Google Cloud

Hyperparameter Tuning

Fereshteh Mahvar

Machine Learning on
Google Cloud Platform

The Art of ML

Hyperparameter Tuning

A Pinch of Science

The Science of Neural Networks

Embeddings

Custom Estimator

Learn how to...

Differentiate between
parameters and hyperparameters

Learn how to...

Differentiate between parameters and hyperparameters

Think beyond simple grid search algorithms

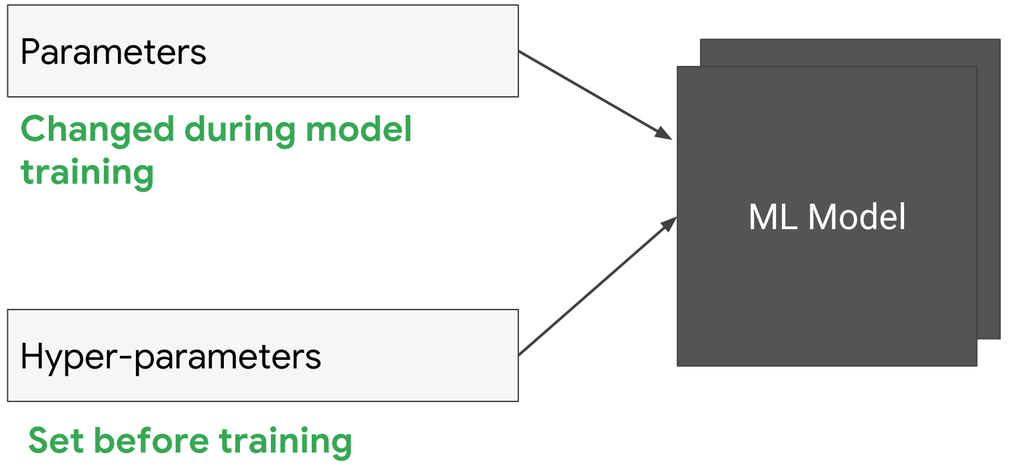
Learn how to...

Differentiate between parameters and hyperparameters

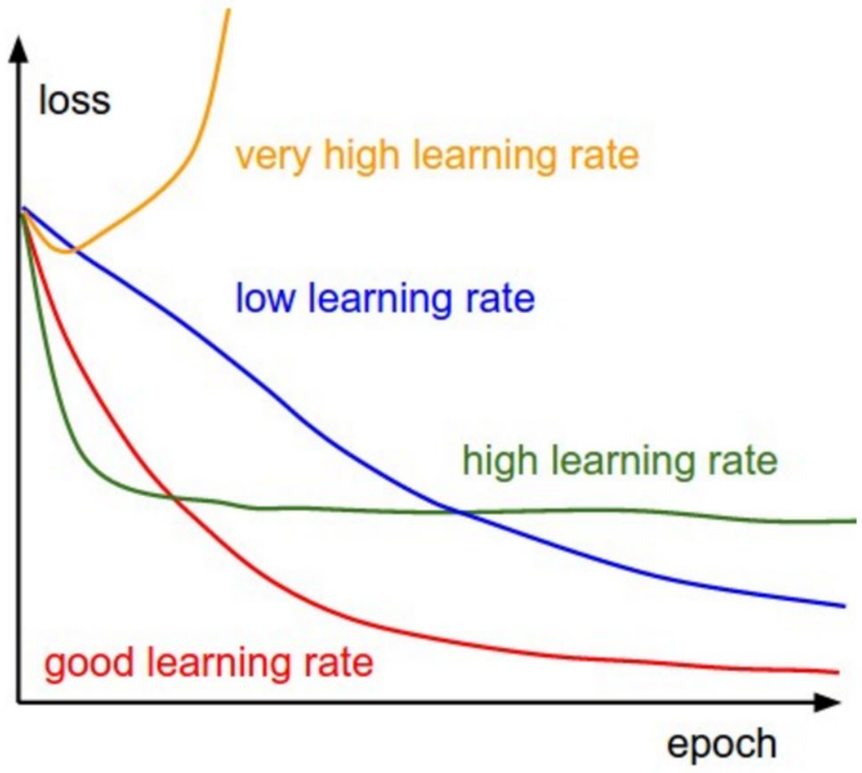
Think beyond simple grid search algorithms

