TensorFlow

# Broadcasting

主讲人: 龙良曲

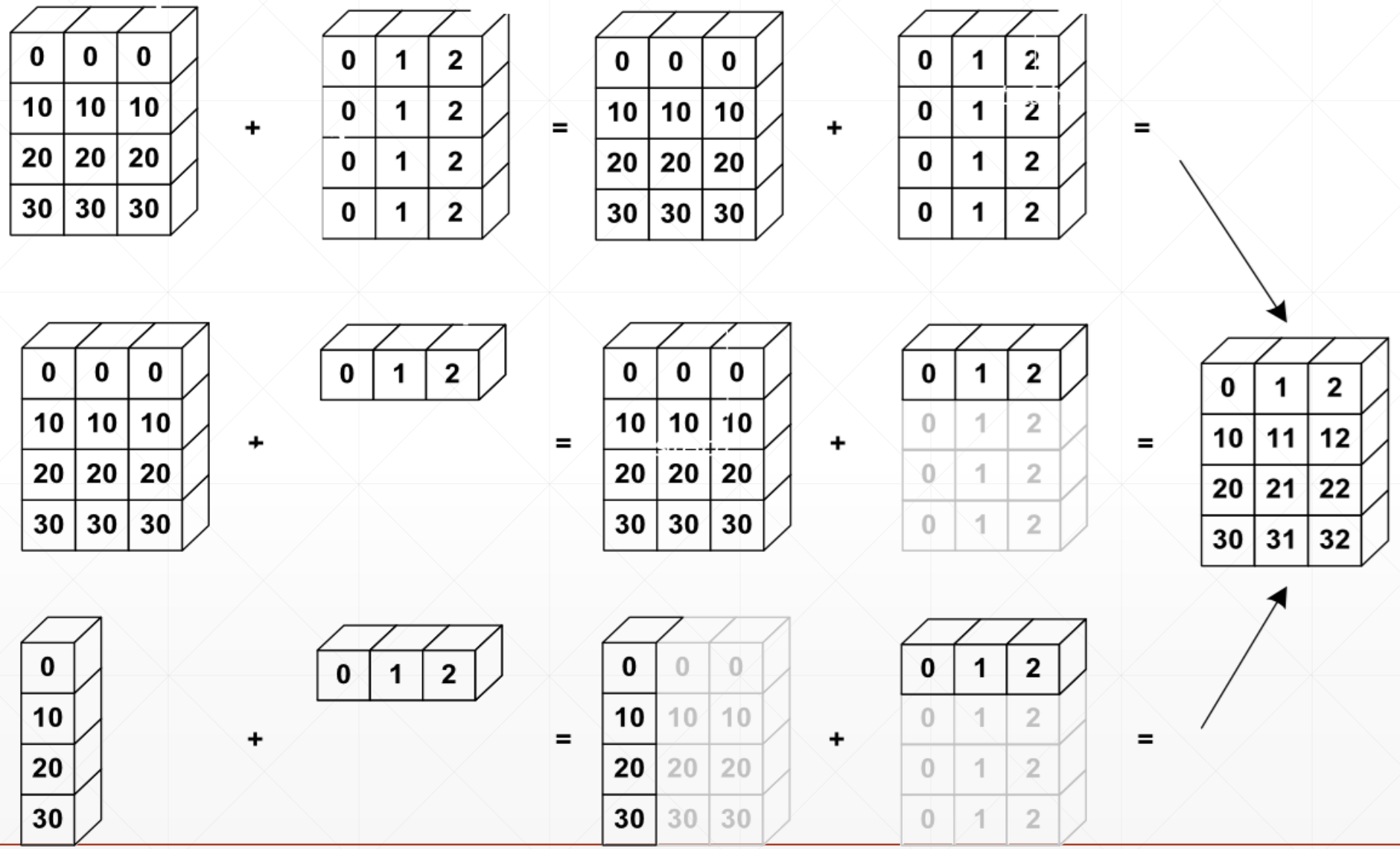
# Broadcasting

- expand
- without copying data
  - VS tf.tile
- tf.broadcast_to

https://blog.openai.com/generative-models/

# Key idea

- Insert 1 dim ahead if needed
- Expand dims with size 1 to same size
- Feature maps:  [4, 32, 32, 3]
- Bias: [3] → [1, 1, 1, 32] → [4, 32, 32, 3]


