These notes have been drafted to give you some pointers on how to debug your scripts when you start using TensorFlow. Sometimes the errors printed out are helpful, and the script will terminate at the line where the problem occurs. Sometimes it won't. You can save yourself a lot of time in debugging if you make some simple choices to follow good practice when writing code.

Common errors:

This list of common errors may be extended during the course, based on input from the cohort.

This is a common error that will occur when you try and use a tensor that has not been initialised. Learn to read the message from back to front. If you go to the end of the error you will be able to see the variable name that is being used, and the error message is sufficient to tell you what the issue is. The relevant text is highlighted in **red** for clarity. Unfortunatley the line number where the error occurred is not displayed. This is why it is good practice to **always name your tensor objects**.

```
Traceback (most recent call last):
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1334, in _do_call
    return fn(*args)
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1319, in _run_fn
    options, feed_dict, fetch_list, target_list, run_metadata)
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1407, in
_call_tf_sessionrun
    run_metadata)
tensorflow.python.framework.errors_impl.FailedPreconditionError: Attempting
to use uninitialized value x
      [[{{node _retval_x_0_0}}]]

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 929, in run
    run_metadata_ptr)
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1152, in _run
    feed_dict_tensor, options, run_metadata)
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1328, in _do_run
    run_metadata)
  File "/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/client/session.py", line 1348, in _do_call
    raise type(e)(node_def, op, message)
tensorflow.python.framework.errors_impl.FailedPreconditionError: Attempting
to use uninitialized value x
      [[{{node _retval_x_0_0}}]]
```

You can fix this error by initialising the tensors. See the introductory tensor flow slide deck for more information on the various ways to do this.

```
ValueError: Cannot evaluate tensor using `eval()`: No default session is
registered. Use `with sess.as_default()` or pass an explicit session to
`eval(session=sess)`
```

For this error the function `eval()` has been called for a tensor object without having specified a default session to use for the evaluation. Either revert to `sess.run(x)` to evaluate the tensor, or change the code to include

```
with sess.as_default():
    x.eval()
```

to fix this error. Alternatively change the code to use the `sess.run(tensor)` approach to evaluating tensors.

The following type of error occurs when there is a mis-match in shape. The first line highlighted is where the mismatch occurs (e.g. in model building). The last line tells you what the issue is. This case has a weight set of shape [1, 100] multiplied into the data input of shape [?,1] (which is fine) and the problem is that the bias parameter has been initialised with the wrong shape – there are only 50 bias offsets, where 100 are needed for the syntax to be correct. Everything between the highlighted red lines just refers to tensorflow code that is expecting the user to ensure the shapes match.

```
Traceback (most recent call last):
  File "./Example_ShapeError.py", line 90, in <module>
    layer_1     = tf.nn.sigmoid(tf.add(tf.matmul(x_,w_layer_1),bias_layer_1))
  File "/Library/Python/2.7/site-packages/tensorflow/python/ops/gen_math_ops.py", line 73, in add
    result = _op_def_lib.apply_op("Add", x=x, y=y, name=name)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/op_def_library.py", line 759, in
apply_op
    op_def=op_def)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/ops.py", line 2242, in create_op
    set_shapes_for_outputs(ret)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/ops.py", line 1617, in
set_shapes_for_outputs
    shapes = shape_func(op)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/ops.py", line 1568, in
call_with_requiring
    return call_cpp_shape_fn(op, require_shape_fn=True)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/common_shapes.py", line 610, in
call_cpp_shape_fn
    debug_python_shape_fn, require_shape_fn)
  File "/Library/Python/2.7/site-packages/tensorflow/python/framework/common_shapes.py", line 675, in
_call_cpp_shape_fn_impl
    raise ValueError(err.message)

ValueError: Dimensions must be equal, but are 100 and 50 for 'Add' (op: 'Add') with input shapes: [?,100], [50].
```

Spyder gives no output on running a script.

Try re-running the script, or restarting Spyder. Make sure you save any changes to your work before restarting Spyder to ensure no loss of work.

Spyder stops running a TensorFlow model training scripts without any warning or error. The problem arises in the loop over training epochs.

