

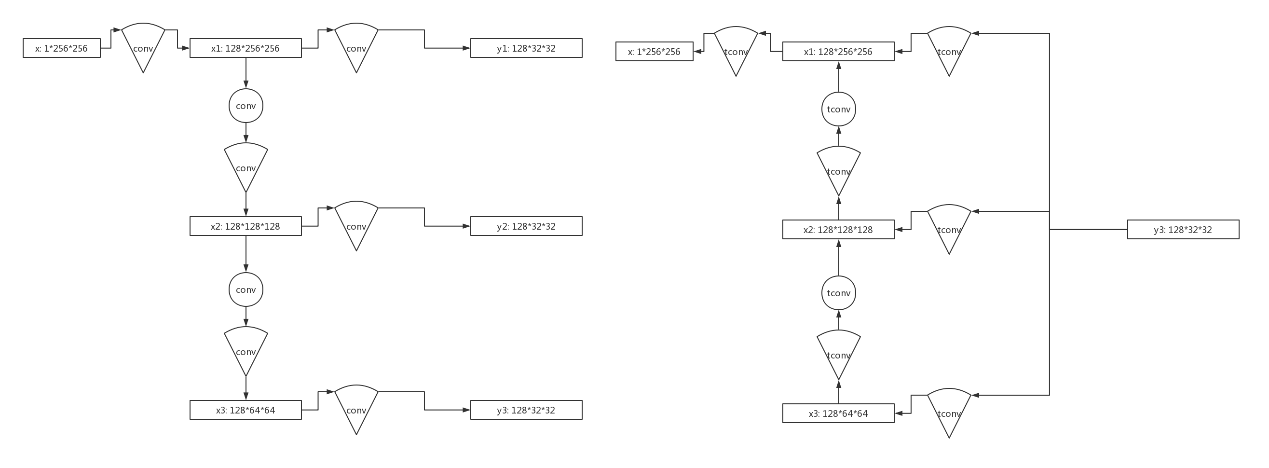
(L+2\*padding-ksize)/stride + 1

5.py结构

三角形conv表示的是降采样conv，圆形的则是没有降采样

编码器的三角形conv后会有一个gdn

解码器的三角形conv前会有一个gdn

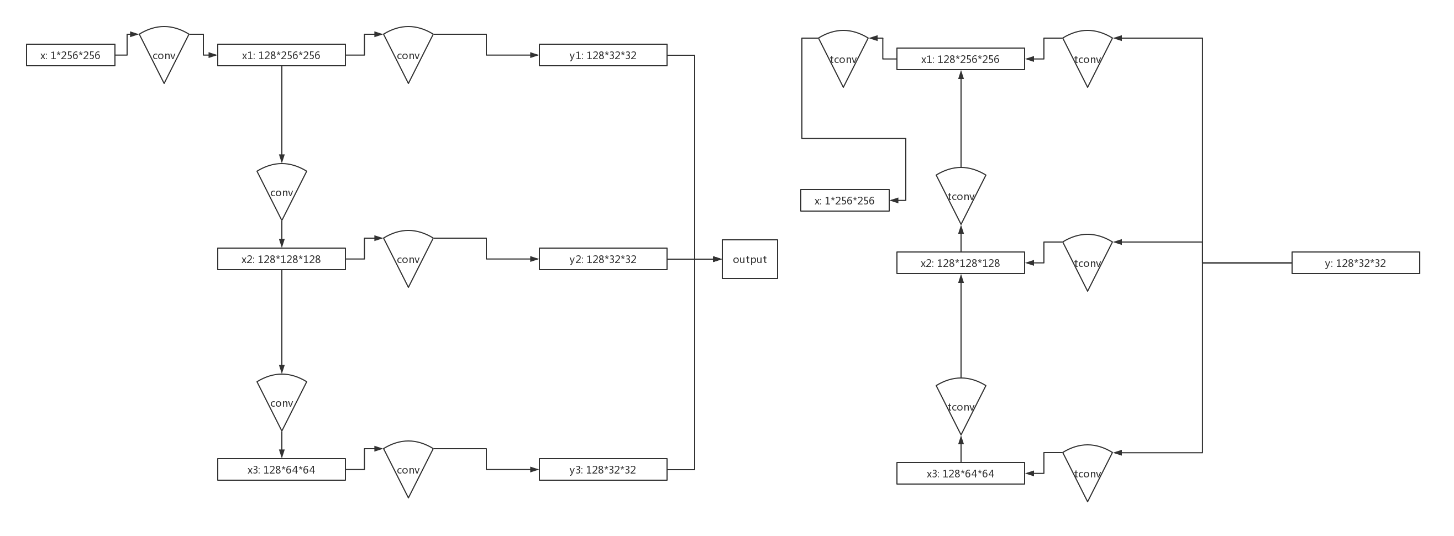


5\_1.py的改进为，增加了圆形conv层数量，但效果反而不如5.py

5\_2.py将竖直方向的加号改为减号

x1A = x1  
x1 = F.leaky\_relu(self.conv1\_0(x1))  
x1 = F.leaky\_relu(self.conv1\_1(x1))  
x1 = F.leaky\_relu(self.conv1\_2(x1))  
x1 = F.leaky\_relu(self.conv1\_3(x1))  
x1 = x1 - x1A

