

# MATLAB

## 深度学习简介

# 什么是深度学习?

深度学习是机器学习的一个类型,该类型的模型直接从图像、文本或声音中学习执行分类任务。通常使用神经网络架构实现深度学习。“深度”一词是指网络中的层数 — 层数越多,网络越深。传统的神经网络只包含 2 层或 3 层,而深度网络可能有几百层。

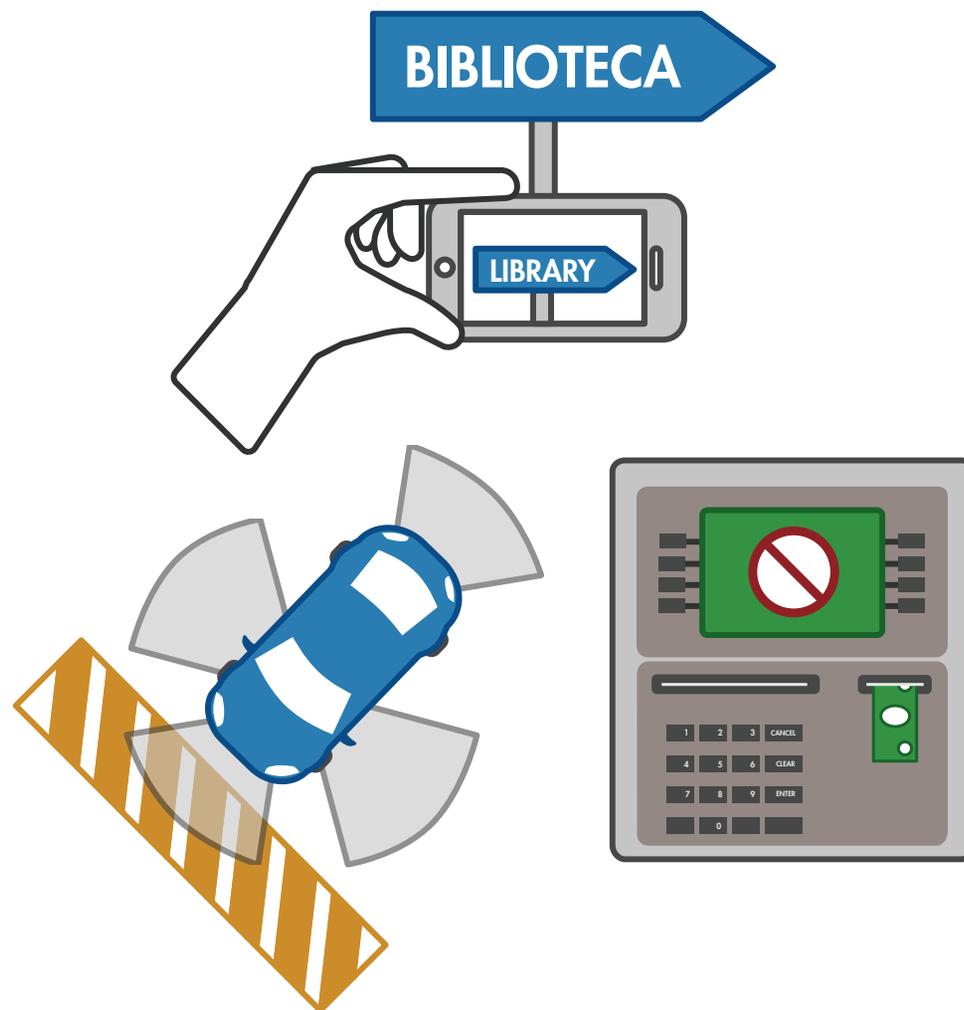


# 深度学习应用场景

下面只是深度学习发挥作用的几个例子：

- 无人驾驶汽车在接近人行横道线时减速。
- ATM 拒收假钞。
- 智能手机应用程序即时翻译国外路标。

深度学习特别适合鉴别应用场景，比如人脸辨识、文本翻译、语音识别以及高级驾驶辅助系统（包括车道分类和交通标志识别）。



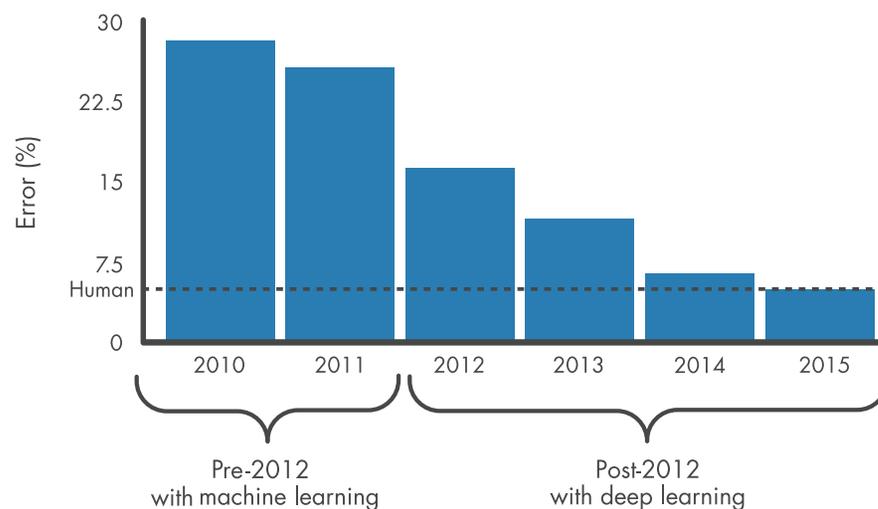
# 什么让深度学习如此先进？

简言之，精确。先进的工具和技术极大改进了深度学习算法，达到了很高的水平，在图像分类上能够超越人类，能打败世界最优秀的围棋选手，还能实现语音控制助理功能，如 Amazon Echo® 和 Google Home，可用来查找和下载您喜欢的新歌。

