Google Cloud

Partner Certification Academy

Professional Machine Learning Engineer

We will begin in:

<<15:00->>

The information in this presentation is classified:

Google confidential & proprietary

⚠️ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

Source Materials

Some of this program's content has been sourced from the following resources:

- Google Cloud certification site
- Google Cloud documentation
- Google Cloud console
- Google Cloud courses and workshops
- Google Cloud white papers
- Google Cloud Blog
- Google Cloud YouTube channel
- Google Cloud samples
- Google codelabs
- Google Cloud partner-exclusive resources

Google Cloud

Google Cloud Skills Boost for Partners

- Google Cloud Fundamentals: Core Infrastructure
- Logging, Monitoring and Observability in Google Cloud

Google Cloud
Partner Advantage

- Identity Management Technical Deep Dive
- Access Management Technical Deep Dive
- Cloud Foundations: Cost Control Technical Deep Dive [PSO Y22]

Session Logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.

- These slides are available in the Student Lecture section of your Qwiklabs classroom.

- The session is **not recorded.**

- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Google Cloud Partner Learning Programs

- Partner Certification Academy
- Partner Delivery Readiness Index (DRI)
- Cloud Skills Boost for Partners
- Partner Advantage

Learner Commitment

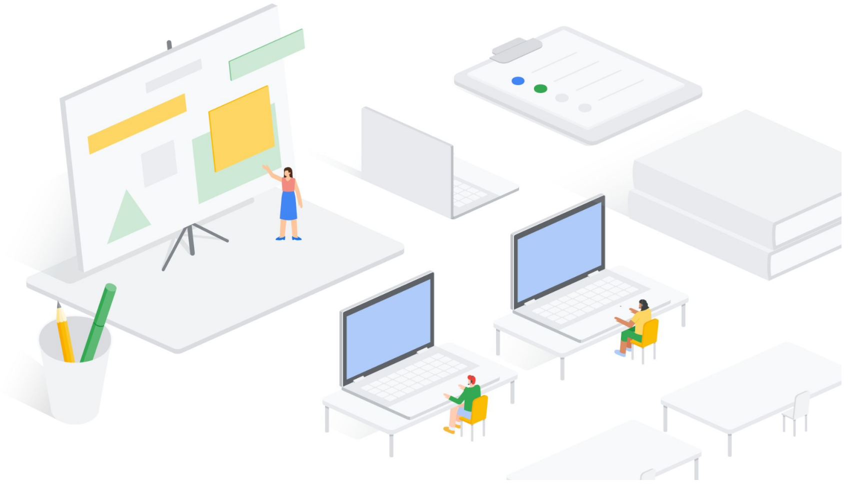
Each week, learners are to complete the learning path's course content, Cloud Skills Boost for Partner Quests/Challenge Labs and material that the mentor has recommended that will support learning.

- **Workshop Day:** Meet for the cohort's weekly 'general session'. **(≅ 2 hours)**
- **During the week:** Complete the week's course, perform hands-on labs, review any additional material suggested material for the week. **(≅ 8 - 16 hours)**

➔ **Important:** Learners must allocate time between each weekly session to study and familiarize themselves with any prerequisite knowledge they may lack. It is also recommended that learners complete the next week's course prior to the scheduled workshop.

Path to Service Excellence

Certification



Advanced Solutions Training

Partner
DRI

Delivery Readiness Index

Benchmark your skills with DRI

Partner DRI

Assess: Partner Proficiency and Delivery Capability
Benchmark Partner individuals, project teams and practices GCP capabilities

Analyze: Individual Partner Consultants’ GCP Readiness
Showcase Partner individuals GCP knowledge, skills, and experience

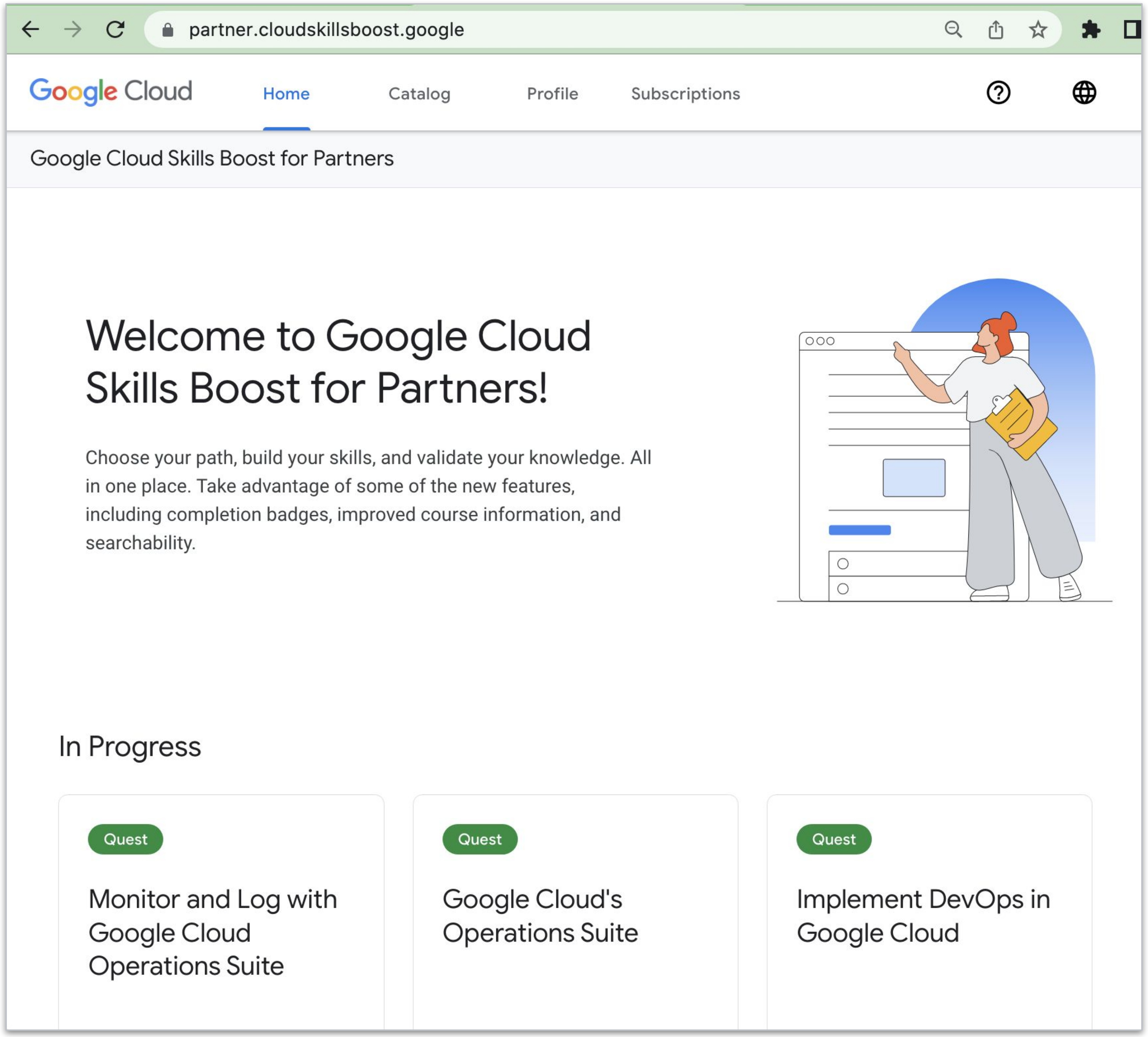
Advise: Google Assurance for Partner Delivery
Packaged offerings to bridge specific capability gaps

Action: Tailored L&D Plan for Account Based Enablement
Personalized learning & development recommendations per individual consultant

Google Cloud Skills Boost for Partners

https://partner.cloudskillsboost.google/

- On-demand course content
- Hands-on labs
- Skill Badges
- **FREE** to Google Cloud Partners!



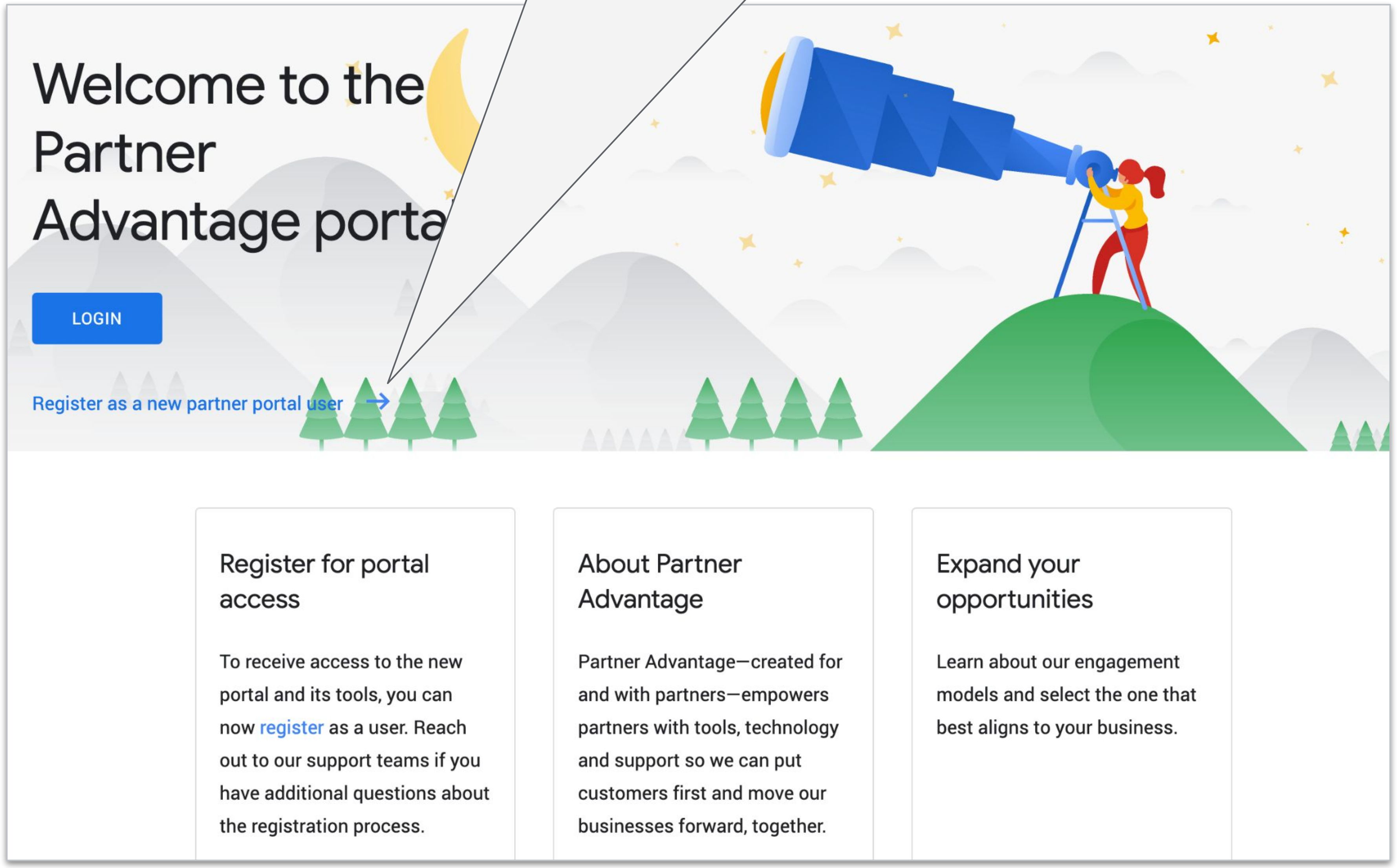
Google Cloud Partner Advantage

- Resources for Google Cloud partner organizations:
  - Recent announcements
  - Solutions/role-based training
  - Live/pre-recorded webinars on various topics
    - Partner Advantage Live Webinars

- Complements the certification self-study material presented on Google Cloud Skills Boost for Partners

- Helpful Links:
  - Getting started on Partner Advantage
  - Join Partner Advantage
  - Get help accessing Partner Advantage

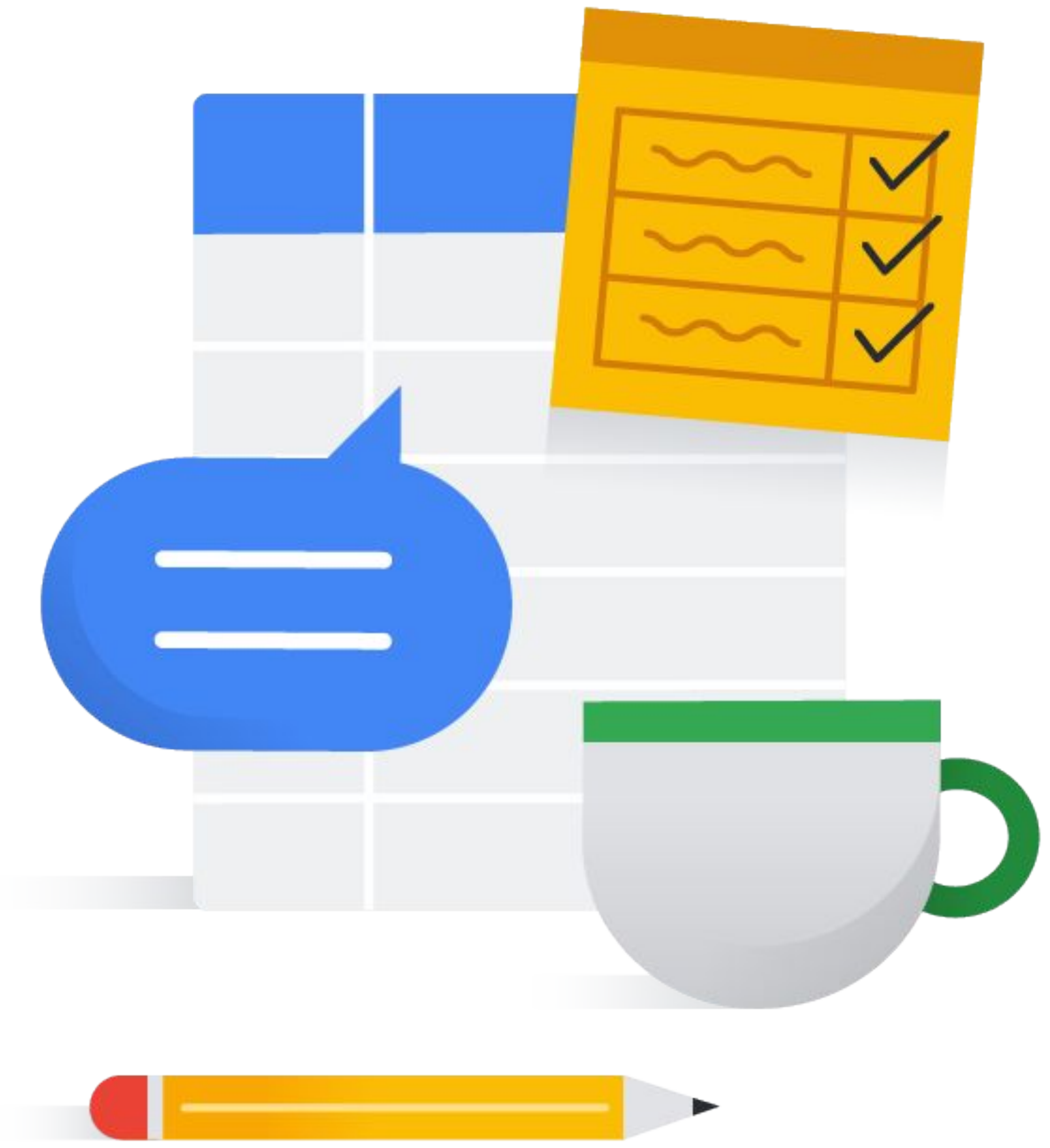
Create a login using your company email. Your organization must verify your request prior to granting you access.



https://www.partneradvantage.goog

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



Data Ingestion, Feature Engineering and Distributed Training with TensorFlow

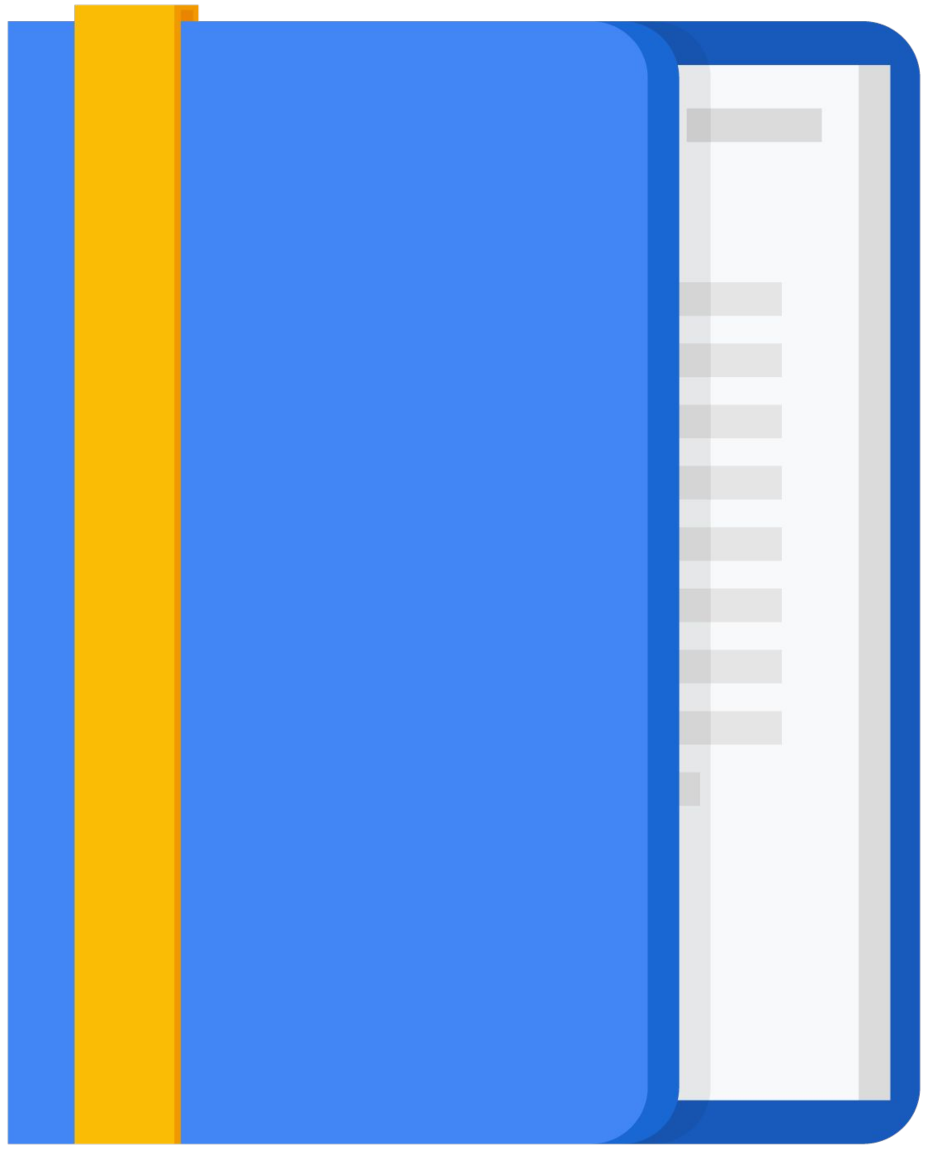
Instructor: Ben Ahmed

Course Agenda

Optimize Data Input Pipeline with TensorFlow

Feature Engineering with TFT

Distributing Strategies with TF

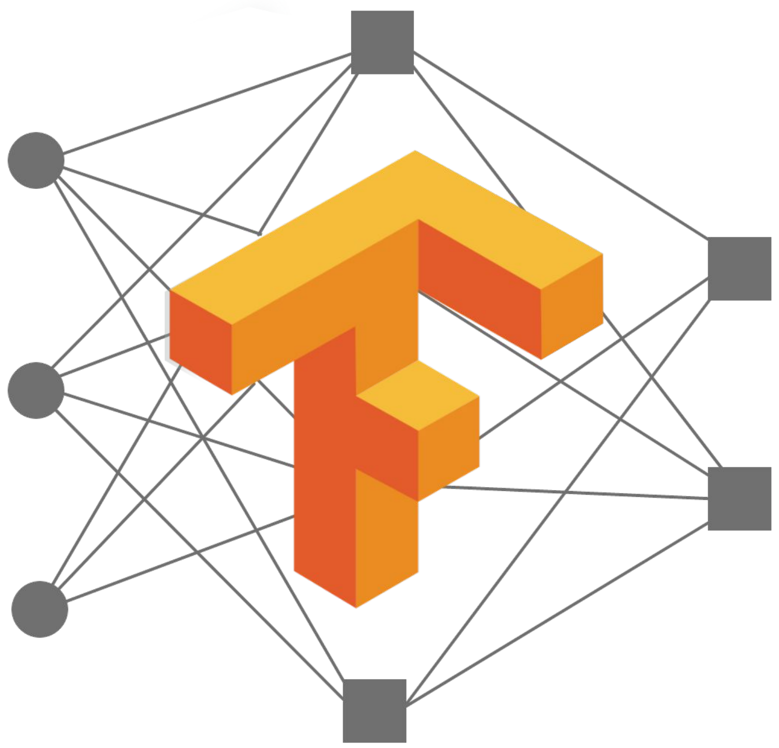


Performing a Training Step Involves:

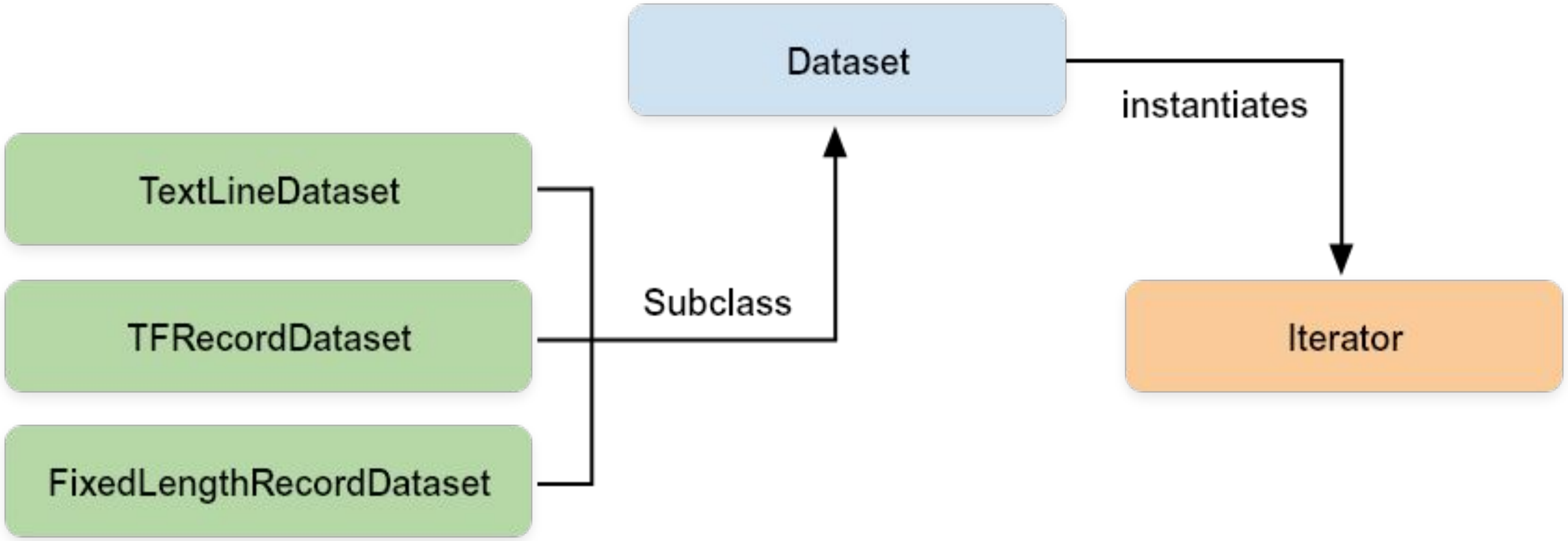
1. Opening a file (if it isn't open already)
2. Fetching a data entry from the file
3. Using the data for training

tf.data API

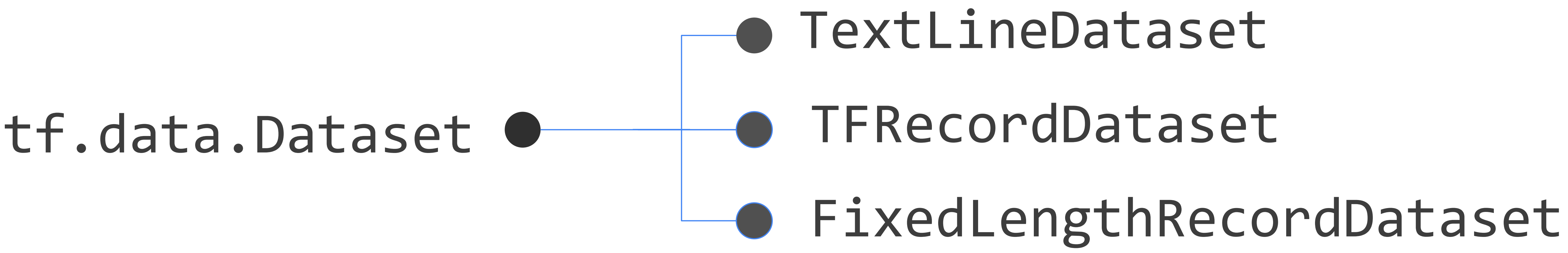
1. Build complex input pipelines from simple, reusable pieces.
2. Build pipelines for multiple data types.
3. Handle large amounts of data; perform complex transformations.



Multiple ways to feed TensorFlow models with data



Datasets can be created from different file formats.



A tf.data.Dataset allows you to

- Create data pipelines from
  - in-memory dictionary and lists of tensors
  - out-of-memory sharded data files

- Preprocess data in parallel (and cache result of costly operations)

```
dataset = dataset.map(preproc_fun).cache()
```

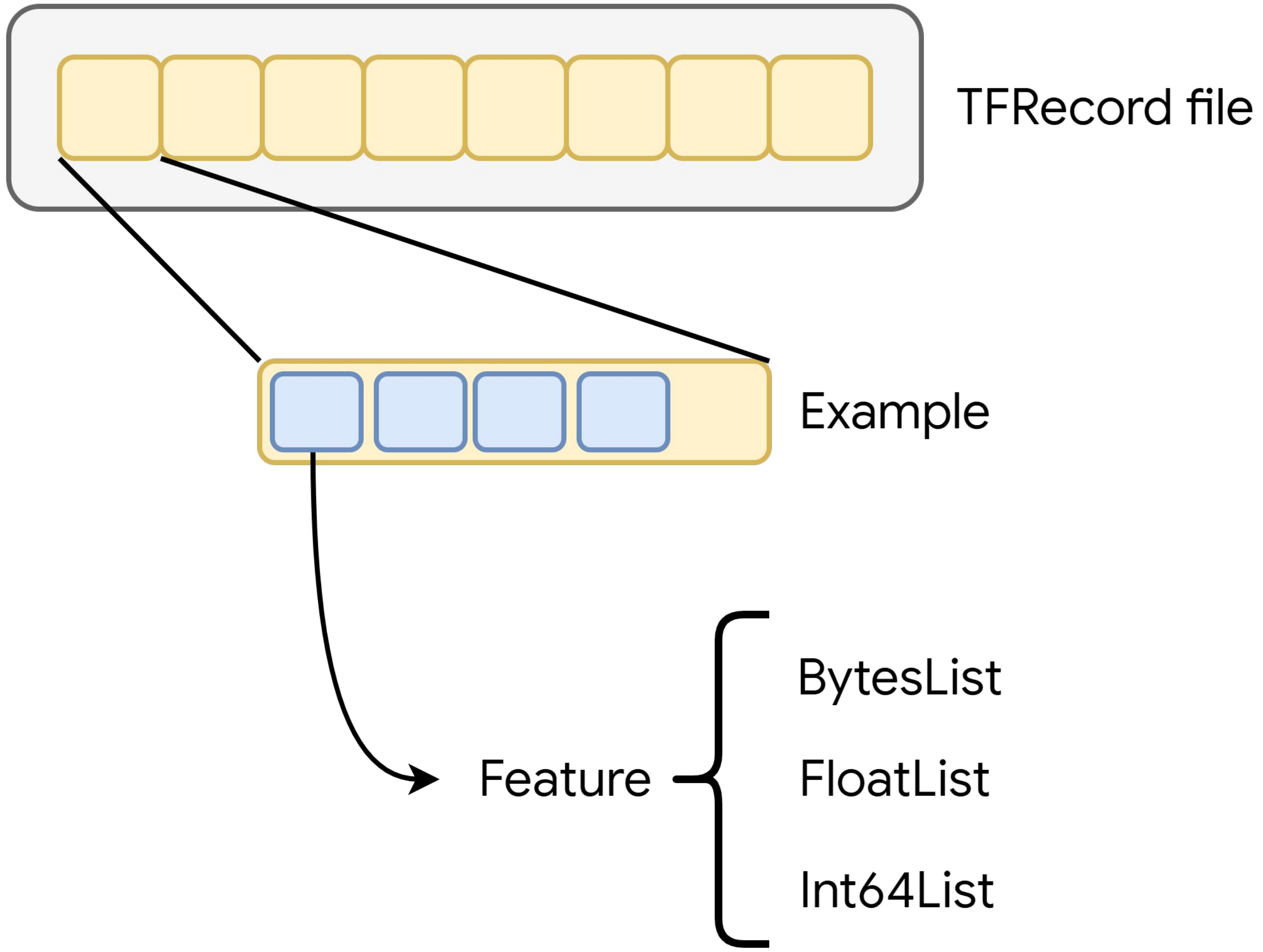
- Configure the way the data is fed into a model with a number of chaining methods

```
dataset = dataset.shuffle(1000).repeat(epochs).batch(batch_size, drop_remainder=True)
```

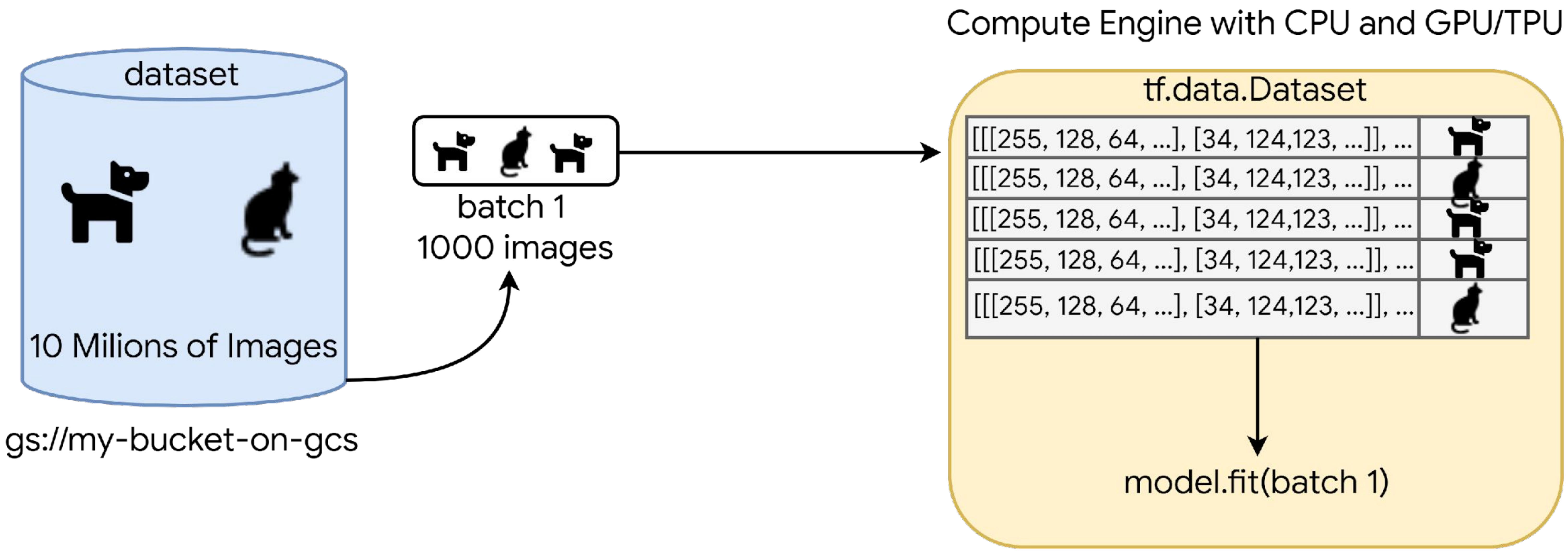
in a easy and very compact way

TFRecord data structure

- Data is stored as sequence of samples that has the name of Example.
- Example contains a list of Features
- Features can have different tf.dtype:
 BytesList, FloatList, Int64List
- You must convert your data into Features which are inner components of an Example
- Main advantages of using TFRecords:
 Fast streaming as it is a sequence of bytes
 Easy of batching, shuffling, caching etc
 Easy to distribute the dataset