5\_3.py在5.py基础上直接去掉了圆形conv层



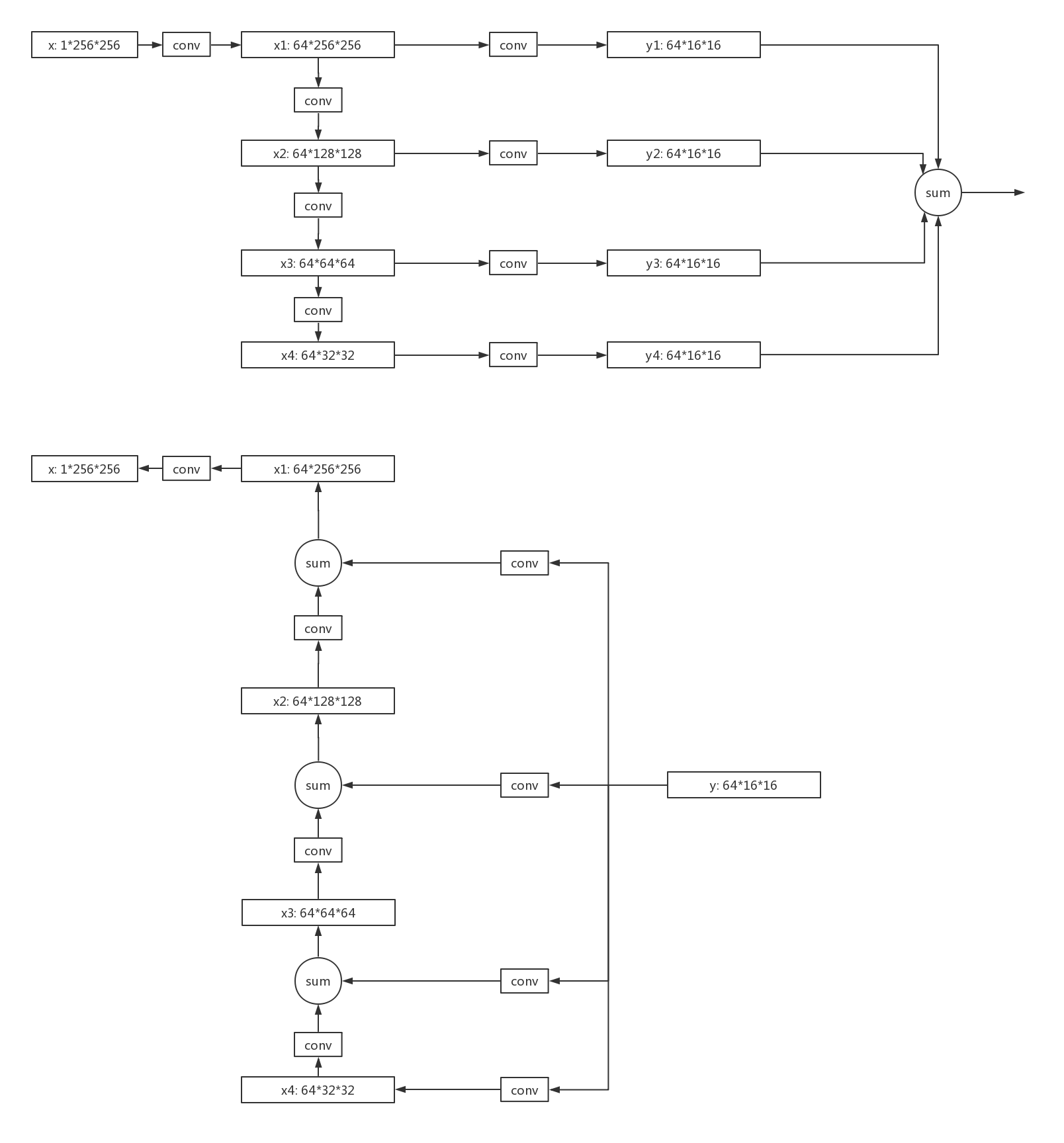
5\_4.py在横向的三角形conv前也加入了圆形conv层

5\_3\_3.py在5\_3.py的基础上，减小通道数进行测试

5\_3\_1.py在5\_3.py的基础上，输出降低为128\*16\*16

5\_3\_4.py 减小通道数到32

6\_1结构与下图类似，输出是64\*32\*32，下图实际上是6\_3：



6\_2改为32\*32\*32

6\_3改为64\*16\*16

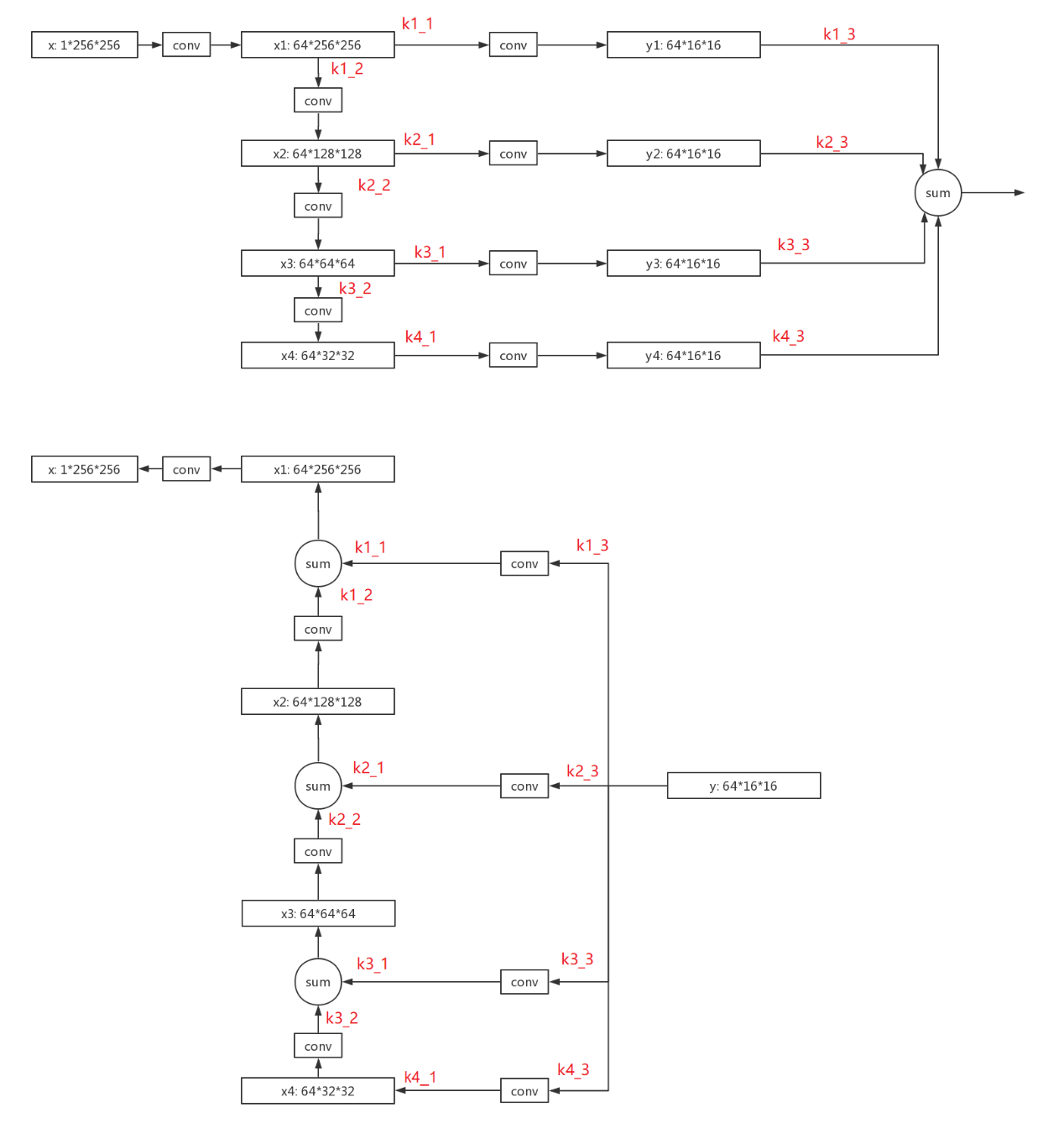
6\_3\_1在6\_3基础上，输出增加一个不降维的全连接层

6\_3\_2改为100\*16\*16

6\_3\_3改为128\*16\*16

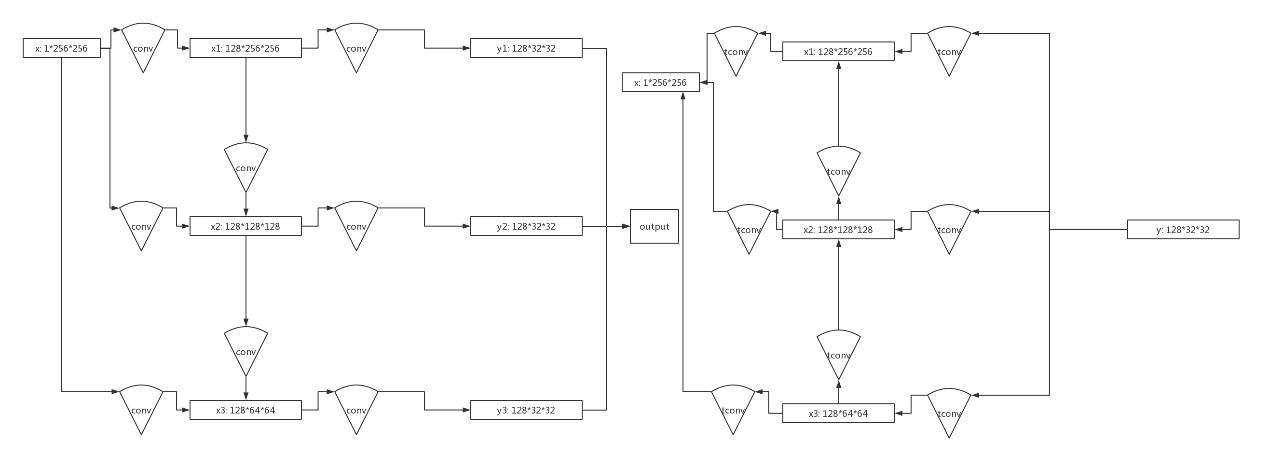
6\_4改为48\*32\*32

6\_3\_8的改动：



7.py

结构如下所示，编码器输出为64\*16\*16



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模型 | 编码器输出 | 相对尺寸/65536 | MSE | MS-SSIM | 备注 |
| 6\_1 | 64\*32\*32 | 1 | 85.505 | 0.961 |  |
| 6\_2 | 32\*32\*32 | 0.5 | 396.320 | 0.871 |  |
