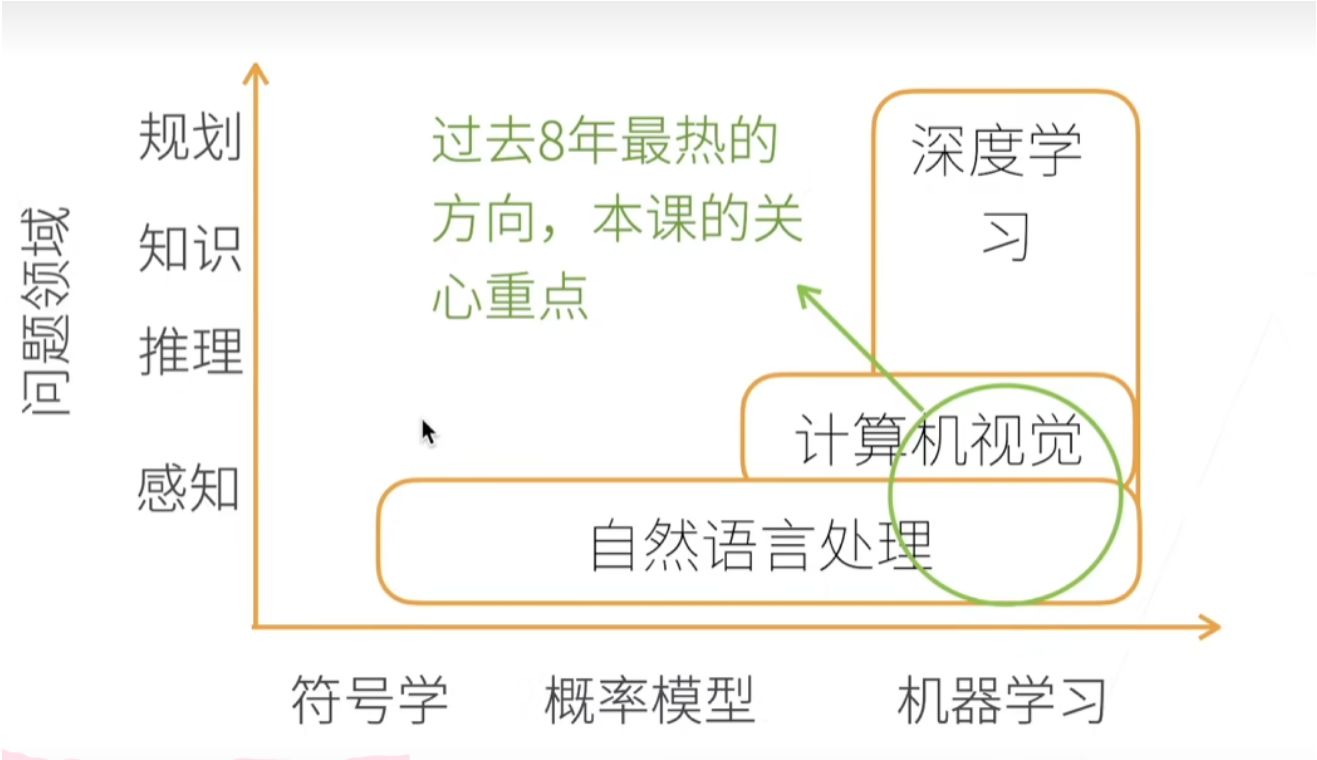
01.序言：关于深度学习

AI地图

看如下一张图：



这张图将AI划分为两个维度，在X轴包含的是三个不同的模式，在Y轴包含四个问题领域，其中包含符号学、概率模型、机器学习三个模式，以及感知、推理、知识、规划四个问题领域。

其中问题领域中的**感知指形如看到、听到环境中内容的能力。**
推理是指基于感知的内容推理出相关的内容，从已有的资料迭代出新的资料。
知识则是通过感知和推理来形成自己的知识体系，相当于模型的训练，通过感知和推理数据集来形成一个特定的知识体系。
规划是基于已有的知识体系来对未来进行规划整理。

