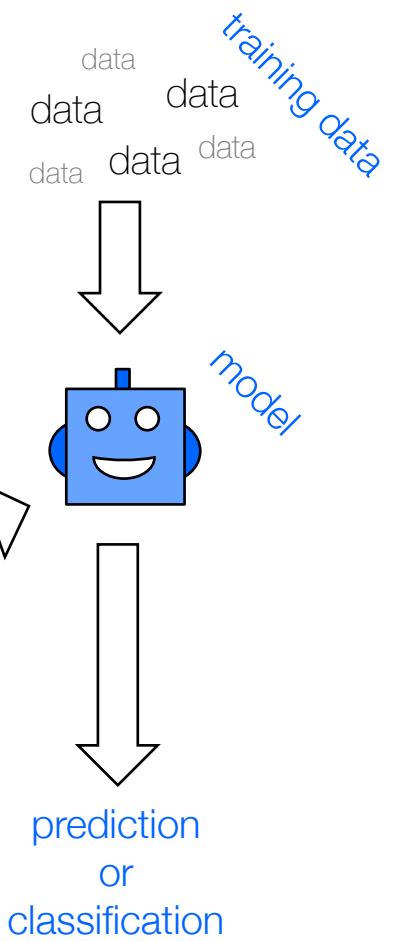
How does machine learning work? 

Machine learning requires
TONS OF DATA 

- 1 A machine learning model is trained to recognize patterns in historical data

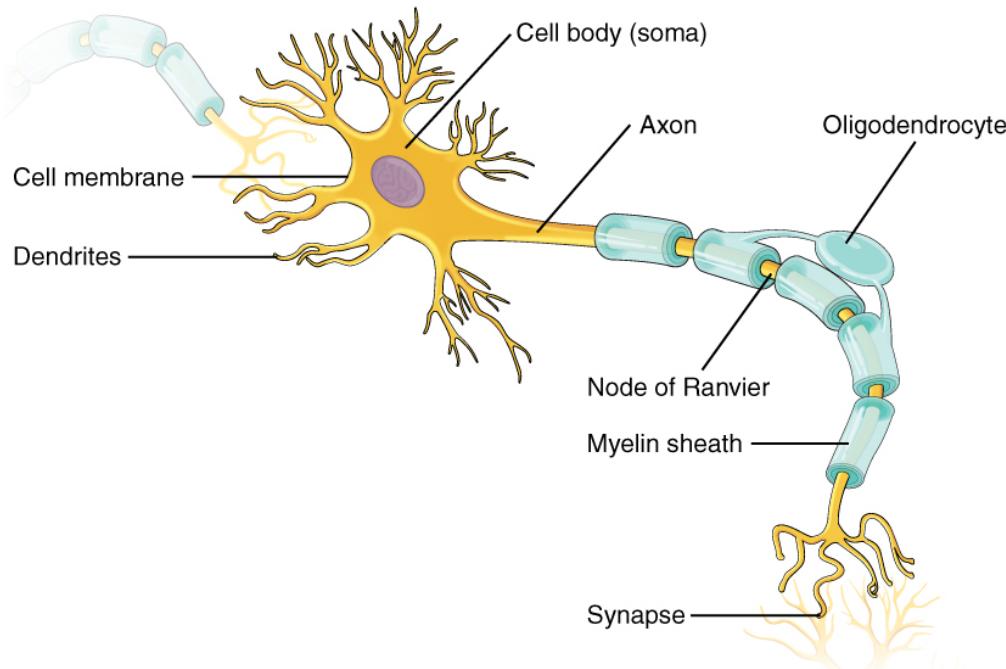
- 2 The model is then shown new data and asked to predict or classify it.

- 3 If patterns in the new data match the training data then the model makes accurate predictions



Deep Learning = Training Artificial Neural Networks

Based on biological neurons. Artificial neurons learn by recognizing patterns in data.



A human brain has:

- 200 billion neurons
 - 32 trillion connections between them
- ➡ Artificial neural networks have far fewer

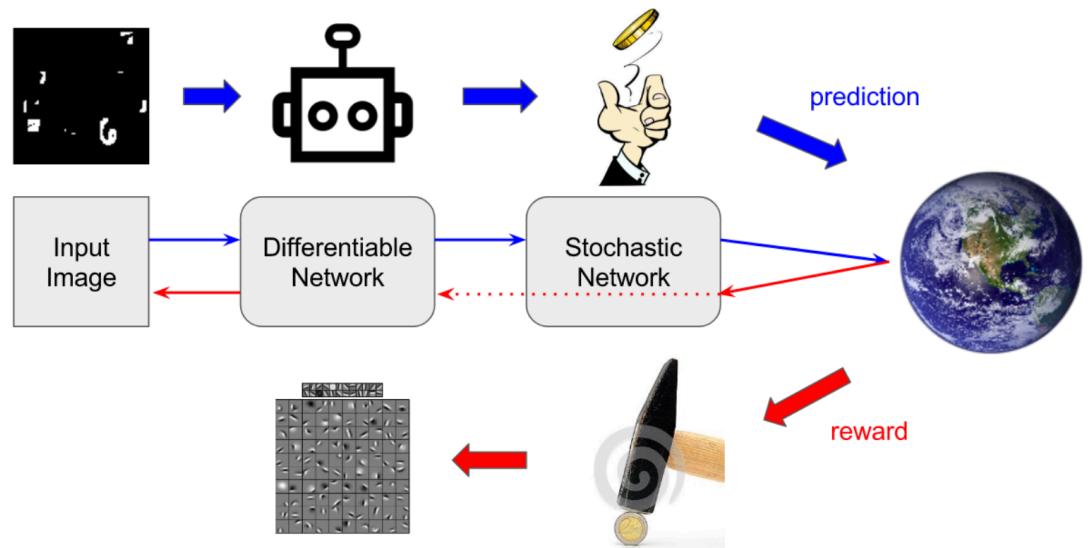
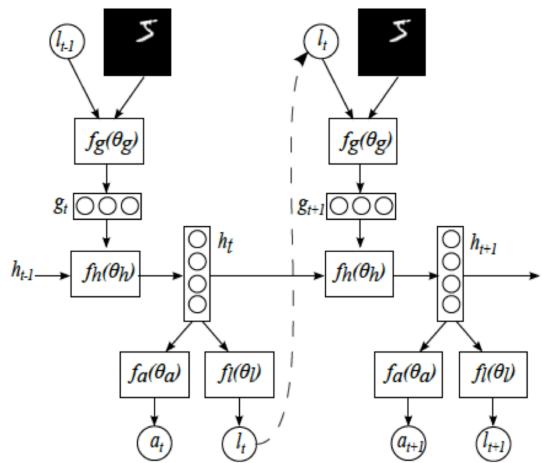
Machine Learning is algorithm selection

Deep learning is neural network design

AI is systems architecture

Augmented MNIST^[1]

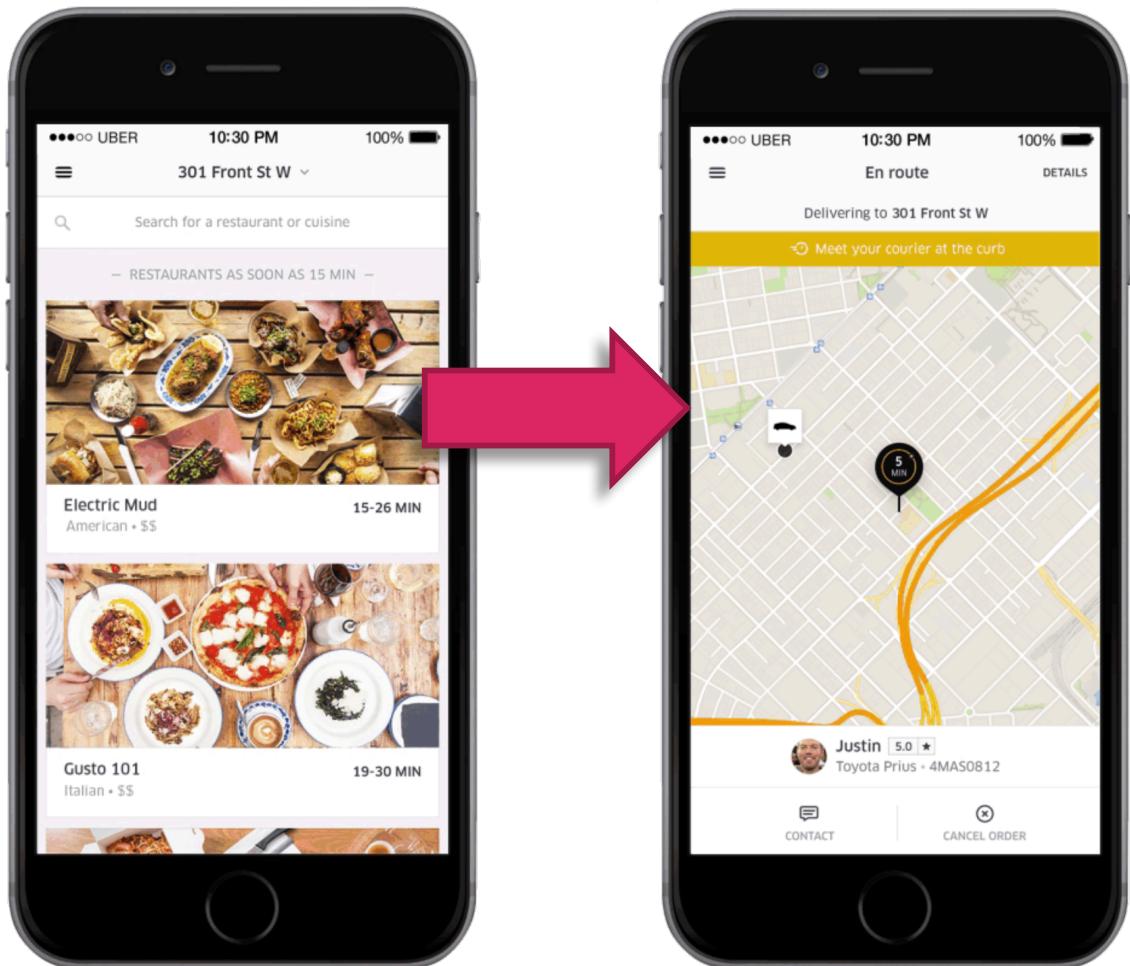
glimpse sensor
 location network
 activation network
 recurrent neural network
 LSTM
 reinforcement learning
 policy objective





Uber Eats requires multiple models to ensure accuracy of food delivery

- ✓ search autocomplete
- ✓ search rankings
- ✓ restaurant rankings
- ✓ meal delivery time predictions
- ✓ nearby driver selection
- ✓ driver incentives
- ✓ customer satisfaction analysis



<https://eng.uber.com/machine-learning/>
<https://eng.uber.com/michelangelo/>

Machine learning requires
a data science **team**.

