GridGain

Achieving Continuous Machine and Deep Learning with Apache Ignite and TensorFlow

Yuriy Babak

Speaker

Yuriy Babak

- Head of ML/DL framework development at GridGain
- Apache Ignite committer

Agenda

- ML introduction
- Apache Ignite ML overview
- Integration with TensorFlow
- Demo

Apache Ignite ML overview

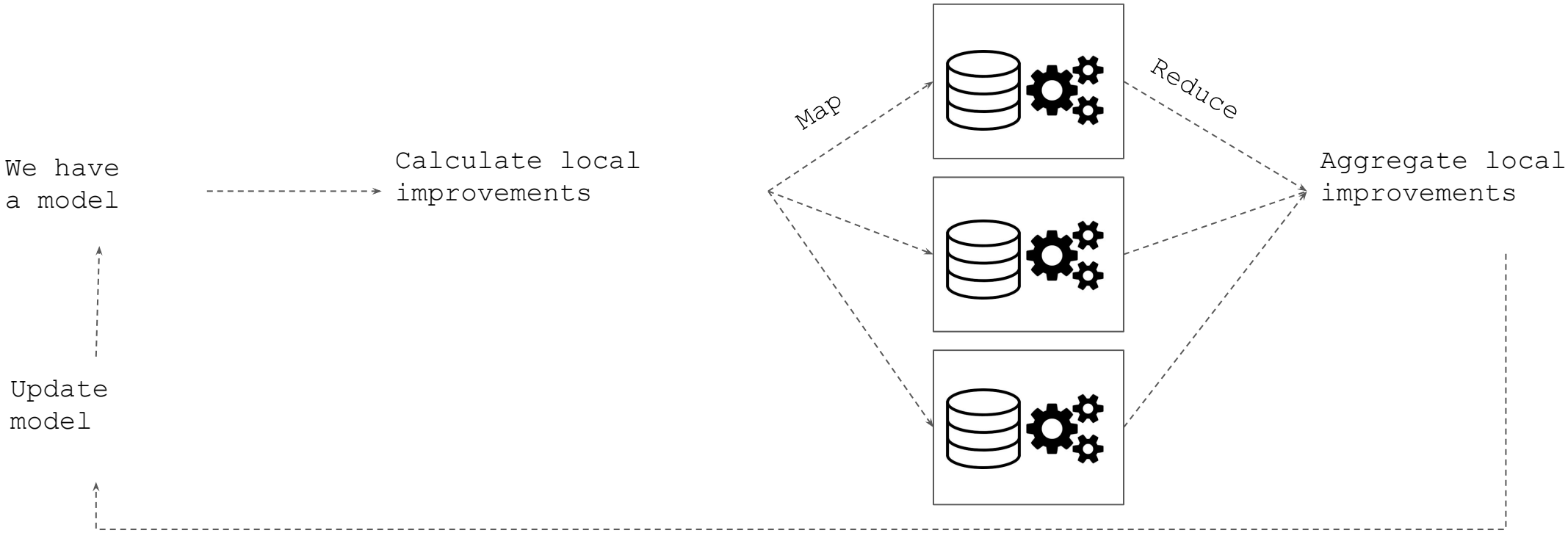
ML Terms

- **Raw Data** - data of business (logs, transactions, …)
- **Training/Test set** - preprocessed data from raw data, numeric features (set: age, user type id, gender, ...)
- **Training algorithm** - produce model by training set (GDB, Gibbs Sampling, Decision Tree building)
- **Model** - formula for producing answers on new business data (decision tree, SVM, linear regression)
- **Meta algorithm** - combinator of several models (boosting, bagging, stacking)
- **Evaluation/Validation** - estimation of model given test/validation set (Precision, MSE, ROC AUC)
- **Inference** - using ML model for prediction

Typical ML tasks

1. Regression
2. Time-series regression
3. Classification (binary, multiclass)
4. Clusterization
5. Ranking
6. Entity extraction, structure recognition
7. Generative models

