Constants and variables

INTRODUCTION TO TENSORFLOW IN PYTHON

Isaiah Hull
Economist

datacamp

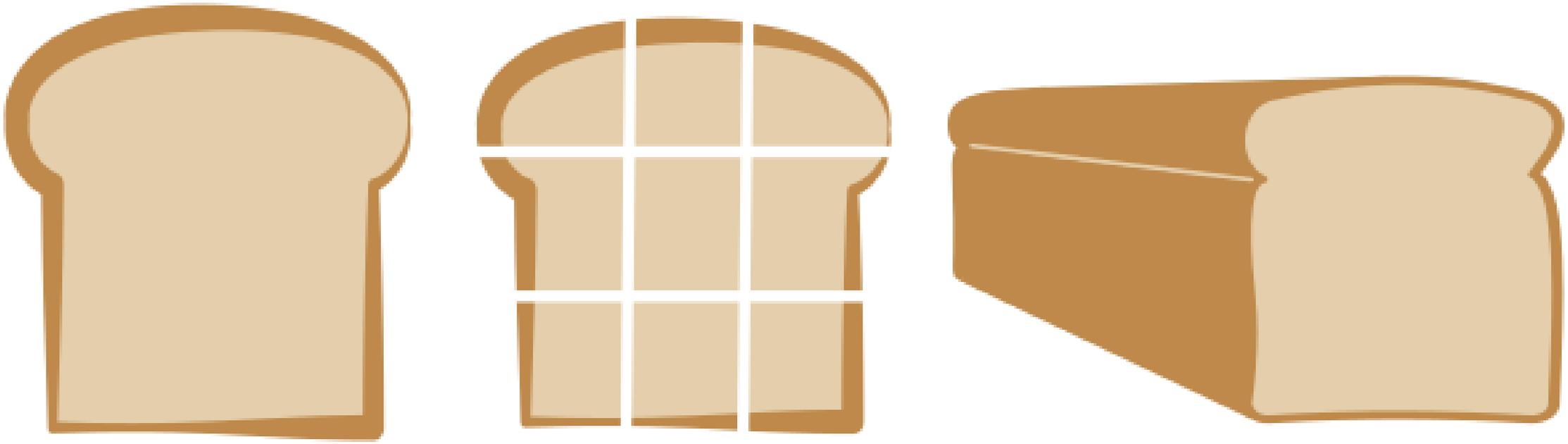
What is TensorFlow?

- **Open-source library for graph-based numerical computation**
  - Developed by the Google Brain Team
- **Low and high level APIs**
  - Addition, multiplication, differentiation
  - Machine learning models
- **Important changes in TensorFlow 2.0**
  - Eager execution by default
  - Model building with Keras and Estimators

What is a tensor?

- Generalization of vectors and matrices
- Collection of numbers
- Specific shape

What is a tensor?



Source: Public Domain Vectors

Defining tensors in TensorFlow

```
import tensorflow as tf

# 0D Tensor
d0 = tf.ones((1,))
```

```
# 1D Tensor
d1 = tf.ones((2,))
```

```
# 2D Tensor
d2 = tf.ones((2, 2))
```

```
# 3D Tensor
d3 = tf.ones((2, 2, 2))
```

Defining tensors in TensorFlow

```
# Print the 3D tensor
print(d3.numpy())
```

```
[[[1. 1.]
  [1. 1.]]

 [[1. 1.]
  [1. 1.]]]
```

Defining constants in TensorFlow

- **A constant is the simplest category of tensor**
  - Not trainable
  - Can have any dimension

```
from tensorflow import constant

# Define a 2x3 constant.
a = constant(3, shape=[2, 3])
```

```
# Define a 2x2 constant.
b = constant([1, 2, 3, 4], shape=[2, 2])
```

