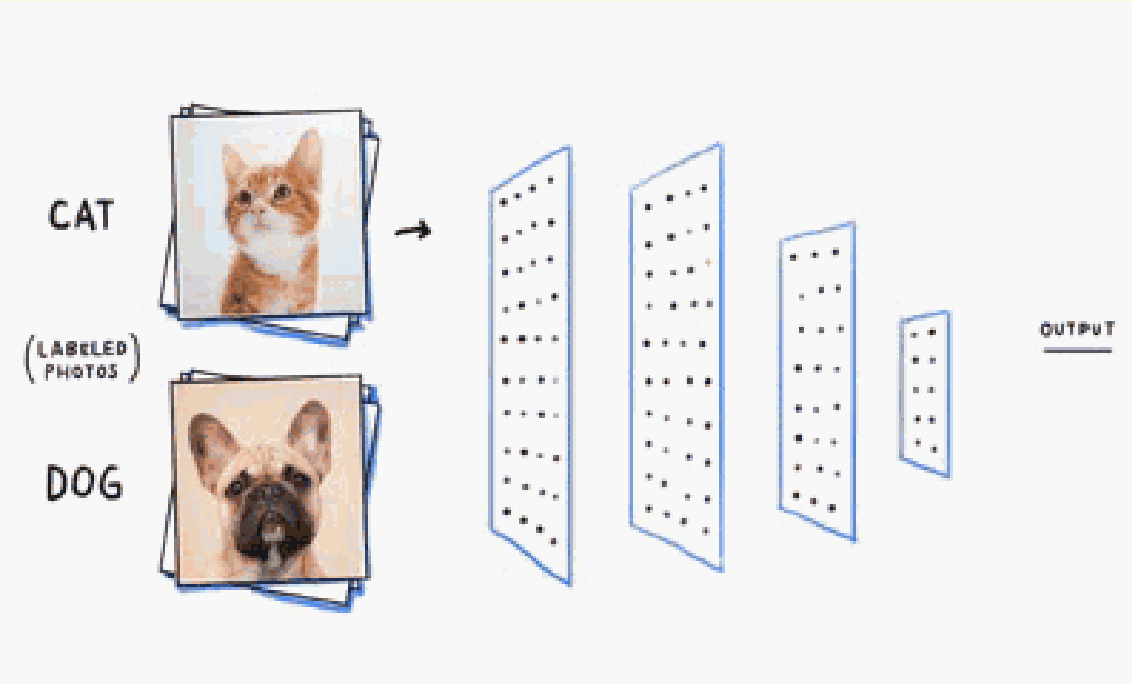
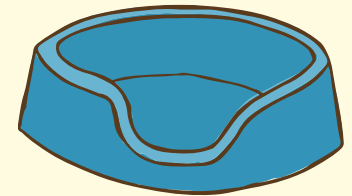




# 猫狗图像识别

Cats VS. Dogs



# CONTENTS

1



赛题介绍  
与  
数据预处理

---

2



多层感知器

---

3



卷积神经网络

---

4



模型预测  
与  
模型集成

---

1



# 赛题介绍 与 数据预处理

---

为了维护网站服务的安全，许多网站选择使用提问方式来识别访问者的身份，这一类问题对于人类来讲很简单、但对机器很难作答。例如网站的**验证码**或者**人类交互模式证明**（HIP for Human Interactive Proof），都是为了维护网站安全而设置的。

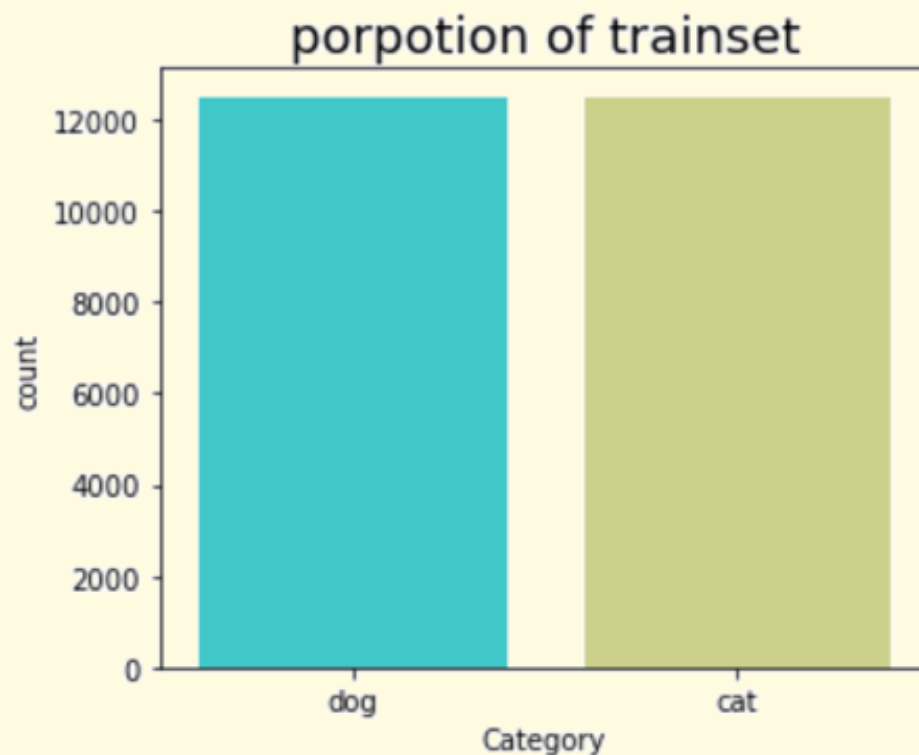
其中，ASIRRA（Animal Species Image Recognition for Restricting Access）就是一种人类交互模式证明，其工作原理就是**向人们提供猫和狗的图片**，以完成图片识别的速度准确率**区分人类和机器**。

