

# 贪心科技 CV 课第一章学习笔记

作者:hangtingter 组别: 第一组

## 摘要

本章主要内容为机器学习以及深度学习的简介，包括机器学习和人工神经网络简介；深度学习的发展历程；深度学习在计算机视觉中的应用；以及深度学习模型开发环境的配置。

**关键词：**机器学习 深度学习 简介 环境搭建

## 一、引言

机器学习是什么？深度学习又是什么？我们为什么要用深度学习？本章从现实生活出发，介绍机器学习以及深度学习的基本概念，深度学习的发展历程，最后介绍本门课程需要使用到的开发环境配置方法。

## 二、机器学习、深度学习简介

**机器学习：**通过算法使得机器能从大量历史数据中学习规律从而对新的样本做预测。如语音识别，构建一个函数，输入一段语音信号，输出一段文字；如图片识别，构建一个函数，输入一张图片，输出该图片的识别结果。其他类型如图 1 所示。

机器学习 ≈ 构建一个函数



•语音识别

$$f(\text{[声波图]}) = \text{"人工智能"}$$

•图像识别

$$f(\text{[猫的照片]}) = \text{"猫"}$$

•围棋

$$f(\text{[棋盘]}) = \text{"5-5" (落子位置)}$$

•对话系统

$$f(\text{"今天天气如何"}) = \text{"小雨转晴, 23度, 微风"}$$

图 1 机器学习简介

**深度学习**: 深度学习是机器学习的子集，是基于神经网络的一种机器学习算法，主要有以下优势：

- 1) 深度神经网络在多个计算机视觉任务的比赛中获得了很好的成绩；
- 2) 深度神经网络模型可以用于不同的领域(语言识别, 下围棋, 玩游戏, 预测比赛结果, 计算机视觉等等)；
- 3) 在特定数据集特定任务上面超过普通人类的能力；
- 4) 活跃的社区, 大量的资源；
- 5) 计算机硬件软件技术的进步提供了技术能力支撑。

**深度学习的发展历程**: 深度学习模型起源于古老的感知机以及神经网络模型, 发展历程如图 2 所示。

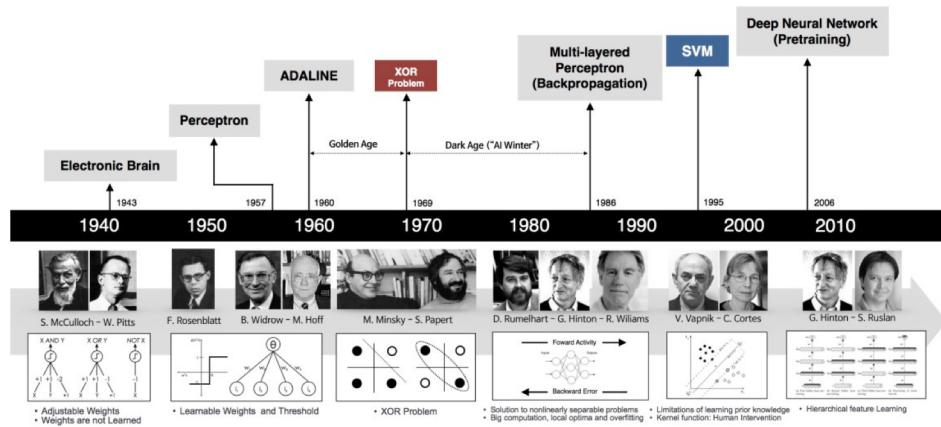


图 2 深度学习发展历程

- 1) 感知机模型, 对输入加权求和, 激活函数是阶跃函数 (Step Function), 但是属于线性函数, 无法解决异或问题

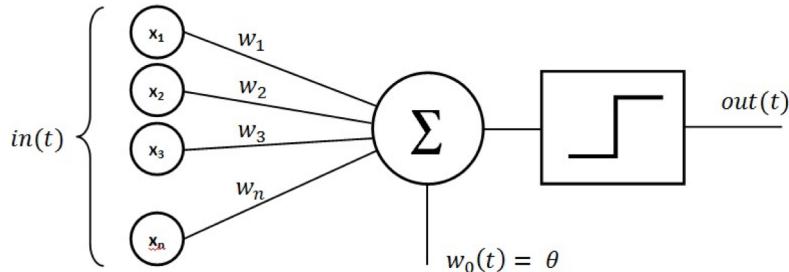


图 3 感知机

- 2) 1960 年提出人工神经网络, 多个感知器, 多层结合, 可以拟合任意函数, 可以解决异或问题。

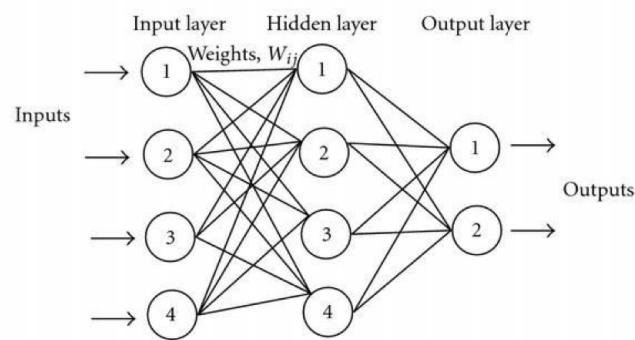


图 4 人工神经网络

- 3) 1982 年提出 BP 算法 (误差向后传递算法, Backpropagation), 误差向后传递, 可以使用梯度下降法等优化, 可以自动训练多层的网络。

