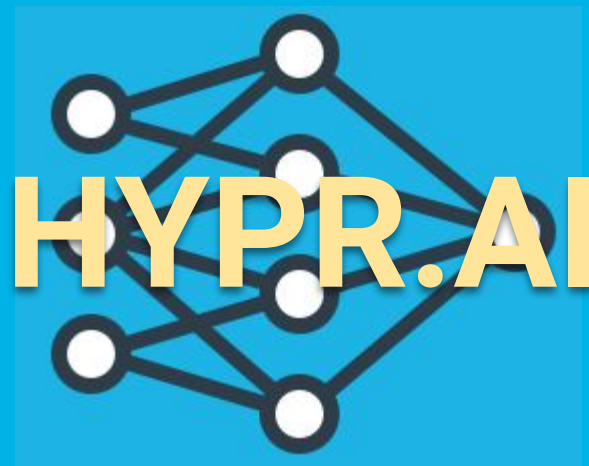


HYPR.AI

Cloud-based AutoML

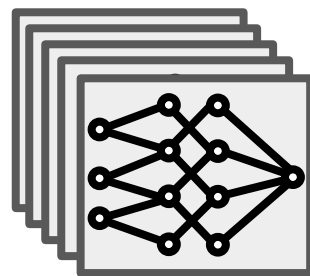
Ryan Reece
Insight AI Fellow

Also with Jiangming “Jimmy” Hu
Previous Insight Fellow



Intro

- AI is an *experimental* science
- Search & test many models: “*hyperparameters*”
- Better tools can streamline this process
- Leads to discovering and organizing better models
- Jeff Dean: tools for “ML 2.0” or “*AutoML*”

