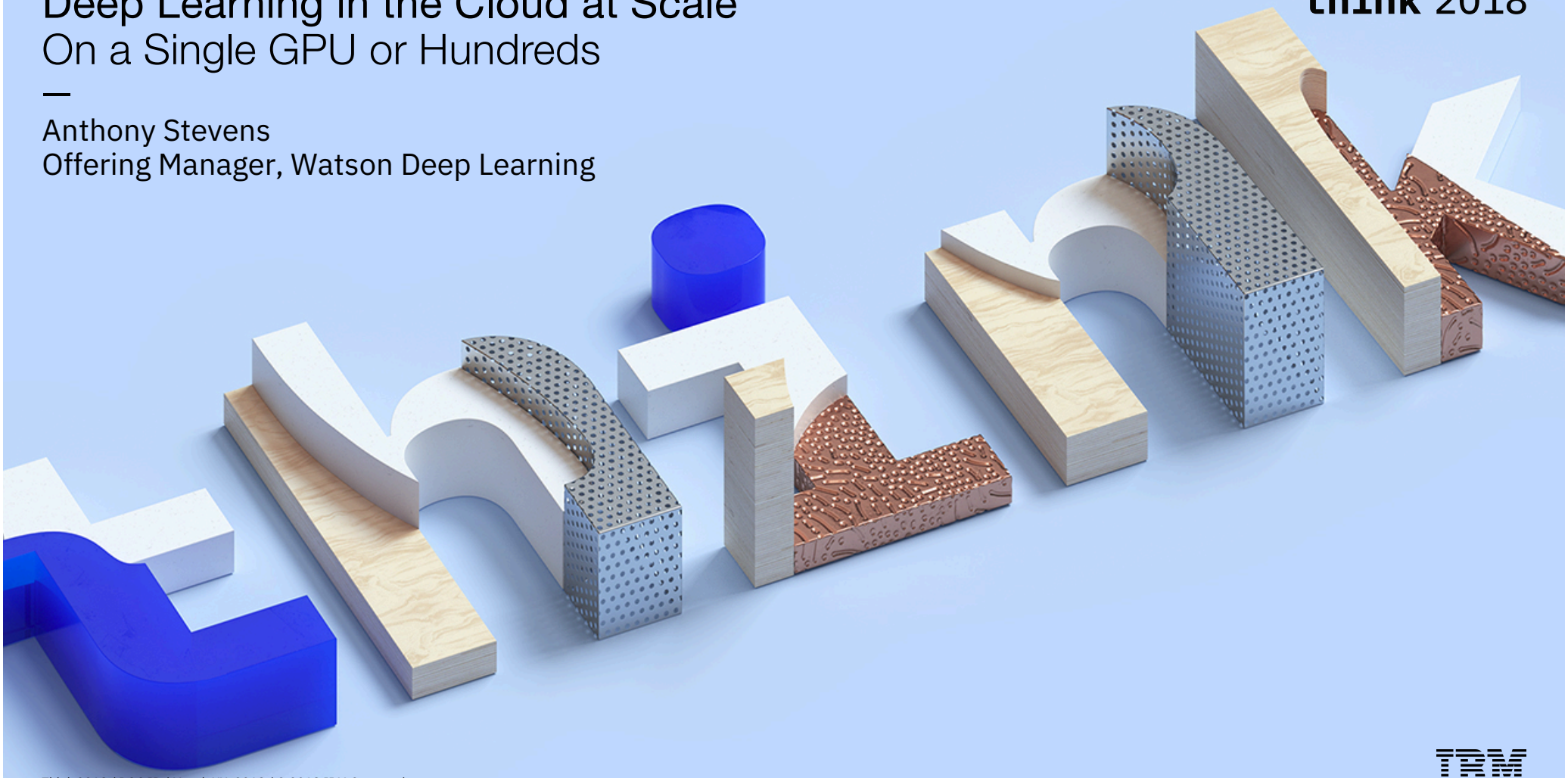
Deep Learning in the Cloud at Scale
On a Single GPU or Hundreds

—

Anthony Stevens
Offering Manager, Watson Deep Learning

think 2018



IBM

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.

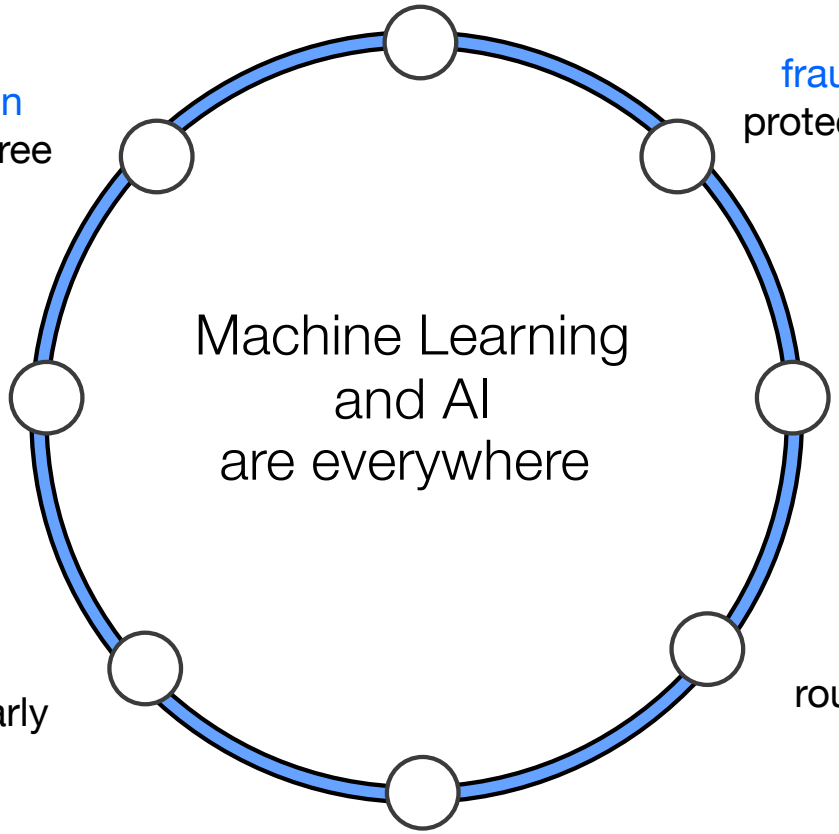
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

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

The future is now



facial recognition
unlocks your phone

fraud detection
protects your credit

recommendations
help you shop faster

chat bots
route calls quicker

spam detection
unclogs your Inbox

machine vision
detects cancer early

autonomous vehicles
detect pedestrians

speech recognition
lets you go hands-free

Use Cases for **Deep Learning**

Cyber Defense

Drug Discovery

Fraud Detection

Aeronautics

IoT

Medical Decision-Making

Smart Cities

Earth Monitoring

Weather Forecasting

Disease Diagnostics

Climate Change

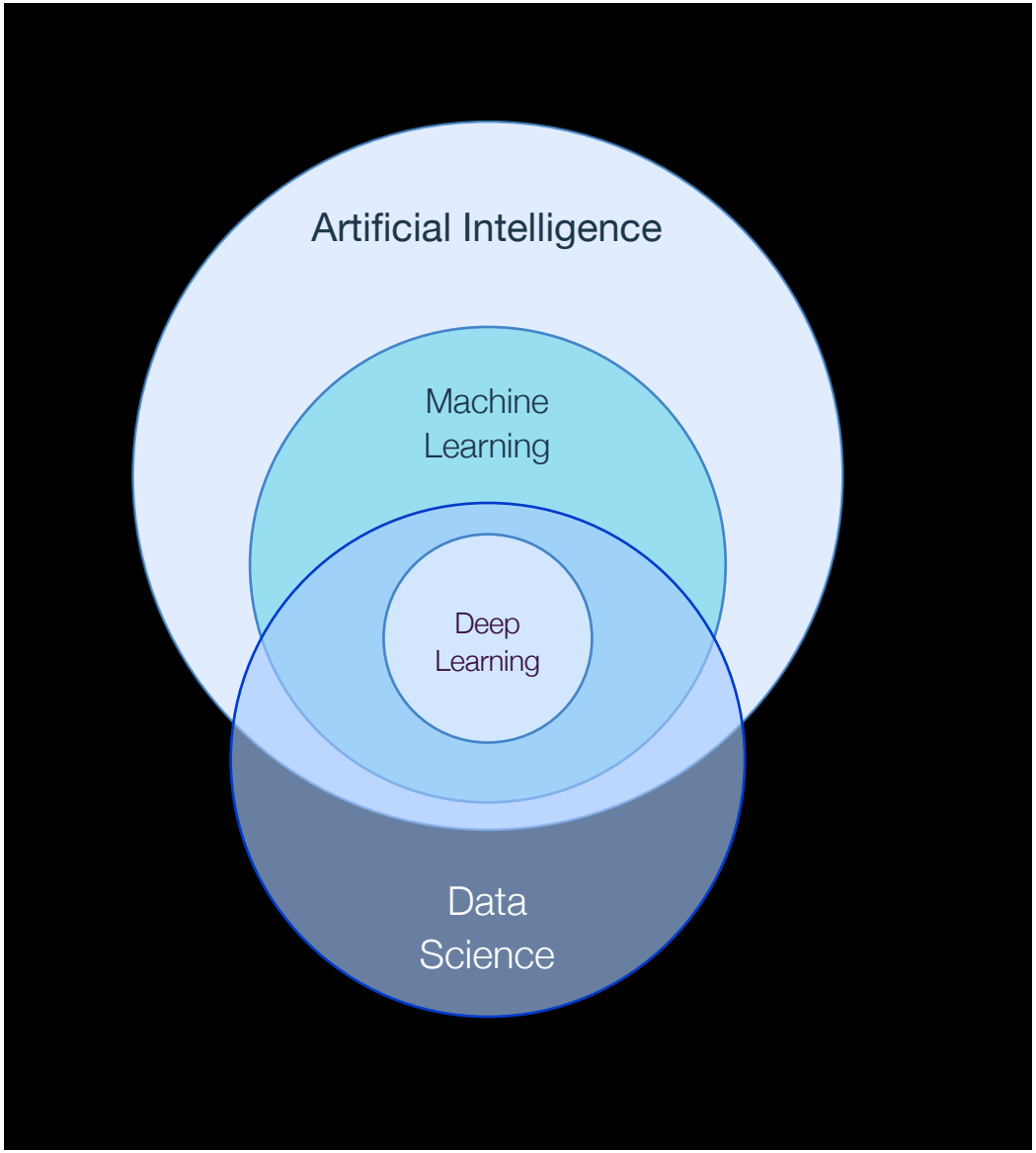
Advanced Physics Research

Robotics

Supply Chain Management

Media Analytics

How does machine learning relate to data science and AI?