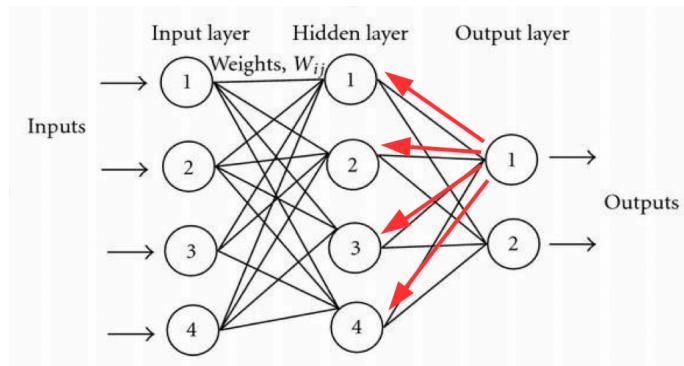


图 5 误差向后传递算法

- 4) 1989 年提出卷积与池化的概念, 相较于普通全连接层, 参数更少, 更容易训练, 在当时已经应用于计算机视觉问题——手写数字识别问题中。

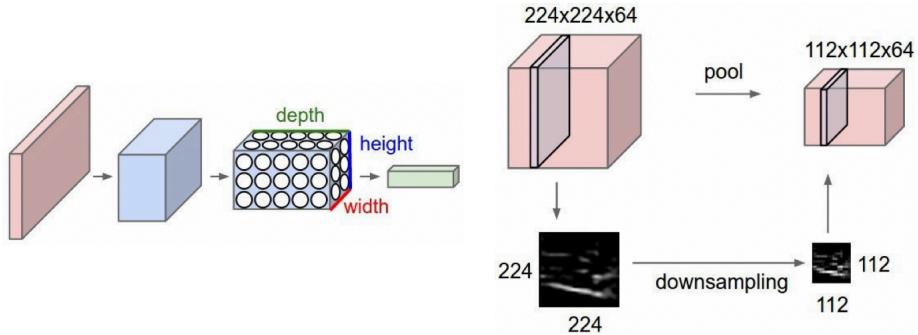


图 6 卷积层与池化层

- 5) 2011 年深度学习得到了很好的发展, 更好的初始化, 新的激活函数 ReLU, 更多的训练数据 (千万级别的 ImageNet), 更好的计算机硬件, 并带来了一系列的进步。

总之，深度学习是一种综合技术。包括卷积 Conv-、池化 Pool- 和全连接层 Fully-Connected Layers ReLU 激活函数；更深的网络层次，更多的参数；新的网络层 (DWConv, SPConv, Group Convolution) 和结构 (Skip Connection, Dense Connection)；新的防止过拟合和技术 (DropOut, Image Augmentation)；更大的训练数据和计算基础设施 (千万级别的数据, GPU 分布式计算) 等等。

### 三、几个典型的深度神经网络结构

1) AlexNet (2012)，共有 62378344 个权值 (250MB)，24 层。

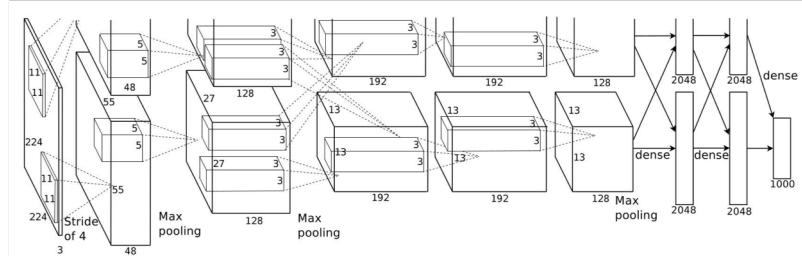


图 7 AlexNet 神经网络结构

2) VGGNet (2013)，102908520 个权值 (412MB)，23 层。

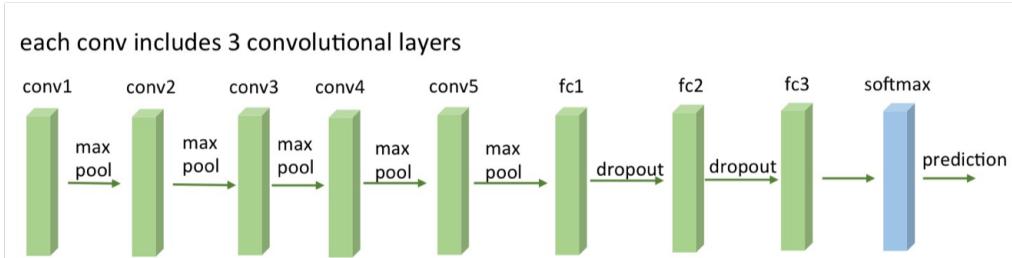


图 8 VGGNet 神经网络结构

3) GoogLeNet(2014), 6998552 个权值 (28MB)，143 层，首次提出了 Inception 的概念，图 10 给出了 Inception 的基本结构，主要贡献有两个：一是使用  $1 \times 1$  的卷积来进行升降维；二是在多个尺寸上同时进行卷积再聚合，即提取了更多的信息。