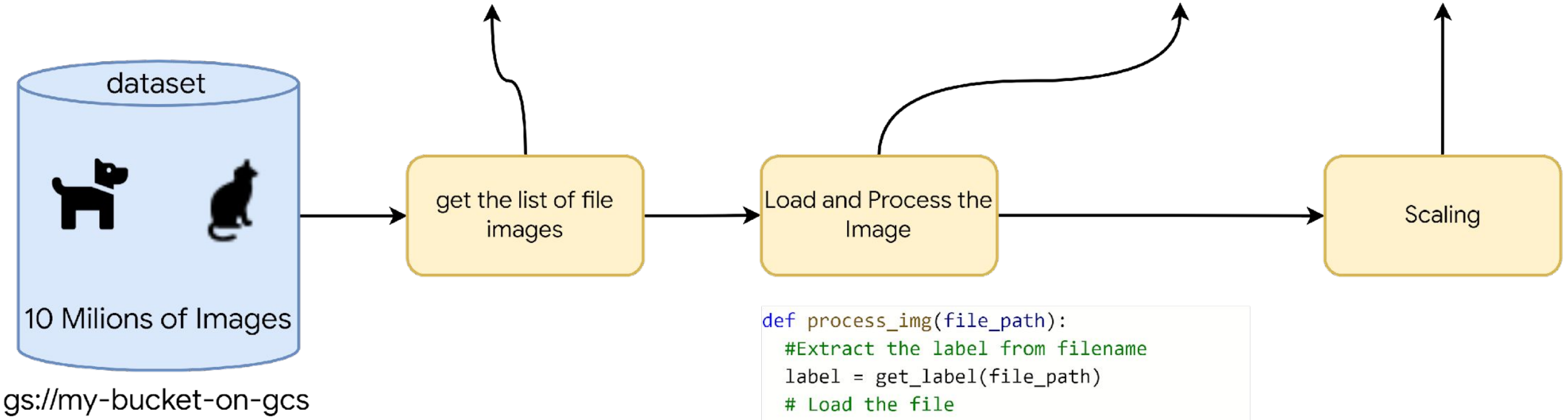


Introducing tf.data API



What’s the matter with this pipeline?

ds = tf.data.Dataset.list_files("gs://my-bucket-on-gcs/images/*").map(process_img).map(lambda x: x/255)




```
def process_img(file_path):
  #Extract the label from filename
  label = get_label(file_path)
  # Load the file
  image = tf.io.read_file(file_path)
  #Decode the image, jpeg in this case
  image = tf.image.decode_jpeg(image)
  # Resize the image
  image = tf.image.resize(image, [128,128])
  return image, label
```

In reality an Optimized Input Pipeline includes…

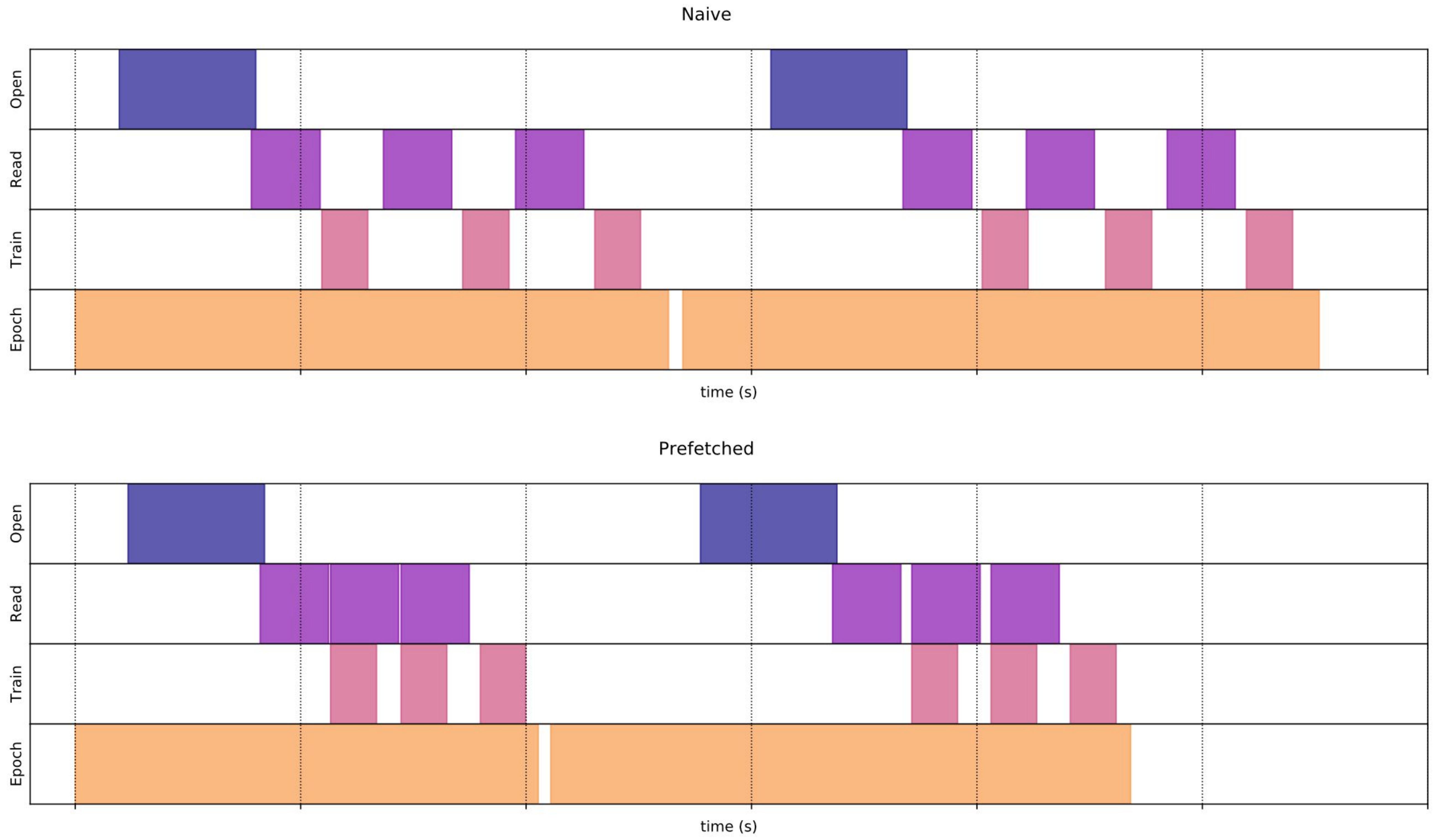
```
ds = tf.data.Dataset.list_files("gs://my-bucket-on-gcs/images/*")
ds = ds.interleave(load_function, num_parallel_calls=tf.data.AUTOTUNE)
ds = ds.batch(BATCH_SIZE)
ds = ds.map(process_img, num_parallel_calls=tf.data.AUTOTUNE)
ds = ds.cache()
ds = ds.prefetch(tf.data.AUTOTUNE)
```

```
ds = tf.data.Dataset.list_files("...").interleave(...).batch(...).map(num_parallel_calls=...).cache().prefetch(...)
```

Best practice in TF pipeline performance optimization

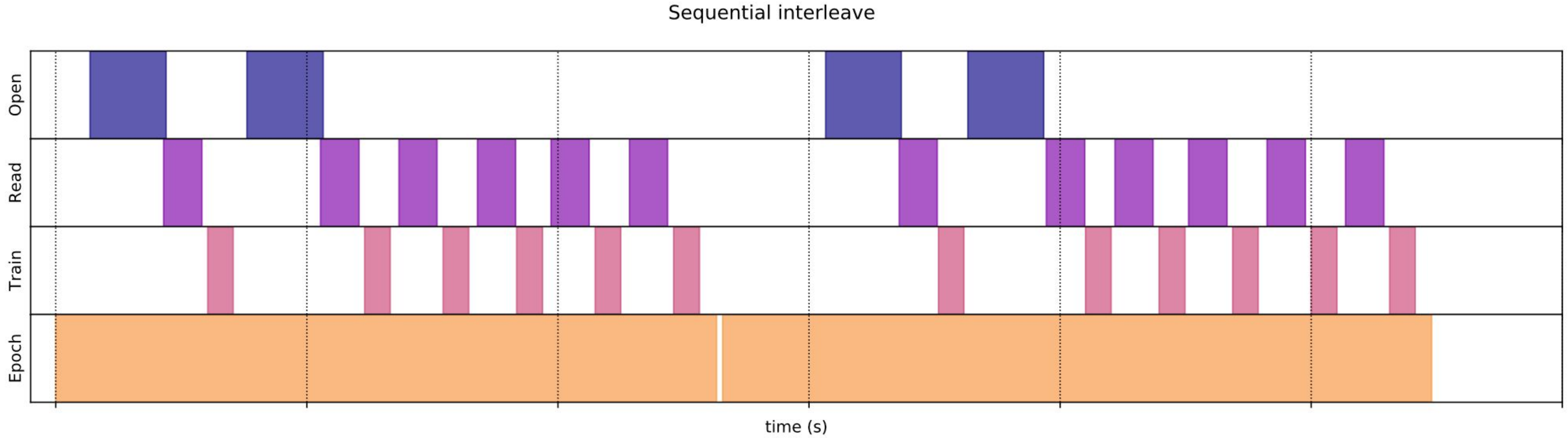
- Prefetch the data while doing training
- Parallelize data extraction from multiple (remote) datasets using interleaving
- Parallelize data transformation (map) using num_parallel_calls
- Cache transformation during the first epoch
- Batch inputs before doing transformation

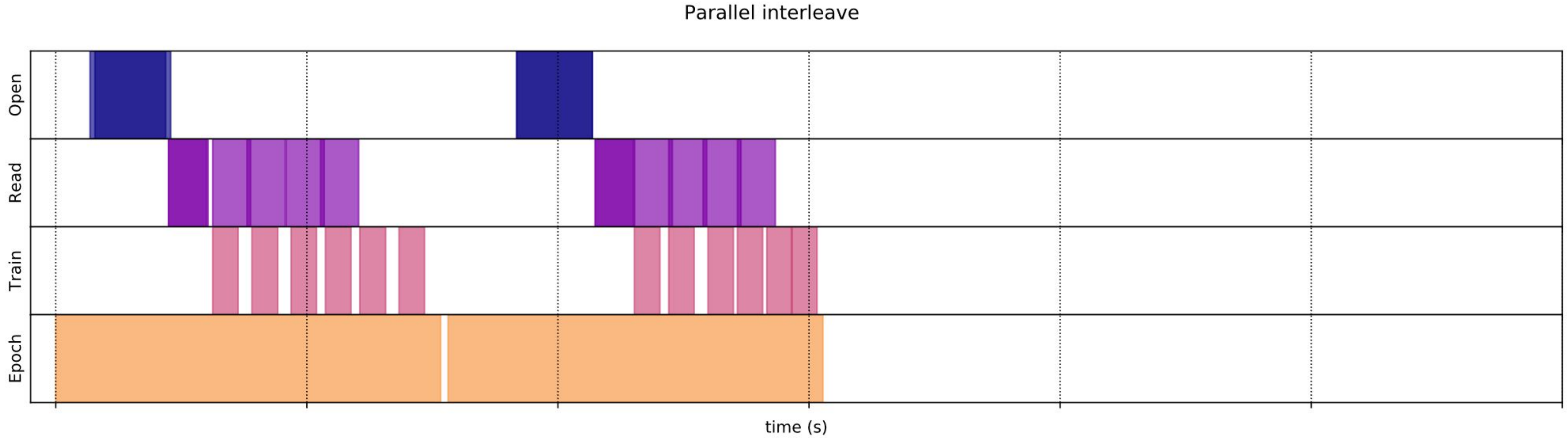
Prefetching



Google Cloud

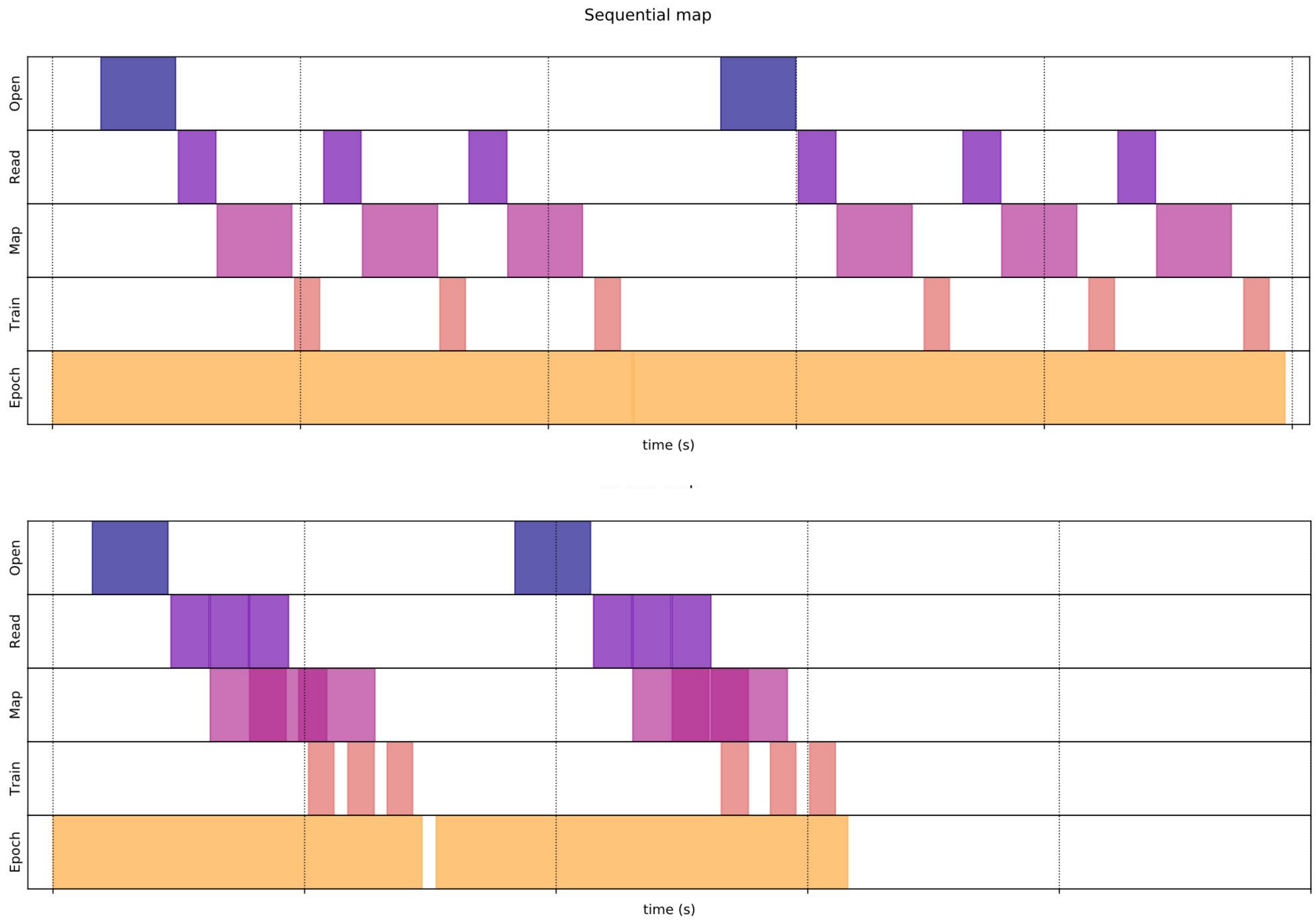
Interleaving





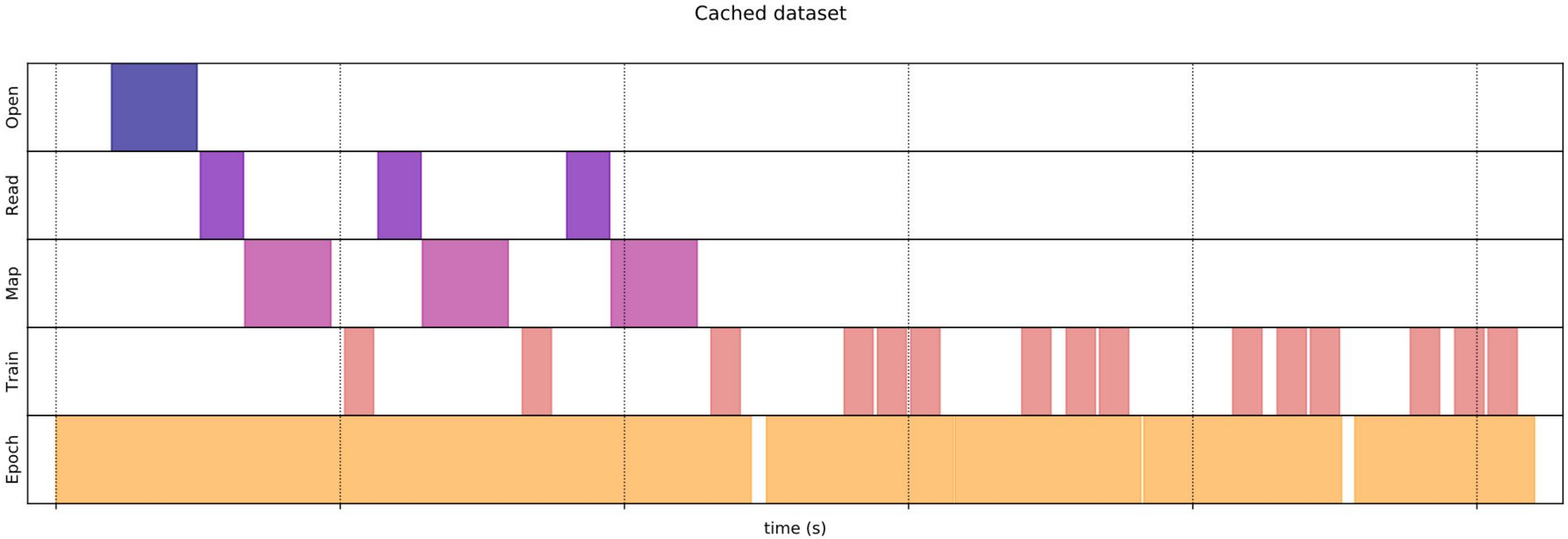
Google Cloud

Parallelize Transformations



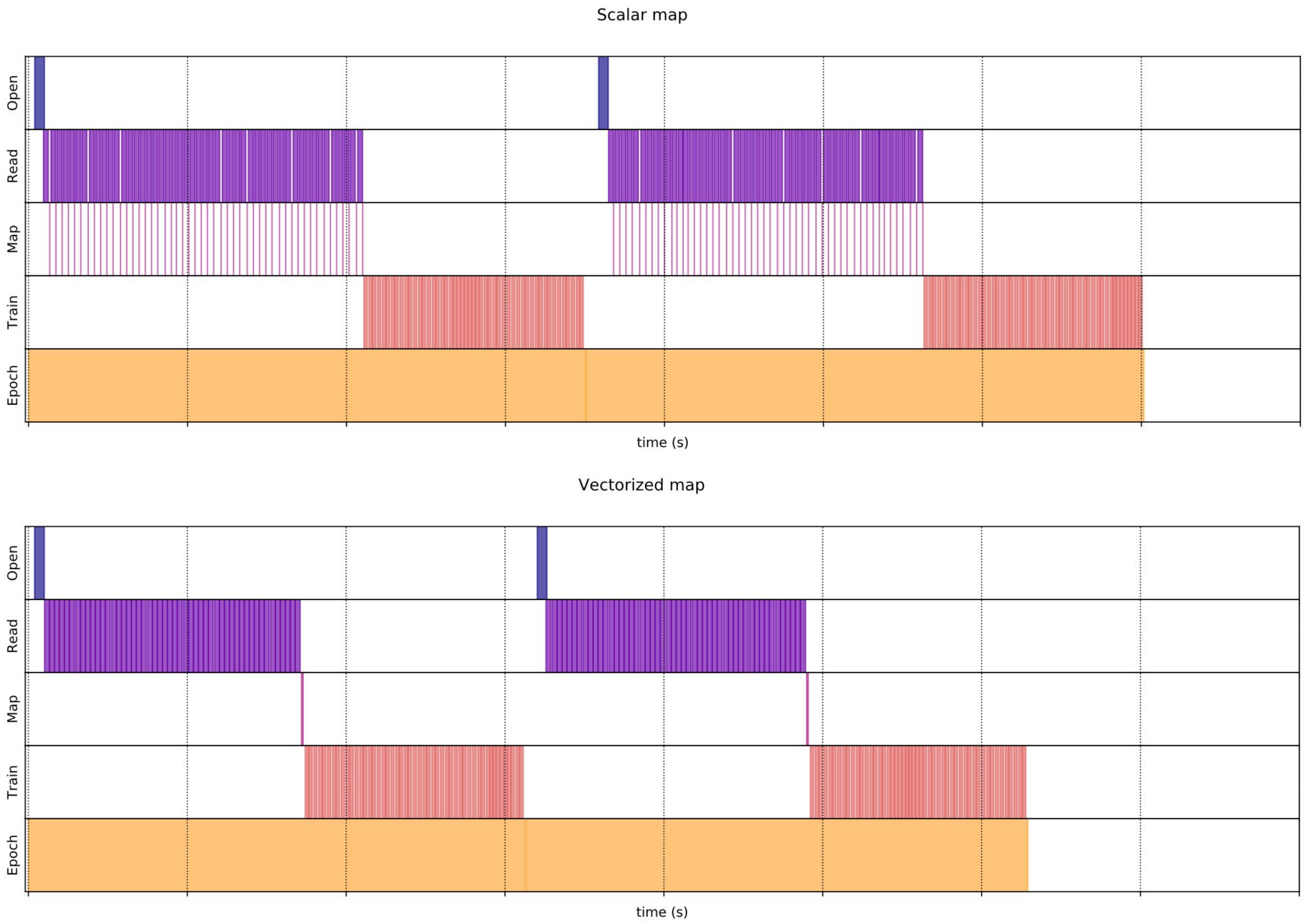
Google Cloud

Caching



Note: use caching only for those map() that are expensive and can fit into memory or local storage.

Batching


Scalar map
Open
Read
Map
Train
Epoch
time (s)
Vectorized map
Open
Read
Map
Train
Epoch
time (s)

Optimizing the input Data Pipeline in TF

```
ds = tf.data.Dataset.list_files("gs://my-bucket-on-gcs/images/*")
ds = ds.interleave(load_function, num_parallel_calls=tf.data.AUTOTUNE)
ds = ds.batch(BATCH_SIZE)
ds = ds.map(process_img, num_parallel_calls=tf.data.AUTOTUNE)
ds = ds.cache()
ds = ds.prefetch(tf.data.AUTOTUNE)
```

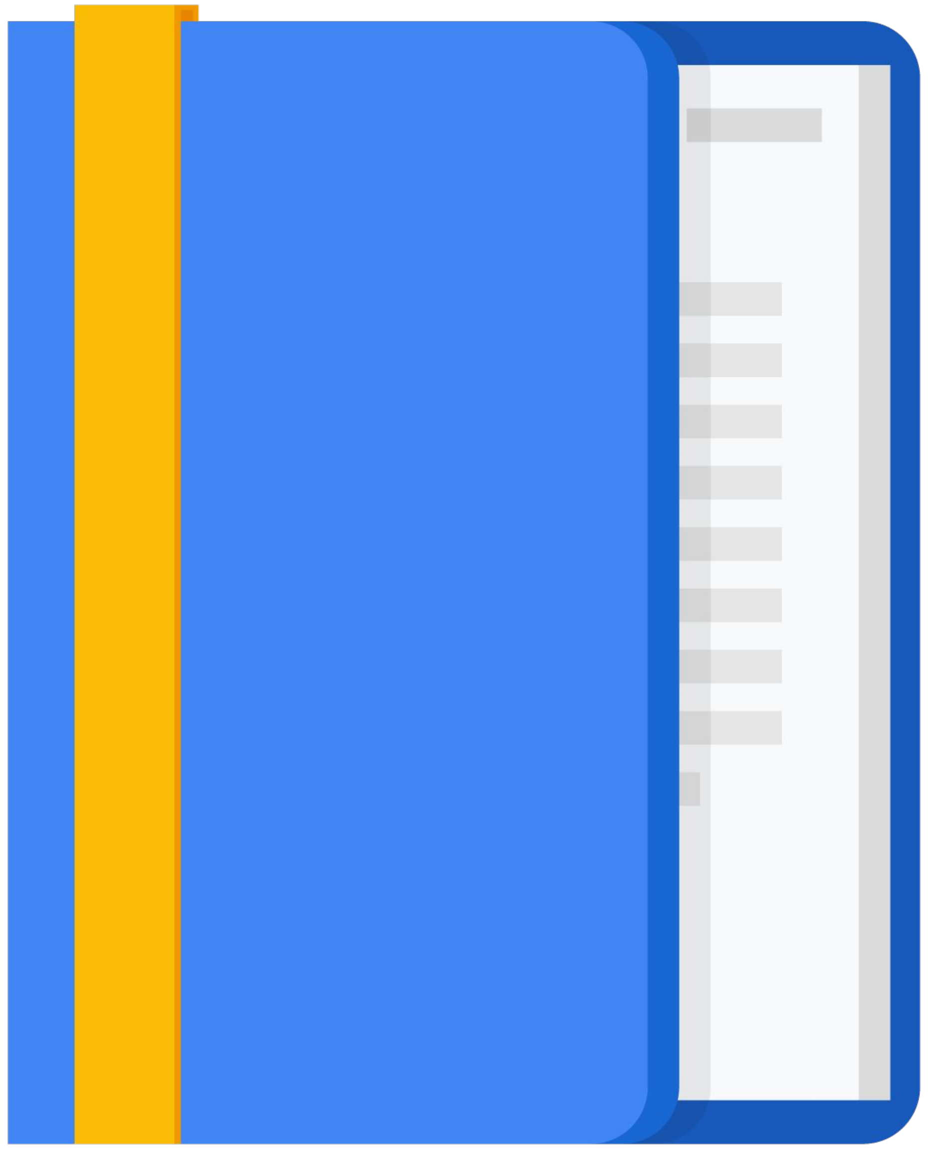
```
ds = tf.data.Dataset.list_files("...").interleave(...).batch(...).map(num_parallel_calls=...).cache().prefetch(...)
```

Course Agenda

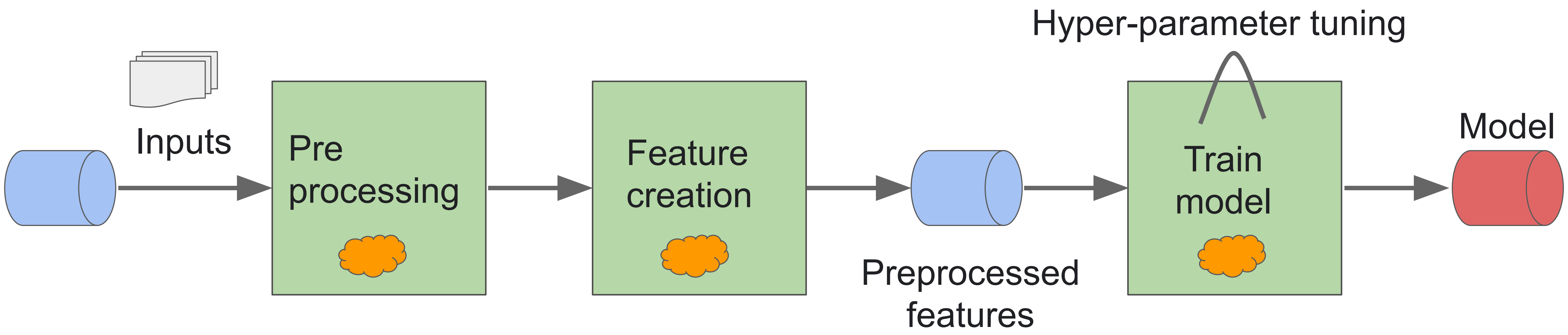
Optimize Data Input Pipeline with TensorFlow

Feature Engineering with TFT

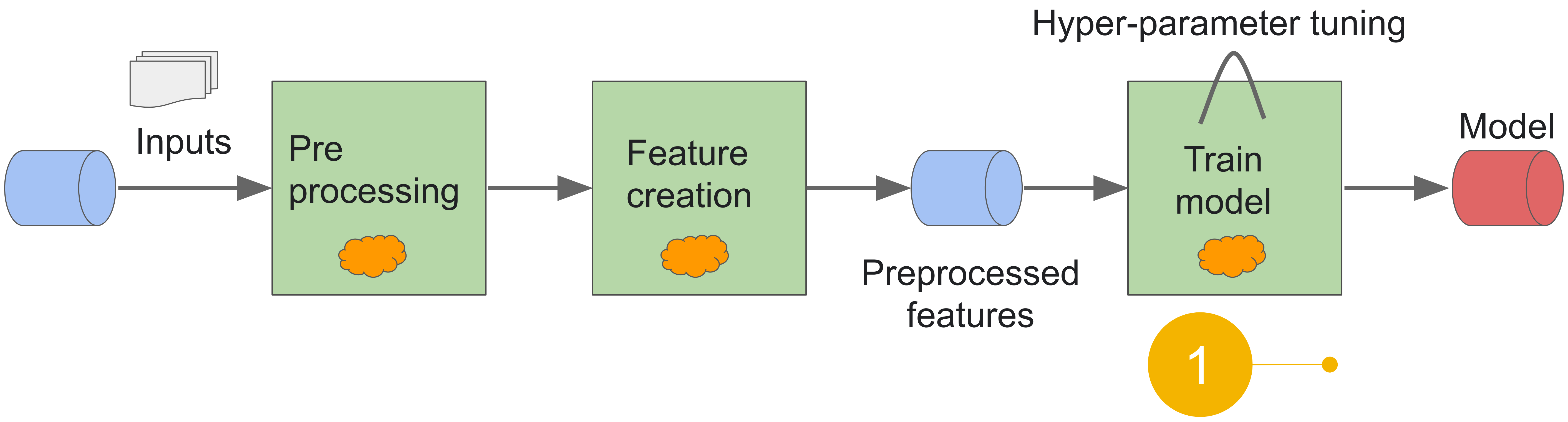
Distributing Strategies with TF



There are three possible places to do feature engineering, each of which has its pros and cons



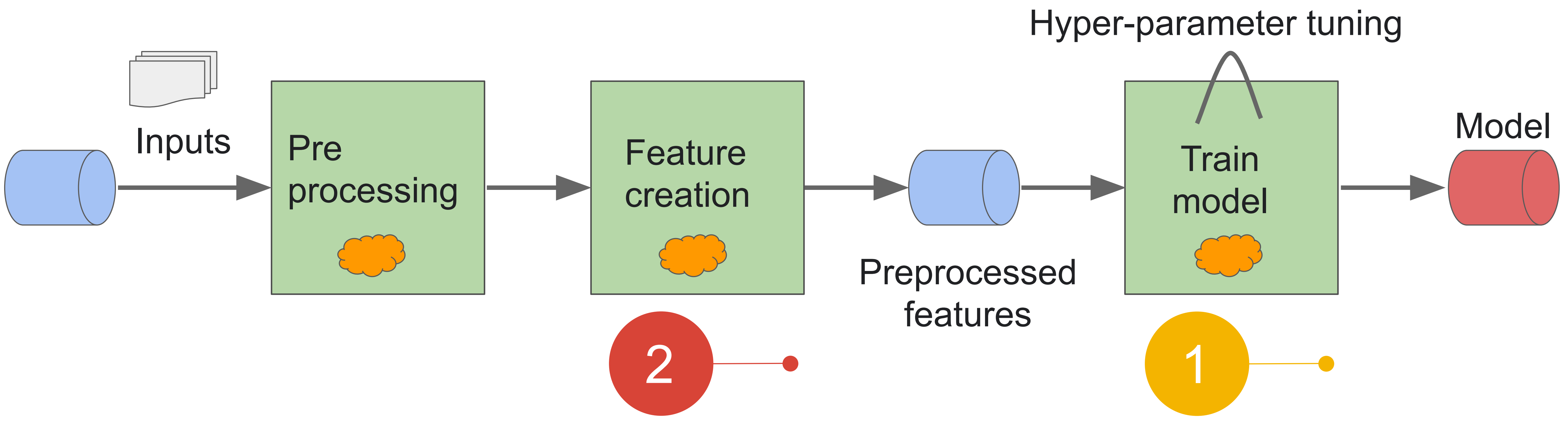
There are three possible places to do feature engineering, each of which has its pros and cons



TensorFlow*
*efficient
*tf methods only
*this input only

Google Cloud

There are three possible places to do feature engineering, each of which has its pros and cons