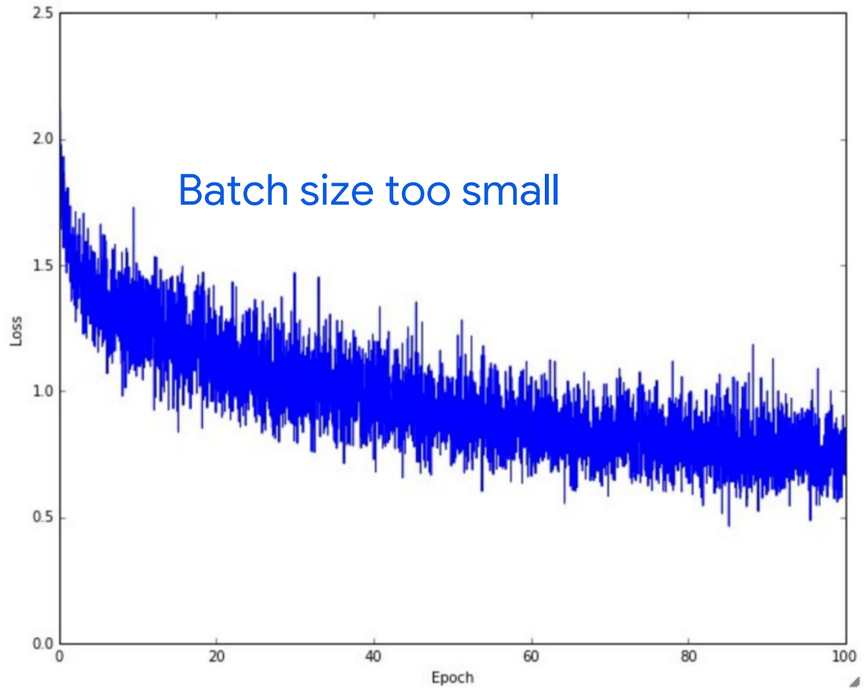
Take advantage of Cloud ML Engine for hyperparameter tuning

ML models are mathematical functions with parameters and hyper-parameters



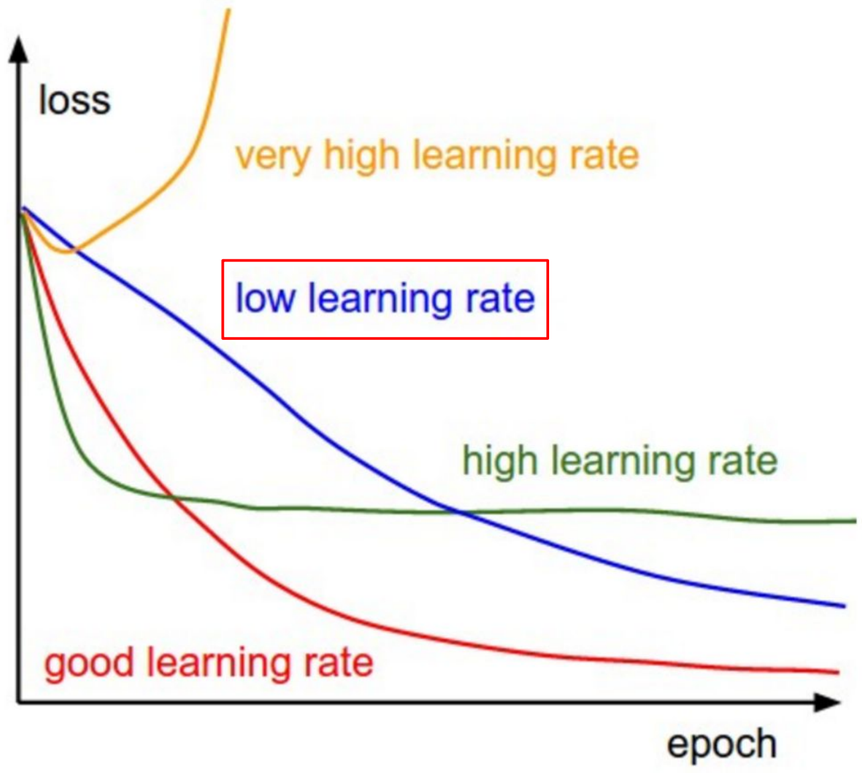
Model improvement is very sensitive to batch_size and learning_rate