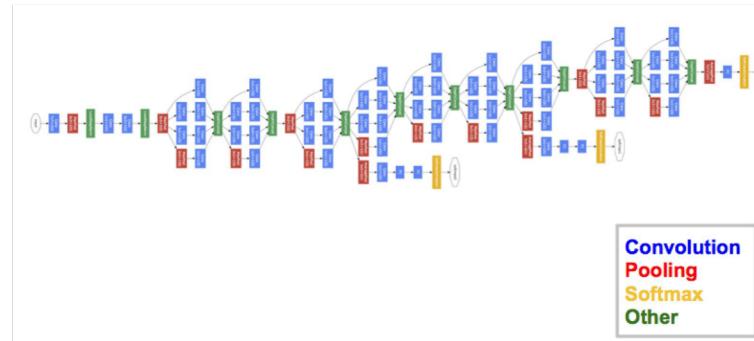


图 9 GoogLeNet 神经网络结构

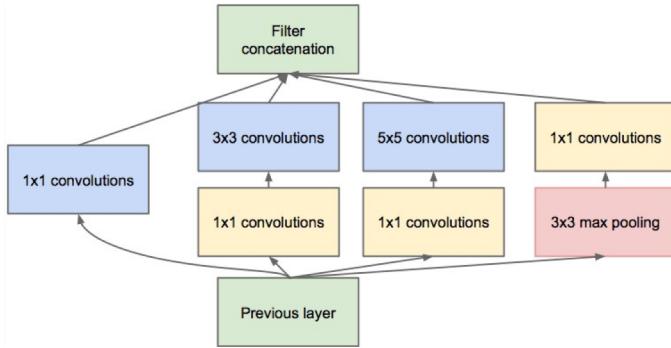


图 10 Inception Module

思考：使用  $1 \times 1$  卷积的作用是什么？

$1 \times 1$  卷积神经网络的主要作用是为了在整合不同通道的信息，结合不同尺度信息进行升降维，如图 11 所示，其数学表示如下所示。

转载自<https://blog.csdn.net/renhaofan/article/details/82721868>

$$\text{rawpicture} = R\text{layer} + G\text{layer} + B\text{layer}$$

$$\text{purplelayer} = \alpha \times \text{rawpicture}$$

$$\text{yellowlayer} = \beta \times \text{rawpicture}$$

$$\text{newpicture} = \text{purplelayer} + \text{yellowlayer} = (\alpha + \beta)(R + G + B)$$

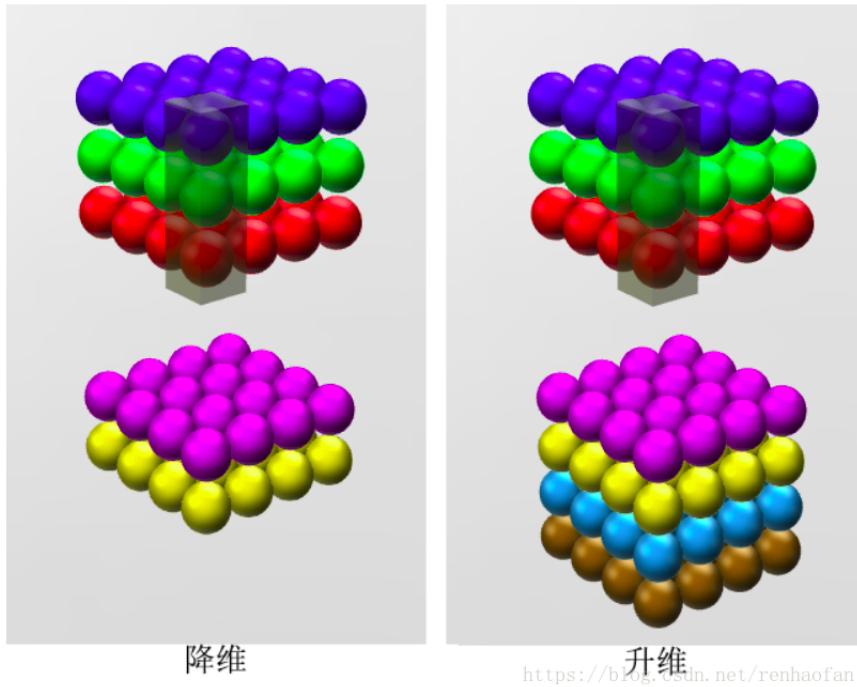


图 11  $1 \times 1$  卷积层的作用

4) ResNet (2015)，创造了 Residual 残差学习,152 层

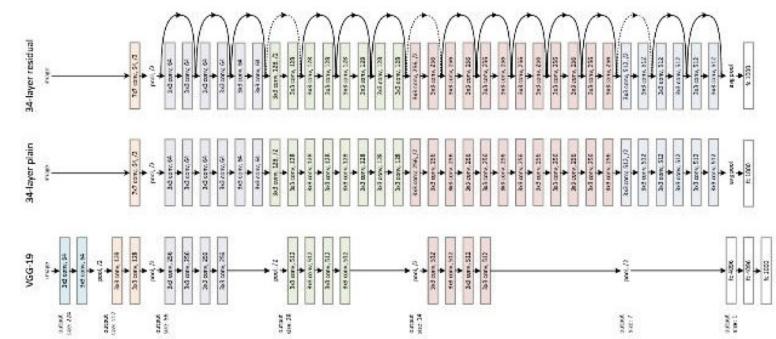


图 12 ResNet 神经网络模型

#### 四、深度学习在计算机视觉中的应用

深度学习在计算机视觉中主要有四个方面：图片分类（单目标），目标定位（单目标），目标检测（多目标），语义分割（多目标）。

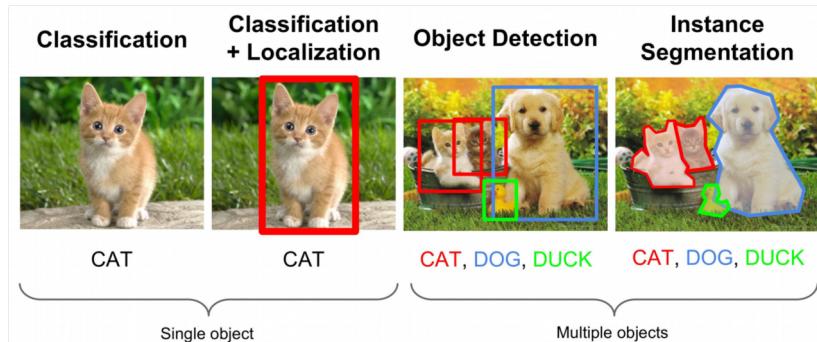


图 13 深度学习在计算机视觉中的四大应用

除此之外，还有压缩，Auto-encoders, Self-organizing maps，生成图像描述 Image Captioning，结合递归神经网络 NLP，图像风格化转移 Image Stylization，基于内容的图像 Image Retrieval 等方面的应用。