UCLA 研究人员建造了一种高级显微镜，能产生高维的数据集，用来训练深度学习网络，在组织检体中识别癌细胞。



### ILSVRC TOP-5 ERROR ON IMAGENET

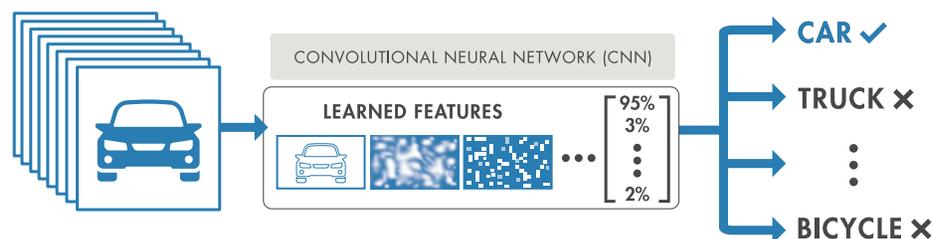


# 什么让深度学习如此先进？（续）

三个技术助推器让这种精确度成为可能：

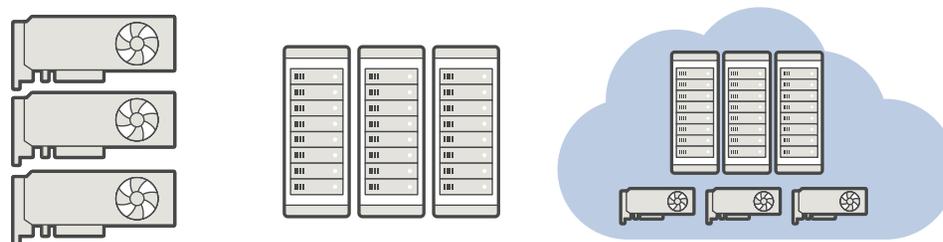
## 易于访问大规模带标签的数据集

ImageNet 和 PASCAL VoC 等数据集可以免费使用，对于许多不同类型的对象训练十分有用。



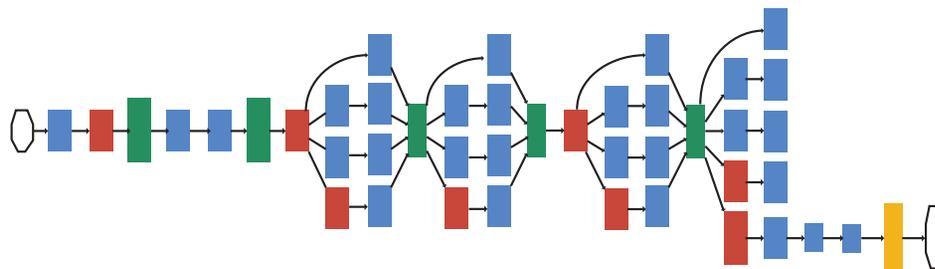
## 增大计算能力

高性能 GPU 加快了深度学习所需的海量数据的训练速度，训练时间从几星期减少到几小时。



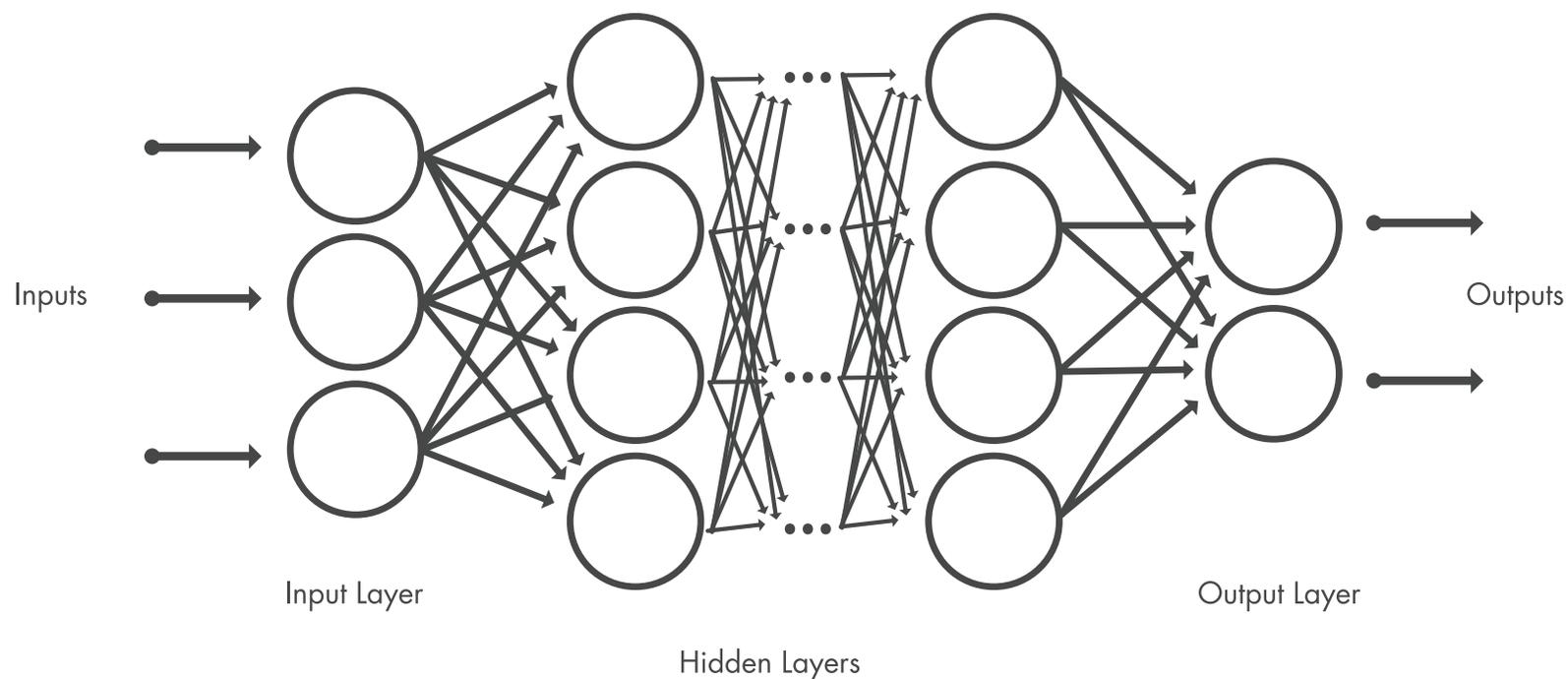
## 由专家构建的预先训练好的模型

可以重新训练 AlexNet 之类的模型，使用名为迁移学习的技术执行新识别任务。虽然使用了 130 万张高分辨率图像训练 AlexNet 来识别 1000 个不同的对象，但可以使用较小的数据集实现精确的迁移学习。



# 在深度神经网络内部

深度神经网络结合多个非线性处理层，并行使用简单元素操作，受到了生物神经系统的启发。它由一个输入层、多个隐藏层和一个输出层组成。各层通过节点或神经元相互连接，每个隐藏层使用前一层的输出作为其输入。