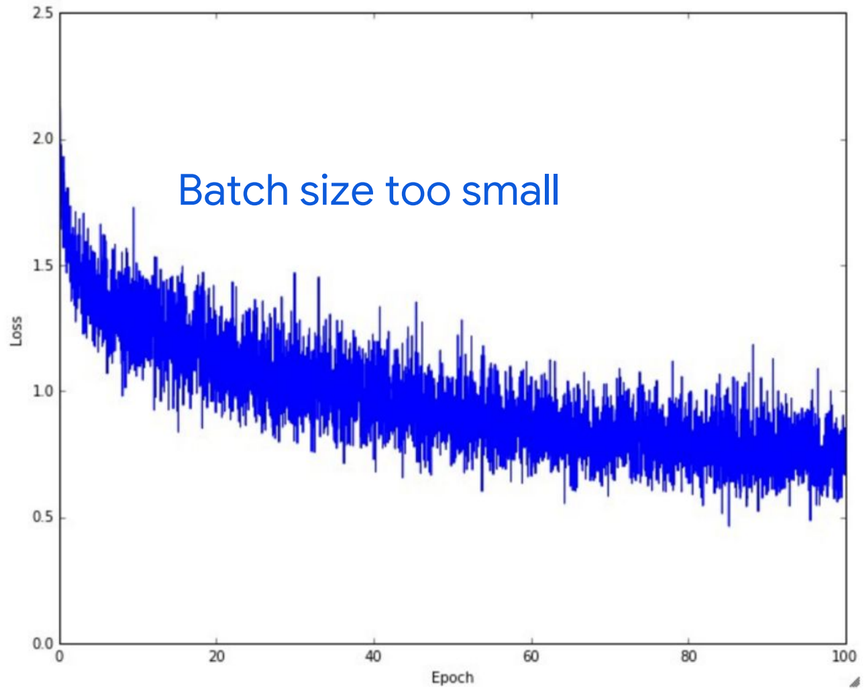




Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

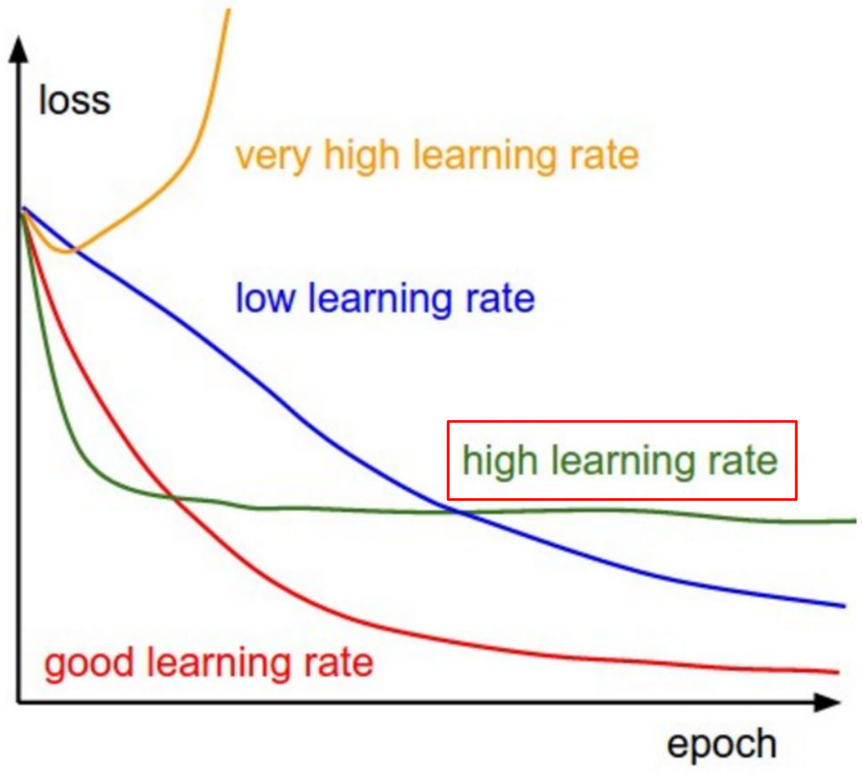
Model improvement is very sensitive to batch_size and learning_rate