本次比赛的数据由与ASIRRA合作的世界最大的为**宠物安家网站Petfinder提供**，所提供的图片分为训练集和测试集，训练集提供了每张图片猫或狗的类标识，测试集没有。参赛者需要以训练集图片训练模型，用已训练好的模型对测试集图片进行判别分类，并提交测试集分类结果。

将训练集和测试集压缩包解压，使用python中os包读取训练集文件夹中每个图片的文件名，将其对应路径与对应的类一同存于列表中。

题目数据集中共有**25000**个有类标的图片，**12500**没有类标的图片。

|   | Image         | Category |
|---|---------------|----------|
| 0 | dog.8011.jpg  | dog      |
| 1 | cat.5077.jpg  | cat      |
| 2 | dog.7322.jpg  | dog      |
| 3 | cat.2718.jpg  | cat      |
| 4 | cat.10151.jpg | cat      |
| 5 | cat.3406.jpg  | cat      |
| 6 | dog.1753.jpg  | dog      |
| 7 | cat.4369.jpg  | cat      |
| 8 | cat.7660.jpg  | cat      |
| 9 | dog.5535.jpg  | dog      |



数据集中提供的图片为**RGB色彩模式**图片，每个像素的颜色由R、G、B三个通道对应的数值决定，每个通道的数值**取值范围**为**0-255**。

而对于神经网络的计算来讲，255数值过大，可能在误差反向传递的时候造成**梯度爆炸问题**，同时过大的梯度使**学习率的选择**变得困难，所以需要图片进行**Min-Max归一化**处理，将像素值映射到[0, 1]范围内，具体转换如下公式所示：

$$newpixel = \frac{pixel}{255}$$

以**8 : 2**的比例将有类标的**25000张**图片随机划分成训练集和验证集，  
得到包含**20000张**图片的**训练集**和包含**5000张**图片的**验证集**。



多层感知器



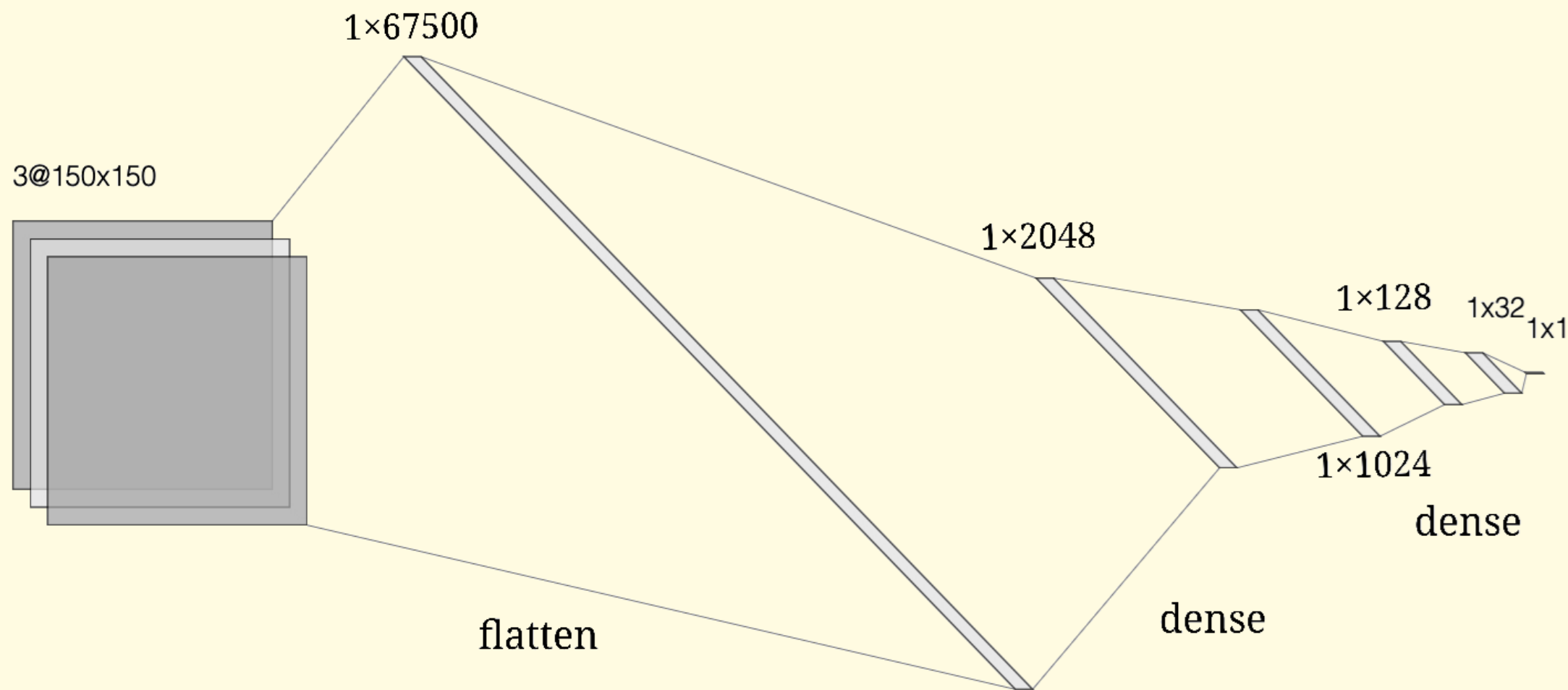
- ✓ 输入数据的尺寸： **$150 \times 150 \times 3$** ，输出的尺寸： **$1 \times 1$**
- ✓ 网络共有**10层**，包括全联接层**5个**，批标准化层**5个**
- ✓ 虽然在数据预处理部分，我们已经将数据归一化，但在网络拟合过程中，权重的更新会导致数据的分布发生改变，仍然存在发生梯度消失或梯度爆炸问题的可能，所以在多层感知器的每两个全联接层中间加入**批标准化层**，让数据在计算过程中仍保持分布
- ✓ 前4个全联接层激活函数为**ReLU函数**，最后一个全联接层激活函数为**Sigmoid函数**
- ✓ 可训练待估参数共**140,617,049个**