在图中比较热门的三个方向，他们就是基于这是三个模式和四个问题领域。

自然语言处理（NLP）：

自然语言处理虽然取得了较大的进展，但是目前仍然处于简单的感知层次，相当于人脑对自然语言的感知和一系列处理，例如机器翻译就是自然语言处理中的一个重要应用。

（这里由于课程是21年的，放在2024年，根据我个人的理解，自然语言处理似乎已经突破了感知的问题领域，现在LLM完全能够基于给定的自然语言输入来推理出对应的输出，甚至在一些比较领先的大模型中，例如chatGPT，完全可以在一个会话中进行知识体系的构建。）

NLP的发展经历了符号学、概率模型、机器学习三个模式阶段，因为语言本质就是符号，通过传统的符号学就可以对自然语言进行处理，而概率模型和机器学习，是将NLP的效率进一步提高了，或者说拓展的NLP的应用范围。

计算机视觉：

计算机视觉，相比自然语言处理，最直观的体现是图像和文字的差异，图像是由不同的像素点组成的，也就是不同的RGB或者HSV颜色数据，对图像的处理很显然通过符号学是难以达到的，颜色的表示，似乎不像文字那样有一个标准的符号。因此计算机视觉大部分是基于概率模型和机器学习实现的。

不过符号学并非完全不能应用于计算机视觉领域，例如最著名的Opencv这一开源框架是完全结合了这三个模式，如边缘检测（Canny边缘检测）、图像滤波（高斯滤波、均值滤波等）和形态学操作（膨胀、腐蚀等）这些图像处理技术中就采用了符号学的方案。

