

# “斗转星移” — 使用小数据集玩转深度学习

尹志



云朵网  
YDOOL.COM

# “斗转星移” — 使用小数据集玩转深度学习

1. 深度学习简介
2. 卷积神经网络
3. 小数据集玩转深度学习技巧
  - 3.1. 直接训练小数据集
  - 3.2. 数据增强
  - 3.3. 基于预训练网络进行特征提取
  - 3.4. 基于预训练网络进行精细调参
4. 斗转星移，更进一步

A: 听说你用深度学习做图片分类? 帮我分类下我家狗狗品种?

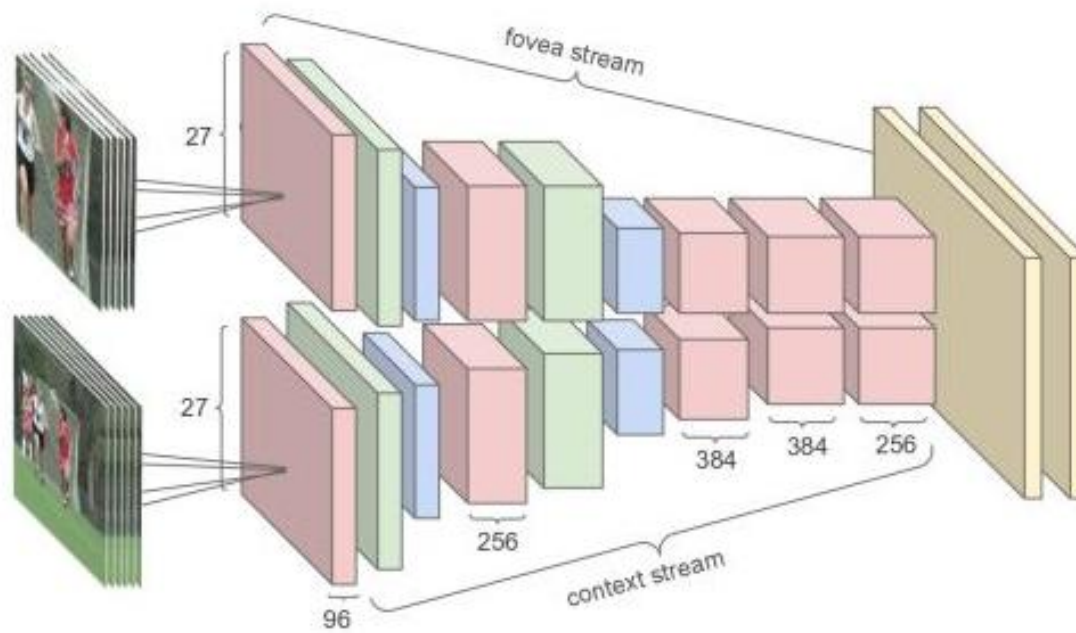




目瞪口呆



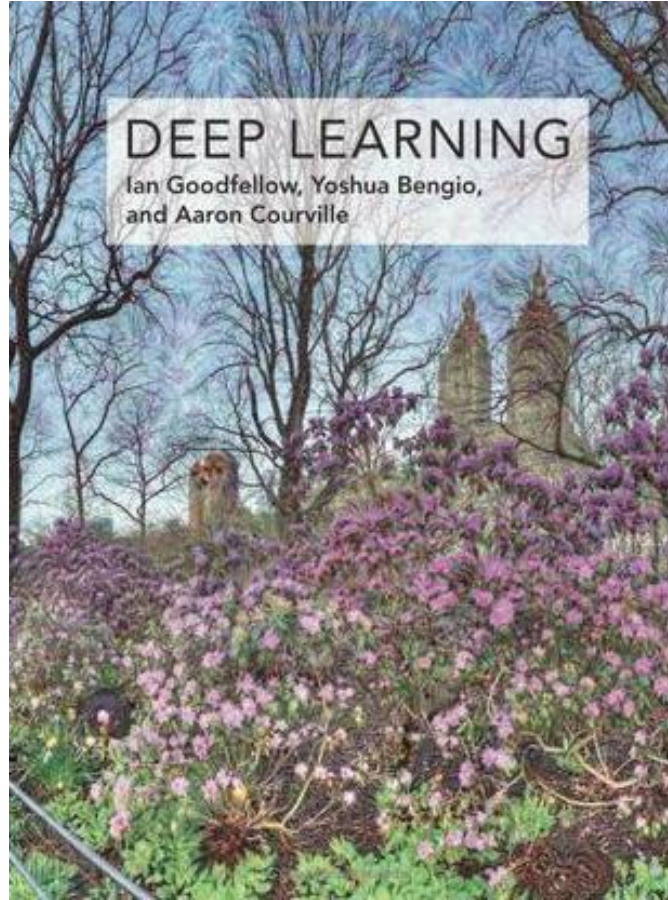
# 深度学习简介



# 深度学习简介



# 深度学习简介



# 深度学习简介

深度学习是近年来兴起的机器学习范式，.....深度学习利用多层神经网络结构，从大数据中学习现实世界中各类事物能被用于计算机计算的表示形式（比如图像中的事物、音频中的声音等），被认为是智能机器可能的“大脑结构”