## 2 多层感知器——模型构建

MODEL1 多层感知器结构

| 序号 | 层        | 输入尺寸      | 输出尺寸    | 激活函数    | 待估参数个数    |
|----|----------|-----------|---------|---------|-----------|
| 1  | flatten层 | 150*150*3 | 1*67500 | ReLU    | 0         |
| 2  | 批标准化层    | 1*67500   | 1*67500 | —       | 270000    |
| 3  | 全联接层     | 1*67500   | 1*2048  | ReLU    | 138242048 |
| 4  | 批标准化层    | 1*2048    | 1*2048  | —       | 8192      |
| 5  | 全联接层     | 1*2048    | 1*1024  | ReLU    | 2098176   |
| 6  | 批标准化层    | 1*1024    | 1*1024  | —       | 4096      |
| 7  | 全联接层     | 1*1024    | 1*128   | ReLU    | 131200    |
| 8  | 批标准化层    | 1*128     | 1*128   | —       | 512       |
| 9  | 全联接层     | 1*128     | 1*32    | ReLU    | 4128      |
| 10 | 批标准化层    | 1*32      | 1*32    | —       | 128       |
| 11 | 全联接层     | 1*32      | 1*1     | Sigmiod | 33        |

可训练待估参数合计：140, 617, 049





模型在**第5次迭代**之后，验证集的准确率下降，损失上升，模型出现过拟合现象。选择最优模型，即迭代次数为5的模型作为最终模型进行保存。

### 径向基网络

| 模型<br>编号 | 激活函数 | 批数据量 | Dropout<br>层 | 批标准化<br>层 | 最佳迭代<br>次数 | 总用时<br>(秒) | 验证集准确率 |
|----------|------|------|--------------|-----------|------------|------------|--------|
| 1        | ReLU | 64   | —            | 使用        | 5          | 3986       | 65.82% |

注：模型总用时为截止到过拟合前最后一个epoch所用时间，后文出现总用时含义均于此处相同。

- ✓ 多层感知器理论上可以做图像识别，但会存在**诸多问题**：
- ✓ 第一，图像识别的输入数据常为矩阵形式，但多层感知器需要将输入数据降维，在降维的过程中，**数据结构遭到破坏**，原本紧连的像素点被拆开，可能导致一些**数据特征被忽略**；
- ✓ 第二，图像识别中的一张图片所包含的输入数据量很大，导致全联接层中的**待估参数非常多**，不仅计算**速度慢**，而且需要**很大的数据集**来防止模型过拟合。