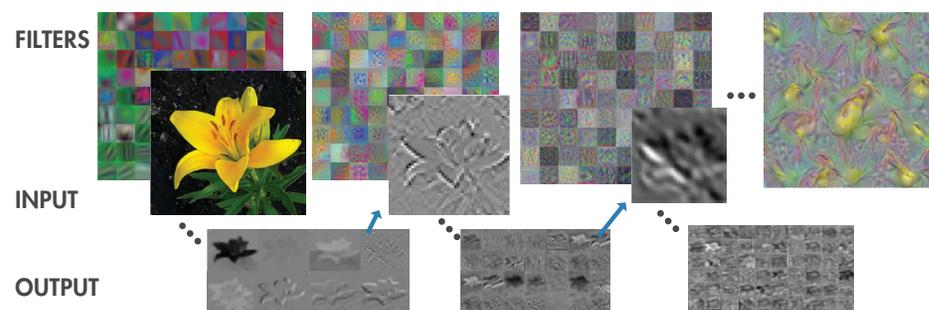
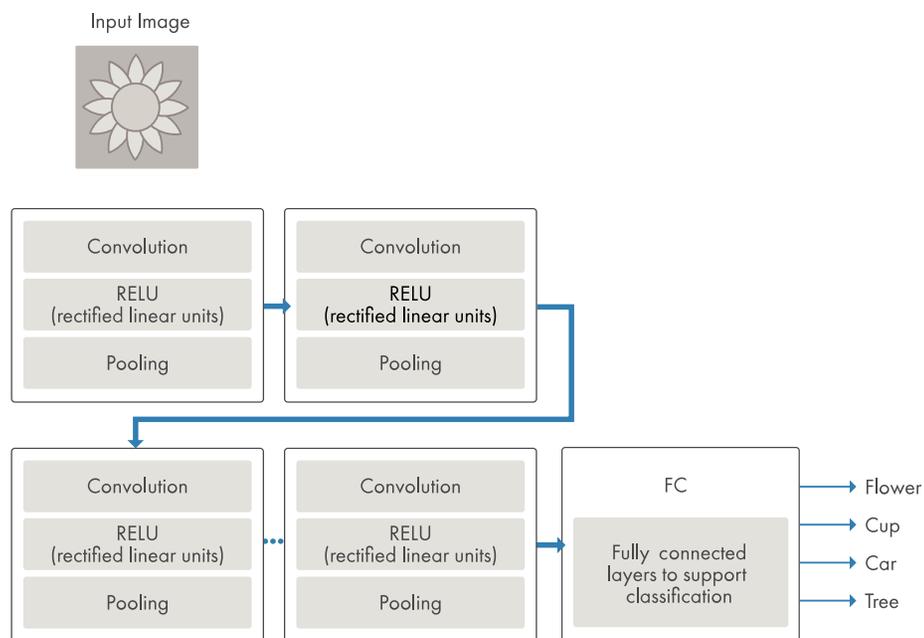
# 神经网络如何学习