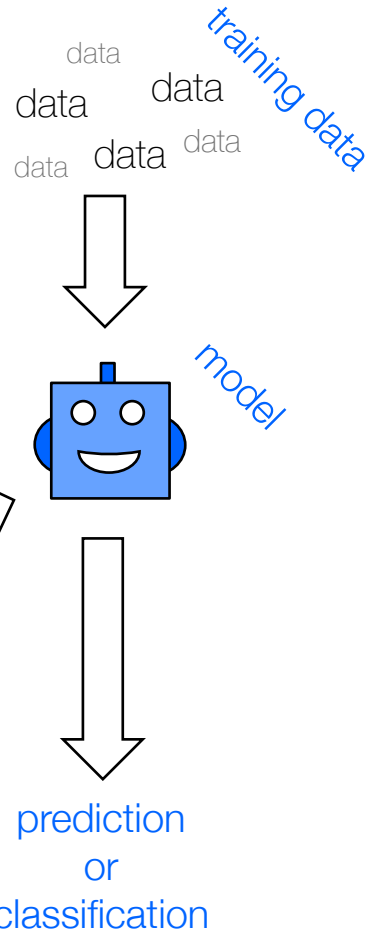
How does machine learning work?

Machine learning requires TONS OF DATA

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

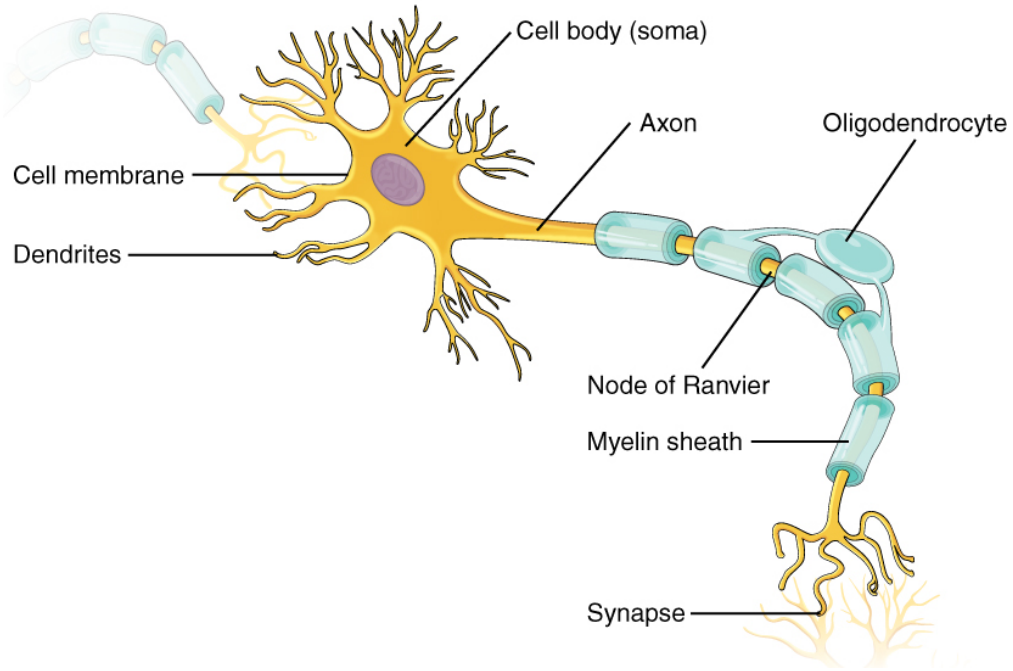
2. The model is then shown new data and asked to predict or classify it.

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

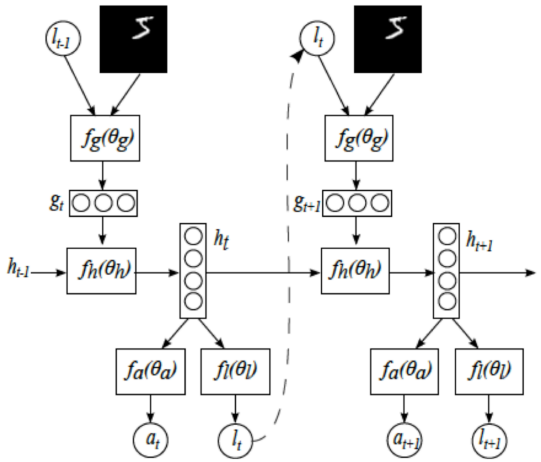
Machine Learning is algorithm selection

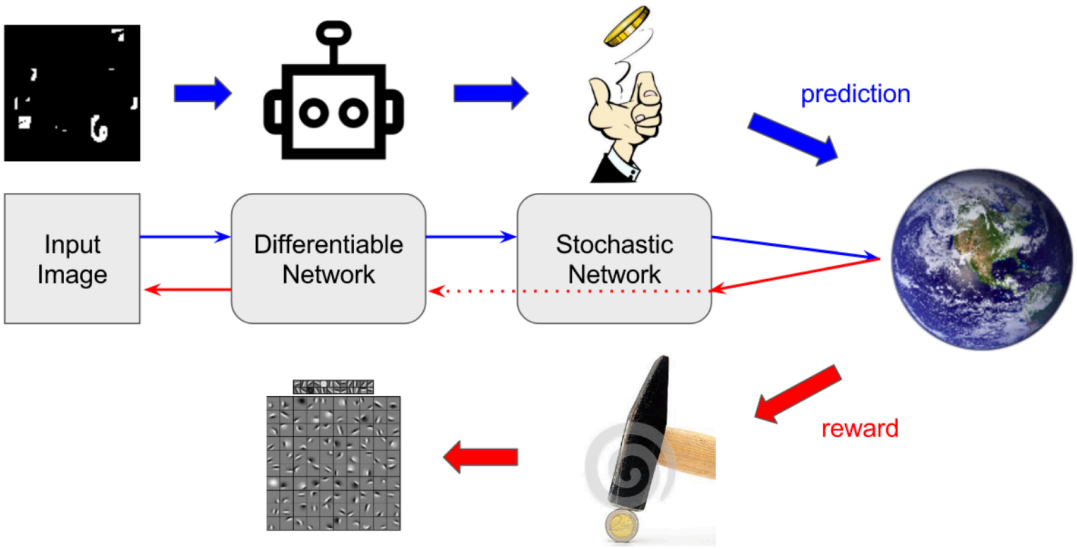
Deep learning is neural network design

AI is systems architecture

Augmented MNIST[1]

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

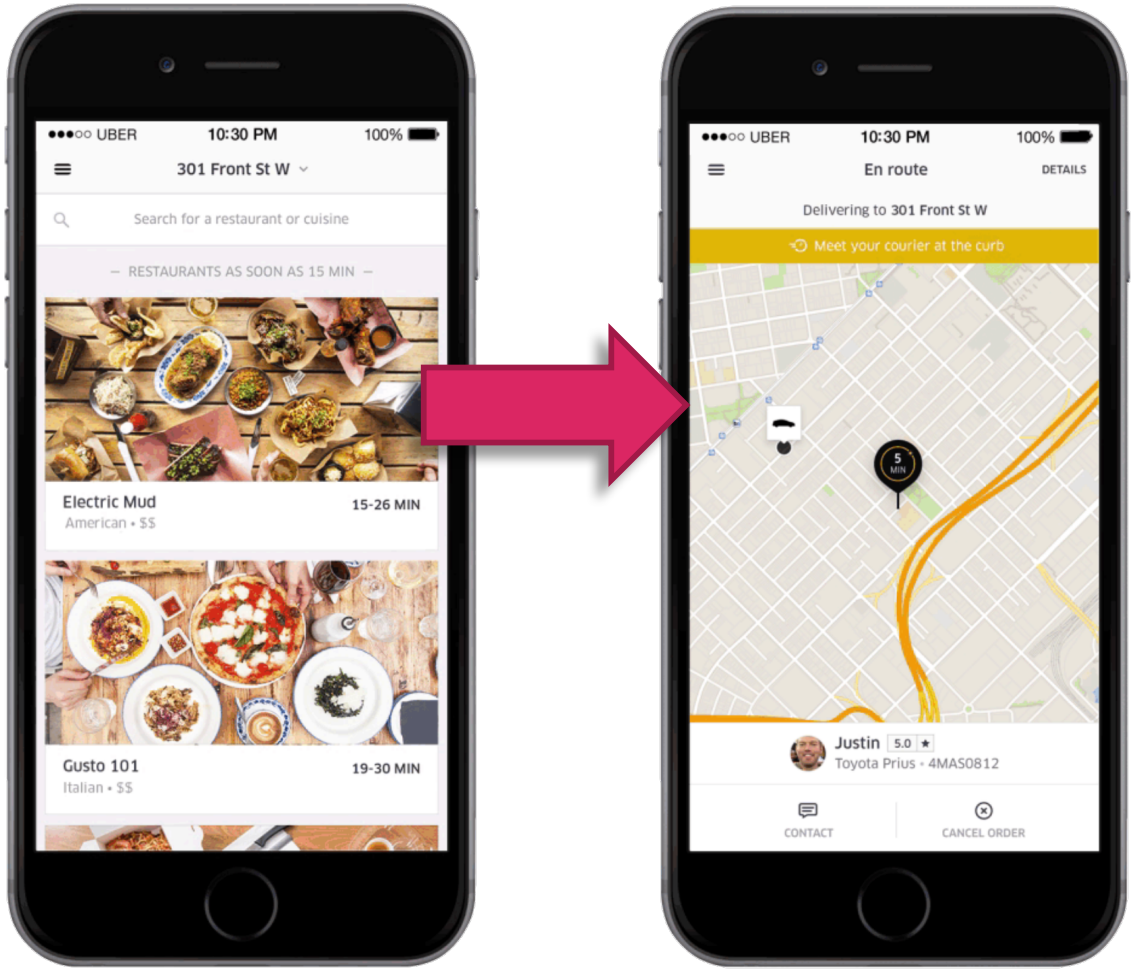




Source 1. https://towardsdatascience.com/visual-attention-model-in-deep-learning-708813c2912c