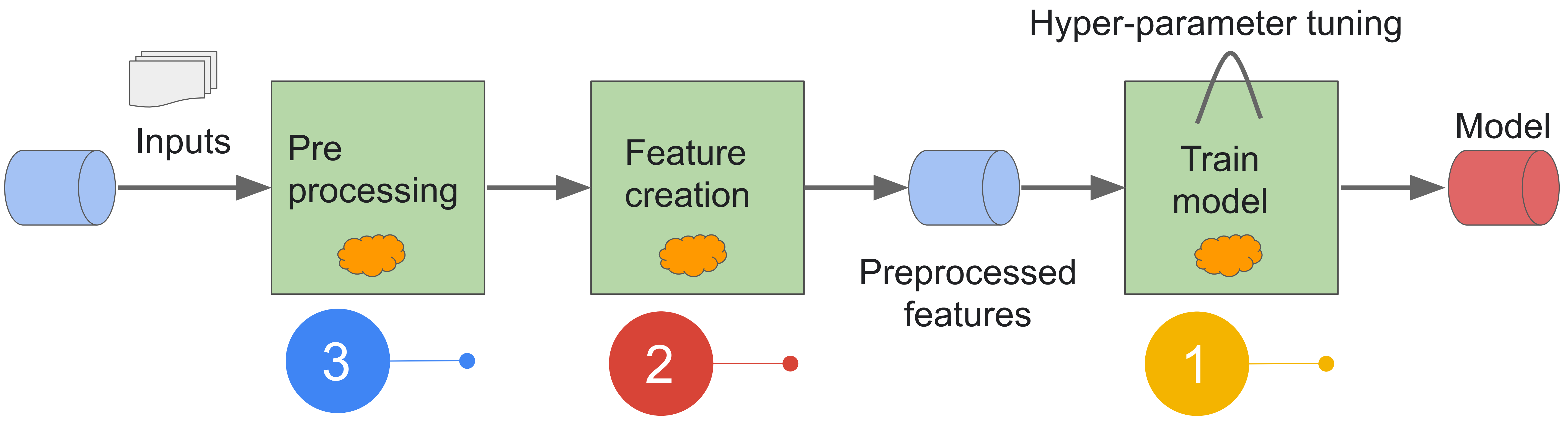


Beam/Dataflow*
*in a pipeline
*Python/Java code
*time-windows

TensorFlow*
*efficient
*tf methods only
*this input only

Google Cloud

There are three possible places to do feature engineering, each of which has its pros and cons

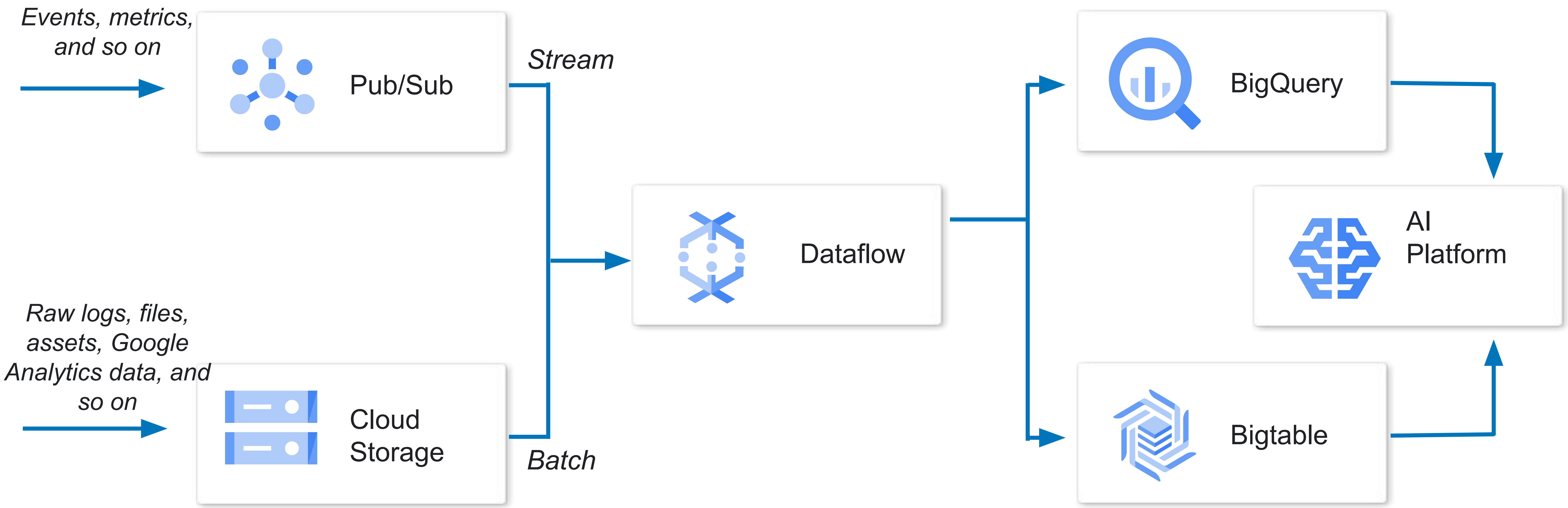


tf.transform*
*efficient
*tf methods only
*aggregates

Beam/Dataflow*
*in a pipeline
*Python/Java code
*time-windows

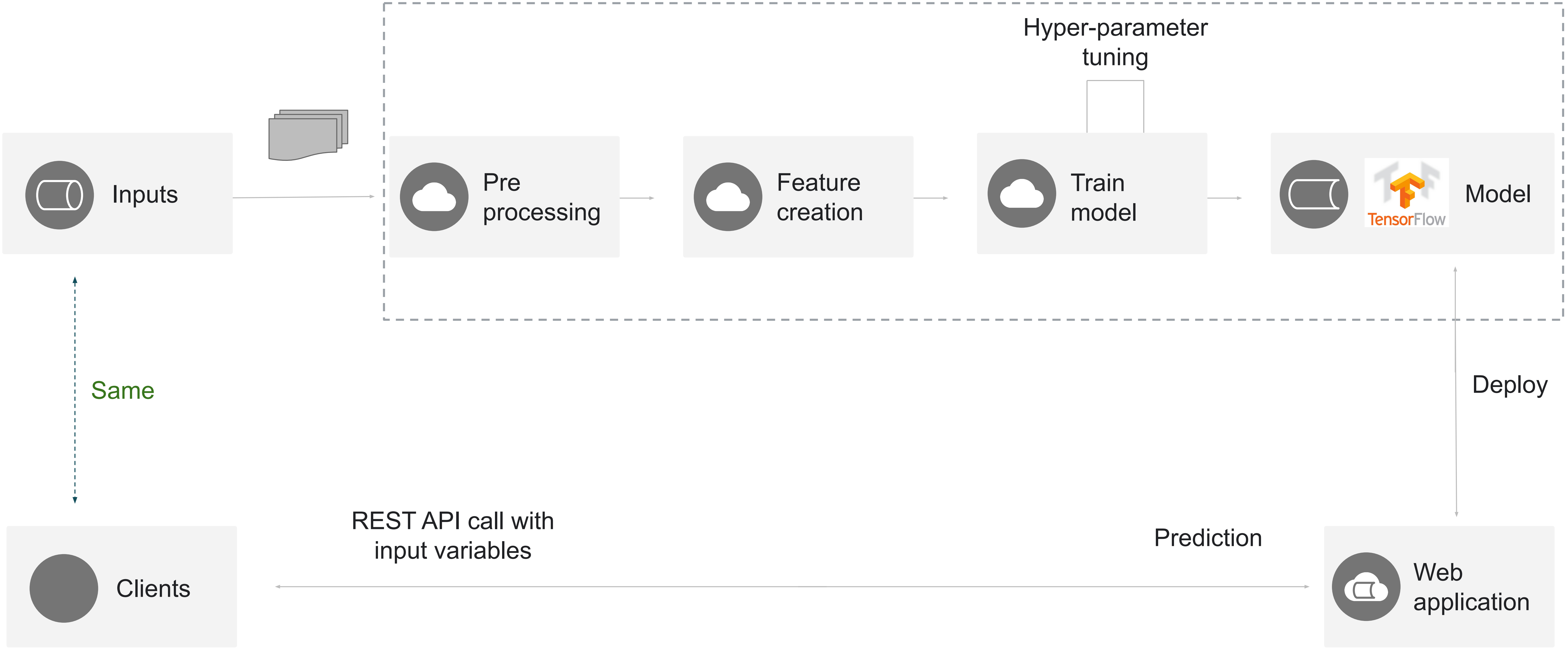
TensorFlow*
*efficient
*tf methods only
*this input only

Beam/Dataflow preprocessing works in the context of a pipeline



Google Cloud

TensorFlow is good for on-demand, on-the-fly processing



Google Cloud

TensorFlow Transform: A part of TFX

Productionizing Machine Learning requires more than just a learning algorithm.

TFX is an end-to-end ML platform based on TensorFlow.

Shared Configuration | Data Visualization | Job Orchestration | Monitoring | Workflow Tools | ...

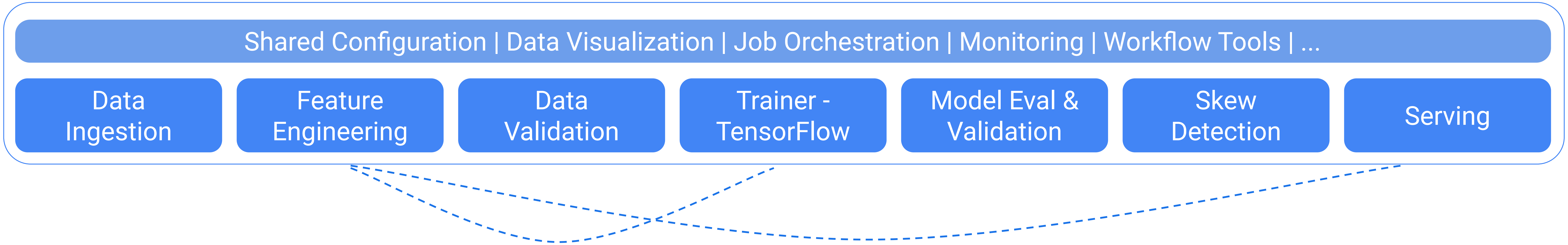
Data Ingestion | Feature Engineering | Data Validation | Trainer - TensorFlow | Model Eval & Validation | Skew Detection | Serving

Tf.Transform is the component used to **analyze** and **transform** training data

TensorFlow Transform: A part of TFX

Productionizing Machine Learning requires more than just a learning algorithm.

TFX is an end-to-end ML platform based on TensorFlow.

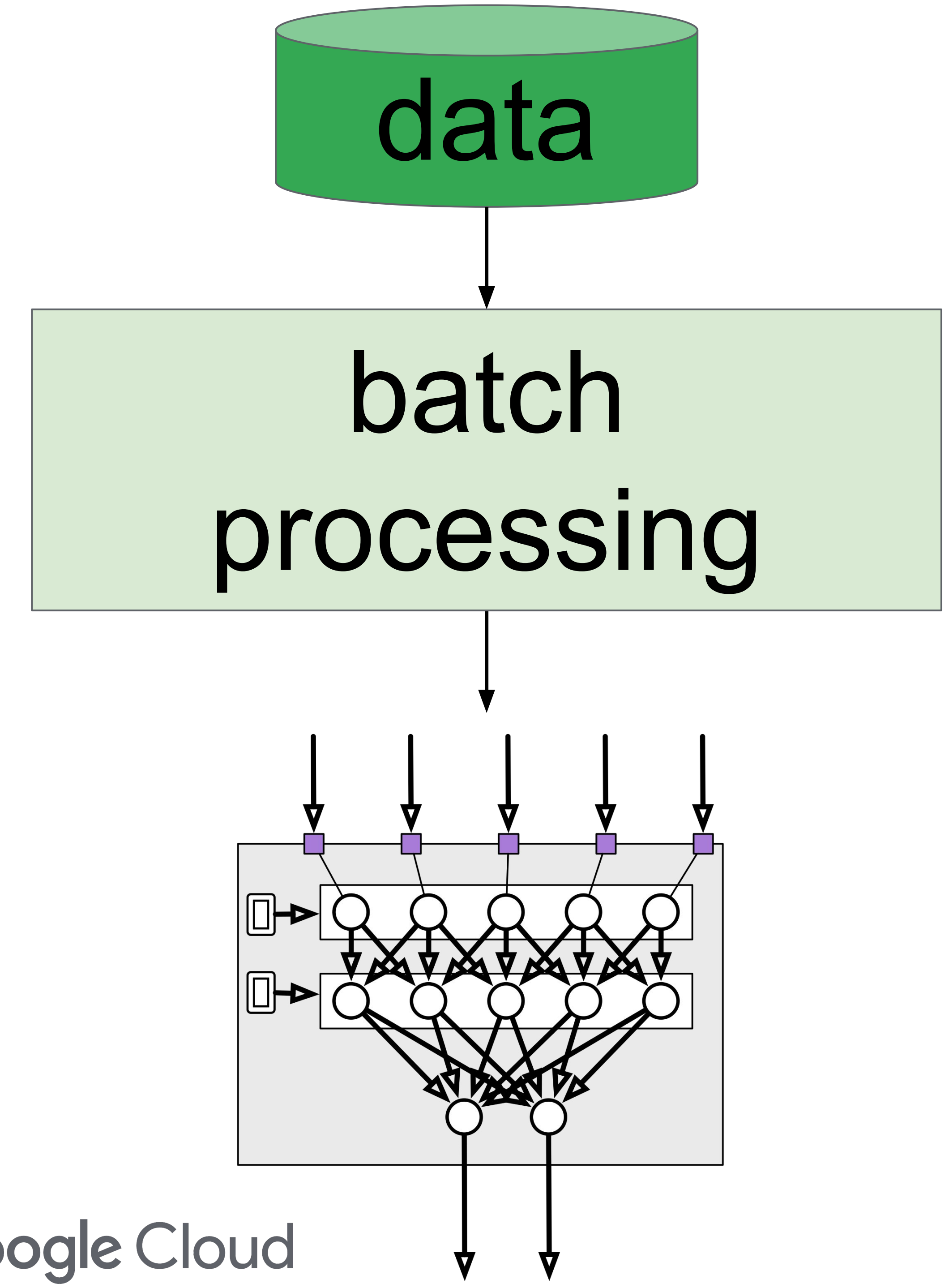


Artifacts produced by tf.Transform's are consumed at both **training** and **serving** time to avoid skew.

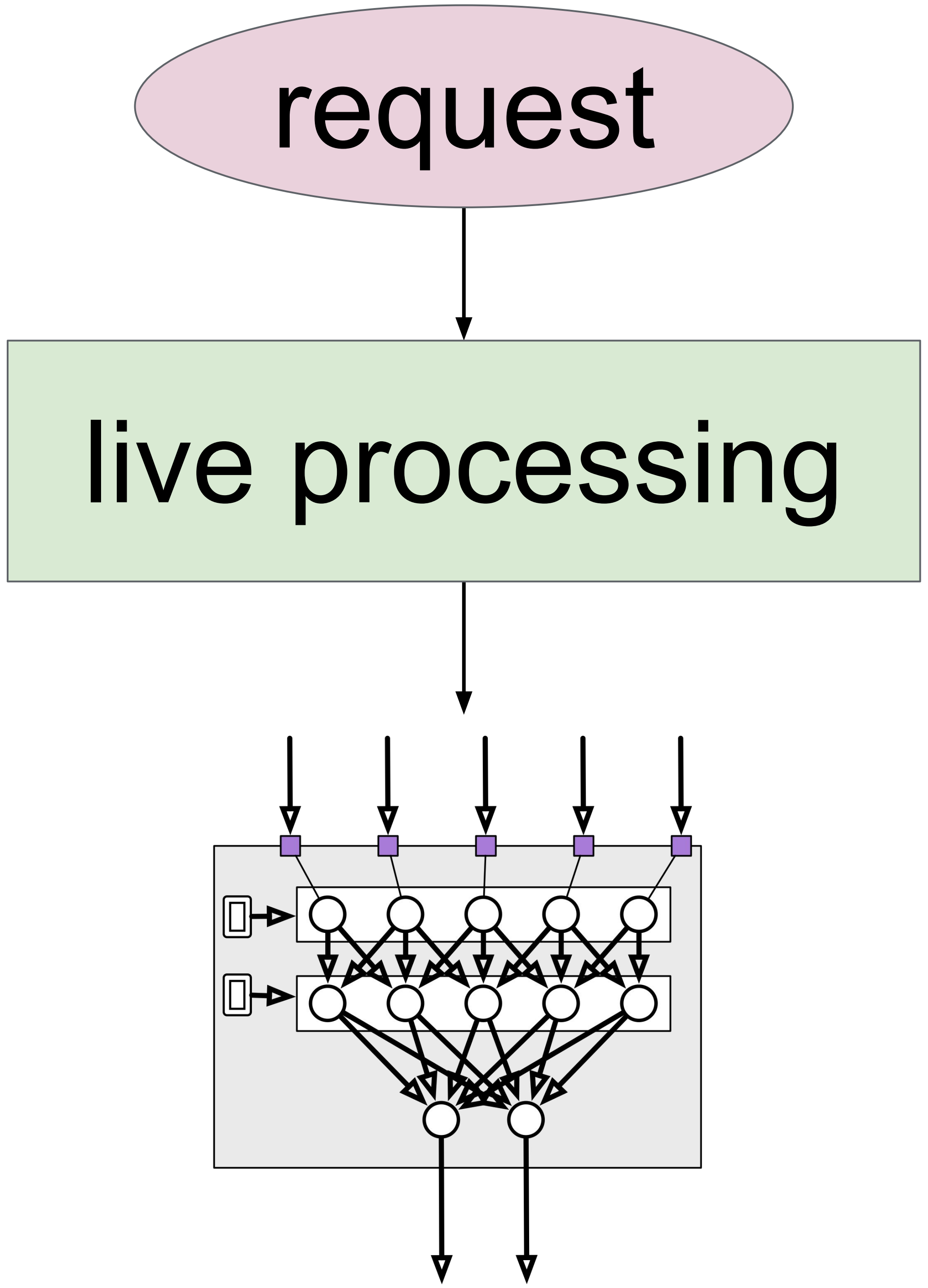
Google Cloud

Typical ML Pipeline

During training



During serving



Problems

- Need to keep batch and live processing in sync.
- All other tooling (e.g. evaluation) must also be kept in sync with batch processing.

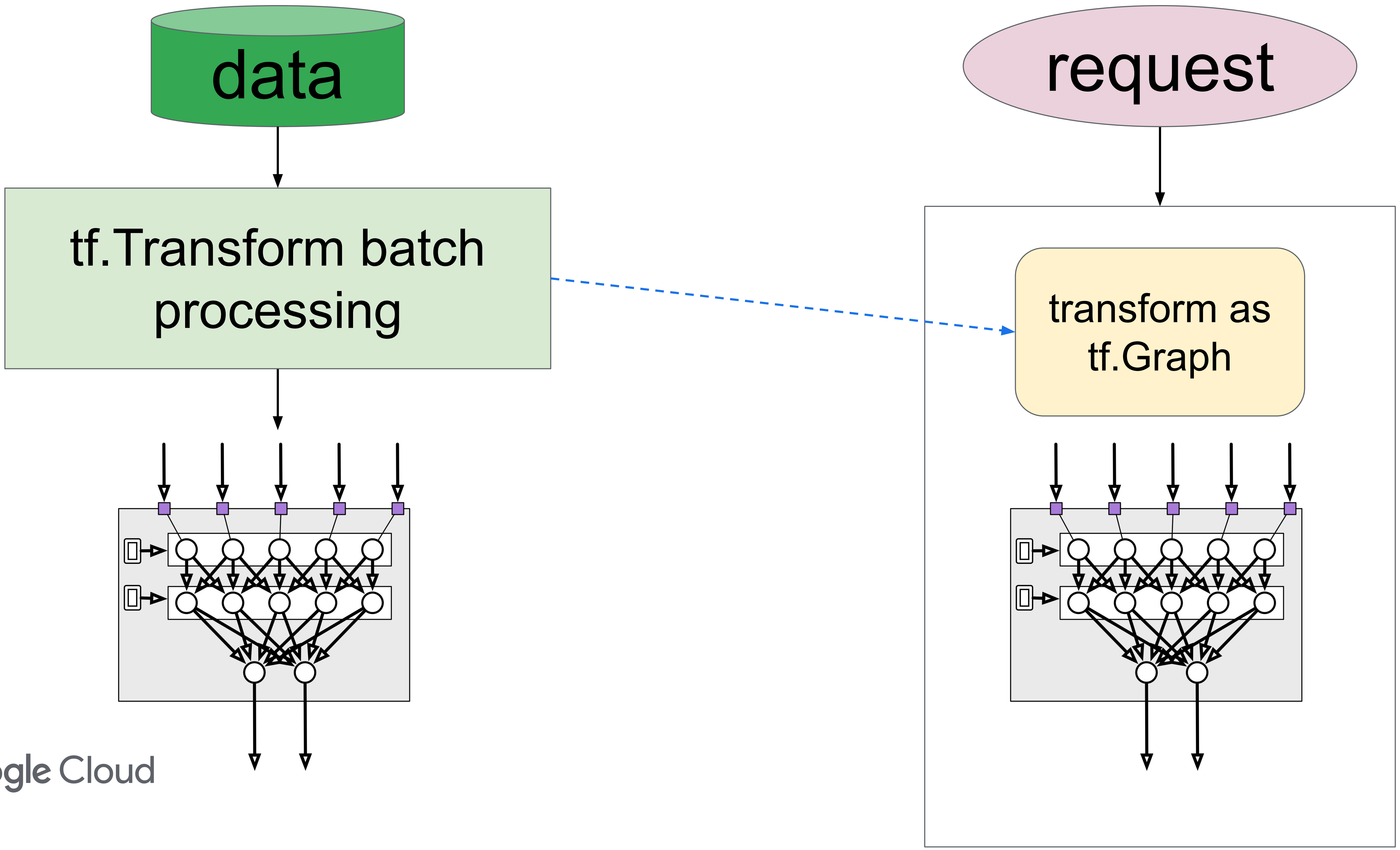
Partial Solutions

- Do everything in the training graph
- Do everything in the training graph + using statistics/vocabs generated from raw data.

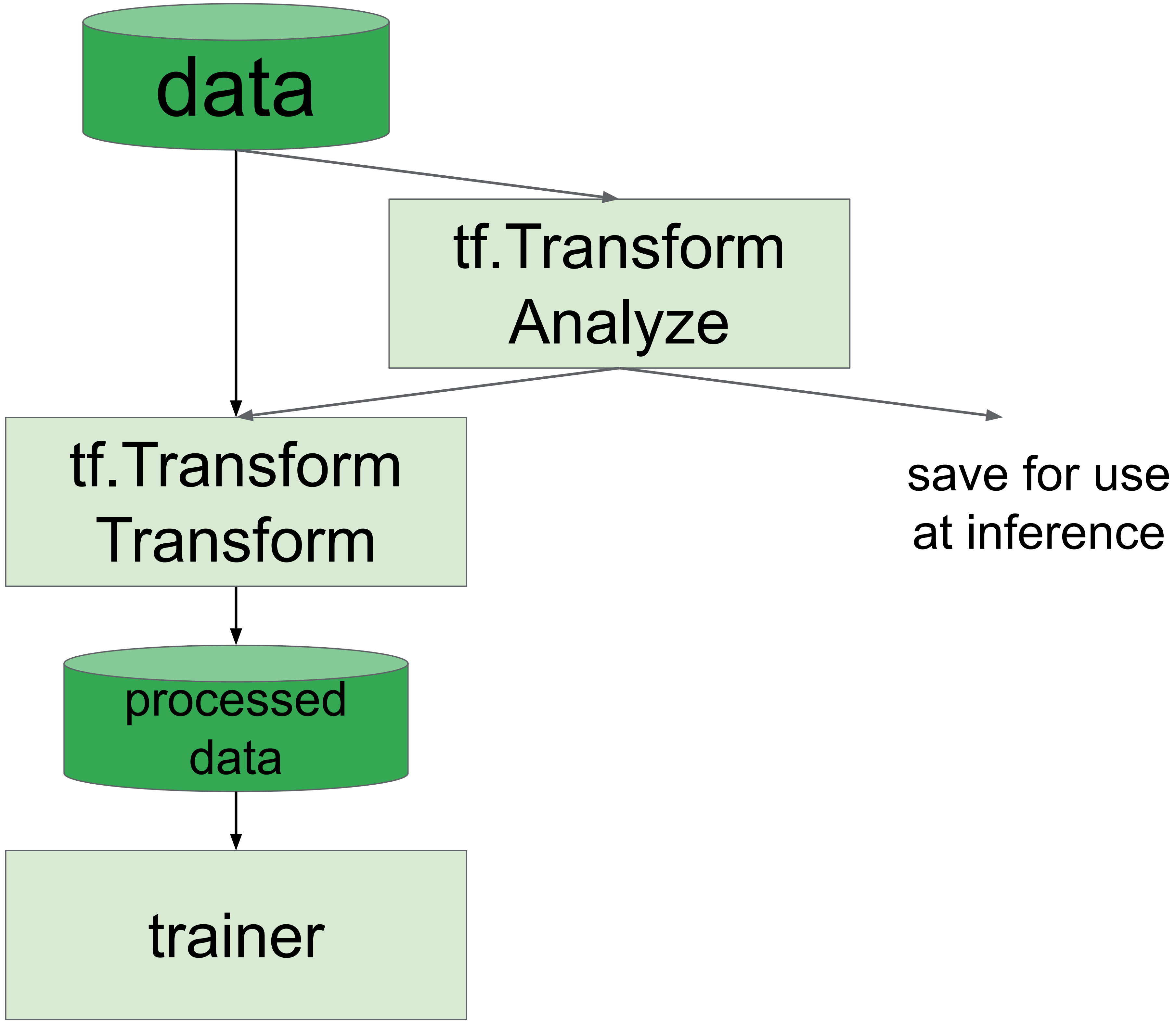
tf.Transform

During training

During serving

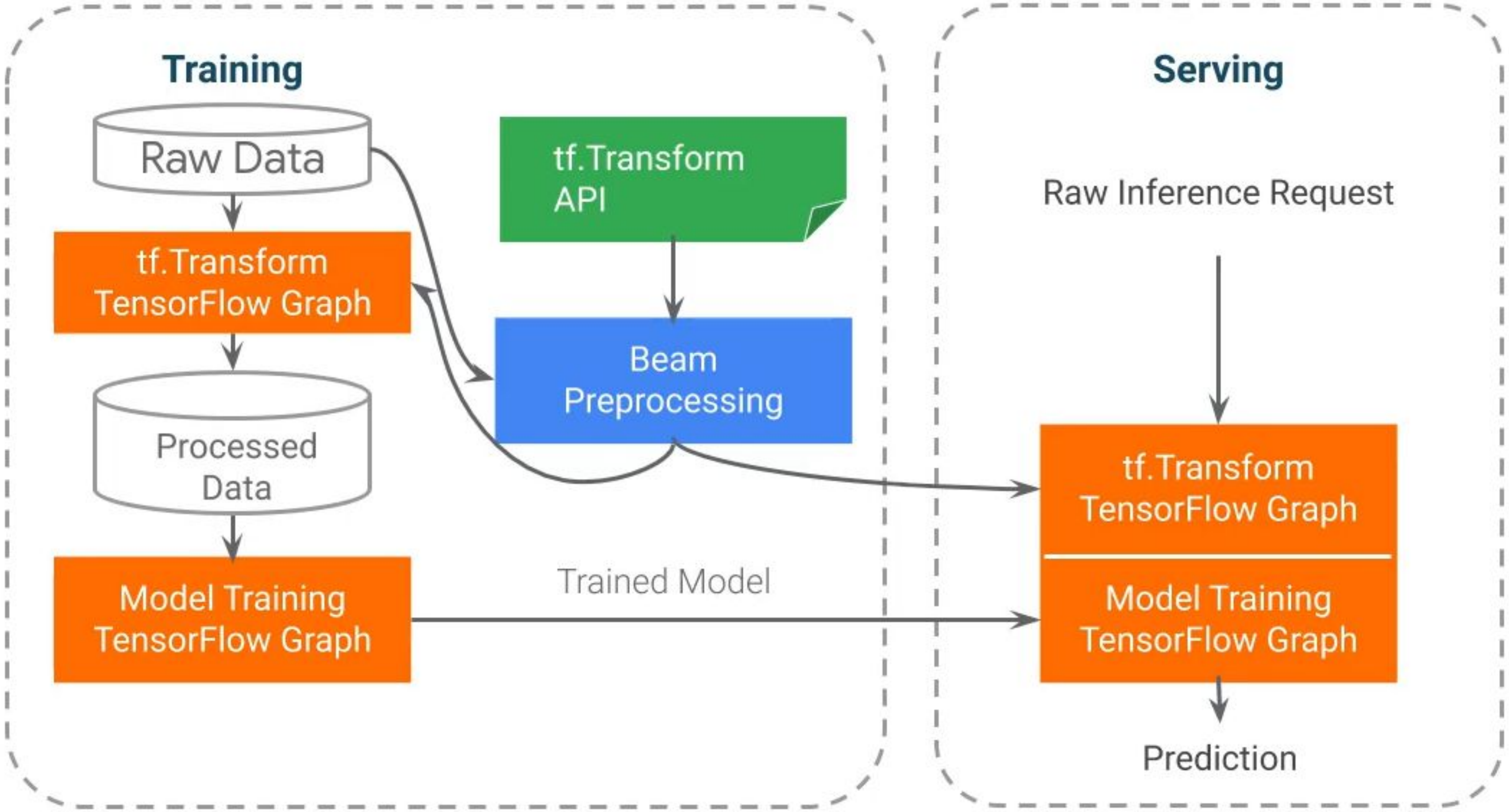


tf.Transform



Google Cloud

Component: **Transform**



ML project lifecycle benefits

- Generates SavedModel for consistent training and serving feature engineering
- Backed by Apache Beam for scalable distributed data processing to large datasets

TensorFlow Transform (TFT) library for preprocessing data and feature engineering with TensorFlow

Analyzers

- max
- min
- mean
- sum
- var
- covariance
- quantiles
- size
- vocabulary
- pca

Mappers

- bucketize
- apply_buckets
- hash_string
- ngram
- scale_0+1
- scale_z_score
- Scale_max_min
- tfidf
- compute_and_apply_vocabulary

tf.transform is a hybrid of Apache Beam and TensorFlow

Analyze	Transform
Find min/max value of a numeric feature	Scale inputs by the min & max
Find all the unique values of a categorical feature	One-hot encode inputs based on set of unique values
Beam	TensorFlow

Google Cloud

tf.transform provides two PTransforms

`AnalyzeAndTransformDataset`

- Executed in Beam to create the training dataset.
- Similar in purpose to Scikit-learn's `fit_transform` method

`TransformDataset`

- Executed in Beam to create the evaluation dataset
- The underlying transformations are executed in TensorFlow at prediction time
- Similar in purpose to Scikit-learn's `tranform` method.

tf.transform has two phases

Analysis phase (compute min/max/vocab etc. using Beam)

- Executed in Beam while creating training dataset

Transform phase (scale/vocabulary etc. using TensorFlow)

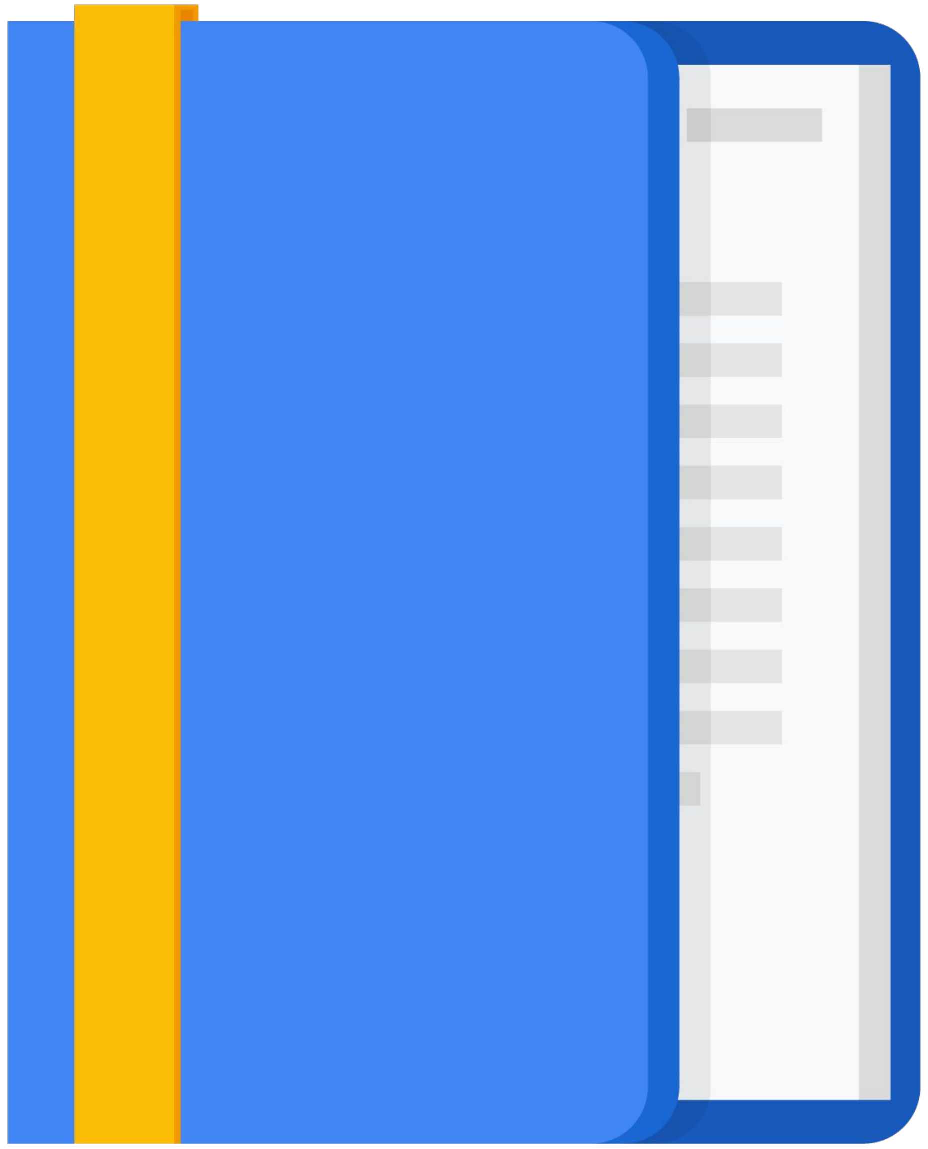
- Executed in TensorFlow during prediction
- Executed in Beam to create training/evaluation datasets