比如说，我们有一组图像，每个图像包含四种不同类别对象的一种，我们想让深度学习网络自动识别每个图像中有哪个对象。我们给图像加标签，这样就有了网络的训练数据。

使用此训练数据，网络随后能开始理解对象的具体特征，并与相应的类别建立关联。

网络中的每一层从前面一层吸取数据，进行变换，然后往下传递。网络增加了复杂度和逐层学习的详细内容。

注意，网络直接从数据中学习 — 我们对学习到的具体特征没有影响。



# 关于卷积神经网络

卷积神经网络 (CNN 或 ConvNet) 是图像和视频深度学习的最流行算法之一。

像其他神经网络一样, CNN 由一个输入层、一个输出层和中间的多个隐藏层组成。

## 特征检测层

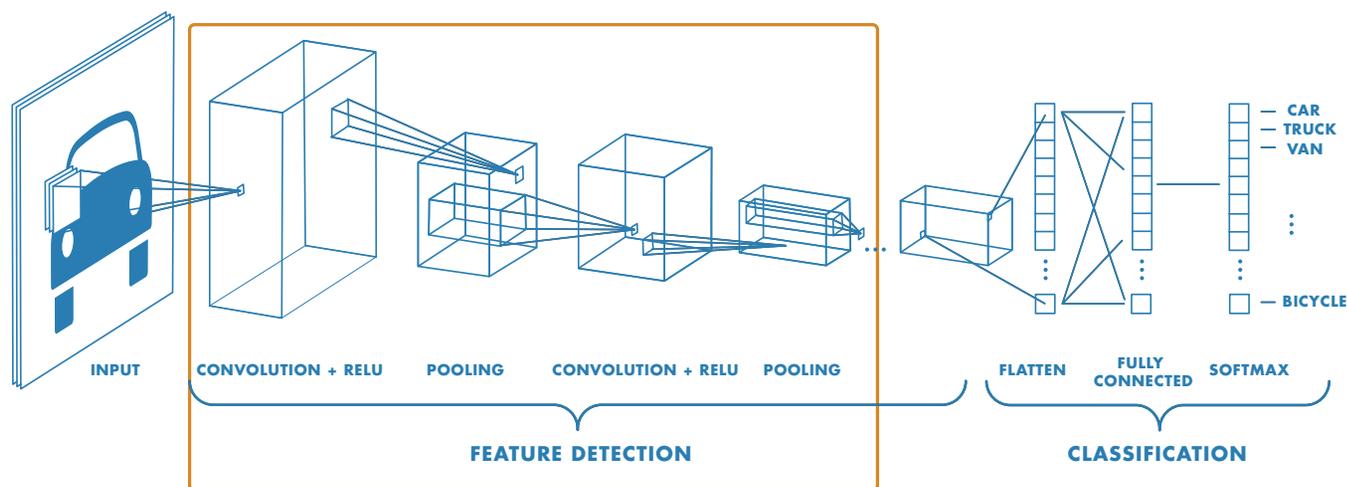
这些层对数据执行三种类型操作中的一种, 即卷积、池化或修正线性单元 (ReLU)。

