1 Example of working with Tensorflow Keras Models

While none made any huge breakthroughs on the Moments in Time state-of-the-art, due to the computational costs of training these models on such a large dataset, we would like to make them available to others studying similar problems where transfer learning might be applicable.

1.1 Model Overview

Three types of "off-the'shelf" models are included in this "Model Zoo": 2D CNNs (C2D (https://en.wikipedia.org/wiki/Convolutional_neural_network#Image_recognition)), 3D CNNs (C3D (https://arxiv.org/pdf/1412.0767.pdf), one-stream I3D (https://arxiv.org/abs/1705.07750)), and an LRCN (https://arxiv.org/pdf/1411.4389.pdf) (CNN+LSTM).

C2D models were trained by uniformly randomly sampling frames from the input video. C3D, I3D, and LRCN models were trained by using 16 dense frames randomly sampled from the input video.

1.1.1 Naming Convention

Each of the models is named in the following way: (backbone_name)-(input_shape)-(output_classes)-(training_history).h5

1.1.2 Descriptions

The original backbones from which these "off-the-shelf" models were created and trained are linked to in the table below. The following is a brief description of the models:

Model	Name	Input Shape	# Classes	Training History
C3D-16x224x224x3-339-m.h5	C3D (https://github.com/axon-research/c3d-keras) (source license (https://github.com/axon-research/c3d-keras/blob/master/LICENSE.md))	(16,224,224,3)	339	Moments in Time
D169-224x224x3-339-im.h5	DenseNet169 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/DenseNet169) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time

Model	Name	Input Shape	# Classes	Training History
D201-224x224x3-339-im.h5	DenseNet201 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/DenseNet201) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time
I3DIv1-16x224x224x3-339-ikm.h5	Inflated Inception-v1 3D ConvNet (https://github.com/deepmind/kinetics-i3d) (source license (https://github.com/deepmind/kinetics-i3d/blob/master/LICENSE))	(16,224,224,3)	339	ImageNet and Kinetics (pretrained weights sets) Moments in Time
I3DIv1-32x224x224x3-339-ikm.h5	Inflated Inception-v1 3D ConvNet (https://github.com/deepmind/kinetics-i3d) (source license (https://github.com/deepmind/kinetics-i3d/blob/master/LICENSE))	(32,224,224,3)	339	ImageNet and Kinetics (pretrained weights sets) Moments in Time
IRv2-224x224x3-339-ikm.h5	Inception-ResNet-v2 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/InceptionResNetV2) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time
IRv2avg-64x224x224x3-339-ikm.h5	Inception-ResNet-v2 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/InceptionResNetV2) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(64,224,224,3)	339	ImageNet (pretrained weights set) Moments in Time
Iv3-224x224x3-339-im.h5	Inception-v3 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/InceptionV3) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time
LRCN-16x224x224x3-339-m6h5	Long-term Recurrent Convolutional Network (https://github.com/harvitronix/five-video-classification-methods/blob/master/models.py) (source license (https://github.com/harvitronix/five-video-classification-methods/blob/master/LICENSE))	(16,224,224,3)	339	Moments in Time
M-224x224x3-339-im.h5	MobileNet (https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNet) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time
Mv2-224x224x3-339-im.h5	MobileNet-v2 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV2) (source license (https://www.apache.org/licenses/LICENSE-2.0))	(224,224,3)	339	ImageNet (pretrained weights set) Moments in Time

Model	Name	Input Shape	# Classes	Training History
R50-224x224x3-339-im.h5	ResNet50 (https://www.tensorflow.org/api_docs/python/tf/keras/applications/ResNet50) (source license (https://www.apache.org/licenses	(224,224,3)	339	ImageNet (pretrained weights set) Moments in

1.2 Working With These Models

1.2.1 Loading

Note that it may be neccessay to disable HDF5 file locking as done below

In [ ]:
```
# Disable file locking
import os
os.putenv("HDF5_USE_FILE_LOCKING", "FALSE")
os.system("export $HDF5_USE_FILE_LOCKING")

# Load the model
from tensorflow.python.keras.models import load_model
model_file = ### TODO: add file path here ###
model = load_model(model_file)

# View loaded model
```

1.2.2 Replacing an output layer

Once a model has been loaded, it is possible to modify it by accessing its layers feature. Below is an example of replacing the last layer of model (i.e. the dense classifier) with a new classification layer. This would be necessary when transfering learning from one dataset to another which have different numbers of classes.

In [ ]:
```
# Remove the top layer and replace with a new output layer
from tensorflow.python.keras.models import Model
from tensorflow.python.keras.layers import Dense
x = model.layers[-2].output
new_output_classes = ### TODO: add number of classes in new dataset he
x = Dense(new_output_classes, activation="softmax", name='dense_classi
new_model = Model(inputs=old_model.layers[0].input, outputs=x)

# View modified model
```

1.2.3 Averaging over frames

Often, it is useful to average the prediction across multiple frames if using a C2D base. Here is an example of making this ensemble type model using the model above as a base.

In [ ]:
```
base_model = model
frames = 64
```

```
vid_in = Input(shape=(frames,224,224,3), name='video_input')
x = TimeDistributed(base_model)(vid_in)
x = GlobalAveragePooling1D()(x)
new_avg_model = Model(inputs=[vid_in],outputs=[x])
```

1.2.4 Saving

Once you have modified or trained a model, to save it use:

In [ ]:
```
output_path = ### TODO: add output model path here ###
```

1.3 Questions

Any questions can be directed to Matthew Hutchinson at hutchinson@alum.mit.edu (mailto:hutchinson@alum.mit.edu).

Python license: https://docs.python.org/3/license.html (https://docs.python.org/3/license.html)

TensorFlow license: https://github.com/tensorflow/tensorflow/blob/master/LICENSE (https://github.com/tensorflow/tensorflow/blob/master/LICENSE)

In [ ]:
```
```