其中还有背景减法算法用于运动检测，卡尔曼滤波器用于目标跟踪，贝叶斯分类器用于模式识别等处理方案使用到了概率模型。

使用Opencv实现的目标检测、图像分类的应用，就采用了SVM（支持向量机）、KNN（K邻近算法）等典型机器学习算法。

深度学习：

在这里首先明确一点，关于人工智能、机器学习、深度学习这三个概念的关系。

首先，人工智能即AI（Artificial Intelligence）是一门研究和开发用于模拟、延伸和扩展人类智能的理论、方法、技术及应用系统的科学。

人工智能是一门学科，他涉及的内容非常广泛，其中包含：专家系统、机器人学、自然语言处理、计算机视觉、机器学习等。

也就是说，机器学习是人工智能的一个子领域，他专注于研究和开发使计算机能够通过数据进行学习和改进性能的算法和技术。

而深度学习又是机器学习的一个子领域，相对于传统的机器学习，深度学习则利用多层神经网络（通常包含数十层甚至数百层）进行特征提取和模式识别。

深度学习中常见的模型有：卷积神经网络（CNN）、循环神经网络（RNN）、生成对抗网络（GAN）等
所以，这三个概念的关系是：

$$DL \in ML \in AI$$

然后，深度学习这个重要的领域，他也可以实现之前的自然语言处理和计算机视觉，以及其他的具体应用，例如语音和音频处理、医学和生物学、自动驾驶等。

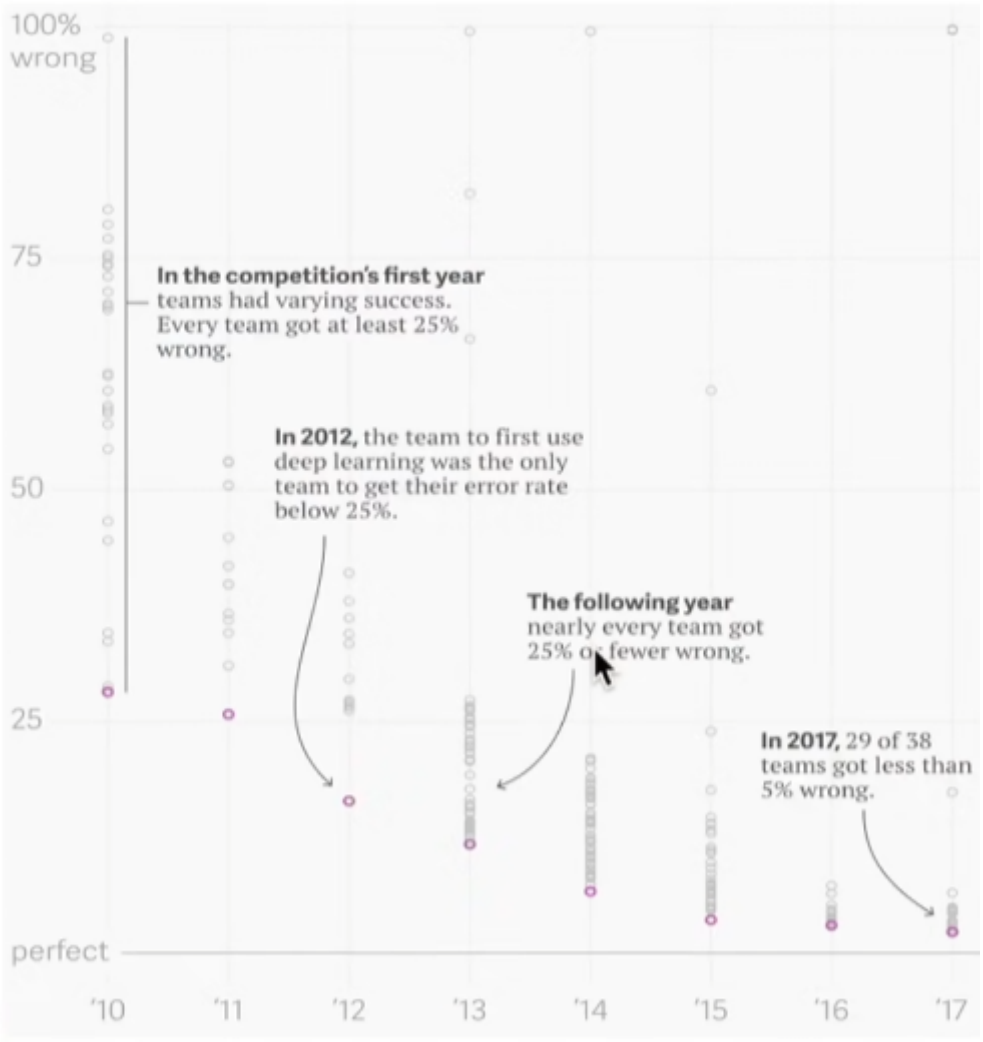
深度学习的领域突破

深度学习的产生和发展对一些应用领域产生的极大的影响，通过深度学习技术，使这些领域产生的革命性的突破。

图片分类

在图片分类领域（计算机视觉的一个重要应用场景），深度学习带来了极大的收益，在深度学习对图像分类的任务中，数据集是一个重要的因素。

在 image-net 这个网站中，包含了数百万张不同的图片数据集，这些图片可供任何人进行非盈利性的使用。



上图表示图片分类领域中的错误率变化趋势，在深度学习开始发展时，图像分类任务的错误率极大的降低了。

物体检测和分割

Mask R-CNN 是一个强大的物体检测和分割的开源项目，下图所示即为通过Mask R-CNN进行物体检测和分割的结果：

其中标注出了人和飞机的位置，这里再解释一下，物体检测即在图像中检测某个物体的位置，即他所处的图像区域，而分割则是判断图像中的一个像素点属于哪个物体。

样式迁移

MXNet Gluon Style Transfer GitHub Repository 是一个基于MXnet的样式迁移项目，样式迁移即通过给定两个输入图像，一张作为原始图像，一张作为参考图像，通过样式迁移可以将参考图像的风格样式迁移到原始图像，输出是处理后的原始图像。