Distributed model training

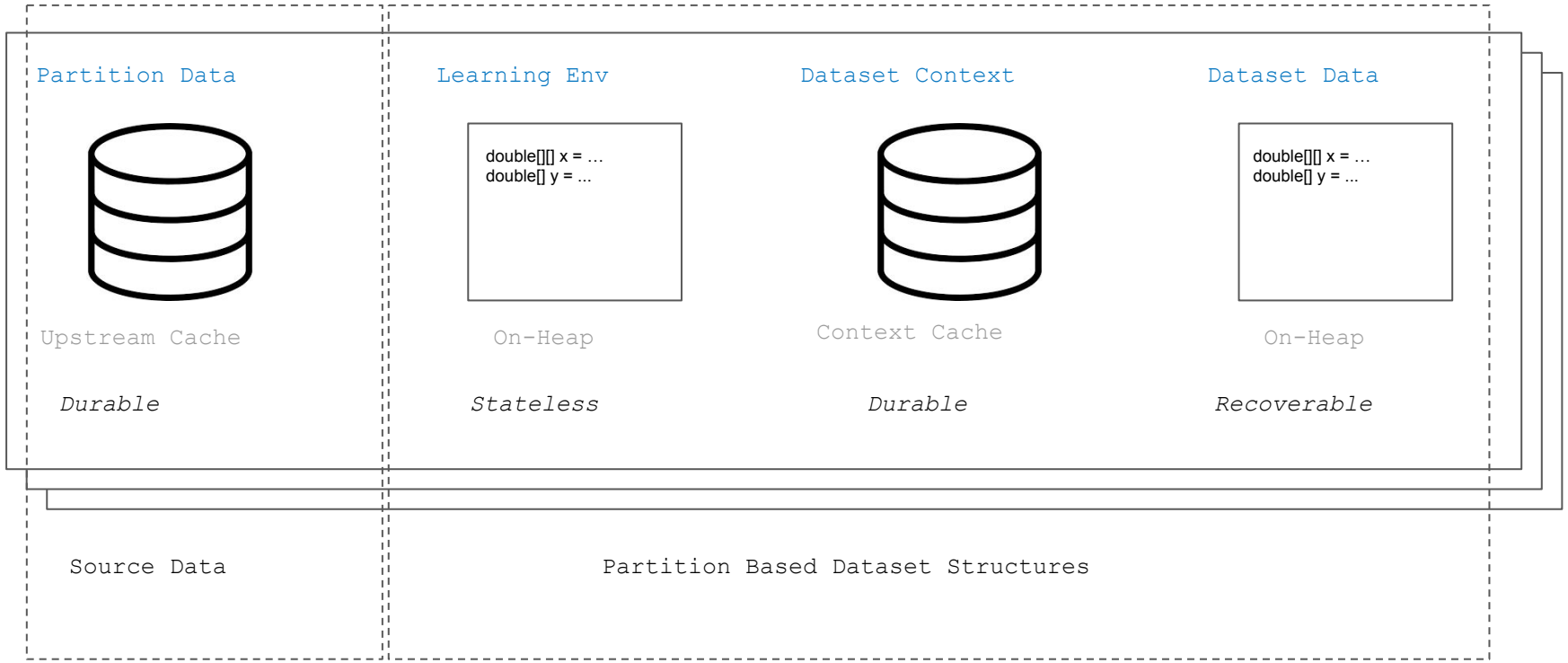


$$\nabla[\sum_{1}^{n}(y-\hat{y})^2] = \nabla[\sum_{1}^{n_1}(y-\hat{y})^2] + \nabla[\sum_{n_1}^{n_2}(y-\hat{y})^2] + \ldots + \nabla[\sum_{n_{k-1}}^{n}(y-\hat{y})^2]$$

Ignite partition-based dataset

Dataset dataset = … // Partition based dataset, internal API

dataset.compute((env, ctx, data) -> map(...), (r1, r2) -> reduce(...))



Apache Ignite ML API

```
LinearRegressionLSQRTrainer trainer = new LinearRegressionLSQRTrainer();

Vectorizer<Integer, Vector, Integer, Double> extractor = new DummyVectorizer<Integer>()
    .labeled(Vectorizer.LabelCoordinate.FIRST);

LinearRegressionModel mdl = trainer.fit(ignite, dataCache, extractor);

double rmse = Evaluator.evaluate(
    dataCache,
    mdl,
    CompositionUtils.asFeatureExtractor(extractor),
    CompositionUtils.asLabelExtractor(extractor),
    new RegressionMetrics()
);
```

List pre-build models

Regression: Linear Regression (LSQR and SGD), Decision Tree, Random Forest, GBT, Nearest Neighbours (KNN), MLP.