Course Agenda

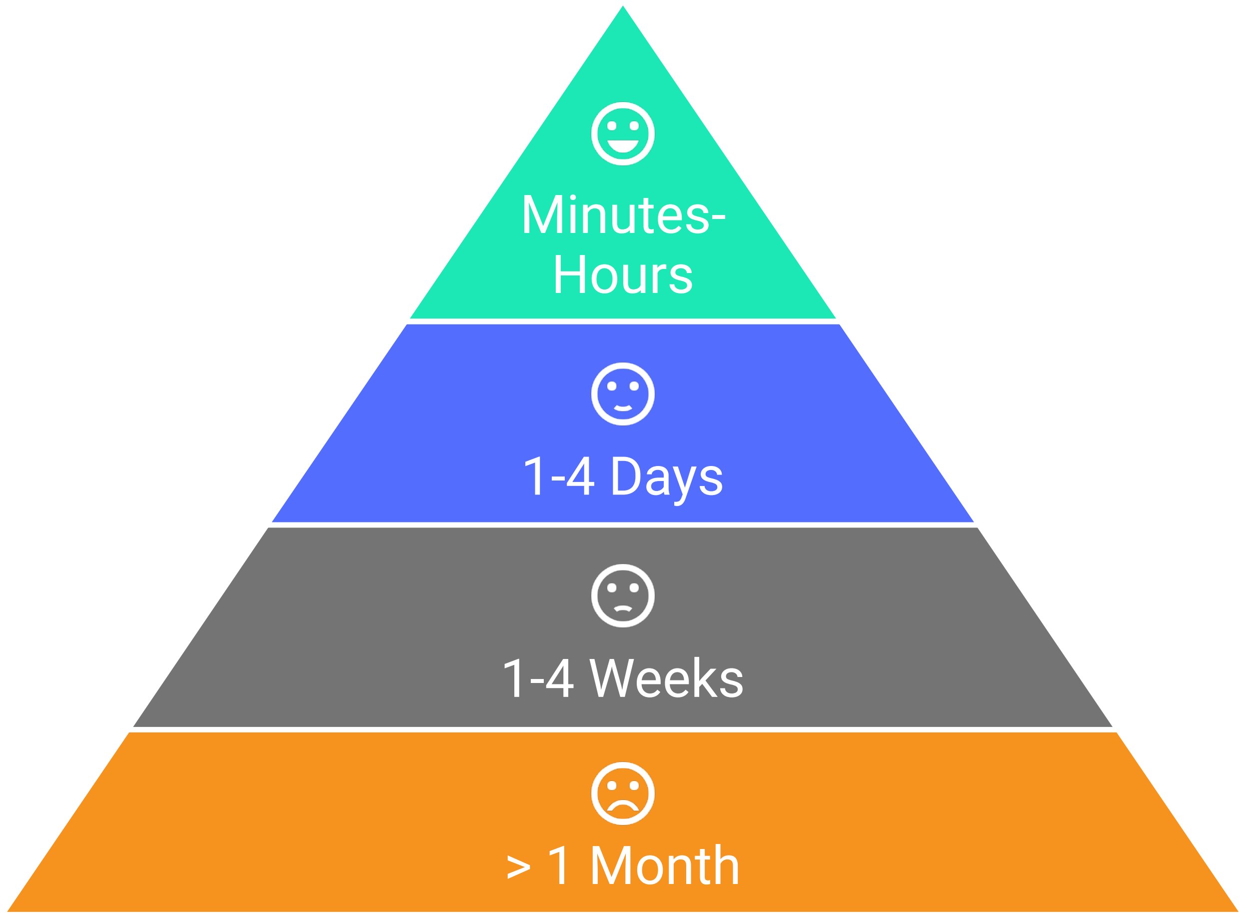
Optimize Data Input Pipeline with TensorFlow

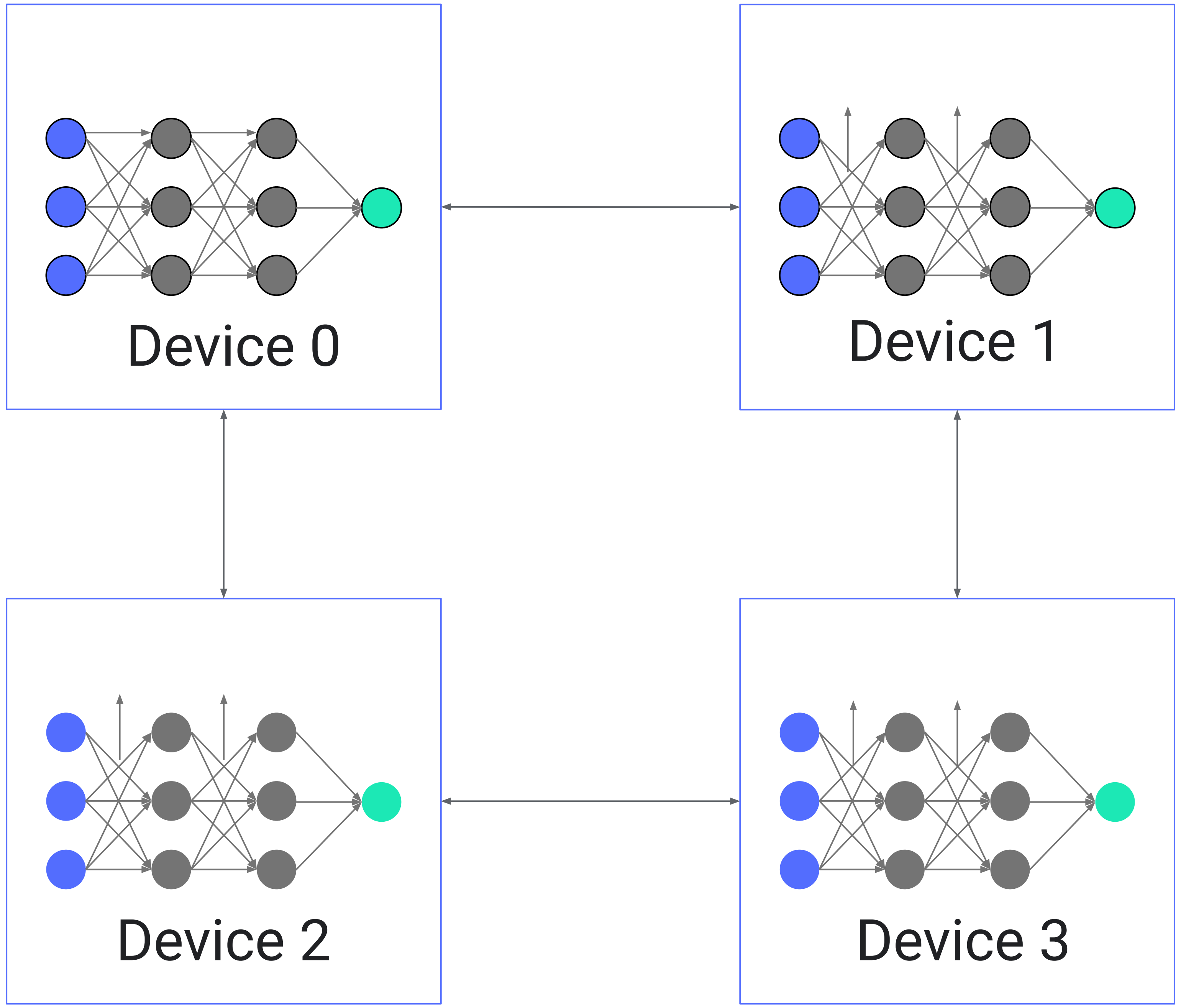
Feature Engineering with TFT

Distributing Strategies with TF

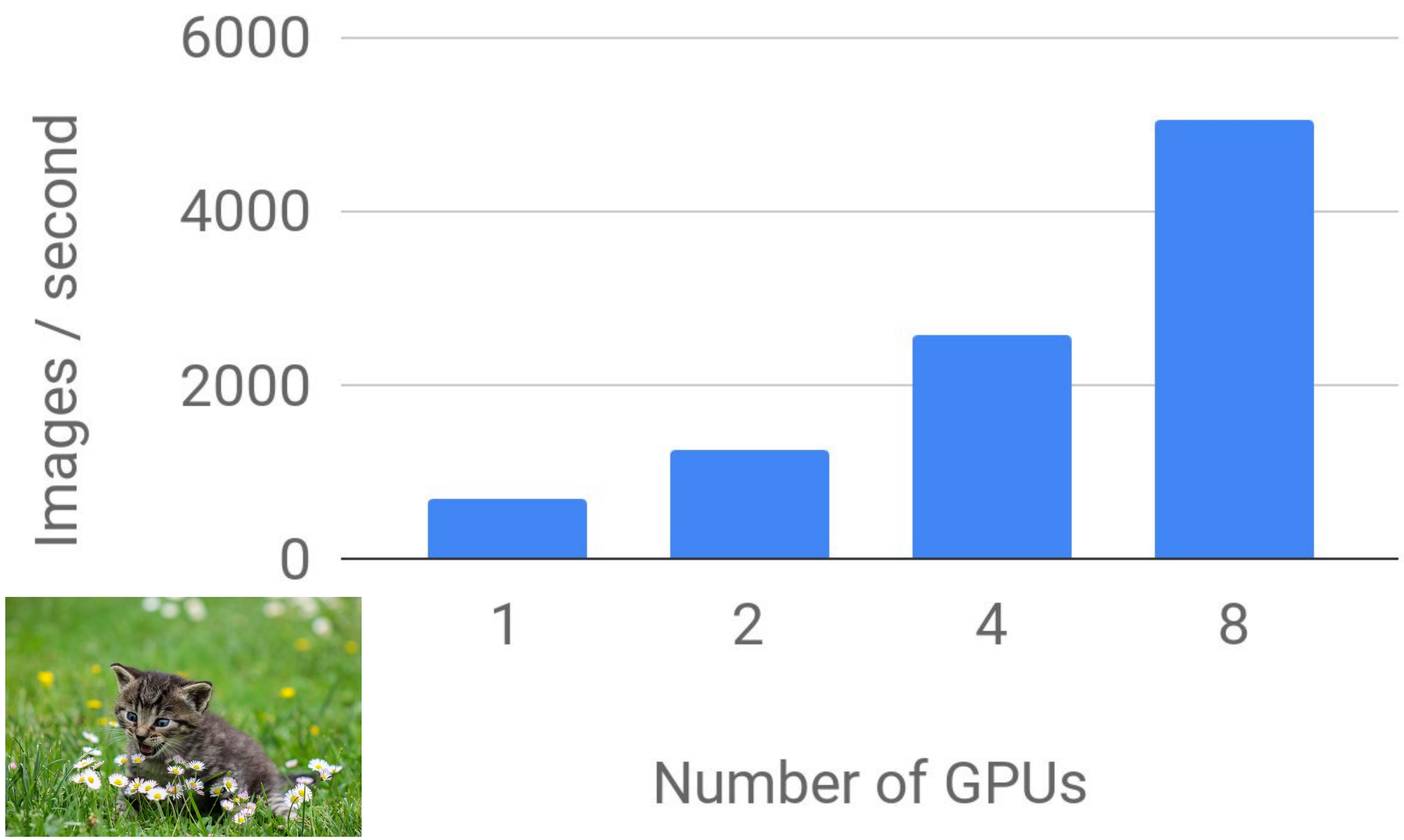


How can you make model training faster?

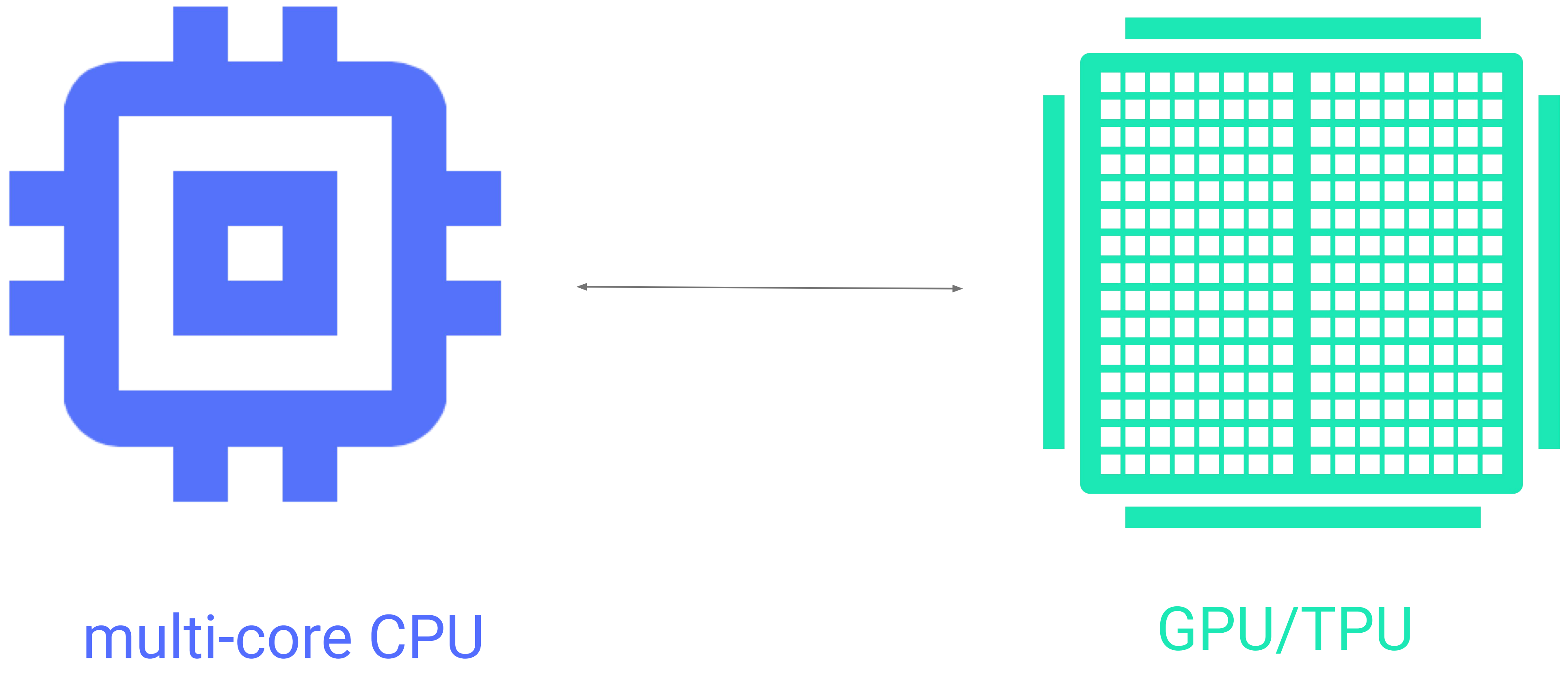




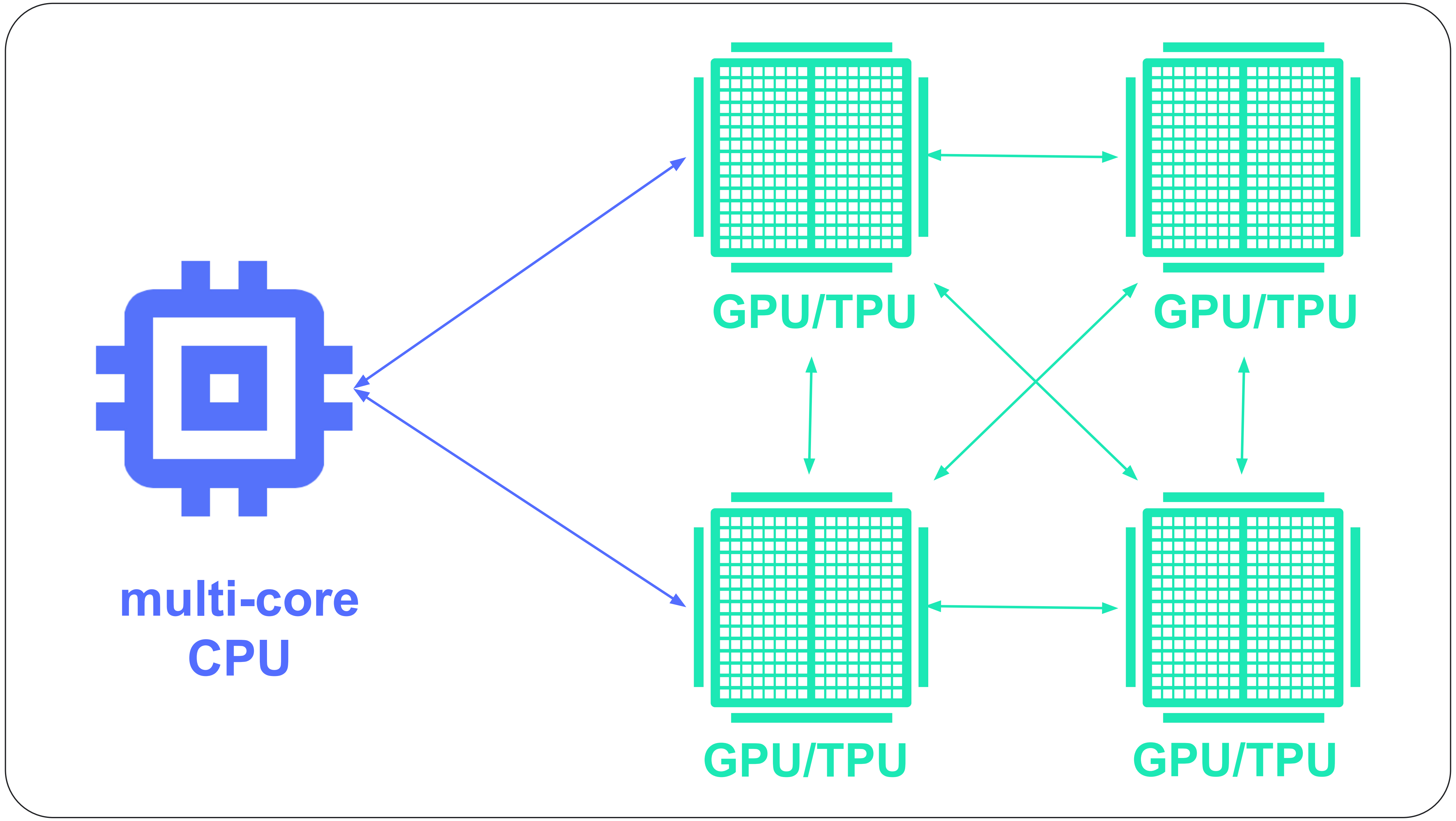
Scaling with distributed training



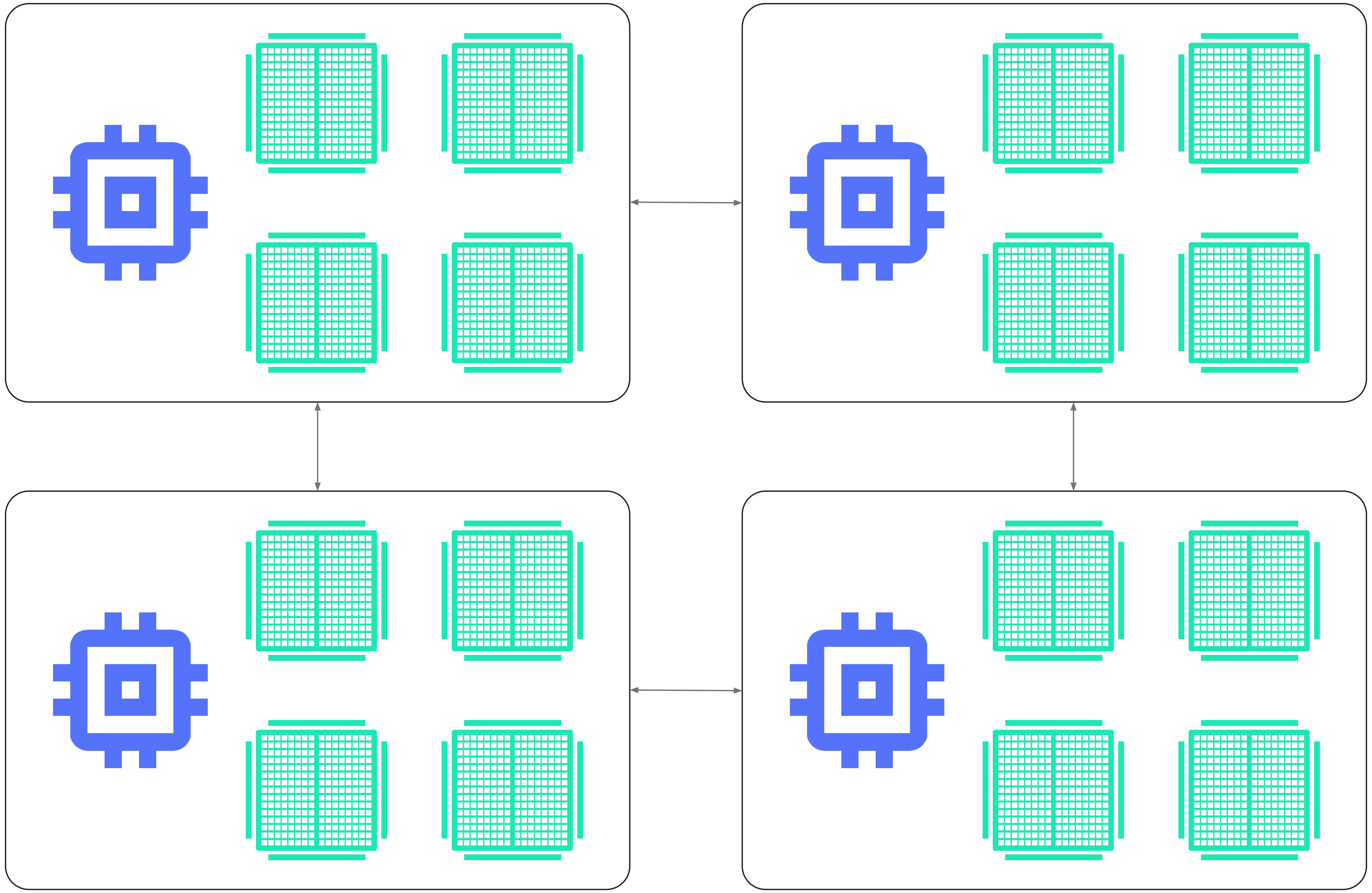
Start training on a machine with a multi-core CPU , then add a single accelerator



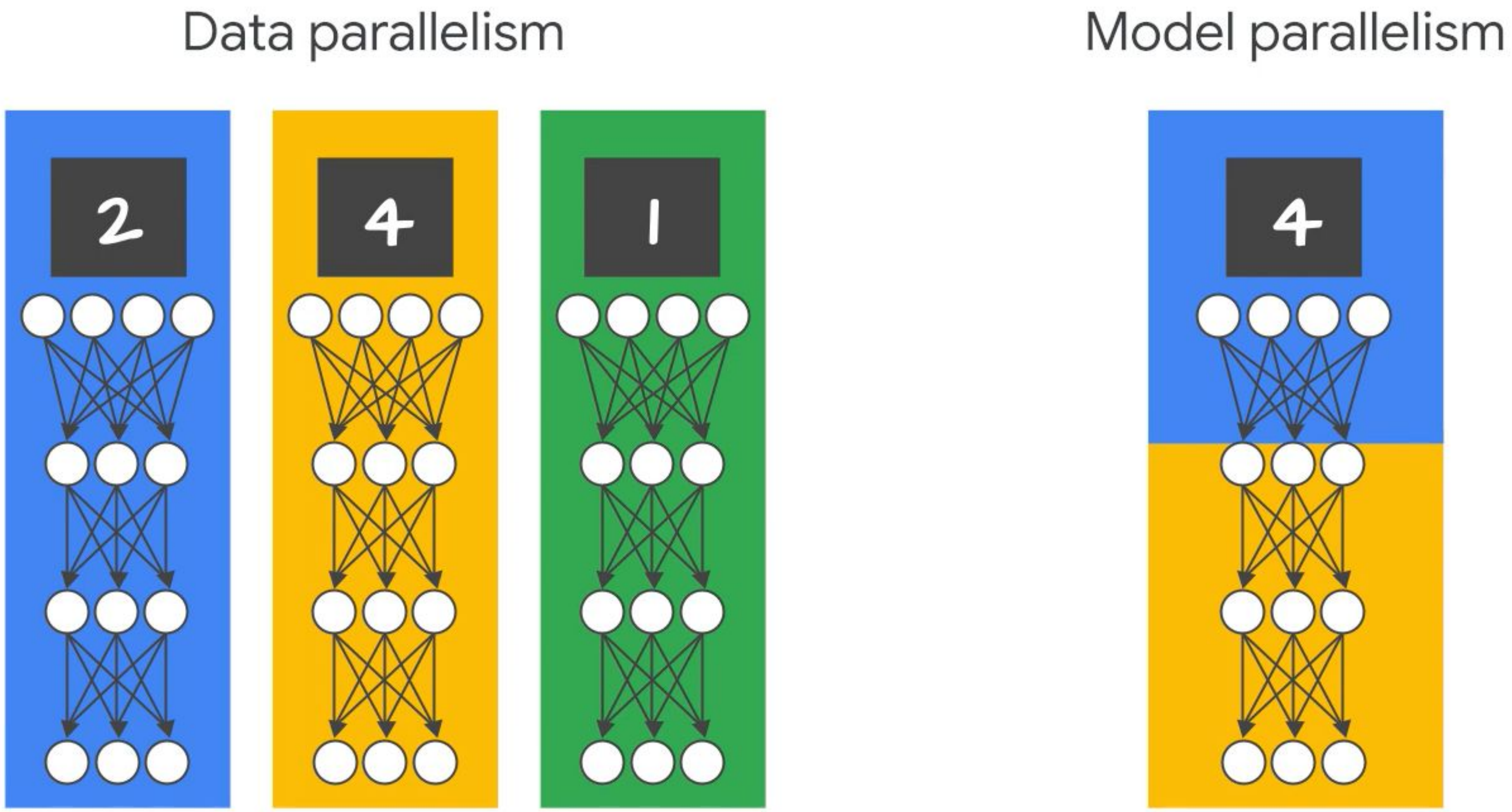
Adding many accelerators to a single device



Adding many machines with many possible devices



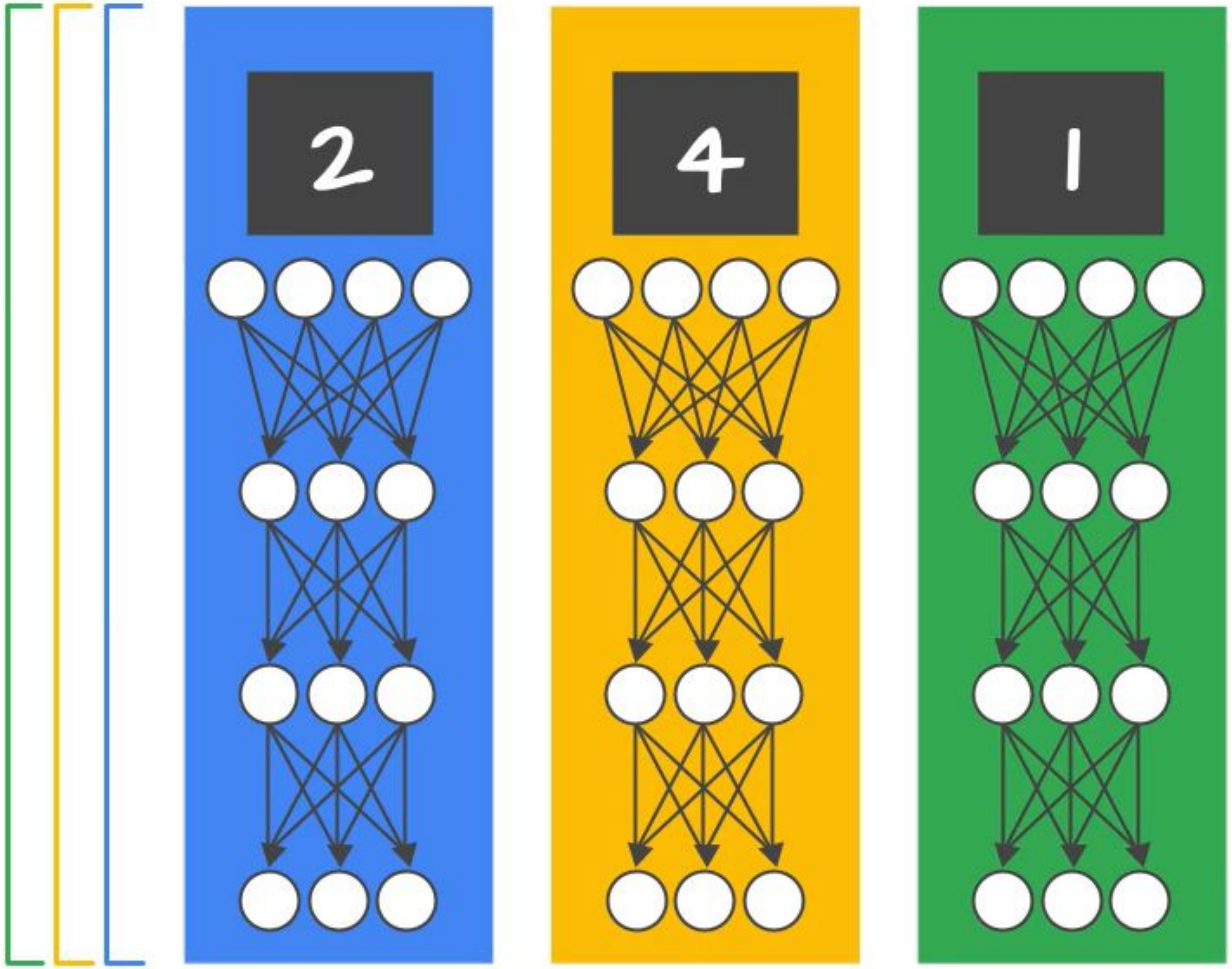
Distributed Training Architectures



Data Parallelism

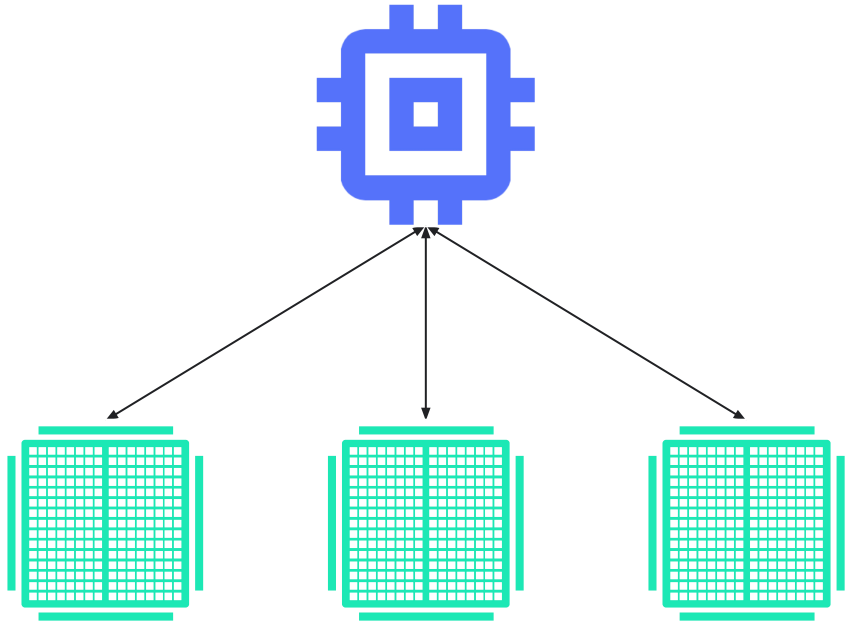
Each device computes loss and gradients based on the training samples.

Gradient measures the change in all weights.



