A Simple Guide

DeepLearning.AI

How to Build Your Career in AI

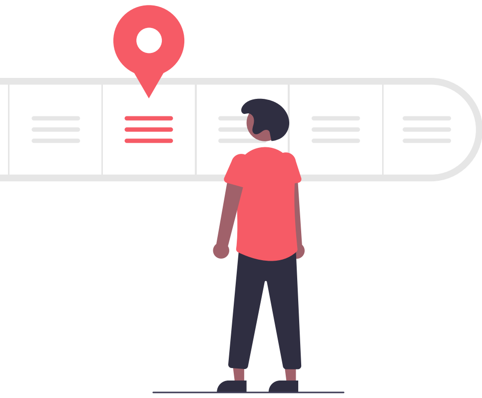
Collected Insights from Andrew Ng

Founder, DeepLearning.AI

"We believe that AI
is the new electricity
and will transform
and improve nearly all
areas of human lives."

Andrew Ng

Table of Contents



Coding AI is the New Literacy

Today we take it for granted that many people know how to read and write. Someday, I hope, it will be just as common that people know how to write code, specifically for AI.
Several hundred years ago, society didn't view language literacy as a necessary skill. A small number of people learned to read and write, and everyone else let them do the reading and writing. It took centuries for literacy to spread, and now society is far richer for it.

Words enable deep human-to-human communication. Code is the deepest form of human-to-machine communication. As machines become more central to daily life, that communication becomes ever more important.

Traditional software engineering — writing programs that explicitly tell a computer sequences of steps to execute — has been the main path to code literacy. But AI, machine learning, and data science offer a new paradigm in which computers extract knowledge from data. This technology offers another pathway to coding — one that strikes me as even more promising.
Many Sundays, I buy a slice of pizza from my neighborhood pizza parlor. The gentleman behind the counter may have little reason to learn how to build software applications (beyond personal growth and the pleasure of gaining a new skill).

But AI and data science have great value even for a pizza maker. A linear regression model might enable him to better estimate demand so he could optimize the restaurant's staffing and supply chain. He could better predict sales of Hawaiian pizza — my favorite! — so he could make more Hawaiian pies in advance and reduce the amount of time customers had to wait for them.

Uses of AI and data science can be found in almost any situation that produces data, and I believe that a wide variety of professions will find more uses for custom AI applications and data-derived insights than for traditional software engineering. This makes literacy in AI-oriented coding even more valuable than traditional skills. It could enable countless individuals to harness data to make their lives richer.

I hope the promise of building basic AI applications, even more than that of building basic traditional applications, encourages more people to learn how to code. If society embraces this new form of literacy as it has the ability to read and write, we will all benefit.

CHAPTER 1

Keys to Building a Career in AI

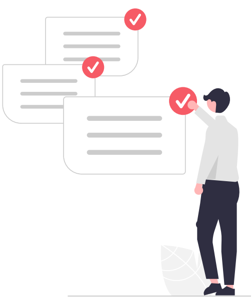
Swetha Mandava

Deep Learning Engineer at Nvidia

"I had a traditional entry into deep learning by pursuing a Master's Degree, but there are boot camps and online courses that allow people to build a foundation outside traditional paths."

AI continues to create numerous exciting career opportunities, and I know that many of you aim to develop a career in the field. While taking online courses in technical topics is an important step, being an AI professional requires more than technical skills. Lately I've been thinking about how to do more to support all of you who are looking to build a career in AI.

Considering individuals at a variety of stages in their careers, what are some of the keys to success?



1. Technical skills

When learning a new skill, taking an online course or reading a textbook — in which an expert presents important concepts into an easy-to-digest format — is one of the most efficient paths forward.

2. Practical experience

After gaining a skill, it's necessary to practice it — and learn tricks of the trade — by applying that skill to significant projects. Machine learning models that perform well in the lab can run into trouble in the real world. Practical project experience remains an important component in overcoming such problems.

3. Project selection

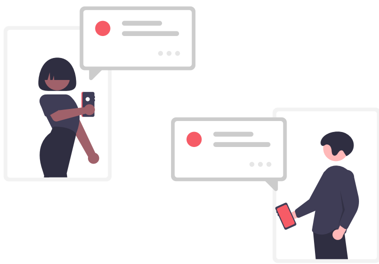
Choosing projects to work on is one of the hardest skills in AI. We can only work on so many projects at a time, and scoping ones that are both feasible and valuable — so they have a good chance of achieving meaningful success — is an important step that has to be done repeatedly in the course of a career.

4. Teamwork:

When we tackle large projects, we succeed better by working in teams than individually. The ability to collaborate with, influence, and be influenced by others is critical. This includes both interpersonal and communication skills. (I used to be a pretty bad communicator, by the way.)