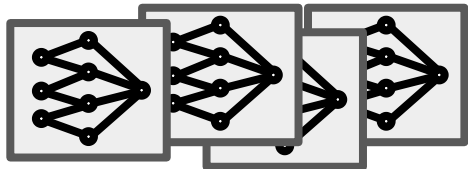


Google Cloud Platform

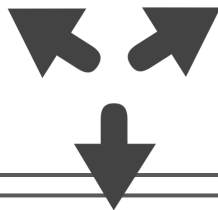
H₂O.ai

Typical ML workflow today

Models



Compute



```
Train the model!
This is the fun part: you can feed the training data loaded in earlier into this model and it will learn to classify digits

In [9]: model.fit(X_train, Y_train,
                  batch_size=128, nb_epoch=4,
                  show_accuracy=True, verbose=1,
                  validation_data=(X_test, Y_test))

Train on 60000 samples, validate on 10000 samples
Epoch 1/4
60000/60000 [=====] - 10s - loss: 0.2521 - acc: 0.9245 - val_loss: 0.1131
- val_acc: 0.9651
Epoch 2/4
60000/60000 [=====] - 10s - loss: 0.1016 - acc: 0.9687 - val_loss: 0.0827
- val_acc: 0.9746
Epoch 3/4
60000/60000 [=====] - 11s - loss: 0.0711 - acc: 0.9779 - val_loss: 0.0668
- val_acc: 0.9791
Epoch 4/4
60000/60000 [=====] - 11s - loss: 0.0557 - acc: 0.9816 - val_loss: 0.0642
- val_acc: 0.9805
Out[9]: <keras.callbacks.History at 0x12655bbe0>

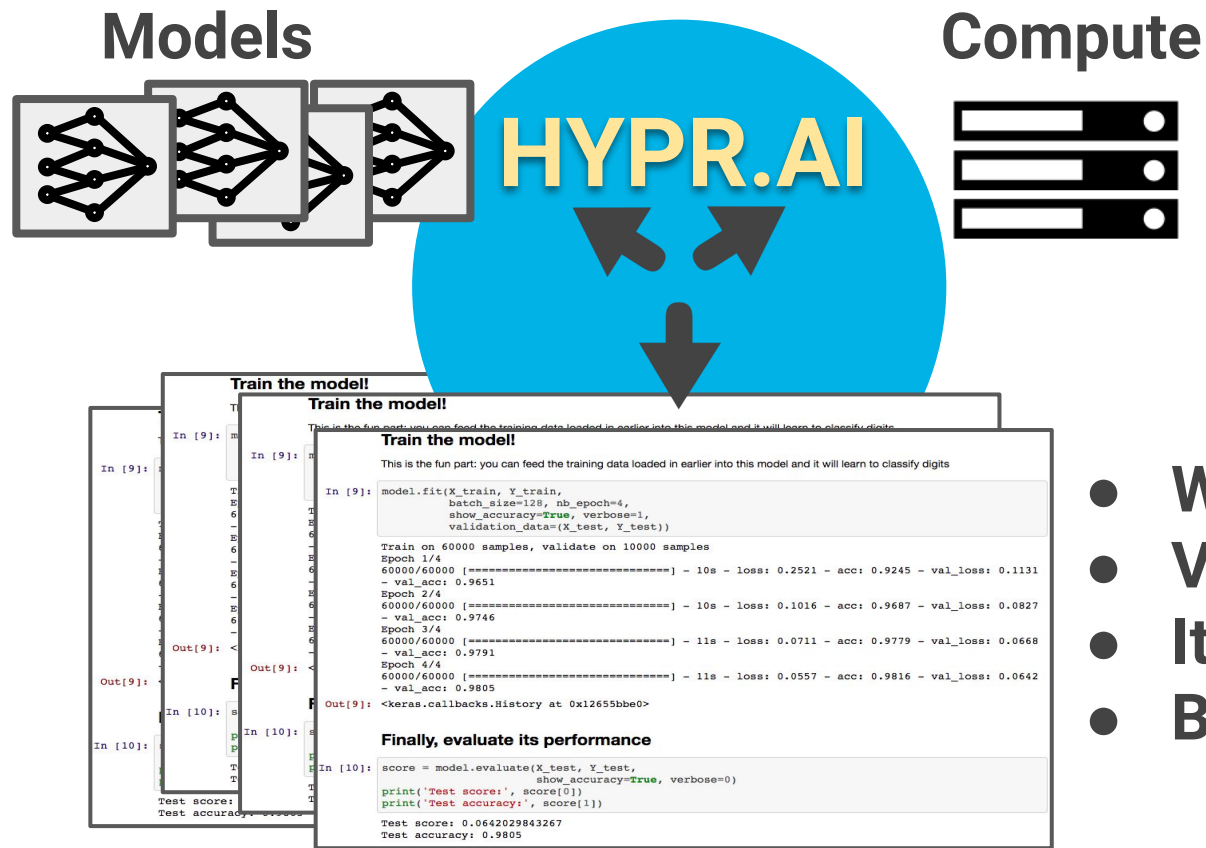
Finally, evaluate its performance

In [10]: score = model.evaluate(X_test, Y_test,
                                show_accuracy=True, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

Test score: 0.0642029843267
Test accuracy: 0.9805
```

- Workflow
- Visualization
- Iterate
- Bookkeeping?

Typical ML workflow today



- Workflow
- Visualization
- Iterate
- Bookkeeping?

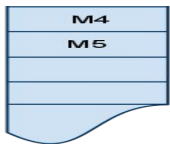
HYPER.AI provides



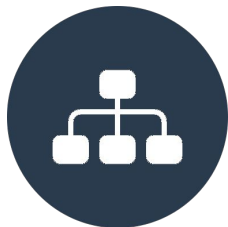
Website UI: construct or upload keras models