# 卷积神经网络

---

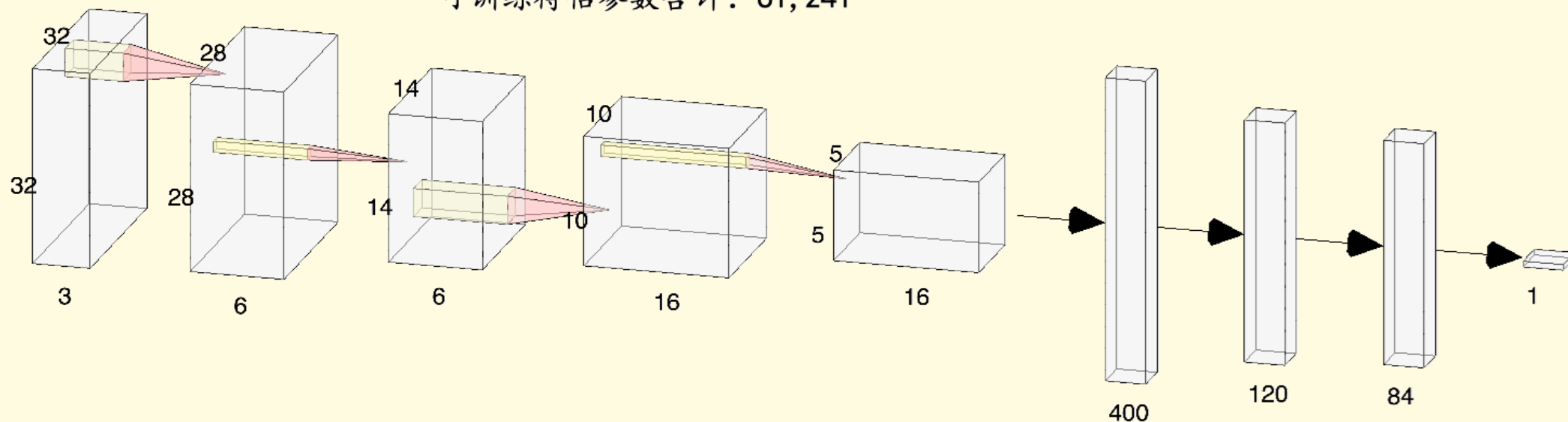
### 3 LeNet5——模型构建

✓ 按照经典卷积神经网络LeNet-5的结构构建网络。

MODEL2 LeNet-5

| 序号 | 层        | 卷积核    | 输入尺寸     | 输出尺寸     | 激活函数    | 待估参数个数 |
|----|----------|--------|----------|----------|---------|--------|
| 1  | 卷积层      | 5*5*6  | 32*32*3  | 28*28*6  | tanh    | 456    |
| 2  | 最大池化层    | -      | 28*28*6  | 14*14*6  | -       | 0      |
| 3  | 卷积层      | 5*5*16 | 14*14*6  | 10*10*16 | tanh    | 2416   |
| 4  | 最大池化层    | -      | 10*10*16 | 5*5*16   | -       | 0      |
| 5  | flatten层 | -      | 5*5*16   | 1*400    | -       | 0      |
| 6  | 全联接层     | -      | 1*400    | 1*120    | tanh    | 48120  |
| 7  | 全联接层     | -      | 1*120    | 1*84     | tanh    | 10164  |
| 8  | 全联接层     | -      | 1*84     | 1*1      | sigmoid | 85     |

可训练待估参数合计：61,241



- ✓ 原LeNet-5网络中激活函数为tanh函数和sigmoid函数
- ✓ 尝试将tanh函数换成**收敛速度快、不存在梯度消失问题**、具有**稀疏性**的ReLU函数
- ✓ 由于猫狗分类为二分类问题，选择保留原网络中输出响应变量的sigmoid函数



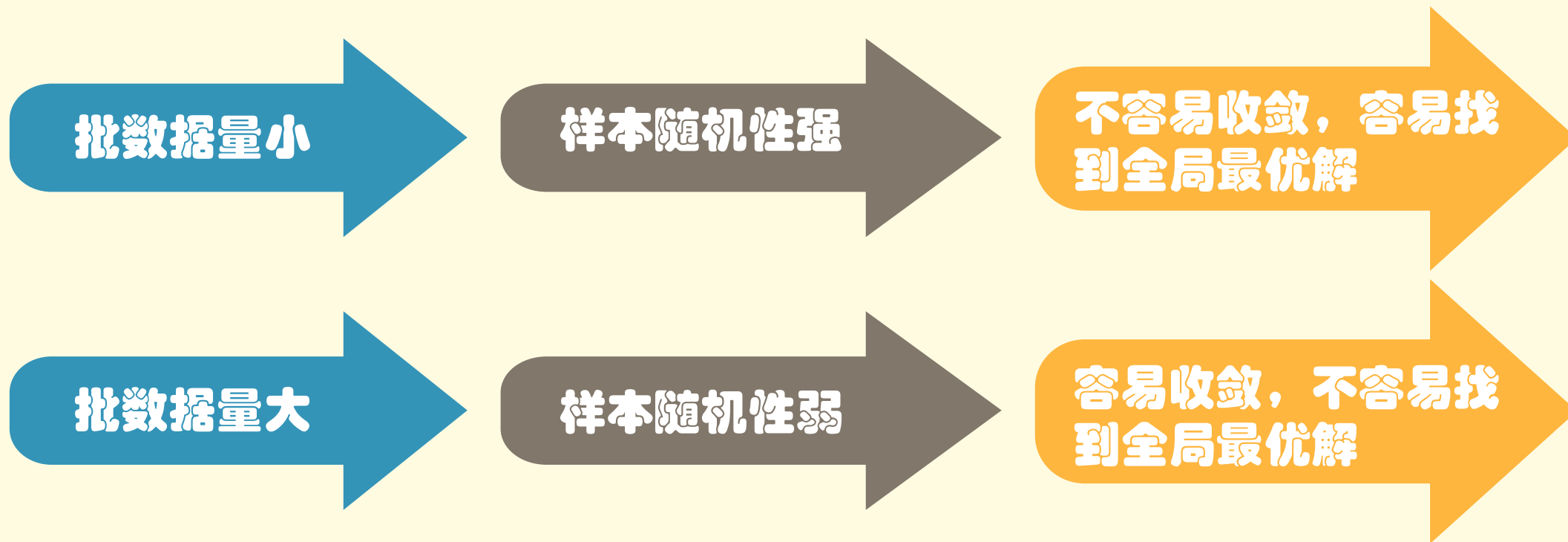
### 3 LeNet5——调整激活函数

LeNet-5 修改后模型对比

| 模型编号 | 模型                | 最佳迭代次数 | 总用时<br>(秒) | 验证集准<br>确率 |
|------|-------------------|--------|------------|------------|
| 1    | LeNet-5           | 8      | 576        | 72.44%     |
| 2    | 激活函数为ReLU的LeNet-5 | 6      | 474        | 74.42%     |