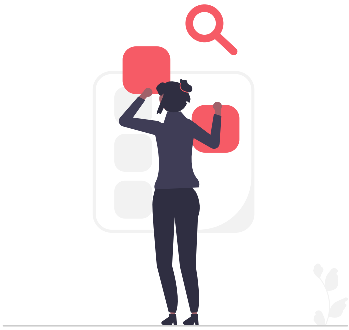
5. Networking:

I hate networking! As an introvert, having to go to a party to smile and shake as many hands as possible is an activity that borders on horrific. I'd much rather stay home and read a book. Nonetheless, I'm fortunate to have found many genuine friends in AI; people I would gladly go to bat for and who I count on as well. No person is an island, and having a strong professional network can help propel you forward in the moments when you need help or advice.



6. Job search

Of all the steps in building a career, this one tends to receive the most attention. Unfortunately, I've found a lot of bad advice about this on the internet. (For example, many articles seem to urge taking an adversarial attitude toward potential employers, which I don't think is helpful). Although it may seem like finding a job is the ultimate goal, it's just one small step in the long journey of a career.



7. Personal discipline

Few people will know if you spend your weekends learning or binge watching TV (unless you tell them on social media!), but they will notice the difference over time. Many successful people develop good habits in eating, exercise, sleep, personal relationships, work, learning, and self-care. Such habits help them move forward while staying healthy.

8. Altruism

I find that individuals who aim to lift others during every step of their own journey often achieve better outcomes for themselves. How can we help others even as we build an exciting career for ourselves?

While each of these items is a complex subject worthy of an entire book, in this eBook I try to summarize the most important points as succinctly as possible.

CHAPTER 2

Three Steps to Career Growth

Adama Diallo

AI developer at Paradigm

"I spend at least 30 minutes a day actively reading up on state-of-the-art AI models"

The rapid rise of AI has led to a rapid rise in AI jobs, and many people are building exciting careers in this field. A career is a decades-long journey, and the path is not always straightforward. Over many years, I've been privileged to see thousands of students as well as engineers in companies large and small navigate careers in AI. In this and the next few chapters, I'd like to share a few thoughts that might be useful in charting your own course.

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

1	2	3
Initially, you focus on learning foundational technical skills.	After having gained foundational skills, you lean into project work. During this period, you'll probably keep learning.	Later, you might occasionally carry out a job search. Throughout this process, you'll probably continue to learn and work on meaningful projects.

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

Learning 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.

Project work often means working 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.

What makes an AI job search unique:

While searching for a job in AI can be similar to searching for a job in other sectors, there are some 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.

Throughout these steps, a supportive community is a big help. Having a group of friends and allies who can help you — and whom you strive to help — makes the path easier. This is true whether you're taking your first steps or you've been on the journey for years.

CHAPTER 3

Learning Technical Skills For a Promising AI Career

Apala Guha

Engineer at Tenstorrent, Inc.

"Working in AI makes me feel more passionate about what I do because I feel connected to what is going on in society."

In the previous chapter, I introduced the three key steps for building a career in AI: learning technical skills, doing project work, and searching for a job, all of which is supported by being part of a community. In this chapter, I'd like to dive more deeply into the first step: learning technical skills.

More papers have been published on AI than any person can read in a lifetime. So, in your efforts to learn, it's critical to prioritize topic selection. I believe the most important topics for a technical career in machine learning are:

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.

This is a lot to learn!

Even after you master everything in this list, I hope you'll keep learning and continue to deepen your technical knowledge. I've known many machine learning engineers who benefitted from deeper skills in an application area such as natural language processing or computer vision, or in a technology area such as probabilistic graphical models or building scalable software systems.

How do you gain these skills? There's a lot of good content on the internet, and in theory reading dozens of web pages could work. But when the goal is deep understanding, reading disjointed web pages is inefficient because they tend to repeat each other, use inconsistent terminology (which slows you down), vary in quality, and leave gaps. That's why a good course — in which a body of material has been organized into a coherent and logical form — is often the most time-efficient way to master a meaningful body of knowledge. When you've absorbed the knowledge available in courses, you can switch over to research papers and other resources.

Finally, keep in mind that no one can cram everything they need to know over a weekend or even a month. Everyone I know who's great at machine learning is a lifelong learner. In fact, given how quickly our field is changing, there's little choice but to keep learning if you want to keep up. How can you maintain a steady pace of learning for years? You can cultivate the habit of learning a little bit every week, you can make significant progress with what feels like less effort.