There isn’t one right answer, but here are some considerations

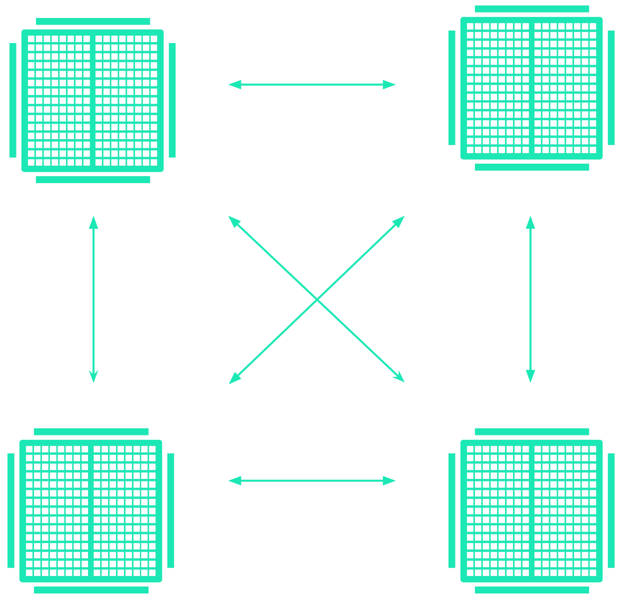
Consider async parameter server if...



Many low-power or unreliable workers.

More mature approach.
Constrained by I/O.

Consider sync allreduce if...

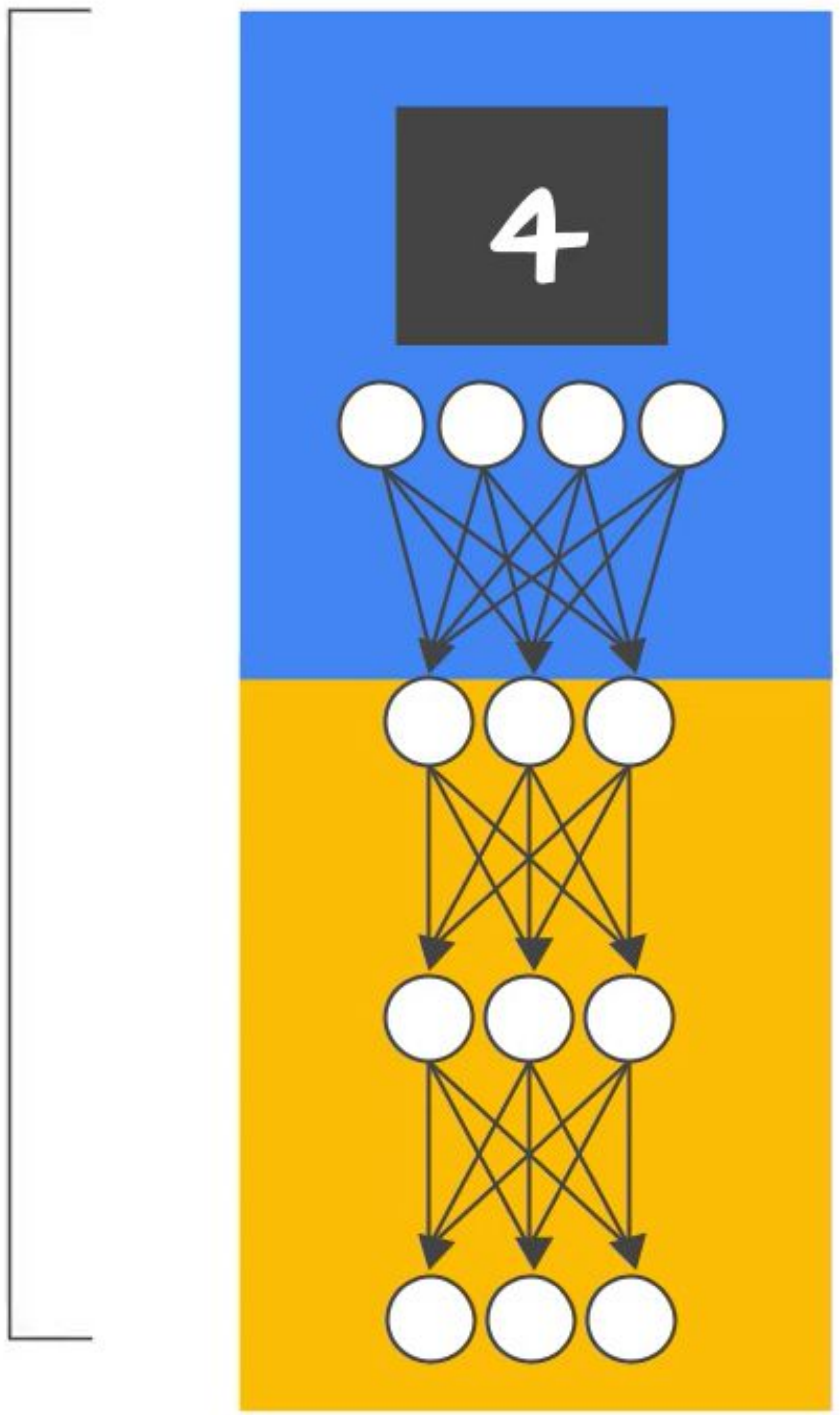


Multiple devices on one host.
Fast devices with strong links (e.g. TPUs).

Better for multiple GPUs.
Constrained by compute power.

Model parallelism

Each GPU has different parameters, and computation, of different parts of a model.



Training Distribution challenges

How will you distribute the data across the different devices?

How will you accumulate the gradients during backpropagation?

How will the model parameters be updated?

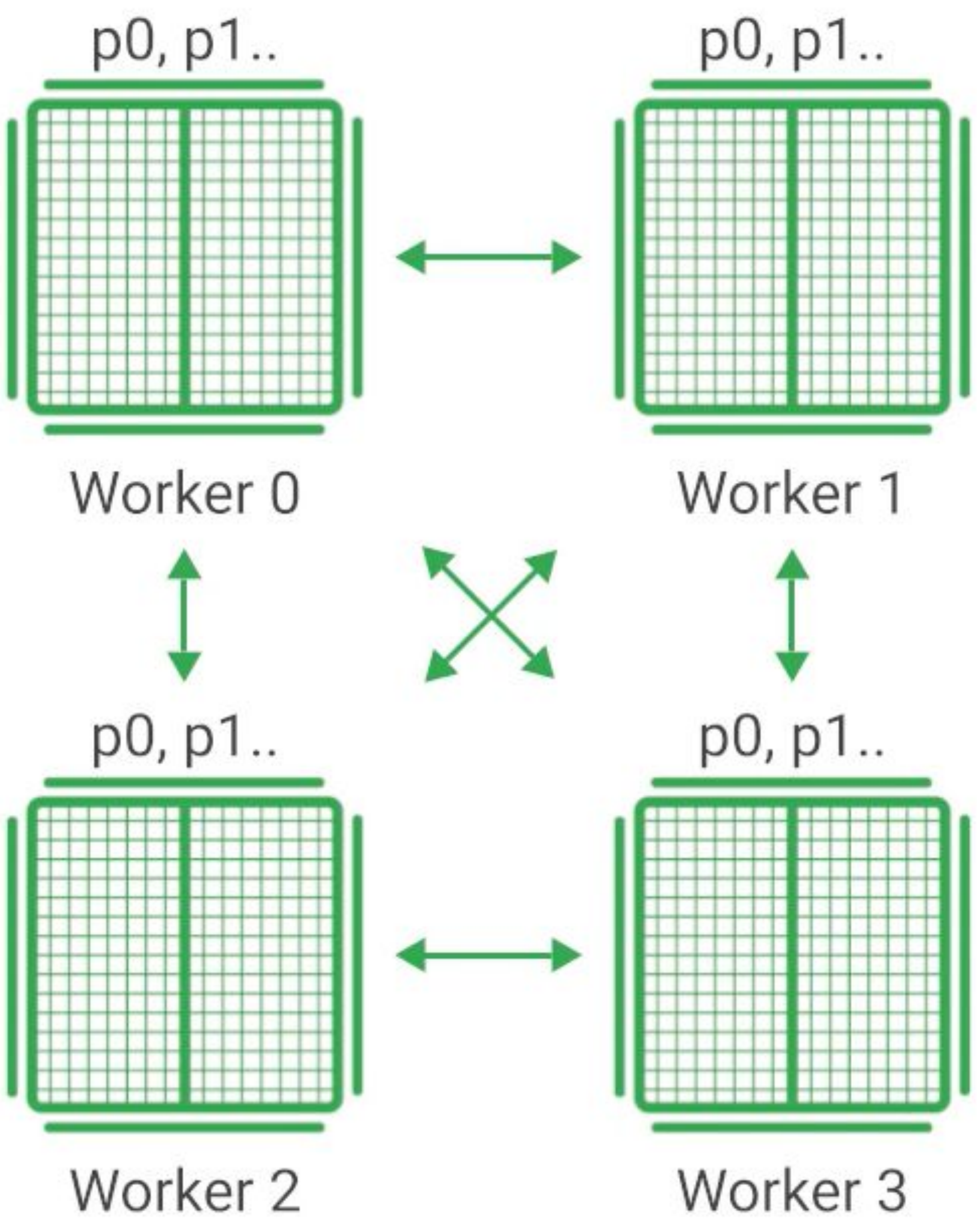
tf.distribute.Strategy API

TensorFlow distributed training strategies

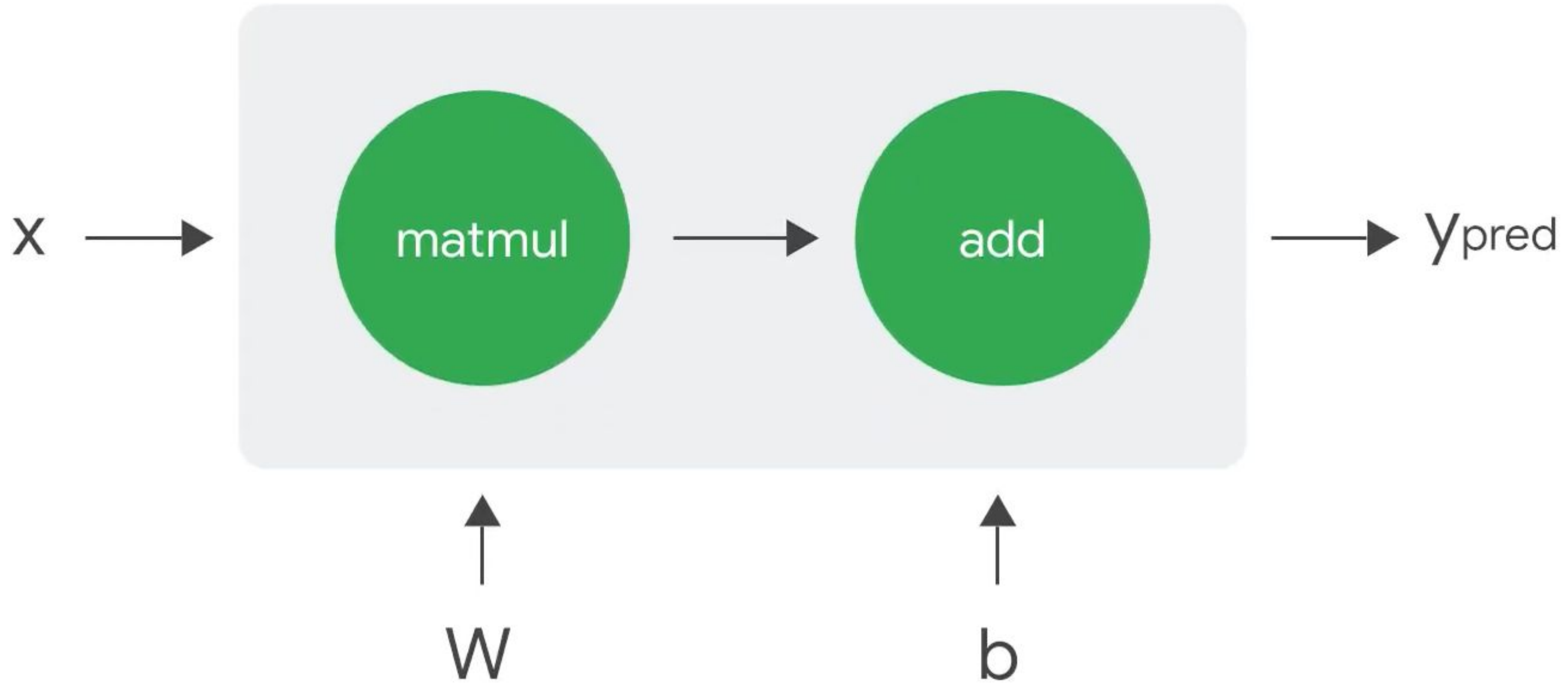
- Mirrored strategy
- Multi-worker mirrored strategy
- TPU strategy
- Parameter server strategy

Mirrored Strategy

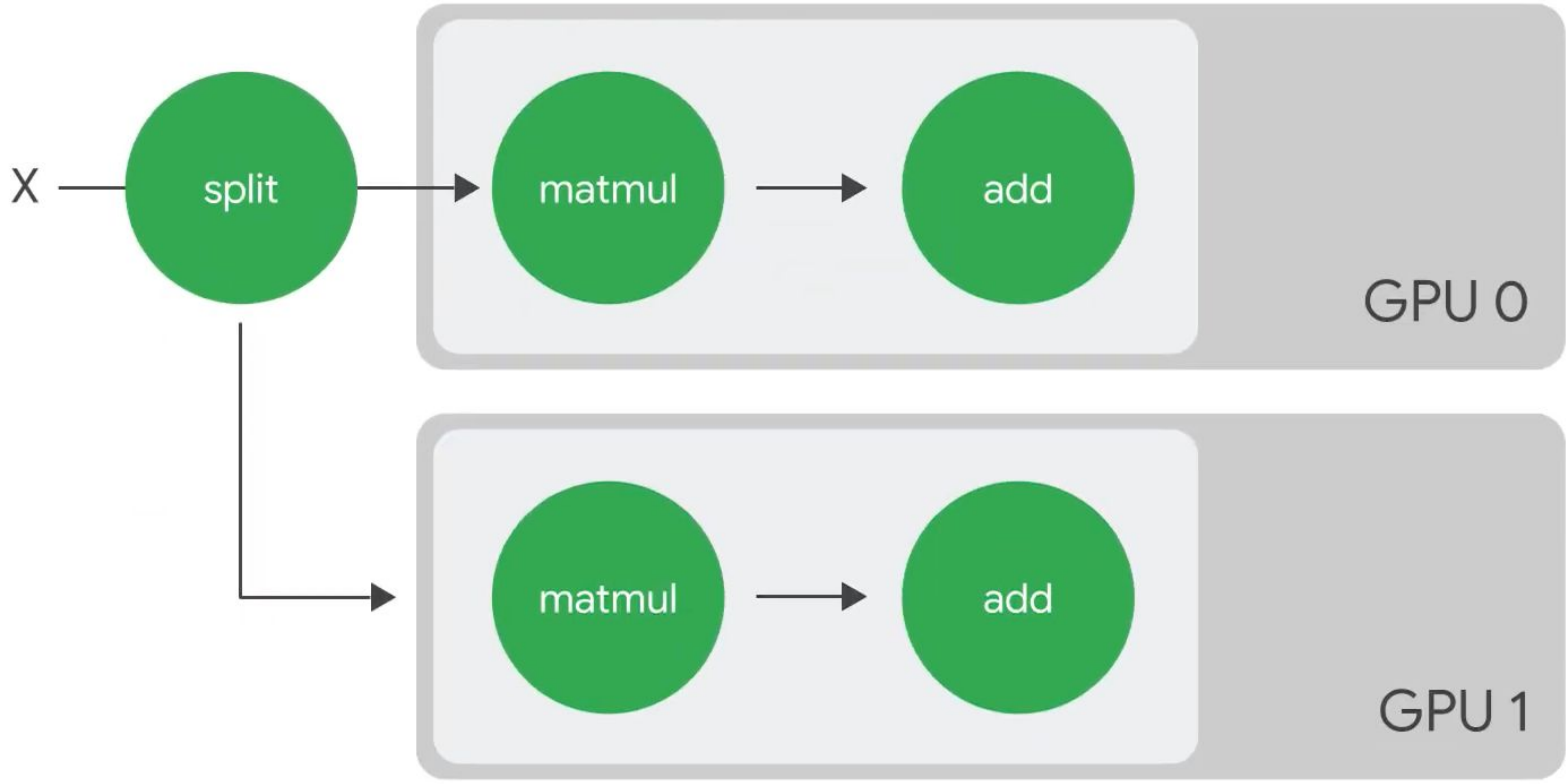
- Synchronous Strategy one machine with many accelerators
- Creates a replica of the model on each GPU
- Mini batch is split into n
- Data distribution and Gradient updates are automatically updated



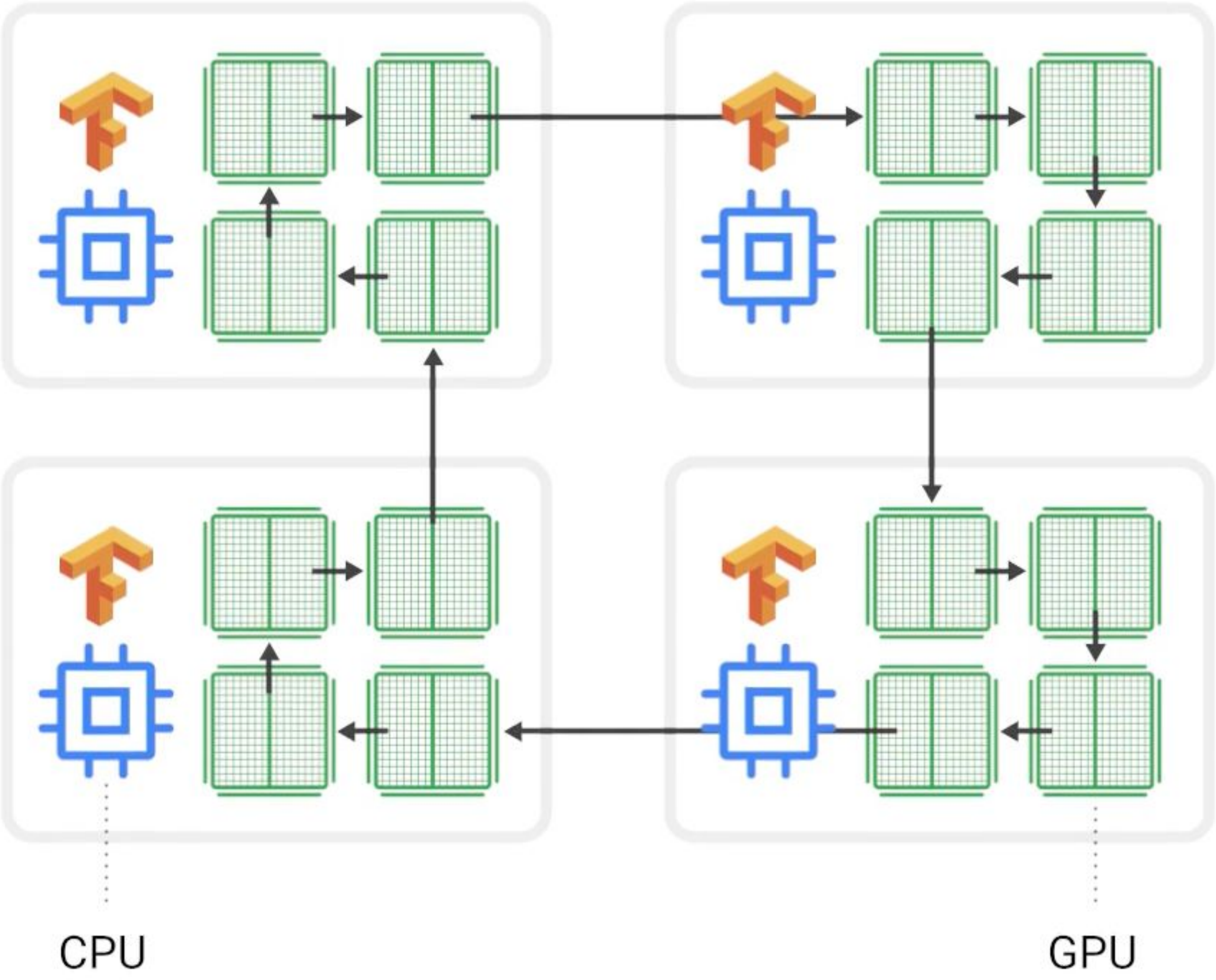
DAG without Mirrored Strategy



DAG with Mirrored Strategy



Multi-worked Mirrored Strategy



Multi-worked Mirrored Strategy

```
os.environ["TF_CONFIG"] = json.dumps({
    "cluster": {
        "chief": ["host1:port"],
        "worker": ["host2:port", "host3:port"],
    },
```

Multi-worked Mirrored Strategy

tf.distribute

1. Create a strategy object.

```
strategy =
tf.distribute.MultiWorkerMirroredStrategy()
```

2. Wrap the creation of the model parameters within the Scope of the strategy.

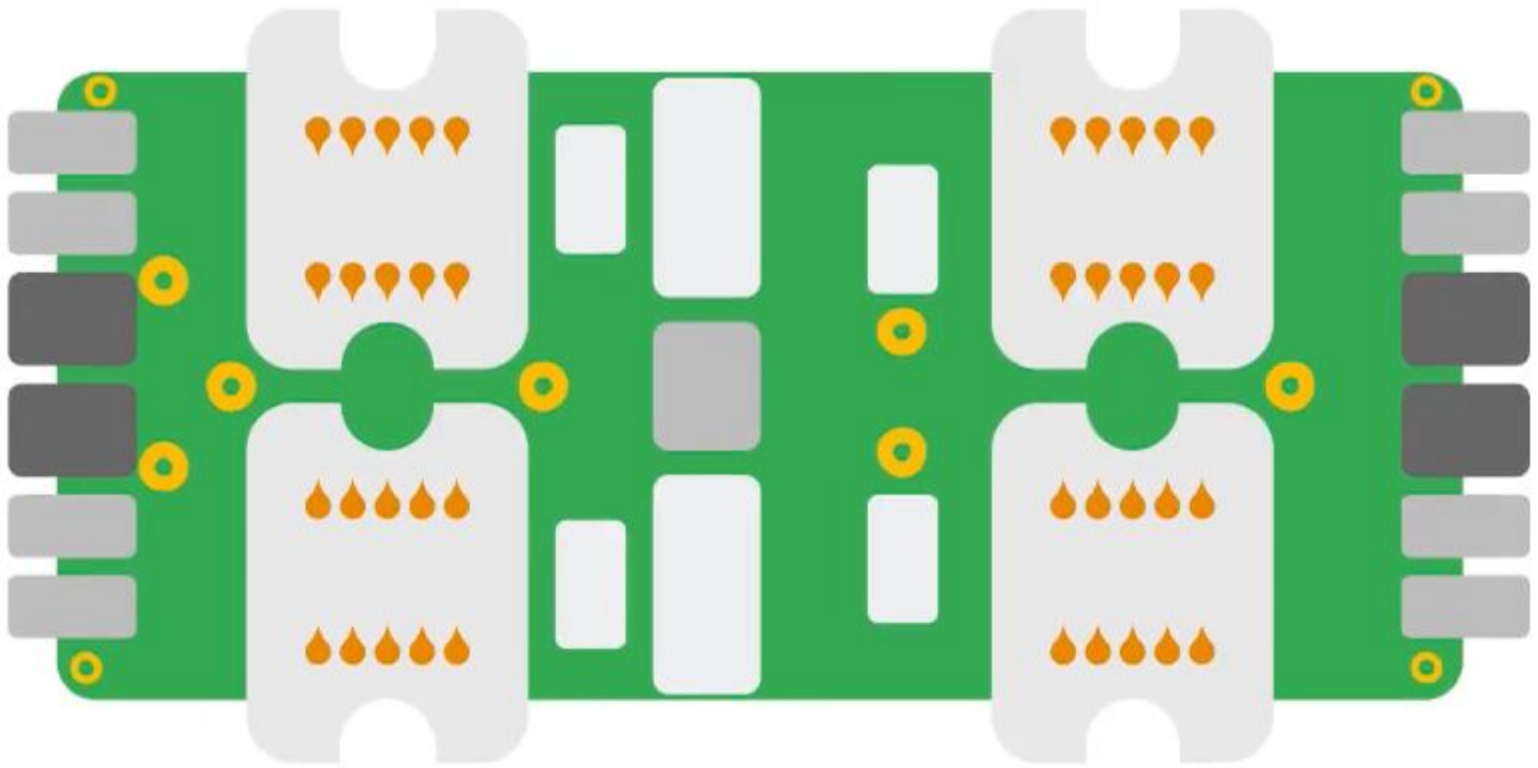
```
with strategy.scope():
  model = create_model()
  model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(0.0001),
    metrics=['accuracy'])
```

3. Scale the batch size by the number of replicas in the cluster.

```
per_replica_batch_size = 64
global_batch_size =
per_replica_batch_size *
strategy.num_replicas_in_sync
```

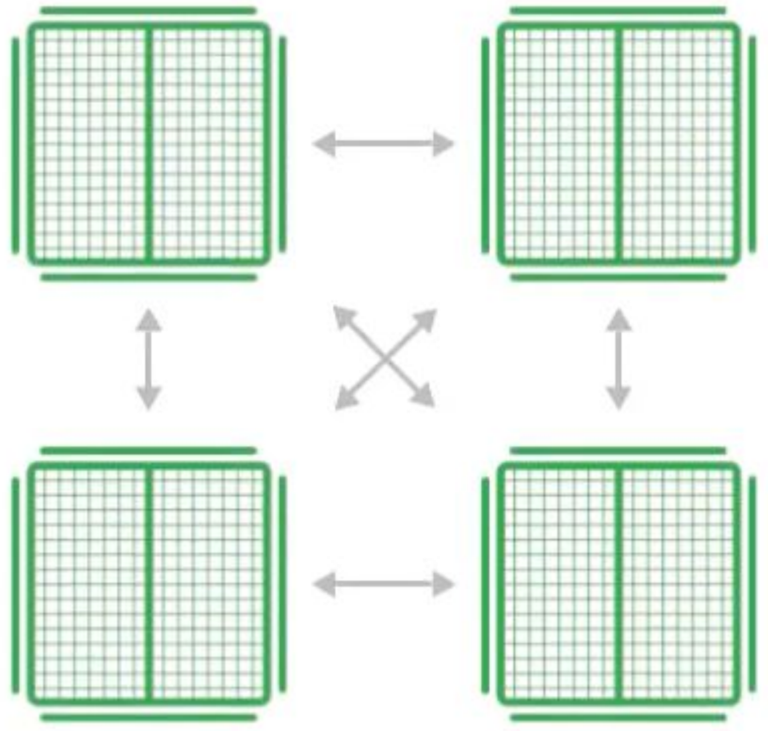
TPU Strategy

TPUStrategy



All-reduce across TPU cores

MirroredStrategy

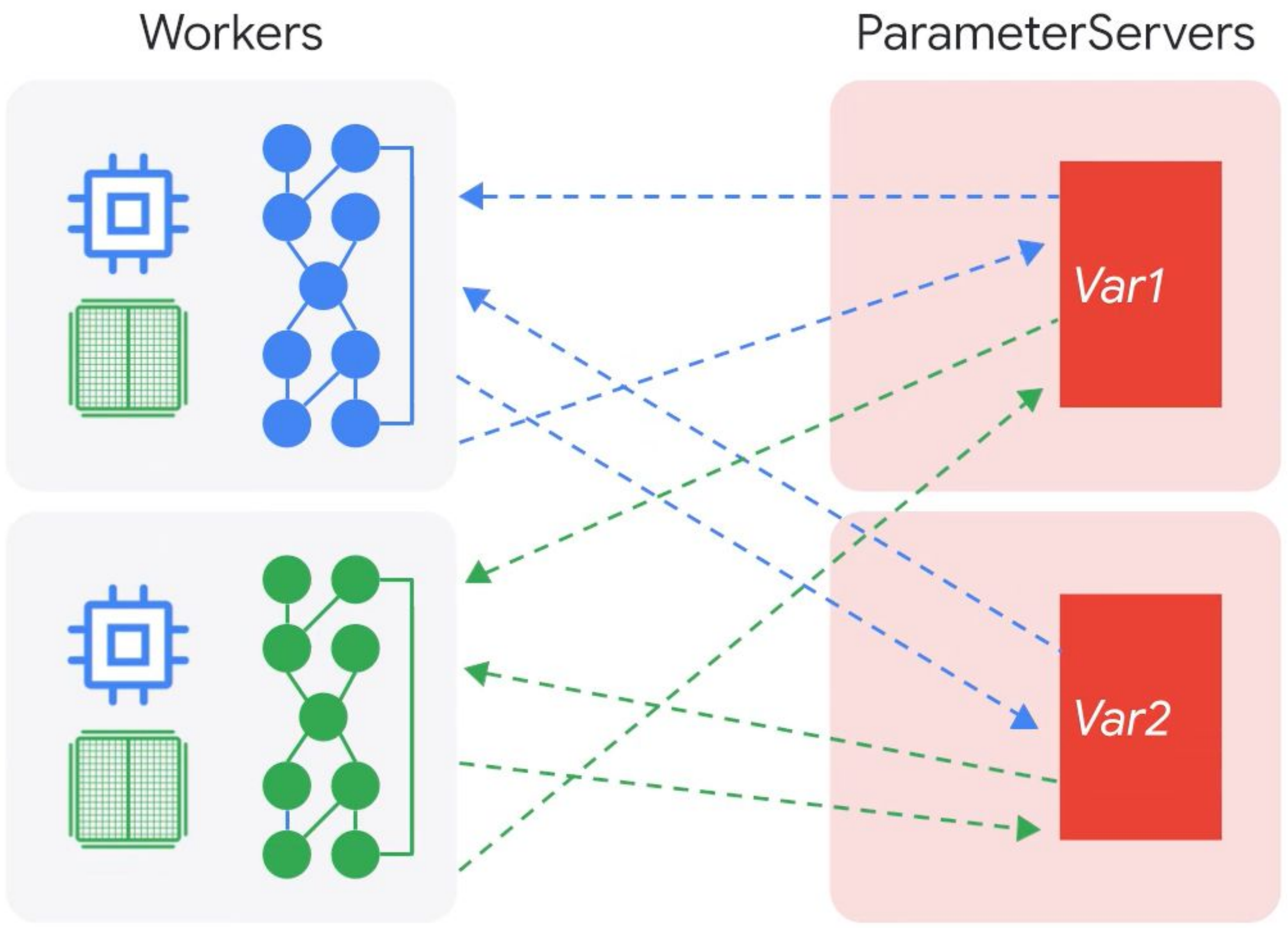


All-reduce across devices

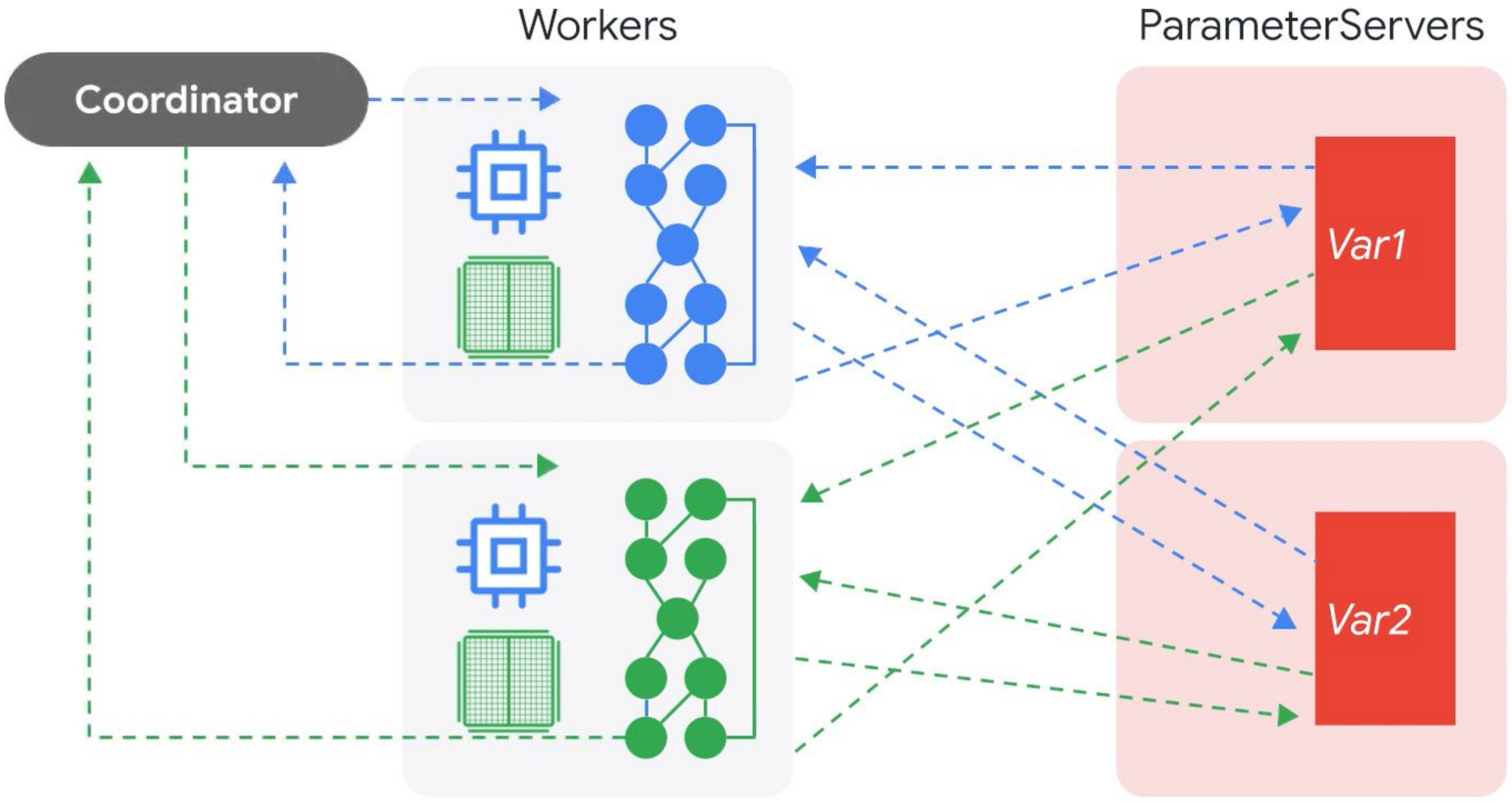
TPU Strategy Implementation

```
strategy = tf.distribute.TPUStrategy()



with strategy.scope():
  model = tf.keras.Sequential(...)
  model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(0.0001),
    metrics=['accuracy'])
```