And **teams** require tools
to collaborate.



Data Engineer



Data Scientist



Business Analyst



ML Ops



Application
Developer

Most AI solutions require neural networks that don't yet exist

pre-trained
neural network



Application
Developer

transfer learning



SME

+



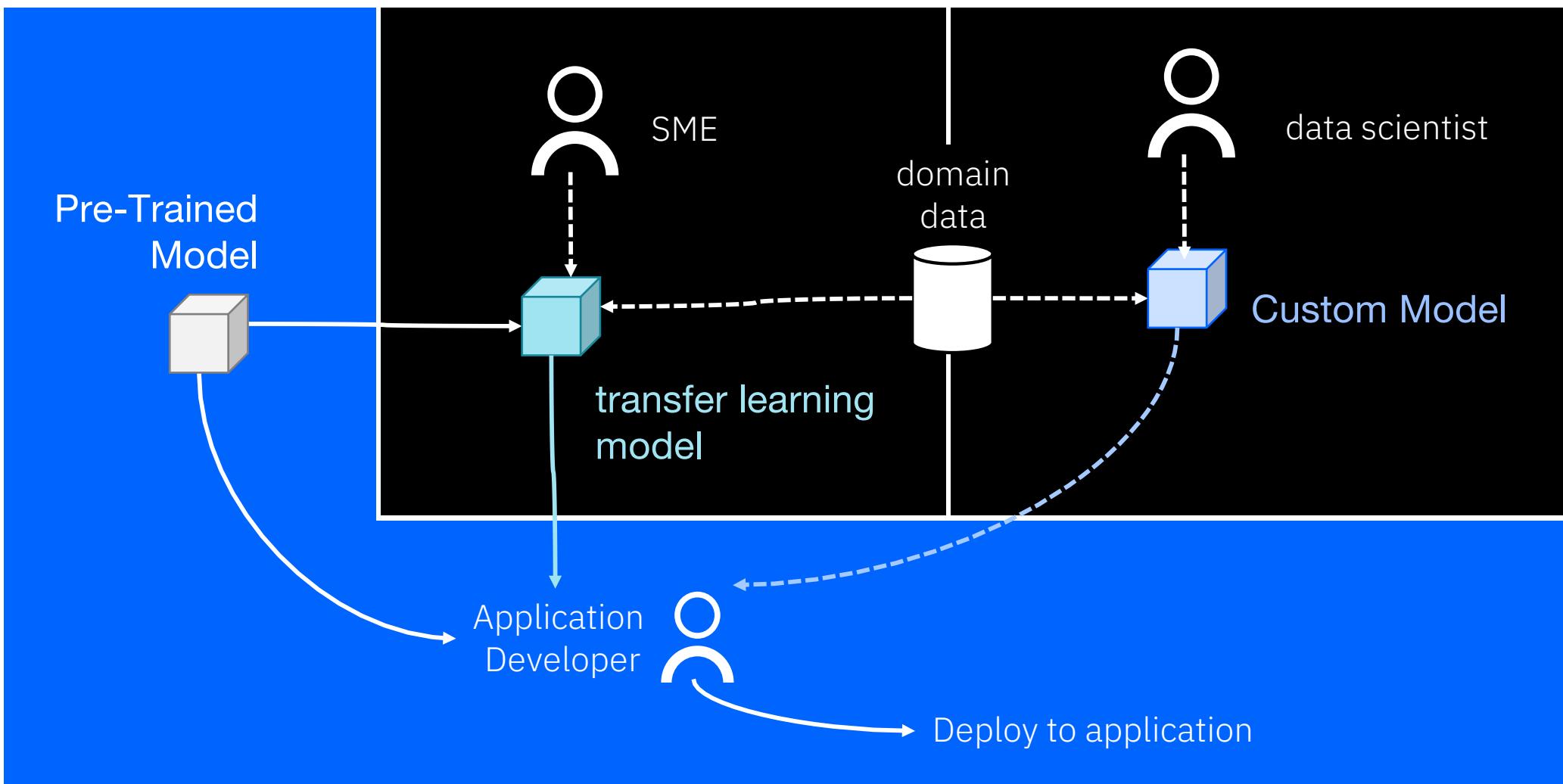
domain
data

custom
neural networks
required



data scientist

How do the paths to AI-powered apps relate?



Deep Learning in Watson Studio

Neural Network
Design

Experiment
Management

Technical
Deep Dive

Model
Hosting

Neural Network Modeler (beta)

An intuitive drag-and-drop, no-code interface for designing neural network structures using the most popular deep learning frameworks. Quickly capture your network design then single click export for experimental optimization.

Supported Frameworks



Caffe



The screenshot shows the Deep Learning Editor interface with the following annotations:

- Drag-and-drop network layers**: A bracket on the left side of the interface points to the search bar and the main canvas area where nodes are connected by arrows.
- Real-time validation of network flow**: An arrow points from the top center to the network diagram.
- Define layer configuration**: An arrow points to the right panel for a **Dense** layer, which includes fields for Weight Regularizer (L1, L2, L1-L2, null), Weight LR Multiplier (1), Weight Decay Multiplier (1), Bias Constraint (maxnorm, nongeq, unifnorm, null), Bias Regularizer (L1, L2, L1-L2, null), Bias Decay Multiplier (1), Activity Regularizer (L1, L2, L1-L2, null), and a **Save** button.
- Choose optimizer params**: An arrow points to the right panel for a **Dense** layer, specifically highlighting the **Weight Regularizer** section.
- Generate CPU or GPU compatible code**: An arrow points to the bottom left of the interface.
- Save as popular framework code**: An arrow points to the bottom right of the interface.
- Export as a python notebook**: An arrow points to the bottom right of the interface.
- Execute as batch experiment**: An arrow points to the bottom right of the interface.

Neural Network
Design

Experiment
Management

Technical
Deep Dive

Model
Hosting

Discovering a single optimal neural network requires exploring
1000's of hyperparameter combinations