The Best Way to Build a New Habit

One of my favorite books is BJ Fogg's, Tiny Habits: The Small Changes That Change Everything. Fogg explains that the best way to build a new habit is to start small and succeed, rather than starting too big and giving up. For example, rather than trying to exercise for 30 minutes a day, he recommends aspiring to do just one push-up, and doing it consistently.

This approach may be helpful to those of you who want to spend more time studying. If you hold yourself accountable for watching, say, 10 seconds of an educational video every day — and you do so consistently — the habit of studying daily will grow naturally. Even if you learn nothing in that 10 seconds, you're establishing the habit of studying a little every day. On some days, maybe you'll end up studying for an hour.

CHAPTER 4

Overcoming Imposter Syndrome

Matt Struble

Machine Learning Engineer

"I always felt like an outsider until I took AI courses, then I achieved things that previously felt impossible."

I'd like to address the serious matter of newcomers to AI sometimes experiencing imposter syndrome, where someone — regardless of their success in the field — wonders if they're a fraud and really belong in the AI community. I want to make sure this doesn't discourage you or anyone else.

> Let me be clear: If you want to be part of the AI community, then I welcome you with open arms. If you want to join us, you fully belong with us!

An estimated 70 percent of people experience some form of imposter syndrome at some point. Many talented people have spoken publicly about this experience, including former Facebook COO Sheryl Sandberg, U.S. first lady Michelle Obama, actor Tom Hanks, and Atlassian co-CEO Mike Cannon-Brookes. It happens in our community even among accomplished people. If you've never experienced this yourself, that's great! I hope you'll join me in encouraging and welcoming everyone who wants to join our community.

AI is technically complex, and it has its fair share of smart and highly capable people. But, of course, it is easy to forget that to become good at anything, the first step is to suck at it. If you've succeeded at sucking at AI — congratulations, you're on your way!

I once struggled to understand the math behind linear regression. I was mystified when logistic regression performed strangely on my data, and it took me days to find a bug in my implementation of a basic neural network. Today, I still find many research papers challenging to read, and just yesterday I made an obvious mistake while tuning a neural network hyperparameter (that fortunately a fellow engineer caught and fixed).

So if you, too, find parts of AI challenging, it's okay. We've all been there. I guarantee that everyone who has published a seminal AI paper struggled with similar technical challenges at some point.

Here are some things that can help.

- Do you have supportive mentors or peers? If you don't yet, attend Pie & AI or other events, use discussion boards, and work on finding some. If your mentors or manager don't support your growth, find ones who do. I'm also working on how to grow a supportive AI community and hope to make finding and giving support easier for everyone.

- No one is an expert at everything. Recognize what you do well. If what you do well is understand and explain to your friends one-tenth of the articles in The Batch, then you're on your way! Let's work on getting you to understand two-tenths of the articles.

My three-year-old daughter (who can barely count to 12) regularly tries to teach things to my one-year-old son. No matter how far along you are — if you're at least as knowledgeable as a three-year-old — you can encourage and lift up others behind you. Doing so will help you, too, as others behind you will recognize your expertise and also encourage you to keep developing. When you invite others to join the AI community, which I hope you will do, it also reduces any doubts that you are already one of us.

AI is such an important part of our world that I would like everyone who wants to be part of it to feel at home as a member of our community. Let's work together to make it happen.

CHAPTER 5

Scoping Successful AI Projects

Aleksandr Gontcharov

Software Engineer, Microsoft

"When changing your career, there’s always the fear that you’ll be surrounded by people who know more than you. But learning the right things can keep you from drowning."

One of the most important skills of an AI architect is the ability to identify ideas that are worth working on. 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 effectively.

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 a matter of 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.

Is there a domain that excites you where AI might make a difference? I hope these steps will guide you in exploring it — even if you don't yet have deep expertise in that field. AI won't solve every problem, but as a community, let's look for ways to make a positive impact wherever we can.") and making the remaining sentence ("This is an iterative process. If, at any step, you find that the current direction is infeasible, return to an earlier step and proceed with your new understanding.

CHAPTER 6

Finding Projects that Compliment Your Career Goals

Kulsoom Abdullah

Data Scientist, Anthem, Inc.

"I like to practice new concepts I learn on real-life problems."

It goes without saying that we should only work on projects that are responsible,ethical, and that benefit people. But those limits leave a large variety to choose from. In the previous chapter, I wrote about how to identify and scope AI projects. This chapter and and the next have a slightly different emphasis: picking and executing projects with an eye toward career development.

A fruitful career will include many projects, hopefully growing in scope, complexity, and impact over time. Thus, it is fine to start small. Use early projects to learn and gradually step up to bigger projects as your skills grow.

When you're starting out, don't expect others to hand great ideas or resources to you on a platter. Many people start by working on small projects in their spare time. With initial successes — even small ones — under your belt, your growing skills increase your ability to come up with better ideas, and it becomes easier to persuade others to help you step up to bigger projects.