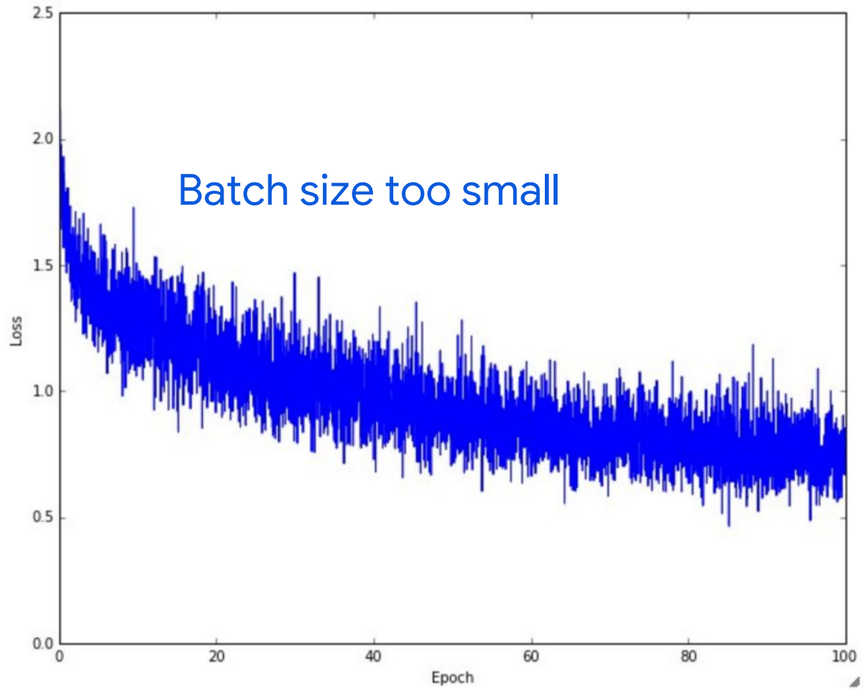




Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

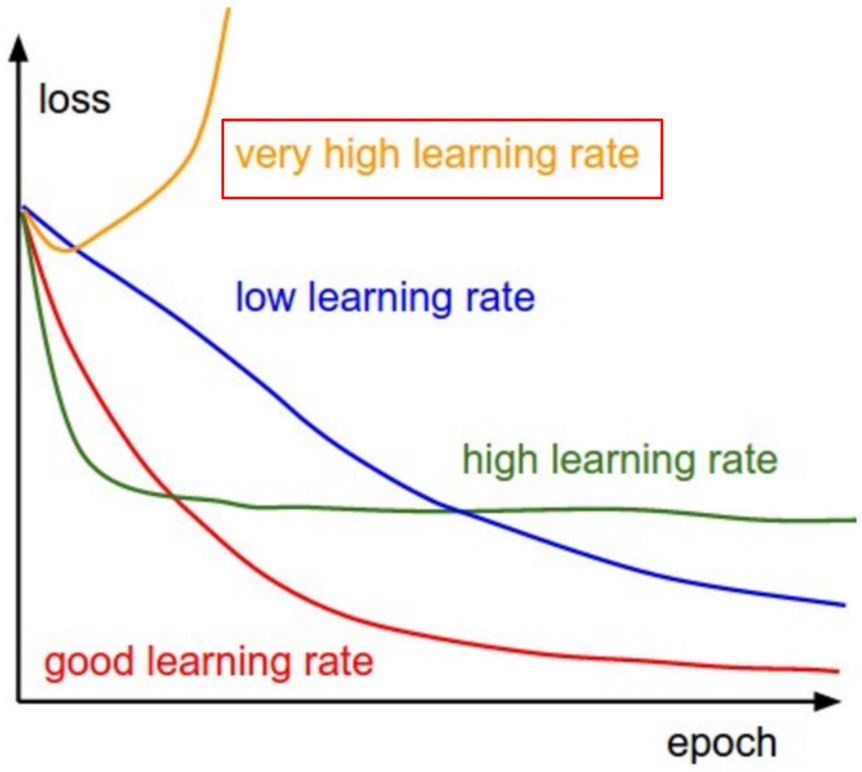
Model improvement is very sensitive to batch_size and learning_rate