Using convenience functions to define constants

Operation	Example
`tf.constant()`	`constant([1, 2, 3])`
`tf.zeros()`	`zeros([2, 2])`
`tf.zeros_like()`	`zeros_like(input_tensor)`
`tf.ones()`	`ones([2, 2])`
`tf.ones_like()`	`ones_like(input_tensor)`
`tf.fill()`	`fill([3, 3], 7)`

Defining and initializing variables

```
import tensorflow as tf

# Define a variable
a0 = tf.Variable([1, 2, 3, 4, 5, 6], dtype=tf.float32)
a1 = tf.Variable([1, 2, 3, 4, 5, 6], dtype=tf.int16)
```

```
# Define a constant
b = tf.constant(2, tf.float32)
```

```
# Compute their product
c0 = tf.multiply(a0, b)
c1 = a0*b
```

Let's practice!

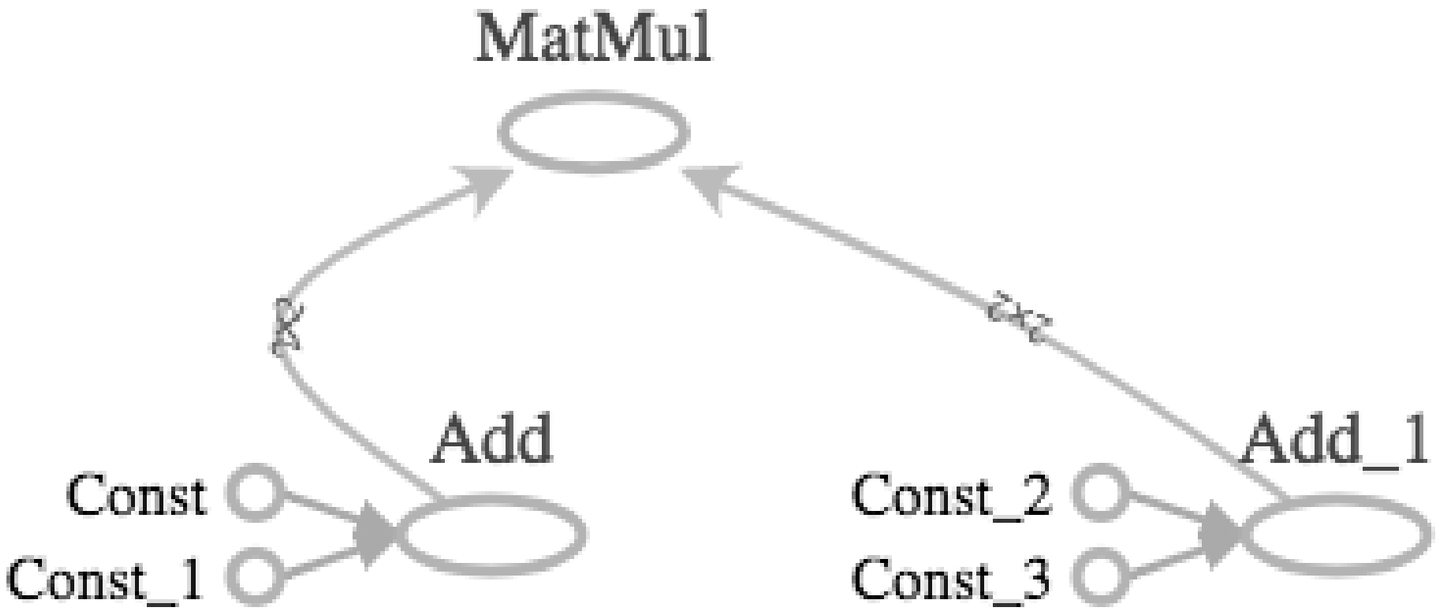
INTRODUCTION TO TENSORFLOW IN PYTHON

Basic operations

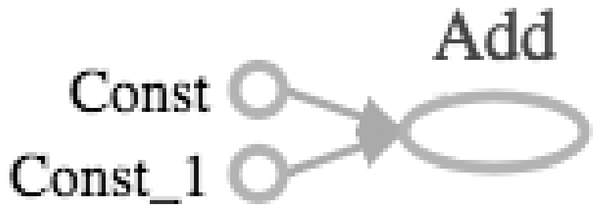
INTRODUCTION TO TENSORFLOW IN PYTHON

Isaiah Hull
Economist

What is a TensorFlow operation?