which means 100's of experiments

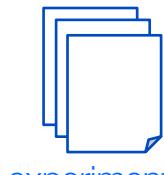
which requires 1000's of training runs

Accelerate your
experimental workflow

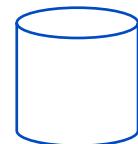
You focus on experiment design



source code



experiment
definition



dataset

We'll manage the rest

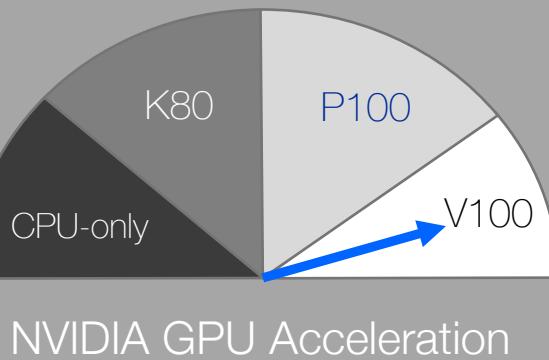
EXPERIMENTS

10's-100's

>>

TRAINING RUNS

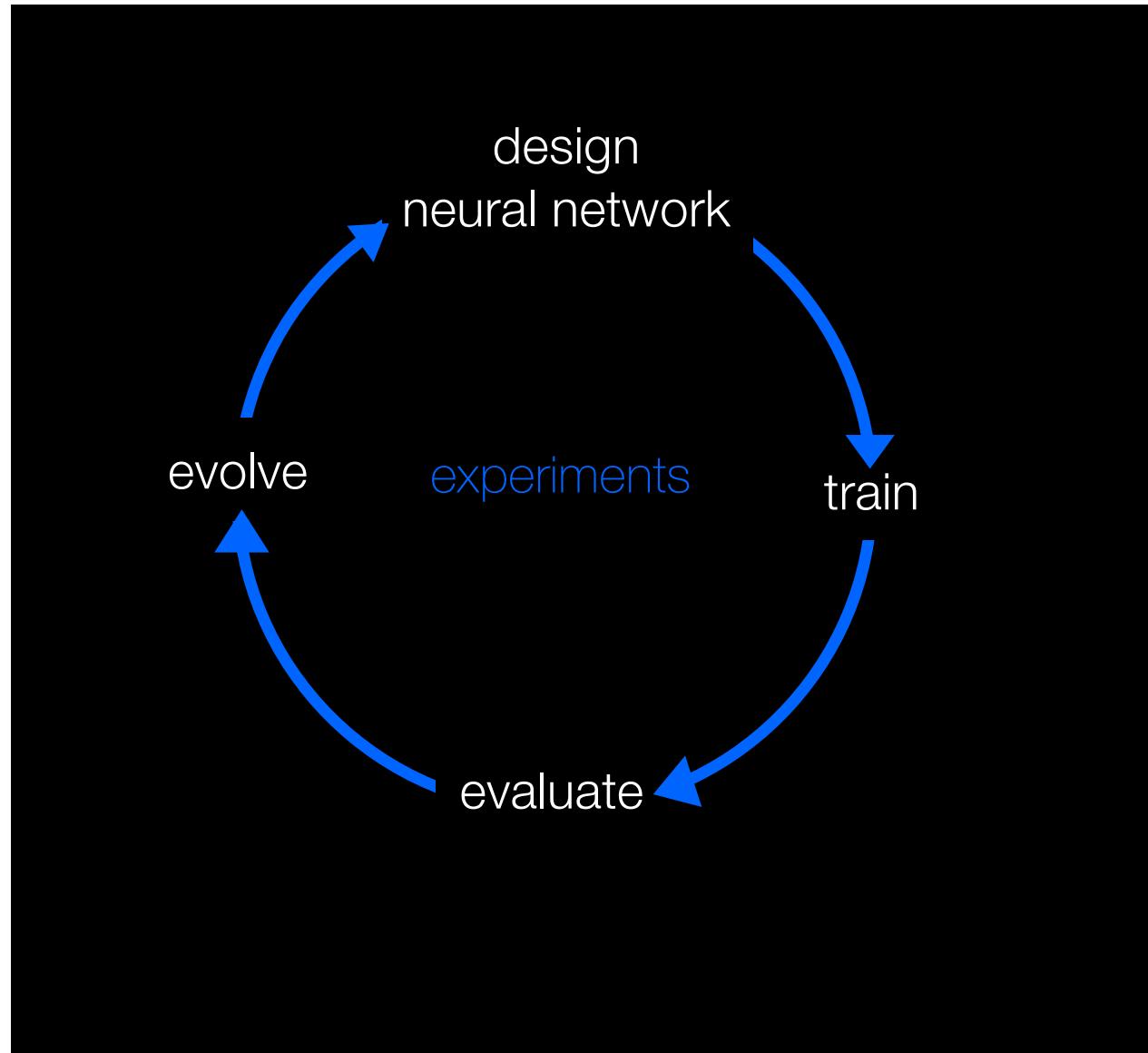
100's-10000's

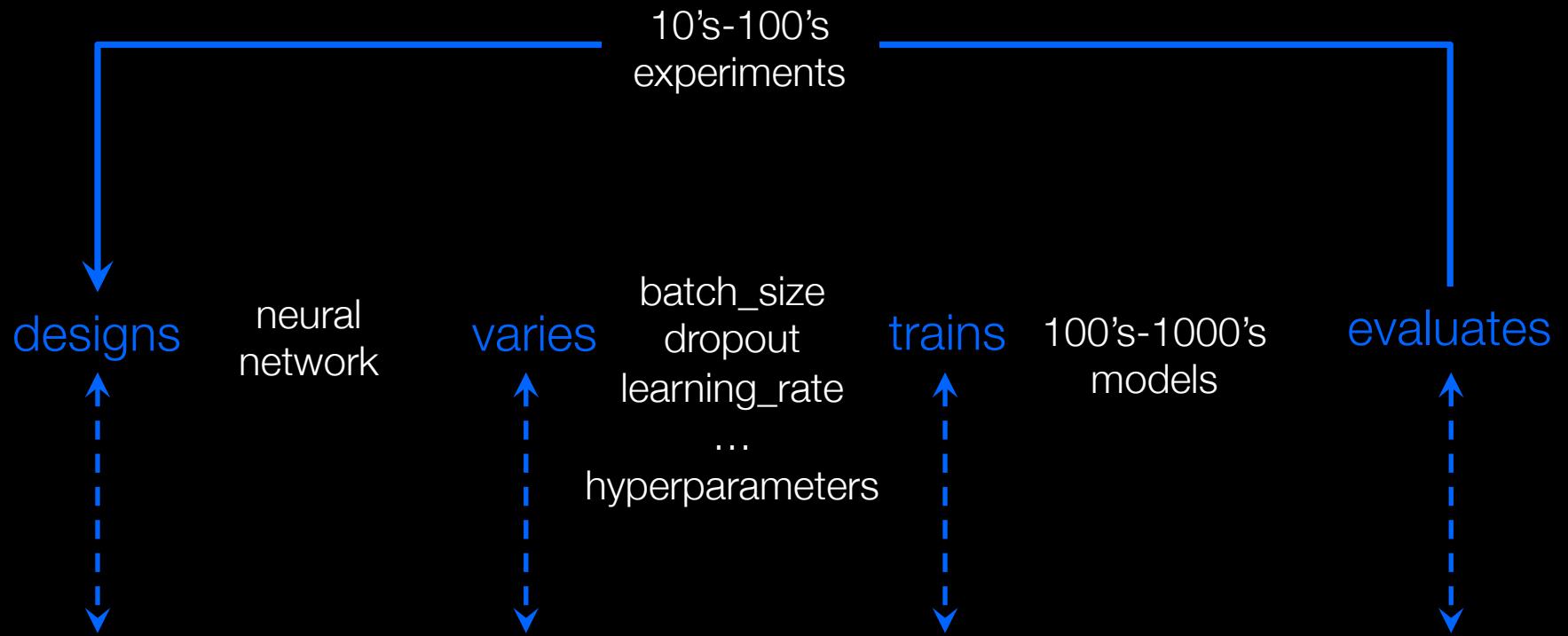
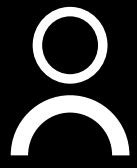


Experiment Assistant
accelerates discovering the
optimal neural networks

100's of experiments
10,000's of training runs

It's here to serve you





Experiment Assistant

supports the end-to-end workflow

Experiment Assistant

A suite of tools that manage your training runs. Each run is automatically started, monitored, and stopped upon completion. Training history and assets are tracked then automatically transferred to the customer's Object Storage for quick access.

Initiate Experiments

The screenshot shows the 'Define experiment details' section with a 'Name' field containing 'Fraud detection analysis - network #1'. A 'Description' field is present below it. On the right, there's a 'Associate training definitions' section with a 'COMPUTE PLAN' button and a note about no training definitions associated. At the bottom, there are 'Cancel' and 'Create and run' buttons.

CLI or Python Client

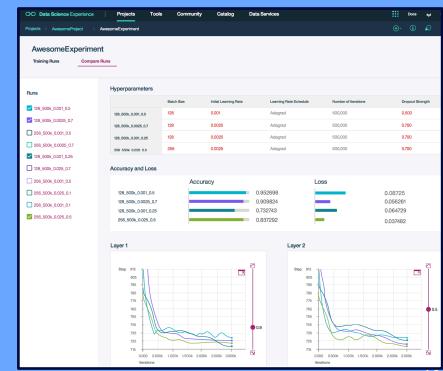
The terminal window displays a list of training runs for a 'keras-fraud-detect-csv (batch=10)' experiment. The output includes columns for ID, name, status, framework, and submission time. The status column shows various states like 'pending', 'running', and 'completed'. The terminal prompt is 'data_science -- bash - 10b.v47'.

Think 2018 / DOC ID / Month XX, 2018 / © 2018 IBM Corporation

Monitor Training in Real-Time

The interface shows a summary of the 'AwesomeExperiment' with 9 runs in total, 4 in progress, and 16 hours and 27 minutes of total running time. It lists runs under 'Queued', 'In Progress', and 'Completed' sections. Each run entry includes a status icon, name, duration, completion time, learning rate, test accuracy, and iteration count.

Compare Model Performance



Accessing Experiment Assistant

Python Client

```
File Edit View Insert Cell Kernel Help Trusted | Python [default]

[+] Run Code

python3 experiment_1.py --trainImagesFile ${DATA_DIR}/train-images-idx3-ubyte.gz --trainLabelsFile ${DATA_DIR}/train-labels-idx1-ubyte.gz --testImagesFile ${DATA_DIR}/t10k-images-idx3-ubyte.gz --testLabelsFile ${DATA_DIR}/t10k-labels-idx1-ubyte.gz --batch_size=8 --optimizer=SGD --epochs=10 --dropout=0.5

5 Define training Runs

Now you're ready to store each training runs from the grid search so they can be executed later as part of your experiment.

In [220]: def store_runs_for_grid_search(parameter_variants, experiment_number):

    training_run_definitions = []
    print("Creating %d training run definitions" % len(parameter_variants))
    for index, hyperparameters in enumerate(parameter_variants):

        # Provide a unique name
        run_num = index+1

        # Replace placeholders with run values
        command_for_run = get_command(experiment_number)
        for key, value in hyperparameters.items():
            placeholder = "%" + key + "%"
            command_for_run = command_for_run.replace(placeholder, key + value)

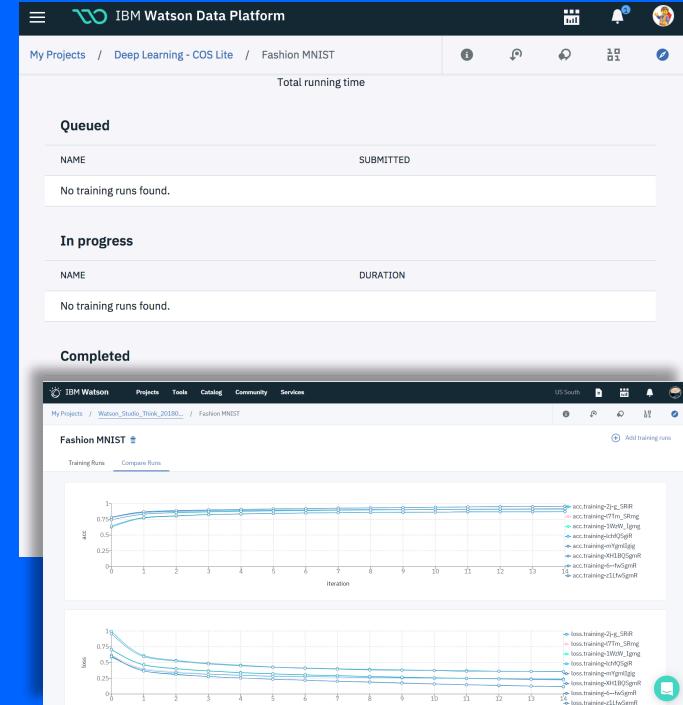
        # Store training run definition
        run = TrainingRun(hyperparameters)
        metadata = {
            client.repository.DefinitionMetaNames.NAME: run.name,
            client.repository.DefinitionMetaNames.AUTHOR_EMAIL: "pe"
            client.repository.DefinitionMetaNames.FRAMEWORK_NAME: "TensorFlow"
            client.repository.DefinitionMetaNames.FRAMEWORK_VERSION: "1.14.0"
        }
```

