

AI基础：走进深度学习

原创：机器学习初学者 机器学习初学者 3天前

0. 导语

深度学习是机器学习的一个子集，自从AlexNet出现后，深度学习越来越热门。我最近在编写AI基础系列，第一部分已经完成并提供下载（[下载说明](#)）。现在开始更新深度学习部分。（黄海广）

目前已经发布：

AI 基础：简易数学入门

AI 基础：Python开发环境设置和小技巧

AI 基础：Python 简易入门

AI 基础：Numpy 简易入门

AI 基础：Pandas 简易入门

AI 基础：Scipy(科学计算库) 简易入门

AI基础：数据可视化简易入门（matplotlib和seaborn）

AI基础：机器学习库Scikit-learn的使用

AI基础：机器学习简易入门

AI基础：机器学习的损失函数

AI基础：特征工程-类别特征

AI基础：特征工程-数字特征处理

AI基础：特征工程-文本特征处理

AI基础：词嵌入基础和Word2Vec

AI基础：图解Transformer

AI基础：一文看懂BERT

AI基础：入门人工智能必看的论文

后续持续更新

1.深度学习入门的资料

深度学习的资料太多了，我只推荐吴恩达老师的深度学习课，刚开始学深度学习，不宜看太难的，以免打击信心。吴恩达老师的课，适合初学者，而且非常有代码和测试题。这个课程看懂了，以后看别的课程就不会太吃力了。

一、吴恩达深度学习教程中文笔记

github地址：

https://github.com/fengdu78/deeplearning_ai_books

包含笔记、视频等资源。

课程概述

<https://www.deeplearning.ai>

这些课程专为已有一定基础（基本的编程知识，熟悉**Python**、对机器学习有基本了解），想要尝试进入人工智能领域的计算机专业人士准备。介绍显示：“深度学习是科技业最热门的技能之一，本课程将帮你掌握深度学习。”

在这5堂课中，学生将可以学习到深度学习的基础，学会构建神经网络，并用在包括吴恩达本人在内的多位业界顶尖专家指导下创建自己的机器学习项目。**Deep Learning Specialization**对卷积神经网络 (**CNN**)、递归神经网络 (**RNN**)、长短期记忆 (**LSTM**) 等深度学习常用的网络结构、工具和知识都有涉及。

课程中也会有很多实操项目，帮助学生更好地应用自己学到的深度学习技术，解决真实世界问题。这些项目将涵盖医疗、自动驾驶、和自然语言处理等时髦领域，以及音乐生成等等。

Coursera上有一些特定方向和知识的资料，但一直没有比较全面、深入浅出的深度学习课程——《深度学习专业》的推出补上了这一空缺。

课程的语言是**Python**，使用的框架是**Google**开源的**TensorFlow**。最吸引人之处在于，课程导师就是吴恩达本人，两名助教均来自斯坦福计算机系。完成课程所需时间根据不同的学习进度，大约需要3-4个月左右。学生结课后，**Coursera**将授予他们**Deep Learning Specialization**结业证书。

“我们将帮助你掌握深度学习，理解如何应用深度学习，在人工智能业界开启你的职业生涯。”
吴恩达在课程页面中提到。

有同学提供了一个离线视频的下载：链接：

<https://pan.baidu.com/s/1ciq3qHo0lgoD3MLRwfeqnA> 密码：0kim

我还把笔记做了一个在线版本，可以在手机上阅读。（[在线阅读](#)）

深度学习课程的测试题，我翻译成了中文，可以做做看。（[深度学习测试题](#)）

2.主流深度学习框架和入门

TensorFlow、**Keras**和**Pytorch**是目前深度学习的主要框架，也是入门深度学习必须掌握的三大框架，但是官方文档相对内容较多，初学者往往无从下手。本人从github里搜到三个非常不错的学习资源，并对资源目录进行翻译，强烈建议初学者下载学习，这些资源包含了大量的代码示例（含数据集），个人认为，只要把以上资源运行一次，不懂的地方查官方文档，很快就能理解和运用这三大框架。

这部分资源整合后的github地址：

https://github.com/fengdu78/machine_learning_beginner/tree/master/deep-learning-with-tensorflow-keras-pytorch