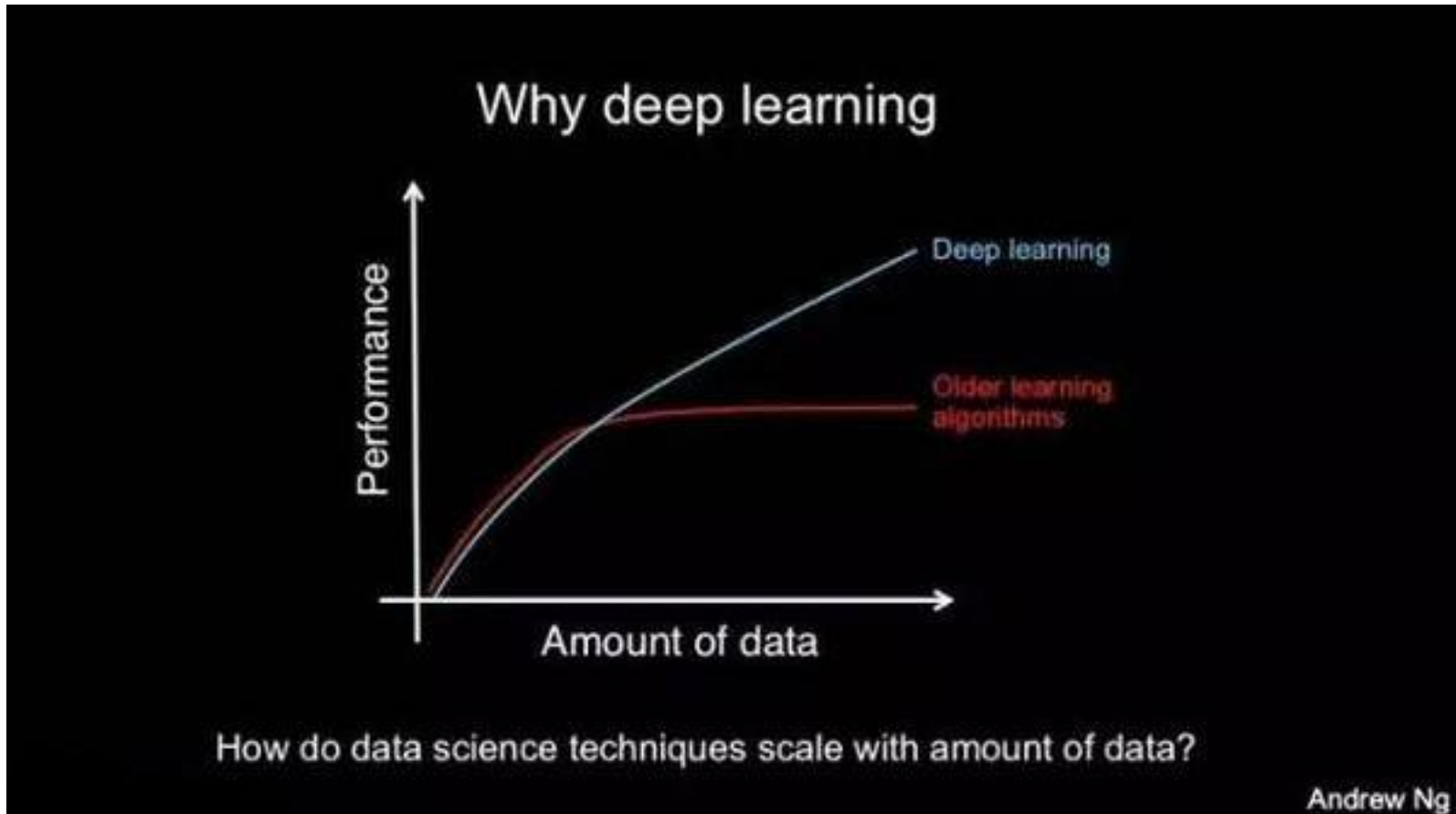
----- 《大数据智能》



# 深度学习简介

1. 通过算法，在少量人为经验情况下，自动从数据中抽取出合适的特征
2. 能够处理更多数据，更大维度
3. 能够处理更加复杂的数据，更加复杂的任务（图像、文本、语音等）
4. 端到端服务
5. 通用性很强，一个深度网络模型一旦训练出来，可能可以适合很多的场景

# 深度学习简介



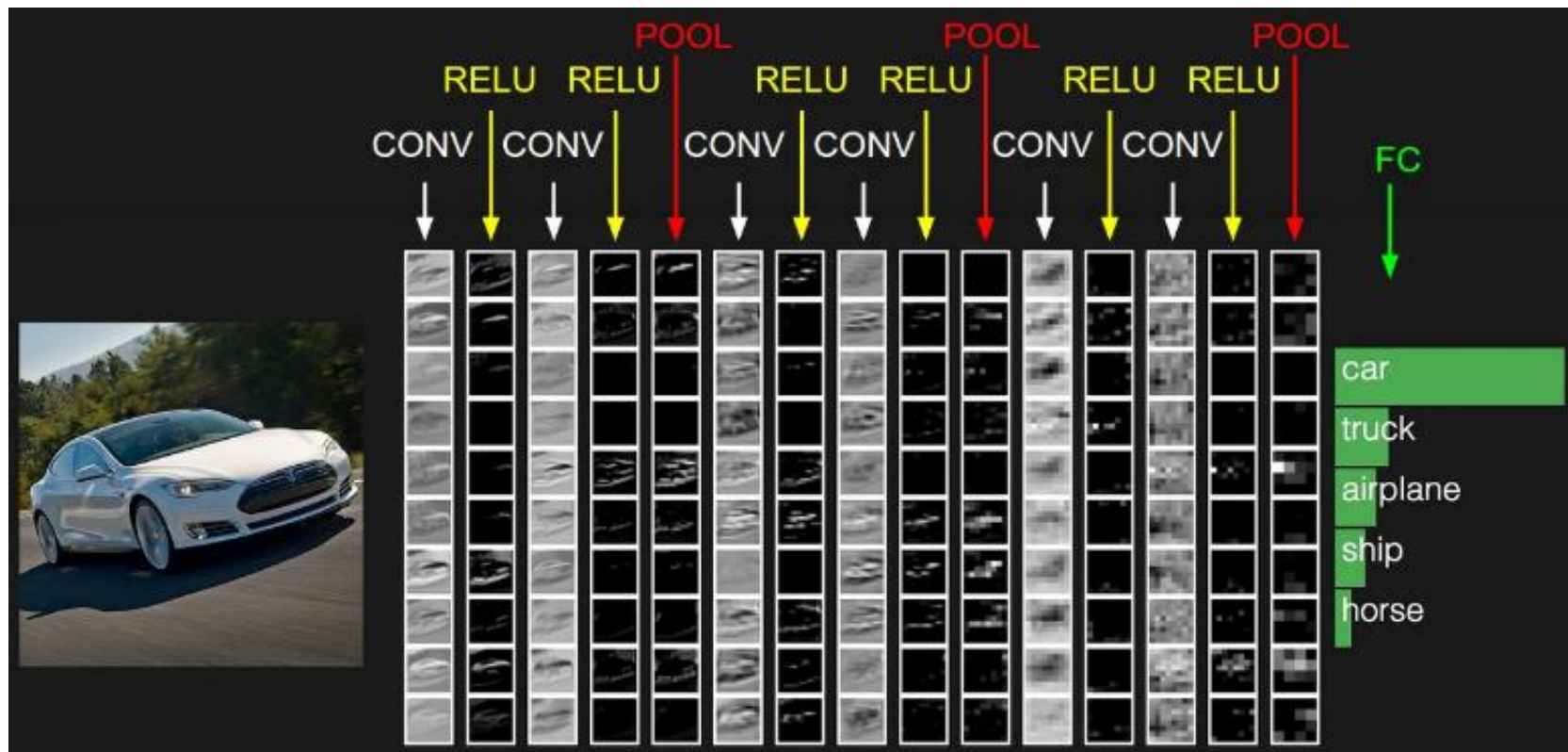
# 深度学习简介

1. 复杂的模型-深度神经网络 (A)
2. 大量的数据 (B)
3. 大量的计算能力 (C)