**卷积**将输入图像放进一组卷积过滤器, 每个过滤器激活图像中的某些特征。

**池化**通过执行非线性下采样, 减少网络需要学习的参数个数, 从而简化输出。

**修正线性单元 (ReLU)** 通过将负值映射到零和保持正数值, 实现更快、更高效的训练。

这三种操作在几十层或几百层上反复进行, 每一层都学习检测不同的特征。



# 关于卷积神经网络 (续)

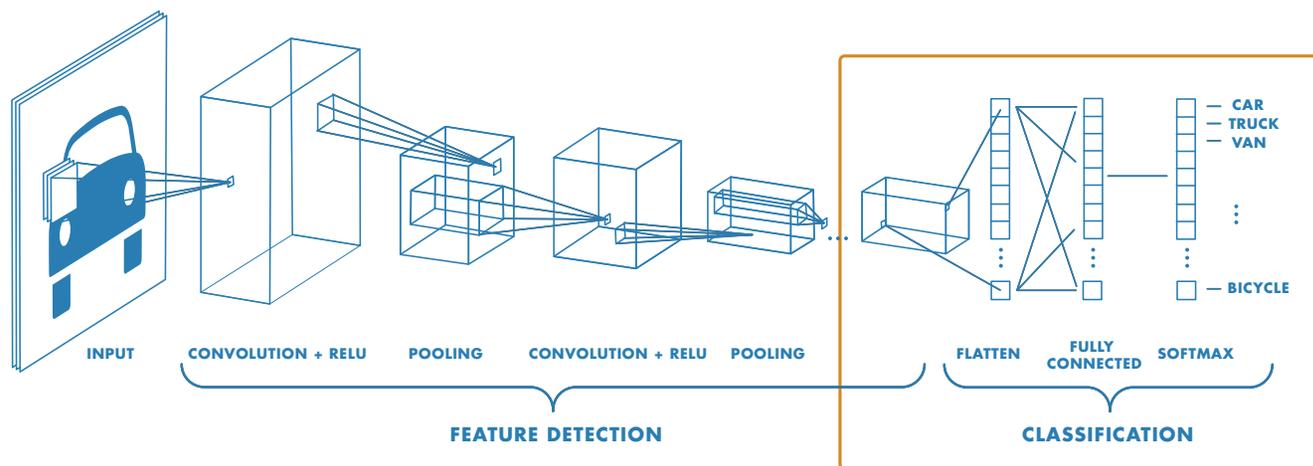
## 分类层

在特征检测之后, CNN 的架构转移到分类。

倒数第二层是**全连接层 (FC)**, 输出  $K$  维度的向量, 其中  $K$  是网络能够预测的类数量。此向量包含任何图像的对每个类进行分类的概率。

CNN 架构的最后一层使用 **softmax** 函数提供分类输出。