一、TensorFlow

推荐资料

a.吴恩达深度学习笔记中的TensorFlow部分

这部分可以说是一个科普入门，学完后对TensorFlow会有一定的理解，接下来会容易点（第二门课第三周第3.11节，对应笔记p247-253，笔记可以在我的github下载：[github介绍](#)）

笔记截图：

希望这个让你对 **TensorFlow** 程序的大致结构有了了解，当你做编程练习，使用更多 **TensorFlow** 代码时，我这里用到的一些函数你会熟悉起来，这里有个地方要注意， w 是我们想要优化的参数，因此将它称为变量，注意我们需要做的就是定义一个损失函数，使用这些 `add` 和 `multiply` 之类的函数。**TensorFlow** 知道如何对 `add` 和 `multiply`，还有其它函数求导，这就是为什么你只需基本实现前向传播，它能弄明白如何做反向传播和梯度计算，因为它已经内置在 `add`，`multiply` 和平方函数中。

对了，要是觉得这种写法不好看的话，**TensorFlow** 其实还重载了一般的加减运算等等，因此你也可以把 `cost` 写成更好看的形式，把之前的 `cost` 标成注释，重新运行，得到了同样的结果。

```
In [5]: w=tf.Variable(0,dtype=tf.float32)
#cost=tf.add(tf.add(w**2,tf.multiply(-10.,w)),25)
cost=w**2-10*w+25
train=tf.train.GradientDescentOptimizer(0.01).minimize(cost)

init=tf.global_variables_initializer()
session=tf.Session()
session.run(init)
print(session.run(w))
```

0.0

```
In [6]: session.run(train)
print(session.run(w))
```

0.1

```
In [7]: for i in range(1000):
        session.run(train)
        print(session.run(w))
```

图：深度学习笔记截图

b.TensorFlow2.0样例 (github标星34000+)

TensorFlow推出2.0版本后，TF2.0相比于1.x版本默认使用Keras、Eager Execution、支持跨平台、简化了API等。这次更新使得TF2.0更加的接近PyTorch，一系列烦人的概念将一去不复返。推荐一位大神写的TF2.0的样例代码，推荐参考。

资源地址：

<https://github.com/aymericdamien/TensorFlow-Examples>

资源介绍：

本资源旨在通过示例轻松深入了解TensorFlow。 为了便于阅读，它包括notebook和带注释的源代码。

它适合想要找到关于TensorFlow的清晰简洁示例的初学者。除了传统的“原始”TensorFlow实现，您还可以找到最新的TensorFlow API实践（例如layers, estimator, dataset,）。

最后更新(08/17/2019)：添加新示例（TF2.0）。

配置环境：

python 3.6以上，TensorFlow 1.8+

资源目录：

0 - 先决条件

- 机器学习简介
- MNIST数据集简介

1 - 简介

- Hello World(包含notebook和py源代码)。非常简单的例子，学习如何使用TensorFlow打印“hello world”。
- 基本操作(包含notebook和py源代码)。一个涵盖TensorFlow基本操作的简单示例。
- TensorFlow Eager API基础知识(包含notebook和py源代码)。开始使用TensorFlow的Eager API。

2 - 基础模型

- 线性回归(包含notebook和py源代码)。使用TensorFlow实现线性回归。
- 线性回归（eager api）(包含notebook和py源代码)。使用TensorFlow的Eager API实现线性回归。
- Logistic回归(包含notebook和py源代码)。使用TensorFlow实现Logistic回归。
- Logistic回归（eager api）(包含notebook和py源代码)。使用TensorFlow的Eager API实现Logistic回归。
- 最近邻(包含notebook和py源代码)。使用TensorFlow实现最近邻算法。
- K-Means(包含notebook和py源代码)。使用TensorFlow构建K-Means分类器。
- 随机森林(包含notebook和py源代码)。使用TensorFlow构建随机森林分类器。
- Gradient Boosted Decision Tree（GBDT）(包含notebook和py源代码)。使用TensorFlow构建梯度提升决策树（GBDT）。
- Word2Vec（词嵌入）(包含notebook和py源代码)。使用TensorFlow从Wikipedia数据构建词嵌入模型（Word2Vec）。