# 卷积神经网络

任务：图像分类

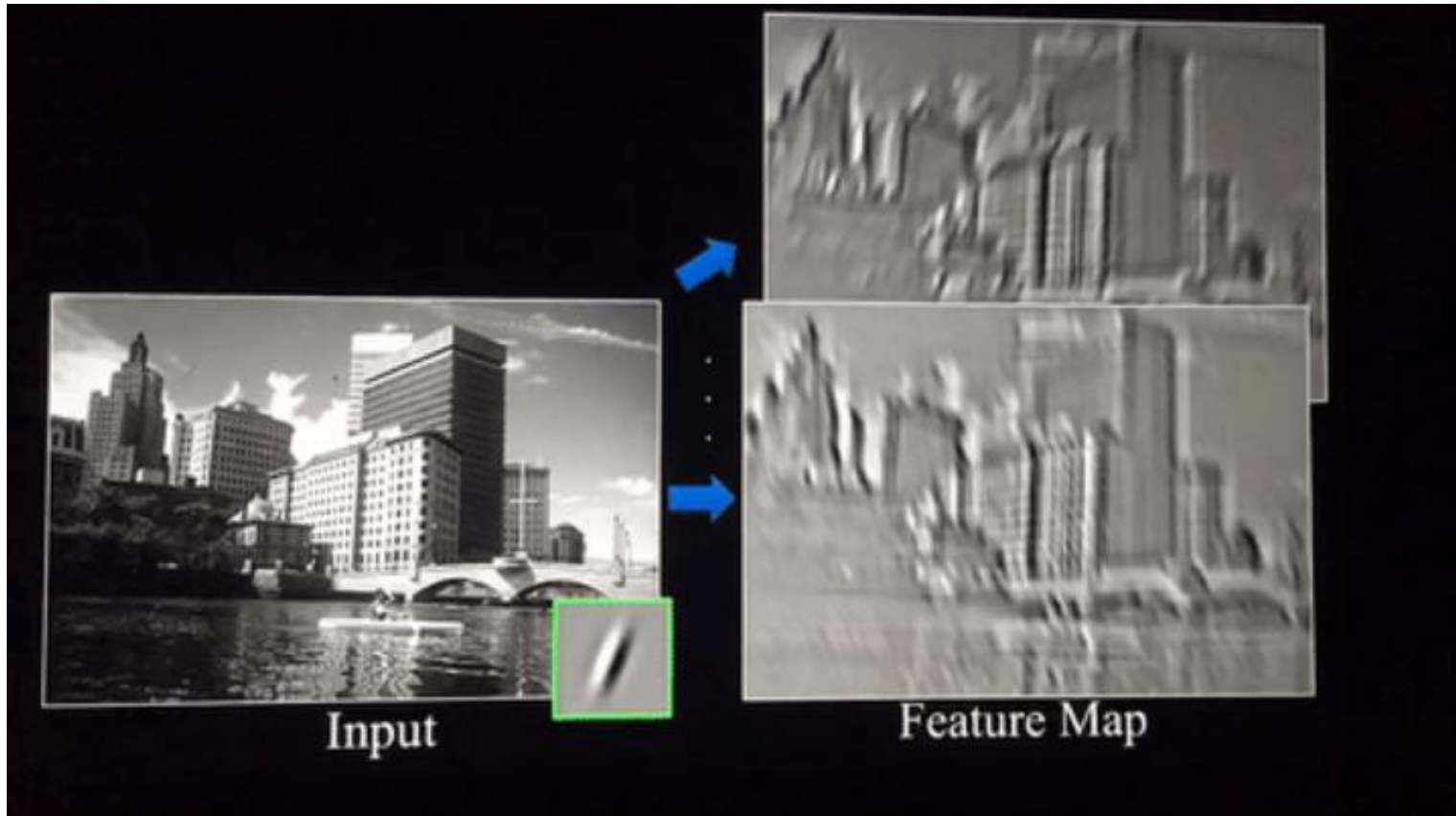
# 卷积神经网络



# 卷积神经网络

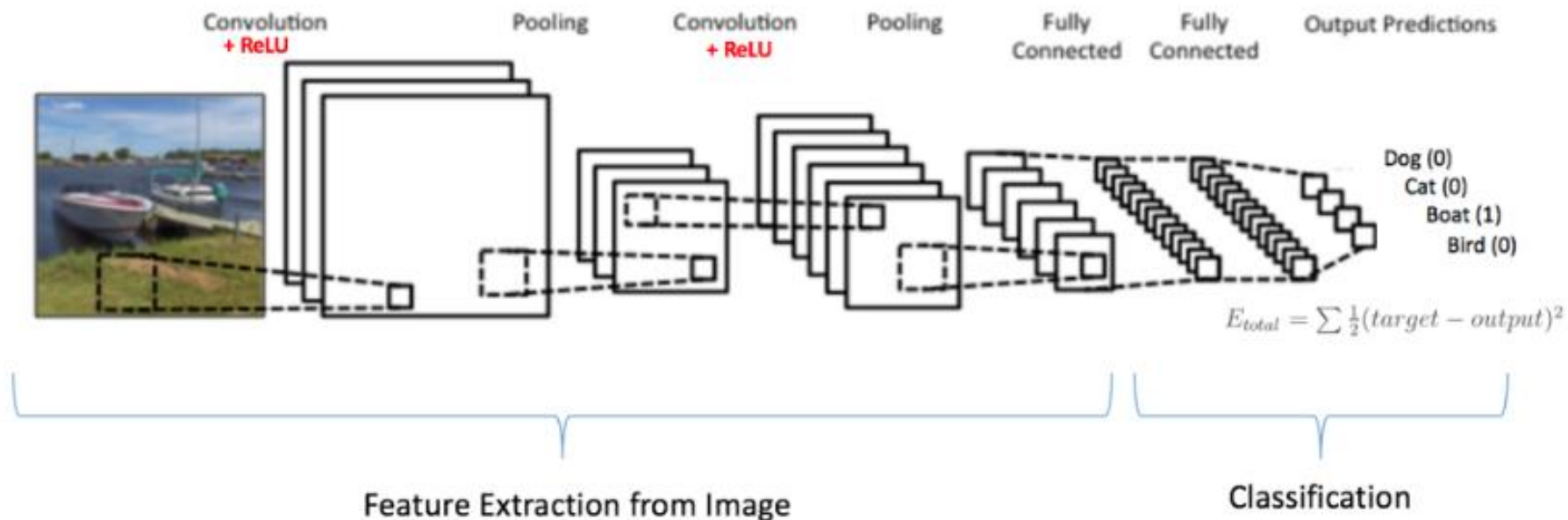
1. Convolution
2. Non Linearity (ReLU)
3. Pooling
4. Classification (Fully Connected Layer)