- ✓ 从上表可以看出，以ReLU函数作为模型的激活函数，无论从**收敛速度**还是**验证集准确率**方面评价，都要优于tanh函数
- ✓ 最终选择**ReLU函数**作为模型的激活函数

✓ 批数据量 (batch size) 是指一次训练所取的样本数



✓ 所以，理论上存在一个比较合适的批数据量，使得训练能够收敛的相对较快且收敛效果较好。

### 3 LeNet5——调整批数据量

选择尝试训练批数据量为**1**和**64**，两种LeNet5网络进行拟合：

| LeNet-5 修改后模型对比 |                |        |        |        |
|-----------------|----------------|--------|--------|--------|
| 编号              | 模型             | 最佳迭代次数 | 总用时（秒） | 验证集准确率 |
| 2               | LeNet-5        | 5      | 576    | 74.42% |
| 3               | LeNet-5批数据量为1  | 6      | 317    | 74.04% |
| 4               | LeNet-5批数据量为64 | 7      | 521    | 73.20% |

批数据量**没有**使模型**准确率提高**，难以此时就选择最优批数据量，将两种不同批数据量继续应用于后续模型，由后续模型准确率对模型批数据量进行选择。

### 3 LeNet5——Dropout层

LeNet5网络的体量不大，但也有**61241**↑待估参数，训练数据偏少，有可能会引起**过拟合**。

为了防止过拟合问题，提高网络的泛化能力，在LeNet5中的两个池化层后边各添加一个Dropout层，Dropout层随机使**20%**的检测器停止工作。

### 3 LeNet5——Dropout层

LeNet-5 添加Dropout层前后模型对比

| 模型编号 | 模型                        | 最佳迭代次数 | 总用时<br>(秒) | 验证集准确率 |
|------|---------------------------|--------|------------|--------|
| 2    | LeNet-5                   | 5      | 576        | 74.42% |
| 5    | LeNet-5添加Dropout层，批数据量为1  | 21     | 1562       | 75.48% |
| 6    | LeNet-5添加Dropout层，批数据量为64 | 13     | 916        | 75.84% |

- ✓ 添加Dropout层之后模型在过拟合前的**迭代次数明显上升**，模型5的迭代次数是模型2的**4倍以上**，模型6的迭代次数也是模型2的**2.5倍以上**。
- ✓ 批数据量过小的**不稳定性**在此处有所体现，批数据量为1的模型经过**21次**迭代才趋于稳定，是模型2的**4倍**以上。
- ✓ 加入Dropout层后，两个模型验证集准确率分别提高了**1.44%**和**2.64%**，模型6的准确率较之前最优模型（模型2）提高了**1.42%**。

### 3 LeNet5——批标准化层

- ✓ 在数据预处理中，为防止梯度爆炸问题，我们已经对图像像素数据进行了**Min-Max归一化**处理，使输入数据分布在  $[0,1]$  区间上。
- ✓ 但当网络开始训练之后，参数就会发生变化，高层的**输入分布**变化会非常剧烈（ Internal Covariate Shift ）
- ✓ 为使网络中数据分布更加稳定，以保证机器学习原假设**训练集分布和测试集分布相同**，添加**标准化层**。

### 3 LeNet5——批标准化层

- ✓ Keras中提供了两种标准化，**层标准化**和**批标准化**，
- ✓ 两种方法所选取用来计算均值和方差的总体是不同的，层标准化是对所有维度同时进行标准化，若各个维度表示的**特征量纲不一致**（比如R、G、B层的量纲），那么会导致模型的**表达能力下降**，所以此处选择为模型加入批标准化层。
- ✓ 在使用Dropout层的LeNet-5中，第二个卷积层前和第一个全联接层前，分别加入一个批标准化层。
- ✓ 批标准化层将输入数据的分布改为**标准正态分布**，为防止过多数据落入**ReLU函数的饱和区域**，导致**dead ReLU**问题，将模型激活函数换为**elu函数**进行拟合。

### 3 LeNet5——批标准化层

LeNet-5 添加Batch\_normalization层前后模型对比

| 模型编号 | 模型   | 最佳迭代<br>次数 | 总用时<br>(秒) | 验证集<br>准确率 |
|------|--|------------|------------|------------|
| 7    | LeNet-5添加Dropout层，<br>批数据量为64                              | 13         | 916        | 75.84%     |
| 8    | LeNet-5添加<br>Batch_normalization层，<br>批数据量为64              | 12         | 802        | 75.98%     |
| 9    | LeNet-5添加<br>Batch_normalization层，<br>批数据量为1               | 11         | 790s       | 75.18%     |
| 10   | LeNet-5添加<br>Batch_normalization层，<br>批数据量为64，激活函数<br>为elu | 15         | 1216s      | 75.56%     |