Uber Eats requires multiple models to ensure accuracy of food delivery

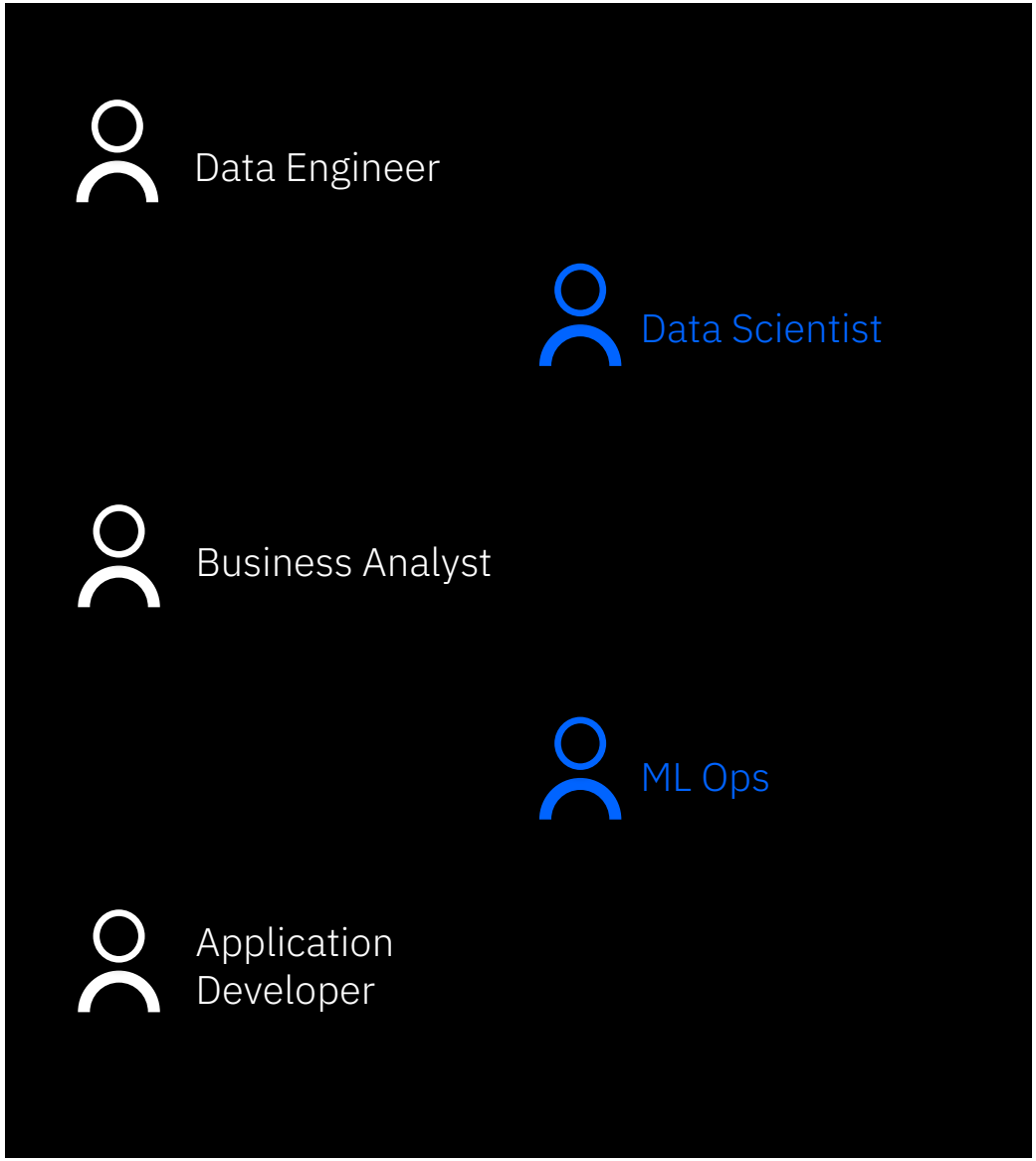
- ✓ search autocomplete
- ✓ search rankings
- ✓ restaurant rankings
- ✓ meal delivery time predictions
- ✓ nearby driver selection
- ✓ driver incentives
- ✓ customer satisfaction analysis



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

Machine learning requires a data science team.

And teams require tools to collaborate.



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

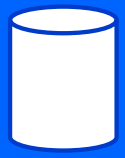
pre-trained neural network

Application Developer

transfer learning

SME

+

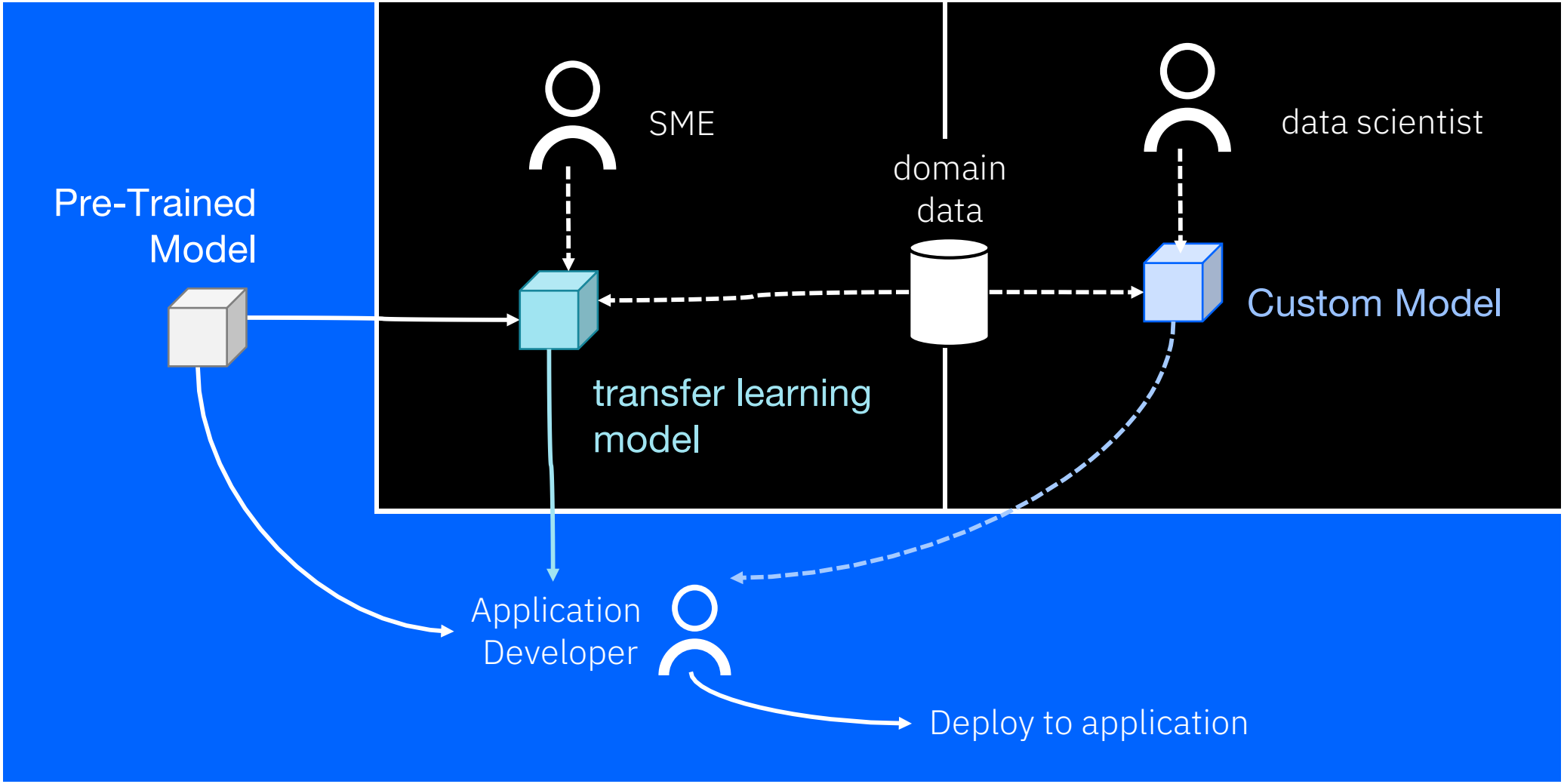


domain data

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

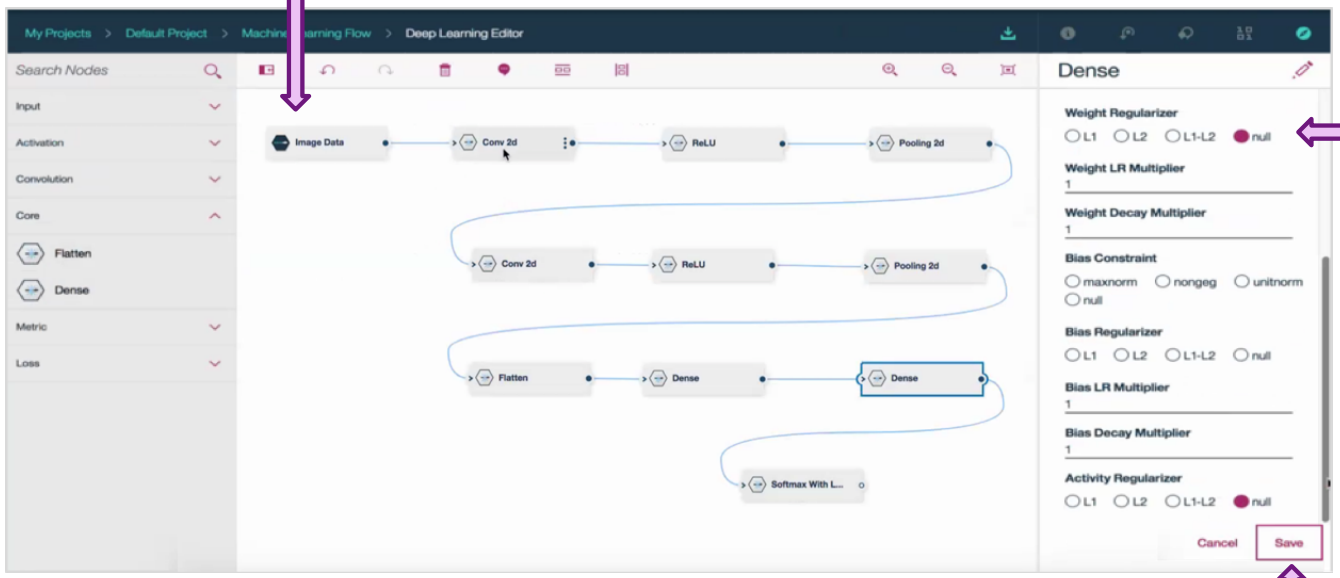
Neural Network Modeler (beta)

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

TensorFlow™ PYTORCH Caffe Keras

Real-time validation of network flow



Drag-and-drop network layers

- Define layer configuration
- Choose optimizer params

- Generate CPU or GPU compatible code

- Save as popular framework code
- Export as a python notebook
- Execute as batch experiment

Neural Network Design

Experiment Management

Technical Deep Dive

Model Hosting

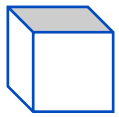
Discovering a single optimal neural network requires exploring 1000's of hyperparameter combinations