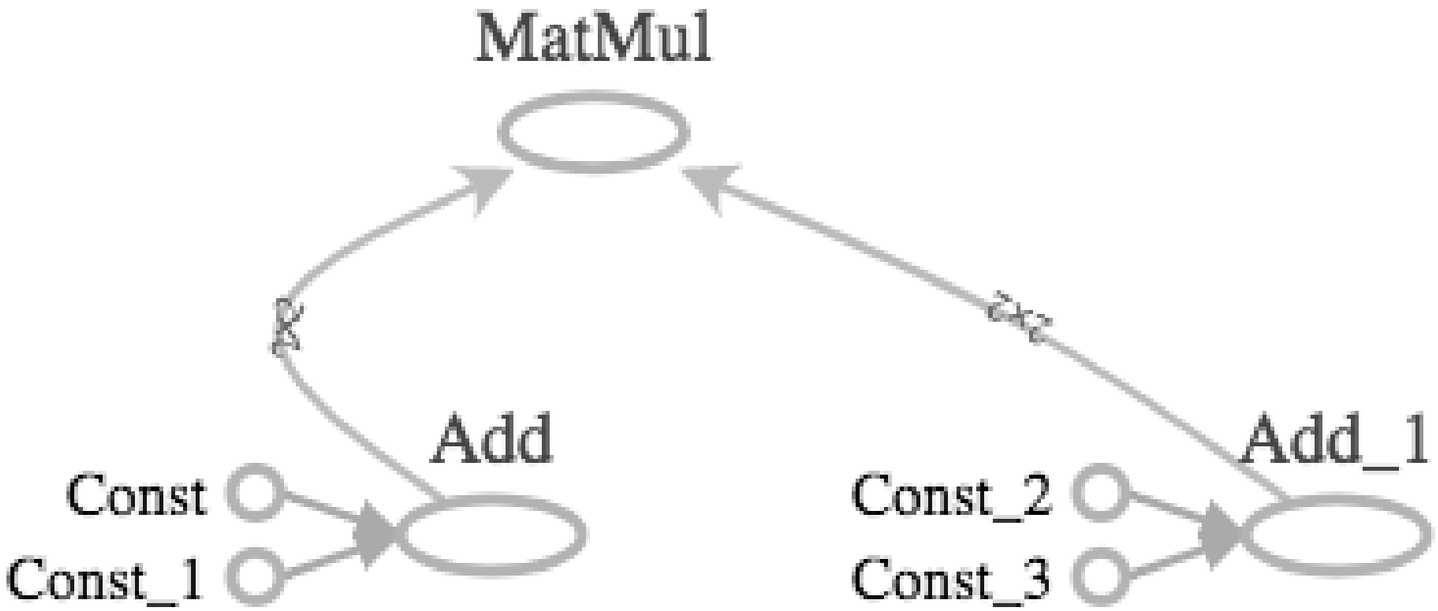


What is a TensorFlow operation?



What is a TensorFlow operation?

What is a TensorFlow operation?



