
人工智能课程综述

目 录

一、学习之道.....	2
二、学习之径.....	4
三、课程目的.....	5
四、选择 TensorFlow 的原因.....	5
五、课程（项目）的安排.....	6

【说在前面的话】

君子不器

子曰：「書不盡言，言不盡意。然則聖人之意，其不可見乎。」

子曰：「聖人立象以盡意，設卦以盡情偽，繫（系）辭以盡其言，變而通之以盡利，鼓之舞之以盡神。」

乾坤其易之緼（yùn）邪？

乾坤成列，而易立乎其中矣。乾坤毀，則無以見易，易不可見，則乾坤或幾乎息矣。是故，形而上者謂之道，形而下者謂之器。

化而裁之謂之變，推而行之謂之通，舉而錯之天下之民，謂之事業。

【注】真理、知识，向上凝为“道”，落地实为“器”。道，是相通的，器却不同。深度学习有很多的框架、方法，“道”相同，“器”不同罢了。同学们在研究深度学习\机器学习知识时，不要纠结于使用哪种框架、哪个软件包，更不要被某一个编程语言所限制，它们之间总是存在一个隐藏的“道”，“道”理解了，“器”自然水到渠成。希望跟同学们共勉。

一、学习之道

我们先来聊一聊入门学习这件事。

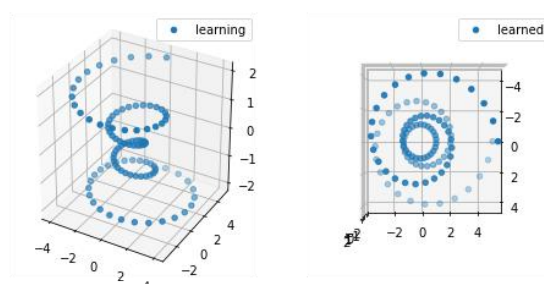
在我们进入一个新的领域的开始学习的时候，可能会遇到很多很多的困难和障碍，包括我们自己所读的一些书，它可能会让我们反而变得更加的迷茫。

原因是什么？为什么会发生这样的事？根据我们多年的课程设计开发经验，总结出来原因不外乎有三个：

1. 入门最难（概念膨胀）
2. 缺少学习路径
3. 知识大爆炸（与项目链接困难）

我们从一个领域切换到另外一个领域的时候，第一个要克服的困难就是关键词的爆炸。我们需要了解整个领域里边的概念，也就是它的行内话。这些术语可能有好多，由于我们还是门外汉，没有办法去理解术语和术语之间隐含的关系，很多时候只能靠死记硬背。而一般情况，行业内部的人去写的书或技术文章，他都会假定读者已经知道了，或者说他认为读者肯定会知道的内容，这部分内容就会直接被作者略过，这样就导致了这些门外汉，学习很努力，但不得其门而入。

举一个例子，大家就会一目了然。如下这张图，左侧代表的是螺旋式学习上升过程，右侧代表学习至顶端，站在顶端往下看知识的分布情况。



先来看左侧的螺旋式学习上升过程。在入门的阶段，我们会遇到很多的名词解释、技术知识点，这些知识点看起来都是离散的。随着时间的推移，我们开始逐渐的入门，开始对于各个知识点之间建立一定的联系。在某些知识点或者某些方面，我们会反复的去研读，然后过了一遍又一遍地加深理解。当我们经历了整个学习过程，也就是过了这个门槛之后，就变成了形而下者谓之器，我们开始去思考具体的应用、变通，也就是开始运用阶段，站在顶端了。

再来看右侧站在知识顶端向下看的过程。我们思考之前看过一些材料、书或者其他学习内容，你会发现，其实这些作者都具有很丰富的经验背景，也就是说他们已经是门内人了，站在了知识的顶端往下看，达到了如何应用或者如何推广这个阶段。当他们再往回推的时候，看到的是一个一个的知识点，这些知识点分布在不同的领域中。忽略了立体学习的过程，他就可能会把难度不同的 N 个知识点聚在一起，在同一个章节里边给大家讲，或者同一个部分的时候，让大家去了解和学习。这就导致刚开始学习时，还算轻松，越往后学习，越发困难和糊涂。其实，也就是缺少一个合理的学习路径，也就是第二个原因了。