which means 100's of experiments

which requires 1000's of training runs

Accelerate your experimental workflow

You focus on experiment design

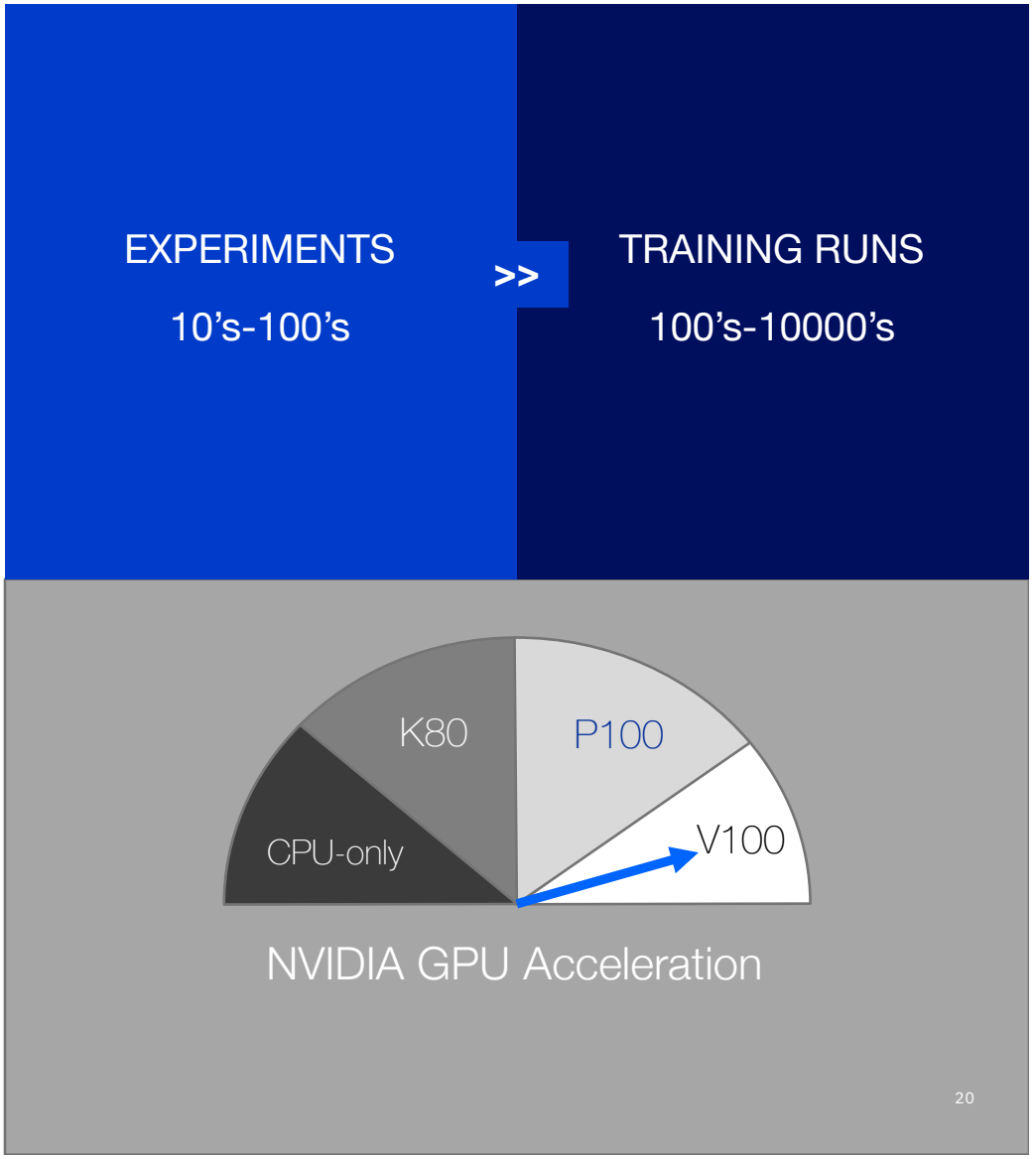


source code

experiment definition

dataset

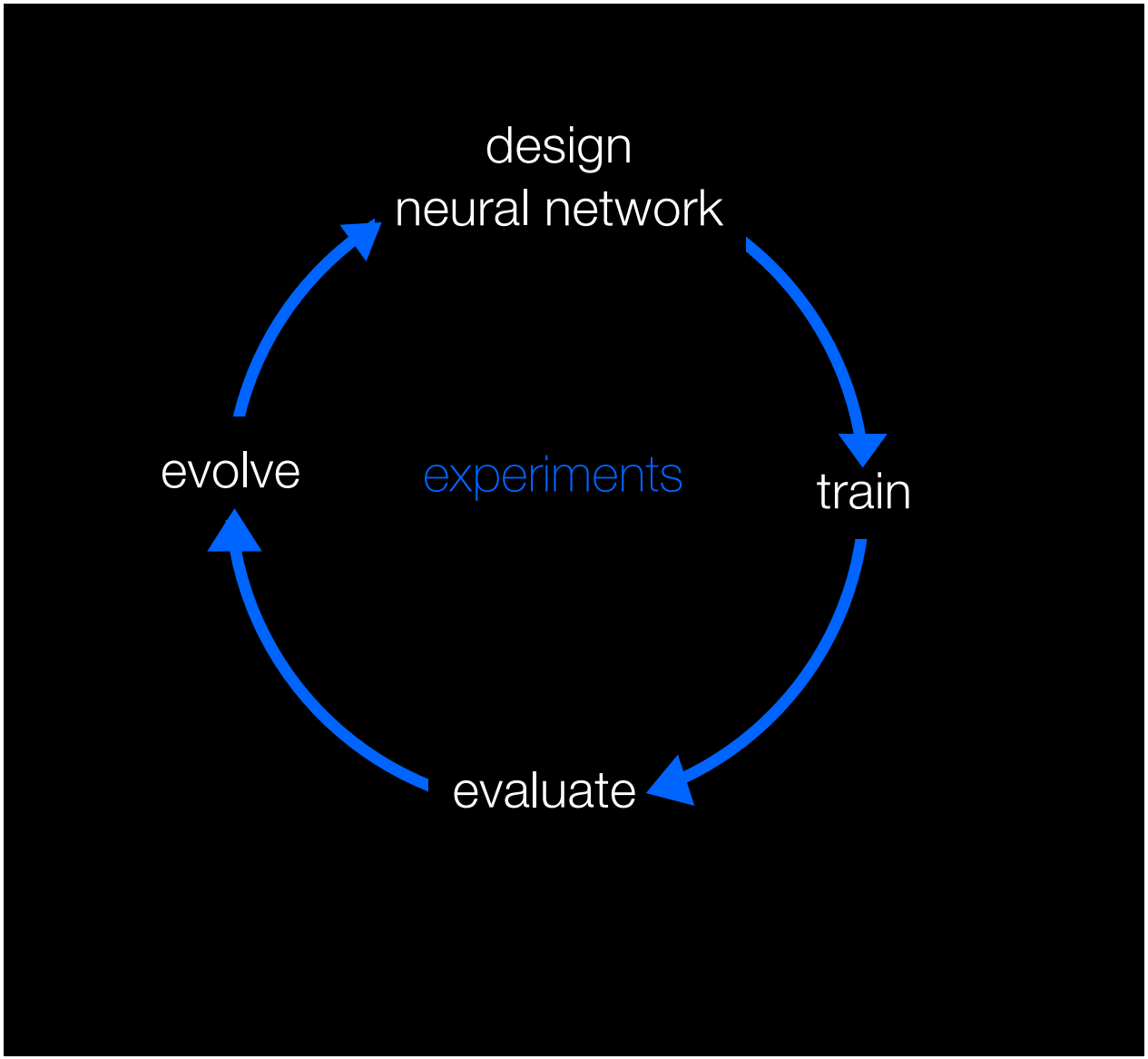
We'll manage the rest

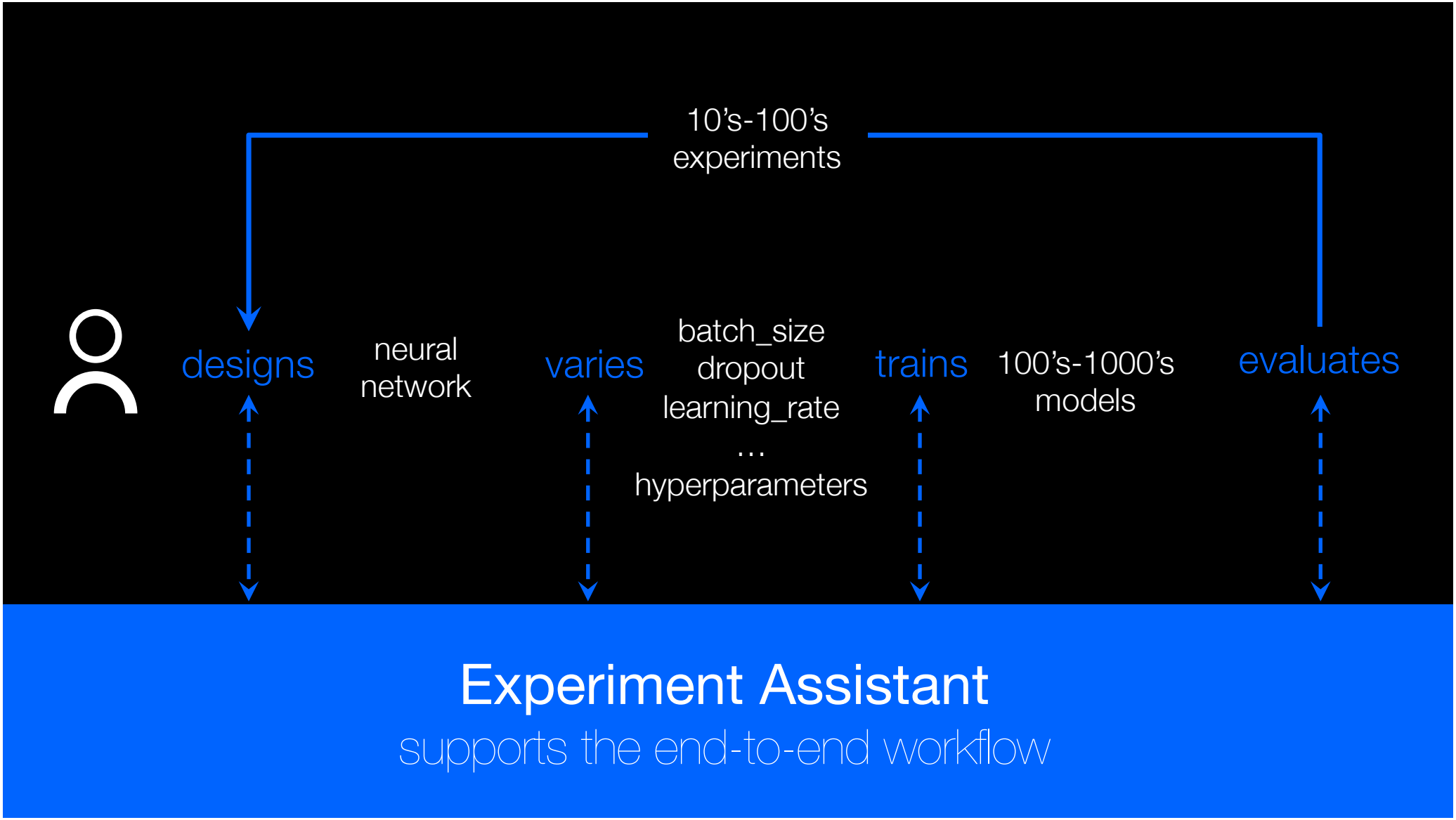


Experiment Assistant
accelerates discovering the optimal neural networks

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

It's here to serve you

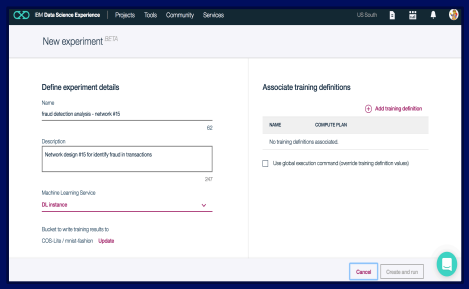



10's-100's
experiments
designs
neural
network
varies
batch_size
dropout
learning_rate
…
hyperparameters
trains
100's-1000's
models
evaluates
Experiment Assistant
supports the end-to-end workflow

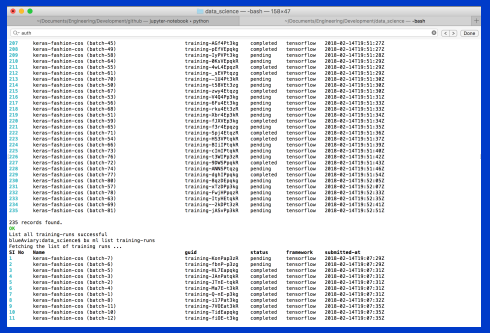
Experiment Assistant

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.