Parameter Server Strategy



Parameter Server Strategy with Coordinator



Hyperparameter Tuning and Explainability in Google Cloud

Week 6

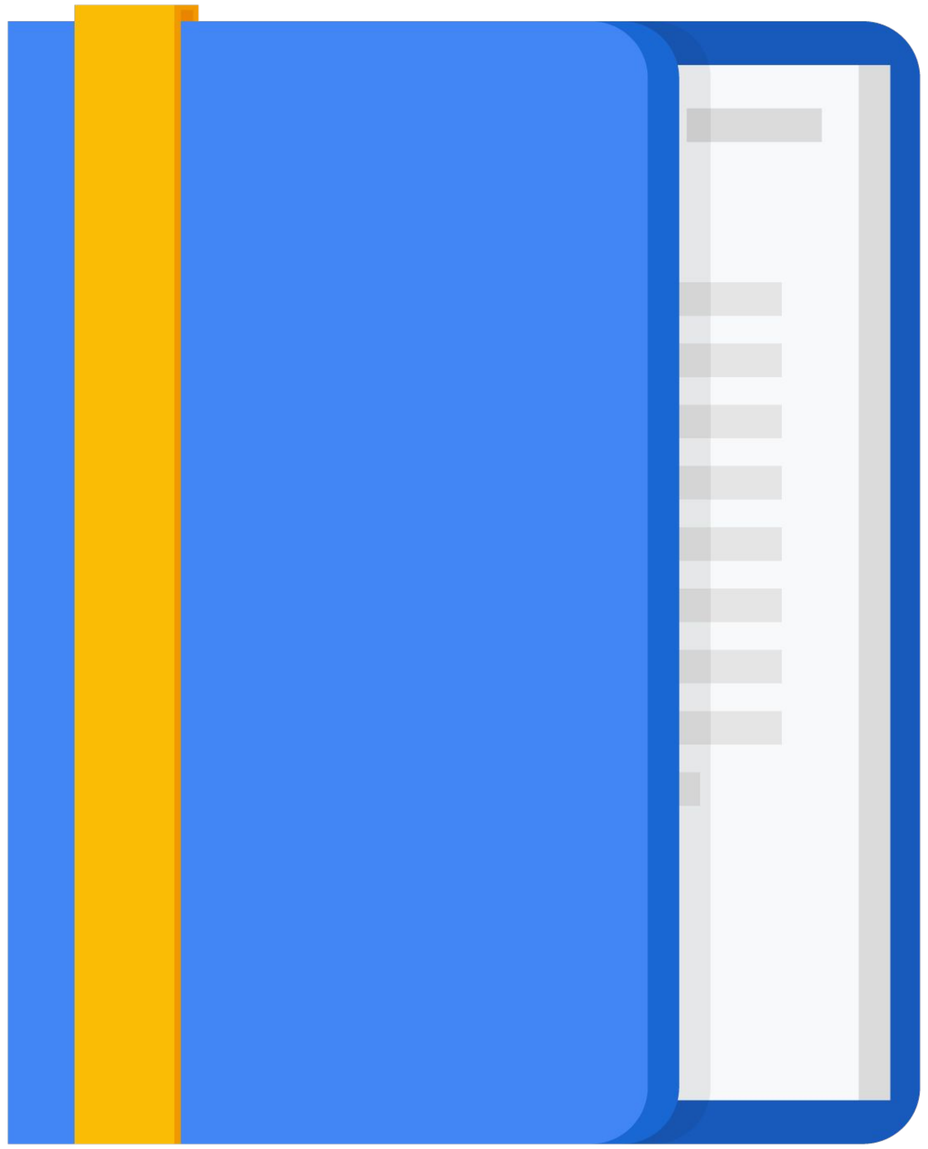
Instructor:
Ben Ahmed

Agenda

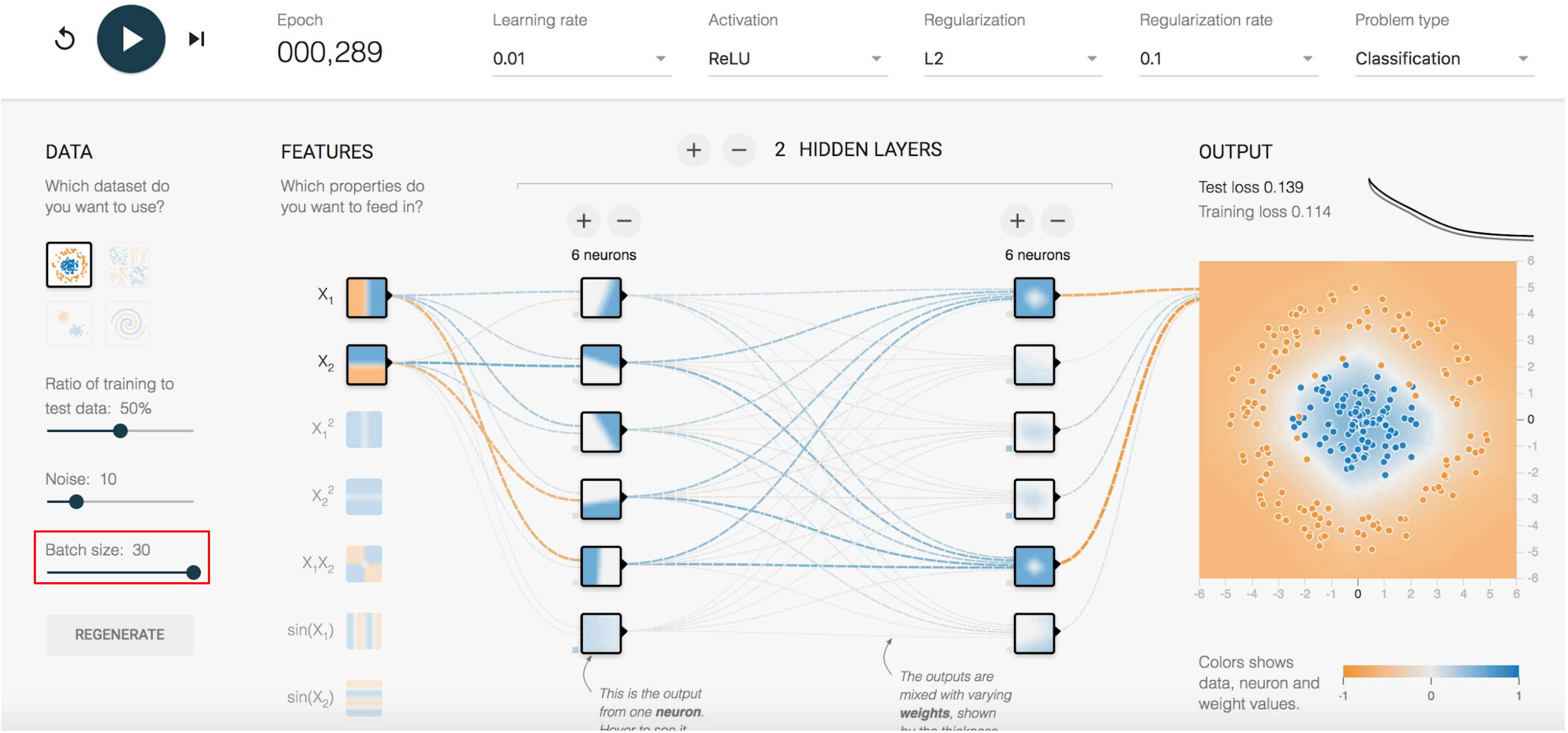
Learning Rate and Batch Size

Hyperparameter Tuning

AI Explanations

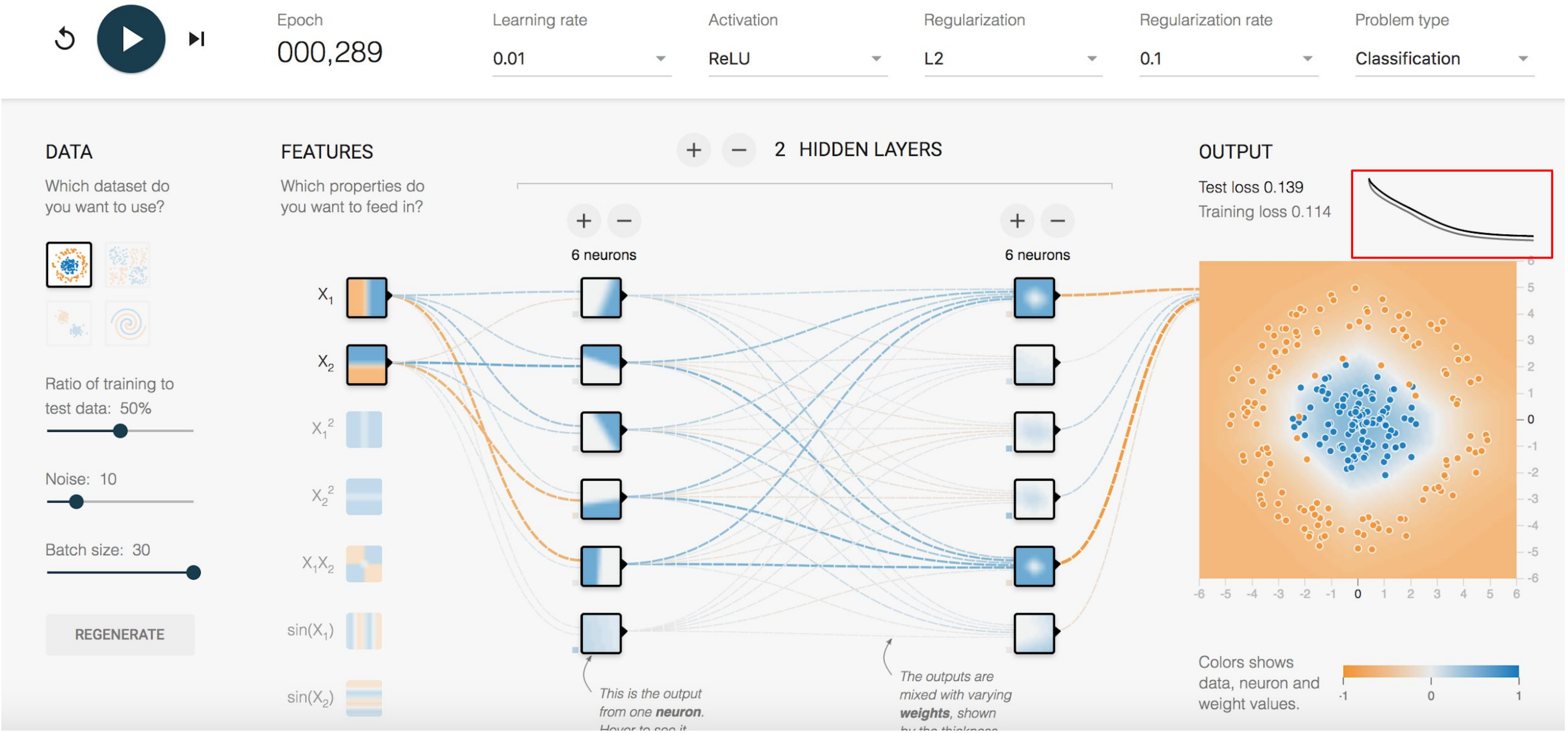


How can you define model complexity?

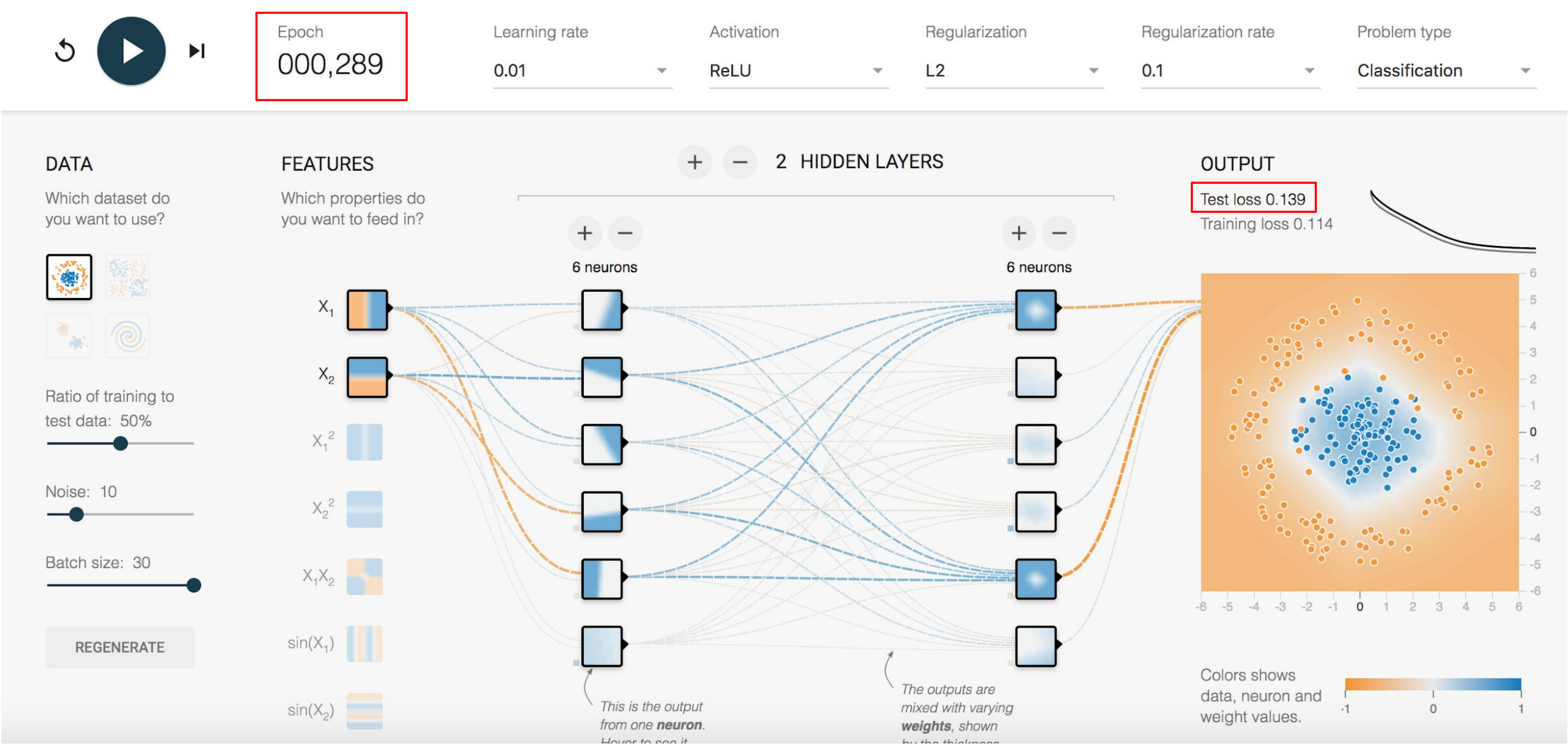


How can you define model complexity?

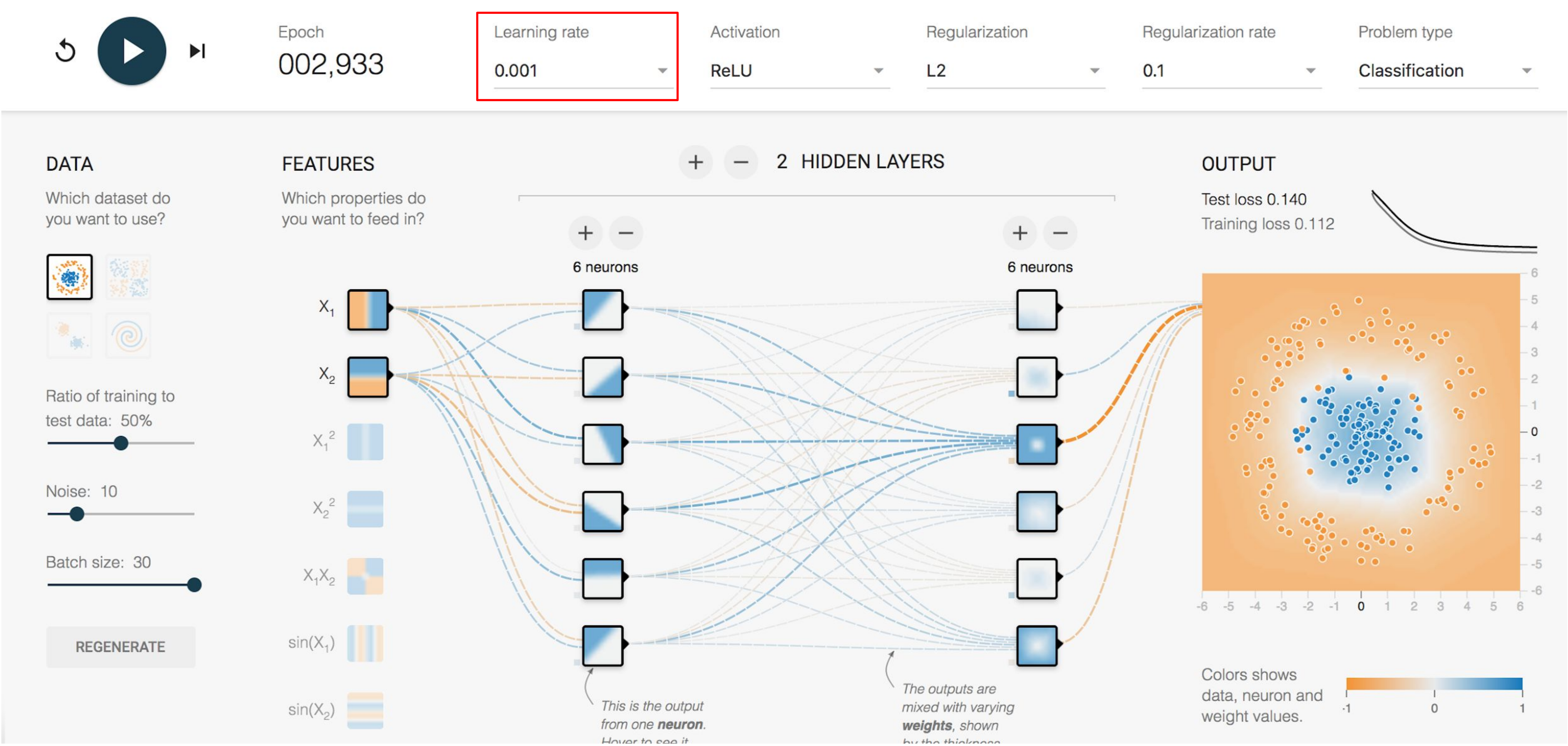
How can you define model complexity?



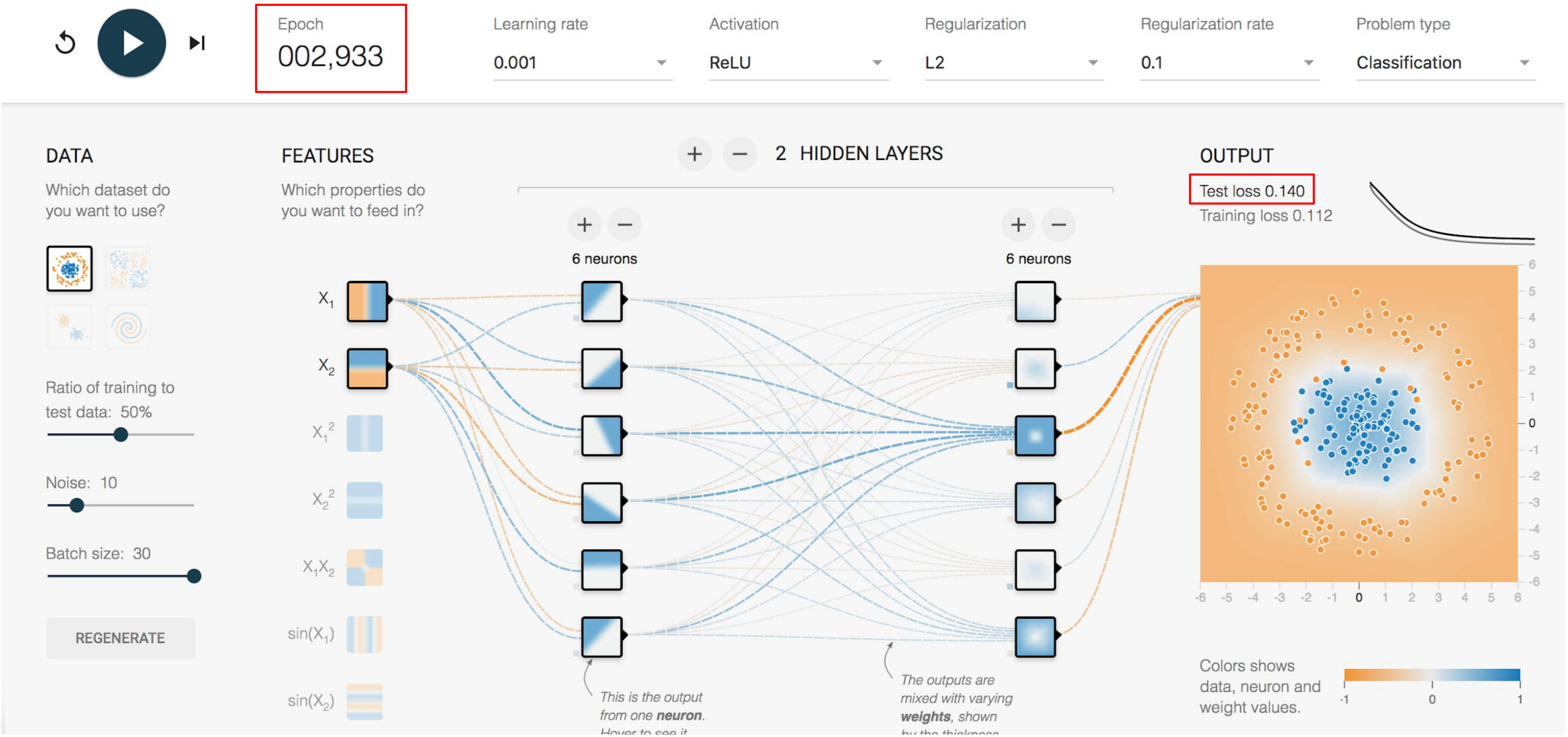
How can you define model complexity?



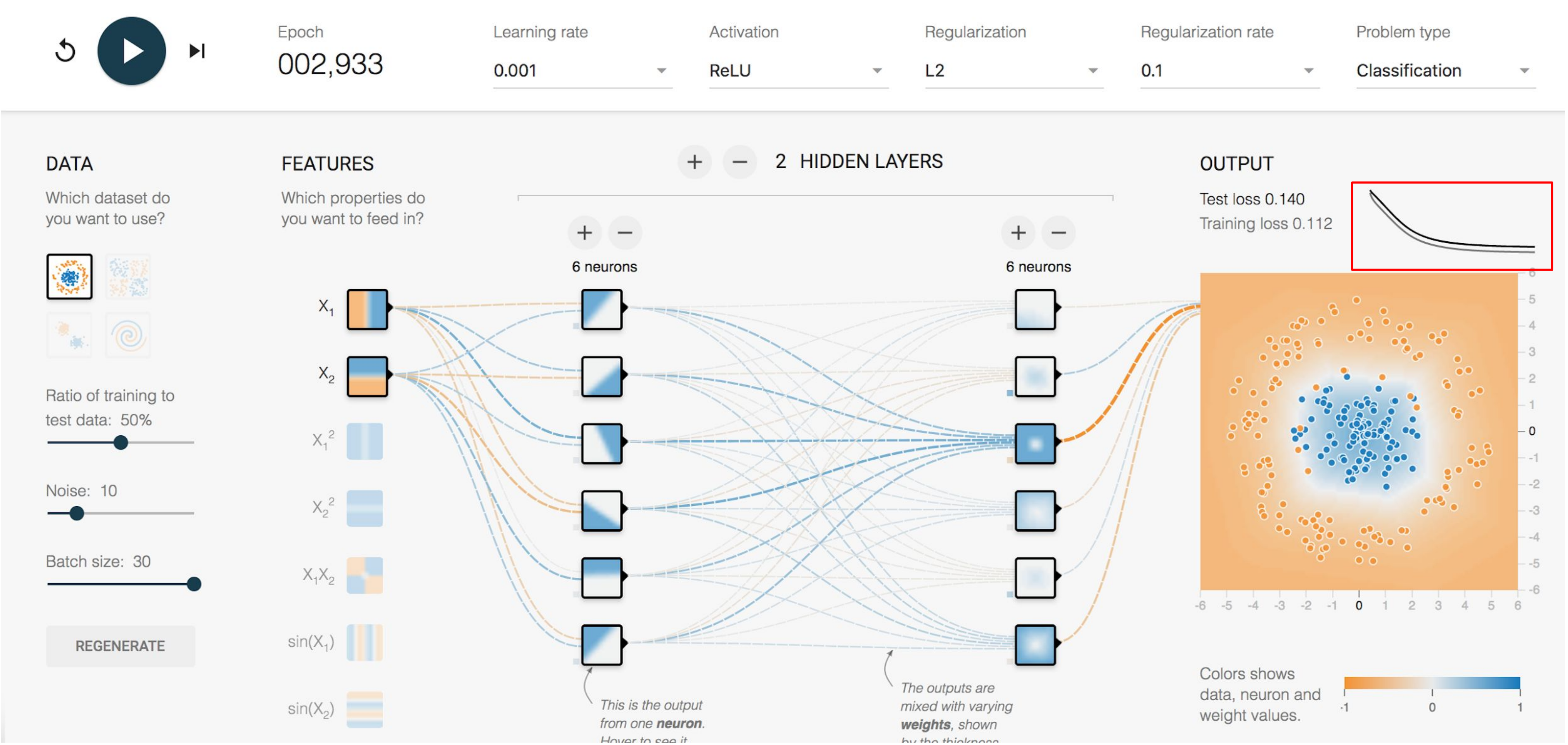
How can you define model complexity?



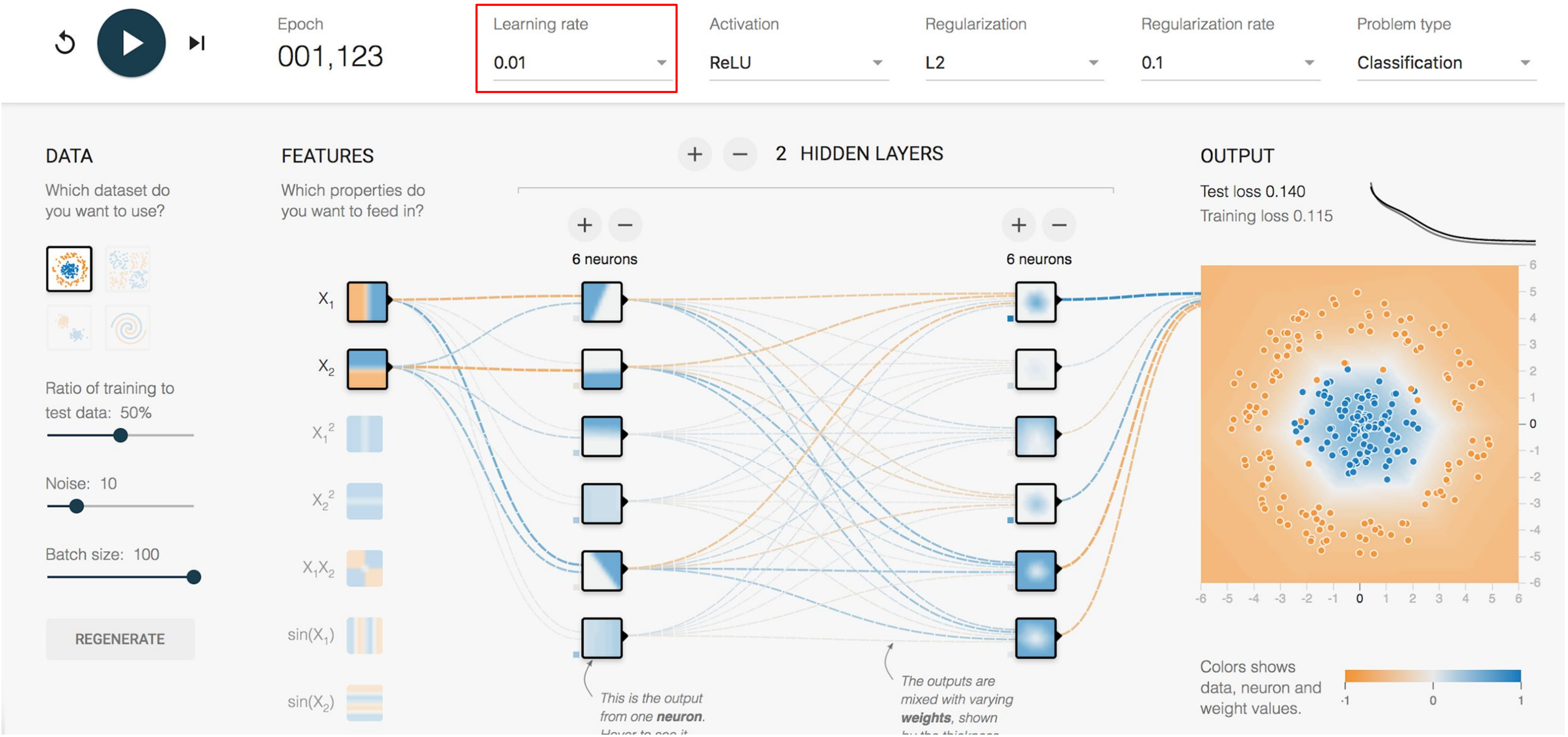
How can you define model complexity?



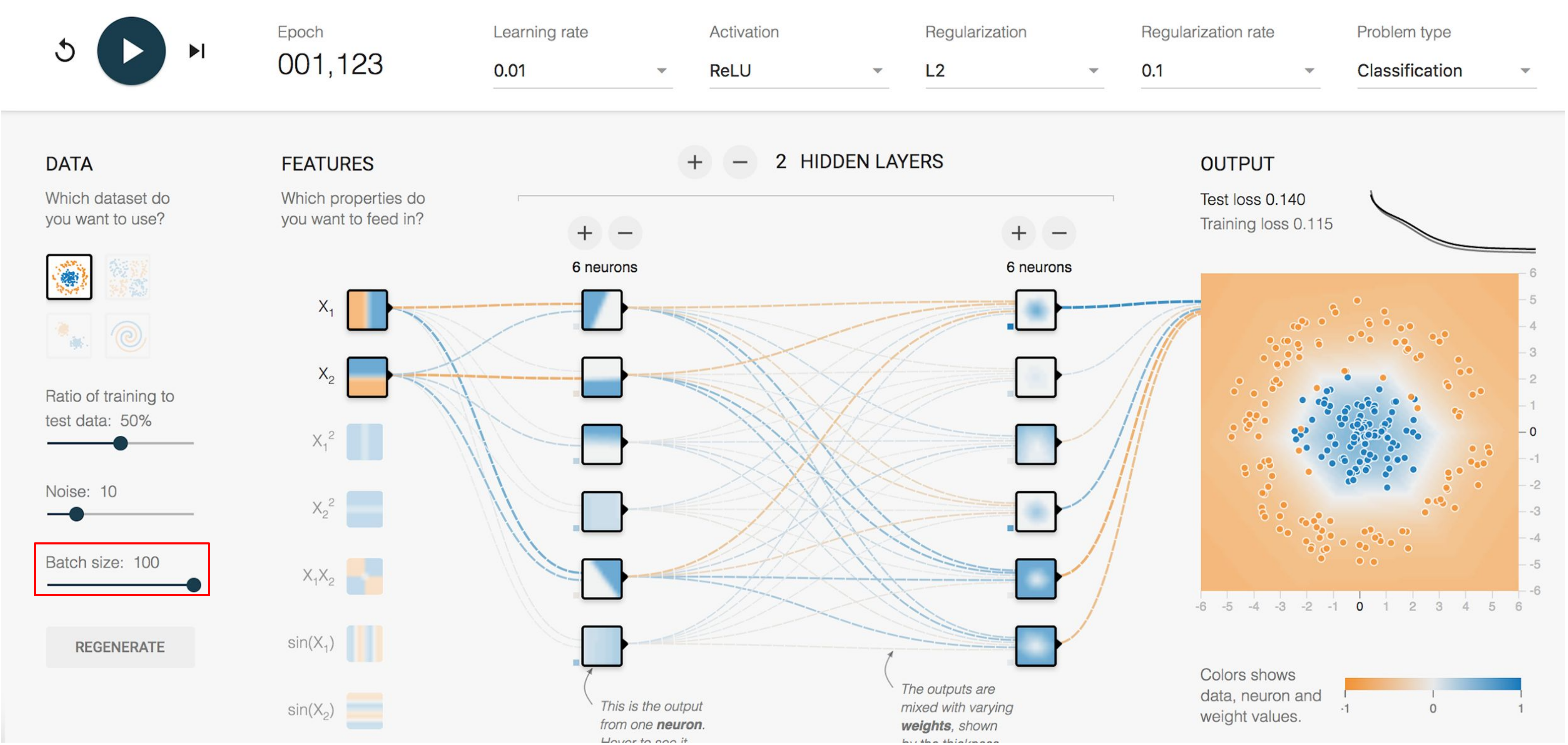
How can you define model complexity?