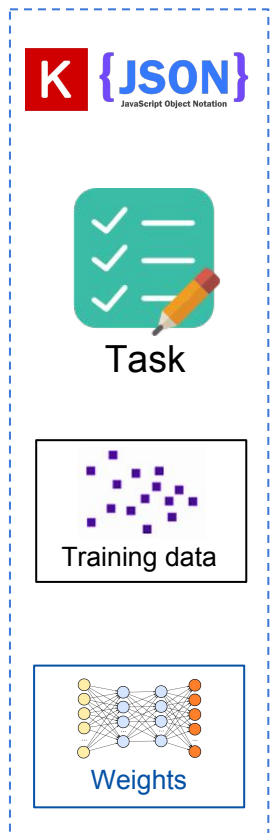
Database for storing models and results



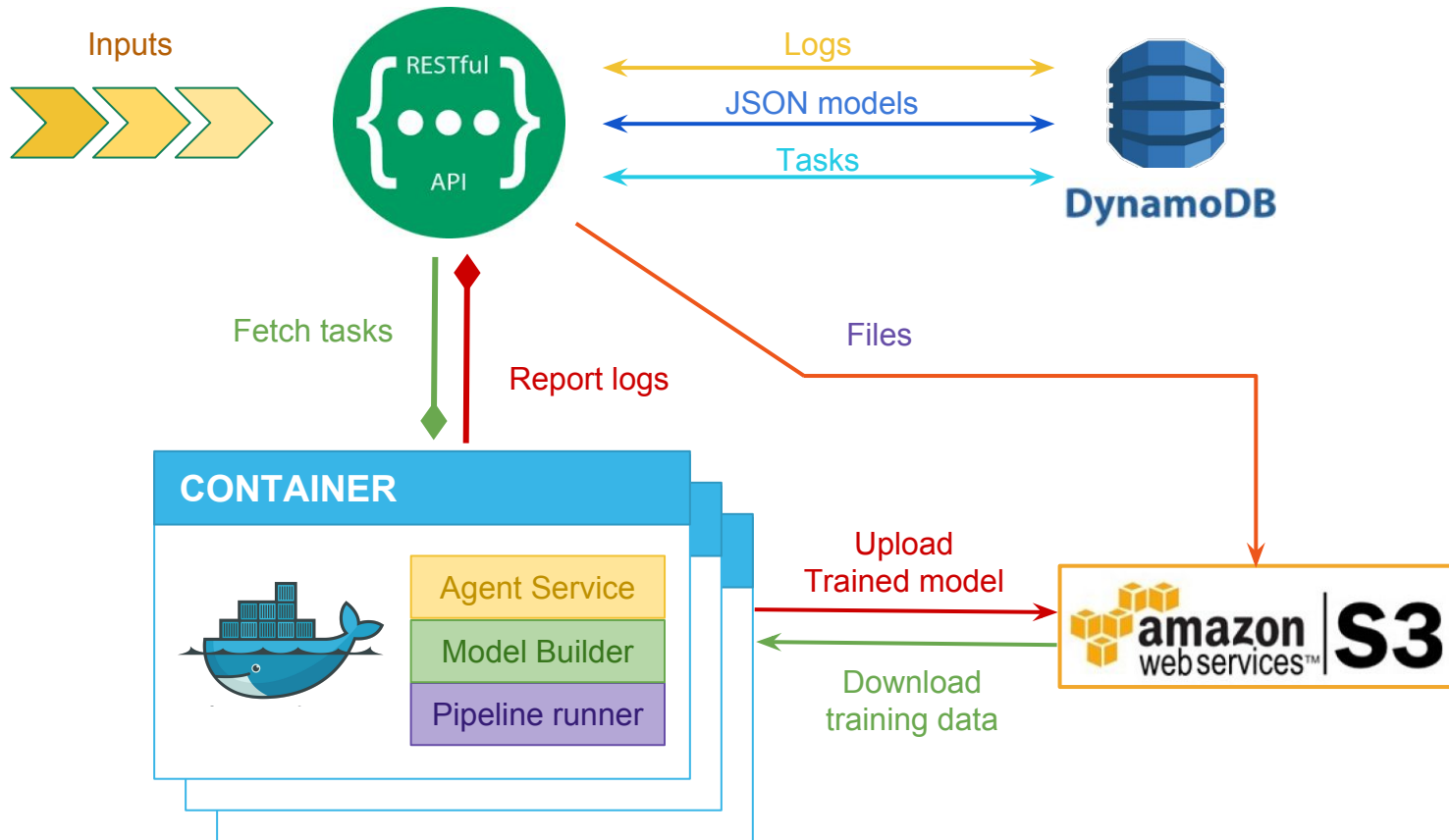
Task queue for training many models



Compute back-ends for AWS & Paperspace



You





HYPR.AI

Cloud-based AutoML

Models

Tasks

Results

Storage

Cluster

Settings



Models

+ New



CNN-Base

Convolution2D (conv1) - inputs: [null,32,32,3] fi...

