CB02-6 A Career In AI &
Sum of CB Series V1

Three key steps of career growth are **learning foundational skills, working on projects** (to deepen your skills, build a portfolio, and create impact), and **finding a job.** These steps stack on top of each other:

LEARNING → **PROJECTS** → **JOB**

Initially, you focus on **learning foundational skills.** Chapters with the 🔒 cover topics about learning foundational technical skills.

After having gained foundational technical skills, you will begin **working on projects.** During this period, you'll also keep learning. Chapters with the 🤖 focus on projects.

Later, you will work on **finding a job.** Throughout this process, you'll continue to learn and work on meaningful projects. Chapters with the 💻 focus on a job search.

These phases apply in a wide range of professions, but AI involves unique elements. For example:

LEARNING

Learning foundational skills is a career-long process:

AI is nascent, and many technologies are still evolving. While the foundations of machine learning and deep learning are maturing — and coursework is an efficient way to master them — beyond these foundations, keeping up-to-date with changing technology is more important in AI than fields that are more mature.

PROJECTS

Working on projects often means collaborating with stakeholders who lack expertise in AI:

This can make it challenging to find a suitable project, estimate the project's timeline and return on investment, and set expectations. In addition, the highly iterative nature of AI projects leads to special challenges in project management: How can you come up with a plan for building a system when you don't know in advance how long it will take to achieve the target accuracy? Even after the system has hit the target, further iteration may be necessary to address post-deployment drift.

JOB

Inconsistent opinions on AI skills and jobs roles:

While searching for a job in AI can be similar to searching for a job in other sectors, there are also important differences. Many companies are still trying to figure out which AI skills they need, and how to hire people who have them. Things you've worked on may be significantly different than anything your interviewer has seen, and you're more likely to have to educate potential employers about some elements of your work.

2024/7/25

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

Career In AI

Learn → Project → Career

Foundational machine learning skills: For example, it's important to understand models such as linear regression, logistic regression, neural networks, decision trees, clustering, and anomaly detection. Beyond specific models, it's even more important to understand the core concepts behind how and why machine learning works, such as bias/variance, cost functions, regularization, optimization algorithms, and error analysis.

Deep learning: This has become such a large fraction of machine learning that it's hard to excel in the field without some understanding of it! It's valuable to know the basics of neural networks, practical skills for making them work (such as hyperparameter tuning), convolutional networks, sequence models, and transformers.

Math relevant to machine learning: Key areas include linear algebra (vectors, matrices, and various manipulations of them) as well as probability and statistics (including discrete and continuous probability, standard probability distributions, basic rules such as independence and Bayes' rule, and hypothesis testing). In addition, exploratory data analysis (EDA) — using visualizations and other methods to systematically explore a dataset — is an underrated skill. I've found EDA particularly useful in data-centric AI development, where analyzing errors and gaining insights can really help drive progress! Finally, a basic intuitive understanding of calculus will also help. The math needed to do machine learning well has been changing. For instance, although some tasks require calculus, improved automatic differentiation software makes it possible to invent and implement new neural network architectures without doing any calculus. This was almost impossible a decade ago.

Software development: While you can get a job and make huge contributions with only machine learning modeling skills, your job opportunities will increase if you can also write good software to implement complex AI systems. These skills include programming fundamentals, data structures (especially those that relate to machine learning, such as data frames), algorithms (including those related to databases and data manipulation), software design, familiarity with Python, and familiarity with key libraries such as TensorFlow or PyTorch, and scikit-learn.

How much math do you need to know to be a machine learning engineer?

Is math a foundational skill for AI? It's always nice to know more math! But there's so much to learn that, realistically, it's necessary to prioritize. Here's how you might go about strengthening your math background.

To figure out what's important to know, I find it useful to ask what you need to know to make the decisions required for the work you want to do. At DeepLearning.AI, we frequently ask, "What does someone need to know to accomplish their goals?" The goal might be building a machine learning model, architecting a system, or passing a job interview.

