1、网络参数包格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 偏移 | 内容 | 长度（字节） | 要求 | 建筑物网络示例 |
| 0 | 网络层数N | 4 | [1, 128] | 5 |
| 4 | 第一层网络配置layer1 | 32 | 详见Layer\_Param定义 |  |
| 36 | 第二层网络配置 | 32 |  |  |
| …… | …… |  |  |  |
| (N-1) \*32+4 | 第N层网络配置 | 32 |  |  |
| N\*32+4 | 第一层权值数据 | layer1. inputChannel\*  layer1. outputChannel\*  layer1. kernelRow\*  layer1. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] | char weight1  [1][64][3][3] |
|  | 第一层偏置数据 | layer1. biasFlag \*  layer1. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] | int bias1[64] |
|  | 第二层权值数据 | layer2. inputChannel\*  layer2. outputChannel\*  layer2. kernelRow\*  layer2. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] | char weight2  [64][128][3][3] |
|  | 第二层偏置数据 | layer2. biasFlag \*  layer2. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] | int bias2[128] |
|  | 第三层权值数据 | layer3. inputChannel\*  layer3. outputChannel\*  layer3. kernelRow\*  layer3. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] | char weight3  [128][256][3][3] |
|  | 第三层偏置数据 | layer3. biasFlag \*  layer3. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] | int bias3[256] |
|  | 第四层权值数据 | layer4. inputChannel\*  layer4. outputChannel\*  layer4. kernelRow\*  layer4. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] | char weight4  [256][256][3][3] |
|  | 第四层偏置数据 | layer4. biasFlag \*  layer4. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] | int bias4[256] |
|  | 第五层权值数据 | layer5. inputChannel\*  layer5. outputChannel\*  layer5. kernelRow\*  layer5. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] | unsigned short weight5  [256][32][3][3] |
|  | 第五层偏置数据 | layer5. biasFlag \*  layer5. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] | int bias5[32] |
|  | …… | …… |  |  |
|  | 第N层权值数据 | layerN. inputChannel\*  layerN. outputChannel\*  layerN. kernelRow\*  layerN. kernelCol | 排列方式为  [inputChannel]  [ outputChannel]  [ kernelRow]  [ kernelCol] |  |
|  | 第N层偏置数据 | layerN. biasFlag \*  layerN. outputChannel\*  BIAS\_SIZE | 排列方式为  [ outputChannel] |  |

注：

偏置数据32bit，BIAS\_SIZE=4

typedef struct

{

unsigned short inputChannel; //输入特征图通道数 1-1024

unsigned short outputChannel; //输出特征图通道数 1-1024

unsigned short kernelRow; //CNN参数--卷积核行数 1-5

unsigned short kernelCol; //CNN参数--卷积核列数 1-5

short inputLayer;

//RES参数--融合层1 0-MAX\_LAYER\_NUM-1 （-1 表示上一层）

//CAT参数--串联层1 0-MAX\_LAYER\_NUM-1 （-1 表示上一层）

//CNN参数--本层输入来自哪一层输出 0-MAX\_LAYER\_NUM-1 （-1 表示上一层）

short resLayer;

//RES参数--融合层2 0-MAX\_LAYER\_NUM-1

//CAT参数--串联层2 0-MAX\_LAYER\_NUM-1 （-1表示只串联一层）

//CNN参数--keepFlag=1时指定占位层数（TINY-YOLO用）

unsigned char computeType;

//计算类型： 0：卷积CNN； 1：融合RES；

// 2：串联CAT； 3：上采样UPSAMPLE;

// 0xff：无操作占位（TINY-YOLO用）

unsigned char biasFlag; //CNN参数--偏置标志,0：本层无偏置；1：本层有偏置

unsigned char actFlag; //CNN参数--激活标志,0：本层无激活；1：relu；

2：leakyRelu；3：非线性激活；4：部分非线性激活

unsigned char poolFlag; //CNN参数--池化标志,0：本层无池化；1：本层有池化

unsigned char keepFlag; //CNN参数--保留标志,0：本层中间结果无保留；1：本层中间结果（卷积后池化前）保留（TINY-YOLO用）

unsigned char cnnStride; //CNN参数--卷积步长，目前支持1和2

unsigned char poolStride; //CNN参数--池化步长，目前支持1和2

unsigned char poolSize; //CNN参数--池化尺寸，目前支持2和3

unsigned char cnnPadFlag; //CNN参数--卷积pad标志,0：本层卷积计算无pad；1：本层卷积计算有pad

unsigned char poolPadFlag; //CNN参数--池化pad标志,0：本层池化无pad；

1：本层池化有pad

unsigned char truncBit; //CNN/RES参数--数据截断位数

unsigned char upsampleStride; //UPSAMPLE参数--上采样倍数 1~2

unsigned char outputFlag; //输出层标志，1：该层不进行截断，输出32位定点数（机场算法）；2：该层按照16位输出截断，输出16位定点数（建筑算法）

unsigned char rsv[7]; //保留字

}Layer\_Param