0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0 1 2
10 11 12
20 21 22
30 31 32
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=
0
10
20
30
+
0 1 2
=
0 0 0
10 10 10
20 20 20
30 30 30
+
0 1 2
0 1 2
0 1 2
0 1 2
=

# How to understand?

- When it has no axis
  - Create a new concept
  - [classes, students, scores] + [scores]
- When it has dim of size 1
  - Treat it shared by all
  - [classes, students, scores] + [students, 1]

# Why broadcasting?

- 1. for real demanding
  - [classes, students, scores]
  - Add bias for every student: +5 score
  - [4, 32, 8] + [4, 32, 8]
  - [4, 32, 8] + [5.0]
- 2. memory consumption
  - [4, 32, 8] → 1024
  - bias=[8]: [5.0,5.0,5.0,...] → 8

# Broadcastable?

- Match from Last dim!
  - If current dim=1, expand to same
  - If either has no dim, insert one dim and expand to same
  - otherwise, NOT broadcastable

# Situation 1:

- [4, 32, 14, 14]
- [1, 32, 1, 1] → [4, 32, 14, 14]

# Situation 2

- [4, 32, 14, 14]
- [14, 14] → [1, 1, 14, 14] → [4, 32, 14, 14]

# Situation 3

- [4, 32, 14, 14]
- [**2**, 32, 14, 14]
  - Dim 0 has dim, can NOT insert and expand to same
  - Dim 0 has distinct dim, NOT size 1
  - NOT broadcasting-able

# It's efficient and intuitive!

- [4, 32, 32, 3]
- + [3]
- + [32, 32, 1]
- + [4, 1, 1, 1]

# Broadcasting

```
In [25]: x=tf.random.normal([4,32,32,3])

In [27]: (x+tf.random.normal([3])).shape
Out[27]: TensorShape([4, 32, 32, 3])

In [28]: (x+tf.random.normal([32,32,1])).shape
Out[28]: TensorShape([4, 32, 32, 3])

In [29]: (x+tf.random.normal([4,1,1,1])).shape
Out[29]: TensorShape([4, 32, 32, 3])

In [31]: (x+tf.random.normal([1,4,1,1])).shape
InvalidArgumentError: Incompatible shapes: [4,32,32,3] vs. [1,4,1,1] [Op:Add]
name: add/
```

# tf.broadcast_to

```
In [35]: x.shape
Out[35]: TensorShape([4, 32, 32, 3])

In [36]: (x+tf.random.normal([4,1,1,1])).shape
Out[36]: TensorShape([4, 32, 32, 3])

In [37]: b=tf.broadcast_to(tf.random.normal([4,1,1,1]), [4,32,32,3])

In [38]: b.shape
Out[38]: TensorShape([4, 32, 32, 3])
```

# Broadcast VS Tile

```
In [4]: a=tf.ones([3,4])
In [5]: a1=tf.broadcast_to(a, [2,3,4])
<tf.Tensor: id=7, shape=(2, 3, 4), dtype=float32, numpy=
array([[[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]],

       [[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]]], dtype=float32)>

In [7]: a2=tf.expand_dims(a,axis=0)
Out[8]: TensorShape([1, 3, 4])
In [10]: a2=tf.tile(a2, [2,1,1])
<tf.Tensor: id=12, shape=(2, 3, 4), dtype=float32,
numpy=array([[[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]],

       [[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]]], dtype=float32)>
```

# 下一课时

数学运算

# Thank You.