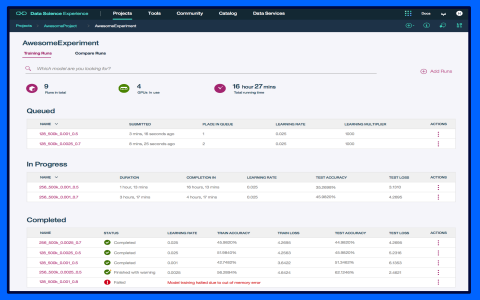
Initiate Experiments



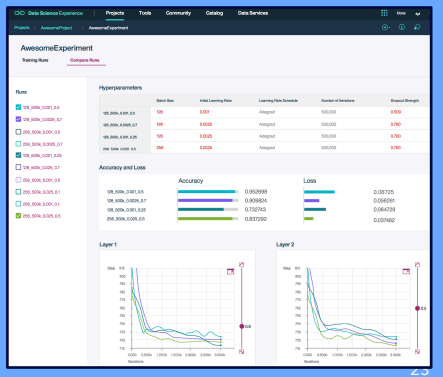
CLI or Python Client



Monitor Training in Real-Time



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-u
byte.gz --trainLabelsFile ${DATA_DIR}/train-labels-idx1-ubyte.gz --testIm
agesFile ${DATA_DIR}/t10k-images-idx3-ubyte.gz --testLabelsFile ${DATA_DI
R}/t10k-labels-idx1-ubyte.gz --%%batch_size%% --%%optimizer%% --%%epochs%
% --%%dropout%%
```

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 + "

        # 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: "
                client.repository.DefinitionMetaNames.FRAMEWORK_VERSION
```