3 - 神经网络

- 监督学习部分
 - 简单神经网络(包含notebook和py源代码)。构建一个简单的神经网络（如多层感知器）来对MNIST数字数据集进行分类。Raw TensorFlow实现。
 - 简单神经网络（tf.layers / estimator api）(包含notebook和py源代码)。使用TensorFlow'layers'和'estimator'API构建一个简单的神经网络（如：Multi-layer

Perceptron) 来对MNIST数字数据集进行分类。

- 简单神经网络 (Eager API) (包含notebook和py源代码)。使用TensorFlow Eager API构建一个简单的神经网络 (如多层感知器) 来对MNIST数字数据集进行分类。
- 卷积神经网络(包含notebook和py源代码)。构建卷积神经网络以对MNIST数字数据集进行分类。Raw TensorFlow实现。
- 卷积神经网络 (tf.layers / estimator api) (包含notebook和py源代码)。使用TensorFlow'layers'和'estimator'API构建卷积神经网络, 对MNIST数字数据集进行分类。
- 递归神经网络 (LSTM) (包含notebook和py源代码)。构建递归神经网络 (LSTM) 以对MNIST数字数据集进行分类。
- 双向LSTM(包含notebook和py源代码)。构建双向递归神经网络 (LSTM) 以对MNIST数字数据集进行分类。
- 动态LSTM(包含notebook和py源代码)。构建一个递归神经网络 (LSTM), 执行动态计算以对不同长度的序列进行分类。

■ 无监督

- 自动编码器(包含notebook和py源代码)。构建自动编码器以将图像编码为较低维度并重新构建它。
- 变分自动编码器 (包含notebook和py源代码)。构建变分自动编码器 (VAE), 对噪声进行编码和生成图像。
- GAN (Generative Adversarial Networks) (包含notebook和py源代码)。构建生成对抗网络 (GAN) 以从噪声生成图像。
- DCGAN (Deep Convolutional Generative Adversarial Networks) (包含notebook和py源代码)。构建深度卷积生成对抗网络 (DCGAN) 以从噪声生成图像。

4 - 工具

- 保存和还原模型(包含notebook和py源代码)。使用TensorFlow保存和还原模型。
- Tensorboard - 图形和损失可视化(包含notebook和py源代码)。使用Tensorboard可视化计算图并绘制损失。
- Tensorboard - 高级可视化(包含notebook和py源代码)。深入了解Tensorboard;可视化变量, 梯度等.....

5 - 数据管理

- 构建图像数据集(包含notebook和py源代码)。使用TensorFlow数据队列, 从图像文件夹或数据集文件构建您自己的图像数据集。
- TensorFlow数据集API(包含notebook和py源代码)。引入TensorFlow数据集API以优化输入数据管道。