1) 分类识别：每副图包含一种类别的物体，最后一层是 Softmax 层



图 14 典型图像分类识别任务

2) 目标定位： Bounding box Regression 回归 + Class Recognition 分类

Overfeat:

<https://arxiv.org/abs/1312.6229>

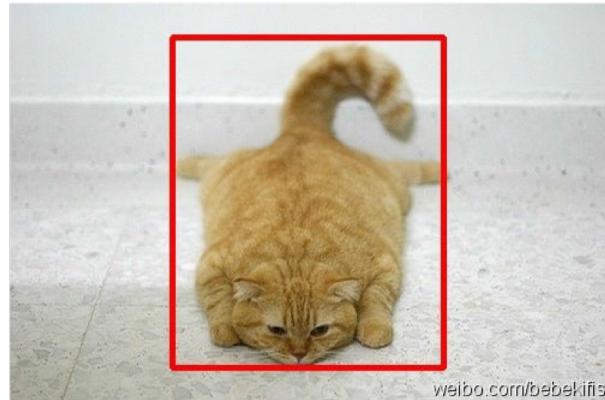


图 15 典型图像目标定位任务

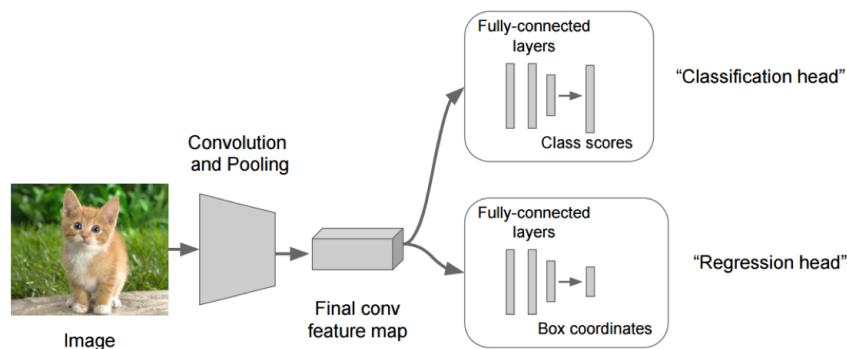


图 16 目标定位任务中的神经网络基本结构

3) 目标检测：一幅图包含多个物体，物体属于多种类别，常用算法有 R-CNN、Faster R-CNN、Faster R-CNN、YOLO、SSD 等

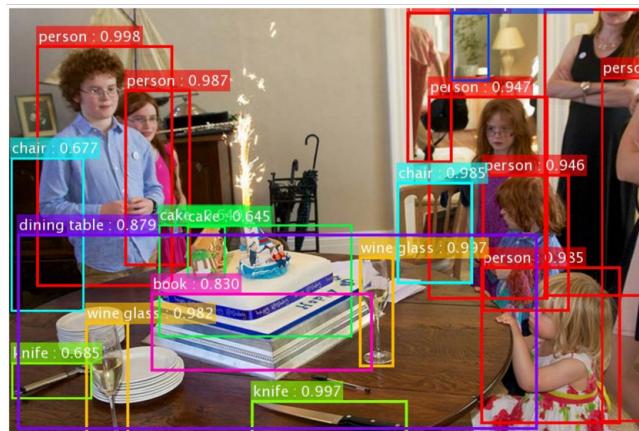


图 17 典型图像目标检测任务

4) 语义分割：计算机根据图像的语义来进行分割，例如让计算机在输入下面左边四只猫咪图像的情况下，能够输出右图将其分割开，神经网络结构参见图 18 下方。

Overfeat:

[https://people.eecs.berkeley.edu/jonlong/long\\_shelhamer\\_fcn.pdf](https://people.eecs.berkeley.edu/jonlong/long_shelhamer_fcn.pdf)

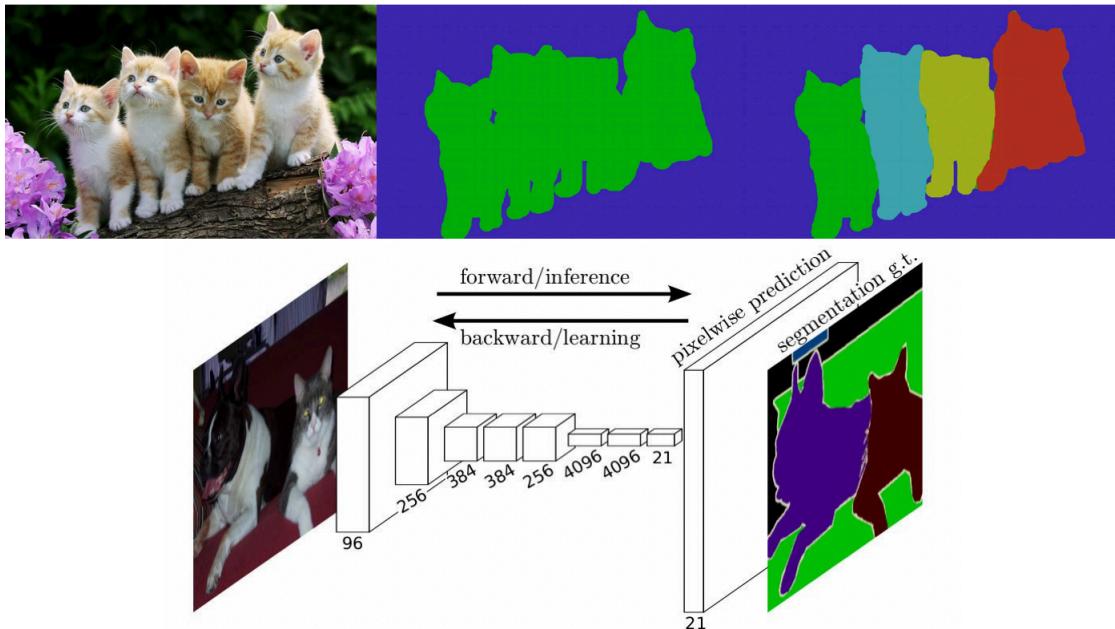


图 18 典型图像语义分割任务

总结，相较于机器学习，深度学习更强大，自动从数据学习特征，无需手工提取特征；节约时间，而且更好（自动提取了空间和图像结合的特征，人脑无法想象）；更深的网络具有较好的能力。同时深度学习的问题如下：需要防止过拟合，需要更大的数据量，对数据质量要求较高，需要进行数据标注。

