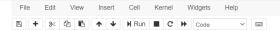
coursera



Jupyter C2_W3_Lab_1_autograph-basics Last Checkpoint: 11/17/2020 (autosaved)



Trusted Python 3 O



Autograph: Basic

In this ungraded lab, you will go through some of the basics of autograph so you can explore what the generated code looks like.

Imports

Addition in autograph

You can use the @tf.function decorator to automatically generate the graph-style code as shown below:

if-statements in autograph

Control flow statements which are very intuitive to write in eager mode can look very complex in graph mode. You can see that in the next examples: first a simple function, then a more complicated one that involves lots of ops and conditionals (fizzbuzz).

```
In [3]: \mbox{\it M} # simple function that returns the square if the input is greater than zero
                  atf.function
                def f(x):
    if x>0:
                      return x
                print(tf.autograph.to_code(f.python_function))
                 def tf__f(x):
                der tT__T(X):
    with ag _.FunctionScope('f', 'fscope', ag__.ConversionOptions(recursive=True, user_requested=True, optional_features=(),
internal_convert_user_code=True)) as fscope:
    do_return = False
        retval_ = ag__.UndefinedReturnValue()
                            def get_state():
    return (x,)
                            def set_state(vars_):
                                  nonlocal x
(x,) = vars_
                            def if body():
                                  nonlocal x
x = (ag_.ld(x) * ag_.ld(x))
                            def else_body():
    nonlocal x
                                  pass
                                 .if_stmt((ag_..ld(x) > 0), if_body, else_body, get_state, set_state, ('x',), 1)
                            ag__..._
try:
    do_return = True
    retval_ = ag__.ld(x)
                            except:
do_return = False
                            return fscope.ret(retval_, do_return)
```

Fizzbuzz in autograph

You may remember implementing $\underline{\text{fizzbuzz}}$ in preparation for a coding interview.

• Imagine how much fun it would be if you were asked to impement the graph mode version of that code!

Fortunately, you can just use $\ensuremath{\operatorname{\textsc{@tf.function}}}$ and then call $\ensuremath{\operatorname{\textsc{tf.autograph.to_code}}}$!

```
print('Fizz')
elif num % 5 == 0:
    print('Buzz')
else:
     print(num)
counter += 1
return counter
 \verb|print(tf.autograph.to_code(fizzbuzz.python_function))|\\
def tf_fizzbuzz(max_num):
    with ag_.FunctionScope('fizzbuzz', 'fscope', ag__.ConversionOptions(recursive=True, user_requested=True, optional_featu
res=(), internal_convert_user_code=True)) as fscope:
    do_return = False
    retval_ = ag__.UndefinedReturnValue()
    counter = 0
           def get_state_3():
                return (counter,)
           def set_state_3(vars_):
                nonlocal counter
(counter,) = vars_
           def loop_body(itr):
                nonlocal counter
num = itr
                def get_state_2():
    return ()
                def set_state_2(block_vars):
                def if_body_2():
    ag__.ld(print)('FizzBuzz')
                def else_body_2():
                     def get_state_1():
    return ()
                     def set_state_1(block_vars):
    pass
                     def if_body_1():
    ag__.ld(print)('Fizz')
                      def else_body_1():
                          def get_state():
    return ()
                          def set_state(block_vars):
    pass
                          def if_body():
    ag__.ld(print)('Buzz')
raise
           return fscope.ret(retval_, do_return)
```