# 卷积神经网络



Convolution

# 卷积神经网络

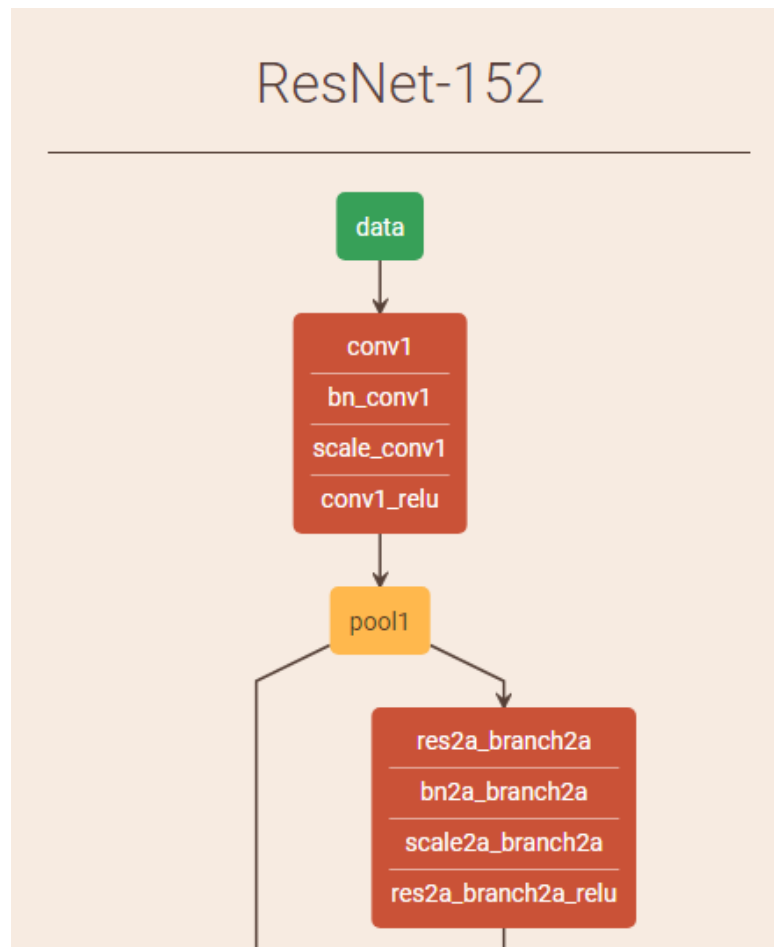




# 卷积神经网络

道理我都懂，  
然而还是训练不出一个好的网络

# 卷积神经网络



ResNet-152

# 卷积神经网络

1. 训练时间长
2. 数据量要求大
3. 计算资源要求高

# 卷积神经网络



怪我咯~

# 小数据集玩转深度学习技巧

1. 直接训练小数据集
2. 数据增强
3. 基于预训练网络进行特征提取
4. 基于预训练网络进行精细调参

# 直接训练小数据集



# 直接训练小数据集

2000训练数据

1000验证集

1000测试集

猫狗各50%

# 直接训练小数据集

最普通的卷积网络

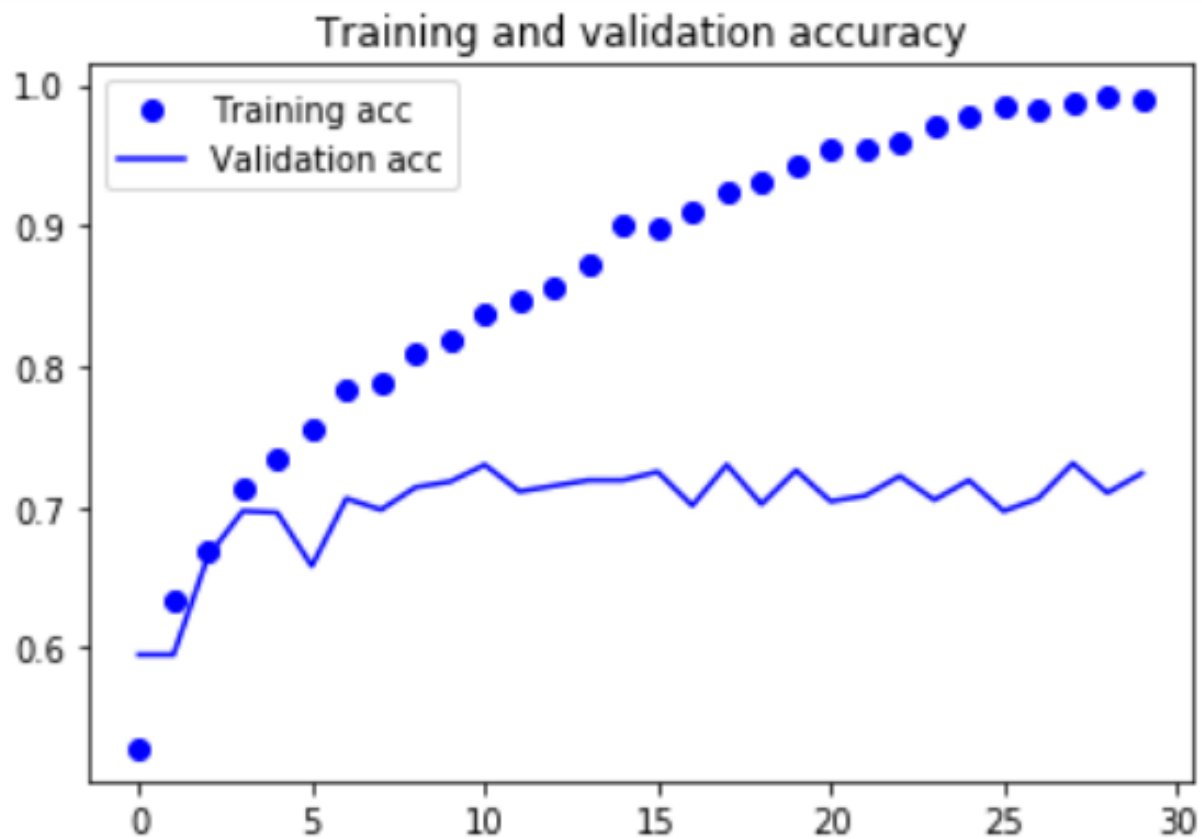
框架：Keras

1. 网络稍微深一点以增强网络容量
2. 减小feature maps大小使得全连接层不至于太大
3. 逐层增加卷积核数量

注意：这里不使用防止过拟合的技巧



# 直接训练小数据集



# 直接训练小数据集

具有一定准确率70%:

卷积网络能够学习局域特征,

而这些特征在感知上非常data-efficient

问题: 过拟合

分类器见到的猫狗太少, 新手上路

# 数据增强

data augmentation



# 数据增强



# 数据增强



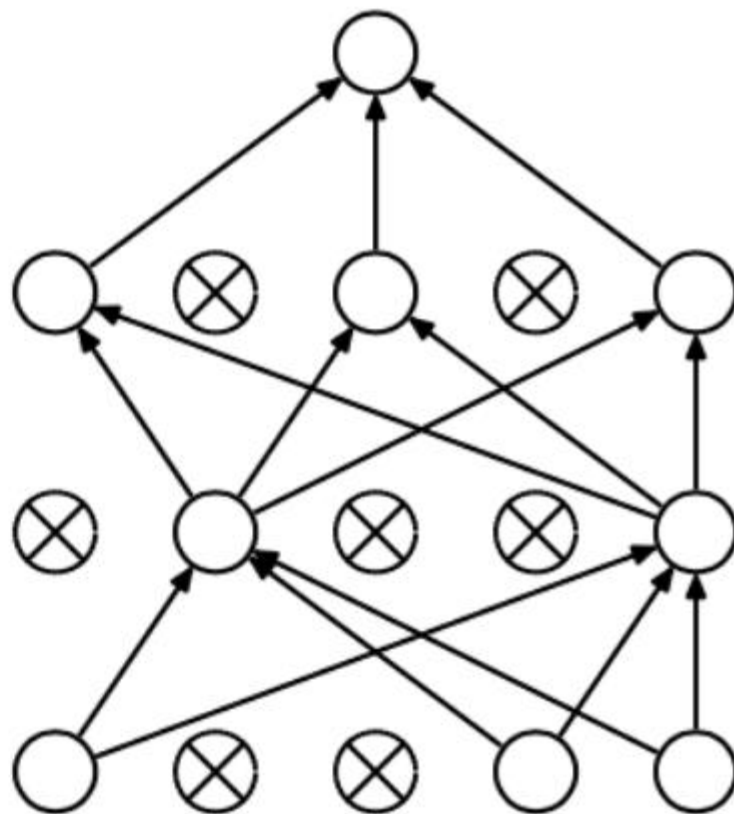
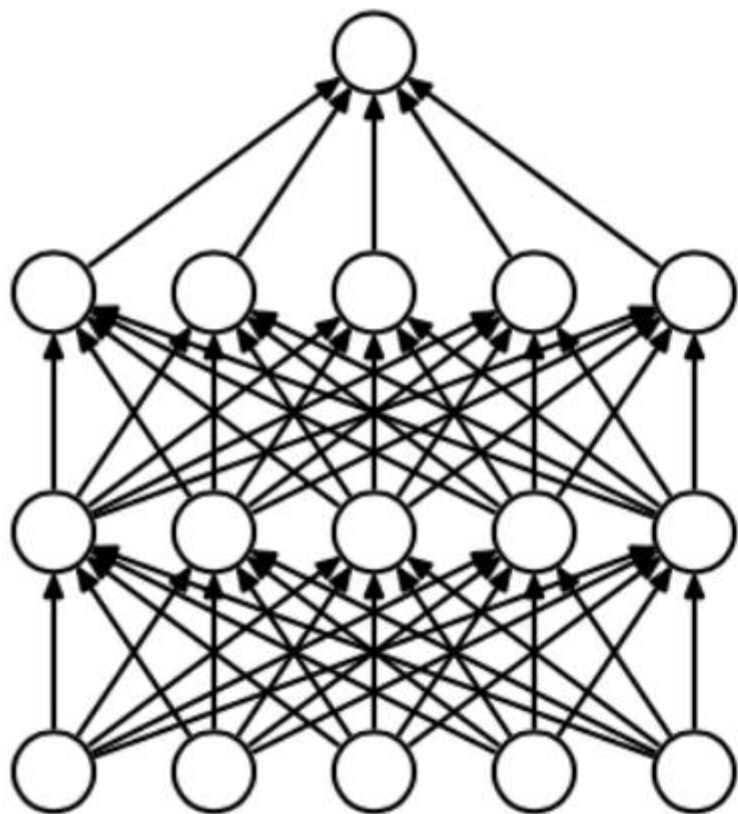
# 数据增强