Understanding the math behind algorithms you use is often helpful, since it enables you to debug them. But the depth of knowledge that's useful changes over time. As machine learning techniques mature and become more reliable and turnkey, they require less debugging, and a shallower understanding of the math involved may be sufficient to make them work.

For instance, in an earlier era of machine learning, linear algebra libraries for solving linear systems of equations (for linear regression) were immature. I had to understand how these libraries worked so I could choose among different libraries and avoid numerical roundoff pitfalls. But this became less important as numerical linear algebra libraries matured.

Deep learning is still an emerging technology, so when you train a neural network and the optimization algorithm struggles to converge, understanding the math behind gradient descent, momentum, and the Adam optimization algorithm will help you make better decisions. Similarly, if your neural network does something funny — say, it makes bad predictions on images of a certain resolution, but not others — understanding the math behind neural network architectures puts you in a better position to figure out what to do.

Of course, I also encourage learning driven by curiosity. If something interests you, go ahead and learn it regardless of how useful it might turn out to be! Maybe this will lead to a creative spark or technical breakthrough.

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

Learned:
1. AI History – CB01
2. NN & BP – CB02-1
3. Coding NN(Py&Np)—CB02-2
4. Tiny Autograd—CB02-3
5. PyTorch—CB02-4
6. Transformers—CB02-5

Accomplishments:
1. Coding: Python, PyTorch, Python
2. Ideas: NN, DL, BP, Engine, Transformers
3. Projects: NN; Autograd; Torc; Transformers
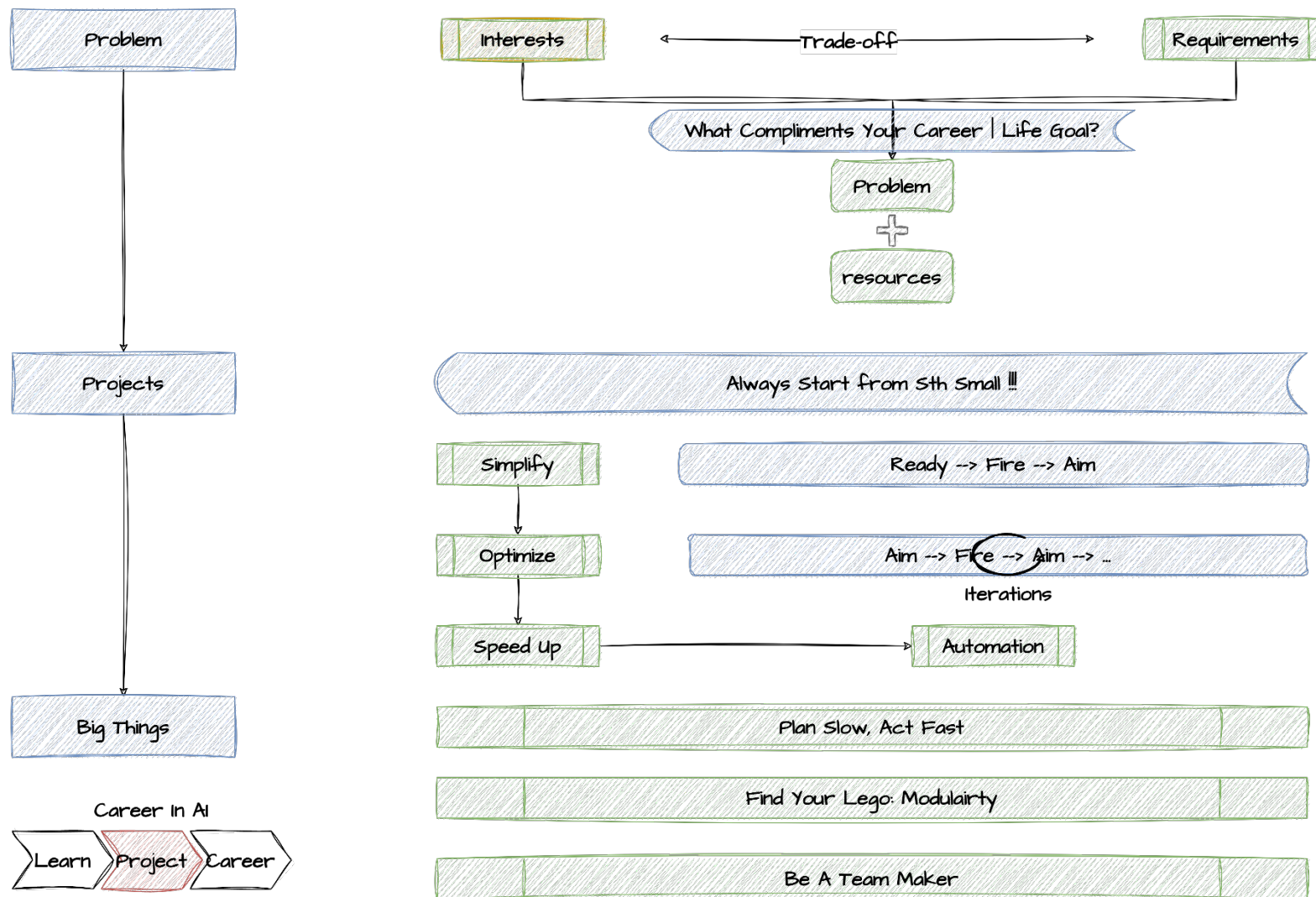
Not Covered:

1. Python(Coding):
   30 days of Python

2. ML & Math:
   Visualize-ML (Iris Series: Visualize Math -- From Arithmetic Basics to Machine Learning)

   3Blue1Brown (bilibili.com)

3. Deep Learning:
   动手学深度学习课程 (d2l.ai)

Problem

Projects

Big Things

Career In AI

Learn → Project → Career

Interests ←———— Trade-off ————→ Requirements

What Compliments Your Career | Life Goal?

Problem

+

resources

Always Start from Sth Small !!!

Simplify

Ready --> Fire --> Aim

Optimize

Aim --> Fire --> Aim --> ...

Iterations

Speed Up ———→ Automation

Plan Slow, Act Fast

Find Your Lego: Modulairty

Be A Team Maker

1. How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

2. Musk's Five Steps

3. Karpathy's Speech at Berkeley

4. - WRB (whereresearchbegins.com)

5. 怎样做成大事 (豆瓣) (douban.com)

6. 为什么伟大不能被计划 (豆瓣) (douban.com)

What if you don't have any project ideas?
Here are a few ways to generate them:

✓ **Join existing projects.** If you find someone else with an idea, ask to join their project.

✓ **Keep reading and talking to people.** I come up with new ideas whenever I spend a lot of time reading, taking courses, or talking with domain experts. I'm confident that you will, too.

✓ **Focus on an application area.** Many researchers are trying to advance basic AI technology — say, by inventing the next generation of transformers or further scaling up language models — so, while this is an exciting direction, it is also very hard. But the variety of applications to which machine learning has not yet been applied is vast! I'm fortunate to have been able to apply neural networks to everything from autonomous helicopter flight to online advertising, partly because I jumped in when relatively few people were working on those applications. If your company or school cares about a particular application, explore the possibilities for machine learning. That can give you a first look at a potentially creative application — one where you can do unique work — that no one else has done yet.

✓ **Develop a side hustle.** Even if you have a full-time job, a fun project that may or may not develop into something bigger can stir the creative juices and strengthen bonds with collaborators. When I was a full-time professor, working on online education wasn't part of my "job" (which was doing research and teaching classes). It was a fun hobby that I often worked on out of passion for education. My early experiences in recording videos at home helped me later in working on online education in a more substantive way. Silicon Valley abounds with stories of startups that started as side projects. As long as it doesn't create a conflict with your employer, these projects can be a stepping stone to something significant.

