AIDL: PROJECTS

•••

Session 8 (2019/06/18): Full model implementation III

Previously on *AIDL: Projects*...

Mapping function

```
dataset = tf.data.Dataset...
    .map(parse_fn, num_parallel_calls=10)
    ...

def parse_fn(image_path, label):
    raw_image = tf.read_file(image_path)
    image = tf.image.decode_jpeg(raw_image, channels=3)
    return image, label
```

Layers

Different ways to create common layers:

- tf.nn: Low level API. You need to create and provide the variables yourself
- tf.layers: High level TF API. It creates everything for you!
- tf.keras.layers: Good-old Keras style layers

Useful links:

https://www.tensorflow.org/api_docs/python/tf/nn

https://www.tensorflow.org/api_docs/python/tf/layers

https://www.tensorflow.org/api_docs/python/tf/keras/layers

Example: Conv2D

Low level API

```
x = …
kernel = tf.get_variable(‘kernel’, shape=[11, 11, 3, 96], dtype=tf.float32)
biases = tf.get_variable(‘biases’, shape=[96], dtype=tf.float32)
output = tf.nn.conv2d(x, kernel, strides=[1, 4, 4, 1], padding="SAME",
name="conv1")
output += biases
output = tf.nn.relu(output)
```

High level API

```
x = …
output = tf.layers.conv2d(x, filters=96, kernel_size=(11, 11), strides=(4,
4), padding=”SAME”, activation=tf.nn.relu, name=”conv1”)
```

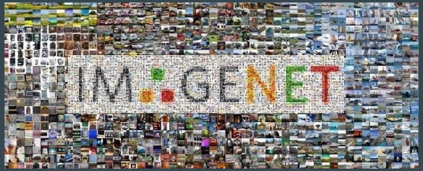
Keras

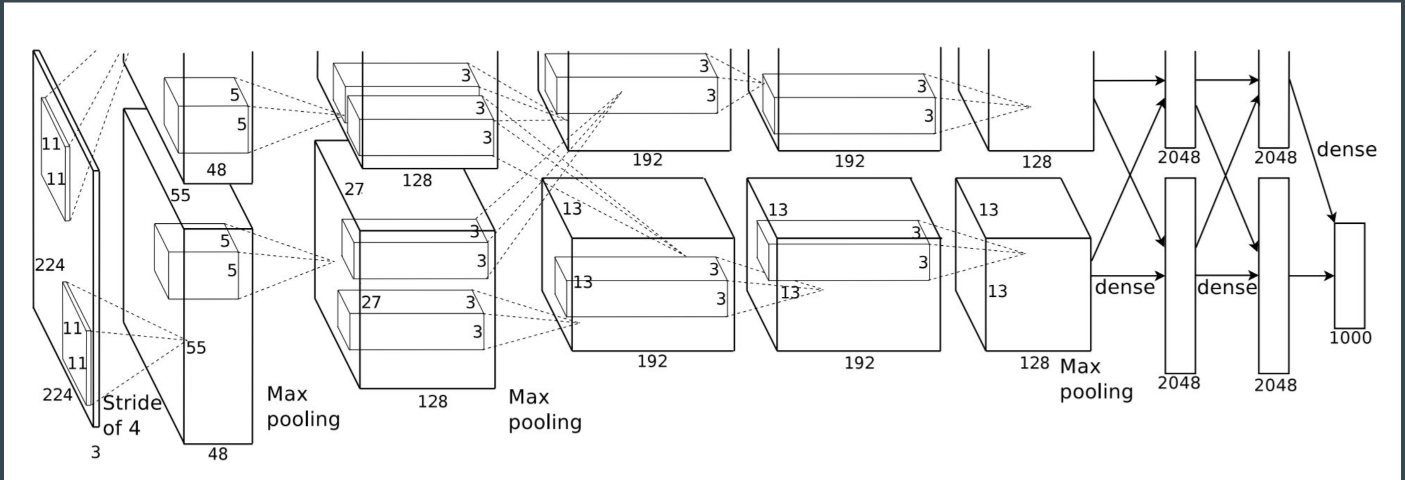
```
x = …
output = tf.keras.layers.Conv2d(filters=96, kernel_size=(11, 11), strides=(4,
4), padding=”SAME”, activation=’relu’, name=”conv1”)(x)
```

Model implementation

AlexNet

- Winner of ILSVRC 2012
- Beginning of CNN revolution





“All of our experiments suggest that our results can be improved simply by waiting for faster GPUs and bigger datasets to become available”

AlexNet implementation

Slight modification over original architecture

DO

- Activation function: ReLU
- Local Response Normalization (check tf.nn)
- Overlapping max-pooling
- L2 weight decay, a.k.a kernel regularization
- Random normal initialization of kernel weights

DONT

- Multi-GPU implementation → use single layer per stage with merged number of filters
- Momentum optimizer → use ADAM as a sane default
- Use 1000 output units → use the number of units appropriate for the task

OPTIONAL

- Dropout
- Bias initialization
- Match dimensions with padding

Exercise VI

Implement AlexNet

- Implement proposed AlexNet
- Connect with input pipeline and train

Cheatsheet

Management

- tf.Session:
  - run
- tf.Graph:
  - as_default_graph
- tf.variable_scope
- tf.name_scope
- tf.summary.FileWriter