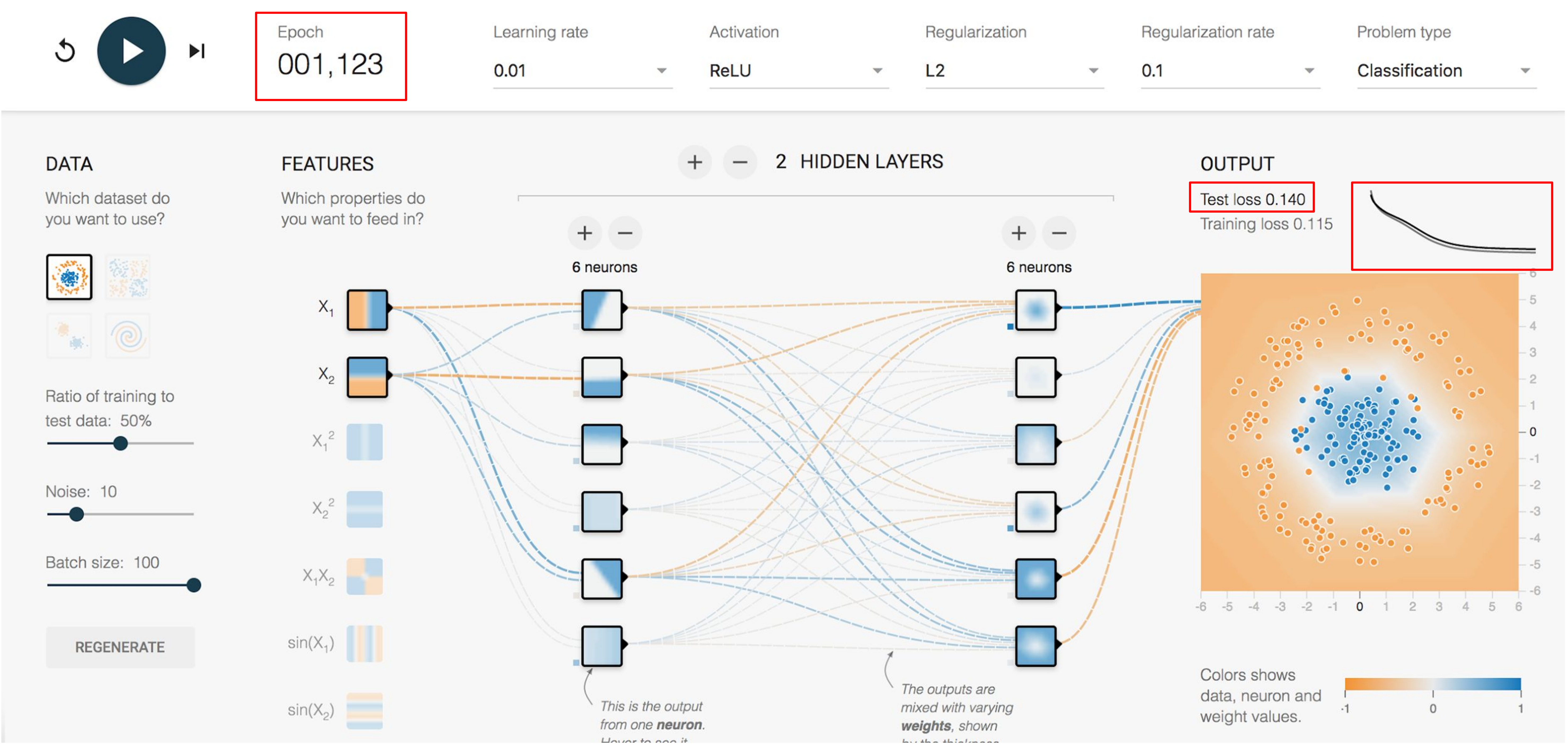
How can you define model complexity?



How can you define model complexity?



How can you define model complexity?



How can you define model complexity?

We have several knobs
that are dataset-dependent

Learning rate controls the size of the step in weight space

If too small, training will take a long time

If too large, training will bounce around

Default learning rate in Estimator's LinearRegressor is smaller of 0.2 or 1/sqrt(num_features) -- this assumes that your feature and label values are small numbers

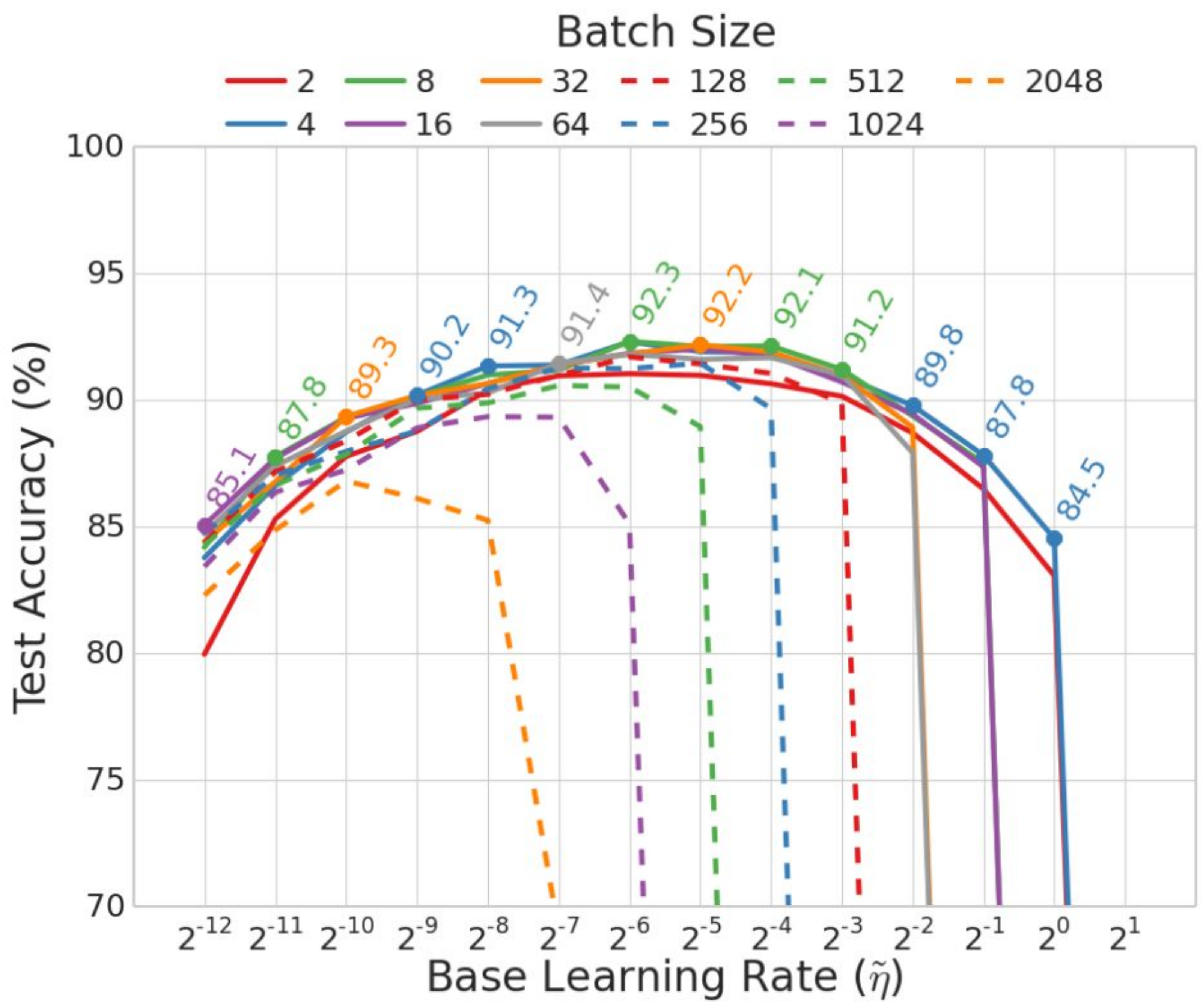
The batch size controls the number of samples that gradient is calculated on.

If too small, training will bounce around

If too large, training will take a very long time

40-100 tends to be a good range for batch size
Can go up to as high as 500

Larger batch sizes require smaller learning rates



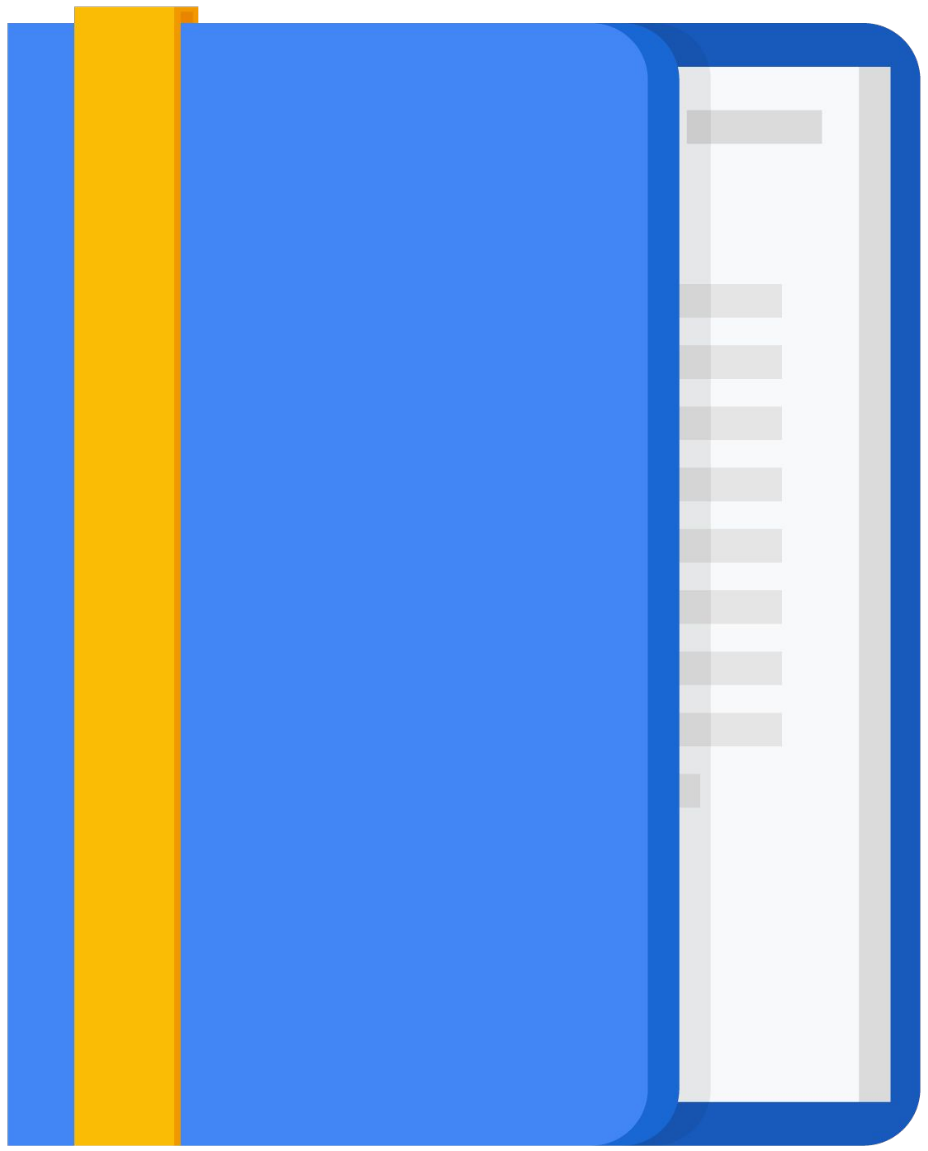
Revisiting Small Batch Training for Deep Neural Networks, Masters and Luschi, 2018

Agenda

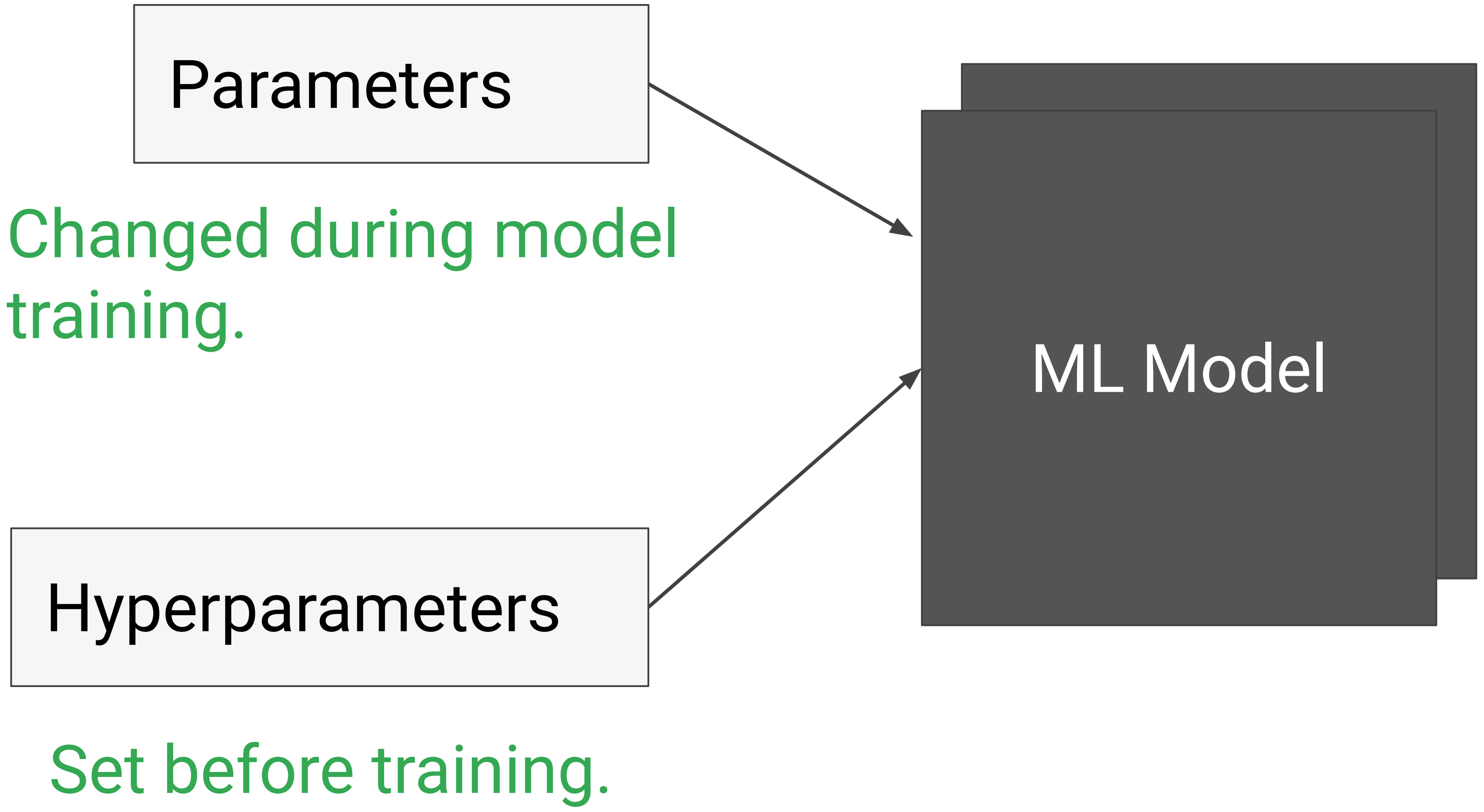
Learning Rate and Batch Size

Hyperparameter Tuning

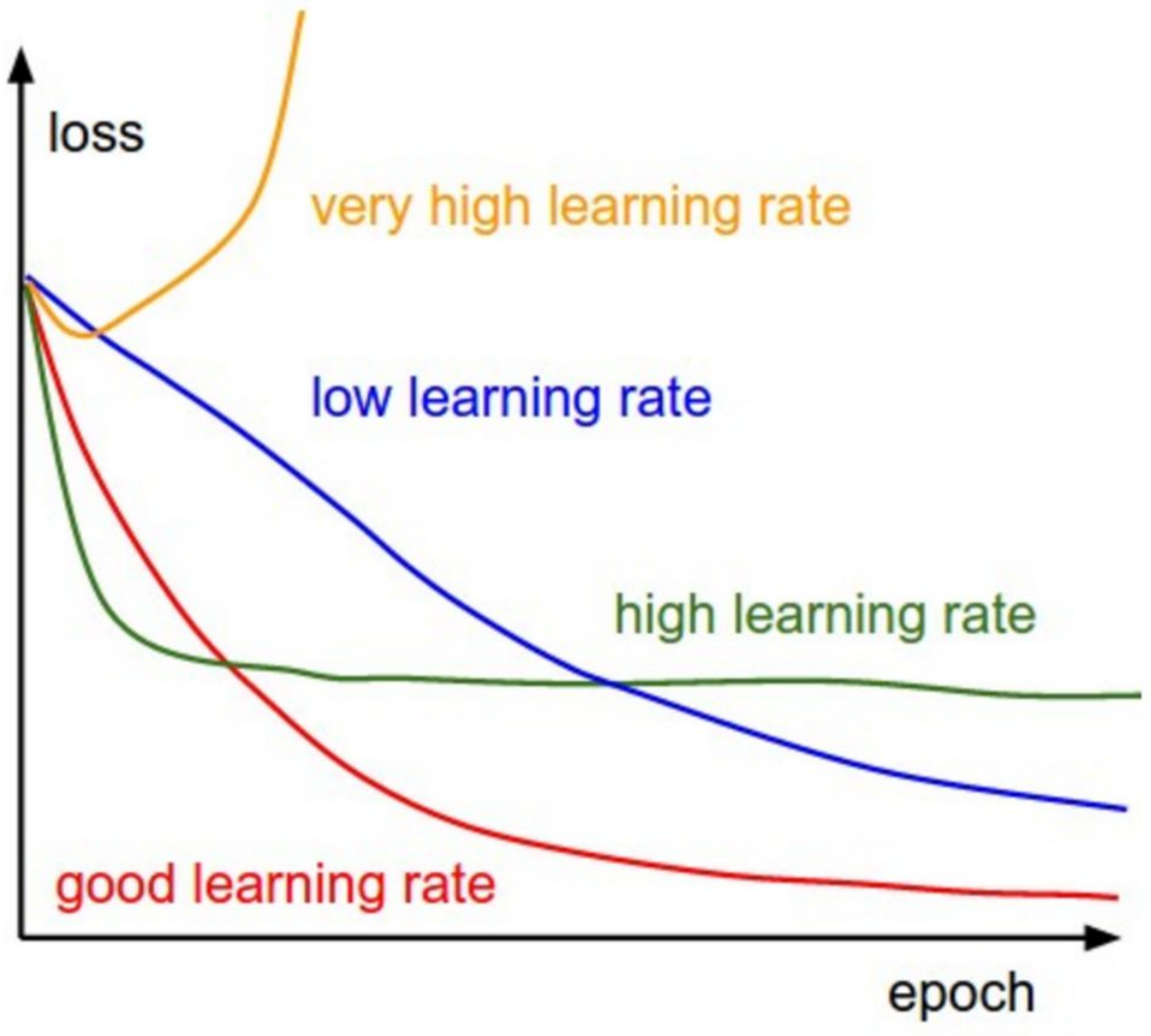
AI Explanations

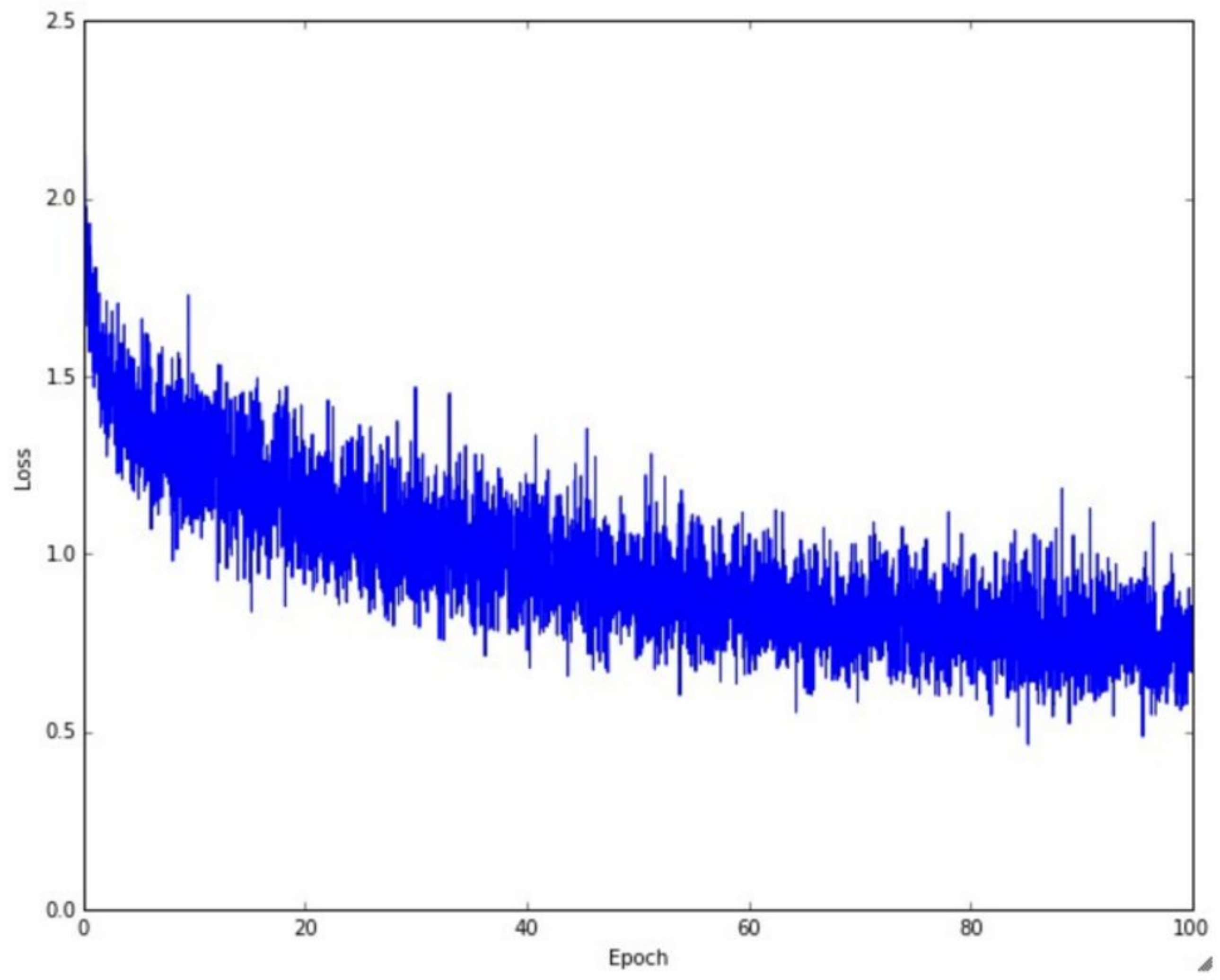


ML models are mathematical functions with parameters and hyperparameters



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?

Fear 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.

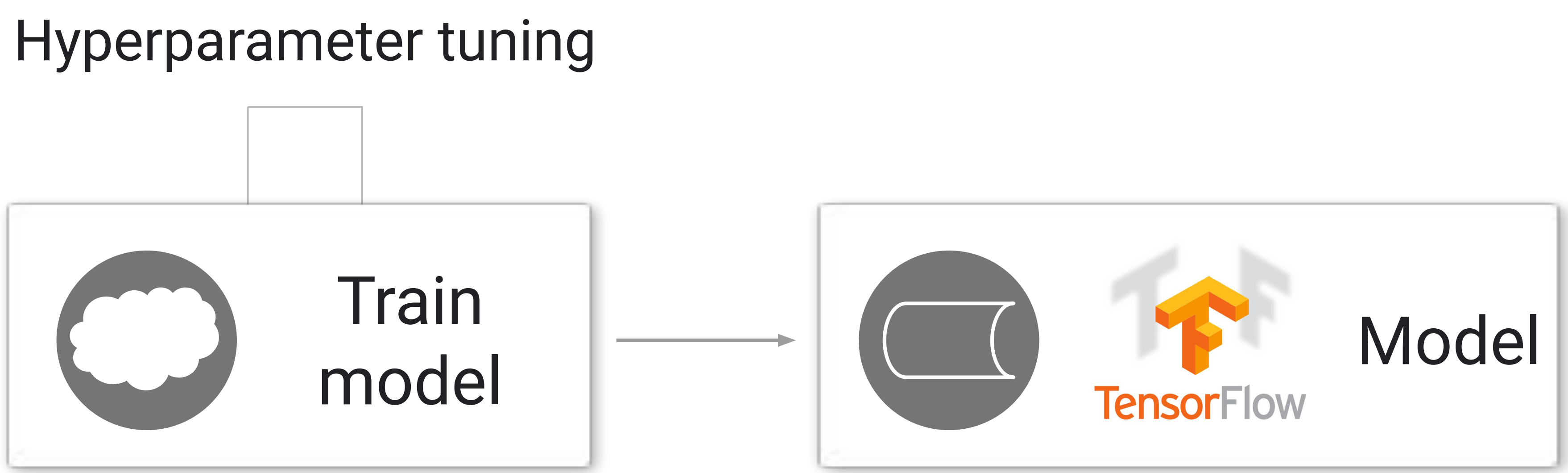
1.1 Related Work

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 AI Platform 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.

