17级种子班

数字图像处理课设

Week5-CNN总结

队伍成员：张志宇、李勉、刘羿

CONTENT

整体网络结构

使用框架：TensorFlow2.1

整体网络层数：

算上batch_normalization层及池化层共30层

不算batch_normalization层及池化层共13层

初始化：Xavier

损失函数：crossentropy

激活函数：Relu

优化器：Adam

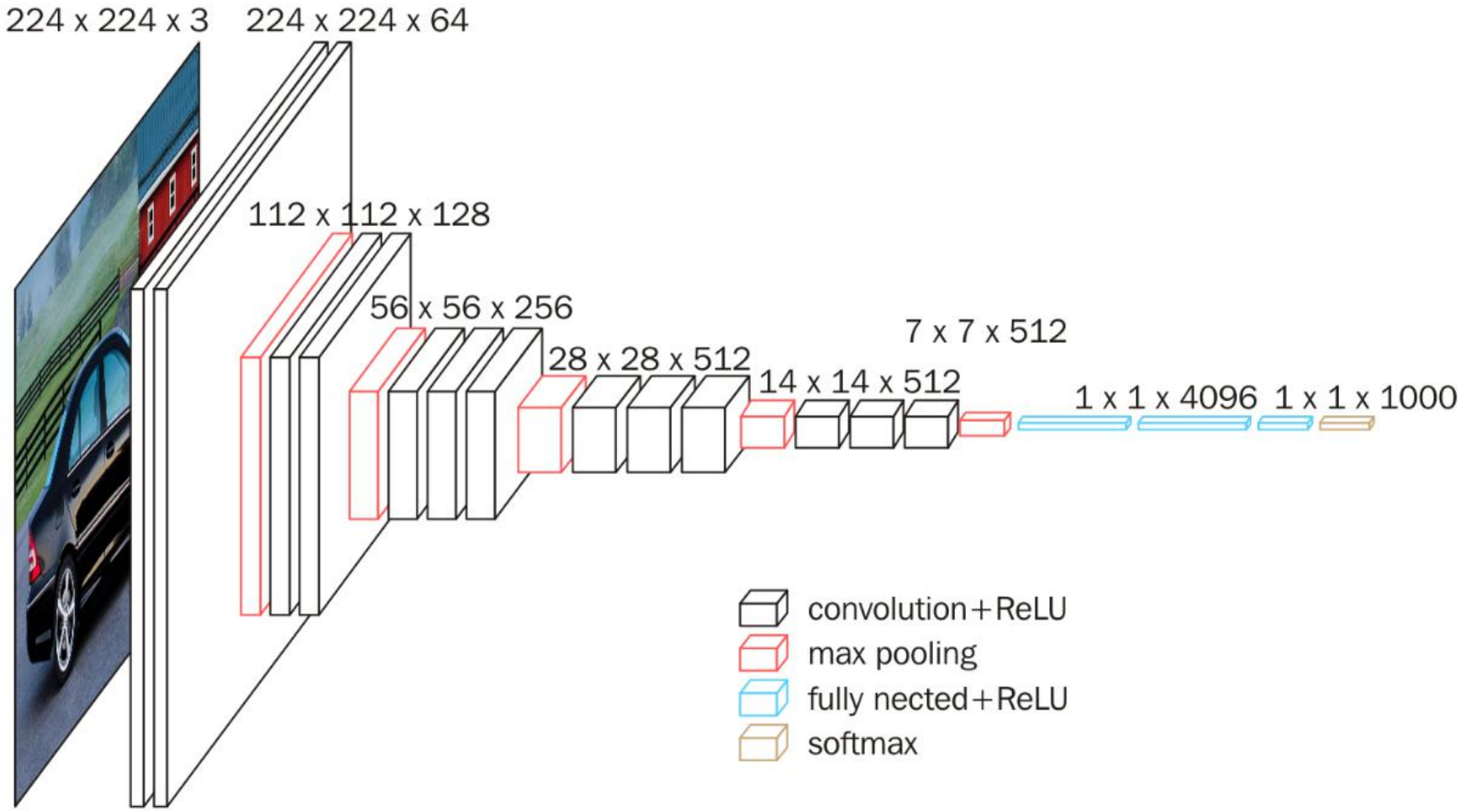
Model: "sequential"