这里我克隆了一下这个项目，并在本地部署运行，记录一下过程：
这里我首先从github上克隆项目：

```
git clone https://github.com/zhanghang1989/MXNet-Gluon-Style-Transfer.git
```

然后在conda中创建虚拟环境：

```
conda create --name Style-Transfer python=3.6
```

这里踩坑了，由于Style-Transfer是基于MXnet框架的，我起初创建的虚拟环境是3.11版本，但是在安装mxnet时一直报错，提示无法编译numpy，但是我已经安装好numpy，网上搜索了一大堆解决方案都尝试没有用，后面仔细看了一下报错信息，有一个依赖项的需要的Python版本是3.6，于是我将环境改成3.6版本，重新安装依赖，这时候安装上了mxnet，但是我想使用GPU版本的mxnet，查看我的CUDA版本：

```
nvidia-msi -q
```

CUDA版本是12.5，我尝试：

```
pip install mxnet-cu125
```

但是无法找到包，我在阿里镜像站中也没有搜索到MX对应CUDA12.5版本的包。

这里我看到一篇文章安装的是1.10版本的，我试着硬着头皮安装：

```
pip install mxnet-cu110
```

这个版本的可以安装，但是和我的CUDA版本不匹配，很明显，无法执行脚本。

接着卸载MXnet和GPU版本的MXnet，重新安装。

只能通过CPU运算了，在Readme中给出了执行main.py的参数:

- `--content-image`: path to content image you want to stylize.
- `--style-image`: path to style image (typically covered during the training).
- `--model`: path to the pre-trained model to be used for stylizing the image.
- `--output-image`: path for saving the output image.
- `--content-size`: the content image size to test on.
- `--cuda`: set it to 1 for running on GPU, 0 for CPU.

将--cuda参数设置为0即可只使用CPU进行运算，但是在执行的时候还是报出了一个错误：

```
warnings.warn("load_params is deprecated. Please use load_parameters.")
```

这个就很好解决，打开报错文件，找到位置，发现是在调用 `load_params` 时报出的这个错误，这个函数只在main.py中有两个引用，根据报错信息，将其改为 `load_parameters` 即可。

完整运行一下：

```
python main.py eval --content-image images/content/venice-boat.jpg --style-image images/styles/candy.jpg --model models/21styles.params --content-size 1024 --cuda=0
```

