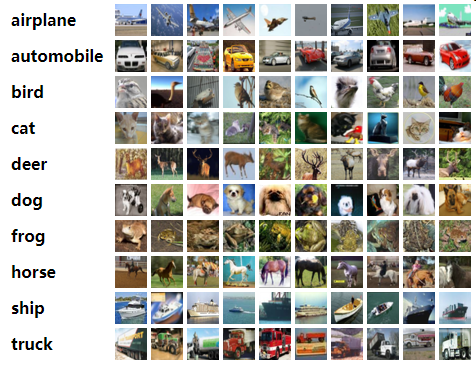
基于CNN神经网络的分类任务

基于Cifar-10数据集

计算机1604 王殊 1611640413

# 数据集描述

Cifar-10 的数据集包括训练集和测试集，样本的大小分别是50000和10000。样本被归为10类，分别是：



数据格式

1. 导入的数据格式：Python 使用 Pickle 序列化的数据。
2. 图像数据包括是 32x32 的 uint8 的图像数据。是一个10000x3072的uint8矩阵。每一行的前1024个uint8 长度是R分量，2048开始的是G分量和3072的是B分量。
3. 使用行优先顺序存储。

数据处理

1. Tensorflow 默认的卷积层的数据输入格式是4-D张量维度信息是：

[样本数量，矩阵高度，矩阵长度，通道数量]

所以需要在原数据的基础上进行转置操作。

1. 官方的样本数低于送入到神经网络的样本总数，所以需要对原数据增广处理同时对每个样本进行长度和高度缩放。。

# 神经网络

参考了基于VGG19的Cifar10修改版本的网络模型，总共包含xxx参数。

超参数

|  |  |  |
| --- | --- | --- |
| 参数名 | 值 | 意义 |
| conv\_strides | [1,1,1,1] | 过滤器的在输入样本的各个分量的移动步数 |
| pool\_kernel\_size | [1,3,3,1] | 最大池化层的过滤器大小 |
| pool\_strides | [1,2,2,1] | 最大池化层的过滤器移动的步长 |
| keep\_prob | 0.5 | L2正则化保留的比例 |
| regularize\_ratio | 0.0002 | 对给定过滤器的L2正则化器的缩放比例(将乘以过滤器) |
| batch\_size | 128 | 送入神经网络的一批次样本数量 |

网络结构

## VGG19网络：

针对于Cifar-10数据，不同于VGG19默认的以244x244像素的图片。需要做调整。

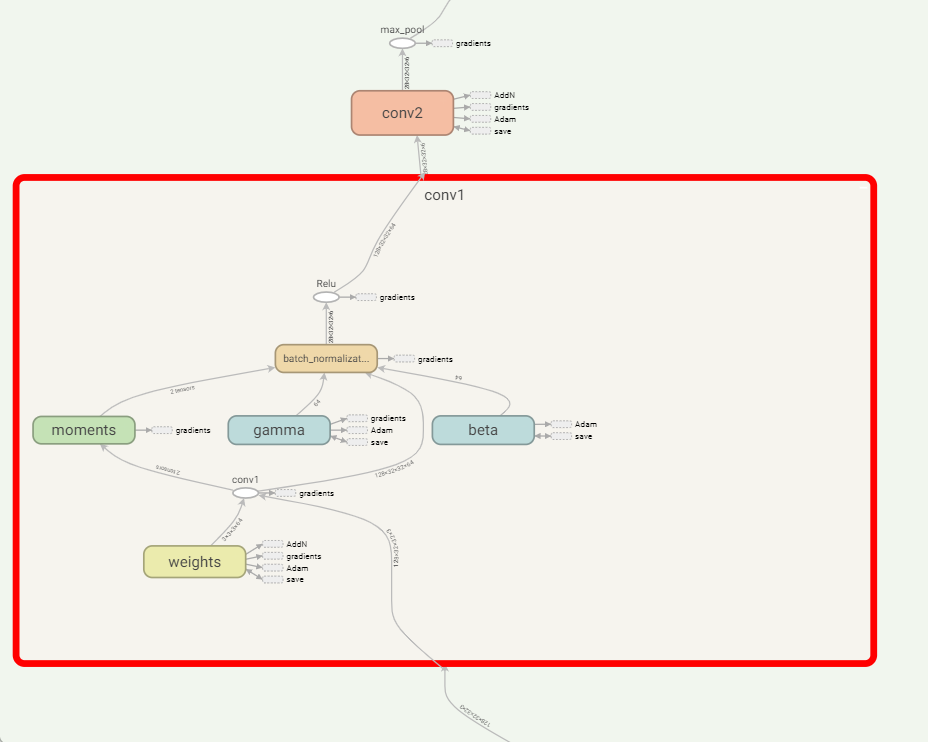
1. 输入的数据格式

|  |  |
| --- | --- |
| 名称 | 维度 |
| train\_images\_input | [128 32 32 3] |

1. 采用的结构

VGG19 网络中主要是由5个Block 结构构成：

* Block 结构：



每一个 Block 中包含有若干个卷积层以及最外层卷积层的最大池化层。具体的Block 配置：

采用的结构如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block | 层类别 | 输出大小 | 配置（大小，输出通道） | 激活函数 |
| Block1 | Conv1 | 32x32 | 大小：3x3，输出通道：64 | ReLU |
| Conv1 | 32x32 | 大小：3x3，输出通道：64 | ReLU |
| MaxPool | 16x16 | 步长大小：2x2，填充0保持信息不变 |  |
| Block2 | Conv1 | 16x16 | 大小：3x3，输出通道：128 | ReLU |
|  | Conv2 | 16x16 | 大小：3x3，输出通道：128 | ReLU |
|  | MaxPool | 8x8 | 步长大小：2x2，填充0保持信息不变 |  |
| Block3 | Conv1 | 8x8 | 大小：3x3，输出通道：256 | ReLU |
|  | Conv2 | 8x8 | 大小：3x3，输出通道：256 | ReLU |
|  | Conv3 | 8x8 | 大小：3x3，输出通道：256 | ReLU |
|  | Conv4 | 8x8 | 大小：3x3，输出通道：256 | ReLU |
|  | MaxPool | 4x4 | 步长大小：2x2，填充0保持信息不变 |  |
| Block4 | Conv1 | 4x4 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv2 | 4x4 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv3 | 4x4 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv4 | 4x4 | 大小：3x3，输出通道：512 | ReLU |
|  | MaxPool | 2x2 | 步长大小：2x2，填充0保持信息不变 |  |
| Block5 | Conv1 | 2x2 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv2 | 2x2 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv3 | 2x2 | 大小：3x3，输出通道：512 | ReLU |
|  | Conv4 | 2x2 | 大小：3x3，输出通道：512 | ReLU |
| Flatten |  | 1x1 | 大小：2x2x512 |  |
| FC1 |  | 1x1 | 大小：2048x4096 | ReLU |
|  | Dropout |  | 保留的比例：0.5 |  |
| FC2 |  | 1x1 | 大小：4096x4096 | ReLU |
|  | Dropout |  | 保留的比例：0.5 |  |
| Softmax |  | 1x1 | 大小：4096x10 | Softmax |

Tensorboard 输出的结构：

Xxx

## 损失函数

Softmax 输出的是10类的可能的概率，通过将输入的样本的标签书输入可以定义Softmax交叉熵，损失函数就是Softmax交叉熵的损失。

## 训练过程

首先是需要统计真实的图片标签与Softmax的输出 logits 的误差值加入到tf可损失集合中。这个集合包括了使用L2正则化的各层的卷积核的过滤器防止过拟合的张量。