Convolution2D (conv2) - filters: 64 kernel_size: ...

MaxPooling2D (pool1) - pool_size: [2,2] strides: ...

Dropout (dropout1) - rate: 0.25

Convolution2D (conv3) - filters: 128 kernel_size:...

MaxPooling2D (pool2) - pool_size: [2,2] strides: ...

Convolution2D (conv4) - filters: 128 kernel_size:...

MaxPooling2D (pool3) - pool_size: [2,2] strides: ...

Dropout (dropout2) - rate: 0.25

Flatten

Dense (dense1) - units: 1024 activation: relu

Dropout (dropout3) - rate: 0.5

Dense (softmax1) - units: 10 activation: softmax

Compiler optimizer: {"adam":{"lr":0.0001}} loss: ...



CNN-RMSprop

From: CNN-Base

input softmax1, Compiler: optimizer: rmsprop loss...



HYPR.AI

Cloud-based AutoML

Mo
Ta
Re
Sto
Clu



Models

+ New



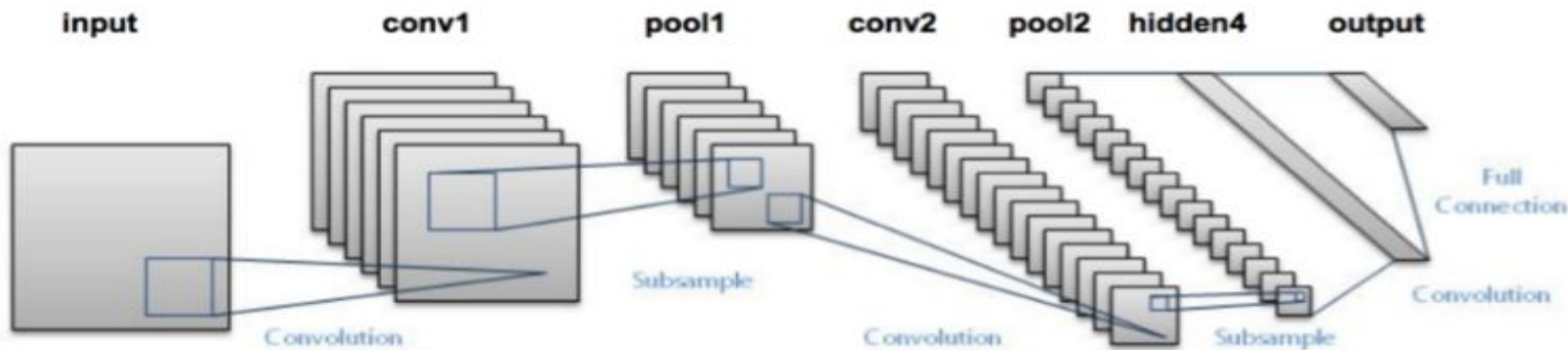
CNN-Base

Convolution2D (conv1) - inputs: [null,32,32,3] fi...



CNN-RMSprop

From: CNN-Base



Dense (dense1) - units: 1024 activation: relu

Dropout (dropout3) - rate: 0.5

Dense (softmax1) - units: 10 activation: softmax

Compiler optimizer: {"adam":{"lr":0.0001}} loss: ...