Command Line

```
... biospher --- bash - 117x57
..._github -- jupyter-notebook > python ... L... biospher numbers/data --bash ~ --bash
525 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=0+05_d_3=0,15_d_d=4+05_d_5=0.05 training-GWlWt6zg running
tensorflow _2018-02-27T05:58:02Z
526 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=2+05_d_3=0,15_d_d=4+05_d_5=0.15 training-SAX20p6Kr running
tensorflow _2018-02-27T05:58:03Z
527 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=2+05_d_3=0,15_d_d=4+05_d_5=0.25 training-R9620pezR pending
tensorflow _2018-02-27T05:58:07Z
528 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=2+05_d_3=0,15_d_d=4+05_d_5=0.05 training-YNkMOpzg pending
tensorflow _2018-02-27T05:58:08Z
529 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=2+05_d_3=0,15_d_d=4+05_d_5=0.15 training-PamMwt6zR running
tensorflow _2018-02-27T05:58:11Z
530 bs=256_opt=&adam_epochs=50_d_l=0,25_d_d=2+05_d_3=0,15_d_d=4+05_d_5=0.25 training-TQWMwt6Kr running
tensorflow _2018-02-27T05:58:12Z

530 records found.
OK
List all training-runs successful
blueViary> biospher ml list training-runs
Fetching the list of training runs ...
No Name Framework submitted-at guid status
1 bs=256_drop=0_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:27Z training-eSENaZekg completed
2 bs=256_drop=0.05_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:30Z training-5BBN-Wekg completed
3 bs=256_drop=0.1_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:33Z training-JUhAzegz completed
4 bs=256_drop=0.15_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:35Z training-yGChAwEZ completed
5 bs=256_drop=0.2_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:36Z training-MwqBa6Kg completed
6 bs=256_drop=0.25_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:36Z training-IKxD-2z6x completed
7 bs=256_drop=0.3_epochs=5_opt=&adam tensorflow _2018-02-23T08:03:42Z training-zApvnpzeg completed
8 bs=256_drop=0.35_epochs=5_opt=&adam tensorflow _2018-02-27T01:11:25Z training-b_oVnt6kg completed
9 bs=256_drop=0.1_epochs=5_opt=&adam tensorflow _2018-02-27T01:11:39Z training-zsJ3nt6zg completed
10 bs=256_drop=0.15_epochs=5_opt=&adam tensorflow _2018-02-27T01:11:42Z training-tR-Dpztek completed
11 bs=256_drop=0.2_epochs=5_opt=&adam tensorflow _2018-02-27T01:11:42Z training-XrBpdn6Kr completed
12 bs=256_drop=0.25_epochs=5_opt=&adam tensorflow _2018-02-27T01:11:42Z training-iOpVntekg completed
13 bs=256_drop=0_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:35Z training-Heu74t6Kk completed
14 bs=256_drop=0.05_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:40Z training-G8qo4pezR completed
15 bs=256_drop=0.1_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:41Z training-1NKAVtekg completed
16 bs=256_drop=0.15_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:43Z training-8SGd4teKR completed
17 bs=256_drop=0.20_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:45Z training-vW2D4tezg completed
18 bs=256_drop=0.25_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:47Z training-W7HA4t6zg completed
19 bs=256_drop=0.3_epochs=5_opt=&adam tensorflow _2018-02-27T01:32:47Z training-tMT4Dqek completed
```

GUI





Save time: not just money. Use your preferred IDE and existing workflows. CLI, python library, and REST access is balanced by visual debugging tools. Design and optimize your networks better and faster.



Hyperparameter Optimization: Efficiently automate searching your network's hyperparameter space to ensure the best model performance with the fewest training runs.

Code with your favorite frameworks and tools

Use your personal IDE or a Jupyter notebook hosted in DSX while using the most popular deep learning frameworks



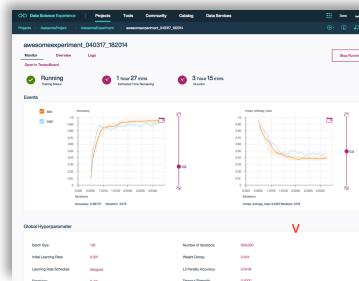
Don't be constrained. Select the framework appropriate to the unique requirements of your problem domain and skills of your team. Or try a simpler approach and use a high-level meta-framework like Keras to accelerate model exploration by abstracting away low level complexity.



Validated and Evolving Cloud Infrastructure: optimized for enterprise production environments and running on the same infrastructure hosting IBM Watson's own cognitive services.



Collaborate with team members: Share your deep learning experiments, debug your neural network structures, access common data within hosted object stores, forward versioned models to your development team then let them submit new data into your continuous learning flow.



Graphs not Log Files: Don't stare at text logs when you can overlay accuracy and loss graphs to dive deeper into the training of your neural networks. Track then view model hyperparameters so you can start understanding the progress of training individual layers in your network.

