

Keras

tensorflow

- yay bazelisk

```
pip install -U --user pip six numpy wheel setuptools mock  
pip install -U --user keras_applications --no-deps  
pip install -U --user keras_preprocessing --no-deps
```

```
git clone https://github.com/tensorflow/tensorflow.git  
cd tensorflow
```

```
wget  
https://git.archlinux.org/svntogit/community.git/plain/trunk/Add-grpc-fix-for-gettid.patch?h=packages/tensorflow -O Add-grpc-fix-for-gettid.patch  
patch -Np1 -i Add-grpc-fix-for-gettid.patch
```

edit configure.py and modify _TF_MAX_BAZEL_VERSION = '1.2.1'

```
export TF_NEED_CUDA=1  
.configure
```

this produce .tf_configure.bazelrc

```
build --action_env PYTHON_BIN_PATH="/lab/gianoobserver/lib/bin/python"  
build --action_env  
PYTHON_LIB_PATH="/lab/gianoobserver/lib/lib/python3.8/site-packages"  
build --python_path="/lab/gianoobserver/lib/bin/python"  
build:xla --define with_xla_support=true  
build --config=xla  
build --action_env CUDA_TOOLKIT_PATH="/usr/local/cuda"  
build --action_env TF_CUDA_COMPUTE_CAPABILITIES="6.1"  
build --action_env LD_LIBRARY_PATH="/lab/dnn/lib/lib"  
build --action_env GCC_HOST_COMPILER_PATH="/usr/bin/gcc-8"  
build --config=cuda  
build:opt --copt=-march=native  
build:opt --copt=-Wno-sign-compare  
build:opt --host_copt=-march=native  
build:opt --define with_default_optimizations=true  
test --flaky_test_attempts=3  
test --test_size_filters=small,medium  
test --test_tag_filters=-benchmark-test,-no_oss,-gpu,-oss_serial  
test --build_tag_filters=-benchmark-test,-no_oss,-gpu  
build --action_env TF_CONFIGURE_IOS="0"
```

compile

```
bazel \
  build --config=opt \
    //tensorflow:libtensorflow.so \
    //tensorflow:libtensorflow_cc.so \
    //tensorflow:install_headers \
    //tensorflow/tools/pip_package:build_pip_package
./bazel-bin/tensorflow/tools/pip_package/build_pip_package
/tmp/tensorflow_pkg
pip install /tmp/tensorflow_pkg/tensorflow*.whl
```

from keras to opencv

- https://github.com/adoval4/keras_to_OpenCV_tensorflow

Export protobuf

prereq

```
bazel build tensorflow/python/tools:freeze_graph
```

```
import argparse
ap = argparse.ArgumentParser()
ap.add_argument('--input', required=True)
ap.add_argument('--output', default="/tmp/model/ckpt")
args = ap.parse_args()

from keras import backend as K
from keras.models import load_model
import tensorflow as tf

K.set_learning_phase(0)

# create model and load weights or ...
# model = create_model()
# model.load_weights(args.input)

# ... load h5 model
load_model(args.input)

saver = tf.train.Saver()
saver.save(K.get_session(), args.default)
print("Output node needed in freeze_graph: %s" % model.output.op.name)
print("Model created in: %s" % args.output)
```

result

```
/tmp/model
```

```
└── checkpoint  
    ├── ckpt.data-00000-of-00001  
    ├── ckpt.index  
    └── ckpt.meta
```

freeze

```
bazel-bin/tensorflow/python/tools/freeze_graph \  
--input_meta_graph=/tmp/model/ckpt.meta \  
--input_checkpoint=/tmp/model/ckpt \  
--output_graph=frozen_graph.pb \  
--input_binary \  
--output_node_names="previous output node name"
```

From:
<https://wiki.csgalileo.org/> - **Galileo Labs**



Permanent link:
<https://wiki.csgalileo.org/tips/keras?rev=1575970713>

Last update: **2019/12/10 10:38**