| 6\_3 | 64\*16\*16 | 0.25 | 364.660 | 0.890 |  |
| 6\_3\_1 | 64\*16\*16 | 0.25 | 6580.907 | 0.234 | 在6\_3基础上，编码器输出增加一个全连接层 |
| 6\_3\_2 | 100\*16\*16 | 0.390625 | 330.638 | 0.913 |  |
| 6\_3\_3 | 128\*16\*16 | 0.5 | 277.429 | 0.932 |  |
| 6\_3\_4 | 64\*16\*16 | 0.25 | 491.261 | 0.846 | 在6\_3\_3的编码器输出后，多层卷积缓慢降低通道数 |
| 6\_3\_5 | 64\*16\*16 | 0.25 | 322.488 | 0.881 | 在6\_3的编码器中最低降到8\*8，然后升采样叠加到输出；但解码器并没有增加类似的结构 |
| 6\_3\_6 | 32\*16\*16 | 0.125 | 485.108 | 0.830 |  |
| 6\_3\_7 | 64\*16\*16 | 0.25 | 253.927 | 0.957 | 与6\_3相同，但是用MS-SSIM训练 |
|  |  |  | 198.678 | 0.973 | 去掉量化 |
| 6\_3\_8 | 64\*16\*16 | 0.25 | 378.772 | 0.926 | 与6\_3相同，但是所有参与求和的项，增加可以学习的权重。具体见图 |
| 6\_3\_9 | 64\*16\*16 | 0.25 | 445.668 | 0.875 | 与6\_3相同，损失函数换成频域相关 |