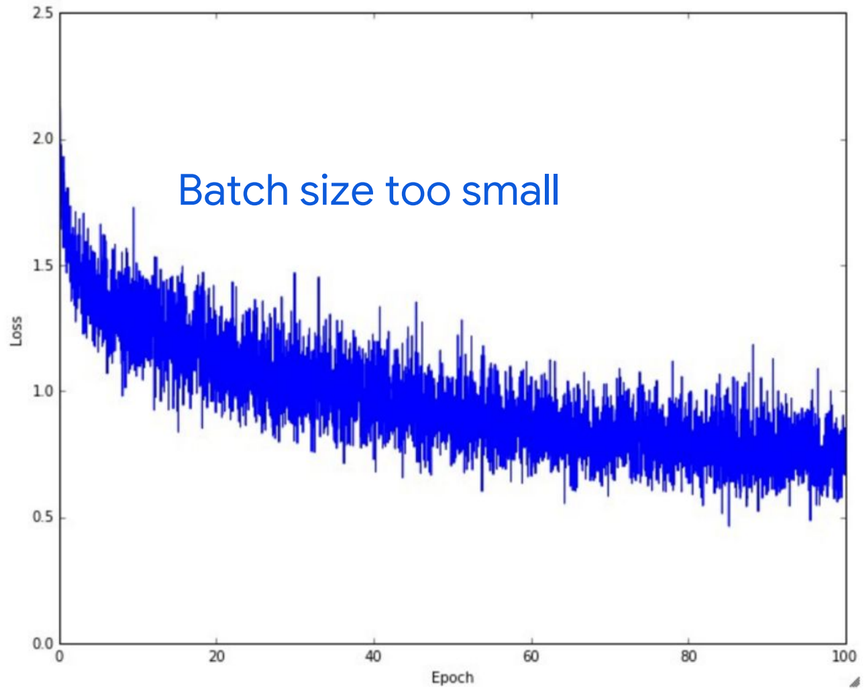




Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

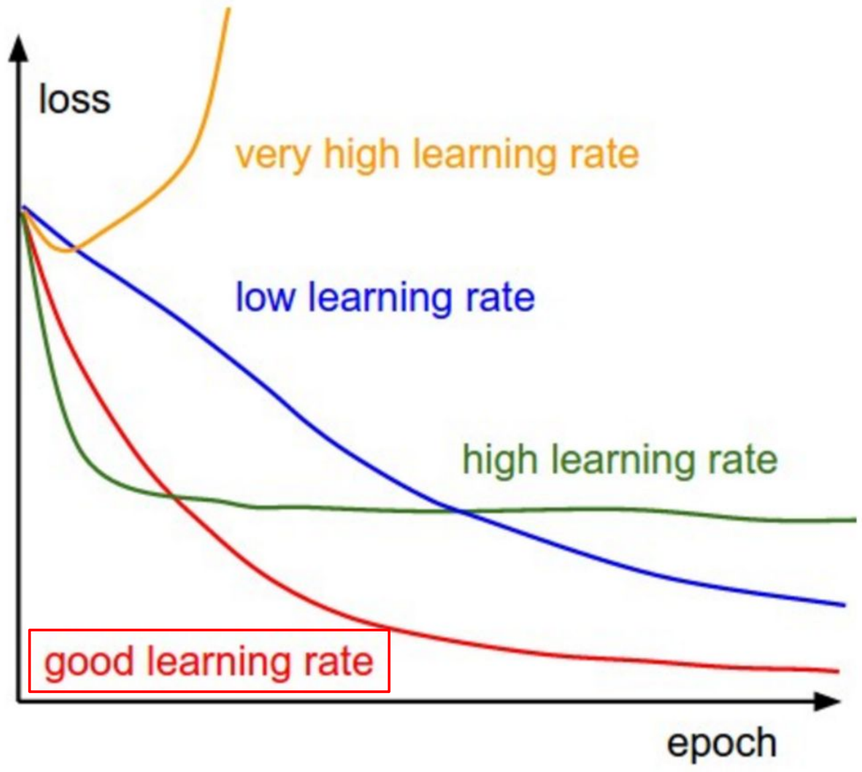
Model improvement is very sensitive to batch_size and learning_rate