关于第三个原因，知识大爆炸，与项目链接困难，可能同学们体会更加深刻。在信息化飞速发展的今天，各种各样的技术铺天盖地，纷繁杂乱，可以说没有合理的体系性引导，完全可能迷失在各类技术当中，到底要学什么？学的东西能用吗？用在哪里？怎么用？往往同学们在自学过程中，会一股脑儿学好多好多的知识，每天有看不完的资料，一个知识点接着一个知识点，越深入越庞大，仿佛构建了一个庞大的知识王国，但是，当遇到一个具体项目问题时，却卡了壳，不知从何下手，这也就是非常严重的理论应用不到实践（项目），也是市面上的各种资料、课程较为常见的问题所在，项目应用偏弱，知识罗列偏多，不能将繁多的知识点合理的融合到真正的项目里，甚至单纯为了知识点设计不结合实际的案例。所以，入门时需要一个合理的体系性引导，或者说好的学习路径规划，以及真正的融合知识点后的企业级项目设计，是多么的重要。

二、学习之径

第二个部分，看看我们课程设计的学习路径安排。

首先来看一下人工智能的本质。

简单说来人工智能在数学上的表达式，实际上是对于某个复杂函数的拟合。举个例子，有一句歌词：下雨要打伞。这个歌词告诉我们说，发现“下雨”这件事情，就要执行“打伞”这个动作，这个过程结构就可以表示成一个函数。人工智能就是通过类似传感器发现“下雨”，找到“伞并打开伞”。我们人其实在大脑中潜意识地进行了这些函数的拟合和执行。人工智能就是想办法用计算机的方法去对这些函数进行求解。所谓求解，就是找到这个函数的拟合的表达式，然后执行这种拟合。

再来看一下传统的学习路径状况。

要实现人工智能，就不得不从数学的角度来讲上面“打伞”的这件事，在高等数学中一个概念——泰勒级数，泰勒级数可以在平面上画出任意的曲线。推而广之，我们可以用类似的方法去拟合任意的曲面。当这个表达方法足够复杂的时候，就可以通过数学的方法来拟合任意复杂的函数。然后我们想办法把这些东西用编程的方法去实现了它。要想实现这些技术，我们在进入到人工智能这门课程之前，需要先学习高等数学，知道什么是导数、曲线曲面积分，什么是偏导，然后还需要学习线性代数，搞清楚什么是矩阵的秩、逆矩阵……等等的东西，还需要学习概率论，搞清楚什么是概率、贝叶斯公式，什么是概率分布函数，什么是概率密度函数……

当所有这些知识学完了之后，会告诉你，OK 你现在可以开始搞人工智能了。你可以去进行梯度下降的求导、进行图优化、对于函数进行拟合，进行拟合时，你还需要注意不要过拟合，需要加上 L1 因子、加上 L2 因子……，这条路子整个下来跑下来了之后，你就变成了一个可以进行人工智能开发的一个工程技术人员。

这一条传统学习路线的难度是在于：在实际有产出之前，你需要去了解掌握无穷多的概念，而且了解到多深你也不清楚，以及对于后边开发是否有必要你也不知道。更要命的情况是有些人就没有数学的一些天赋（但可能有写代码的天赋）。这一系列的数学上的障碍极有可能把你档在了门外。

那么，在这种情况下，我们又对人工智能很热爱，我们怎么能够入这个门呢？明确的答复大家：可以！可以通过编程这条路进人工智能的大门。

我们课程设计的學習路径关键点。

我们课程设计对于学习路径的关键点：惰性求值。根据事物的结果去倒推，要达到这一步，需要进行哪些努力，找到末来求本。这条路要求大家对于程序，特别是读别人代码的能力一定要强，简单说来，就是先来写代码，进入编程阶段，再来理解原理知识。学习的过程必然是痛苦的，我们开这门课的目的就是帮助大家把这个阶段度过去，让大家可以通过一个相对比较简单、相对比较轻松的路线，通过编程这条路，或者通过技术这个视角，通过计算机这个视角去倒推，去看一下 AI 实际是什么样的，从而走上 AI 之路。

三、课程目的

1. 了解深度学习的能力与局限
2. 了解机器学习领域的工作流程、常见术语
3. 能够使用专业术语与深度学习领域的资深研究人员进行技术交流
4. 能够完成深度学习领域的工程应用