Command Line

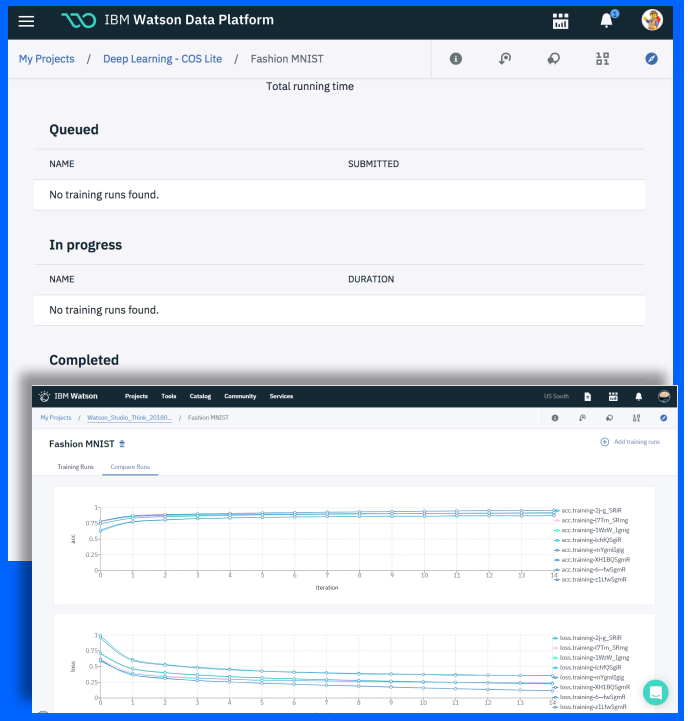
biosopher — -bash — 117×57

...ent/github — jupyter-notebook ▸ python ... | ...L/street house numbers/data — -bash | ~ — -bash

```
525     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.15_d__5=0.05   training-GnlWwt6zg   running
 tensorflow   2018-02-27T06:58:02Z
526     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.15_d__5=0.15   training-SAXZQp6kR   running
 tensorflow   2018-02-27T06:58:03Z
527     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.15_d__5=0.25   training-R96ZQpezR   pending
 tensorflow   2018-02-27T06:58:07Z
528     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.25_d__5=0.05   training-YNkMQpezg   pending
 tensorflow   2018-02-27T06:58:08Z
529     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.25_d__5=0.15   training-PamMwt6zR   running
 tensorflow   2018-02-27T06:58:11Z
530     bs=256_opt=adam_epochs=50_d__1=0.25_d__2=0.05_d__3=0.15_d__4=0.25_d__5=0.25   training-TQWMwt6kR   running
 tensorflow   2018-02-27T06:58:12Z

530 records found.
OK
List all training-runs successful
blueAviary:~ biosopher$ bx ml list training-runs
Fetching the list of training runs ...
SI No   Name                                                                    guid                 status
  framework    submitted-at
1       bs=256_drop=0_epochs=5_opt=adam                                         training-eSENaZekg   completed
 tensorflow   2018-02-23T08:03:27Z
2       bs=256_drop=0.05_epochs=5_opt=adam                                      training-S88N-Wekg   completed
 tensorflow   2018-02-23T08:03:30Z
3       bs=256_drop=0.1_epochs=5_opt=adam                                       training-_7uHaZezg   completed
 tensorflow   2018-02-23T08:03:32Z
4       bs=256_drop=0.15_epochs=5_opt=adam                                      training-ygCHaWezR   completed
 tensorflow   2018-02-23T08:03:35Z
5       bs=256_drop=0.20_epochs=5_opt=adam                                      training-nWgvaZ6kg   completed
 tensorflow   2018-02-23T08:03:36Z
6       bs=256_drop=0.25_epochs=5_opt=adam                                      training-IKzD-Z6zR   completed
 tensorflow   2018-02-23T08:03:38Z
7       bs=256_drop=0_epochs=5_opt=adam                                         training-zApvnpezg   completed
 tensorflow   2018-02-27T01:11:35Z
8       bs=256_drop=0.05_epochs=5_opt=adam                                      training-b_ovnt6kg   completed