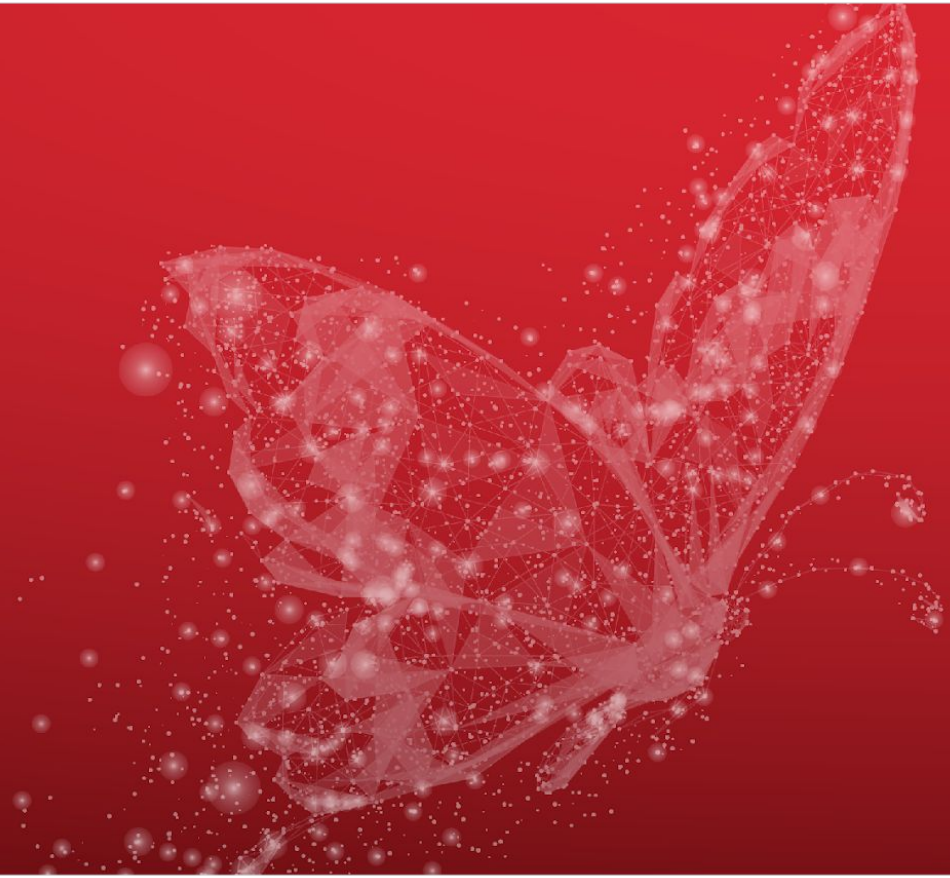
Classification: SVM, Nearest Neighbours (ANN, KNN), Decision Tree, MLP, GBT, Logistic Regression.

Clustering: K-Means, Gaussian Mixture Model (GMM).

Preprocessing: Normalization, Binarization, Imputer, One-Hot Encoder, String Encoder, MinMax Scaler, MaxAbsScaler.

Ensambles: Boosting, Stacking, Bagging of any model.

TensorFlow integration



Apache Ignite as data source

```
    Returns:
      A dataset that can be used for iteration.
    """
-   filenames = get_filenames(is_training, data_dir)
-   dataset = tf.data.FixedLengthRecordDataset(filenames, _RECORD_BYTES)
+   dataset = IgniteDataset("TEST_DATA", local=True).map(lambda row: row['val'])

    return resnet_run_loop.process_record_dataset(
        dataset=dataset,
```

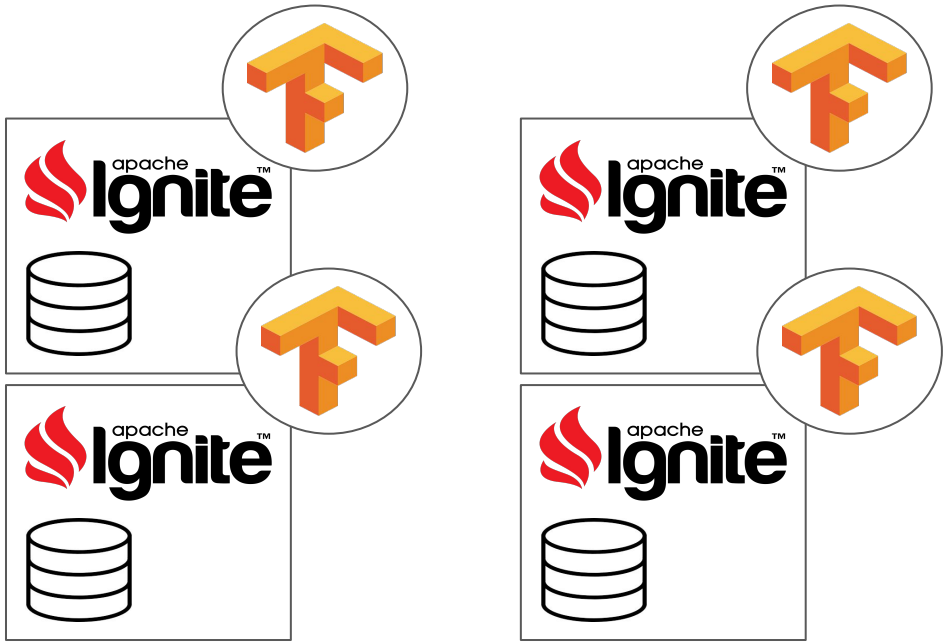
Distributed training

$$\nabla[\sum_{1}^{n}(y-\hat{y})^2] = \nabla[\sum_{1}^{n_1}(y-\hat{y})^2] + \nabla[\sum_{n_1}^{n_2}(y-\hat{y})^2] + \ldots + \nabla[\sum_{n_{k-1}}^{n}(y-\hat{y})^2]$$

