**Anchor arrange:**

**Experiment:**

Take a pyramid feature map with size of (3, 2), and generate 2 bbox at one pixel as an example.

Then the output size of convolutional layer is (3, 2, 2\*4)

“Batch” axis is not mentioned because is unchanged.

|  |  |
| --- | --- |
| [a1, b1, c1, d1, a2, b2, c2, d2] | [a3, b3, c3, d3, a4, b4, c4, d4] |
| [a5, b5, c5, d5, a6, b6, c6, d6] | [a7, b7, c7, d7, a8, b8, c8, d8] |
| [a9, b9, c9, d9, a10, b10, c10, d10] | [a11, b11, c11, d11, a12, b12, c12, d12] |

After reshaped to (3\*2\*2, 4), the tensor will become:

[a1, b1, c1, d1]

[a2, b2, c2, d2]

[a3, b3, c3, d3]

[a4, b4, c4, d4]

[a5, b5, c5, d5]

[a6, b6, c6, d6]

[a7, b7, c7, d7]

[a8, b8, c8, d8]

[a9, b9, c9, d9]

[a10, b10, c10, d10]

[a11, b11, c11, d11]

[a12, b12, c12, d12]

**Conclusion:**

The anchors is arranged in order of anchors, transvers, longitude

(for Pixeli,j in pyramid feature map)

Pixel0,0-anchor1, Pixel0,0-anchor2, ...

Pixel0,1-anchor1, Pixel0,1-anchor2, ...

Pixel0,2-anchor1, Pixel0,2-anchor2, ...

...

Pixel1,0-anchor1, Pixel1,0-0\_anchor2, ...

Pixel1,1-anchor1, Pixel1,1-1\_anchor2, ...

Pixel1,2-anchor1, Pixel1,2-2\_anchor2, ...

…