- ✓ 可看出模型的验证集准确率并没有太大改观。
- ✓ 可能由于LeNet-5的模型  
层数比较浅，隐藏层中数  
据分布的变动不大，不需  
要标准化也可以很好的拟  
合。

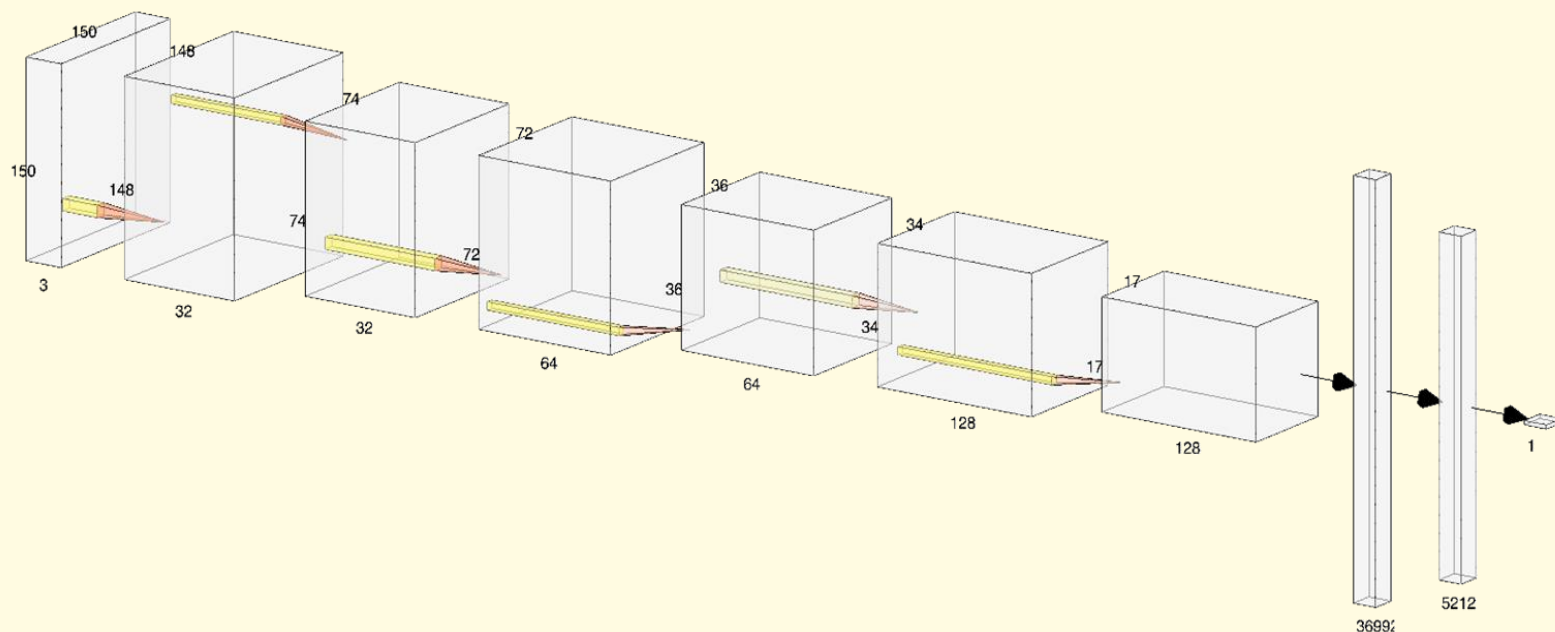


### 3 LeNet5——模型汇总

| 模型编号 | 激活函数 | 批数据量 | Dropout层 | 批标准化层 | 最佳迭代次数 | 总用时（秒） | 验证集准确率 |
|------|------|------|----------|-------|--------|--------|--------|
| 2    | tanh | —    | —        | —     | 6      | 576    | 72.22% |
| 3    | ReLU | —    | —        | —     | 5      | 474    | 74.42% |
| 4    | ReLU | 1    | —        | —     | 6      | 317    | 74.04% |
| 5    | ReLU | 64   | —        | —     | 7      | 521    | 73.20% |
| 6    | ReLU | 1    | 使用       | —     | 21     | 1562   | 75.48% |
| 7    | ReLU | 64   | 使用       | —     | 13     | 916    | 75.84% |
| 8    | ReLU | 64   | 使用       | 使用    | 12     | 802    | 75.98% |
| 9    | ReLU | 1    | 使用       | 使用    | 11     | 790    | 75.18% |
| 10   | elu  | 64   | 使用       | 使用    | 15     | 1216   | 75.56% |

- ✓ 经过多次模型调整后，模型验证集的准确率提高了**3.76%**
- ✓ 验证集准确率最高的模型为模型8，激活函数为ReLU、批数据量为64、使用Dropout层、使用批标准化层的LeNet5模型，其验证集准确率为**75.98%**
- ✓ 此模型的拟合效果并不十分理想，但由于模型仅有61241个待估参数，也有网络体量小，拟合速度快等优点。

