Add New Op in TensorFlow

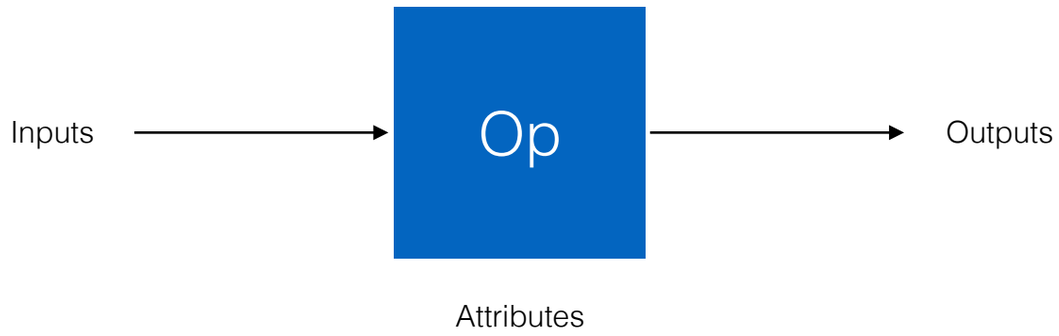
20 March 2017

Steps

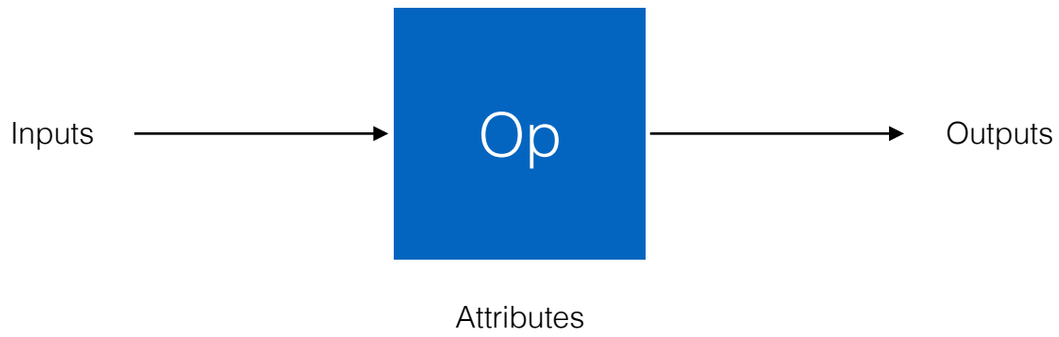
1. Create Interface
2. Implement Interface
3. Compilation
4. Implement Gradient (Python)
5. Call New Op (Python)

Create Interface

Create Interface



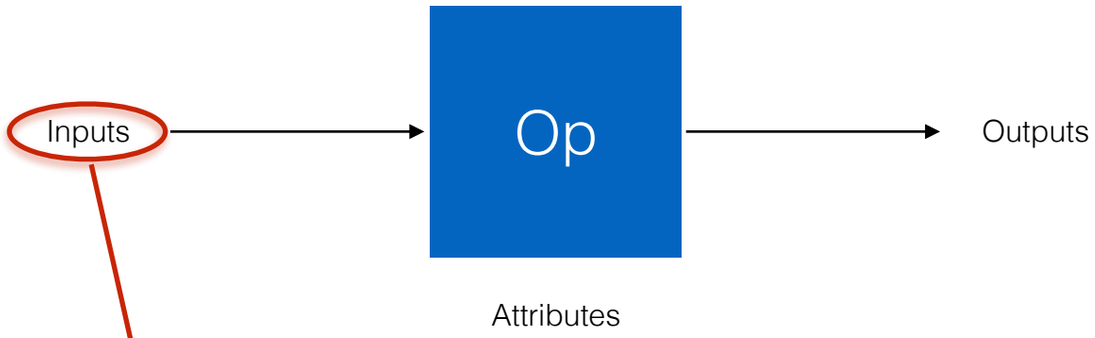
Create Interface



Example (copy input to output)

```
REGISTER_OP("CopyMat")
    .Input("in: int16")
    .Output("out: int16")
    .SetShapeFn([](::tensorflow::shape_inference::InferenceContext* c)
    {
                    c->set_output(0, c->input(0));
                    return Status::OK();
    }); // Set the shape property for the output tensor
```