Add new hyper task



Task Name

test

Max Epoch

50

JSON model

CNN-Base



Dataset

dataset/cifar-10



Hyperparameters

{'conv2/filters' : ['ci', [16, 32, 64]], 'conv4/filters' : ['ci', [64, 128, 256]], 'dropout2/rate' : ['cf', [0.25]

Weights (pretrained model)

NONE



Cancel

Save

Settings



reece-test-02

CNN-Base

dataset/cifar-10

10

completed

2018-01-19



HYPR.AI

Cloud-based AutoML

Models

Tasks

Results

Storage

Cluster

Settings



	test-012	CNN-Base-test-012	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-011	CNN-Base-test-011	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-010	CNN-Base-test-010	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-009	CNN-Base-test-009	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-008	CNN-Base-test-008	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-007	CNN-Base-test-007	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-006	CNN-Base-test-006	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-005	CNN-Base-test-005	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-004	CNN-Base-test-004	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-003	CNN-Base-test-003	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-002	CNN-Base-test-002	dataset/cifar-10	50	initial	2018-01-30 22:43
	test-001	CNN-Base-test-001	dataset/cifar-10	50	training	2018-01-30 22:43



HYPR.AI

Cloud-based AutoML

Models

Tasks

Results

Storage

Cluster

Settings



Results



Name	Loss (train/test)	Accuracy (train/test)	Best Epoch	Created
test-025	0.092113 / 0.003807	0.96932 / 0.9999	52 / 52	2018-01-31 05:05
test-024	0.30405 / 0.044943	0.8915 / 0.9949	99 / 100	2018-01-31 05:05
test-023	0.37652 / 0.082246	0.86504 / 0.987	100 / 100	2018-01-31 05:05
test-022	0.56593 / 0.23273	0.79544 / 0.9376	100 / 100	2018-01-31 05:05
test-021	0.17968 / 0.0084701	0.93818 / 0.9998	99 / 99	2018-01-31 05:05
test-020	0.14708 / 0.0075892	0.94836 / 0.9999	82 / 82	2018-01-31 05:05
test-019	0.14485 / 0.01072	0.95078 / 0.9999	80 / 81	2018-01-31 05:05
test-018	0.13193 / 0.0083731	0.9543 / 0.9999	59 / 60	2018-01-31 05:05
test-017	0.069662 / 0.0028221	0.97752 / 0.9999	51 / 51	2018-01-31 05:05