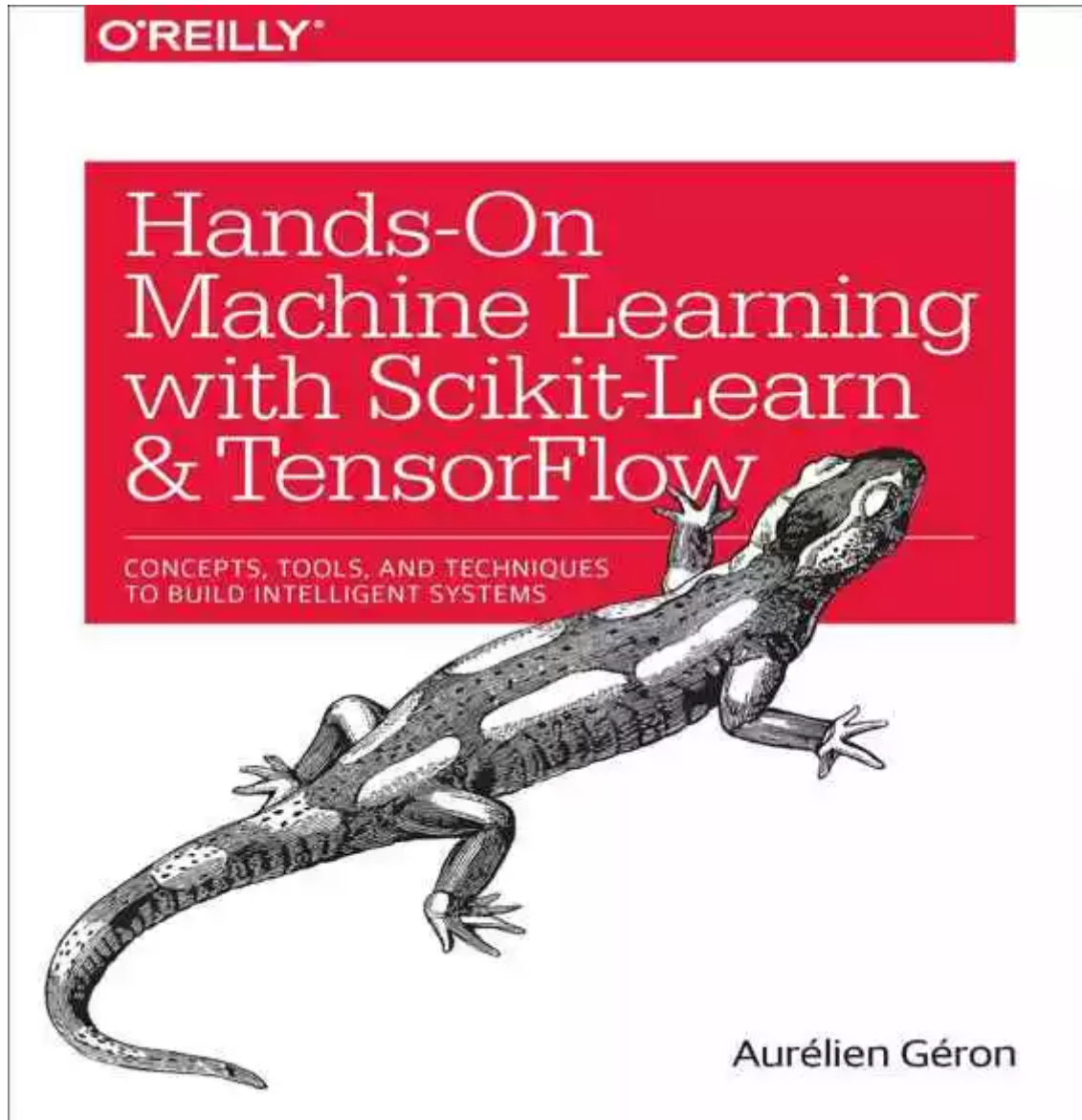
6 - 多GPU

- 多GPU的基本操作(包含notebook和py源代码)。在TensorFlow中引入多GPU的简单示例。
- 在多GPU上训练神经网络(包含notebook和py源代码)。一个清晰简单的TensorFlow实现, 用于在多个GPU上训练卷积神经网络。

数据集

- 一些示例需要MNIST数据集进行训练和测试。官方网站：
<http://yann.lecun.com/exdb/mnist/>

c. 《机器学习实战：基于Scikit-Learn和TensorFlow》



推荐一本机器学习和TensorFlow入门的好书：《机器学习实战：基于Scikit-Learn和TensorFlow》。

本书主要分为两部分，第一部分（第1~8章）涵盖机器学习的基础理论知识和基本算法，从线性回归到随机森林等，可以帮助你掌握Scikit-Learn的常用方法；第二部份（第9~16章）探讨深度学习和常用框架TensorFlow，手把手教你使用TensorFlow搭建和训练深度神经网络，以及卷积神经网络。

