An Introduction to Deep Learning With Python

[2.3] The gears of neural networks: tensor operations

Prof. Yuzo Iano

pgs: 38 - 43

Element-wise operations

In [1]:
```
def naive_relu(x):
    assert len(x.shape) == 2
    x = x.copy()
    for i in range (x.shape[0]):
        for j in range (x.shape[1]):
            x[i, j] = max(x[i, j], 0)
    return x
```

In [2]:
```
def naive_add(x, y):
    assert len(x.shape) == 2
    assert x.shape == y.shape

    x = x.copy()
    for i in range (x.shape[0]):
        for j in range (x.shape[1]):
            x[i, j] += y[i, j]
    return x
```

In [3]:
```
import numpy as np
x = np.arange(320).reshape(32, 10)
y = np.arange(10).reshape(1,10)

z = x + y
z = np.maximum(z, 0.)
z.shape
```

Out[3]: (32, 10)

Broadcasting

In [4]:
```
import numpy as np
x = np.random.random((64, 3, 32, 10))
y = np.random.random((32, 10))

z = np.maximum(x, y)
print('z = ', z.shape)
```

```
z =  (64, 3, 32, 10)
```

Tensor dot

In [5]:
```
def naive_vector_dot(x, y):
    assert len(x.shape) == 1
    assert len(y.shape) == 1
    assert x.shape[0] == y.shape[0]
    z = 0.
    for i in range (x.shape[0]):
        z += x[i] * y[i]
    return z
```

In [6]:
```
def naive_matrix_vector_dot(x, y):
    assert len(x.shape) == 2
    assert len(y.shape) == 1
    assert x.shape[1] == y.shape[0]

    z = np.zeros(x.shape[0])
    for i in range (x.shape[0]):
        for j in range (x.shape[1]):
            z[i] += x[i, j] * y[j]
    return z
```

In [7]:
```
def naive_matrix_vector_dot(x, y):
    z = np.zeros(x.shape[0])
    for i in range (x.shape[0]):
        z[i] = naive_vector_dot(x[i, :], y)
    return z
```

In [8]:
```
def naive_matrix_dot(x, y):
    assert len(x.shape) == 2
    assert len(y.shape) == 2
    assert x.shape[1] == y.shape[0]

    z = np.zeros((x.shape[0], y.shape[1]))
    for i in range (x.shape[0]):
        for j in range (y.shape[1]):
            row_x = x[i, :]
            column_y = y[:, j]
            z[i, j] = naive_vector_dot(row_x, column_y)
    return z
```

Tensor reshaping

In [9]:
```
x = np.array([[0., 1.],
              [2., 3.],
              [4., 5.]])
print('x = ', x)
print('x shape = ',x.shape)
```

```
x =  [[0. 1.]
 [2. 3.]
 [4. 5.]]
x shape =  (3, 2)
```

In [10]:
```
x = x.reshape((6, 1))
print('x = ', x)
print('x shape = ',x.shape)
```

```
x =  [[0.]
 [1.]
 [2.]
 [3.]
 [4.]
 [5.]]
x shape =  (6, 1)
```

In [11]:
```
x = x.reshape((2, 3))
print('x = ', x)
print('x shape = ',x.shape)
```

```
x =  [[0. 1. 2.]
 [3. 4. 5.]]
x shape =  (2, 3)
```

In [12]:
```
x = np.zeros((300, 20))
x = np.transpose(x)
print('x shape', x.shape)
```

```
x shape (20, 300)
```

Pablo Minango

- pablodavid218@gmail.com