Applying the addition operator

```
#Import constant and add from tensorflow
from tensorflow import constant, add

# Define 0-dimensional tensors
A0 = constant([1])
B0 = constant([2])
```

```
# Define 1-dimensional tensors
A1 = constant([1, 2])
B1 = constant([3, 4])
```

```
# Define 2-dimensional tensors
A2 = constant([[1, 2], [3, 4]])
B2 = constant([[5, 6], [7, 8]])
```

Applying the addition operator

```
# Perform tensor addition with add()
C0 = add(A0, B0)
C1 = add(A1, B1)
C2 = add(A2, B2)
```

Performing tensor addition

- The `add()` operation performs **element-wise addition** with two tensors
- Element-wise addition requires both tensors to have the same shape:
  - Scalar addition: $1 + 2 = 3$
  - Vector addition: $[1, 2] + [3, 4] = [4, 6]$
  - Matrix addition: $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$
- The `add()` operator is overloaded

How to perform multiplication in TensorFlow

- **Element-wise multiplication** performed using `multiply()` operation
  - The tensors multiplied must have the same shape
  - E.g. [1,2,3] and [3,4,5] or [1,2] and [3,4]
- **Matrix multiplication** performed with `matmul()` operator
  - The `matmul(A,B)` operation multiplies A by B
  - Number of columns of A must equal the number of rows of B

Applying the multiplication operators

```
# Import operators from tensorflow
from tensorflow import ones, matmul, multiply

# Define tensors
A0 = ones(1)
A31 = ones([3, 1])
A34 = ones([3, 4])
A43 = ones([4, 3])
```

- What types of operations are valid?
  - `multiply(A0, A0)`, `multiply(A31, A31)`, and `multiply(A34, A34)`
  - `matmul(A43, A34)`, but not `matmul(A43, A43)`

Summing over tensor dimensions

- The `reduce_sum()` operator sums over the dimensions of a tensor
  - `reduce_sum(A)` sums over all dimensions of A
  - `reduce_sum(A, i)` sums over dimension i

```
# Import operations from tensorflow
from tensorflow import ones, reduce_sum

# Define a 2x3x4 tensor of ones
A = ones([2, 3, 4])
```

Summing over tensor dimensions

```
# Sum over all dimensions
B = reduce_sum(A)


# Sum over dimensions 0, 1, and 2
B0 = reduce_sum(A, 0)
B1 = reduce_sum(A, 1)
B2 = reduce_sum(A, 2)
```

Let's practice!

INTRODUCTION TO TENSORFLOW IN PYTHON

Advanced operations

INTRODUCTION TO TENSORFLOW IN PYTHON

Isaiah Hull
Economist

Overview of advanced operations

- We have covered basic operations in TensorFlow
  - `add()`, `multiply()`, `matmul()`, and `reduce_sum()`
- In this lesson, we explore advanced operations
  - `gradient()`, `reshape()`, and `random()`