这本书受到广大机器学习爱好者的好评，可以说是机器学习入门宝书，豆瓣评分8.2。

这本书假定您有一些 Python 编程经验，并且比较熟悉 Python 的主要科学库，特别是 NumPy, Pandas 和 Matplotlib 。

- 本书作者公开了配套的源代码：

<https://github.com/ageron/handson-ml>

- 国内一个公益组织对原版英文书进行了翻译，制作成markdown文件（md文件推荐使用typora阅读），可以在github下载：

<https://github.com/apacheecn/hands-on-ml-zh>

针对国内下载速度慢，本站对两个资源进行打包，可以在百度云下载：

链接：

<https://pan.baidu.com/s/1jihUZrXblxhrVA5FBGU3RQ>

提取码：0xye

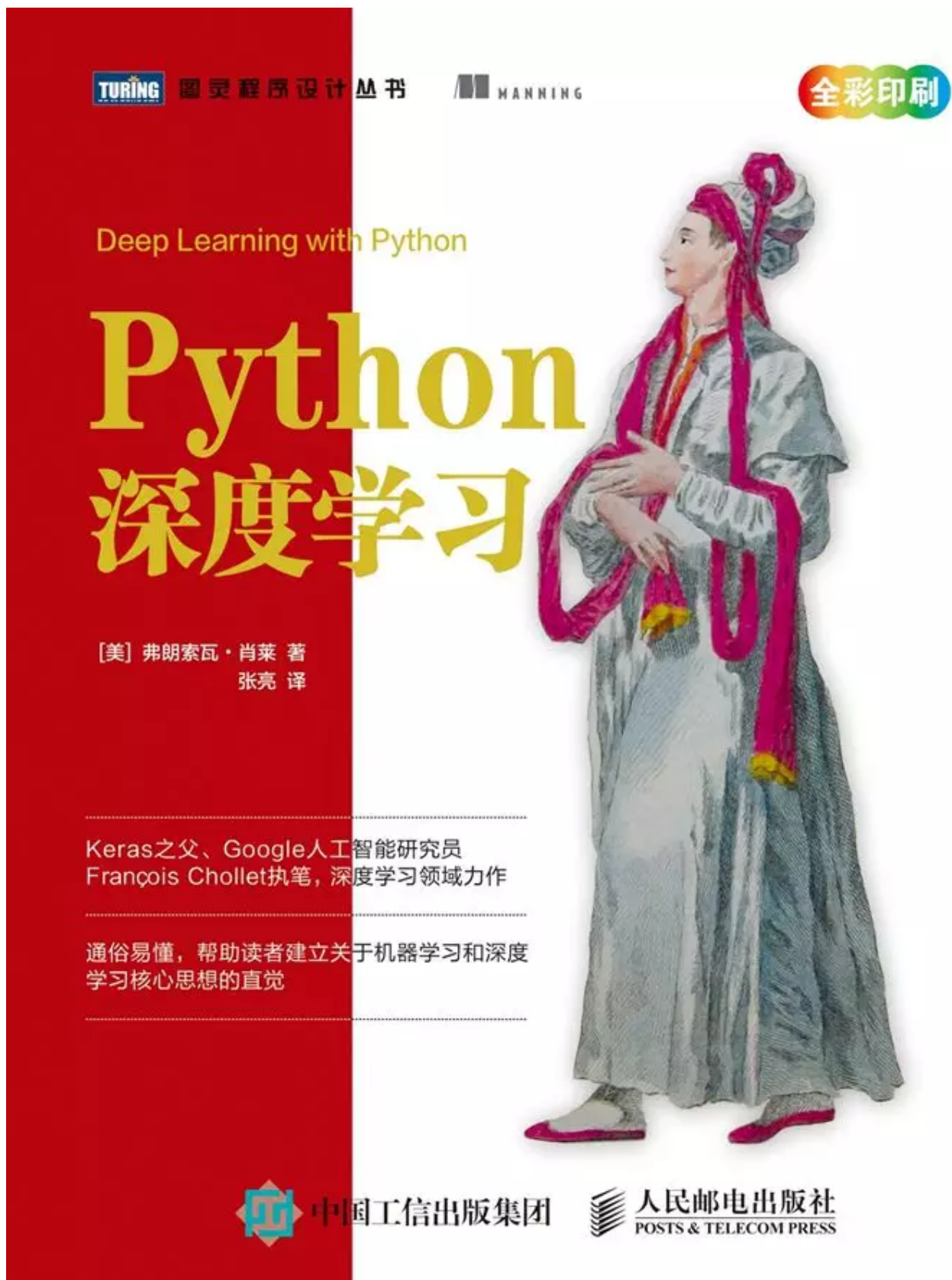
若被和谐请留言。

d. 《python深度学习》

TensorFlow团队的Josh Gordon推荐这本书，TF2.0基于Keras。如果你是一个深度学习新手，最好从这本书入手。（代码翻译注释：黄海广）

当然这本书里的代码需要改一下，但非常简单：

```
import keras -> from tensorflow import keras
```

《python深度学习》由Keras之父、现任Google人工智能研究员的弗朗索瓦·肖莱（François Chollet）执笔，详尽介绍了用Python和Keras进行深度学习的探索实践，包括计算机视觉、自然语言处理、产生式模型等应用。书中包含30多个代码示例，步骤讲解详细透彻。

