

## 问题

在网络模型当中，经常要进行不同通道特征图的信息融合相加操作，以整合不同通道的信息，在具体实现方面特征的融合方式一共有两种，一种是 ResNet 和 FPN 等当中采用的 element-wise add，另一种是 DenseNet 等当中采用的 concat。他们之间有什么区别呢？

## add

以下是 keras 中对 add 的实现源码：

```
1 def _merge_function(self, inputs):
2     output = inputs[0]
3     for i in range(1, len(inputs)):
4         output += inputs[i]
5     return output
```

其中 inputs 为待融合的特征图，inputs[0]、inputs[1].....等的通道数一样，且特征图宽与高也一样。

从代码中可以很容易地看出，add 方式有以下特点：

1. 做的是对应通道对应位置的值的相加，通道数不变
2. 描述图像的特征个数不变，但是每个特征下的信息却增加了。

## concat

阅读下面代码实例帮助理解 concat 的工作原理：

```
1 t1 = [[1, 2, 3], [4, 5, 6]]
2 t2 = [[7, 8, 9], [10, 11, 12]]
3 tf.concat([t1, t2], 0) ==> [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
4 tf.concat([t1, t2], 1) ==> [[1, 2, 3, 7, 8, 9], [4, 5, 6, 10, 11, 12]]
5
6 # tensor t3 with shape [2, 3]
7 # tensor t4 with shape [2, 3]
8 tf.shape(tf.concat([t3, t4], 0)) ==> [4, 3]
9 tf.shape(tf.concat([t3, t4], 1)) ==> [2, 6]
10 -----
11 作者：KUNLI7
12 来源：CSDN
13 原文：https://blog.csdn.net/u012193416/article/details/79479935
14 版权声明：本文为博主原创文章，转载请附上博文链接！
```

在模型网路当中，数据通常为 4 个维度，即 num×channels×height×width，因此默认值 1 表示的是 channels 通道进行拼接。如：

```
1 combine = torch.cat([d1, add1, add2, add3, add4], 1)
```

从代码中可以很容易地看出，concat 方式有以下特点：

1. 做的是通道的合并，通道数变多了
2. 描述图像的特征个数变多，但是每个特征下的信息却不变。

## 多一点理解

---

add相当于加了一种prior，当两路输入可以具有“对应通道的特征图语义类似”的性质的时候，可以用add来替代concat，这样更节省参数和计算量（concat是add的2倍）

## 参考资料

---

[理解concat和add的不同作用](#)

[卷积神经网络中的add和concat区别](#)