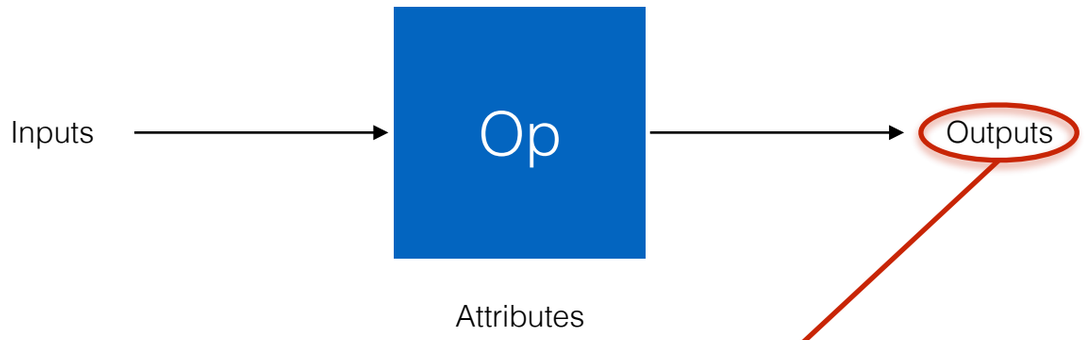
Create Interface



Example (copy input to output)

```
7  REGISTER_OP("CopyMat")
8      .Input("in: int16")
9      .Output("out: int16")
10     .SetShapeFn([](::tensorflow::shape_inference::InferenceContext* c)
11     {
12                     c->set_output(0, c->input(0));
13                     return Status::OK();
14     }); // Set the shape property for the output tensor
15
```

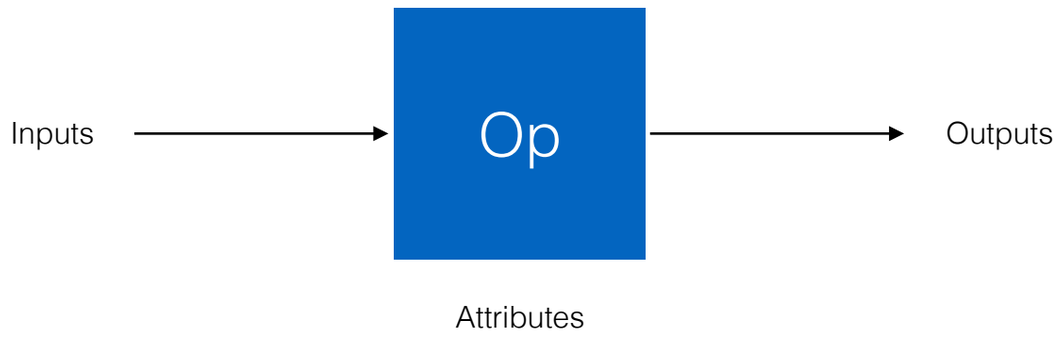
Create Interface



Example (copy input to output)

```
REGISTER_OP("CopyMat")
    .Input("in: int16")
    .Output("out: int16")
    .SetShapeFn([](::tensorflow::shape_inference::InferenceContext* c)
    {
                    c->set_output(0, c->input(0));
                    return Status::OK();
    }); // Set the shape property for the output tensor
```

Create Interface

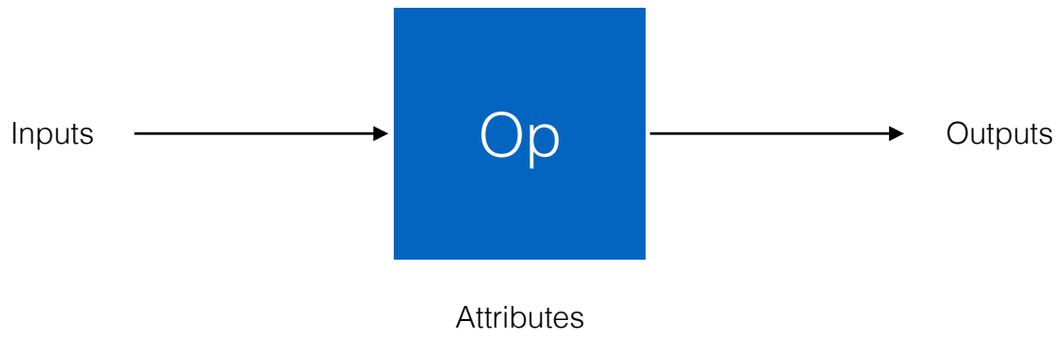


Example (copy input to output)