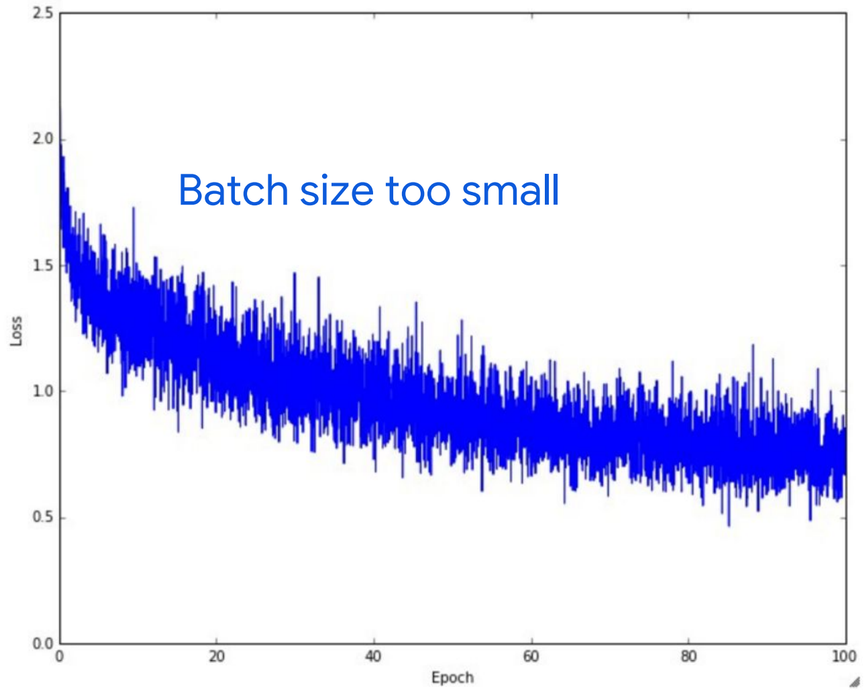




Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

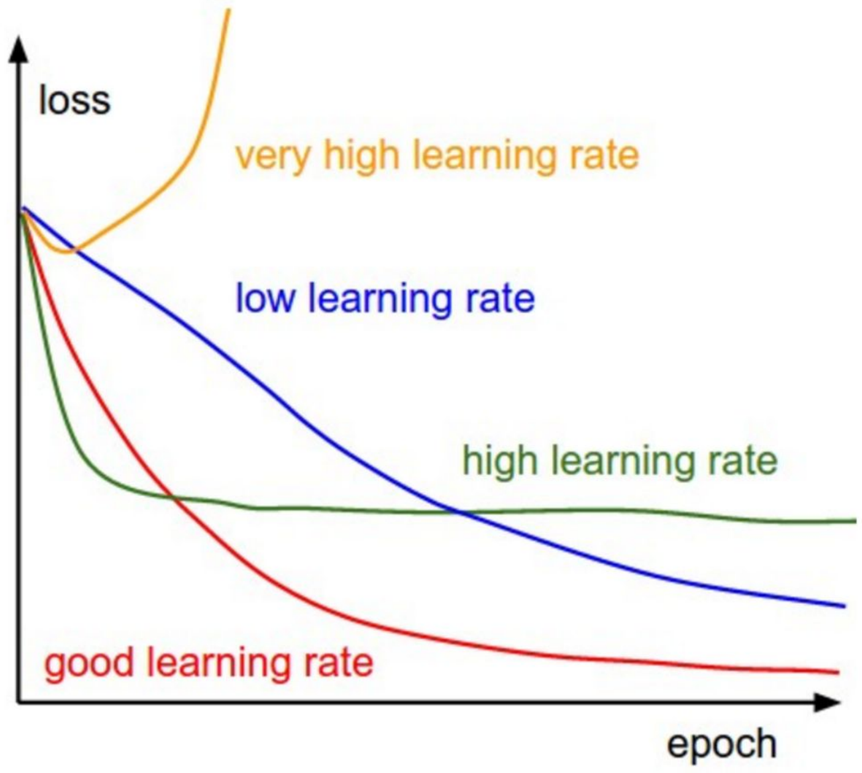
Model improvement is very sensitive to batch_size and learning_rate

