Installing TensorFlow for Java

TensorFlow provides APIs for use in Java programs. These APIs are particularly well-suited to loading models created in Python and executing them within a Java application. This guide explains how to install TensorFlow for Java (https://www.tensorflow.org/api_docs/java/reference/org/tensorflow/package-summary) and use it in a Java application.

Warning: The TensorFlow Java API is *not* covered by the TensorFlow API stability guarantees (https://www.tensorflow.org/programmers_guide/version_semantics).

Supported Platforms

This guide explains how to install TensorFlow for Java. Although these instructions might also work on other variants, we have only tested (and we only support) these instructions on machines meeting the following requirements:

- Ubuntu 16.04 or higher; 64-bit, x86
- macOS 10.12.6 (Sierra) or higher
- Windows 7 or higher; 64-bit, x86

The installation instructions for Android are in a separate Android TensorFlow Support page (https://www.github.com/tensorflow/tensorflow/blob/r1.7/tensorflow/contrib/android). After installation, please see this complete example (https://www.github.com/tensorflow/tensorflow/blob/r1.7/tensorflow/examples/android) of TensorFlow on Android.

Using TensorFlow with a Maven project

If your project uses Apache Maven (https://maven.apache.org), then add the following to the project's `pom.xml` to use the TensorFlow Java APIs:

```
<dependency>
  <groupId>org.tensorflow</groupId>
  <artifactId>tensorflow</artifactId>
  <version>1.7.0</version>
</dependency>
```

That's all.

Example

As an example, these steps will create a Maven project that uses TensorFlow:

1. Create the project's **`pom.xml`**:

```
<project>
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.myorg</groupId>
    <artifactId>hellotf</artifactId>
    <version>1.0-SNAPSHOT</version>
    <properties>
      <exec.mainClass>HelloTF</exec.mainClass>
      <!-- The sample code requires at least JDK 1.7. -->
      <!-- The maven compiler plugin defaults to a lower version -->
      <maven.compiler.source>1.7</maven.compiler.source>
      <maven.compiler.target>1.7</maven.compiler.target>
    </properties>
    <dependencies>
      <dependency>
        <groupId>org.tensorflow</groupId>
        <artifactId>tensorflow</artifactId>
        <version>1.7.0</version>
      </dependency>
    </dependencies>
</project>
```

2. Create the source file (**`src/main/java/HelloTF.java`**):

```
import org.tensorflow.Graph;
import org.tensorflow.Session;
import org.tensorflow.Tensor;
import org.tensorflow.TensorFlow;
```

```
public class HelloTF {
  public static void main(String[] args) throws Exception {
    try (Graph g = new Graph()) {
      final String value = "Hello from " + TensorFlow.version();

      // Construct the computation graph with a single operation, a constan
      // named "MyConst" with a value "value".
      try (Tensor t = Tensor.create(value.getBytes("UTF-8"))) {
        // The Java API doesn't yet include convenience functions for addin
        g.opBuilder("Const", "MyConst").setAttr("dtype", t.dataType()).setA
      }

      // Execute the "MyConst" operation in a Session.
      try (Session s = new Session(g);
           Tensor output = s.runner().fetch("MyConst").run().get(0)) {
        System.out.println(new String(output.bytesValue(), "UTF-8"));
      }
    }
  }
}
```

3. Compile and execute:

```
# Use -q to hide logging from the mvn tool
mvn -q compile exec:java
```

The preceding command should output `Hello from version`. If it does, you've successfully set up TensorFlow for Java and are ready to use it in Maven projects. If not, check Stack Overflow (http://stackoverflow.com/questions/tagged/tensorflow) for possible solutions. You can skip reading the rest of this document.

GPU support

If your Linux system has an NVIDIA® GPU and your TensorFlow Java program requires GPU acceleration, then add the following to the project's **`pom.xml`** instead:

```
<dependency>
  <groupId>org.tensorflow</groupId>
  <artifactId>libtensorflow</artifactId>
  <version>1.7.0</version>
</dependency>
```

```
<dependency>
  <groupId>org.tensorflow</groupId>
  <artifactId>libtensorflow_jni_gpu</artifactId>
  <version>1.7.0</version>
</dependency>
```

GPU acceleration is available via Maven only for Linux and only if your system meets the requirements for GPU (https://www.tensorflow.org/install/install_linux#determine_which_tensorflow_to_install).