HYPR.AI

Cloud-based AutoML

Models

Tasks

Results

Storage

Cluster

Settings



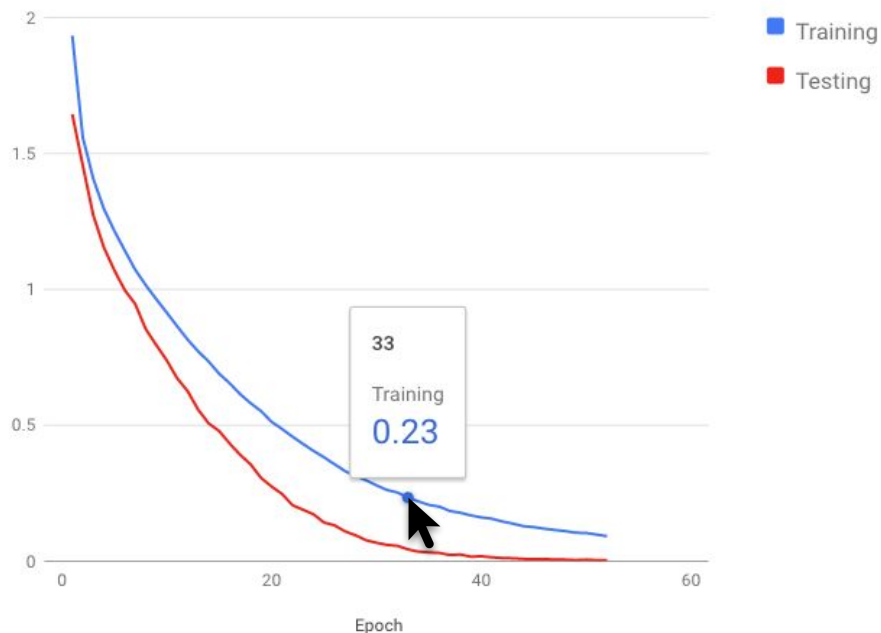
test-025

Use model: CNN-Base-test-025 ----- Max Epochs: 100

Dataset: dataset/cifar-10

Status: completed

Loss vs epoch

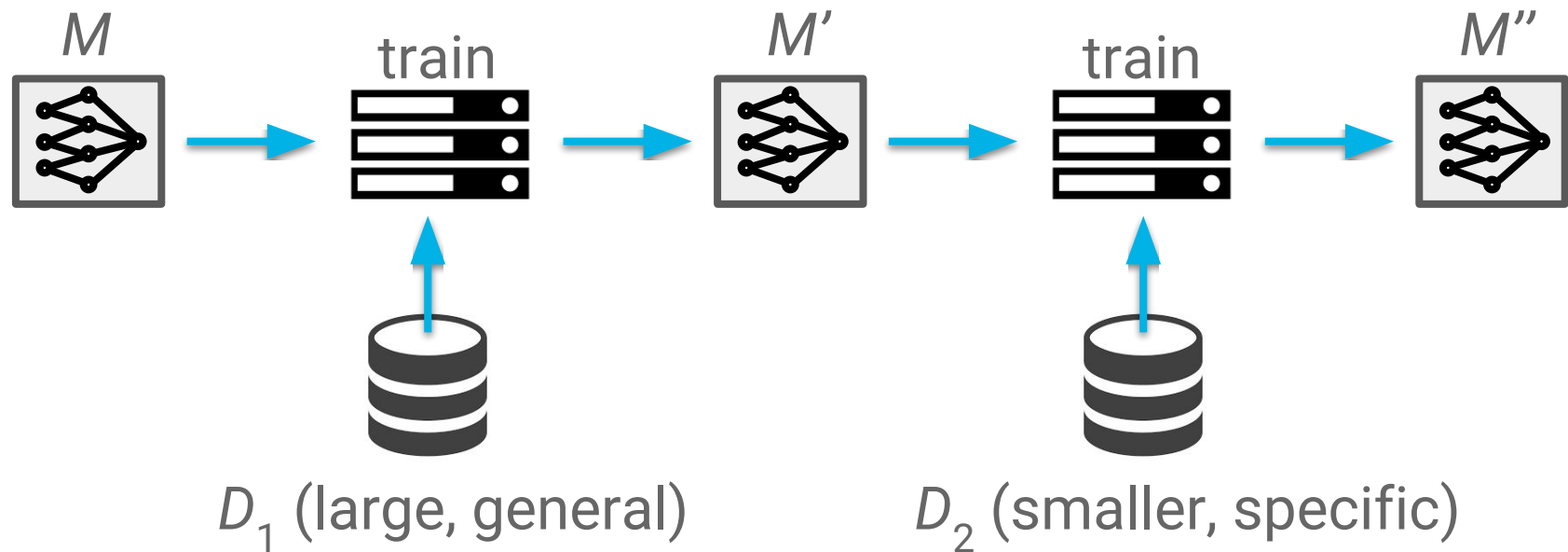


Close

test epoch	Created
2 / 52	2018-01-31 05:05
9 / 100	2018-01-31 05:05
00 / 100	2018-01-31 05:05
00 / 100	2018-01-31 05:05
9 / 99	2018-01-31 05:05
2 / 82	2018-01-31 05:05
0 / 81	2018-01-31 05:05
9 / 60	2018-01-31 05:05
1 / 51	2018-01-31 05:05

Transfer learning

Train a model on one dataset, and then more on another.