Neural Network
Design

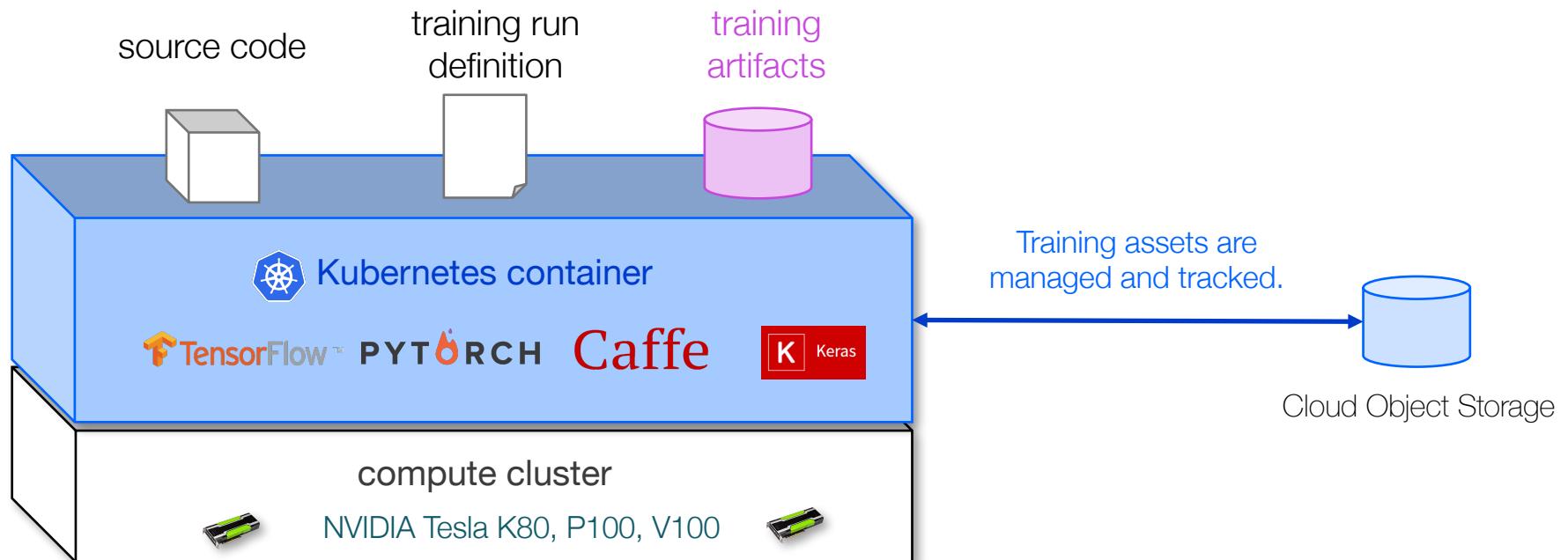
Experiment
Management

Technical
Deep Dive

Model
Hosting

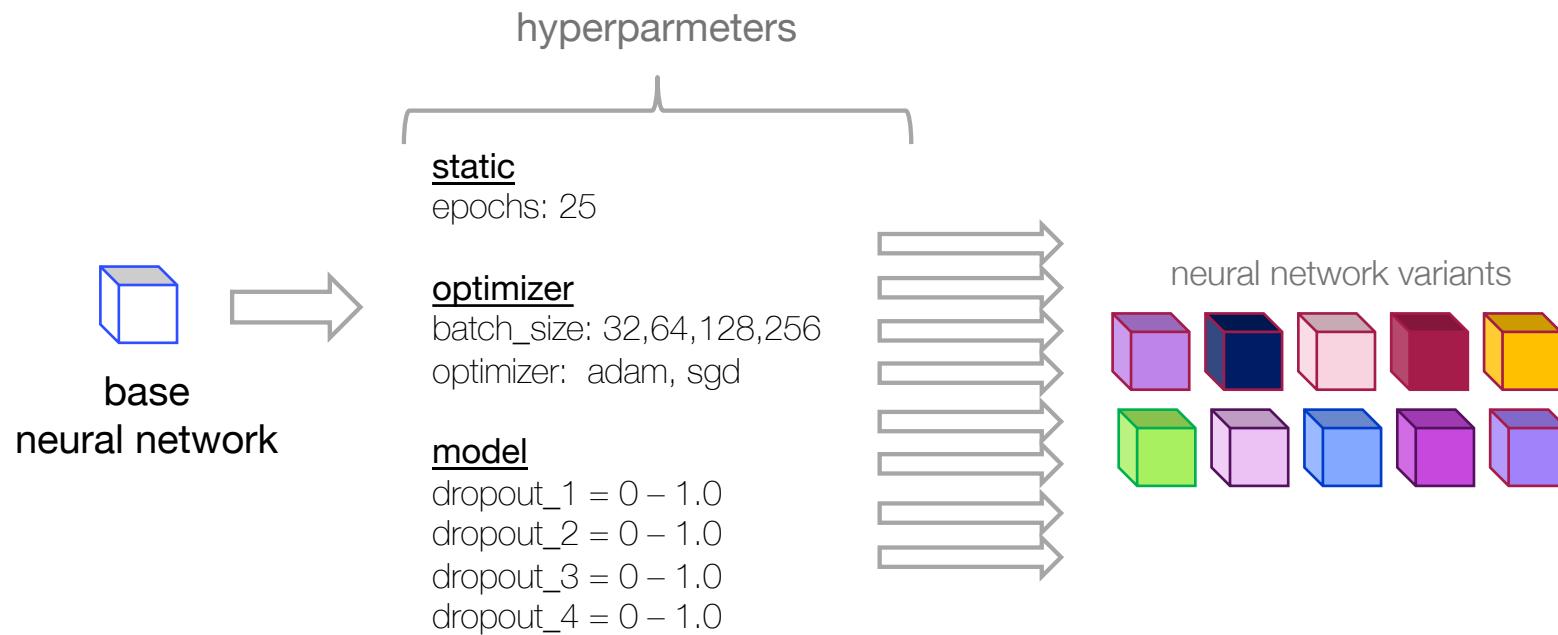
Training Lifecycle Management

Auto-allocation means paying only for what you use



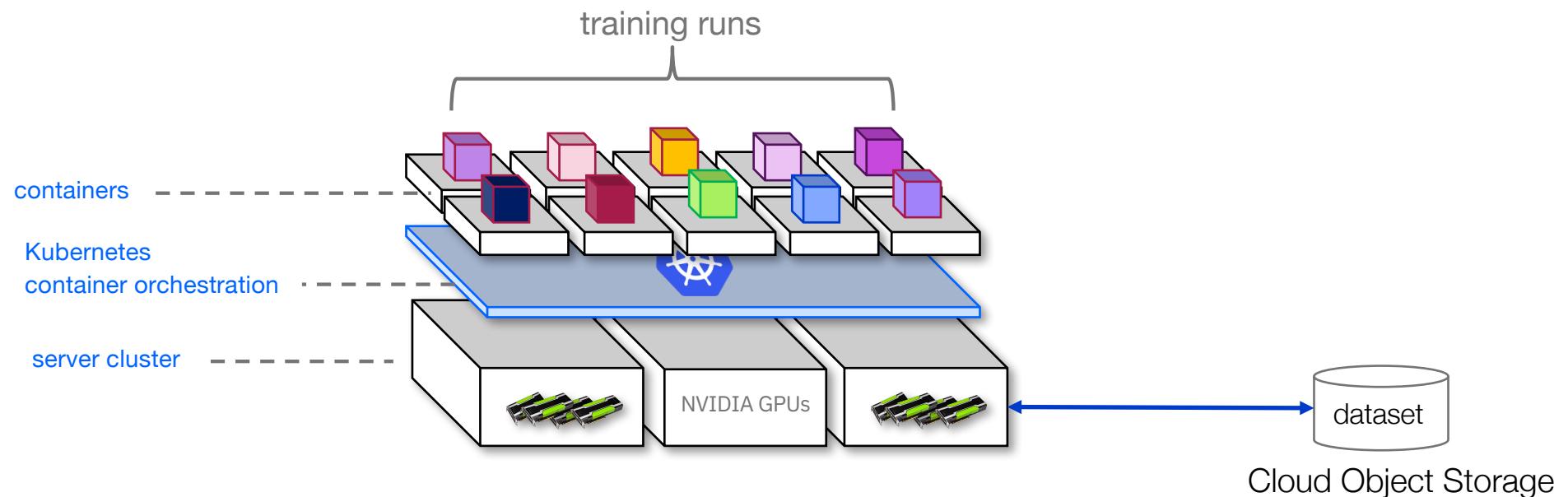
Train 1000's of neural networks

Use Hyperparameter optimization (HPO) or define your own training runs



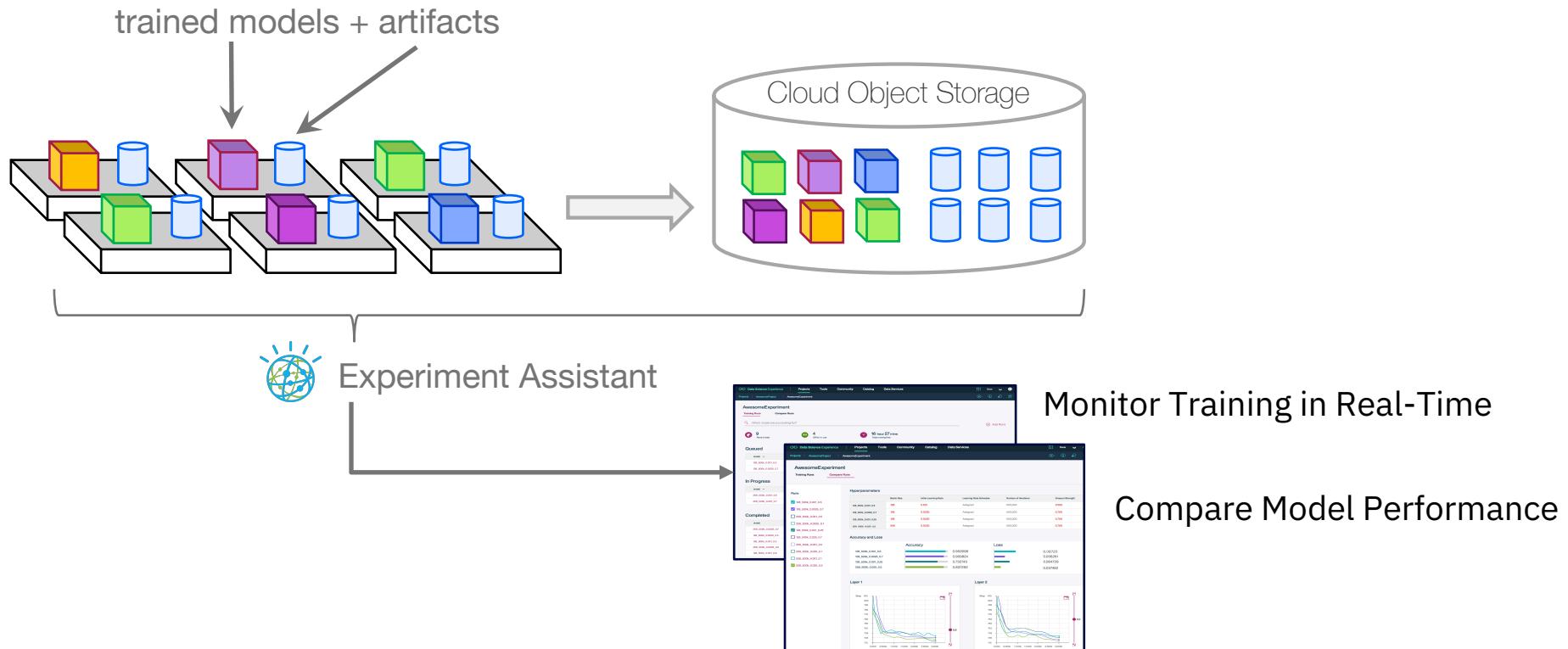
Training distributed across parallel containers

Use Hyperparameter optimization (HPO) or define your own training runs



Experiment assistance manages and archives training progress

Monitor training progress and compare model performance in real-time



Distributed Learning w/Tensorflow

As datasets expand and model grow in complexity, training times increase from hour to days to weeks or longer to complete.

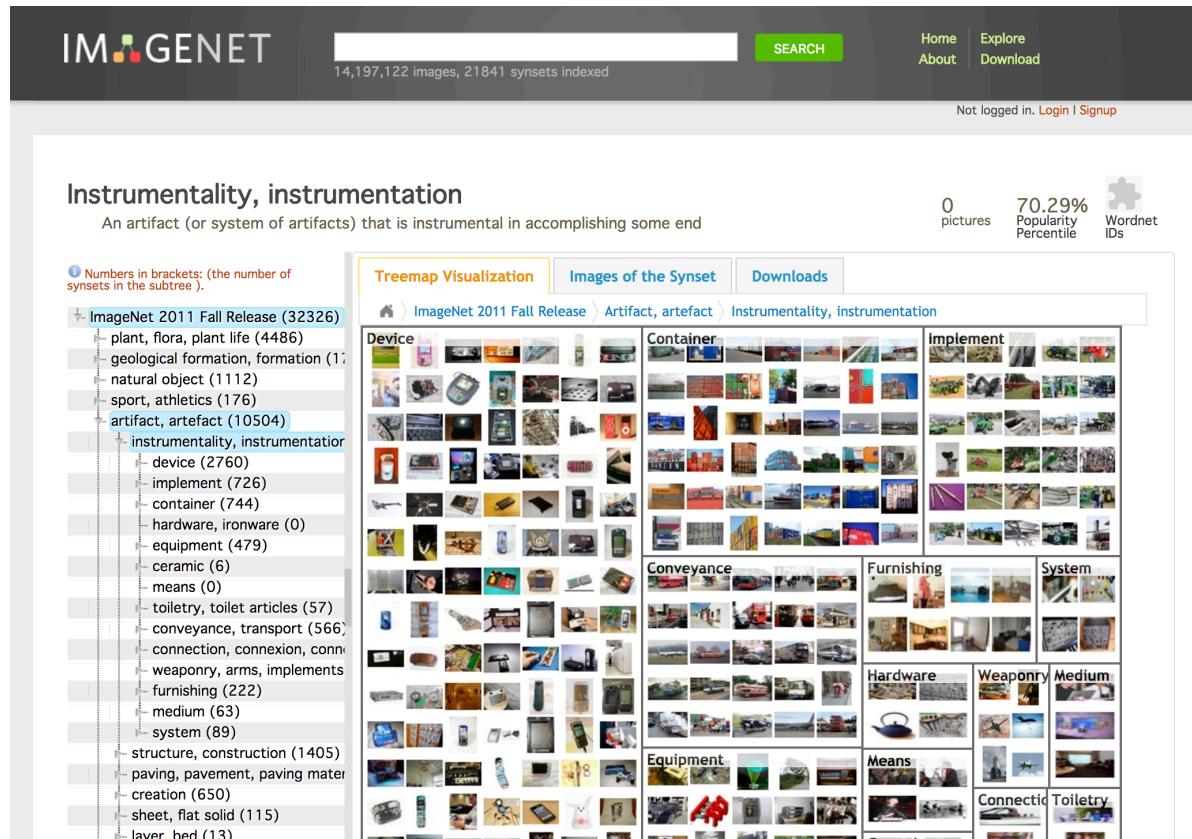
Models no longer fit on one or multiple GPUs within a single server and must be distributed across multiple machines.

