

# 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 `TFMAXBAZELVERSION = '1.2.1'`

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

for atf 1.15

```
echo 0.26.1 > .bazelversion
```

```
this produce .tfconfigure.bazelrc <code> build -actionenv
PYTHONBINPATH="/lab/gianoobserver/lib/bin/python" build -actionenv
PYTHONLIBPATH="/lab/gianoobserver/lib/lib/python3.8/site-packages" build
-pythonpath="/lab/gianoobserver/lib/bin/python" build:xla -define withxlasupport=true build
-config=xla build -actionenv CUDATOOLKITPATH="/usr/local/cuda" build -actionenv
TFCUDACOMPUTECAPABILITIES="6.1" build -actionenv LDLIBRARYPATH="/lab/dnn/lib/lib" build
-actionenv GCCHOSTCOMPILERPATH="/usr/bin/gcc-8" build -config=cuda build:opt -copt=-
march=native build:opt -copt=-Wno-sign-compare build:opt -hostcopt=-march=native build:opt
-define withdefaultoptimizations=true test -flakytestattempts=3 test -testsizefilters=small,medium
test -testtagfilters=-benchmark-test,-nooss,-gpu,-ossserial test -buildtagfilters=-benchmark-test,-
nooss,-gpu build -actionenv TFCONFIGURE_IOS="0" </code>
```

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](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=1576006267>

Last update: **2019/12/10 20:31**