Layer (type)	Output Shape	Param #
batch_normalization (BatchNo	(None, 32, 32, 3)	12
conv2d (Conv2D)	(None, 32, 32, 64)	1792
batch_normalization_1 (Batch	(None, 32, 32, 64)	256
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_2 (Batch	(None, 32, 32, 64)	256
conv2d_2 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
batch_normalization_3 (Batch	(None, 16, 16, 64)	256
conv2d_3 (Conv2D)	(None, 16, 16, 128)	73856
batch_normalization_4 (Batch	(None, 16, 16, 128)	512
conv2d_4 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 128)	0
batch_normalization_5 (Batch	(None, 8, 8, 128)	512
conv2d_5 (Conv2D)	(None, 8, 8, 256)	295168
batch_normalization_6 (Batch	(None, 8, 8, 256)	1024
conv2d_6 (Conv2D)	(None, 8, 8, 256)	590080
batch_normalization_7 (Batch	(None, 8, 8, 256)	1024
conv2d_7 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 256)	0
batch_normalization_8 (Batch	(None, 4, 4, 256)	1024
conv2d_8 (Conv2D)	(None, 4, 4, 512)	1180160
batch_normalization_9 (Batch	(None, 4, 4, 512)	2048
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2359808
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 512)	0
batch_normalization_10 (Batc	(None, 2, 2, 512)	2048
flatten (Flatten)	(None, 2048)	0
batch_normalization_11 (Batc	(None, 2048)	8192
dense (Dense)	(None, 512)	1049088
batch_normalization_12 (Batc	(None, 512)	2048
dense_1 (Dense)	(None, 10)	5130