- ✓ 即使经过了多次模型调整，LeNet5模型拟合的准确率依然不算精确，可能因为**网络结构过于简单，网络层数较少**，导致拟合效果不理想。想要进一步优化模型的图像识别准确率，则要考虑**加深卷积神经网络的深度**。



✓ 添加一个卷积层，共**3个卷积层**

✓ 考虑时间成本问题，去掉一个全联接层，共**2个全联接层**

MODEL3 三个卷积层的卷积神经网络

| 序号 | 层        | 卷积核     | 输入尺寸       | 输出尺寸       | 激活函数    | 待估参数个数   |
|----|----------|---------|------------|------------|---------|----------|
| 1  | 卷积层      | 3*3*32  | 150*150*3  | 148*148*32 | relu    | 896      |
| 2  | 最大池化层    | —       | 148*148*32 | 74*74*32   | —       | 0        |
| 3  | 卷积层      | 3*3*64  | 74*74*32   | 72*172*64  | relu    | 18496    |
| 4  | 最大池化层    | —       | 72*172*64  | 36*36*64   | —       | 0        |
| 5  | 卷积层      | 3*3*128 | 36*36*64   | 34*34*128  | relu    | 73856    |
| 6  | 最大池化层    |         | 34*34*128  | 17*17*128  |         | 0        |
| 7  | flatten层 | —       | 17*17*128  | 1*36922    | —       | 0        |
| 8  | 全联接层     | —       | 1*36922    | 1*512      | relu    | 18940416 |
| 9  | 全联接层     | —       | 1*512      | 1*1        | sigmoid | 513      |

可训练待估参数合计：19, 034, 177

3个卷积层的卷积神经网络

| 编号 | 最佳迭代次数 | 总用时（秒） | 验证集准确率 |
|----|--------|--------|--------|
| 11 | 5      | 2344s  | 82.90% |

- ✓ 此模型验证集准确率为**82.90%**，对比之前拟合的LeNet5模型中最优的模型验证集准确率为**75.98%**，可以看出模型准确率有很大提升。
- ✓ 但同时，此模型共有19,034,177个**待估参数**，是经典LeNet5的**310倍**之多，由于模型结构更加复杂，模型**拟合时间**也变为原来的**5倍以上**。

- ✓ 对3个卷积层的卷积神经网络的调参主要选择从**批数据量**、**dropout层**、**批标准化**三个方面对模型进行优化

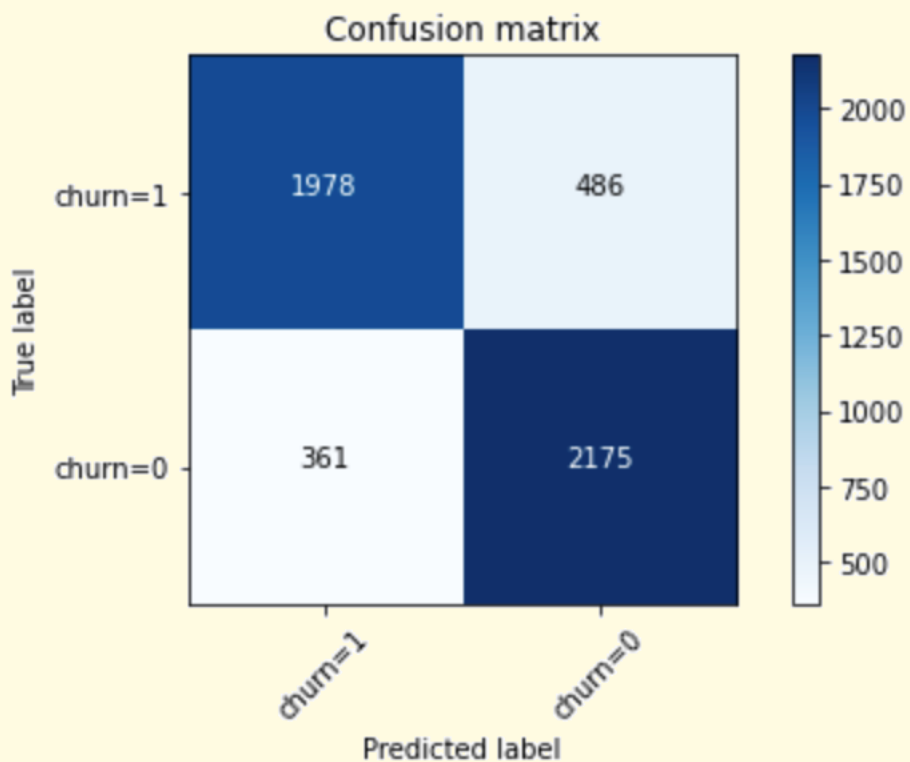
3个卷积层的卷积神经网络模型汇总

| 模型编号 | 激活函数 | 批数据量 | Dropout层 | 批标准化层 | 最佳迭代次数 | 总用时(秒) | 验证集准确率 |
|------|------|------|----------|-------|--------|--------|--------|
| 11   | ReLU | —    | —        | —     | 5      | 2344   | 82.90% |
| 12   | ReLU | 1    | —        | —     | 4      | 1827   | 81.50% |
| 13   | ReLU | 64   | —        | —     | 3      | 1598   | 82.84% |
| 14   | ReLU | 64   | 使用       | —     | 4      | 2050   | 80.80% |
| 15   | ReLU | 64   | 使用       | 使用    | 8      | 5001   | 79.92% |
| 16   | elu  | 64   | 使用       | 使用    | 6      | 3316   | 79.50% |