数据增强只能混合现有信息  
很难增加额外信息

所以还要添加dropout

# 数据增强

## Dropout技术



# 数据增强

```
model.add(layers.Flatten())
```

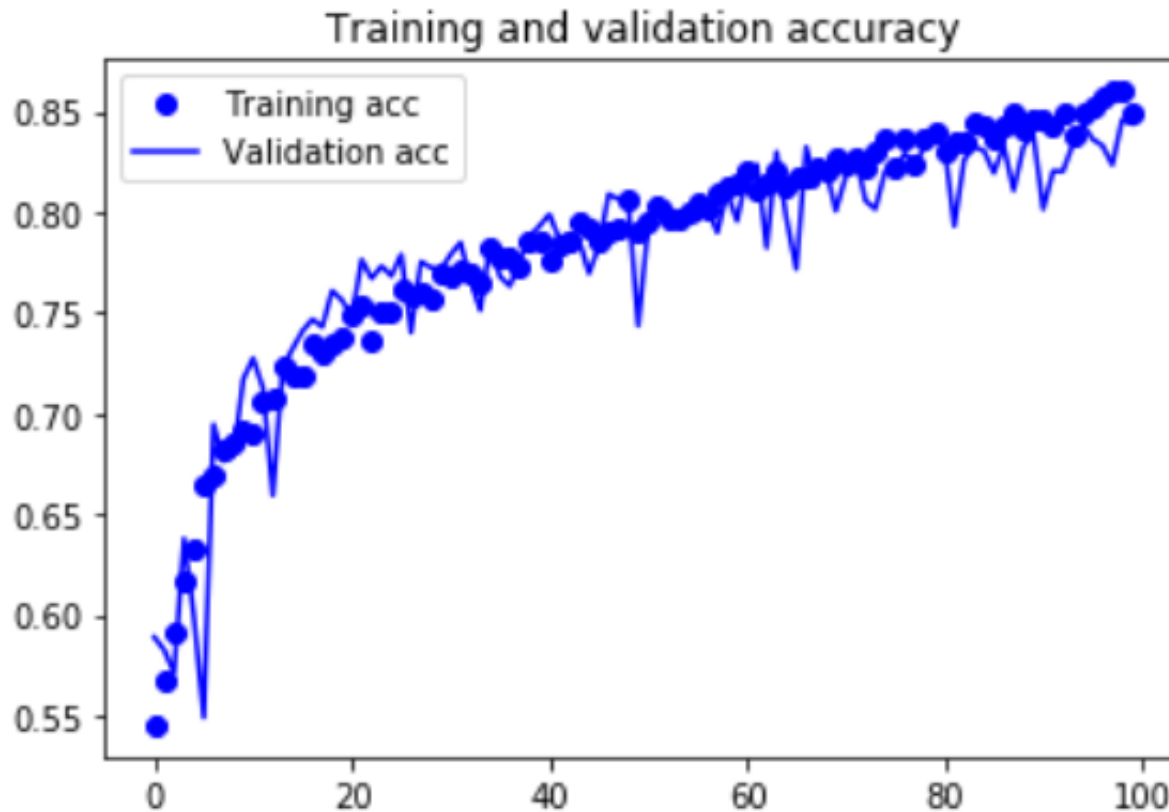
```
model.add(layers.Dropout(0.5))
```

```
model.add(layers.Dense(512, activation='relu'))
```



# 数据增强

data augmentation + dropout版本: **acc-83%**



# 基于预训练网络

办法：引入额外信息

# 基于预训练网络

充分利用pre-trained network

预训练网络：一个在大数据集上已经训练好的网络 (a saved network)

# 基于预训练网络

优势：

- 节省时间

原始数据集足够大且比较general:

- 预训练网络中学到的spatial feature

hierarchy具有很好的一般性

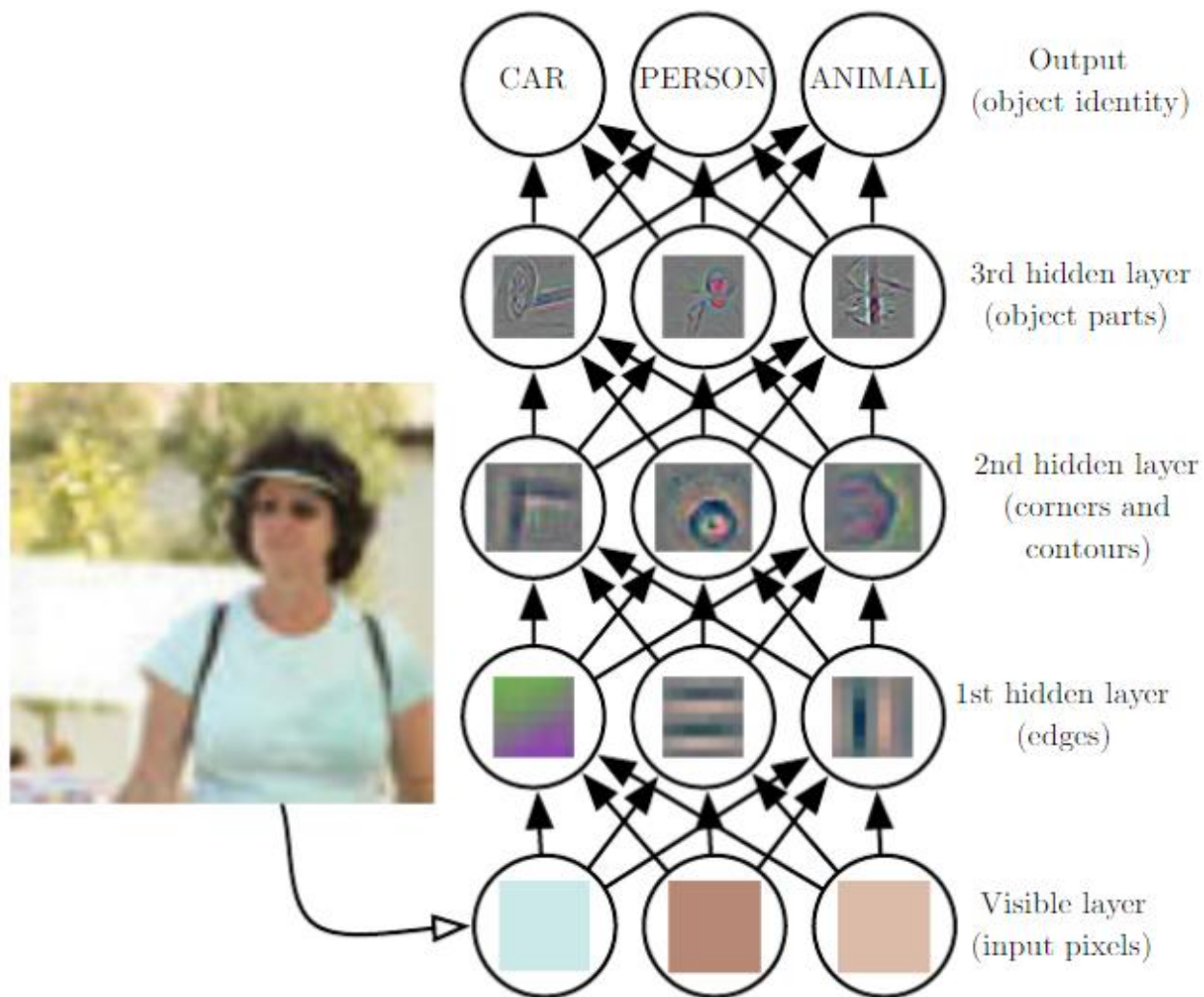
- 就算原始数据里没有的类别的识别能力

# 基于预训练网络

卷积网络识别能力：

1. 底层抽取局域、高度一般化的特征：边、角、纹理、颜色等
2. 高层抽取更抽象的特征：耳朵、眼睛等
3. 更高层抽取：识别对象

# 基于预训练网络



# 基于预训练网络

原始数据： ImageNet dataset

- 1500万张标注过的高清图片
- 22000类
- 10亿张， Amazon Turkey众包标注
- 儿童学习图片的启发

# 基于预训练网络

预训练网络： VGG16 architecture

ILSVRC2014

1. 相对简单的网络结构
2. 效果不错



# 基于预训练网络

基于预训练网络进行**特征提取**

从新的样本中抽取特征

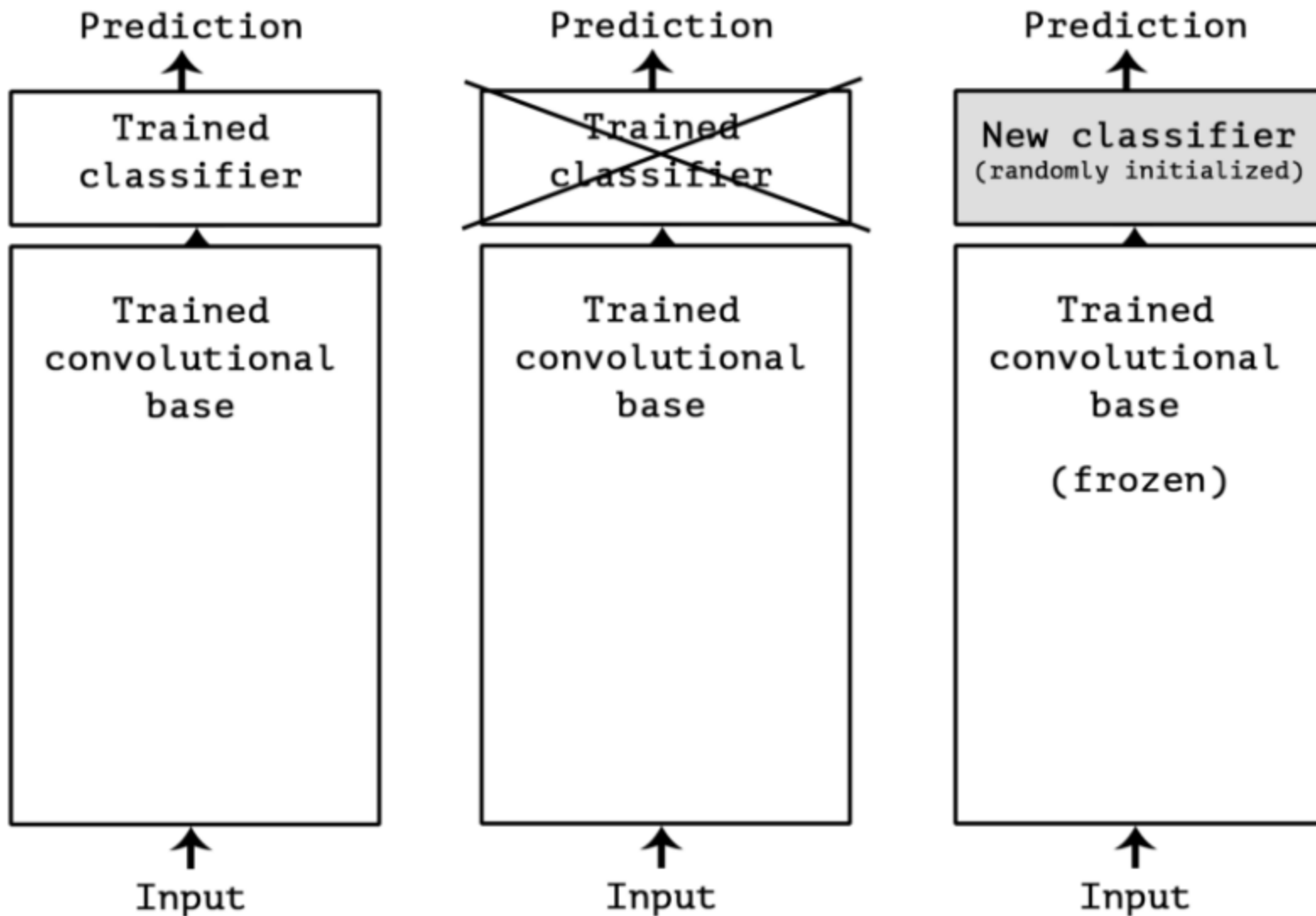
# 基于预训练网络

把整个网络看做两部分：

1. convolutional base
2. 全连接网络(分类器)

- 只训练新数据， convolutional base结构不变
- 只在新数据上抽特征

# 基于预训练网络



# 基于预训练网络

- convolutional base抽取更通用的特征
- 可重用性强
- 迁移学习

# 基于预训练网络

全连接网络分类器，在很多场景下不通用

特别是物体位置特征比较重要的时候，全连接网络抽取特征更不靠谱

注意：如果新数据集和原始数据很不一样，会更多和底层模型有共通处，不需要利用整个卷积层做特征提取，只用前几层即可

# 基于预训练网络

Caffe Model Zoo、Keras提供很多预训练模型：

Xception

InceptionV3

ResNet50

VGG16

VGG19

MobileNet

# 基于预训练网络

```
from keras.applications import VGG16
```

```
conv_base = VGG16(weights='imagenet',  
                    include_top=False,  
                    input_shape=(150, 150, 3))
```

# 基于预训练网络

利用conv base 抽取新数据的特征（特征提取器）

```
features_batch = conv_base.predict(inputs_batch)
```