没有用于选择层数的确切公式。最好的方法是尝试一些层, 查看工作效果如何, 或者使用预先训练好的网络。

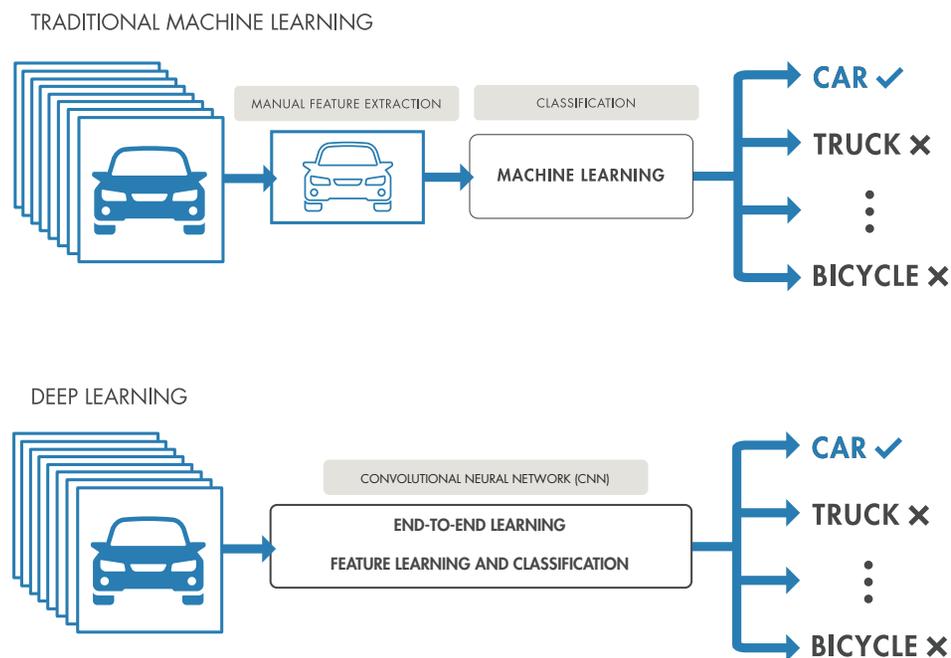


» 浏览 CNN 的架构

# 深度学习与机器学习之间有什么区别？

深度学习是机器学习的子类型。使用机器学习，您手动提取图像的相关特征。使用深度学习，您将原始图像直接馈送给深度神经网络，该网络自动学习特征。

为了获得最佳结果，深度学习通常需要成百上千乃至数百万张图像。而且属于计算密集型，需要高性能 GPU。

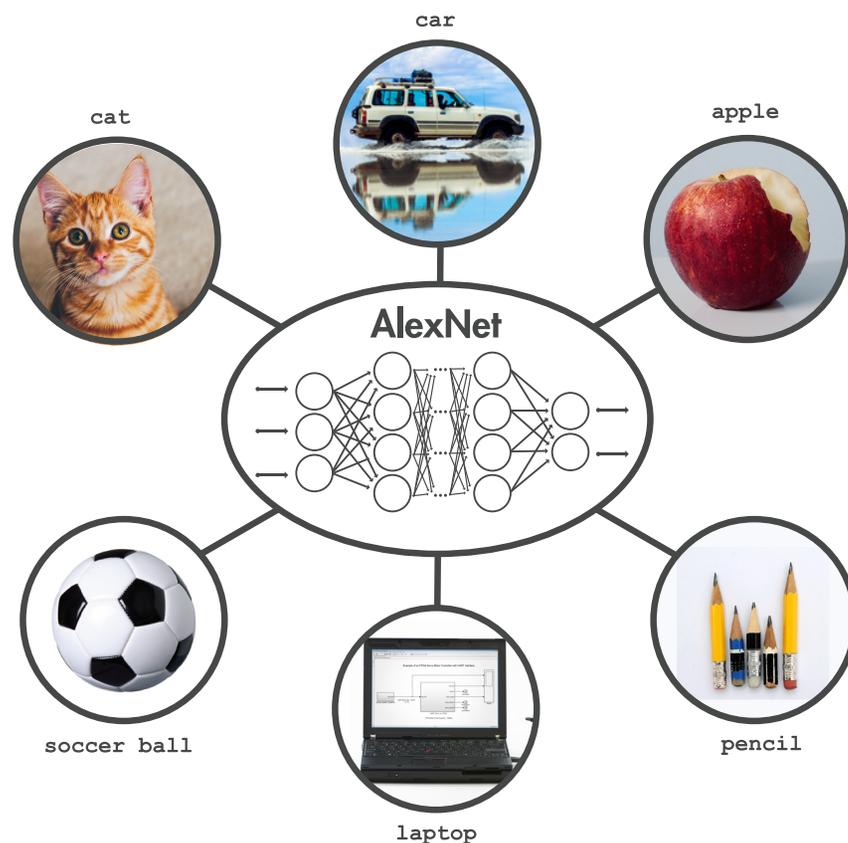


机器学习	深度学习
+ 用小数据集得到好结果	- 需要非常大的数据集
+ 快速训练模型	- 计算密集型
- 需要尝试不同的特征和分类器才能达到最佳结果	+ 自动学习特征和分类器
- 精度进入平台期	+ 精度无限制