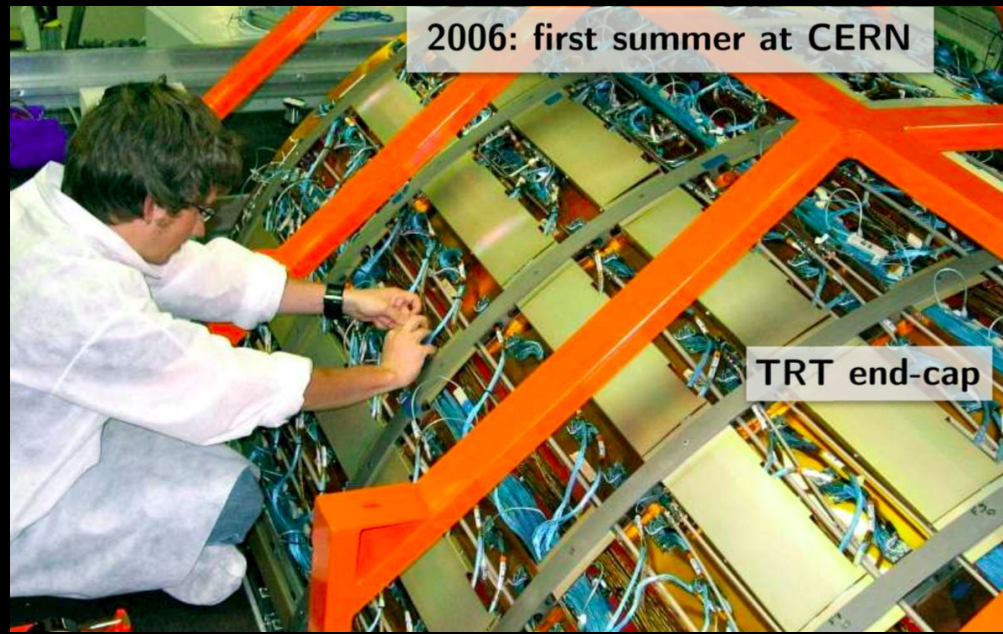
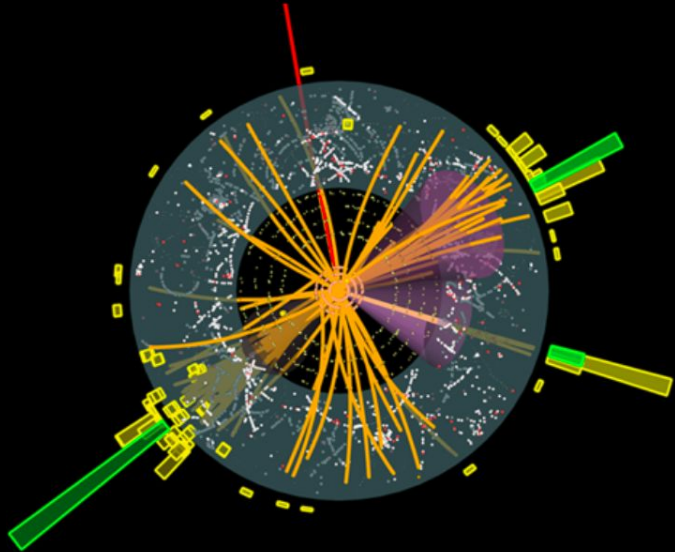


Advantages of using HYPR.AI

- Bookkeeping datasets/models/weights in the cloud
- RESTful API allows back-end independence
- Modular models facilitate transfer learning
- Hyperparameter optimization: currently random search or scan. Possible future plugins: Bayesian?, RL?
- Try out the running example: <http://hypr.umx.io>

Ryan Reece, Ph.D.
Data scientist, AI/ML/Stats

**Previously a particle physicist
with the ATLAS experiment at
the Large Hadron Collider**



CERN



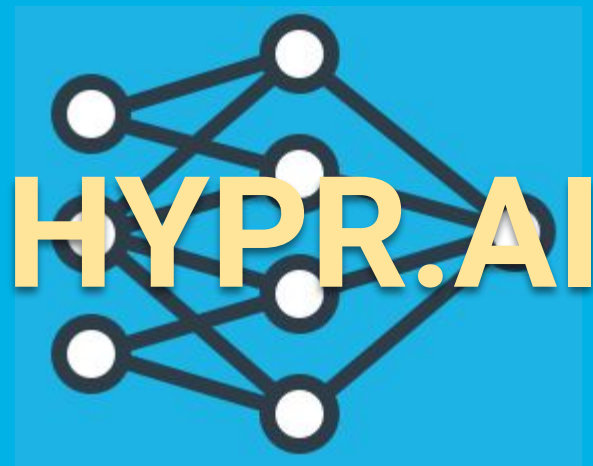
traveler



climber



Backup slides





Google Cloud Platform

CLOUD AUTOML^{ALPHA}

Train high quality custom machine learning models with minimum effort and machine learning expertise