Given a few project ideas, which one should you jump into?
Here's a quick checklist of factors to consider:

✓ **Will the project help you grow technically?** Ideally, it should be challenging enough to stretch your skills but not so hard that you have little chance of success. This will put you on a path toward mastering ever-greater technical complexity.

✓ **Do you have good teammates to work with?** If not, are there people you can discuss things with? We learn a lot from the people around us, and good collaborators will have a huge impact on your growth.

✓ **Can it be a stepping stone?** If the project is successful, will its technical complexity and/or business impact make it a meaningful stepping stone to larger projects? If the project is bigger than those you've worked on before, there's a good chance it could be such a stepping stone.

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

Step 1: Topic Exploration

- 感兴趣的话题或问题是什么
- 立项的研究成果是什么；怎么样才算成功；项目想达到什么目的

Step 2: Explore Yourself

1. 根据Topic，在"数据库/搜索引擎"中搜索相关结果
2. 快速浏览结果列表（不用精读），阅读自己
   - 把让自己"心跳加速"的题目记录下来
   - 同时，把让自己感觉无聊的题目记录到另一张列表里
3. 对心跳列表，向自己提问，为什么这个条目吸引我：
   - 它让我想到了什么
   - 我为什么会注意到它
   - 我脑海中出现了什么问题
4. 对无聊列表，同样提出上述三个问题，并做一句话总结：
   - 比起XX，我更感兴趣的是：
3. 放置一段时间/一天后，用第三方视角观察：
   - 研究者关注点是什么
   - 他们最关心什么问题

Step 3: Ask Questions

写下至少20个与话题相关的问题：要关注小问题

Step 4: Re-phrase Questions

重写一遍问题，注意以下要点：
- 应该清晰、准确，不使用行业属于
- 应该建立在可验证、可证伪的证据与逻辑上
- 保持中立，不偏向任何结果
- 有明确的研究对象
- 思考关键词

Step 5: Make Hypothesis. Update Questions

1. 回顾问题列表
2. 列出关于该问题的原始假设
3. 对假设进行分类
   - A：目前想使用的假设
   - B：马上想抛弃的假设→抛弃之前考虑修改
   - C：不确定的假设
4. 一两句话解释分类原因
5. 对假设进行修改
6. 对问题进行修改

Question	Hypothesis	Class	Reson	Updated H	Updated Q

Step 6: Identify Problem

1. 把你的所有问题都列出来。
2. 先别试图回答这些问题，而是问问自己：这些问题是否有共同的关注点？如果有，是
3. 跳出你自己的身份。如果以别人的眼光来看这些问题，你认为把这些小问题串联在一
4. 把这些深层问题写下来。
5. 有必要的话，可以根据问题的具体程度或笼统程度确定其优先级，比如中级或高级。
   具体的事实性问题要笼统一些

Step 7: Seed Paper

1. 找到一篇种子文章/项目
2. 对文章进行多角度思考

我注意到了什么	可以提出的问题和关注点	下一份资料	更广泛的主题

3. 基于种子文章，向前向后找到相关论文，可以使用如Connected Papers等工具
4. 生成核心论文列表（10+）
   PS: How to read a Paper:

	Pass 1	Pass 2	Pass 3
title	√		every sentence
abstract	√	refs	√
introduction			√
method		Graphs	√
experiment		Graphs	√
conclusion	√		√

Ref: WRB (whereresearchbegins.com)

Over the years, I've had fun applying machine learning to manufacturing, healthcare, climate change, agriculture, ecommerce, advertising, and other industries. How can someone who's not an expert in all these sectors find meaningful projects within them? Here are five steps to help you scope projects.

Step 1

Identify a business problem (not an AI problem). I like to find a domain expert and ask, "What are the top three things that you wish worked better? Why aren't they working yet?" For example, if you want to apply AI to climate change, you might discover that power-grid operators can't accurately predict how much power intermittent sources like wind and solar might generate in the future.