# 深度学习快速入门

如果您刚接触深度学习，快速而轻松的入门方法是使用现有网络，比如 AlexNet，用一百多万张图像训练好的 CNN。AlexNet 最常用于图像分类。它可将图像划分为 1000 个不同的类别，包括键盘、鼠标、铅笔和其他办公设备，以及各个品种的狗、猫、马和其他动物。

AlexNet 于 2012 年首次发布，已成为研究团体中众所周知的模型。



» 了解关于预先训练网络的更多信息

# 使用 AlexNet 的一个示例

可以使用 AlexNet 对任何图像中的对象分类。在本例中，我们使用它对桌上安装的网络摄像头捕获的图像中的对象分类。除了 MATLAB®，我们还使用下列工具：

- Neural Network Toolbox™
- 在 [MATLAB 中使用网络摄像头](#) 的支持包
- 使用 [AlexNet](#) 的支持包

在加载 AlexNet 后，我们连接到网络摄像头并拍摄一张实时图像。

```
camera = webcam; % 连接到摄像头
nnet = AlexNet; % 加载神经网络
picture = camera.snapshot;
% 抓取图片
```

接下来，我们调整图像大小为 227x227 像素，即 AlexNet 所需的大小。

```
picture = imresize(picture,[227,227]);
% 调整图片大小
```

AlexNet 现在可对我们的图像分类。

```
label = classify(nnet, picture);
% 对图片分类
image(picture); % 显示图片
title(char(label)); % 显示标签
```



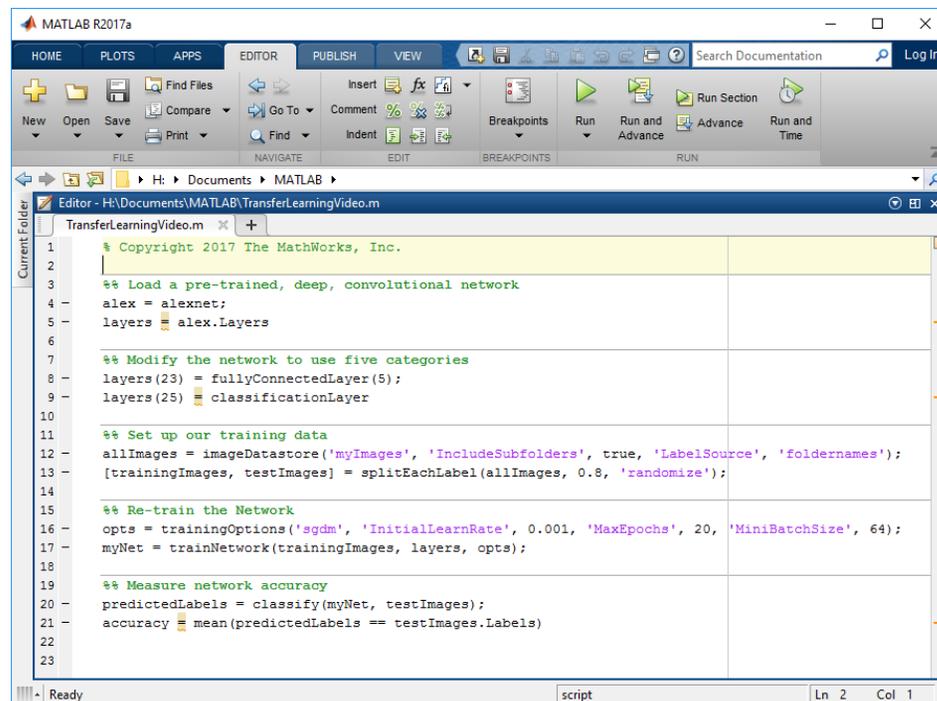
» 观看操作方法视频: [11 行 MATLAB 代码中实现深度学习](#)