Native TensorFlow

IBM Distributed Deep Learning (DDL)

Uber's Horovod

What about massive datasets like ImageNet 22K?

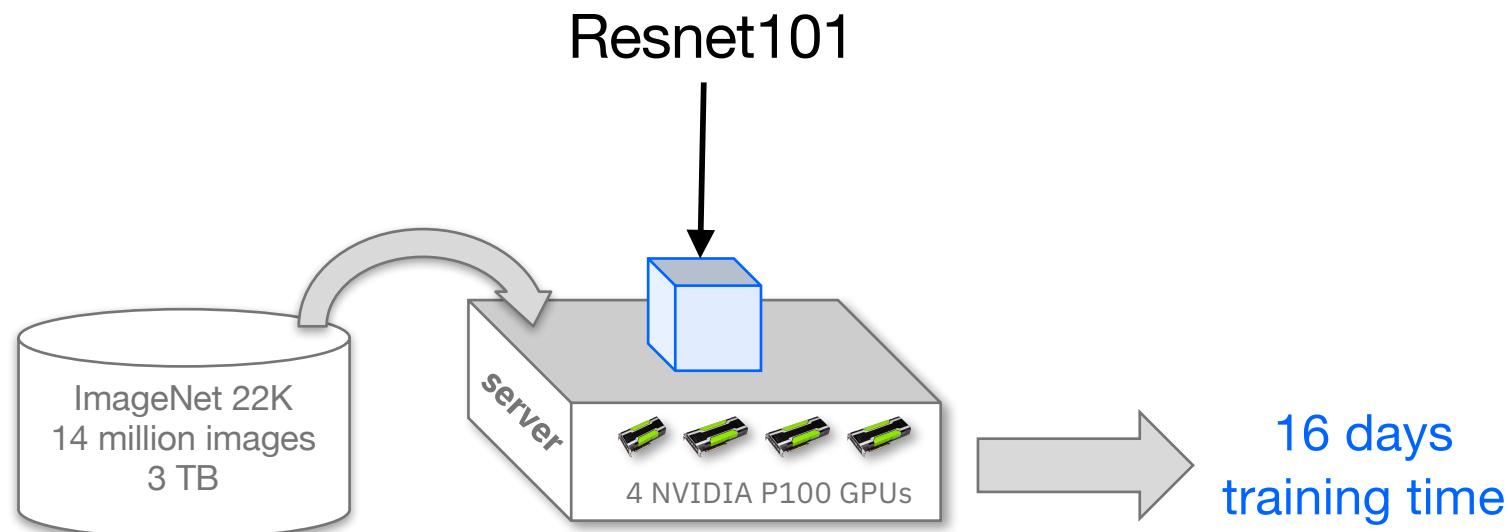


14 million images

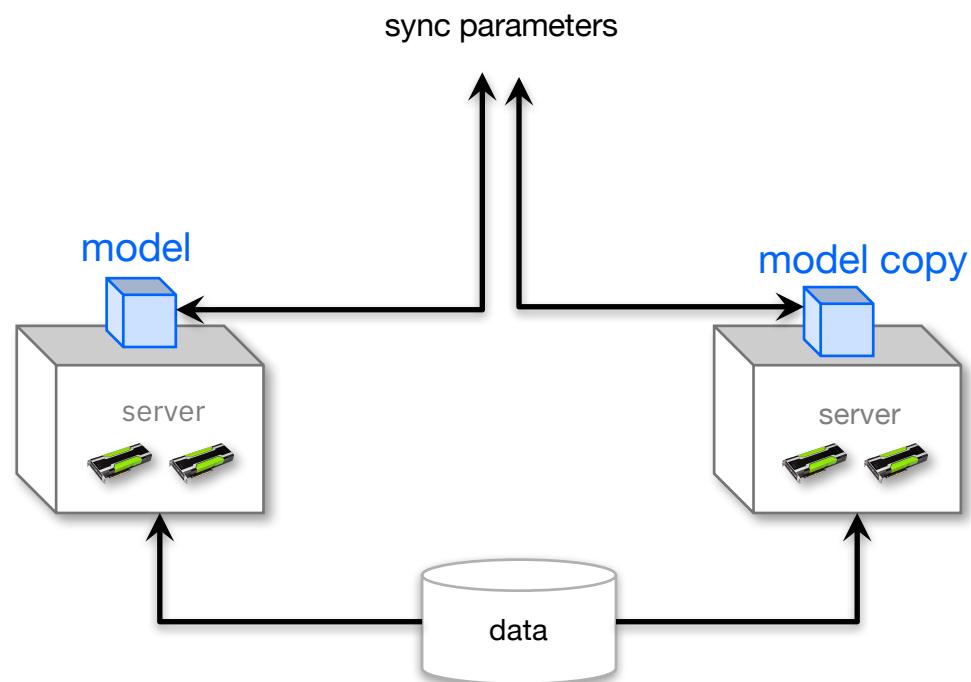
3 Terabytes

21,841 categories

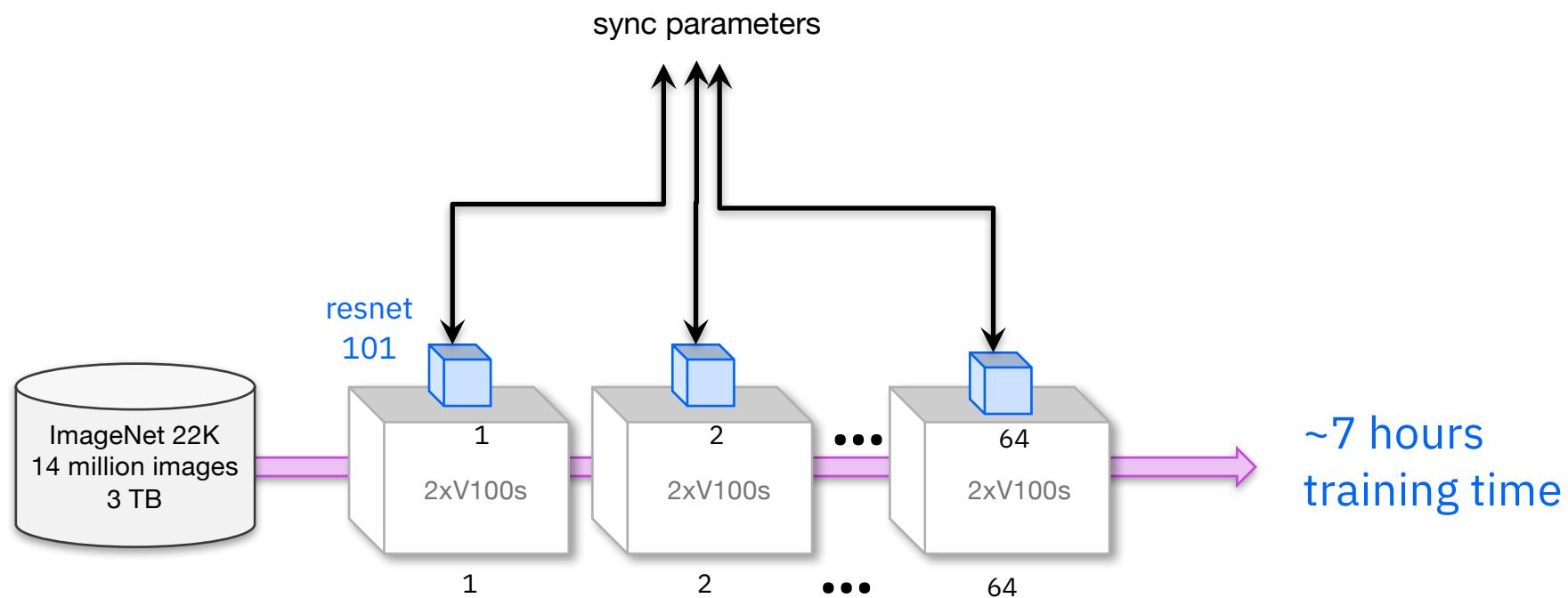
Single servers are too slow for complicated networks and large datasets