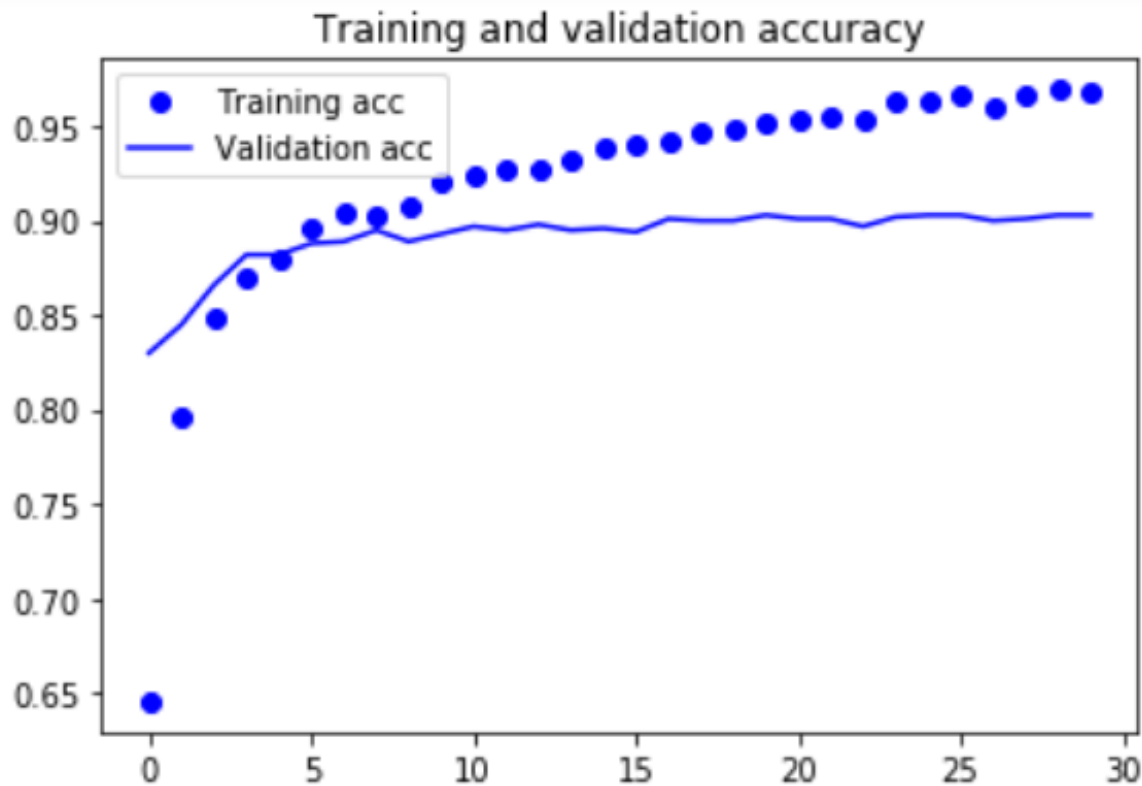


# 基于预训练网络

全连接网络训练（使用dropout防止过拟合）：

```
model = models.Sequential()  
model.add(layers.Dense(256, activation='relu',  
    input_dim=4 * 4 * 512))  
model.add(layers.Dropout(0.5))  
model.add(layers.Dense(1, activation='sigmoid'))
```

# 基于预训练网络



acc-80%，过拟合，dropout作用有限

# 基于预训练网络

数据增强:

conv base + 全连接层 = 一起训练数据增强

使用GPU

# 基于预训练网络

```
from keras import models
```

```
from keras import layers
```

```
model = models.Sequential()
```

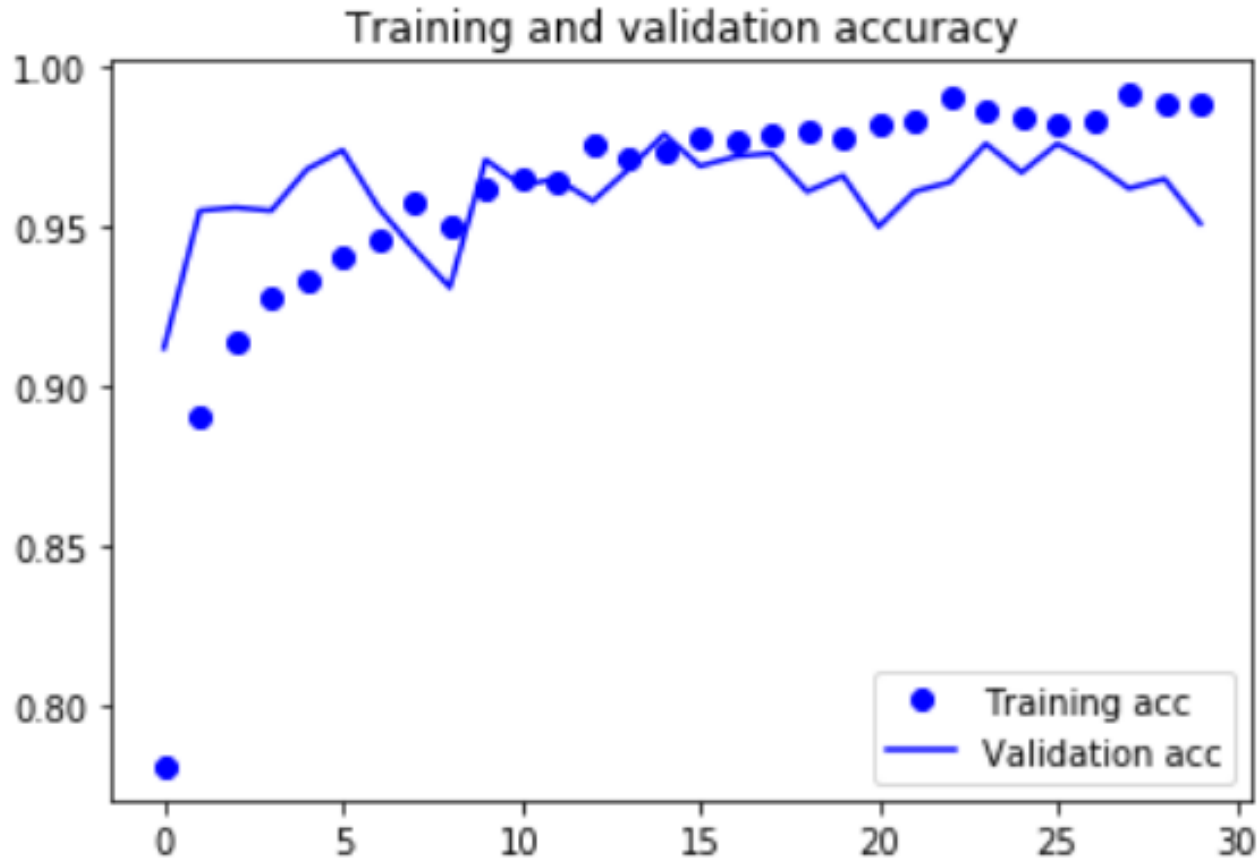
```
model.add(conv_base)
```

```
model.add(layers.Flatten())
```

```
model.add(layers.Dense(256, activation='relu'))
```

```
model.add(layers.Dense(1, activation='sigmoid'))
```