# 重新训练现有网络

在上一个示例中，我们使用的是现成的网络。我们完全不做任何修改，因为 AlexNet 训练所用的图像类似于我们想要分类的图像。

要使用 AlexNet 对未在原有网络中训练的对象分类，我们可以通过迁移学习重新训练。迁移学习是将一个类型问题的知识应用于不同类型但相关的问题的学习方法。在本例中，我们只是剪去网络的最后 3 层，用我们自己的图像重新训练。

如果迁移学习不适合您的应用场景，您可能需要从头开始训练您自己的网络。这种方法产生最精确的结果，但一般需要成百上千张带标签的图像和大量的计算资源。



```
1 % Copyright 2017 The MathWorks, Inc.
2
3 %% Load a pre-trained, deep, convolutional network
4 alex = alexnet;
5 layers = alex.Layers
6
7 %% Modify the network to use five categories
8 layers(23) = fullyConnectedLayer(5);
9 layers(25) = classificationLayer
10
11 %% Set up our training data
12 allImages = imageDatastore('myImages', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
13 [trainingImages, testImages] = splitEachLabel(allImages, 0.8, 'randomize');
14
15 %% Re-train the Network
16 opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
17 myNet = trainNetwork(trainingImages, layers, opts);
18
19 %% Measure network accuracy
20 predictedLabels = classify(myNet, testImages);
21 accuracy = mean(predictedLabels == testImages.Labels)
22
23
```

» 迁移学习快速入门

# 深度学习的计算资源

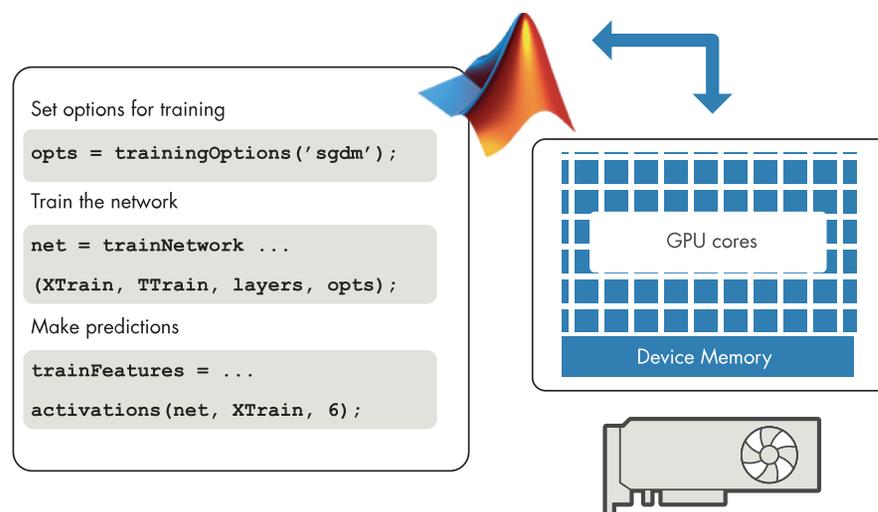
培训深度学习模型可能需要几小时、几天或几星期，具体取决于数据量大小以及可投入使用的处理能力。选择计算资源是您建立工作流程时的重要考虑因素。

当前，有三个计算选项：基于 CPU、基于 GPU 和基于云。

**基于 CPU 的计算**是最简单、最容易得到的选项。前面所述的示例在 CPU 上运算，但我们只对使用预先训练网络的简单示例推荐使用基于 CPU 的计算。

**使用 GPU** 可将网络训练时间从几天缩短到几小时。您可以在 MATLAB 中使用 GPU，无需任何额外编程。我们推荐 NVIDIA® 3.0 计算能力的 GPU。多个 GPU 能更加快处理速度。

**基于云的 GPU 计算**意味着您不必亲自购买和设置硬件。您为了使用本地 GPU 而编写的 MATLAB 代码只需进行一些设置变更，便可扩展为使用云资源。



» 通过 GPU 处理的大数据了解更多深度学习的信息

# 更多深度学习资源

[深度学习简介](#)

[通过 MATLAB 完成深度学习: 快速入门视频](#)

[使用迁移学习更快地开始深度学习](#)

[使用 AlexNet 的迁移学习](#)

[卷积神经网络简介](#)

[创建简单深度学习网络实现分类](#)

[通过 MATLAB 完成计算机视觉的深度学习](#)

[利用深度学习和光子时间拉伸进行癌症诊断](#)