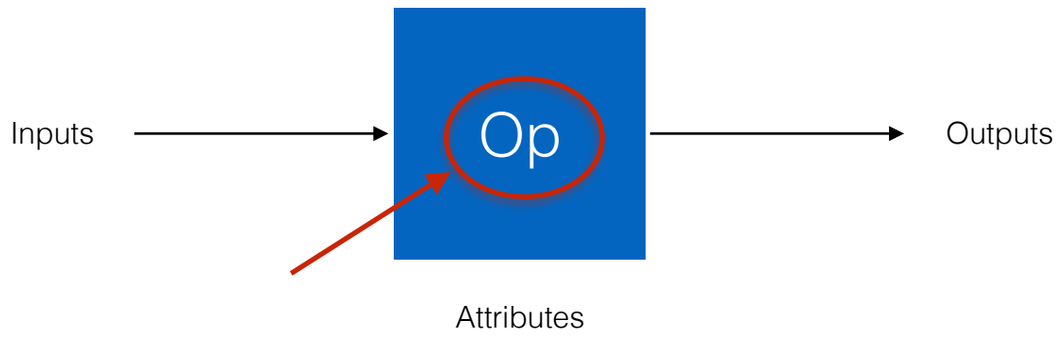
```
7  REGISTER_OP("CopyMat")
8      .Input("in: int16")
9      .Output("out: int16")
10     .SetShapeFn([](::tensorflow::shape_inference::InferenceContext* c)
11     {
12                     c->set_output(0, c->input(0));
13                     return Status::OK();
14     }); // Set the shape property for the output tensor
15
```