Using TensorFlow with JDK

This section describes how to use TensorFlow using the `java` and `javac` commands from a JDK installation. If your project uses Apache Maven, then refer to the simpler instructions above instead.

Install on Linux or macOS

Take the following steps to install TensorFlow for Java on Linux or macOS:

1. Download libtensorflow.jar (https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow-1.7.0.jar), which is the TensorFlow Java Archive (JAR).
2. Decide whether you will run TensorFlow for Java on CPU(s) only or with the help of GPU(s). To help you decide, read the section entitled "Determine which TensorFlow to install" in one of the following guides:
    - Installing TensorFlow on Linux (https://www.tensorflow.org/install/install_linux#determine_which_tensorflow_to_install)
    - Installing TensorFlow on macOS (https://www.tensorflow.org/install/install_mac#determine_which_tensorflow_to_install)
3. Download and extract the appropriate Java Native Interface (JNI) file for your operating system and processor support by running the following shell commands:

```
TF_TYPE="cpu" # Default processor is CPU. If you want GPU, set to "gpu"
OS=$(uname -s | tr '[:upper:]' '[:lower:]')
mkdir -p ./jni
curl -L \
```

```
  "https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow_j
  tar -xz -C ./jni
```

Install on Windows

Take the following steps to install TensorFlow for Java on Windows:

1. Download libtensorflow.jar (https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow-1.7.0.jar), which is the TensorFlow Java Archive (JAR).
2. Download the following Java Native Interface (JNI) file appropriate for TensorFlow for Java on Windows (https://storage.googleapis.com/tensorflow/libtensorflow/libtensorflow_jni-cpu-windows-x86_64-1.7.0.zip)
   .
3. Extract this .zip file.

Validate the installation

After installing TensorFlow for Java, validate your installation by entering the following code into a file named **`HelloTF.java`**:

```
import org.tensorflow.Graph;
import org.tensorflow.Session;
import org.tensorflow.Tensor;
import org.tensorflow.TensorFlow;

public class HelloTF {
  public static void main(String[] args) throws Exception {
    try (Graph g = new Graph()) {
      final String value = "Hello from " + TensorFlow.version();

      // Construct the computation graph with a single operation, a constant
      // named "MyConst" with a value "value".
      try (Tensor t = Tensor.create(value.getBytes("UTF-8"))) {
        // The Java API doesn't yet include convenience functions for adding ope
        g.opBuilder("Const", "MyConst").setAttr("dtype", t.dataType()).setAttr("
      }
```

```
      // Execute the "MyConst" operation in a Session.
      try (Session s = new Session(g);
           Tensor output = s.runner().fetch("MyConst").run().get(0)) {
        System.out.println(new String(output.bytesValue(), "UTF-8"));
      }
    }
  }
}
```

And use the instructions below to compile and run `HelloTF.java`.

Compiling

When compiling a Java program that uses TensorFlow, the downloaded `.jar` must be part of your `classpath`. For example, you can include the downloaded `.jar` in your `classpath` by using the `-cp` compilation flag as follows:

```
javac -cp libtensorflow-1.7.0.jar HelloTF.java
```

Running

To execute a Java program that depends on TensorFlow, ensure that the following two files are available to the JVM:

- the downloaded `.jar` file
- the extracted JNI library

For example, the following command line executes the `HelloTF` program on Linux and macOS X:

```
java -cp libtensorflow-1.7.0.jar:. -Djava.library.path=./jni HelloTF
```

And the following command line executes the `HelloTF` program on Windows:

```
java -cp libtensorflow-1.7.0.jar;. -Djava.library.path=jni HelloTF
```

If the program prints `Hello from` *`version`*, you've successfully installed TensorFlow for Java and are ready to use the API. If the program outputs something else, check Stack Overflow

(http://stackoverflow.com/questions/tagged/tensorflow) for possible solutions.

Advanced Example

For a more sophisticated example, see LabelImage.java (https://www.github.com/tensorflow/tensorflow/blob/r1.7/tensorflow/java/src/main/java/org/tensorflow/examples/LabelImage.java), which recognizes objects in an image.

Building from source code

TensorFlow is open-source. You may build TensorFlow for Java from the TensorFlow source code by following the instructions in a separate document (https://github.com/tensorflow/tensorflow/blob/master/tensorflow/java/README.md).

Last updated April 9, 2018.