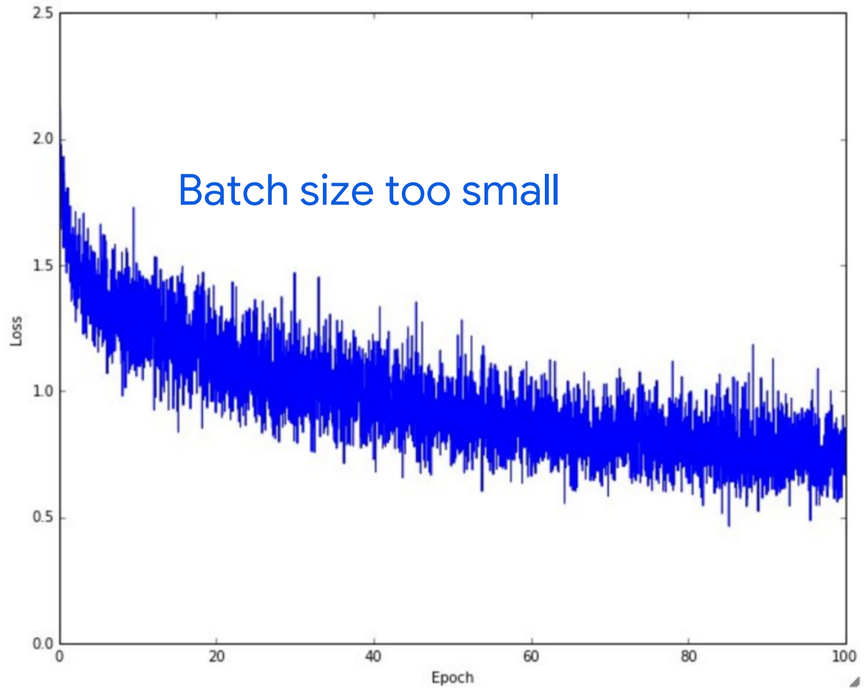




Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

Model improvement is very sensitive to batch_size and learning_rate





Source: http://cs231n.github.io/neural-networks-3/ by Andrej Karpathy

There are a variety of model parameters too

Size of model

Number of hash buckets

Embedding size

Etc.

Wouldn’t it be nice to have the NN training loop do meta-training across all these parameters?

Fearn not! Google Vizier is at your service!

Google Vizier: A Service for Black-Box Optimization

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, D. Sculley

{dgg, bsolnik, smoitra, gpk, karro, dsculley}@google.com
Google Research
Pittsburgh, PA, USA

ABSTRACT

Any sufficiently complex system acts as a black box when it becomes easier to experiment with than to understand. Hence, black-box optimization has become increasingly important as systems have become more complex. In this paper we describe *Google Vizier*, a Google-internal service for performing black-box optimization that has become the de facto parameter tuning engine at Google. Google Vizier is used to optimize many of our machine learning models and other systems, and also provides core capabilities to Google's Cloud Machine Learning *HyperTune* subsystem. We discuss our requirements, infrastructure design, underlying algorithms, and advanced features such as transfer learning and automated early stopping that the service provides.

In this paper we discuss a state-of-the-art system for black–box optimization developed within Google, called *Google Vizier*, named after a high official who offers advice to rulers. It is a service for black-box optimization that supports several advanced algorithms. The system has a convenient Remote Procedure Call (RPC) interface, along with a dashboard and analysis tools. Google Vizier is a research project, parts of which supply core capabilities to our Cloud Machine Learning *HyperTune*[1] subsystem. We discuss the architecture of the system, design choices, and some of the algorithms used.

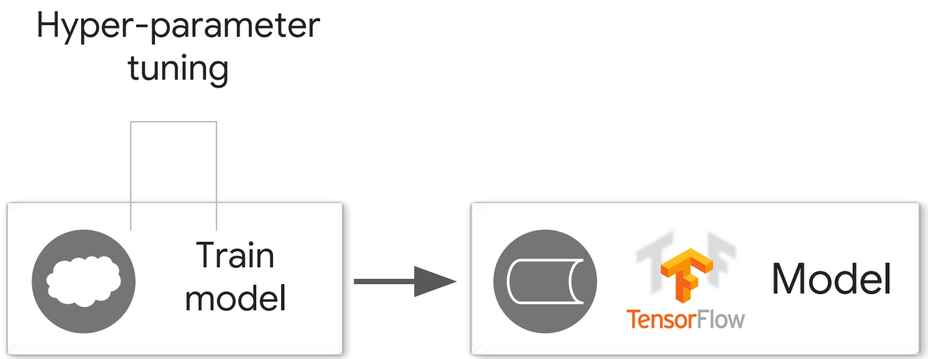
1.1 Related Work

Black–box optimization makes minimal assumptions about the problem under consideration, and thus is broadly appli-