Step 2

Brainstorm AI solutions. When I was younger, I used to execute on the first idea I was excited about. Sometimes this worked out okay, but sometimes I ended up missing an even better idea that might not have taken any more effort to build. Once you understand a problem, you can brainstorm potential solutions more efficiently. For instance, to predict power generation from intermittent sources, we might consider using satellite imagery to map the locations of wind turbines more accurately, using satellite imagery to estimate the height and generation capacity of wind turbines, or using weather data to better predict cloud cover and thus solar irradiance. Sometimes there isn't a good AI solution, and that's okay too.

Step 3

Assess the feasibility and value of potential solutions. You can determine whether an approach is technically feasible by looking at published work, what competitors have done, or perhaps building a quick proof of concept implementation. You can determine its value by consulting with domain experts (say, power-grid operators, who can advise on the utility of the potential solutions mentioned above).

Step 4

Determine milestones. Once you've deemed a project sufficiently valuable, the next step is to determine the metrics to aim for. This includes both machine learning metrics (such as accuracy) and business metrics (such as revenue). Machine learning teams are often most comfortable with metrics that a learning algorithm can optimize. But we may need to stretch outside our comfort zone to come up with business metrics, such as those related to user engagement, revenue, and so on. Unfortunately, not every business problem can be reduced to optimizing test set accuracy! If you aren't able to determine reasonable milestones, it may be a sign that you need to learn more about the problem. A quick proof of concept can help supply the missing perspective.

Step 5

Budget for resources. Think through everything you'll need to get the project done including data, personnel, time, and any integrations or support you may need from other teams. For example, if you need funds to purchase satellite imagery, make sure that's in the budget.

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

1. Class projects:

The first few projects might be narrowly scoped homework assignments with predetermined right answers. These are often great learning experiences!

2. Personal projects

You might go on to work on small-scale projects either alone or with friends. For instance, you might re-implement a known algorithm, apply machine learning to a hobby (such as predicting whether your favorite sports team will win), or build a small but useful system at work in your spare time (such as a machine learning-based script that helps a colleague automate some of their work). Participating in competitions such as those organized by Kaggle is also one way to gain experience.

3. Creating value

Eventually, you will gain enough skill to build projects in which others see more tangible value. This opens the door to more resources. For example, rather than developing machine learning systems in your spare time, it might become part of your job, and you might gain access to more equipment, compute time, labeling budget, or head count.

4. Rising scope and complexity

Successes build on each other, opening the door to more technical growth, more resources, and increasingly significant project opportunities.

Ready, Fire, Aim

Working on projects requires making tough choices about what to build and how to go about it. Here are two distinct styles:

✓ **Ready, Aim, Fire:** Plan carefully and carry out careful validation. Commit and execute only when you have a high degree of confidence in a direction.

✓ **Ready, Fire, Aim:** Jump into development and start executing. This allows you to discover problems quickly and pivot along the way if necessary.

Say you've built a customer-service chatbot for retailers, and you think it could help restaurants, too. Should you take time to study the restaurant market before starting development, moving slowly but cutting the risk of wasting time and resources? Or jump in right away, moving quickly and accepting a higher risk of pivoting or failing?

Both approaches have their advocates, and the best choice depends on the situation.

Ready, Aim, Fire tends to be superior when the cost of execution is high and a study can shed light on how useful or valuable a project could be. For example, if you can brainstorm a few other use cases (restaurants, airlines, telcos, and so on) and evaluate these cases to identify the most promising one, it may be worth taking the extra time before committing to a direction.

Ready, Fire, Aim tends to be better if you can execute at low cost and, in doing so, determine whether the direction is feasible and discover tweaks that will make it work. For example, if you can build a prototype quickly to figure out if users want the product, and if canceling or pivoting after a small amount of work is acceptable, then it makes sense to consider jumping in quickly. When taking a shot is inexpensive, it also makes sense to take many shots. In this case, the process is actually Ready, Fire, Aim, Fire, Aim, Fire, Aim, Fire.