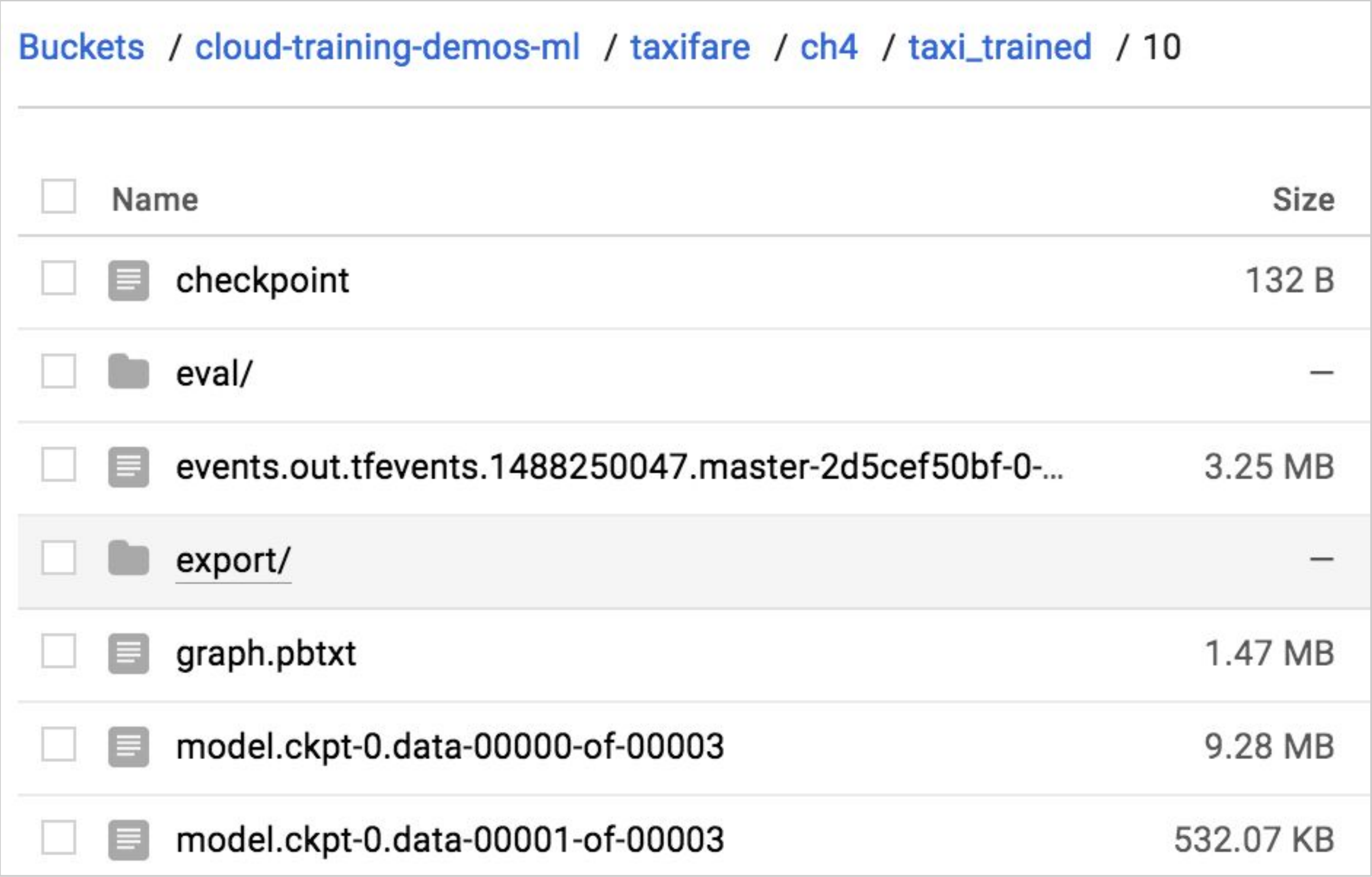


1. Make the hyperparameters as command-line arguments

```
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]
)
```

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', '')
 )
```



3. Supply hyperparameters to training job

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

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

taxifare_211217_120117

Succeeded (57 min 26 sec)

Creation time	Dec 17, 2021, 1:01:19 PM
Start time	Dec 17, 2021, 1:01:22 PM
End time	Dec 17, 2021, 1:58:45 PM
Logs	View Logs
TensorBoard	TensorBoard is available from this page only for models trained with built-in TensorFlow algorithms
Consumed ML units	0.64
Training input	SHOW JSON
Training output	SHOW JSON

HyperTune trials

Filter Enter property name or value

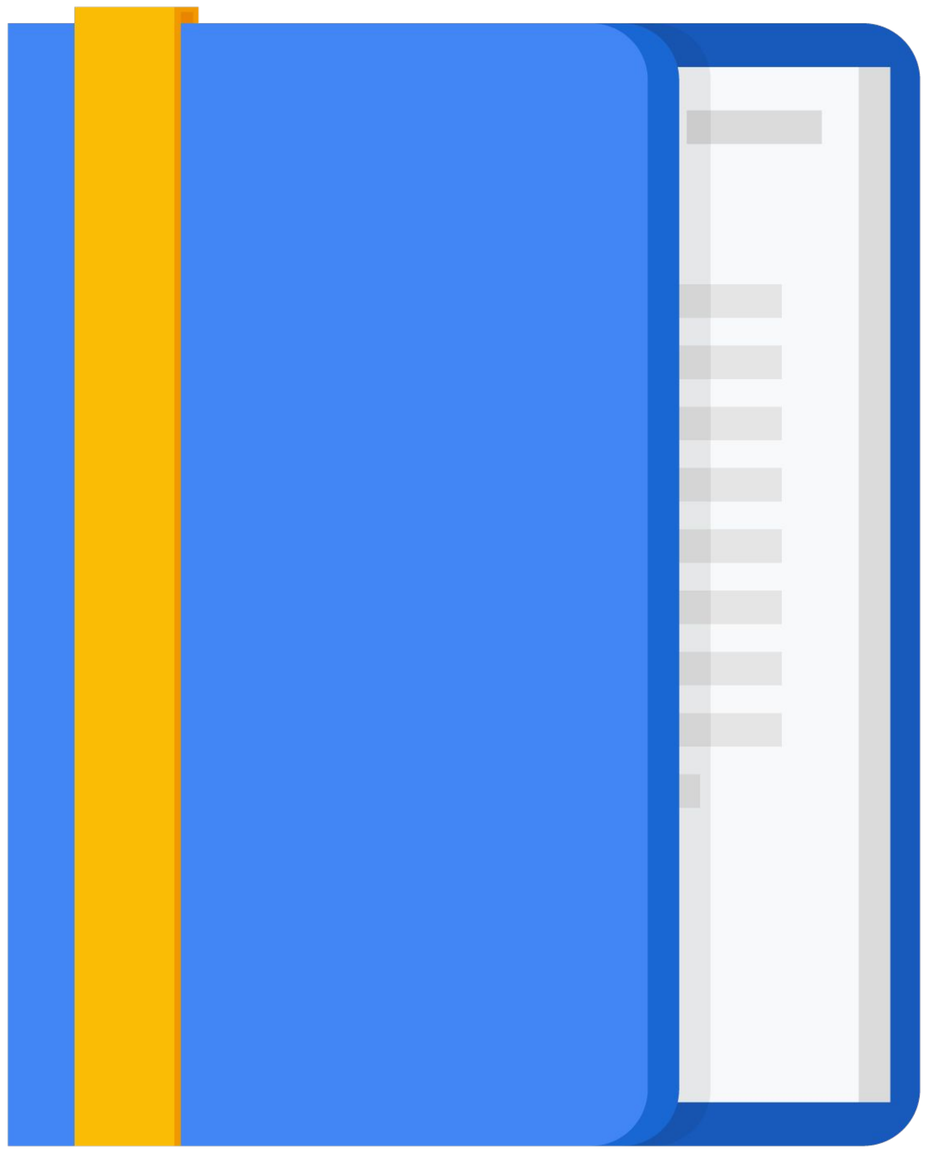
Trial ID	rmse (Objective)	Training step	Elapsed time	lr	nbuckets	batch_size
3	10.96529	10	10 min 9 sec	0.0037	18	50
1	11.70956	10	10 min 9 sec	0.00316	18	30
7	12.6772	10	10 min 27 sec	0.00401	11	30
10	13.03278	10	10 min 9 sec	0.00416	19	50
4	13.78874	10	10 min 29 sec	0.00273	18	15
2	13.87329	10	10 min 30 sec	0.00072	14	15
8	13.96504	10	10 min 58 sec	0.00355	25	50
6	13.96598	10	10 min 10 sec	0.00101	23	50
9	14.1489	10	10 min 19 sec	0.00276	12	50
5	28.39961	10	10 min 9 sec	0.02226	15	50

Agenda

Learning Rate and Batch Size

Hyperparameter Tuning

AI Explanations

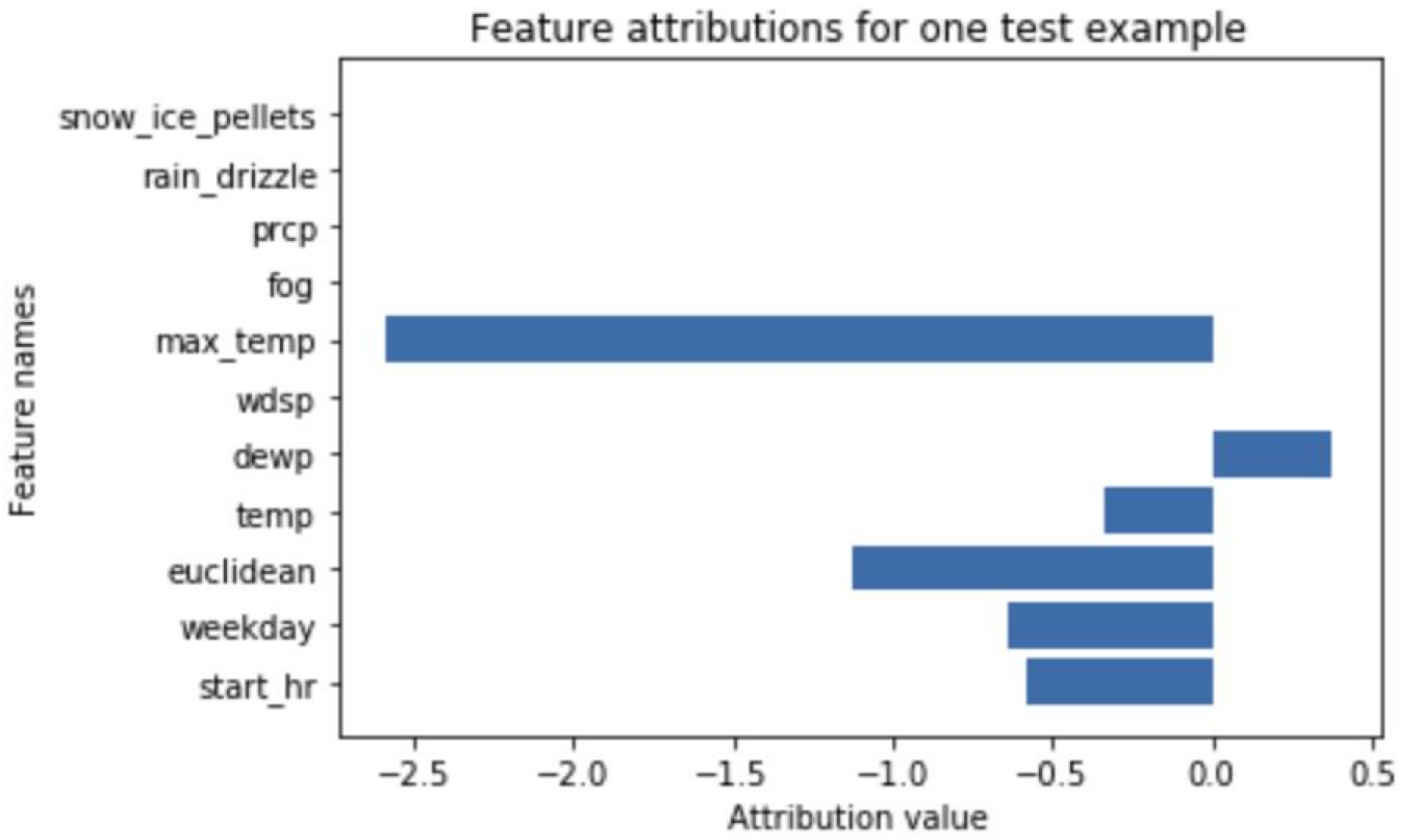


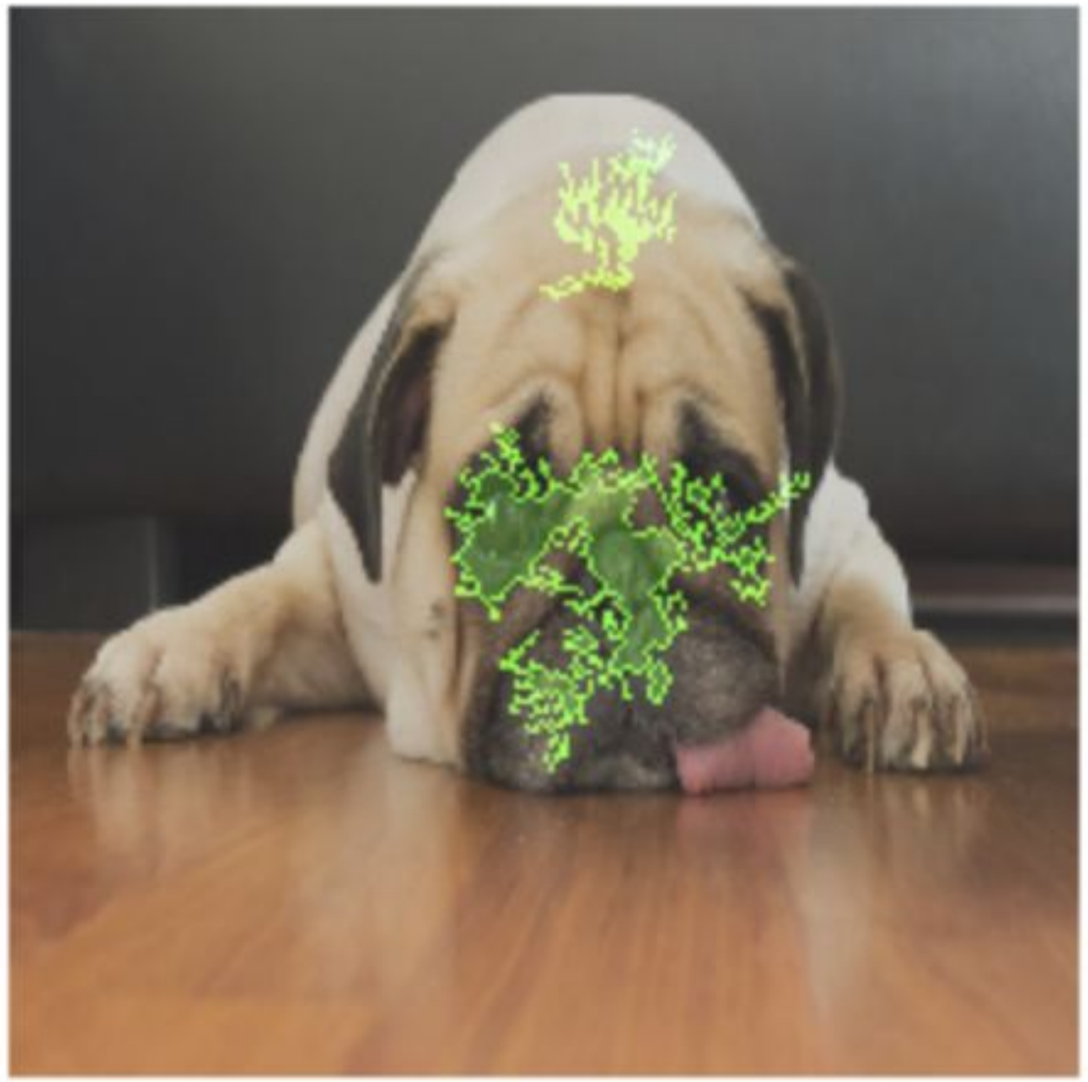
AI Explanations overview

- Tells you how much each feature contributes in the predicted result
- Useful to improve model prediction (removing useless features)
- Helps to recognize bias
- Works for both structured and unstructured dataset and non-linear problems

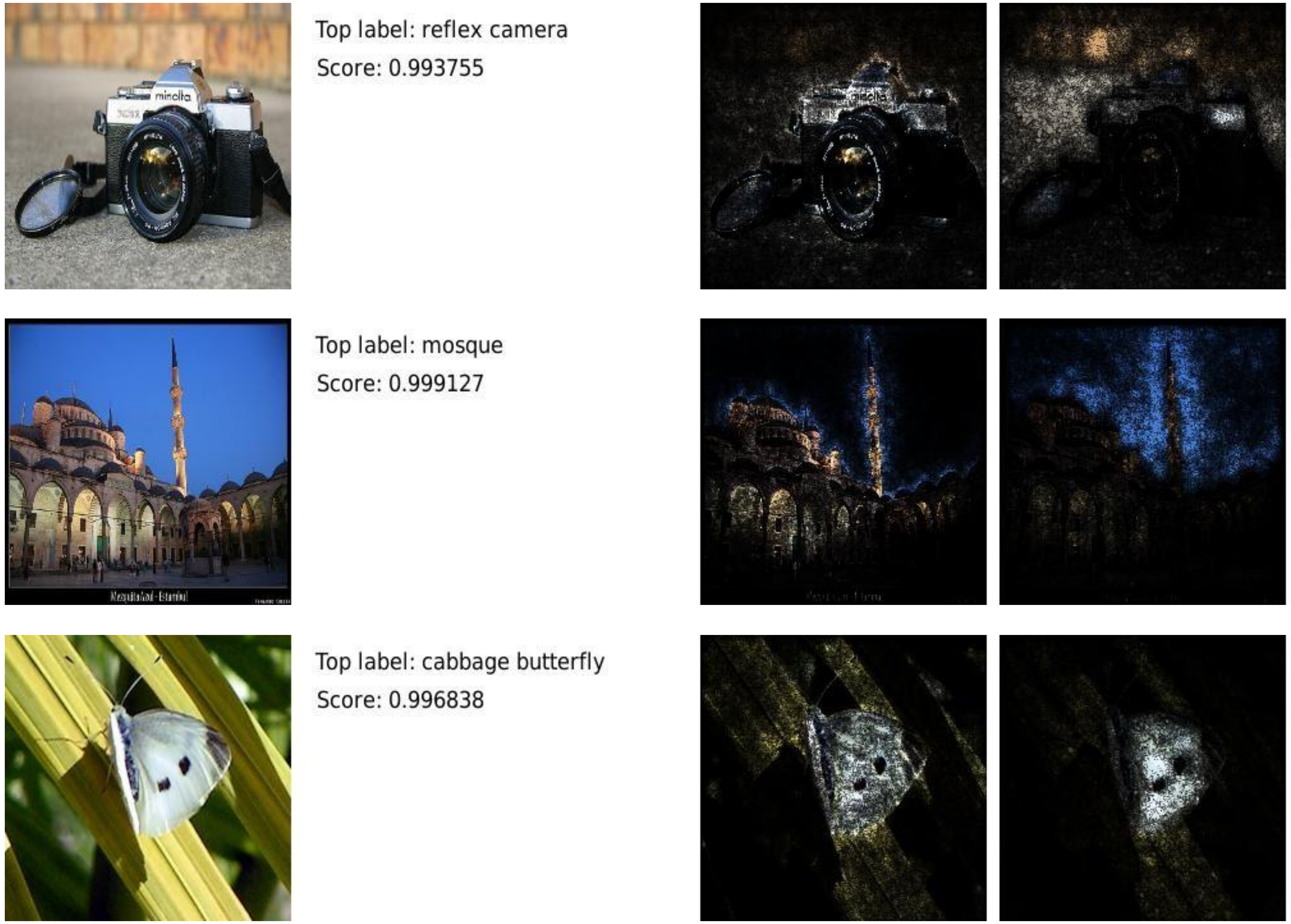
Feature attributions

```
Predicted duration: 11.1651134 minutes
Actual duration: 10.0 minutes
```





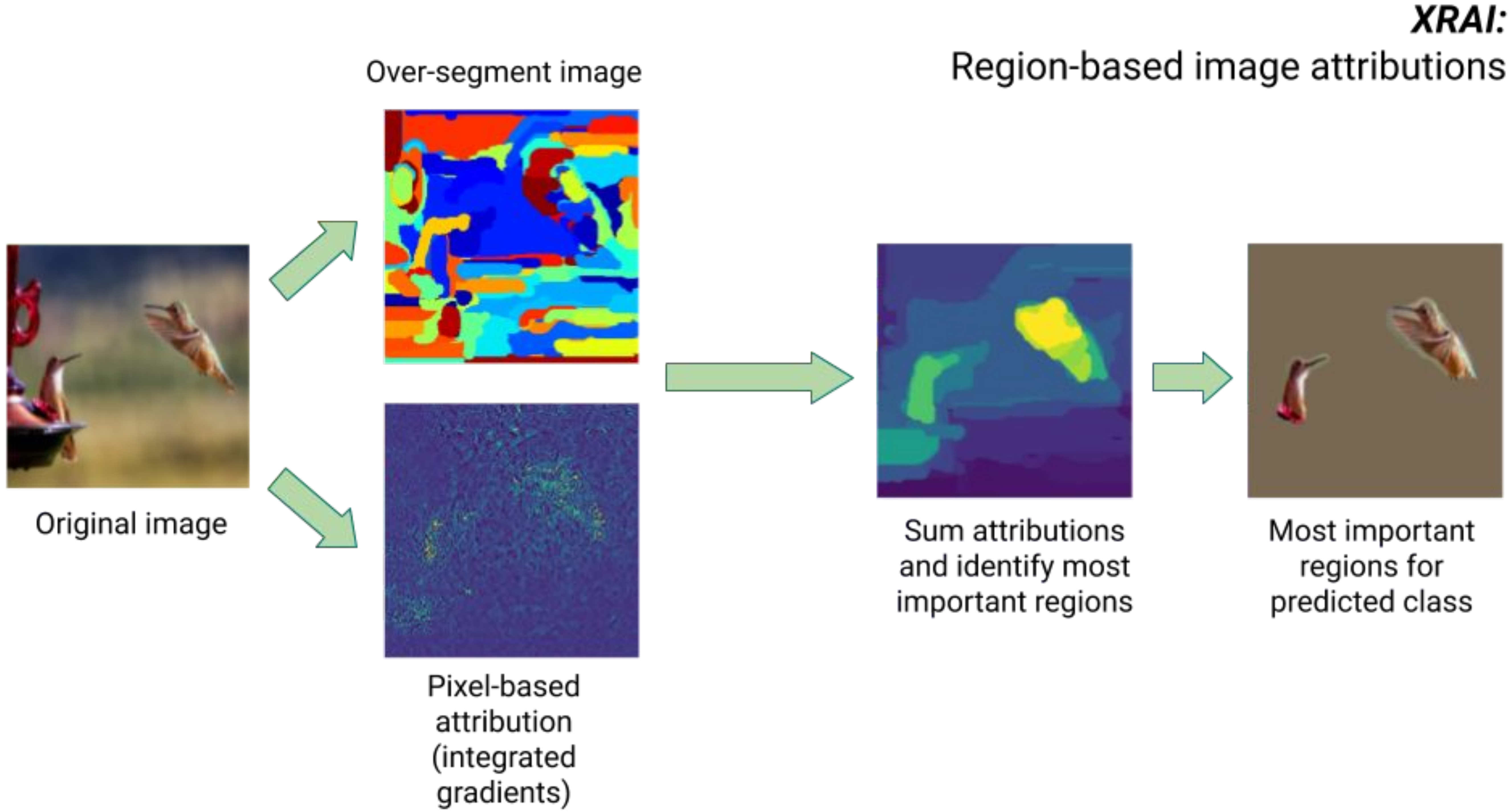
Feature attributions: Integrated Gradients



Integrated Gradients… When to use it

- Differentiable models such as NNs
- Models with large feature spaces
- Low-contrast images such as X-rays
- Works with both classification and regression model on tabular dataset
- Works with classification problem on image dataset

Feature attributions: XRAI (eXplanation Ranked Area Integrals)



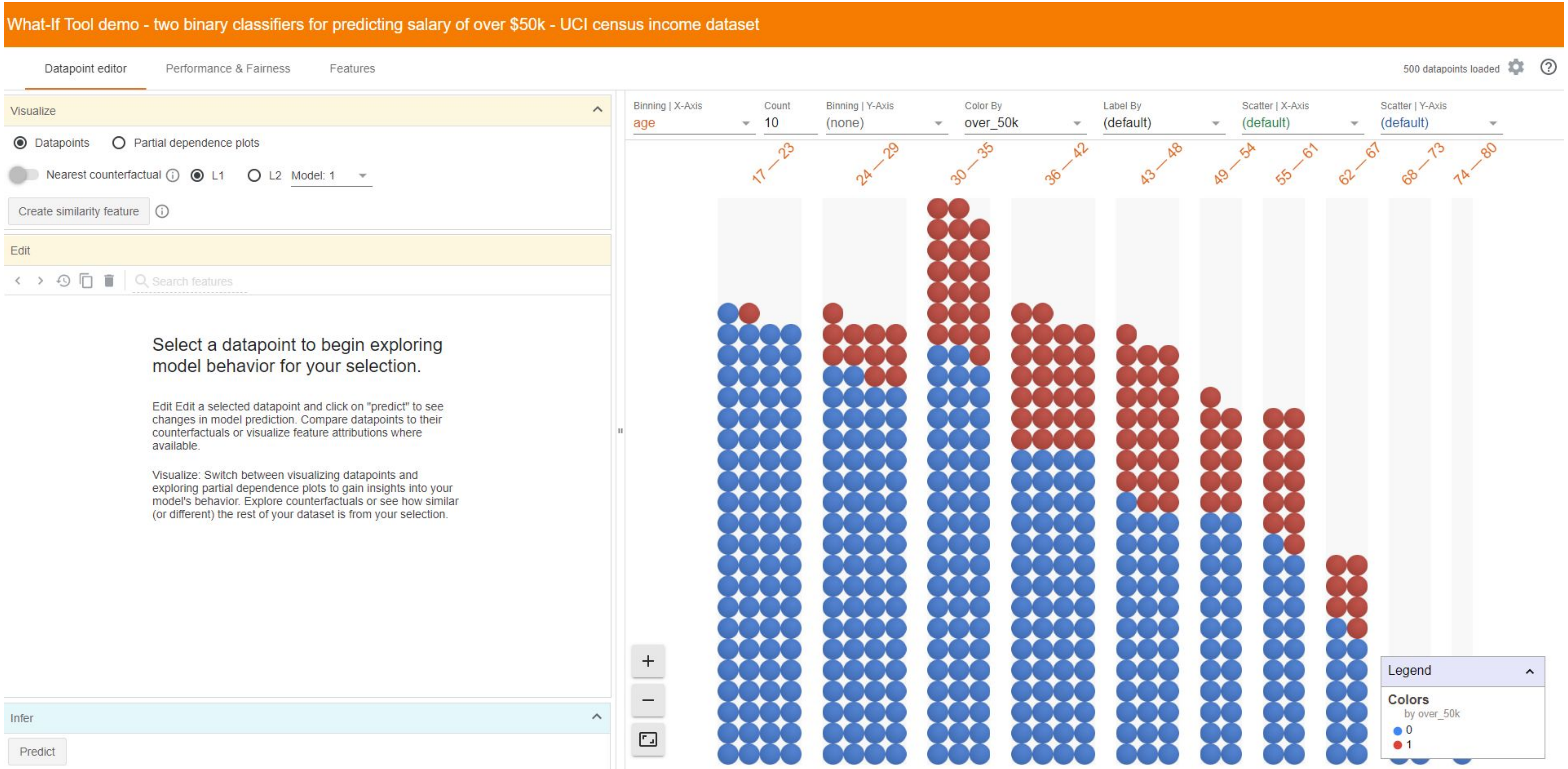
XRAI… When to use it

- Works for only Image Classification problem
- Works for any type of images
- Works well when you need to classify multiple objects

What-If-Tool

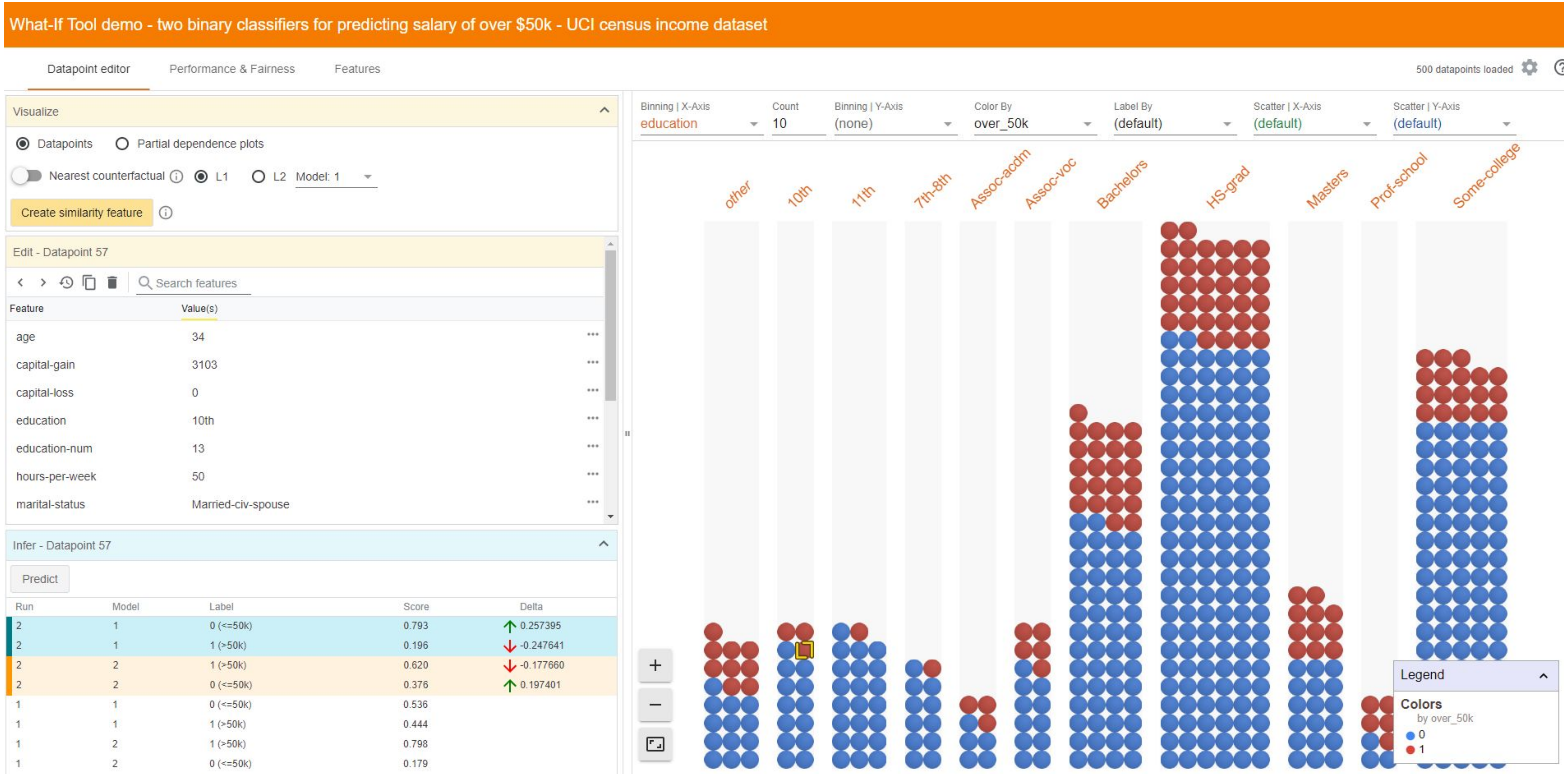
- Great visual tool for monitoring model performance and predictions
- Used for structured dataset and classification and regression model
- Can tell you how much a feature contributes on the model prediction
- Can be used to look about model performance and fairness

What-If-Tool… Facet Dive

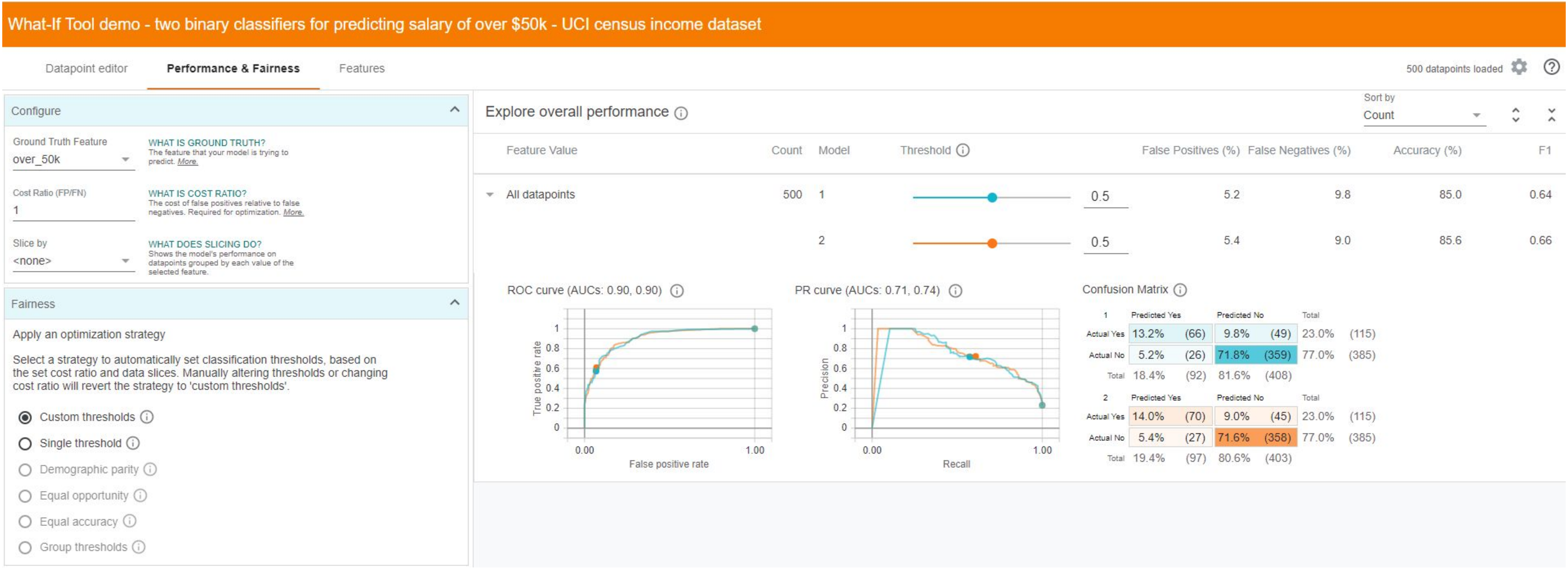


Google Cloud

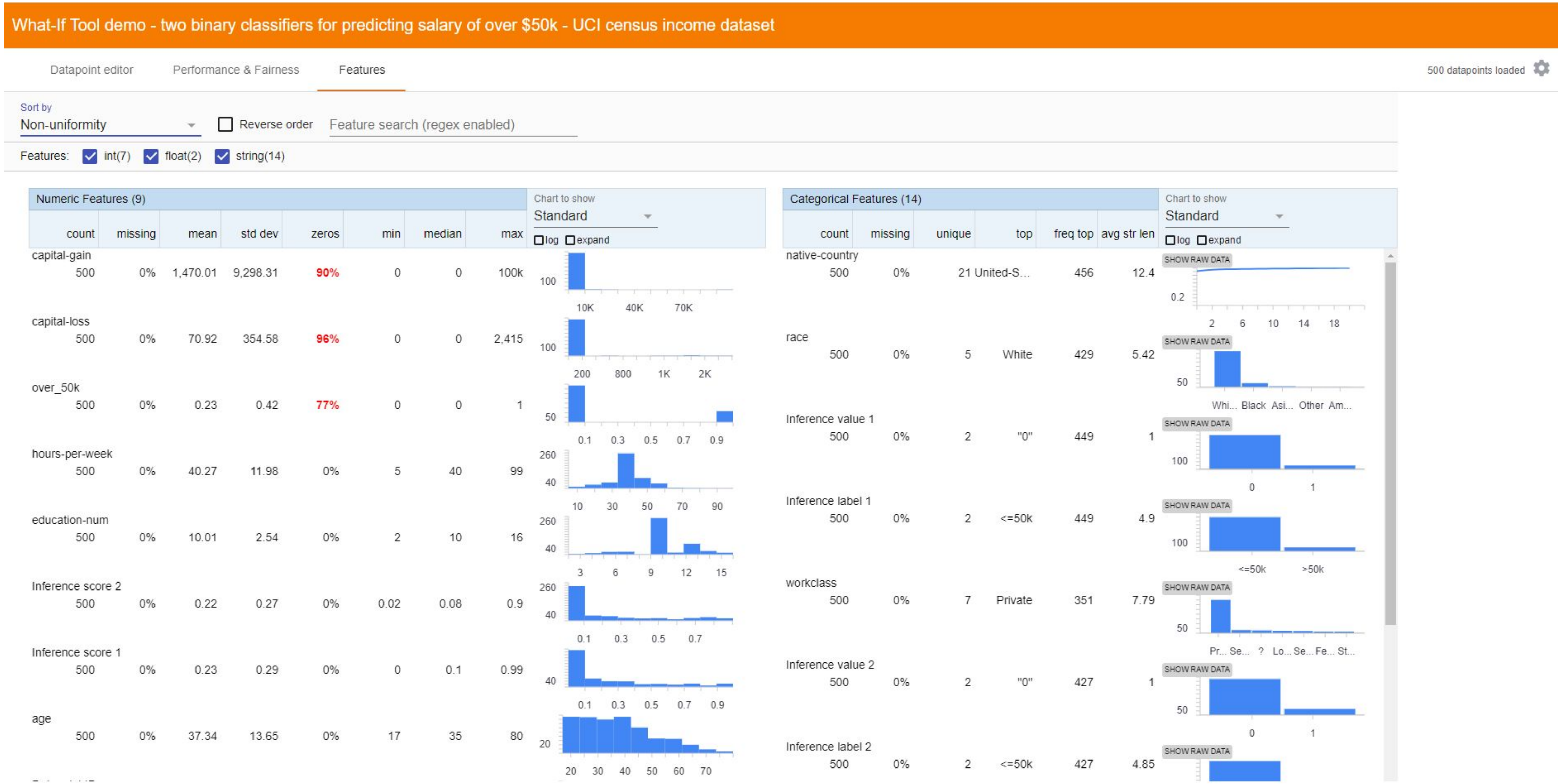
What-If-Tool… Feature contribution



What-If-Tool… Performance & Fairness



What-If-Tool… Feature Distribution



Thank you
End of Session 5

Google Cloud