Source: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/46180.pdf

How to use Cloud ML Engine for hyperparameter tuning

1. Make the parameter a command-line argument
2. Make sure outputs don't clobber each other
3. Supply hyperparameters to training job



1. Make the hyperparameters as command-line arguments

```
parser.add_argument(
    '--nbuckets',
    help = 'Number of buckets into which to discretize lats and lons',
    default = 10,
    type = int
)
parser.add_argument(
    '--hidden_units',
    help = 'List of hidden layer sizes to use for DNN feature columns',
    nargs = '+',
    default = [128, 32, 4]
)
```

1. Make the hyperparameters as command-line arguments

```
parser.add_argument(
    '--nbuckets',
    help = 'Number of buckets into which to discretize lats and lons',
    default = 10,
    type = int
)
parser.add_argument(
    '--hidden_units',
    help = 'List of hidden layer sizes to use for DNN feature columns',
    nargs = '+',
    default = [128, 32, 4]
)
```

1. Make the hyperparameters as command-line arguments

```
parser.add_argument(
    '--nbuckets',
    help = 'Number of buckets into which to discretize lats and lons',
    default = 10,
    type = int
)
parser.add_argument(
    '--hidden_units',
    help = 'List of hidden layer sizes to use for DNN feature columns',
    nargs = '+',
    default = [128, 32, 4]
)
```

2. Make sure that outputs don't clobber each other

```
output_dir = os.path.join(
     output_dir,
     json.loads(
         os.environ.get('TF_CONFIG', '{}')
     ).get('task', {}).get('trial', '')
 )
```

Buckets / cloud-training-demos-ml / taxifare / ch4 / taxi_trained / 10

Name	Size
checkpoint	132 B
eval/	–
events.out.tfevents.1488250047.master-2d5cef50bf-0-...	3.25 MB
export/	–
graph.pbtxt	1.47 MB
model.ckpt-0.data-00000-of-00003	9.28 MB
model.ckpt-0.data-00001-of-00003	532.07 KB

2. Make sure that outputs don't clobber each other

```
output_dir = os.path.join(
     output_dir,
     json.loads(
         os.environ.get('TF_CONFIG', '{}')
     ).get('task', {}).get('trial', '')
 )
```

Buckets / cloud-training-demos-ml / taxifare / ch4 / taxi_trained / 10

Name	Size
checkpoint	132 B
eval/	–
events.out.tfevents.1488250047.master-2d5cef50bf-0-...	3.25 MB
export/	–
graph.pbtxt	1.47 MB
model.ckpt-0.data-00000-of-00003	9.28 MB
model.ckpt-0.data-00001-of-00003	532.07 KB

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ... \
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

3. Supply hyperparameters to training job

```
%writefile hyperparam.yaml
trainingInput:
  scaleTier: STANDARD_1
  hyperparameters:
    goal: MINIMIZE
    hyperparameterMetricTag: rmse
    maxTrials: 30
    maxParallelTrials: 1
    params:
    - parameterName: train_batch_size
      type: INTEGER
      minValue: 64
      maxValue: 512
      scaleType: UNIT_LOG_SCALE
    - parameterName: nbuckets
      type: INTEGER
      minValue: 10
      maxValue: 20
      scaleType: UNIT_LINEAR_SCALE
    - parameterName: hidden_units
      type: CATEGORICAL
      categoricalValues: ["128 64 32", "256 128 16", "512 128 64"]
```

```
gcloud ml-engine jobs submit training $JOBNAME \
   --region=$REGION \
   --module-name=trainer.task \
   ...
   --config=hyperparam.yaml \
   -- \
   --output_dir=$OUTDIR \
   --num_epochs=100
```

Lab

Improve model accuracy by Hyperparameter Tuning with Cloud MLE

Fereshteh Mahvar

Lab

Lab Solution: Improve model accuracy by Hyperparameter Tuning with Cloud MLE

Fereshteh Mahvar