四、选择 TensorFlow 的原因

目前市面上存在很多深度学习的框架，如 TensorFlow、PyTorch、MXNet、CNTK 等等，它们各有特点，但是各种框架大同小异。

选择具体使用框架，一般考虑的要素有：

1. 用户社区。最新的模型是否有社区实现，是否有足够多的用户协助 debug。
2. 性能。能够支持工业规模的大数据量。
3. 移动端支持。是否支持移动端需求。

综合这些因素，我们在进行课程设计时，在 TensorFlow 和 PyTorch 之间进行了严谨的权衡。

✓ PyTorch 的优势

PyTorch 是 Torch 在 Python 上的衍生。因为 Torch 是一个使用 Lua 语言的神经网络库，Torch 很好用，但是 Lua 又不是特别流行，所以开发团队将 Lua 的 Torch 移植到了更流行的语言 Python 上。PyTorch 还有如下的亮点：

- ✧ 2018 年增长很快，很多新的模型、论文，使用 PyTorch 制作最初版本
- ✧ 与 Caffe2 进行了整合，速度有了保障
- ✧ 支持动态图（对 RNN 类的模型支持会更好）

✓ TensorFlow 的优势

- ✧ 工业强度，被广泛验证过
- ✧ 支持 TPU
- ✧ 移动端和嵌入式部署
- ✧ 对新模型的支持不弱于 PyTorch

综合而言，TensorFlow/Keras 相比性价比最高。在通过 TensorFlow 了解深度学习的基本原理后，转用 PyTorch 或其他框架，都相对比较简单、比较容易上手转型。

五、课程（项目）的安排

在课程设计、项目安排上，我们充分运用了认知理论+ARCS 模型（教学中的动机设计模型）理论，综合知识的推导和运用，由简入深，逐渐扩展人工智能的认知领域。

大家知道，人类的认知是有一定规律的。只要掌握了认知规律，抓住认知特点，对知识的掌握就能够起到事半功倍的效果。人类的认知学习，从幼儿到成年，对待事物的认知是从感知世界到理性世界，简单地说，幼儿最初是通过视觉、听觉来感知世界，逐步建立一系列的事物联系，然后语言系统建立，进而建立更加完备的认知体系，认识这个世界。我们的人工智能基本理论就是建立在“人”的基础上，希望机器像人一样智能。遵循规律，我们在课程规划时，引入认知理论+学习动机模型理论，根据人工智能领域应用方向，课程设计采用如下的顺序安排各个领域内容的学习，课程更加符合认知规律：



人工智能应用领域虽然已经非常广泛，但不外乎就是上面这三个部分，我们的课程依据如上顺序进行切入。下面我们再来看一下，从技能的角度，我们是如何切入学习的，也就是技能路线图。

技能路径主线

技能学习路径的安排主线如下：

- 1) 先分别从原生 **Python+TensorFlow** 实现手写体识别入手
- 2) 然后深入浅出讲解透彻机器学习理论
- 3) 再进入深度学习实现项目理论升华
- 4) 实战项目应用开发

这样的安排，充分体现了科学性，符合学习认知规律：

- 1) 先分别从原生 **Python+TensorFlow** 实现手写体识别入手

从原生 Python 入手去实现机器学习领域的手写体识别项目。在当前的市场上，通过调用深度学习框架的 API 可以很便捷的实现这个项目案例，那么为什么我们还要选择采用原生方式呢？

- ✓ 首先，MNIST 这个数据集可谓是机器学习领域的入门经典案例，能让大家快速入门、方便学习，正因为如此，每一个刚入门的学习者基本都是从这个项目入手；
- ✓ 其次，为了“知其然还要知其所以然”。市面上可以很容易的找到深度学习框架实现此案例的源码，但是从原生角度去将事物本质展示出来的却寥寥无几，往往是知其然，不知其所以然。其实采用原生代码来一行一行的实现，对原理的理解帮助非常之大，在一入门时就了解其原理，为今后的开发打好基础，这个项目案例的实现方式具有很强的教学意义，从原生角度去实现这个项目案例也与从编程的角度去学习 AI 相呼应。不需要高大上的理论基础，从编程入手直接切入人工智能，更加符合大部分人的认知规律。
- ✓ 第三，在理解了基本原理基础上，我们可以全身心的把精力投入到机器学习的基础上。在这部分学习中，我们将基础入门级的知识嵌入在项目的各个节点里，在原生Python实现该项目案例时更侧重于思维，随后用 TensorFlow 实现，更侧重数学知识与工程实现的结合。在学习完这些项目后，无论理论实践，还是 TensorFlow 的基础，相信你都会有深入的理解和掌握。