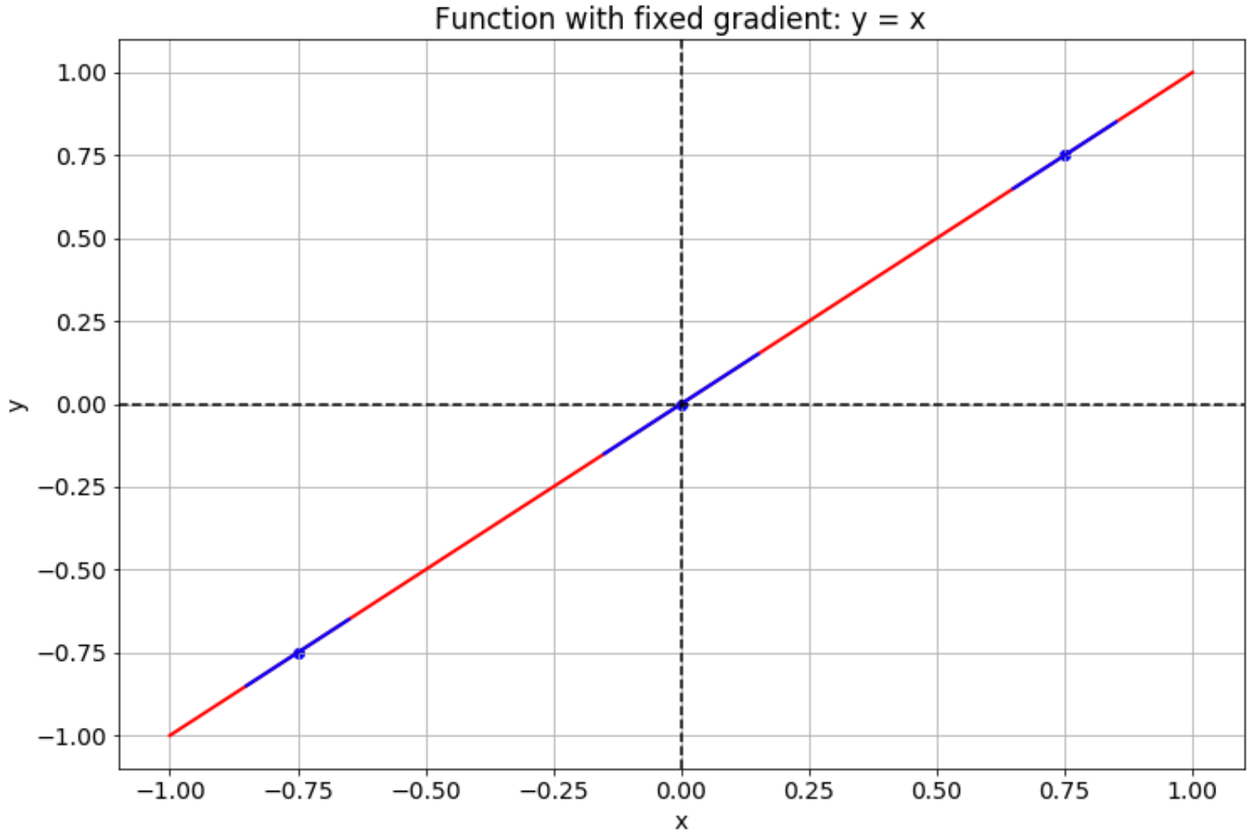
Overview of advanced operations

Operation	Use
`gradient()`	Computes the slope of a function at a point
`reshape()`	Reshapes a tensor (e.g. 10x10 to 100x1)
`random()`	Populates tensor with entries drawn from a probability distribution