REQUEST ACCESS

H₂O.ai



HYPR.AI

Cloud-based AutoML

Models

Tasks

Results

Storage

Cluster

Settings



Results



TODO


#	Name	Loss (train/test)	Accuracy (train/test)	Best Epoch	Created
	test-b-025	0.19725 / 0.032007	0.93328 / 0.998	40 / 40	2018-01-30 19:04
	test-b-024	0.31456 / 0.091606	0.88984 / 0.9894	40 / 40	2018-01-30 19:04
	test-b-023				2018-01-30 19:04
	test-b-022	0.56193 / 0.32403	0.80062 / 0.9151	40 / 40	2018-01-30 19:04
	test-b-021				2018-01-30 19:04
	test-b-020	0.22183 / 0.043774	0.92308 / 0.9971	40 / 40	2018-01-30 19:04
	test-b-019				2018-01-30 19:04
	test-b-018	0.51466 / 0.27159	0.81436 / 0.9313	40 / 40	2018-01-30 19:04




HYPR.AI

Cloud-based AutoML

 Models

 Tasks

 Results

 Storage

Dataset

Weights

 Cluster

Cluster (workers)



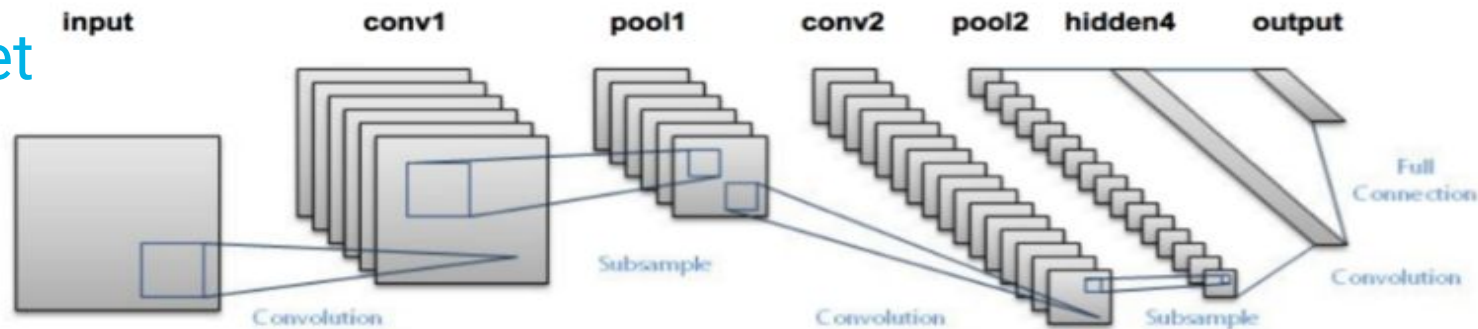
Name	Status	Last seen	Server Info
paperspace	idle	2018-01-26 17:36:27	{}
_paperspace-psrxvede4	offline	2018-01-22 21:15:56	{}
_paperspace_psrxvede4	offline	2018-01-22 21:06:16	{}
54.153.100.55	offline	2018-01-19 04:22:31	{}
AWS-g2.2xlarge	offline	2018-01-12 20:18:01	{}
PAPERSPACE-P5000-1	offline	2018-01-12 20:17:58	{}

Data and models

- As proofs of concept, using standard image classification datasets: CIFAR-10, MNIST

1 1 5 4 3
7 5 3 5 3
5 5 9 0 6
3 5 2 0 0

- Successful models use several layers of CNNs + pooling, e.g. LeNet



- With this platform, easily upload and test many architectures

Goals

- Demonstrate a successful hyperparameter scan
- Use our system to discover a performant model
- Future upgrades could add better optimization (e.g. SigOpt / custom RL?)