2) 在项目中夯实基本数学原理，深入浅出讲解透彻机器学习理论

从编程角度入手，迭代式开发升级项目，接下来进行机器学习的内容。机器学习的基本理论是人工智能的基础，真正的奠基式理论，所以要想研究深度学习，机器学习理论是绕不过的内容。

- ✓ 内容嵌入了进阶的机器学习知识，帮助你了解机器学习中的众多原理、术语及其含义，了解分类结果的评价机制，解决分类问题（如垃圾邮件检测、搜索引擎商品分类处理）
- ✓ 结合项目实践回归、分类、过拟合等的具体问题的解决方法，开发预测数据类项目（如预测房价、预测销售额）。
- ✓ 了解损失函数、过拟合与 L1、L2 正则的意义。
- ✓ 开始使用神经网络的观点去解决问题，并且 TensorFlow 使用熟练之后，会讲解更顶层的深度学习框架 Keras，在训练模型中使用更少的代码来实现应用，把精力集

中于模型的优化调优。

3) 再进入深度学习实现项目理论升华

前两个部分学完之后，我们会抽出精力进行第三次迭代式学习，这次的内容主要是深度学习方向的前部技能，以图像识别内容为切入点，全面剖析深度学习的每一个知识点，从点到面掌握深度学习理论知识：

- ✓ 项目开发中的调优是必须掌握的。从有监督学习到无监督学习，反向传播原理，正则化方法，自适应学习率、梯度下降法、SGD、RMSprop 和 Adam，将图像识别中的参数调优详细介绍给大家，掌握高级优化方法。
- ✓ CNN 和 RNN 在图像识别中的应用。将 AI 图像领域的也教授给大家，在这个过程中有各式各样的项目案例供大家练手学习。除去基础内容外，本次还包含神经网络优化方法及精细调优，高阶的神经网络等。这个阶段与前两阶段结合，学习完之后，你会发现你已不再是门外汉，甚至已经具备工作的能力，基础打牢后进阶到装备升级的项目阶段。
- ✓ 物体定位和检测是图像识别的技术关键点。在这部分内容中，主攻主流的物体检测算法框架如 YOLO、SSD、YOLO3。通过识别车牌号码、路标牌等等项目介绍自然场景下的光学识别技术、文本识别模型 CTPN，彻底解决物体定位识别问题。

4) 实战项目应用开发

学到这里，我们已经具备了扎实的技术功底，本部分进入业界主流项目的实战开发。其实可以认为是 AI 装备的一个升级，至此无论使用哪种框架，万变不离其宗，通过图像识别这块的项目经验积累和学习，我们已经可以灵活运用各种人工智能技术，接下来我们顺利的进入人脸识别、语音识别、自然语言处理方向的开发方向。

- ✓ 首先，在这部分内容中，通过丰富的实战项目，如人脸打卡系统、银行卡识别、身份证识别巩固图像识别技术。
- ✓ 同时，对图像识别算法在其他领域中的扩展进行研究，如语音识别相关的 RNN 技术介绍，基于 DeepSpeech 语音识别等等。
- ✓ 对比传统自然语言处理机制，介绍 Attention 机制，开发基于 Seq2Seq 的聊天机器人等项目，进一步丰富自己的开发经验。
- ✓ 增加就业筹码，解决就业后的技术升级问题。部分项目的开发会采用业界知名常用的开源项目进行，以此帮助学生掌握如何运用和掌握开源项目，学会自我提升的能

力，提升进入职场后，对于技术升级再就业的能力。

所有项目，均来源于第一线企业，在此也一并感谢支持教育事业的第一线人工智能者们，提供给我们企业级开发项目，希望同学们带着敬畏之心进行项目实战，因为，不久的将来，这些项目将是你实际工作中要做的事情。完善的技术、项目学习规划脉络，相信在不久的将来就会迎来你的 AI 职业生涯。

不悔过去，不畏明天，希望一起，为自己，奋斗！