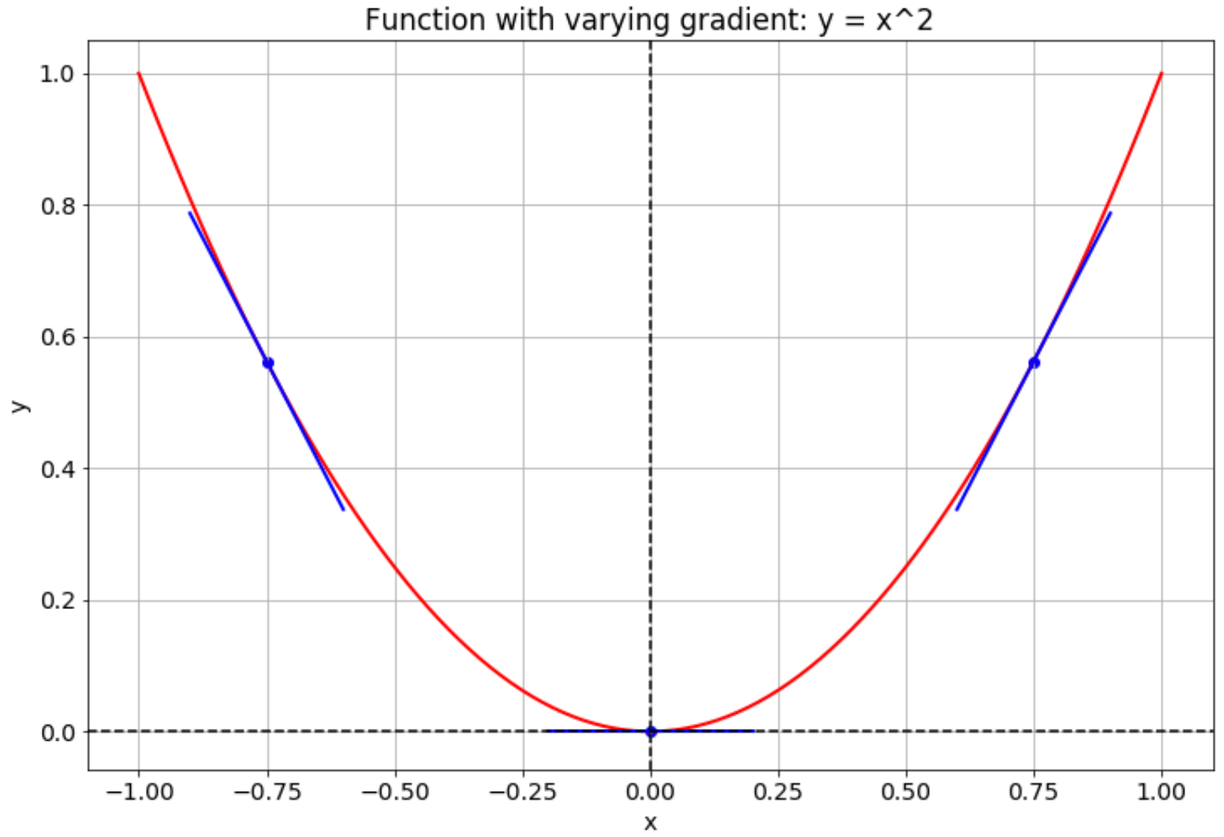
Finding the optimum

- In many problems, we will want to find the optimum of a function.
  - **Minimum**: Lowest value of a loss function.
  - **Maximum**: Highest value of objective function.
- We can do this using the `gradient()` operation.
  - **Optimum**: Find a point where gradient = 0.
  - **Minimum**: Change in gradient > 0
  - **Maximum**: Change in gradient < 0