|  |  |  | 114.430 | 0.967 | 损失函数换成核大小为3的边缘检测的带权MSE |
| 6\_3\_10 | 128\*16\*16 | 0.5 | 84.254 | 0.953 | 与6\_3\_9相同，损失函数带权MSE |
| 6\_3\_11 | 64\*16\*16 | 0.25 |  |  | 64个通道是用4个16通道拼接的 |
| 6\_4 | 48\*32\*32 | 0.75 | 259.919 | 0.951 |  |
| 6\_5 | 256\*8\*8 | 0.25 | 696.348 | 0.676 |  |
| 6\_6 | 128\*8\*8 | 0.125 | 741.043 | 0.702 |  |
| 7 | 64\*16\*16 | 0.25 | 391.095 | 0.881 | 7系列在6系列的基础上增加了连接 |
| 7\_1 | 32\*16\*16 | 0.125 | 474.792 | 0.838 |  |
| 8 | 64\*16\*16 | 0.25 | 363.881 | 0.875 |  |
| 9 | 64\*16\*16 | 0.25 | 134.994 | 0.984 | # 编解码器1 输入trainData 输出decData1 逼近trainData  # 编解码器2 输入decData1 输出decData2 逼近trainData - decData1 |
| 10 | 64\*16\*16 | 0.25 |  |  | 去掉了特征复用  发现效果基本不变 |

8.py的结构如下，在6\_3基础上增加了一些连接：