Implement Interface

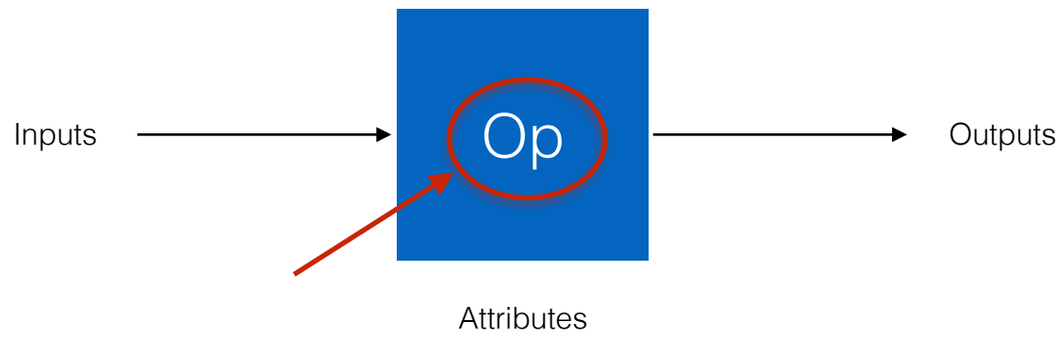
Implement Interface



Implement Interface



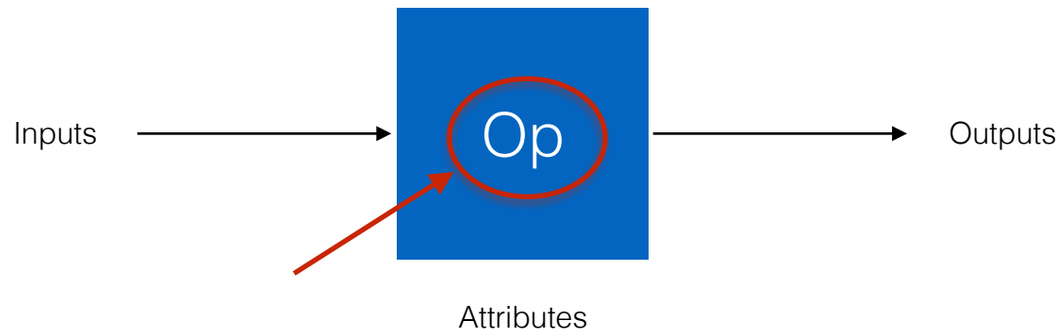
Implement Interface



Example (copy input to output)

```
class CopyMatOp : public OpKernel {
    public:
    explicit CopyMatOp(OpKernelConstruction* context) : OpKernel(context) {}

    void Compute(OpKernelContext* context) override {
        // Grab the input tensor
        const Tensor& input_tensor = context->input(0);
        int num_dims = input_tensor.shape().dims();
        if (num_dims != 3)
            return;
        int depth = input_tensor.shape().dim_size(0);
        int width = input_tensor.shape().dim_size(1);
        int height = input_tensor.shape().dim_size(2);
        auto input = input_tensor.shaped<int16,3>({depth,width,height}); // Conversion to Eigen::Tensor

        // Create an output tensor
        Tensor* output_tensor = NULL;
        OP_REQUIRES_OK(context, context->allocate_output(0, input_tensor.shape(),
                                                         &output_tensor));

        auto output = output_tensor->shaped<int16,3>({depth,width,height}); // Conversion to Eigen::Tensor

        // Copy all elements
        for (int i = 0; i < depth; i++)
        {
            for (int j = 0; j < width; j++)
            {
                for (int k = 0; k < height; k++)
                {
                    output(i,j,k) = input(i,j,k);
                }
            }
        }

    }
};
```

Implement Interface



Example (copy input to output)

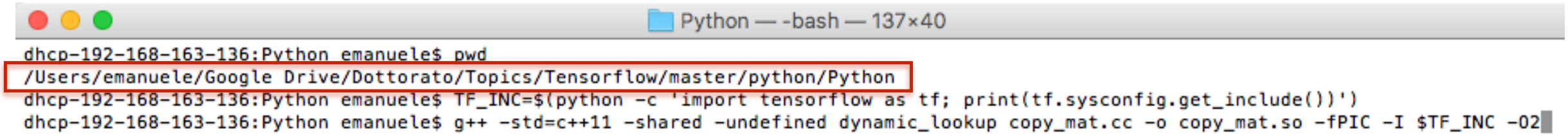
```
16 class CopyMatOp : public OpKernel {
17     public:
18     explicit CopyMatOp(OpKernelConstruction* context) : OpKernel(context) {}
19
20     void Compute(OpKernelContext* context) override {
21         // Grab the input tensor
22         const Tensor& input_tensor = context->input(0);
23         int num_dims = input_tensor.shape().dims();
24         if (num_dims != 3)
25             return;
26         int depth = input_tensor.shape().dim_size(0);
27         int width = input_tensor.shape().dim_size(1);
28         int height = input_tensor.shape().dim_size(2);
29         auto input = input_tensor.shaped<int16,3>({depth,width,height}); // Conversion to Eigen::Tensor
30
31         // Create an output tensor
32         Tensor* output_tensor = NULL;
33         OP_REQUIRES_OK(context, context->allocate_output(0, input_tensor.shape(),
34                                                          &output_tensor));
35
36         auto output = output_tensor->shaped<int16,3>({depth,width,height}); // Conversion to Eigen::Tensor
37
38         // Copy all elements
39         for (int i = 0; i < depth; i++)
40         {
41             for (int j = 0; j < width; j++)
42             {
43                 for (int k = 0; k < height; k++)
44                 {
```

Remember to register the operation using the following command

```
REGISTER_KERNEL_BUILDER(Name("CopyMat").Device(DEVICE_CPU), CopyMatOp);
```

Compilation

Compilation



Change directory containing the C++ implementation of the new operation

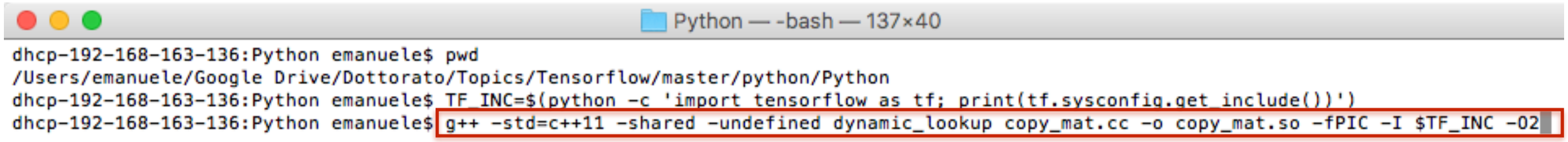
Compilation

Python — -bash — 137×40

```
dhcp-192-168-163-136:Python emanuele$ pwd
/Users/emanuele/Google Drive/Dottorato/Topics/Tensorflow/master/python/Python
dhcp-192-168-163-136:Python emanuele$ TF_INC=$(python -c 'import tensorflow as tf; print(tf.sysconfig.get_include())')
dhcp-192-168-163-136:Python emanuele$ g++ -std=c++11 -shared -undefined dynamic_lookup copy_mat.cc -o copy_mat.so -fPIC -I $TF_INC -O2
```