# 基于预训练网络



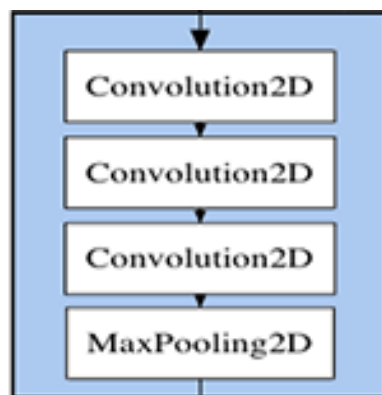
acc-95%

# 基于预训练网络

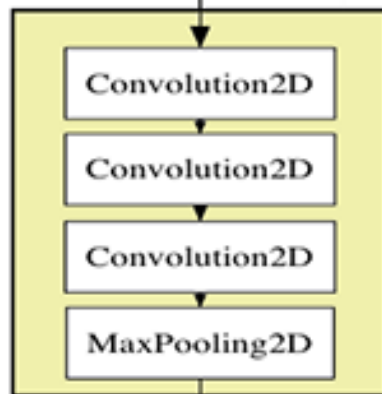
基于预训练网络进行精细调参: fine-tuning

冷冻部分conv base + 数据增强进行训练

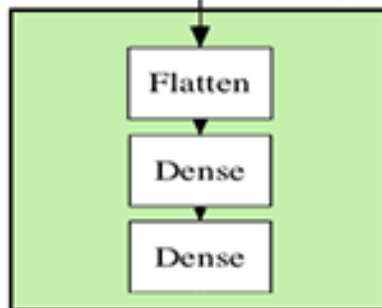
# 基于预训练神经网络



**Conv block 4:  
frozen**



**We fine-tune  
Conv block 5**



**We fine-tune  
our own  
fully-connected  
classifier**

# 基于预训练网络

1. 在卷积层上添加自定义网络层（比如全连接）
2. 冷冻base网络
3. 训练添加的部分（利用第二步的特征抽取能力）
4. 解冻部分base网络
5. 同时训练解冻的网络+自定义网络



# 基于预训练网络

```
conv_base.trainable = True
```

```
set_trainable = False
```

```
for layer in conv_base.layers:
```

```
    if layer.name == 'layer_name':
```

```
        set_trainable = True
```

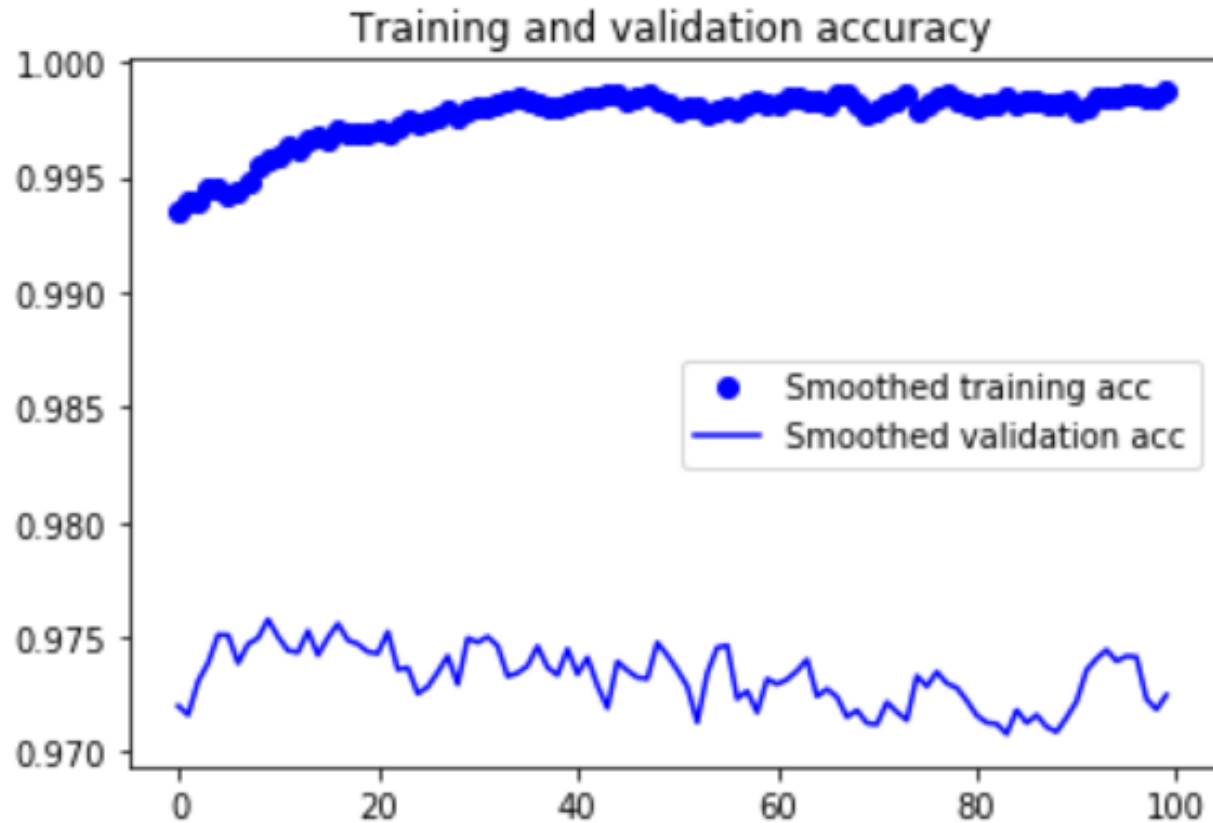
```
    if set_trainable:
```

```
        layer.trainable = True
```

```
    else:
```

```
        layer.trainable = False
```

# 基于预训练网络



acc-97%

# 基于预训练网络

ACC: 70% -- 83% -- 95% -- 97%

# 基于预训练网络



**Invasive Species Monitoring**

Identify images of invasive hydrangea

# 基于预训练网络

## Dog Breed Identification

Determine the breed of a dog in an image



# 基于预训练网络

## Plant Seedlings Classification

Determine the species of a seedling from an image

# 斗转星移，更进一步

斗转星移的思想：

借助具有一般特性抽取能力的模型（网络），  
为特定任务提供帮助

直接使用，快速学习

# 斗转星移，更进一步



拿走不谢~



# 斗转星移，更进一步

软件复用的思想

VS.

模型复用的思想

# 斗转星移，更进一步

## 学件-Learnware

Frontiers of Computer Science, 2016

DOI 10.1007/s11704-016-6906-3

**PERSPECTIVE**

**Learnware: on the future of machine learning**

Zhi-Hua ZHOU

National Key Laboratory for Novel Software Technology, Department of Computer Science  
& Technology, Nanjing University, Nanjing 210023, China

# 斗转星移，更进一步

A learnware is a well-performed **pre-trained machine learning model** with a ***specification*** which explains the purpose and/or specialty of the model.

# 斗转星移，更进一步

## Distilling the Knowledge in a Neural Network

Geoffrey Hinton, Oriol Vinyals, Jeff Dean

*(Submitted on 9 Mar 2015)*

A very simple way to improve the performance of almost any machine learning algorithm is to their predictions. Unfortunately, making predictions using a whole ensemble of models is cuml deployment to a large number of users, especially if the individual models are large neural net compress the knowledge in an ensemble into a single model which is much easier to deploy a technique. We achieve some surprising results on MNIST and we show that we can significant by distilling the knowledge in an ensemble of models into a single model. We also introduce a many specialist models which learn to distinguish fine-grained classes that the full models cor trained rapidly and in parallel.

# 斗转星移，更进一步

## Learnware Store



谢谢!