## 五、深度学习训练环境的配置

NVIDIA GPU + 对应驱动程序 + CUDA + cuDNN + TensorFlow (PyTorch)。

首先，Google 或者百度一下你的电脑是否配有 N 卡，如果没有（比如我），那就不需要考虑下面的环境配置过程了。但是还有别的出路，比如上 Google Colab (需科学上网)、Kaggle 上自己找显卡资源吧。

接下来的教程适用于电脑配备有 N 卡的用户。

1) 下载驱动：进入 Nvidia 官网找到适合你电脑显卡的驱动并下载。

<https://www.geforce.com/drivers>

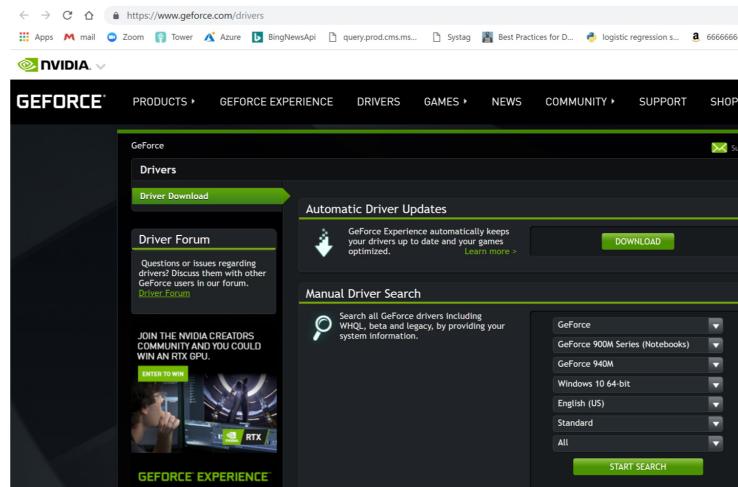


图 19 下载驱动步骤

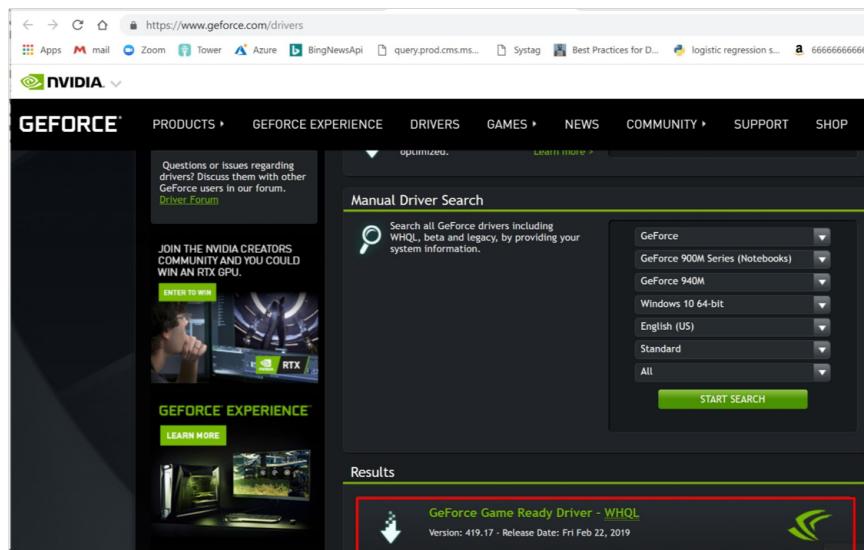


图 20 下载驱动步骤

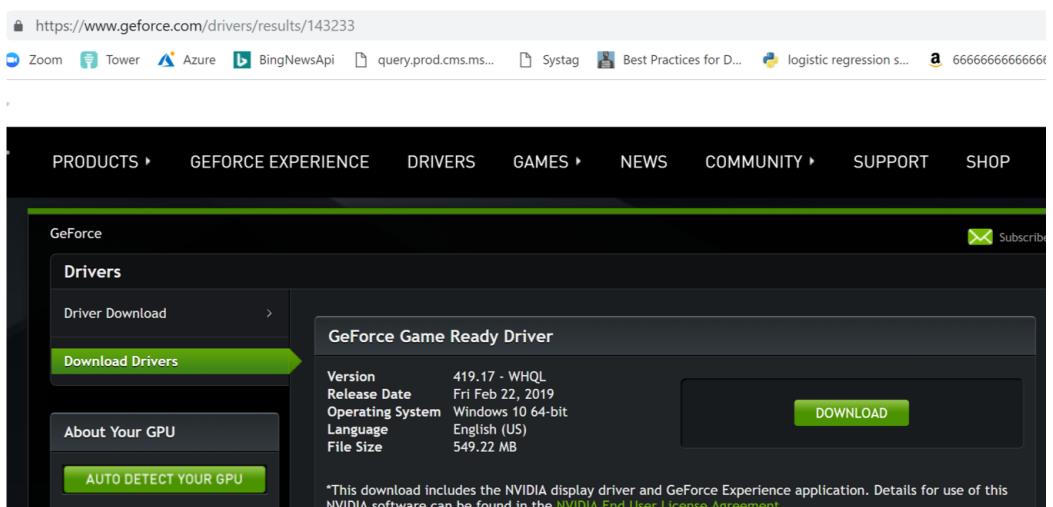


图 21 下载驱动步骤

然后重启电脑，打开任务管理器，看看你的 GPU 能不能正常使用（一般都是能的，不能再重装驱动吧）

2) **下载安装 CUDA**: 这是一种由 NVIDIA 推出的通用并行计算架构，该架构使 GPU 能够解决复杂的计算问题，加速矩阵计算，并且 CUDA 是向后兼容的，即旧版本用新的系统打开依然可以用。只有少数 GPU 有不适用的 CUDA 版本。