TensorFlow is set up to load the OpenMP library multiple times, which will cause the script to crash, unless the environment variable `KMP_DUPLICATE_LIB_OK` is set to true. The

example script, and the tensor flow example scripts found in the scripts directory have been setup to include the following lines:

```
# this is required to permit multiple copies of the OpenMP runtime to be linked
# to the programme.  Failure to include the following two lines will result in
# an error that Spyder will not report.  On PyCharm the error provided will be
#    OMP: Error #15: Initializing libiomp5.dylib, but found libiomp5.dylib already
initialized.
#    ...
import os

os.environ['KMP_DUPLICATE_LIB_OK']='True'
```

to overcome the limitation. If you run the same code in a python terminal there is no issue, and if you run in PyCharm you get an error that will allow you to debug this problem. The simplest solution is to include the above lines of code at the start of your script.

The following is an example of using a tensor of the wrong type, int32, rather than a float64 or float32 to perform a calculation. In this case the offending line is the one computing the inverse of a matrix of integer elements.

```
Traceback (most recent call last):
  File "./ops_matrix.py", line 47, in <module>
    print("A^-1    = ", sess.run(tf.matrix_inverse(A)))
  File "/Library/Python/2.7/site-packages/tensorflow/python/ops/gen_linalg_ops.py", line
330, in matrix_inverse
    name=name)
  File "/Library/Python/2.7/site-
packages/tensorflow/python/framework/op_def_library.py", line 582, in apply_op
    _Attr(op_def, input_arg.type_attr))
  File "/Library/Python/2.7/site-
packages/tensorflow/python/framework/op_def_library.py", line 60, in
_SatisfiesTypeConstraint
    ", ".join(dtypes.as_dtype(x).name for x in allowed_list)))

TypeError: DataType int32 for attr 'T' not in list of allowed values: float64, float32
```

Warnings to ignore:

Generally it is not good to ignore warnings as these often mean something is not working properly. In this case we are ignoring warnings that the TensorFlow API is changing, and when upgrading to a later API the warnings should go away. In practice we don't need to change the code to continue using it and there is no problem. So we can safely ignore these warnings.

```
WARNING:tensorflow:From
/Users/bevan/anaconda/envs/tensorflow_env/lib/python3.7/site-
packages/tensorflow/python/framework/op_def_library.py:263:
colocate_with (from tensorflow.python.framework.ops) is deprecated
and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

For the following we see a performance related warning. The computer being used to run this code has the ability to support a broader range of instructions than supported by the tensorflow binary installed. Again this (and the subsequent warning) can be ignored.

```
2019-07-14 15:03:18.128721: I
tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports
instructions that this TensorFlow binary was not compiled to use:
SSE4.1 SSE4.2 AVX AVX2 FMA
```

```
2019-07-14 15:03:18.129048: I
tensorflow/core/common_runtime/process_util.cc:71] Creating new
thread pool with default inter op setting: 4. Tune using
inter_op_parallelism_threads for best performance.
```

Suggested good practice:

The following notes on suggested good practice are intended to help you make more user friendly and maintainable code. While a strong coder will not have any trouble reading someone else's work (or their own sometime after writing it), for the rest of us, who multiple parts of their job and come back to complicated code only rarely once it is working, these suggestions can help.

Provide sufficient up to date and useful comments on your code. The Example.py script is provided for this course in order to encourage a basic level of documentation – that of providing a header description to a given file. That helps the reader of the code understand what to expect in that file.

When using TensorFlow objects, give them a unique name. This means if the code crashes because some object has the wrong Shape, or for some other reason, you can identify which object of that type has a problem (rather than trying to figure out which variable, placeholder or constant has a shape mis-match from the many that you use in your code.

Ensure that you save your work regularly and maintain backups. This mitigates the chance of losing significant work should a computer fail for some reason, or you logout without saving work done. A simple and secure way to backup work is to send yourself completed scripts as attachments to e-mails. Alternatively save code to a USB stick to ensure that you have a backup other than on the student computer system being used. This is especially important given the compressed teaching schedule for this course, where there is little time to rewrite code assignments to recover from a loss of files.