Save some environment variable

Compilation



Compile

Compilation

Python — -bash — 137×40

```
dhcp-192-168-163-136:Python emanuele$ pwd
/Users/emanuele/Google Drive/Dottorato/Topics/Tensorflow/master/python/Python
dhcp-192-168-163-136:Python emanuele$ TF_INC=$(python -c 'import tensorflow as tf; print(tf.sysconfig.get_include())')
dhcp-192-168-163-136:Python emanuele$ g++ -std=c++11 -shared -undefined dynamic_lookup copy_mat.cc -o copy_mat.so -fPIC -I $TF_INC -O2
```

This should be used only when using Mac OS!

Compilation

Python — -bash — 137×40

```
dhcp-192-168-163-136:Python emanuele$ pwd
/Users/emanuele/Google Drive/Dottorato/Topics/Tensorflow/master/python/Python
dhcp-192-168-163-136:Python emanuele$ TF_INC=$(python -c 'import tensorflow as tf; print(tf.sysconfig.get_include())')
dhcp-192-168-163-136:Python emanuele$ g++ -std=c++11 -shared -undefined dynamic_lookup copy_mat.cc -o copy_mat.so -fPIC -I $TF_INC -O2
```

Example Output

```
In file included from /Users/emanuele/anaconda/lib/python3.5/site-packages/tensorflow/include/google/protobuf/stubs/atomicops.h:199:
/Users/emanuele/anaconda/lib/python3.5/site-packages/tensorflow/include/google/protobuf/stubs/atomicops_internals_macosx.h:173:9: warning
:
      'OSAtomicCompareAndSwap64Barrier' is deprecated: first deprecated in macOS 10.12 - Use std::atomic_compare_exchange_strong() from
      <atomic> instead [-Wdeprecated-declarations]
    if (OSAtomicCompareAndSwap64Barrier(
        ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.12.sdk/usr/include/libkern/OSAtomicDeprecate
d.h:645:9: note:
      'OSAtomicCompareAndSwap64Barrier' has been explicitly marked deprecated here
bool    OSAtomicCompareAndSwap64Barrier( int64_t __oldValue, int64_t __newValue,
        ^
11 warnings generated.
```

Compilation

Python — -bash — 137×40

```
dhcp-192-168-163-136:Python emanuele$ pwd
/Users/emanuele/Google Drive/Dottorato/Topics/Tensorflow/master/python/Python
dhcp-192-168-163-136:Python emanuele$ TF_INC=$(python -c 'import tensorflow as tf; print(tf.sysconfig.get_include())')
dhcp-192-168-163-136:Python emanuele$ g++ -std=c++11 -shared -undefined dynamic_lookup copy_mat.cc -o copy_mat.so -fPIC -I $TF_INC -O2
```

Example Output

```
In file included from /Users/emanuele/anaconda/lib/python3.5/site-packages/tensorflow/include/google/protobuf/stubs/atomicops.h:199:
/Users/emanuele/anaconda/lib/python3.5/site-packages/tensorflow/include/google/protobuf/stubs/atomicops_internals_macosx.h:173:9: warning
:
      'OSAtomicCompareAndSwap64Barrier' is deprecated: first deprecated in macOS 10.12 - Use std::atomic_compare_exchange_strong() from
      <atomic> instead [-Wdeprecated-declarations]
    if (OSAtomicCompareAndSwap64Barrier(
        ^
/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.12.sdk/usr/include/libkern/OSAtomicDeprecate
d.h:645:9: note:
      'OSAtomicCompareAndSwap64Barrier' has been explicitly marked deprecated here
bool    OSAtomicCompareAndSwap64Barrier( int64_t __oldValue, int64_t __newValue,
        ^
11 warnings generated.
```

FILE copy_mat.so GENERATED!

Implement Gradient

Implement Gradient

Easy task, check:

https://www.tensorflow.org/extend/adding_an_op#implement_the_gradient_in_python

Call New Op (Python)

Call New Op (Python)

Example (copy input to output)

```
copy_module = tf.load_op_library('/path/to/file/copy_mat.so')
.. .. ..
a = copy_module.copy_mat(patient)
```