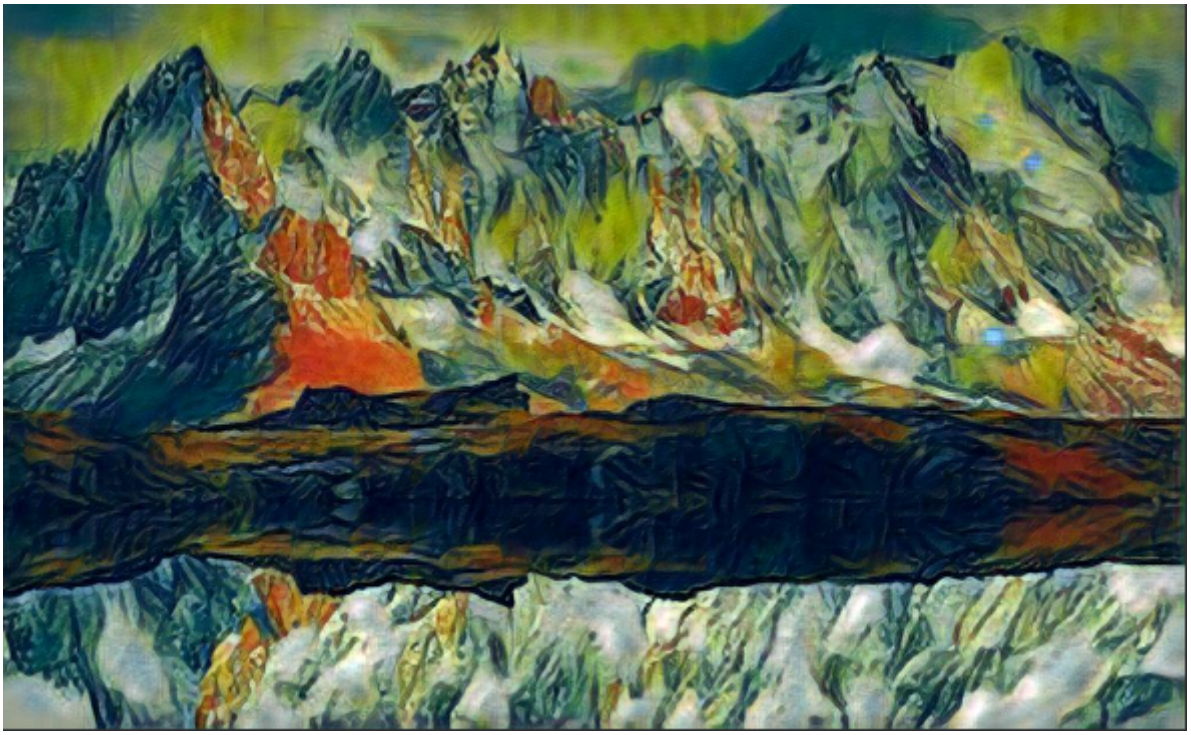
等待几十秒，在项目根目录输出了结果图片，然后加上自定义参数--output-image自定义输出位置，注意这里项目使用的 `fopen()` 是直接创建文件，而没有做目录存在性检测，因此如果需要指定目录，需要先新建目录。
我在根目录下新建了output目录，然后运行：

```
python main.py eval --content-image images/content/img.png --style-image
images/styles/img.png --model models/21styles.params --content-size 1024 --
cuda=0 --output-image images/output/a.jpg
```

这里的原图像是：

参照图像是：

输出的结果：



这里就可以很直观地看出，样式迁移会将参考图片的画风和色调作为滤镜应用到原始图像。

文生图

文生图就是一个热门又经典的应用了，最典型的例子是OpenAI出品的Dell-E，通过给定文字描述产生相关的图片。

文字生成

除了计算机视觉方面，深度学习在NLP中也具有革命性的突破，最典型的就是各种用于语言文本生成的LLM。

无人驾驶

例如特斯拉的无人驾驶，还有最近讨论比较多的百度的那个叫什么的自动驾驶，这里就是将各种人工智能的应用场景综合在一起了，涉及到计算机视觉、机器人系统、专家系统等等。

案例

李沐老师在课程中提及到的是广告点击的案例，这里我发散了一下，根据广告点击案例的思路大概想了一下，如果通过机器学习来实现web防火墙的攻击预测的大概实现步骤。

首先，不管是广告推荐系统还是攻击预测系统，他们的本质实际上都是一类预测问题。

对于预测问题，通过深度学习来解决最关键的仍然是适当的数据集和训练好的模型。

针对一个攻击预测系统的实现，首先需要通过系统访问日志，将访问来源、访问目的、端口、连接类型、连接时间（如果有）、访问URI等信息搜集记录形成数据集，然后通过特征提取来对数据集中的数据提取特征，也就是上面提到的日志中包含的信息的特征，根据这些特征以及对应的神经网络来训练模型，例如在Encoder-Decoder架构就是将输入内容通过RNN或者LSTM来产生上下文向量，并且通过Encoder输出之后作为Decoder的输入。

模型训练完成之后，即可通过验证集来测试模型。例如随机挑选最近几十条访问日志，包含恶意用户和正常用户，通过模型预测他们是否存在攻击行为？在验证准确率较高的情况下，接入系统进行测试。

这里的预测流程是，针对每个访问的用户，获取其与训练集结构一致的字段，即访问来源、目的、端口、连接类型等，根据这些特征输入模型，通过模型预测其是恶意访问的概率来考虑是否放行。