查询 GPU 适用于哪一种 CUDA 的链接:

<https://developer.nvidia.com/cuda-gpus>

CUDA 下载链接（建议使用本地安装包，稳定）：

<https://developer.nvidia.com/cuda-downloads>

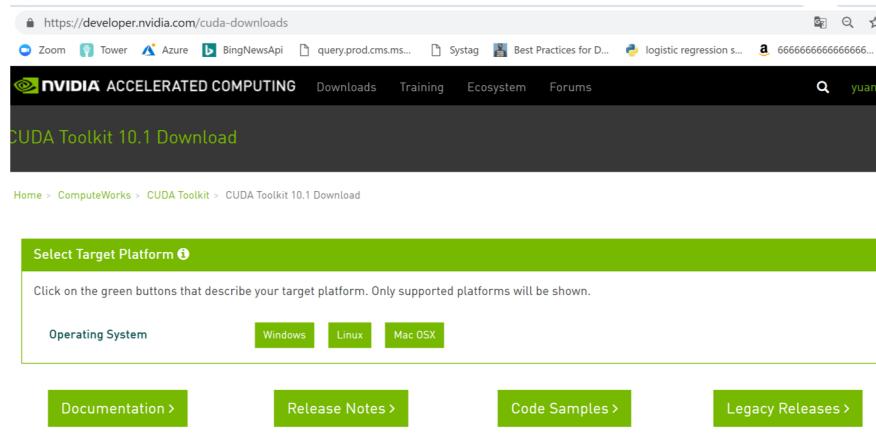


图 22 下载安装 CUDA 步骤

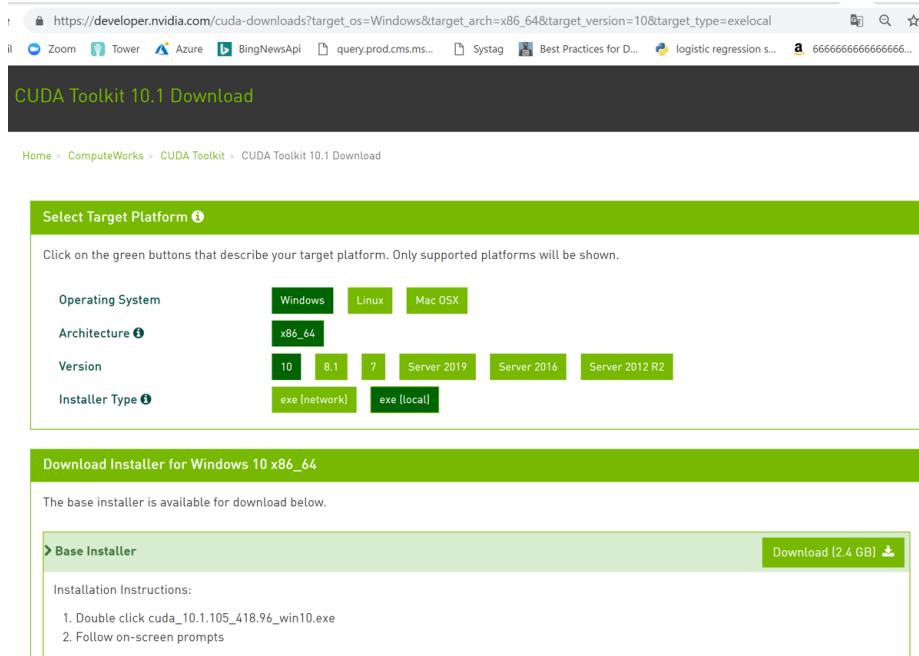


图 23 下载安装 CUDA 步骤

3) 下载安装 cuDNN: cuDNN 是专门用于神经网络加速计算的工具，与 CUDA 配套，版本需要一致。下载完后将其解压缩到 C 盘的 **cuda** 文件夹下，并需要修改注册表，参见下图。

cuDNN 下载链接:

<https://developer.nvidia.com/cudnn>

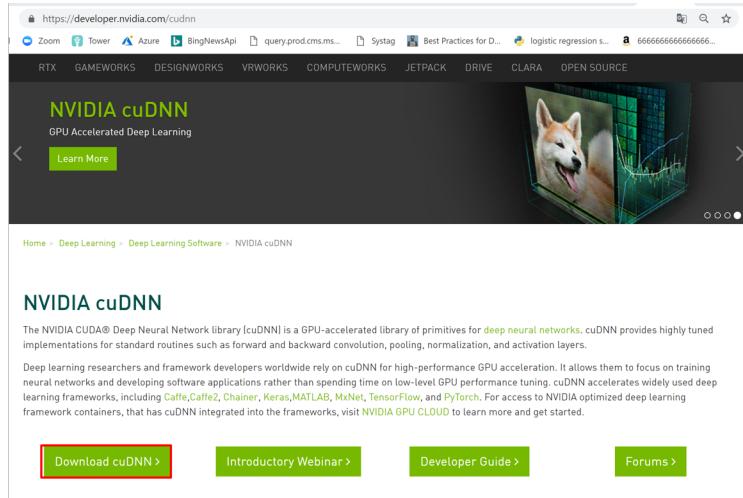


图 24 下载安装 cuDNN 步骤

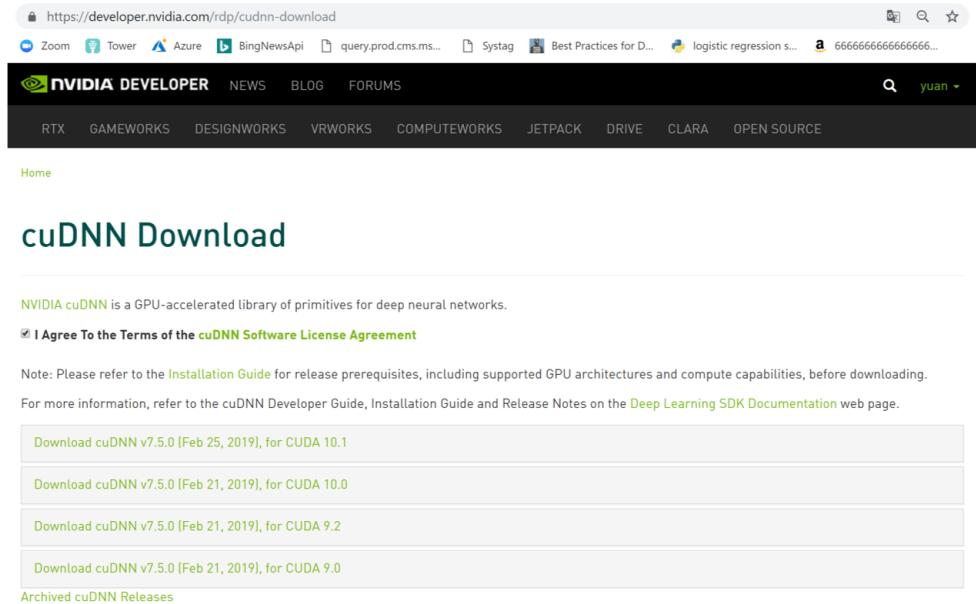


图 25 下载安装 cuDNN 步骤

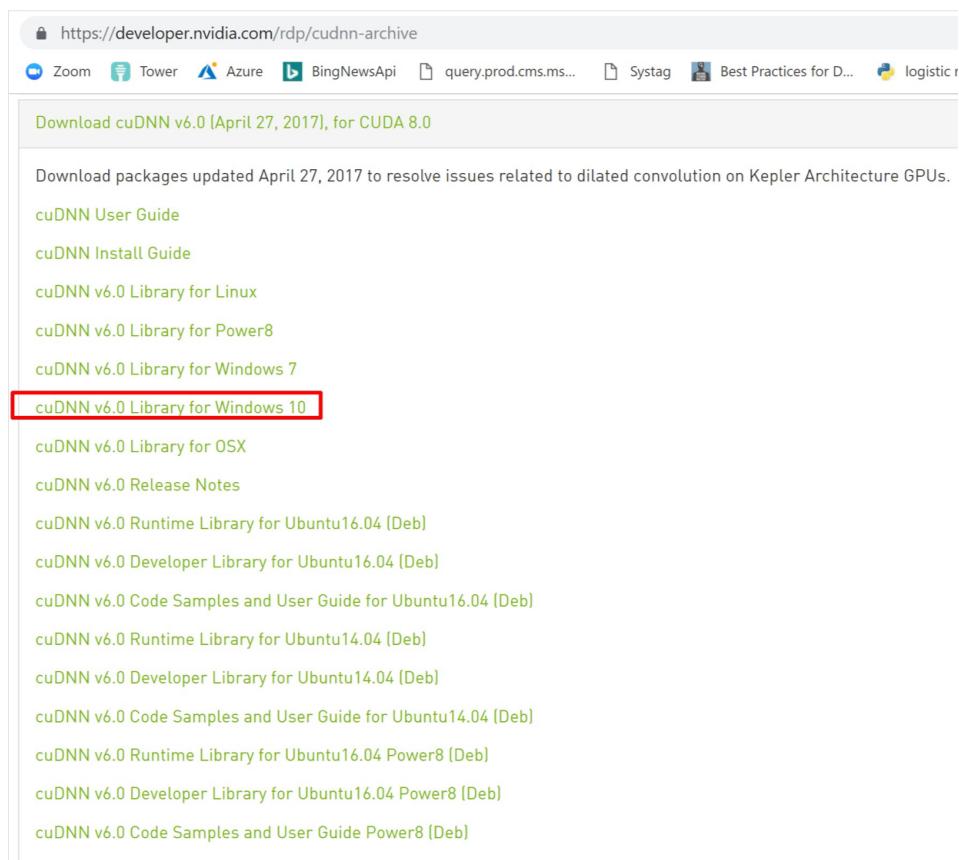


图 26 下载安装 cuDNN 步骤

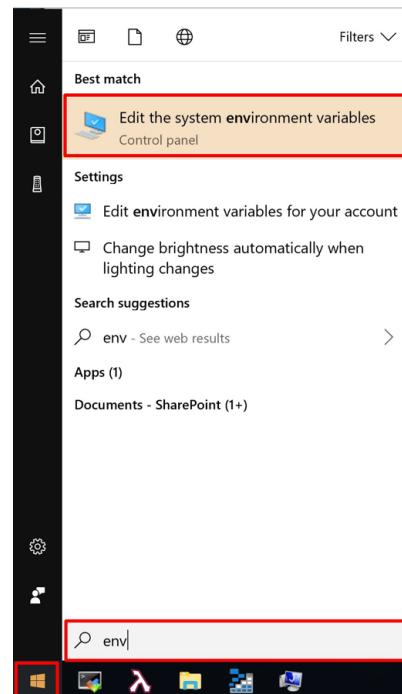


图 27 修改注册表

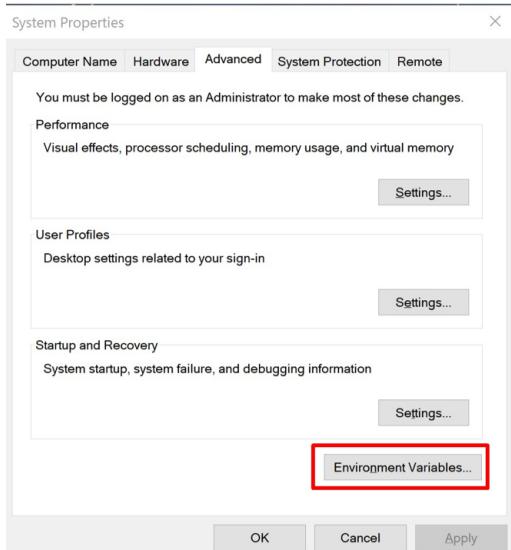


图 28 修改注册表

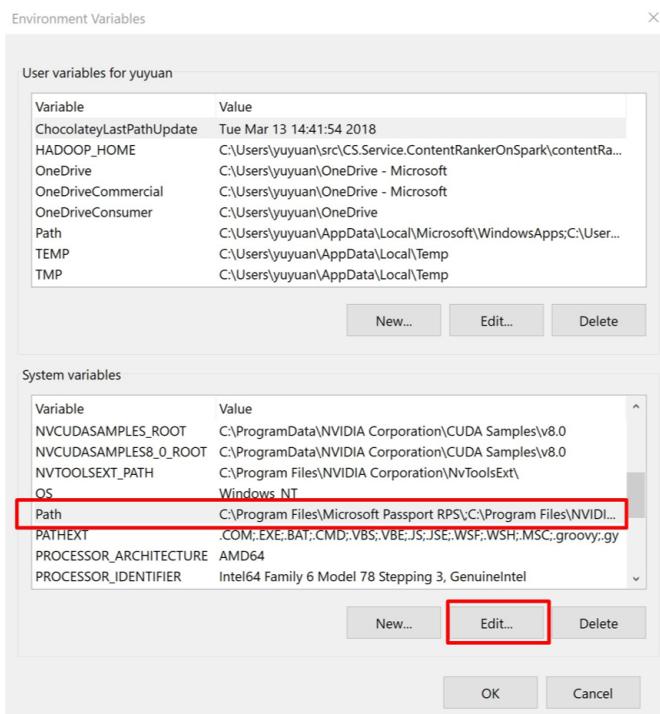


图 29 修改注册表

