Feeding ML models with the data from the databases in real-time

Vojtěch Juránek

Red Hat

Feb. 3rd 2024, FOSDEM, Brussels
MACHINE LEARNING ENGINEERING

DATA PIPELINE

1. Exploration & Validation
2. Wrangling (Cleaning)
3. NEW data from model performance

DATA

TRAIN
TEST

MODEL ENGINEERING
- Feature engineering
- Hyperparameters tuning

MODEL EVALUATION
- Best model selection
- Model performance metrics

MODEL PACKAGING
- Model format
  - ONNX
  - PPL

MODEL

CODE
- Trunk based dev
- Code versioning

BUILD

INTEGRATION TESTING

MONITORING & LOGGING
- Model decay trigger

SOFTWARE CODE PIPELINE

Source: https://ml-ops.org/content/end-to-end-ml-workflow
- Consistent data, no data losses, no dual writes.
- Get all the changes without any delay in the real-time.
- Not overload the DB with the queries.
Change data capture (CDC)
For more details see

- Image classification with Debezium and TensorFlow blog post
- Full example on GitHub
Debezium configuration

```json
{
    "name": "mnist-connector",
    "config": {
        "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
        "tasks.max": "1",
        "database.hostname": "postgres",
        "database.port": "5432",
        "database.user": "postgres",
        "database.password": "postgres",
        "database.dbname": "postgres",
        "topic.prefix": "tf",
        "table.include.list": "public.mnist_.*",
        "key.converter": "org.apache.kafka.connect.storage.StringConverter",
        "value.converter": "org.apache.kafka.connect.storage.StringConverter",
        "transforms": "unwrap, mnist",
        "transforms.unwrap.type": "io.debezium.transforms.ExtractNewRecordState",
        "transforms.mnist.type": "io.debezium.transforms.MnistToCsv"
    }
}
```
# define function for decoding Kafka records

def decode_kafka_stream_record(message, key):
    img_int = tf.io.decode_csv(message, [[0.0] for i in range(NUM_COLUMNS)])

    img_norm = tf.cast(img_int, tf.float32) / 255.
    label_int = tf.strings.to_number(key, out_type=tf.dtypes.int32)
    return (img_norm, label_int)

# define Kafka data stream

test_ds = tfio.experimental.streaming.KafkaGroupIODataset(
    topics=[KAFKA_TEST_TOPIC],
    group_id=KAFKA_CONSUMER_GROUP,
    servers=KAFKA_SERVERS,
    stream_timeout=KAFKA_STREAM_TIMEOUT,
    configuration=[
        "session.timeout.ms=10000",
        "max.poll.interval.ms=10000",
        "auto.offset.reset=earliest"
    ],
)

# read batches of Kafka records

test_ds = test_ds.map(decode_kafka_stream_record)
test_ds = test_ds.batch(BATCH_SIZE)

# make predictions on the data samples

model.evaluate(test_ds)
Thank you!

https://debezium.io
https://debezium.zulipchat.com
https://groups.google.com/g/debezium
https://github.com/debezium
Backup slides
Similar for Apache Spark.

For more details see