Total params: 6,385,814
Trainable params: 6,376,208
Non-trainable params: 9,606

整体网络结构

参考网络结构：VGG16



整体网络结构

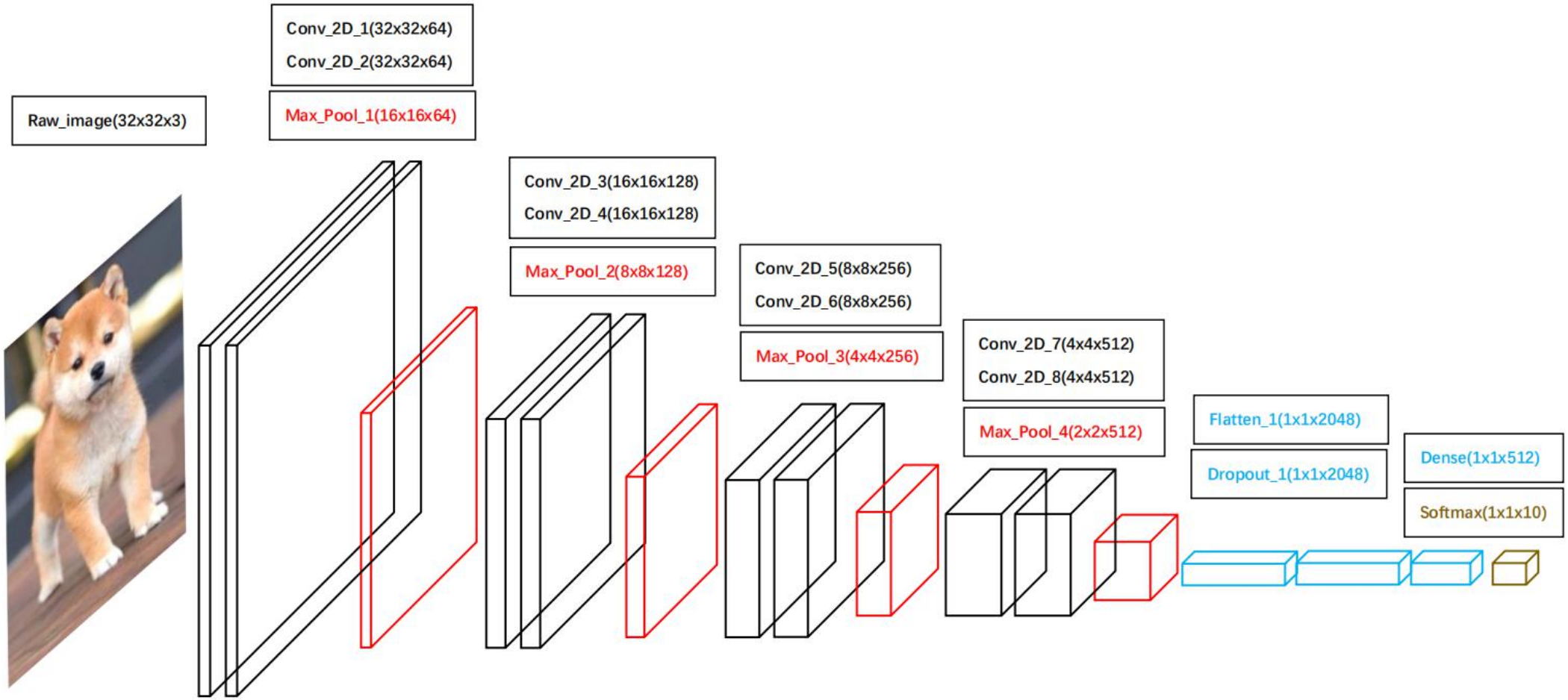
参考网络结构：VGG16

通道数翻倍：　　64 -> 128 -> 256 -> 512 -> 512 -> 512

高和宽变减半(池化层)：　　224 -> 112 -> 56 -> 28 -> 14 -> 7

整体网络结构

我们的网络结构



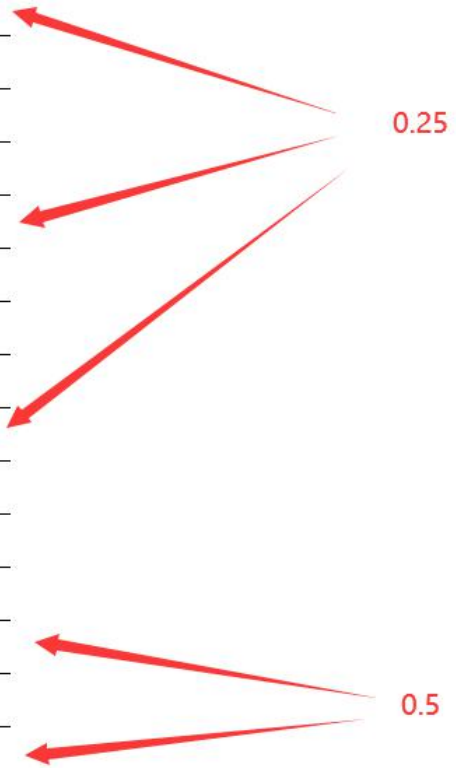
整体网络结构

加入Dropout层(0.25, 0.5)

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	1792
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0
conv2d_6 (Conv2D)	(None, 4, 4, 512)	1180160
conv2d_7 (Conv2D)	(None, 4, 4, 512)	2359808
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 512)	0
dropout_3 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dropout_4 (Dropout)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
dense_1 (Dense)	(None, 10)	5130

Total params: 5,739,594
Trainable params: 5,739,594
Non-trainable params: 0



整体网络结构

Dropout层修改(0.1-0.5)

Model: "sequential"

Layer (type)	Output Shape	Param #	
conv2d (Conv2D)	(None, 32, 32, 64)	1792	
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928	
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0	
dropout (Dropout)	(None, 16, 16, 64)	0	0.1
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856	
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584	
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 128)	0	
dropout_1 (Dropout)	(None, 8, 8, 128)	0	0.2
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168	
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080	
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 256)	0	
dropout_2 (Dropout)	(None, 4, 4, 256)	0	0.3
conv2d_6 (Conv2D)	(None, 4, 4, 512)	1180160	
conv2d_7 (Conv2D)	(None, 4, 4, 512)	2359808	
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 512)	0	
dropout_3 (Dropout)	(None, 2, 2, 512)	0	0.4
flatten (Flatten)	(None, 2048)	0	
dropout_4 (Dropout)	(None, 2048)	0	0.5
dense (Dense)	(None, 512)	1049088	
dense_1 (Dense)	(None, 10)	5130	