作者在github公布了代码，代码几乎囊括了本书所有知识点。在学习完本书后，读者将具备搭建自己的深度学习环境、建立图像识别模型、生成图像和文字等能力。但是有一个小小的遗憾：代码的解释和注释是全英文的，即使英文水平较好的朋友看起来也很吃力。

本站认为，这本书和代码是初学者入门深度学习及Keras最好的工具。

我对全部代码做了**中文解释和注释**，并下载了代码所需要的一些数据集（尤其是“猫狗大战”数据集），并对其中一些图像进行了本地化，代码全部测试通过。（**请按照文件顺序运行，代码前后有部分关联**）。

以下代码包含了全书80%左右的知识点，代码目录：

- 2.1: A first look at a neural network（初识神经网络）
- 3.5: Classifying movie reviews（电影评论分类：二分类问题）
- 3.6: Classifying newswires（新闻分类：多分类问题）
- 3.7: Predicting house prices（预测房价：回归问题）
- 4.4: Underfitting and overfitting（过拟合与欠拟合）
- 5.1: Introduction to convnets（卷积神经网络简介）
- 5.2: Using convnets with small datasets（在小型数据集上从头开始训练一个卷积）
- 5.3: Using a pre-trained convnet（使用预训练的卷积神经网络）
- 5.4: Visualizing what convnets learn（卷积神经网络的可视化）
- 6.1: One-hot encoding of words or characters（单词和字符的 one-hot 编码）
- 6.1: Using word embeddings（使用词嵌入）
- 6.2: Understanding RNNs（理解循环神经网络）
- 6.3: Advanced usage of RNNs（循环神经网络的高级用法）
- 6.4: Sequence processing with convnets（用卷积神经网络处理序列）
- 8.1: Text generation with LSTM（使用 LSTM 生成文本）
- 8.2: Deep dream（DeepDream）
- 8.3: Neural style transfer（神经风格迁移）
- 8.4: Generating images with VAEs（用变分自编码器生成图像）
- 8.5: Introduction to GANs（生成式对抗网络简介）

中文注释与解释如图：

构建网络

在前一个 MNIST 示例中，我们构建了一个小型卷积神经网络，所以你应该已经熟悉这种网络。我们将复用相同的总体结构，即卷积神经网络由 Conv2D 层（使用 relu 激活）和 MaxPooling2D 层交替堆叠构成。

但由于这里要处理的是更大的图像和更复杂的问题，你需要相应地增大网络，即再增加一个 Conv2D+MaxPooling2D 的组合。这既可以增大网络容量，也可以进一步减小特征图的尺寸，使其在连接 Flatten 层时尺寸不会太大。本例中初始输入的尺寸为 150×150（有些随意的选择），所以最后在 Flatten 层之前的特征图大小为 7×7。