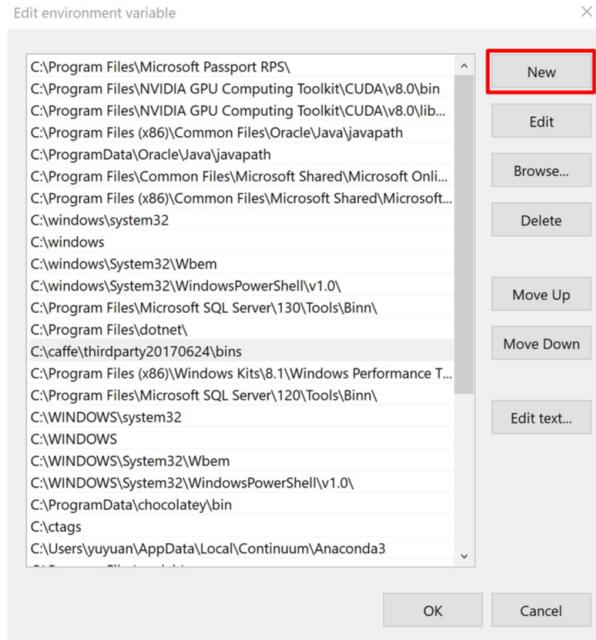


图 30 修改注册表

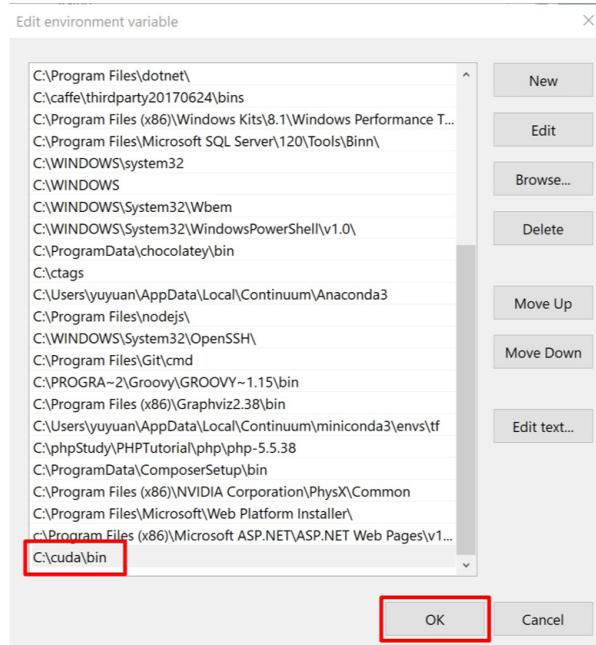


图 31 修改注册表

- 4) 安装 TensorFlow GPU 版本或者 PyTorch GPU 版本：从官网找到对应的 pip 安装代码后，通过命令行安装。

### 检测 TesorFlow GPU 是否有效代码

```
import tensorflow as tf
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
```

## 检测 PyTorch GPU 是否有效代码

```
import torch as T
T.cuda.current_device()
T.cuda.device_count()
T.cuda.get_device_name(0)
```