 tensorflow   2018-02-27T01:11:37Z
9       bs=256_drop=0.1_epochs=5_opt=adam                                       training-zsJDnt6zg   completed
 tensorflow   2018-02-27T01:11:39Z
10      bs=256_drop=0.15_epochs=5_opt=adam                                      training-tR-DnpezR   completed
 tensorflow   2018-02-27T01:11:40Z
11      bs=256_drop=0.20_epochs=5_opt=adam                                      training-XrBDnp6kR   completed
 tensorflow   2018-02-27T01:11:42Z
12      bs=256_drop=0.25_epochs=5_opt=adam                                      training-ioPvntekg   completed
 tensorflow   2018-02-27T01:11:44Z
13      bs=256_drop=0_epochs=5_opt=adam                                         training-HsuT4t6kR   completed
 tensorflow   2018-02-27T01:32:35Z
14      bs=256_drop=0.05_epochs=5_opt=adam                                      training-G8qo4pezR   completed
 tensorflow   2018-02-27T01:32:38Z
15      bs=256_drop=0.1_epochs=5_opt=adam                                       training-1NkAVtekg   completed
 tensorflow   2018-02-27T01:32:41Z
16      bs=256_drop=0.15_epochs=5_opt=adam                                      training-8SG04tekR   completed
 tensorflow   2018-02-27T01:32:43Z
17      bs=256_drop=0.20_epochs=5_opt=adam                                      training-wZV04tezg   completed
 tensorflow   2018-02-27T01:32:45Z
18      bs=256_drop=0.25_epochs=5_opt=adam                                      training-W7HA4t6zg   completed
 tensorflow   2018-02-27T01:32:47Z
19      bs=256_drop=0_epochs=5_opt=adam                                         training-vMT04pekg   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

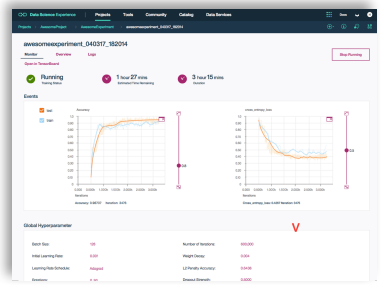
Use your personal IDE or a Jupyter notebook hosted in DSX while using the most popular deep learning frameworks

TensorFlow™ Caffe PYTORCH K Keras

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

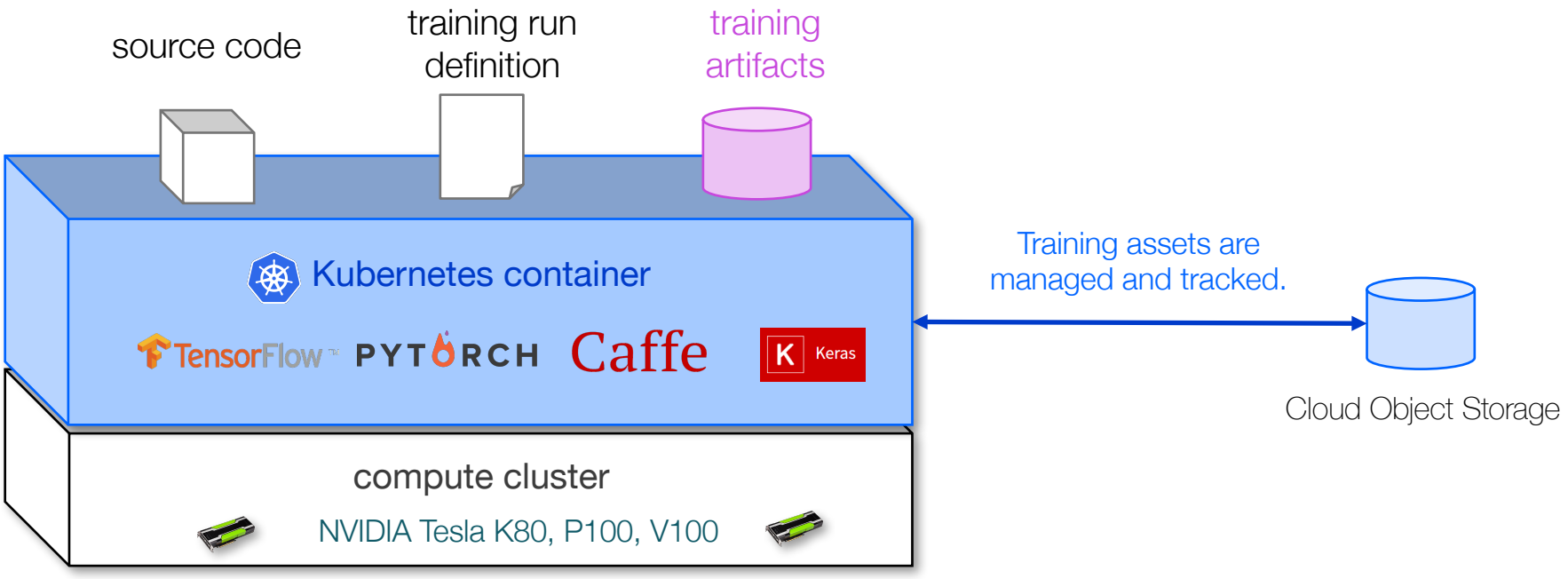
Experiment Management

Technical Deep Dive

Model Hosting

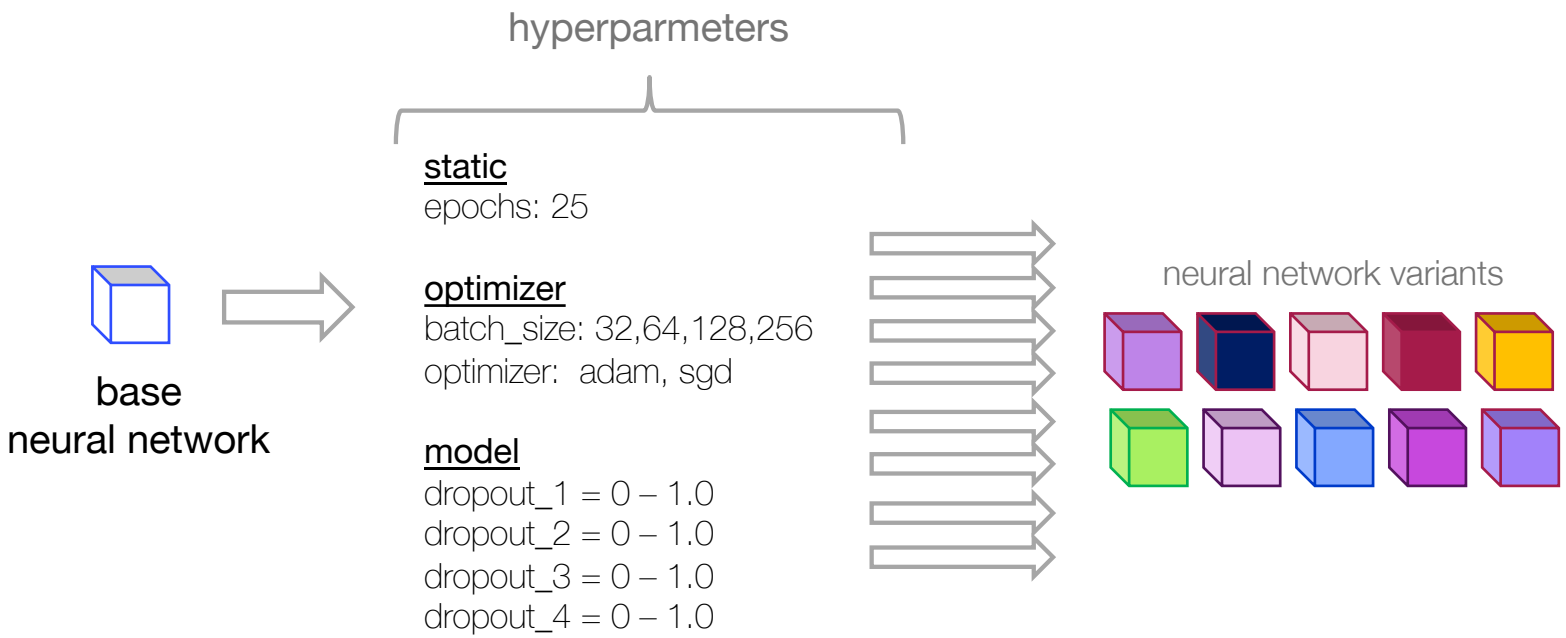
Training Lifecycle Management

Auto-allocation means paying only for what you use



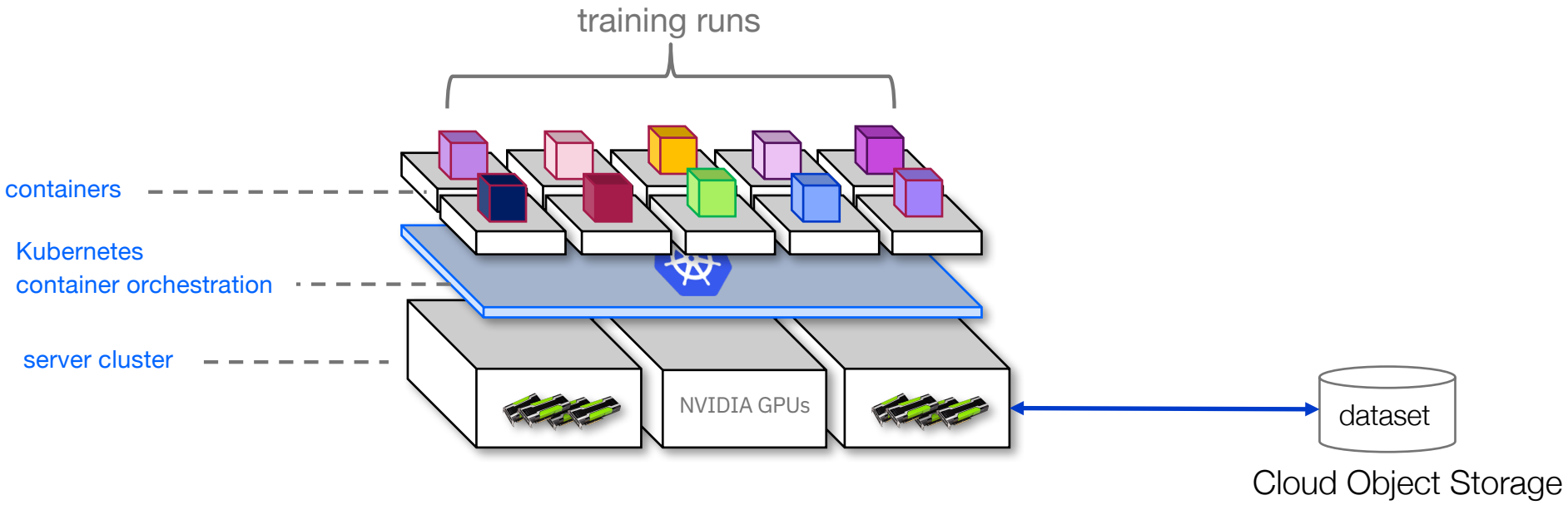
Train 1000's of neural networks

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



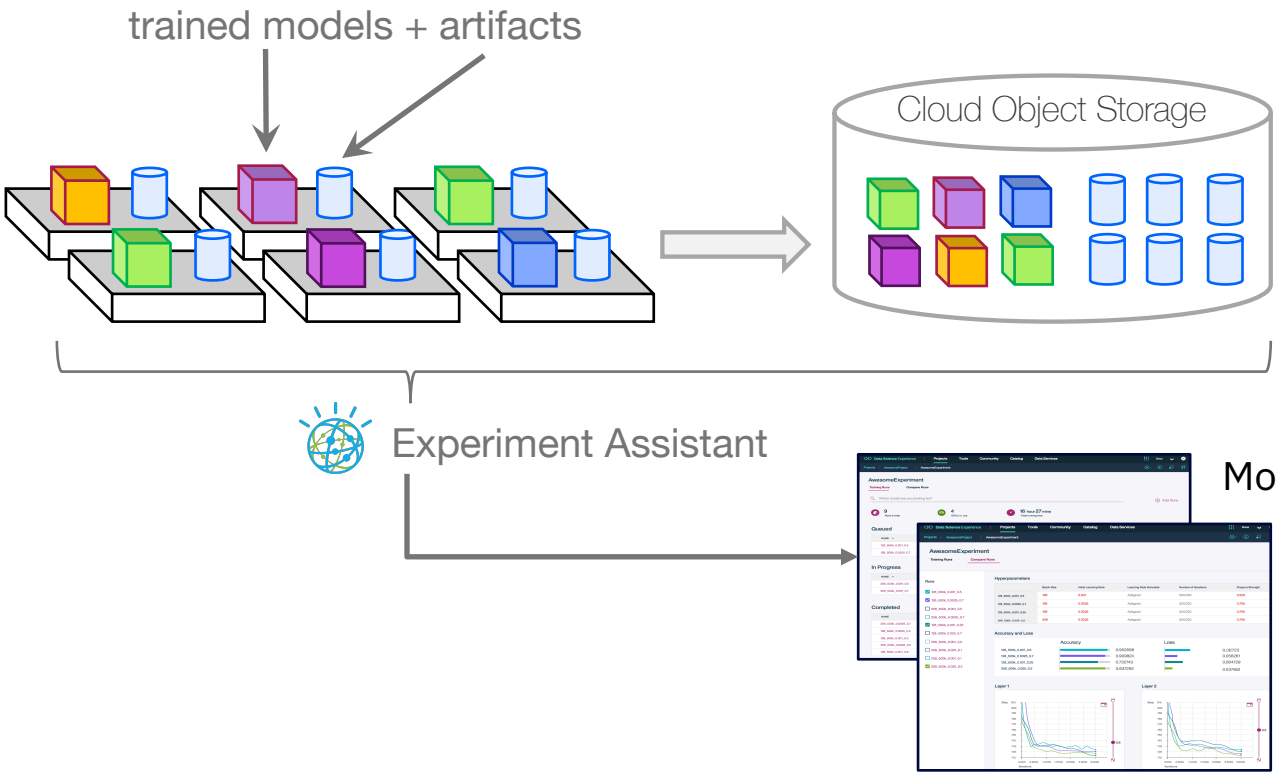
Training distributed across parallel containers