Calculating the gradient



Calculating the gradient



Gradients in TensorFlow

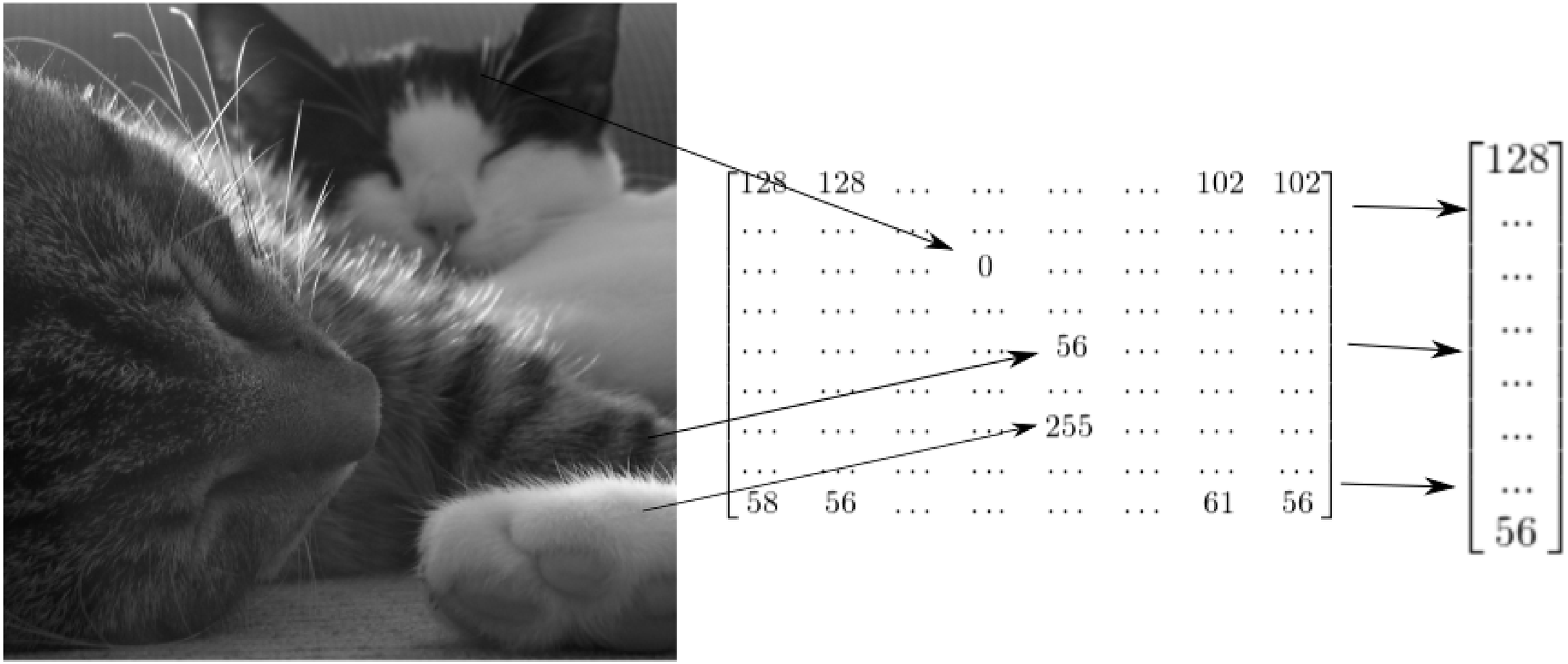
```
# Import tensorflow under the alias tf
import tensorflow as tf

# Define x
x = tf.Variable(-1.0)
```

```
# Define y within instance of GradientTape
with tf.GradientTape() as tape:
    tape.watch(x)
    y = tf.multiply(x, x)
```

```
# Evaluate the gradient of y at x = -1
g = tape.gradient(y, x)
print(g.numpy())
```

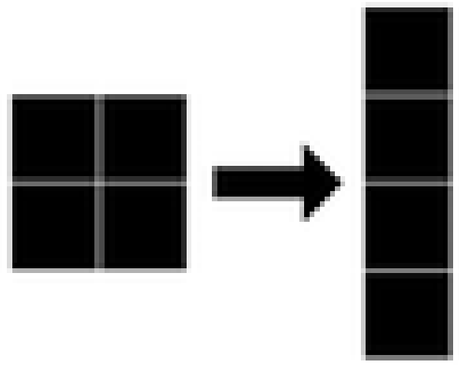
```
-2.0
```

Images as tensors



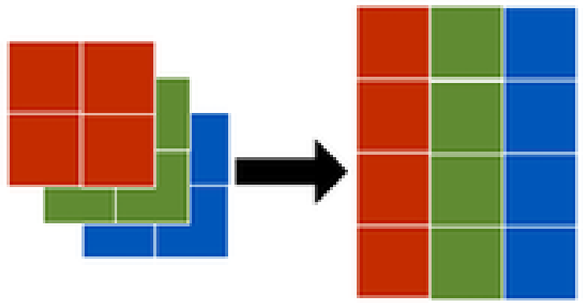
How to reshape a grayscale image

```
# Import tensorflow as alias tf
import tensorflow as tf

# Generate grayscale image
gray = tf.random.uniform([2, 2], maxval=255, dtype='int32')

# Reshape grayscale image
gray = tf.reshape(gray, [2*2, 1])
```



How to reshape a color image

```
# Import tensorflow as alias tf
import tensorflow as tf

# Generate color image
color = tf.random.uniform([2, 2, 3], maxval=255, dtype='int32')

# Reshape color image
color = tf.reshape(color, [2*2, 3])
```



Let's practice!

INTRODUCTION TO TENSORFLOW IN PYTHON