Distributed training across multiple servers



Distributed training at scale



Neural Network
Design

Experiment
Management

Technical
Deep Dive

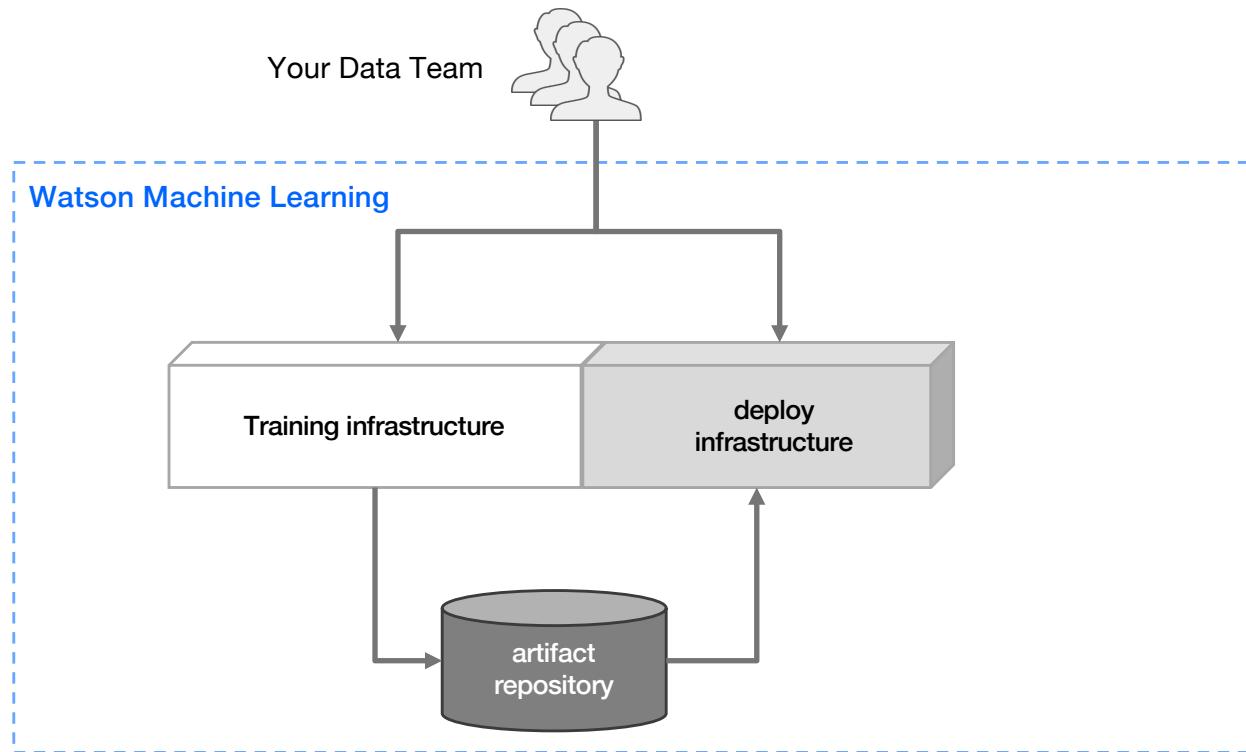
Model
Hosting

Deep Learning

Available through Watson Machine Learning

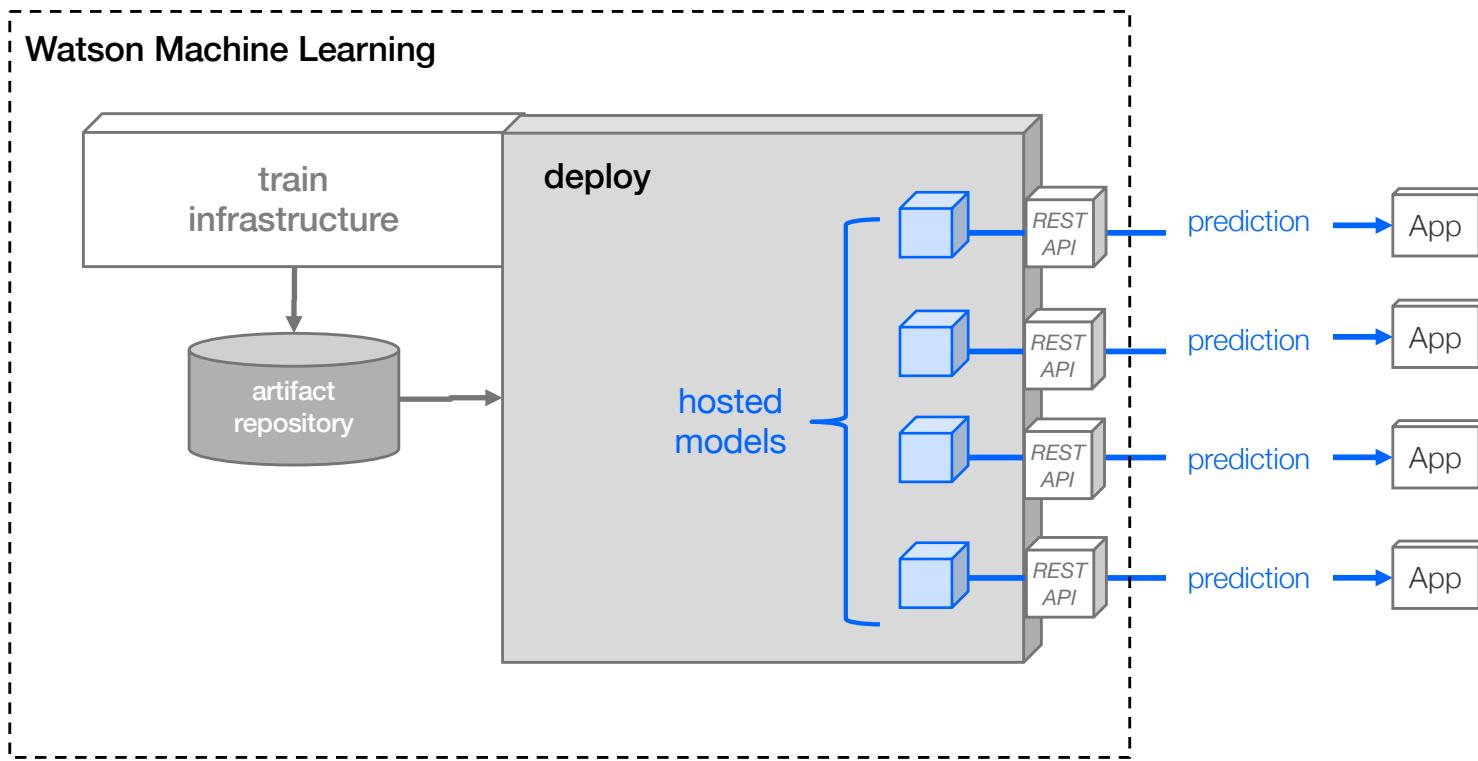
Hosted ML Infrastructure

Watson Machine Learning (WML) is a [managed learning service](#) optimized for production environments that enables train plus deploy of new models.



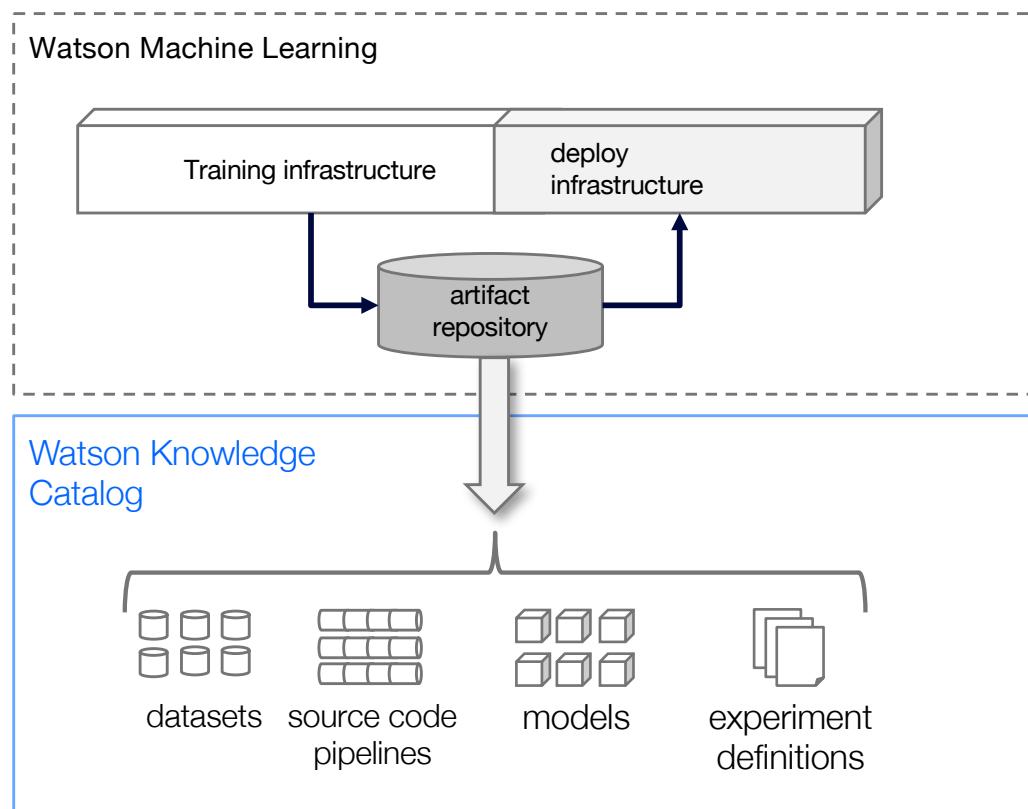
Deploying Trained Models

Download your trained models or deploy your models within Watson Machine Learning



Asset Lifecycle Management

Use the Watson Knowledge Catalog and Watson Studio to manage your machine learning asset or manage them yourself



Model Explanations

In May 2018, the General Data Protection Regulation (GDPR) takes effect and grants consumers the legal “right to explanation” from organizations that use algorithmic decision making.

Audit Trails

Tracking prediction to each model’s unique heritage is critical to regulatory compliance. Enforcing access controls for model sharing and deployment ensure ensures data security and application stability.