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



Monitor Training in Real-Time

Compare Model Performance

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.

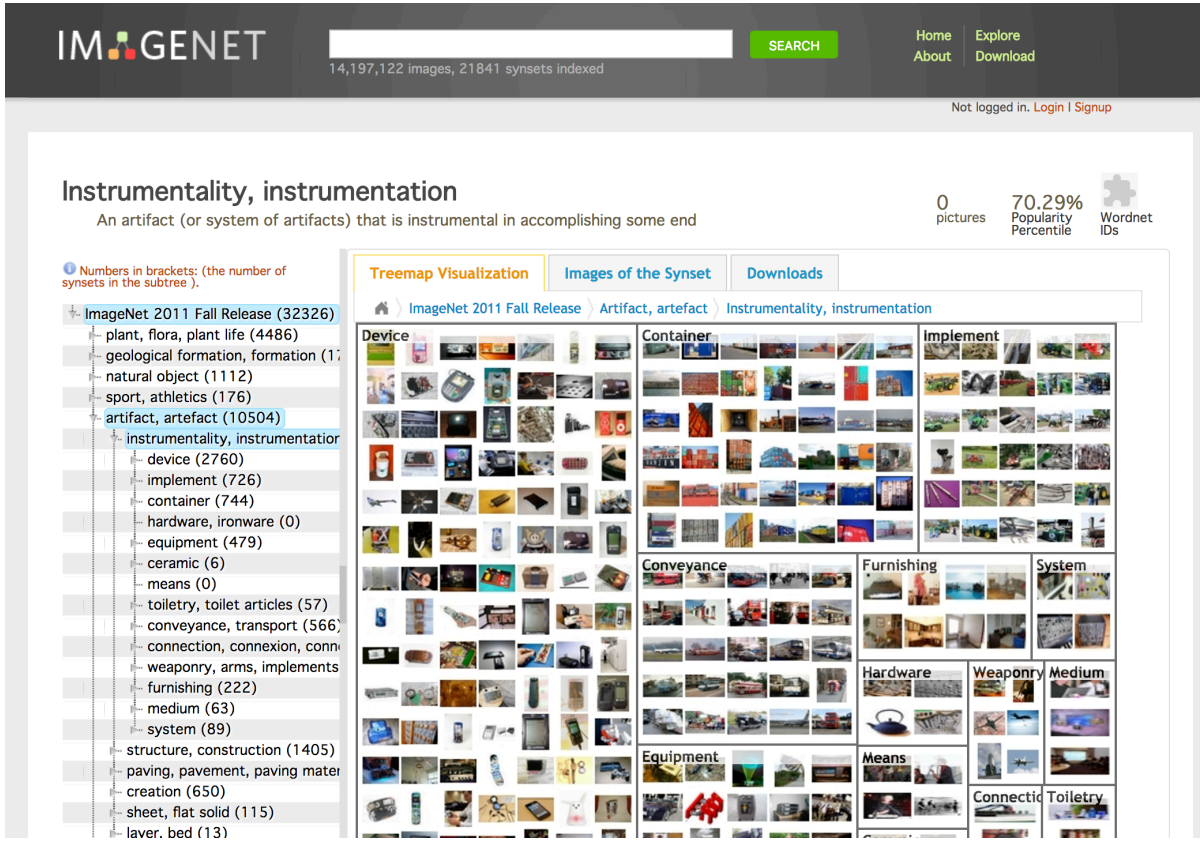
Models no longer fit on one or multiple GPUs within a single server and must be distributed across multiple machines.

Native TensorFlow

IBM Distributed Deep Learning (DDL)

Uber's Horovod

What about massive datasets like ImageNet 22K?

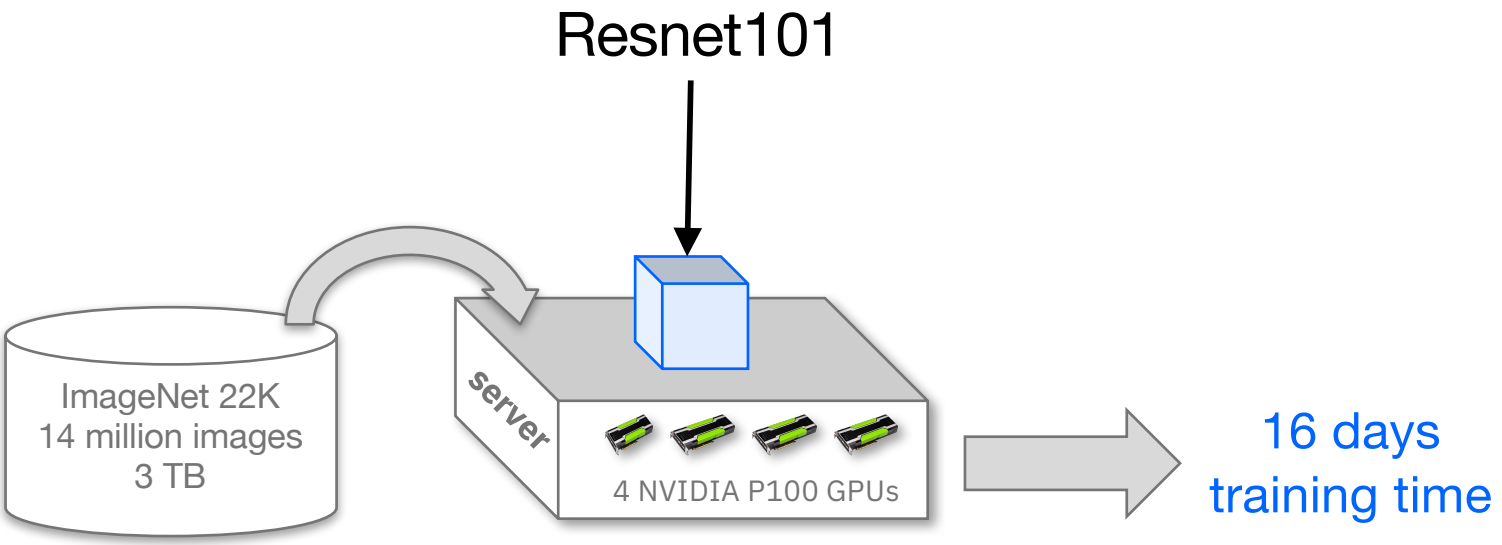


14 million images

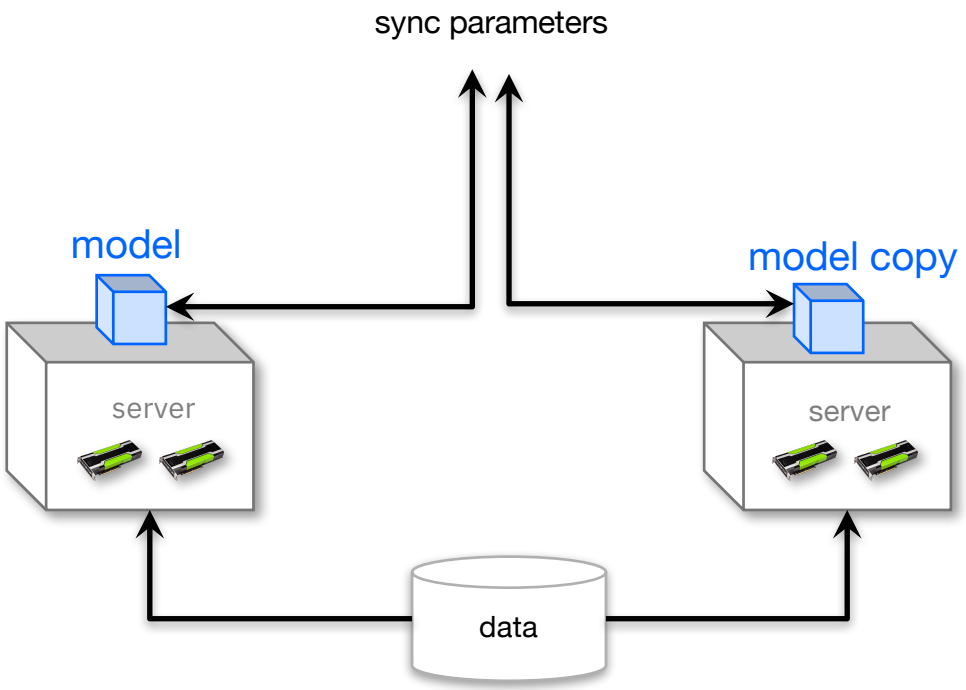
3 Terabytes

21,841 categories

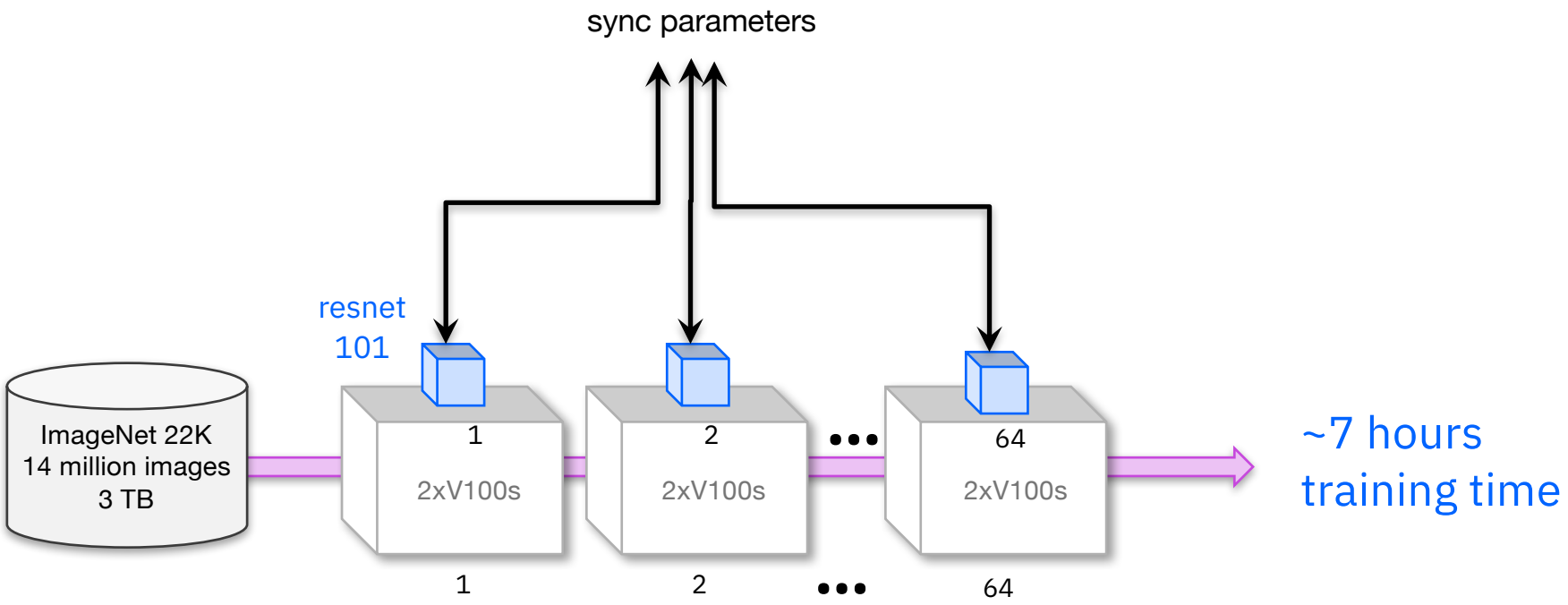
Single servers are too slow for complicated networks and large datasets



Distributed training across multiple servers



Distributed training at scale



Neural Network Design

Experiment Management

Technical Deep Dive

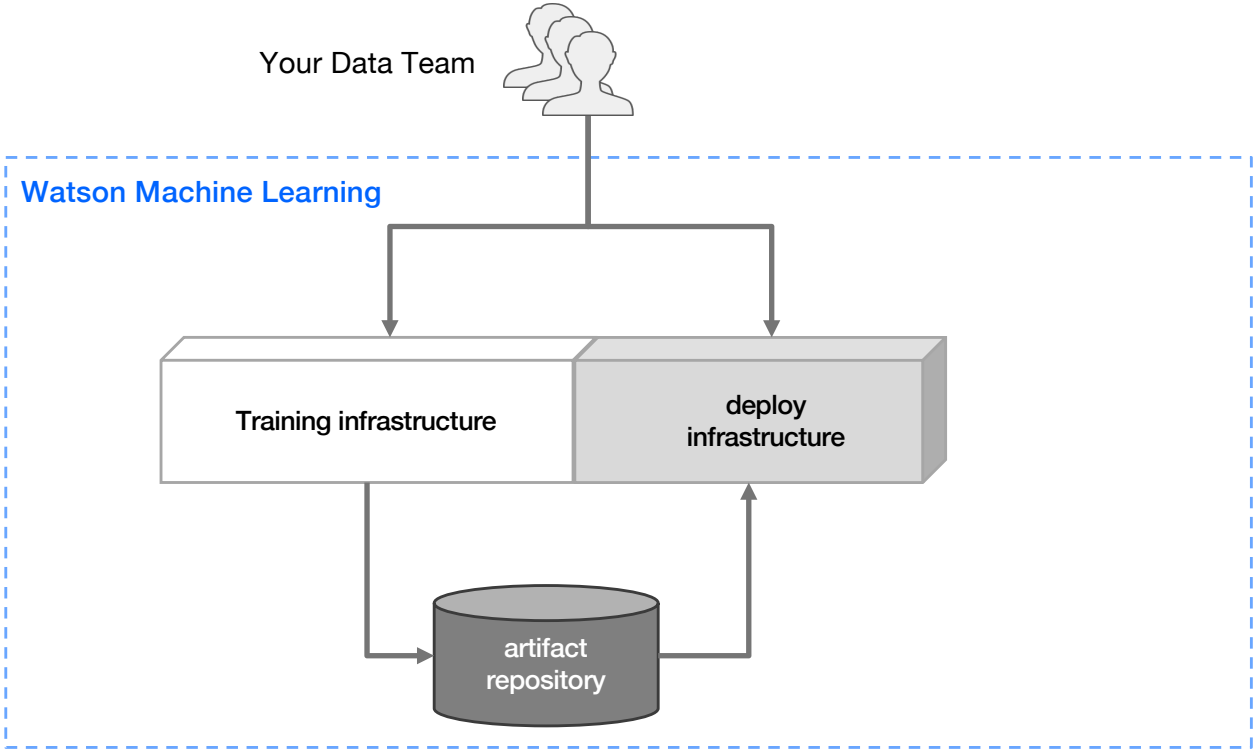
Model Hosting

Deep Learning

Available through Watson Machine Learning

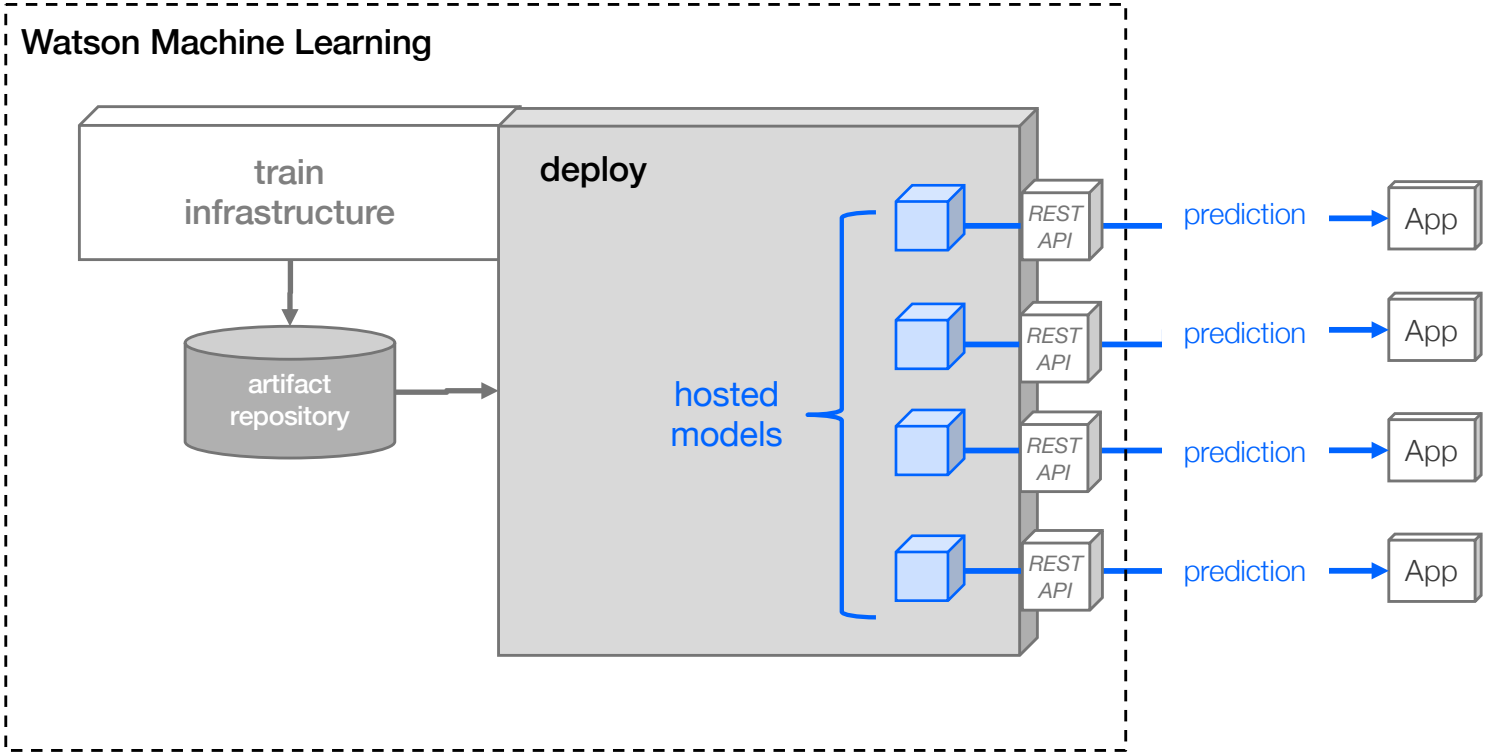
Hosted ML Infrastructure

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



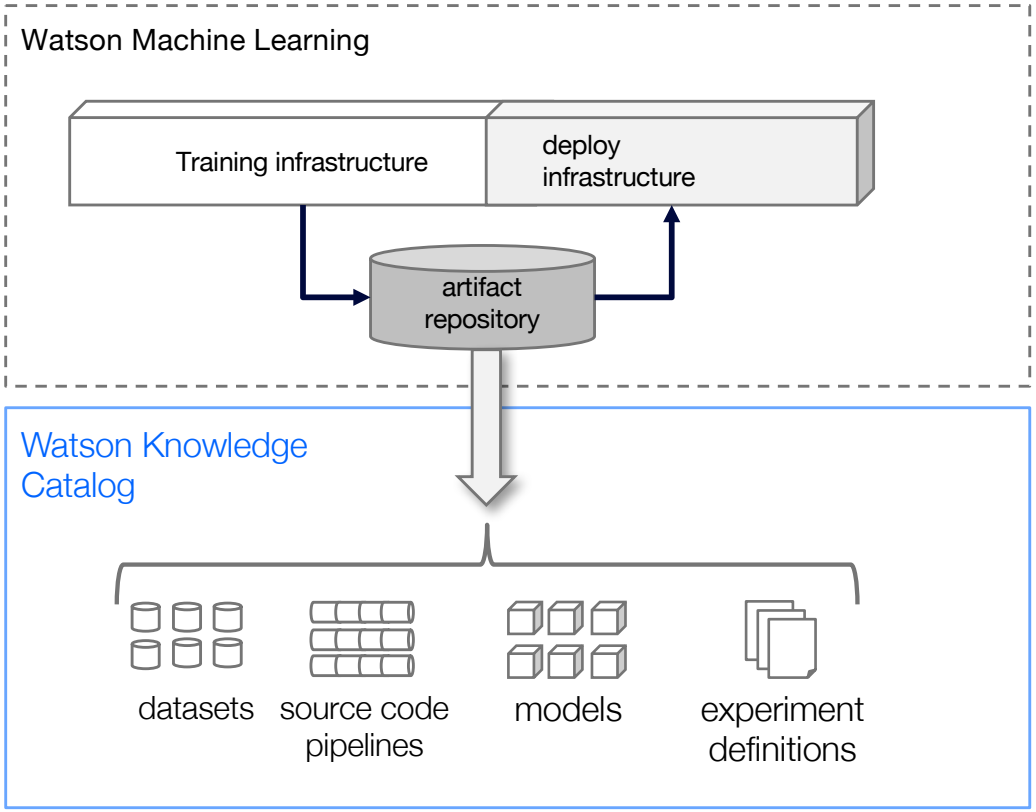
Deploying Trained Models

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 complicance. 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

- 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

Deploy

Move models into production where they improve and remain relevant

- 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

Automate

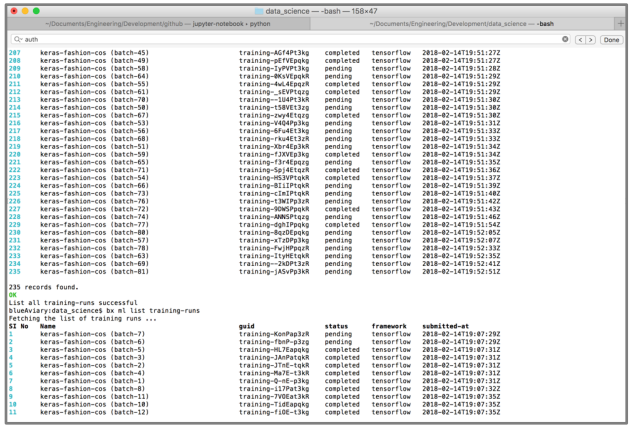
Build better models faster while they continue to learn after

- 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

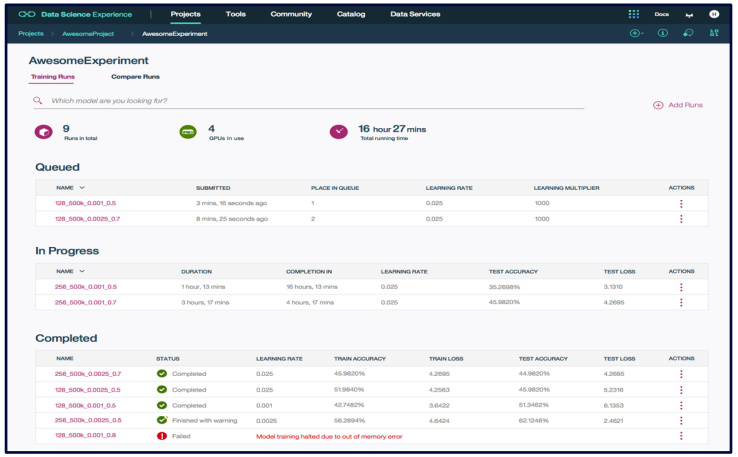
Simplifying deployment & management of ML models in production apps

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.

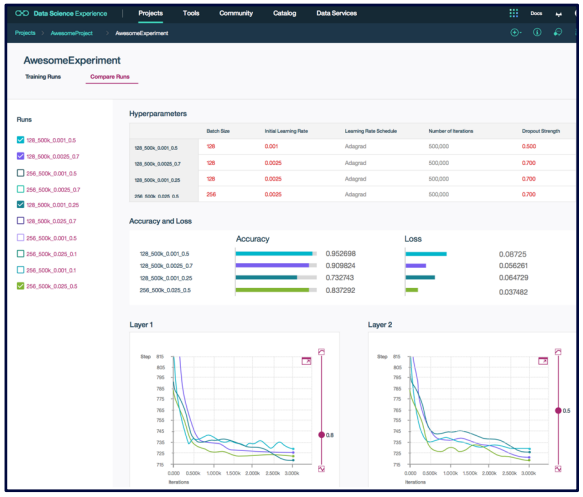


Train neural networks in parallel across NVIDIA GPUs.

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

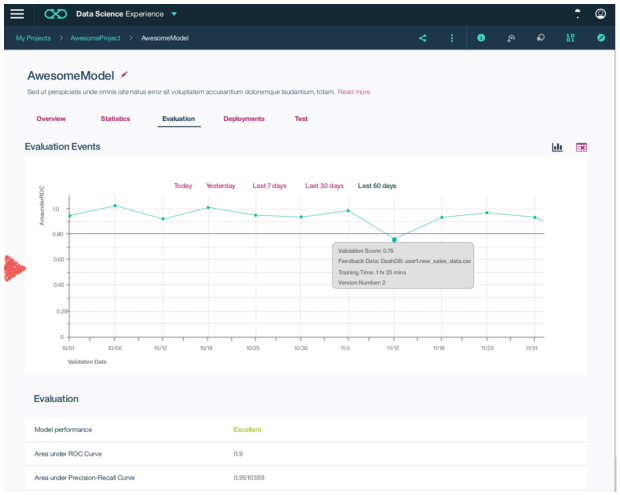


Monitor batch training experiments then compare cross-model performance without worrying about log transfers and scripts to visualize results. You focus on designing your neural networks. We'll manage and track your assets.



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

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.

Thank you

Anthony Stevens
Offering Manager, Watson Deep Learning
—
apsteven@us.ibm.com