注意 网络中特征图的深度在逐渐增大（从 32 增大到 128），而特征图的尺寸在逐渐减小（从 148×148 减小到 7×7）。这几乎是所有卷积神经网络的模式。

你面对的是一个二分类问题，所以网络最后一层是使用 sigmoid 激活的单一单元（大小为1的 Dense 层）。这个单元将对某个类别的概率进行编码。

```
from keras import layers
from keras import models

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```


使用预训练的卷积神经网络

想要将深度学习应用于小型图像数据集，一种常用且非常高效的方法是使用预训练网络。预训练网络（pretrained network）是一个保存好的网络，之前已在大型数据集（通常是大规模图像分类任务）上训练好。如果这个原始数据集足够大且足够通用，那么预训练网络学到的特征的空间层次结构可以有效地作为视觉世界的通用模型，因此这些特征可用于各种不同的计算机视觉问题，即使这些新问题涉及的类别和原始任务完全不同。举个例子，你在 ImageNet 上训练了一个网络（其类别主要是动物和日常用品），然后将这个训练好的网络应用于某个不相干的任 务，比如在图像中识别家具。这种学到的特征在不同问题之间的可移植性，是深度学习与许多早期浅层学习方法相比的重要优势，它使得深度学习对小数据问题非常有效。

本例中，假设有一个在 ImageNet 数据集（140 万张标记图像，1000 个不同的类别）上训练好的大型卷积神经网络。ImageNet 中包含许多动物类别，其中包括不同种类的猫和狗，因此可以认为它在猫狗分类问题上也能有良好的表现。

我们将使用 VGG16 架构，它由 Karen Simonyan 和 Andrew Zisserman 在 2014 年开发。对于 ImageNet，它是一种简单而又广泛使用的卷积神经网络架构。虽然 VGG16 是一个比较旧的模型，性能远比不上当前最先进的模型，而且还比许多新模型更为复杂，但我之所以选择它，是因为它的架构与你已经熟悉的架构很相似，因此无须引入新概念就可以很好地理解。这可能是你第一次遇到这种奇怪的模型名称——VGG、ResNet、Inception、Inception-ResNet、Xception 等。你会习惯这些名称的，因为如果你一直用深度学习做计算机视觉的话，它们会频繁出现。

使用预训练网络有两种方法：特征提取（feature extraction）和微调模型（fine-tuning）。两种方法我们都会介绍。首先来看特征提取。

Feature extraction

Feature extraction consists of using the representations learned by a previous network to extract interesting features from new samples. These features are then run through a new classifier, which is trained from scratch.

As we saw previously, convnets used for image classification comprise two parts: they start with a series of pooling and convolution layers, and they end with a densely-connected classifier. The first part is called the "convolutional base" of the model. In the case of convnets, "feature extraction" will simply consist of taking the convolutional base of a previously-trained network, running the new data through it, and training a new classifier on top of the output.

特征提取

特征提取是使用之前网络学到的表示来从新样本中提取出有趣的特征。然后将这些特征输入一个新的分类器，从头开始训练。

如前所述，用于图像分类的卷积神经网络包含两部分：首先是一系列池化层和卷积层，最后是一个密集连接分类器。第一部分叫作模型的卷积基（convolutional base）。对于卷积神经网络而言，特征提取就是取出之前训练好的网络的卷积基，在上面运行新数据，然后在输出上面训练一个新的分类器（见下图）。

图：代码的中文注释与解释

作者的github：

<https://github.com/fchollet/deep-learning-with-python-notebooks>

中文注释代码：

https://github.com/fengdu78/machine_learning_beginner/tree/master/deep-learning-with-python-notebooks

二、Keras

资源地址：

<https://github.com/erhwenkuo/deep-learning-with-keras-notebooks>

资源介绍：

这个github的repository主要是ErhWen Kuo在学习Keras的一些记录及练习。希望在学习过程中发现到一些好的信息与示例也可以对想要学习使用Keras来解决问题的同学带来帮助。这些notebooks主要是使用Python 3.6与Keras 2.1.1版本跑在一台配置Nivida 1080Ti的Windows 10的机台所产生的结果，但有些部份会参杂一些Tensorflow与其它的函式库的介绍。