Input pipeline

- tf.data.Dataset:
  - from_generator
  - from_tensor_slices
  - shuffle
  - repeat
  - map
  - batch
  - prefetch
  - make_one_shot_iterator
- tf.data.Iterator
  - get_next
- tf.image
  - tf.image.decode_jpeg
- tf.read_file

Models

- tf.layers
  - tf.layers.conv2d
- tf.nn
- tf.keras.layers
- tf.train:
  - tf.train.AdamOptimizer

Checkpointing

- Stateless train loop: **nothing is persisted**
- Solution: model checkpointing and restoration

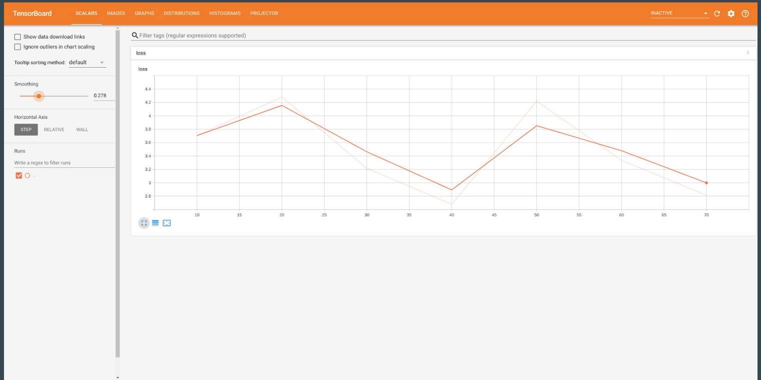
```
saver = tf.train.Saver(var_list=tf.trainable_variables(), max_to_keep=5)
model_checkpoint_path = ‘/tmp/aidl/my_model_checkpoint’

with tf.Session() as sess:
    # Run train op, loss, etc.
    saver.save(sess, save_path=model_checkpoint_path, global_step=global_step)
```

Links:

https://www.tensorflow.org/guide/saved_model#save_and_restore_variables
https://www.tensorflow.org/api_docs/python/tf/train/Saver

Summaries & TensorBoard



□ Show data download links

□ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.278

Horizontal Axis: STEP RELATIVE WALL

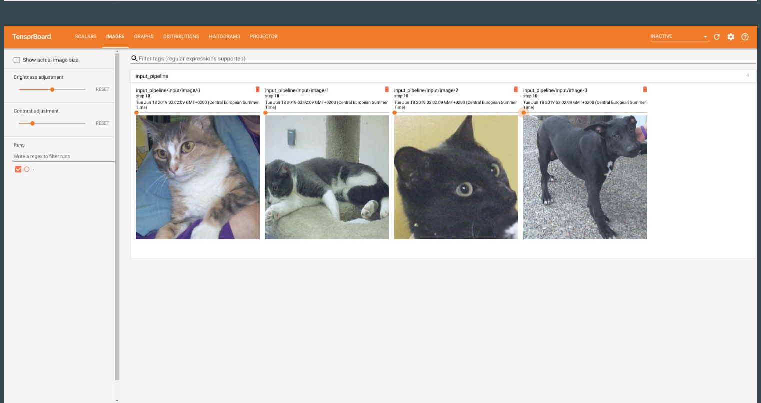
Runs

Write a regex to filter runs

☑ ○ .

Filter tags (regular expressions supported)

loss



□ Show actual image size

Brightness adjustment

Contrast adjustment

Runs

Write a regex to filter runs

☑ ○ .

Filter tags (regular expressions supported)

input_pipeline

input_pipeline/input/image/0
step 10
Tue Jun 18 2019 03:02:09 GMT+0200 (Central European Summer Time)

input_pipeline/input/image/1
step 10
Tue Jun 18 2019 03:02:09 GMT+0200 (Central European Summer Time)

input_pipeline/input/image/2
step 10
Tue Jun 18 2019 03:02:09 GMT+0200 (Central European Summer Time)

input_pipeline/input/image/3
step 10
Tue Jun 18 2019 03:02:09 GMT+0200 (Central European Summer Time)

tf.summary

- tf.summary.scalar
  Scalar values such as loss or metrics.

- tf.summary.image
  Image tensors in grayscale, RGB or RGBA mode.

- tf.summary.histogram
  Histograms of weights, activations or gradients.

- tf.summary.audio
  Audio tensors with a specific number of channels and sample rate

Summaries & TensorBoard

```
# Define summaries along the graph definition
tf.summary.image(‘images’, images, max_outputs=batch_size)

# Create summary writer and merge all summaries into a single op
writer = tf.summary.FileWriter(logdir, graph=tf.get_default_graph())
summary_op = tf.summary.merge_all()

with tf.Session() as sess:
    # Run summary op and global step
    summary, step = sess.run([summary_op, global_step])
    # Add summary to the summary events
    writer.add_summary(summary, global_step=step)
```

Links:

https://www.tensorflow.org/guide/summaries_and_tensorboard
https://www.tensorflow.org/api_docs/python/tf/summary

Questions?