Watson Machine Learning

Simplifying deployment & management of ML models in production apps

Embed AI in your Business

Train

Train custom ML and DL models at scale

Deploy

Move models into production where they improve and remain relevant

Automate

Build better models faster while they continue to learn after

- Use your favorite tools and frameworks such as SPSS, Tensorflow, Keras, Scikit Learn or xgboost.
- Track the progress and performance of your training jobs with the Experiment Assistant interface.
- Create new models without writing a single line of code using the neural network modeler

- Portable models – deploy in the cloud, on devices or on premise
- Import models trained somewhere else and deploy in the cloud
- Transfer models to connected devices with support of Core ML, Tensorflow Lite
- Make millions of predictions in seconds

- Build better models faster with automatic parameter tuning
- Automate the retraining of models with our feedback loop capabilities
- Automate the deployment of models in products
- Automatic algorithm selection for ML models with our CADS technology from Research

Watson Machine Learning

Simplifying deployment & management of ML models in production apps

Python client, command line interface (CLI) or UI? You choose the tooling that best fits your existing workflows. Training history and assets are tracked then automatically transferred to the customer's Object Storage for quick access.

```
  data_science -- bash - 158 at 47
```

```
~Documents/Engineering/Development/github/jupyter-notebooks> python
```

```
~Documents/Engineering/Development/data_science -- bash
```

```
○ auth
```

```
287 keras-fusion-cos [batch-45] training-AGd4P3kq completed tensorflow 2018-02-14T19:51:27Z
288 keras-fusion-cos [batch-49] training-8EVfE5kq completed tensorflow 2018-02-14T19:51:29Z
289 keras-fusion-cos [batch-53] training-1DgXW5kq pending tensorflow 2018-02-14T19:51:29Z
290 keras-fusion-cos [batch-64] training-MKv5Fpkq pending tensorflow 2018-02-14T19:51:29Z
291 keras-fusion-cos [batch-59] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:29Z
292 keras-fusion-cos [batch-61] training_EV9tqzq pending tensorflow 2018-02-14T19:51:29Z
293 keras-fusion-cos [batch-62] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:29Z
294 keras-fusion-cos [batch-58] training-1SWBEl3kq pending tensorflow 2018-02-14T19:51:38Z
295 keras-fusion-cos [batch-56] training-xyHfFtzq pending tensorflow 2018-02-14T19:51:38Z
296 keras-fusion-cos [batch-55] training-1DgXW5kq pending tensorflow 2018-02-14T19:51:38Z
297 keras-fusion-cos [batch-53] training-4ufU1tkq pending tensorflow 2018-02-14T19:51:38Z
298 keras-fusion-cos [batch-50] training-1DgXW5kq pending tensorflow 2018-02-14T19:51:38Z
299 keras-fusion-cos [batch-51] training-9r4g8pkq pending tensorflow 2018-02-14T19:51:38Z
300 keras-fusion-cos [batch-63] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:38Z
301 keras-fusion-cos [batch-65] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:38Z
302 keras-fusion-cos [batch-67] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:38Z
303 keras-fusion-cos [batch-54] training-HNSyPtkq completed tensorflow 2018-02-14T19:51:37Z
304 keras-fusion-cos [batch-66] training-8L1ttPtkq pending tensorflow 2018-02-14T19:51:39Z
305 keras-fusion-cos [batch-70] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:42Z
306 keras-fusion-cos [batch-76] training-3Mq5Ptkq pending tensorflow 2018-02-14T19:51:42Z
307 keras-fusion-cos [batch-72] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:42Z
308 keras-fusion-cos [batch-74] training-AWNS0tkq pending tensorflow 2018-02-14T19:51:46Z
309 keras-fusion-cos [batch-78] training-1JLwQpkq pending tensorflow 2018-02-14T19:51:46Z
310 keras-fusion-cos [batch-80] training-8Dp2pkq pending tensorflow 2018-02-14T19:52:05Z
311 keras-fusion-cos [batch-79] training-1JLwQpkq pending tensorflow 2018-02-14T19:52:32Z
312 keras-fusion-cos [batch-78] training-1JLwQpkq pending tensorflow 2018-02-14T19:52:32Z
313 keras-fusion-cos [batch-63] training-1Ym4tewq pending tensorflow 2018-02-14T19:52:35Z
314 keras-fusion-cos [batch-65] training-1JLwQpkq pending tensorflow 2018-02-14T19:52:35Z
315 keras-fusion-cos [batch-81] training-jAsVpg3kq pending tensorflow 2018-02-14T19:52:51Z
```

```
23 records found, 0s
```

```
List all training-runs successful  
blobstore: https://blobstore.s3.amazonaws.com/list-training-runs  
Fetching the list of training runs ...
```

No	Name	gold	status	framework	submitted-at
1	keras-fusion-cos [batch-7]	Training-KoNPop3zq	pending	tensorflow	2018-02-14T19:07:29Z
2	keras-fusion-cos [batch-6]	Training-7bfDp2zq	pending	tensorflow	2018-02-14T19:07:29Z
3	keras-fusion-cos [batch-5]	Training-1DgXW5kq	pending	tensorflow	2018-02-14T19:07:31Z
4	keras-fusion-cos [batch-3]	Training-AqHs0pkq	completed	tensorflow	2018-02-14T19:07:31Z
5	keras-fusion-cos [batch-4]	Training-1JLwQpkq	completed	tensorflow	2018-02-14T19:07:31Z
6	keras-fusion-cos [batch-1]	Training-1W7e3kq	completed	tensorflow	2018-02-14T19:07:31Z
7	keras-fusion-cos [batch-8]	Training-70083kq	completed	tensorflow	2018-02-14T19:07:32Z
8	keras-fusion-cos [batch-11]	Training-70083kq	completed	tensorflow	2018-02-14T19:07:35Z
9	keras-fusion-cos [batch-10]	Training-1DgXW5kq	pending	tensorflow	2018-02-14T19:07:35Z
10	keras-fusion-cos [batch-12]	Training-1OEl-13kq	completed	tensorflow	2018-02-14T19:07:35Z

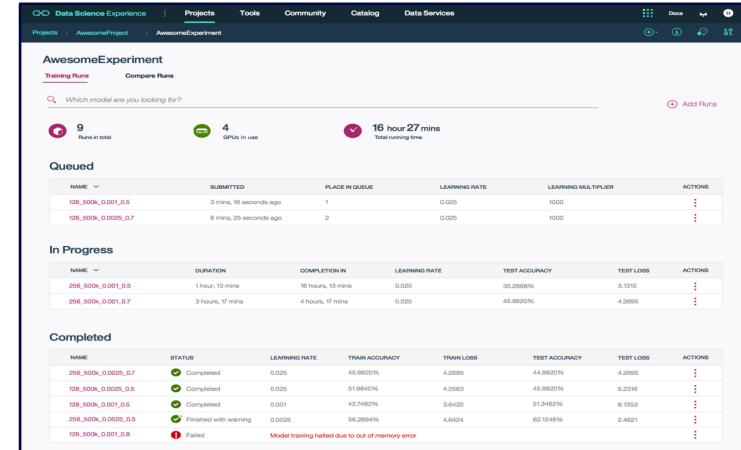
Monitor batch training experiments then compare cross-model performance without worrying about log transfers and scripts to visualize results. You focus on designing your neural networks. We'll manage and track your assets.

The screenshot shows the Data Science Experience interface with the following details:

- Header:** Data Science Experience, Projects, Tools, Community, Catalog, Data Services, Docs.
- Project:** AwesomeProject, AwesomeExperiment.
- Section:** AwesomeExperiment, Training Run, Compare Runs.
- Runs:** A table listing training runs with columns: Name, Status, Batch Size, Initial Learning Rate, Learning Rate Schedule, Number of Iterations, Dropout Strength. The runs include:
 - 128_500k_0.001_0.5 (Status: Success, Batch Size: 128, Learning Rate: 0.001, Iterations: 500,000, Dropout: 0.500)
 - 128_500k_0.005_0.7 (Status: Success, Batch Size: 128, Learning Rate: 0.005, Iterations: 500,000, Dropout: 0.700)
 - 256_500k_0.001_0.5 (Status: Success, Batch Size: 256, Learning Rate: 0.001, Iterations: 500,000, Dropout: 0.500)
 - 256_500k_0.005_0.7 (Status: Success, Batch Size: 256, Learning Rate: 0.005, Iterations: 500,000, Dropout: 0.700)
 - 32k_500k_0.001_0.25 (Status: Success, Batch Size: 32, Learning Rate: 0.001, Iterations: 500,000, Dropout: 0.250)
 - 32k_500k_0.005_0.7 (Status: Success, Batch Size: 32, Learning Rate: 0.005, Iterations: 500,000, Dropout: 0.700)
- Hyperparameters:** A table showing hyperparameter values for each run.
- Accuracy and Loss:** A section comparing Accuracy and Loss across runs.
 - Accuracy:** Success (0.952698), Success (0.905984), Success (0.732743), Success (0.837292).
 - Loss:** Success (0.08725), Success (0.056261), Success (0.064729), Success (0.037482).
- Layer 1:** A line chart showing accuracy over steps (0 to 800) for four runs. The legend indicates:
 - 128_500k_0.001_0.5 (blue)
 - 128_500k_0.005_0.7 (orange)
 - 256_500k_0.001_0.5 (green)
 - 256_500k_0.005_0.7 (red)A vertical red dashed line marks Step 800, and a pink dot highlights the accuracy at Step 800 for the 256_500k_0.005_0.7 run (approx. 0.837).
- Layer 2:** A line chart showing loss over steps (0 to 800) for the same four runs. A vertical red dashed line marks Step 800, and a pink dot highlights the loss at Step 800 for the 256_500k_0.005_0.7 run (approx. 0.037).

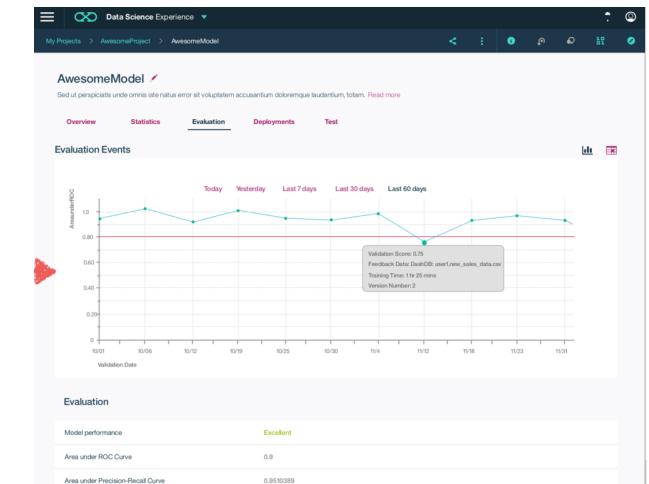
Train neural networks in parallel across NVIDIA GPUs.

Pay only for what you use. Auto-deallocation means no more remembering to shutdown your cloud training instances.



Deploy models into production then monitor them to evaluate performance.

Capture new data for continuous learning and retrain models so they continually adapt to changing conditions.



IBM Watson Studio is for **AI**

Notices and disclaimers

© 2018 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights – use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation. It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.** The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Thank you

Anthony Stevens
Offering Manager, Watson Deep Learning

—
apsteven@us.ibm.com