ignite-tf.sh

Tool that help to submit TensorFlow code into cluster and manage it.

TensorFlow/Ignite Client

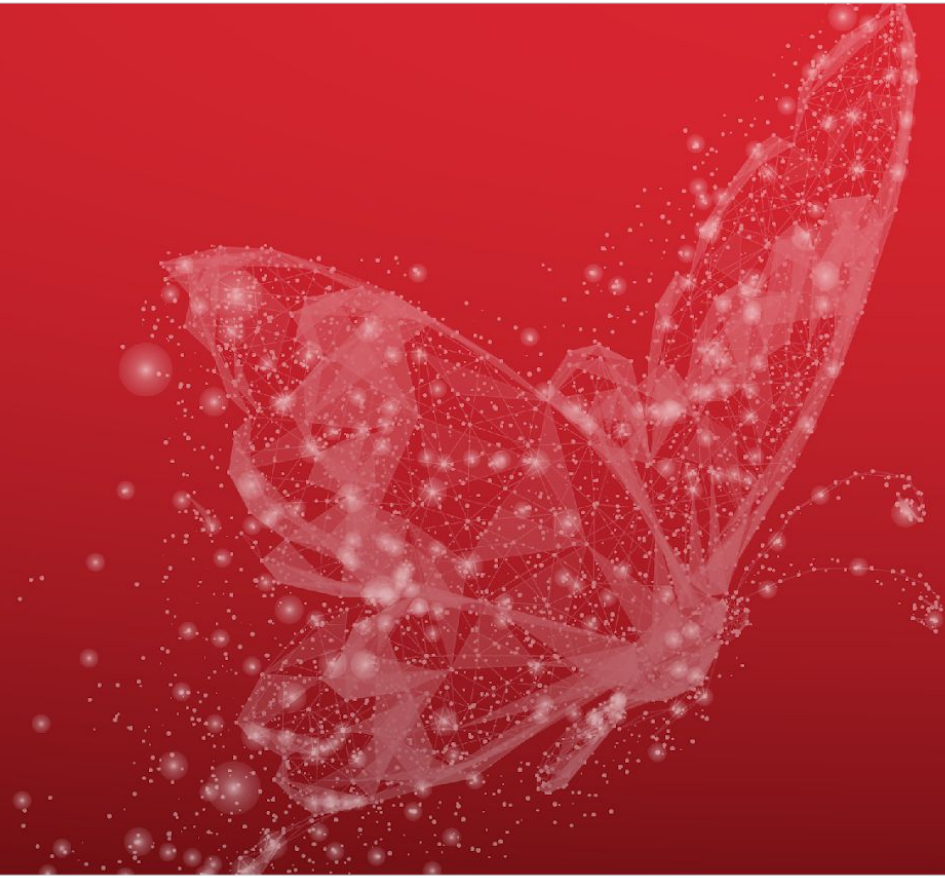


TensorFlow Cluster on top of Apache Ignite Cluster

Distributed training

```
   run_config = tf.estimator.RunConfig(
-      train_distribute=distribution_strategy,
+      experimental_distribute=tf.contrib.distribute.DistributeConfig(
+        train_distribute=tf.contrib.distribute.CollectiveAllReduceStrategy(),
+        eval_distribute=tf.contrib.distribute.MirroredStrategy(),
+        remote_cluster=json.loads(os.environ['TF_CLUSTER'])
+      ),
```

Demo



Demo

Training and inference for pre-build models

Demo

TensorFlow model inference from Apache Ignite

Links

- Apache Ignite docs - https://apacheignite.readme.io/docs
- Apache Ignite sources - https://github.com/apache/ignite
- TensorFlow IO module - https://github.com/tensorflow/io/tree/master/tensorflow_io/ignite

Q&A

Attend the In-Memory Computing Summit

https://www.imcsummit.org/

In-Memory Computing Summit **Europe**

- Next event in London, June 2019

In-Memory Computing Summit **North America**

- Next event in Silicon Valley, Nov. 2019