配置环境：

python 3.6以上，Keras 2.1.1

资源目录：

0.图象数据集/工具介绍

- 0.0: COCO API解说与简单示例
- 0.1:土炮自制扑克牌图象数据集
- 0.2:使用Pillow来进行图像处理

1.Keras API示例

- 1.0:使用图像增强来进行深度学习
- 1.1:如何使用Keras函数式API进行深度学习
- 1.2:从零开始构建VGG网络来学习Keras
- 1.3:使用预训练的模型来分类照片中的物体
- 1.4:使用图像增强来训练小数据集
- 1.5:使用预先训练的卷积网络模型
- 1.6:卷积网络模型学习到什么的可视化
- 1.7:构建自动编码器（Autoencoder）
- 1.8:序列到序列（Seq-to-Seq）学习介绍
- 1.9: One-hot编码工具程序介绍
- 1.10:循环神经网络（RNN）介绍
- 1.11: LSTM的返回序列和返回状态之间的区别
- 1.12:用LSTM来学习英文字母表顺序

2.图像分类（Image Classification）

- 2.0: Julia（Chars74K）字母图像分类
- 2.1:交通标志图像分类
- 2.2:辛普森卡通图像角色分类
- 2.3:时尚服饰图像分类
- 2.4:人脸关键点辨识
- 2.5: Captcha验证码分类
- 2.6: Mnist手写图像分类（MLP）
- 2.7: Mnist手写图像分类（CNN）

3.目标检测（Object Recognition）

- 3.0: YOLO目标检测算法概念与介绍
- 3.1: YOLOv2目标检测示例
- 3.2:浣熊（Raccoon）检测-YOLOv2模型训练与调整

- 3.3:浣熊（Racoon）检测-YOLOv2模型的使用
- 3.4:袋鼠（Kangaroo）检测-YOLOv2模型训练与调整
- 3.5:双手（Hands）检测-YOLOv2模型训练与调整
- 3.6:辛普森卡通图象角色（Simpson）检测-YOLOv2模型训练与调整
- 3.7: MS COCO图象检测-YOLOv2模型训练与调整

4.物体分割（Object Segmentation）

5.关键点检测（Keypoint Detection）

6.图象标题（Image Caption）

7.人脸检测识别（Face Detection/Recognition）

- 7.0:人脸检测- OpenCV（Haar特征分类器）
- 7.1:人脸检测- MTCNN（Multi-task Cascaded Convolutional Networks）
- 7.2:人脸识别-脸部检测、对齐&裁剪
- 7.3:人脸识别-人脸部特征提取&人脸分类器
- 7.4:人脸识别-转换、对齐、裁剪、特征提取与比对
- 7.5:脸部关键点检测（dlib）
- 7.6:头部姿态（Head pose）估计（dlib）

8.自然语言处理（Natural Language Processing）

- 8.0:词嵌入（word embeddings）介绍
- 8.1:使用结巴（jieba）进行中文分词
- 8.2: Word2vec词嵌入（word embeddings）的基本概念
- 8.3:使用结巴（jieba）进行歌词分析
- 8.4:使用gensim训练中文词向量（word2vec）

三、Pytorch

a.Pytorch入门

资源地址：

<https://github.com/yunjey/pytorch-tutorial>

资源介绍：

这个资源为深度学习研究人员提供了学习PyTorch的教程代码大多数模型都使用少于30行代码实现。在开始本教程之前，建议先看完Pytorch官方教程。

配置环境：

python 2.7或者3.5以上，pytorch 0.4

资源目录：

1.基础知识

- PyTorch基础知识
- 线性回归
- Logistic回归
- 前馈神经网络

2.中级

- 卷积神经网络
- 深度残差网络
- 递归神经网络
- 双向递归神经网络
- 语言模型（RNN-LM）

3.高级

- 生成性对抗网络
- 变分自动编码器
- 神经风格转移
- 图像字幕（CNN-RNN）

4.工具

- PyTorch中的TensorBoard

总结

b. 60分钟入门深度学习工具-PyTorch

作者：Soumith Chintala

翻译：黄海广

原文翻译自：https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

github地址：

https://github.com/fengdu78/Data-Science-Notes/tree/master/8.deep-learning/PyTorch_beginner

本教程的目标：

- 在高层次上理解PyTorch的张量(Tensor)库和神经网络
- 训练一个小型神经网络对图像进行分类
- 本教程假设您对numpy有基本的了解

注意：务必确认您已经安装了 `torch` 和 `torchvision` 两个包。

目录

1.Pytorch是什么？

2.AUTOGRAD

3.神经网络

4.训练一个分类器

5.数据并行

结语

这里是AI基础的第二部分，后续还会更新，请大家关注公众号的更新。（黄海广）



往期精彩回顾



- 那些年做的学术公益-你不是一个人在战斗
- 适合初学者入门人工智能的路线及资料下载
- 机器学习在线手册
- 深度学习在线手册

备注：加入本站微信群或者qq群，请回复“加群”

加入知识星球（4500+用户，ID：92416895），请回复“知识星球”

喜欢文章，点个在看