- ✓ 经过优化调参尝试，模型准确率始终没有超过原模型。

| 模型编号 | 模型种类         | 激活函数 | 批数据量 | Dropout层 | 批标准化层 | 最佳迭代次数 | 总用时(秒) | 验证集准确率 |
|------|--------------|------|------|----------|-------|--------|--------|--------|
| 1    | 多层感知器        | ReLU | 64   | —        | 使用    | 5      | 3986   | 65.82% |
| 2    |              | tanh | —    | —        | —     | 6      | 576    | 72.22% |
| 3    |              | ReLU | —    | —        | —     | 5      | 474    | 74.42% |
| 4    |              | ReLU | 1    | —        | —     | 6      | 317    | 74.04% |
| 5    |              | ReLU | 64   | —        | —     | 7      | 521    | 73.20% |
| 6    | LeNet5       | ReLU | 1    | 使用       | —     | 21     | 1562   | 75.48% |
| 7    |              | ReLU | 64   | 使用       | —     | 13     | 916    | 75.84% |
| 8    |              | ReLU | 64   | 使用       | 使用    | 12     | 802    | 75.98% |
| 9    |              | ReLU | 1    | 使用       | 使用    | 11     | 790    | 75.18% |
| 10   |              | elu  | 64   | 使用       | 使用    | 15     | 1216   | 75.56% |
| 11   | 3个卷积层的卷积神经网络 | ReLU | —    | —        | —     | 5      | 2344   | 82.90% |
| 12   |              | ReLU | 1    | —        | —     | 4      | 1827   | 81.50% |
| 13   |              | ReLU | 64   | —        | —     | 3      | 1598   | 82.84% |
| 14   |              | ReLU | 64   | 使用       | —     | 4      | 2050   | 80.80% |
| 15   |              | ReLU | 64   | 使用       | 使用    | 8      | 5001   | 79.92% |
| 16   |              | elu  | 64   | 使用       | 使用    | 6      | 3316   | 79.50% |

✓ 模型11，三个卷积层的神经网络，验证集准确率82.90%



Accuracy = 0.8306

Misclassification = 0.1694

Precision = 0.8576

Recall = 0.8173

F-Measure = 0.8371



4



# 模型预测 与 模型集成

---

- ✓ 为了提高其准确率、稳定性，降低结果的方差，避免过拟合，选择验证集准确率最高的5个模型（模型11、模型12、模型13、模型14、模型15），用**Bagging方法**对其进行模型集成。

$$Score = \frac{1}{5} \sum_{i=1}^5 score_i$$

- ✓ 分别用模型11、模型12、模型13、模型14、模型15对测试集图片类别进行预测，将每个图片的**预测结果进行平均**，将平均后值大于0.5的判断为猫，将小于等于0.5的判断为狗。

| 序号 | 图片       | 模型11   | 模型12   | 模型13   | 模型14   | 模型15   | 平均值    | 分类 |
|----|----------|--------|--------|--------|--------|--------|--------|----|
| 0  | 9733.jpg | 0.9064 | 0.0097 | 1.0000 | 0.0000 | 0.9989 | 0.5830 | 1  |
| 1  | 6400.jpg | 0.9947 | 0.0236 | 0.0000 | 0.0001 | 1.0000 | 0.4037 | 0  |
| 2  | 823.jpg  | 0.9929 | 0.9994 | 0.0720 | 0.4379 | 0.9882 | 0.6981 | 1  |
| 3  | 4217.jpg | 0.0157 | 0.0000 | 0.6740 | 0.9997 | 0.9842 | 0.5347 | 1  |
| 4  | 2666.jpg | 0.9843 | 0.0000 | 0.0006 | 0.8068 | 0.9648 | 0.5513 | 1  |



9733.Jpg / 预测类别: 猫



6400.Jpg / 预测类别: 狗



823.Jpg / 预测类别: 猫



4217.Jpg / 预测类别: 猫



6400.Jpg / 预测类别: 猫



- ✓ 模型训练集仅有20000张图片，**数据较少**，可考虑用**随机剪裁**、**镜像处理**、**图片着色调整**等方法进行图像加强。
- ✓ 由于个别模型训练所需时间过长，模型EarlyStopping的**patience参数**设置为3，patience参数**设置较小**，模型可能**欠拟合**。
- ✓ 由于计算设备计算速度受限，部分模型的**层数据量递减迅速**，尤其神经网络的全联接层，由于层与层之间连接所需的待估参数较多，为了减少待估参数、节约时间，全联接层的神经元数递减过快。可以考虑使用**1\*1的卷积核**减小数据维度，从而在保证全联接层的神经元数逐步递减的同时，提高计算速度。



谢谢欣赏

THANK YOU FOR WATCHING