Total params: 5,739,594
Trainable params: 5,739,594
Non-trainable params: 0

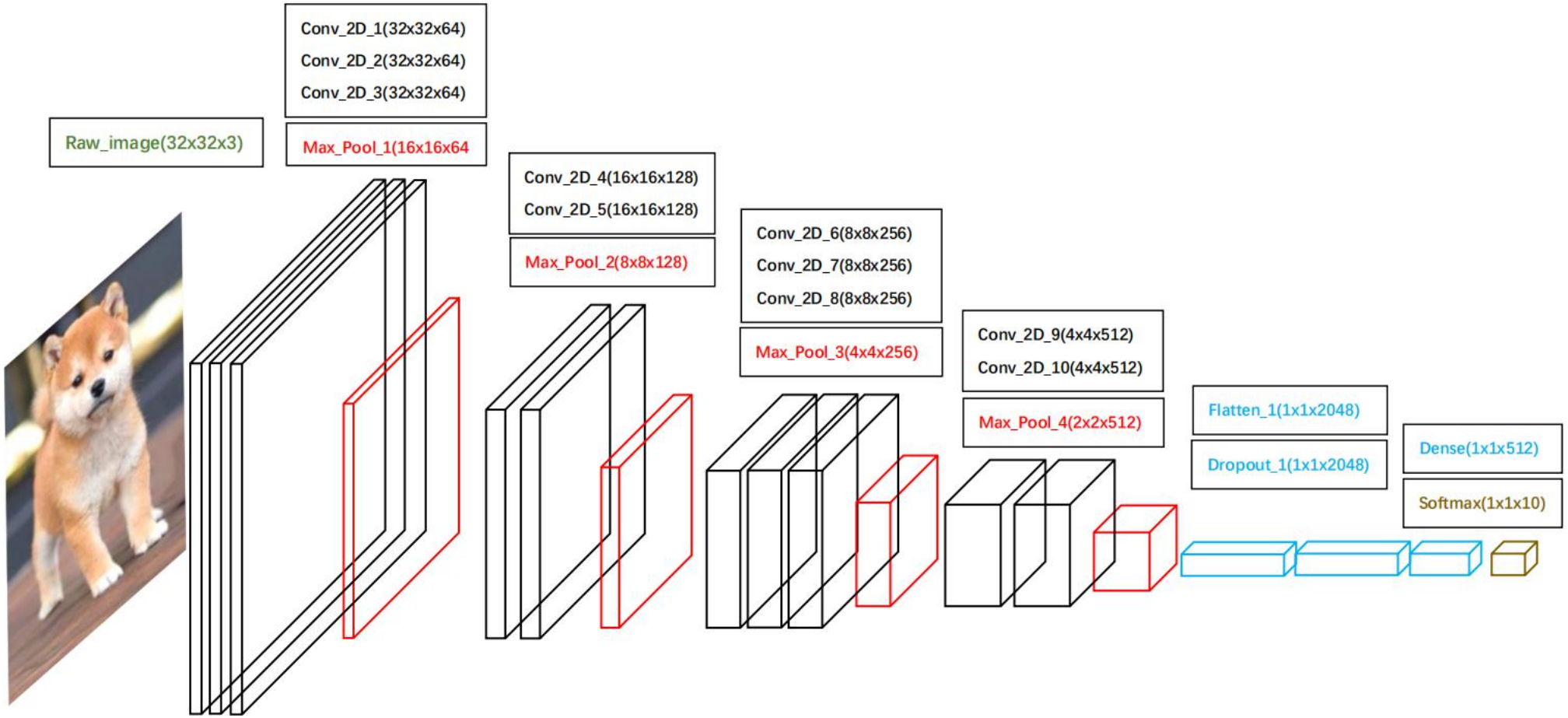
整体网络结构

batch_normalization层取代Dropout层

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization (BatchNo (None, 32, 32, 3)         12
_________________________________________________________________
conv2d (Conv2D)              (None, 32, 32, 64)        1792
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 64)        256
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 16, 128)       512
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 128)       147584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 256)         295168
_________________________________________________________________
batch_normalization_5 (Batch (None, 8, 8, 256)         1024
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 256)         590080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 256)         0
_________________________________________________________________
batch_normalization_6 (Batch (None, 4, 4, 256)         1024
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 4, 4, 512)         1180160
_________________________________________________________________
batch_normalization_7 (Batch (None, 4, 4, 512)         2048
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 4, 4, 512)         2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 512)         0
_________________________________________________________________
batch_normalization_8 (Batch (None, 2, 2, 512)         2048
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
batch_normalization_9 (Batch (None, 2048)              8192
_________________________________________________________________
dense (Dense)                (None, 512)               1049088
_________________________________________________________________
batch_normalization_10 (Batc (None, 512)               2048
_________________________________________________________________
dense_1 (Dense)              (None, 10)                5130
=================================================================
Total params: 5,757,526
Trainable params: 5,748,560
Non-trainable params: 8,966
_________________________________________________________________
```

整体网络结构

网络深度增加



数据预处理及增强

使用ImageDataGenerator

```
train_datagen = keras.preprocessing.image.ImageDataGenerator(
    featurewise_center=True,                    #使输入数据集去中心化(均值为0)
    featurewise_std_normalization=True,         #将输入除以数据集的标准差以完成标准化
    rotation_range=30,                          #图片随机转动的角度
    width_shift_range=0.12,                     #图片随机水平偏移的幅度
    height_shift_range=0.12,                    #图片随机竖直偏移的幅度
    zoom_range=0.12,                            #随机缩放的幅度
    rescale=1./255,                             #重放缩因子(在其他变换前)
    horizontal_flip=True)                       #随机水平翻转

test_datagen = keras.preprocessing.image.ImageDataGenerator(rescale=1./255)
```

实验结果

超参数搜索：

epochs = 30

batch_size_list = [32, 64, 128, 256, 512]

lr_list = [0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005]

搜索结果：

batch_size 越大，每代训练的训练时间越短，但缩短到一定程度就不再下降

batch_size 越大，模型收敛的越慢，直至不能收敛

batch_size 越大，过拟合会越晚体现出来

lr对结果影响不大，较小的lr表现更好

最终采用：

batch_size = 32　　lr = 0.0001　　epochs = 30/100

acc = 91.2%

实验总结

- CNN在图像识别方面的效果不错
- 在实验中深刻感受到了DeepLearning的玄学及不可解释性
- 对深度学习研究的困惑

Question & Answer