After agreeing upon a project direction, when it comes to building a machine learning model that's part of the product, I have a bias toward Ready, Fire, Aim. Building models is an iterative process. For many applications, the cost of training and conducting error analysis is not prohibitive. Furthermore, it is very difficult to carry out a study that will shed light on the appropriate model, data, and hyperparameters. So it makes sense to build an end-to-end system quickly and revise it until it works well.

But when committing to a direction means making a costly investment or entering a one-way door (meaning a decision that's hard to reverse), it's often worth spending more time in advance to make sure it really is a good idea.

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

Career In AI

Learn › Project › Career

Pay attention to the fundamentals. A compelling resume, portfolio of technical projects, and a strong interview performance will unlock doors. Even if you have a referral from someone in a company, a resume and portfolio will be your first contact with many people who don't already know about you. Update your resume and make sure it clearly presents your education and experience relevant to the role you want. Customize your communications with each company to explain why you're a good fit. Before an interview, ask the recruiter what to expect. Take time to review and practice answers to common interview questions, brush up key skills, and study technical materials to make sure they are fresh in your mind. Afterward, take notes to help you remember what was said.

Proceed respectfully and responsibly. Approach interviews and offer negotiations with a win-win mindset. Outrage spreads faster than reasonableness on social media, so a story about how an employer underpaid someone gets amplified, whereas stories about how an employer treated someone fairly do not. The vast majority of employers are ethical and fair, so don't let stories about the small fraction of mistreated individuals sway your approach. If you're leaving a job, exit gracefully. Give your employer ample notice, give your full effort through your last hour on the job, transition unfinished business as best you can, and leave in a way that honors the responsibilities you were entrusted with.

Choose who to work with. It's tempting to take a position because of the projects you'll work on. But the teammates you'll work with are at least equally important. We're influenced by people around us, so your colleagues will make a big difference. For example, if your friends smoke, the odds increase that you, too, will smoke. I don't know of a study that shows this, but I'm pretty sure that if most of your colleagues work hard, learn continuously, and build AI to benefit all people, you're likely to do the same. (By the way, some large companies won't tell you who your teammates will be until you've accepted an offer. In this case, be persistent and keep pushing to identify and speak with potential teammates. Strict policies may make it impossible to accommodate you, but in my mind, that increases the risk of accepting the offer, as it increases the odds you'll end up with a manager or teammates who aren't a good fit.)

Get help from your community. Most of us go job hunting only a small number of times in our careers, so few of us get much practice at doing it well. Collectively, though, people in your immediate community probably have a lot of experience. Don't be shy about calling on them. Friends and associates can provide advice, share inside knowledge, and refer you to others who may help. I got a lot of help from supportive friends and mentors when I applied for my first faculty position, and many of the tips they gave me were very helpful.

I know that the job-search process can be intimidating. Instead of viewing it as a great leap, consider an incremental approach. Start by identifying possible roles and conducting a handful of informational interviews. If these conversations tell you that you have more learning to do before you're ready to apply, that's great! At least you have a clear path forward. The most important part of any journey is to take the first step, and that step can be a small one.

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)

How many days is a typical human lifespan?

- 20,000 days
- 100,000 days
- 1 million days
- 5 million days

When I ask friends, many choose a number in the hundreds of thousands. (Many others can't resist calculating the answer, to my annoyance!)

When I was a grad student, I remember plugging my statistics into a mortality calculator to figure out my life expectancy. The calculator said I could expect to live a total of 27,649 days. It struck me how small this number is. I printed it in a large font and pasted it on my office wall as a daily reminder.

That's all the days we have to spend with loved ones, learn, build for the future, and help others. Whatever you're doing today, is it worth 1/30,000 of your life?

Ref: How to Build Your Career in AI eBook - Andrew Ng Collected Insights (deeplearning.ai)