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 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. So 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.)

Finally, avoid analysis paralysis. It doesn't make sense to spend a month deciding whether to work on a project that would take a week to complete. You'll work on multiple projects over the course of your career, so you'll have ample opportunity to refine your thinking on what's worthwhile. Given the huge number of possible AI projects, rather than the conventional "ready, aim, fire" approach, you can accelerate your progress with "ready, fire, aim."

Ready, Fire, Aim

Building AI projects requires making tough choices about what to build and how to go about it. I've heard of two styles:

- **Ready, Aim, Fire:** Plan carefully and carry out due diligence. 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, but I think 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.

CHAPTER 7

Building a Portfolio of Projects That Shows Skill Progression

Bryan Catanzaro

Vice President of Applied Deep Learning, Nvidia

"I always try to choose projects based on my own internal beliefs about where technology is going and where I think I can make the biggest difference."

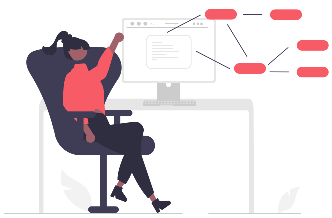
Over the course of a career, you're likely to work not on a single AI project, but on a sequence of projects that grow in scope and complexity. For example:

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.

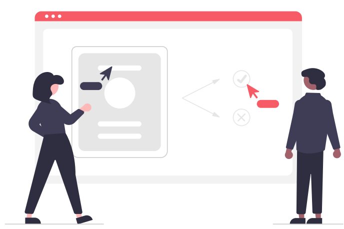


3. Creating value

Eventually, you 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.

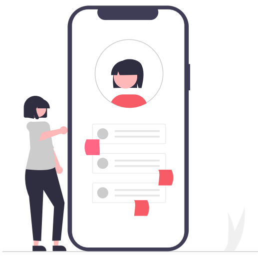


In light of this progression, when picking a project keep in mind that it is only one step on a longer journey, hopefully one that has a positive impact. In addition:

Don't worry about starting too small. One of my first machine learning research projects involved training a neural network to see how well it could mimic the sin(x) function. It wasn't very useful, but was a great learning experience that enabled me to move on to bigger projects.

Communication is key. You need to be able to explain your thinking if you want others to see the value in your work and trust you with resources that you can invest in larger projects. To get a project started, communicating the value of what you hope to build will help bring colleagues, mentors, and managers onboard — and help them point out flaws in your reasoning. After you've finished, the ability to explain clearly what you accomplished will help convince others to open the door to larger projects.

Leadership isn't just for managers. When you reach the point of working on larger AI projects that require teamwork, your ability to lead projects will become more important, whether or not you are in a formal position of leadership. Many of my friends have successfully pursued a technical rather than managerial career, and their ability to help steer a project by applying deep technical insights — for example, when to invest in a new technical architecture or collect more data of a certain type — allowed them to exert leadership that helped the project significantly.



Building a portfolio of projects, especially one that shows progress over time from simple to complex undertakings, will be a big help when it comes to looking for a job.

CHAPTER 8

Should you Learn Math to Get a Job in AI?

Kennedy Wangari

Data Scientist, United Nations Environment Programme

"Before taking any course I always consider whether it will help me achieve my career objectives."

How much math do you need to know to be a machine learning engineer? It's always nice to know more math! But there's so much to learn that, realistically, it's necessary to prioritize. Here are some thoughts about 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.

Sometimes, we're told that an idea is "foundational." While there's a lot to be said for understanding foundations, often this designation is arbitrary and thus not very useful for prioritizing what to study next. For example, computing happens on processors that are packed with transistors. Do you need a deep understanding of how transistors work to write software? It's hard to imagine an AI application where a detailed knowledge of the physics of transistors would affect your decisions.

Rather than accepting an authority's decree that a topic is foundational, it's worth asking what circumstances would require specific knowledge to help you make better decisions.
Of course, I also encourage learning driven by curiosity. If something interests you, go ahead and learn it regardless of how useful it will be in the foreseeable future. Maybe this will lead to a creative spark or technical breakthrough.

CHAPTER 9

A Simple Framework for Starting You AI Job Search

A job search has a few predictable steps including selecting companies to apply to, preparing for interviews, and finally picking a job and negotiating an offer. In this chapter, I'd like to focus on a framework that's useful for many job seekers in AI, especially those who are entering AI from a different field.

If you're considering your next job, ask yourself:

- Are you switching roles? For example, if you're a software engineer, university student, or physicist who's looking to become a machine learning engineer, that's a role switch.
- Are you switching industries? For example, if you work for a healthcare company, financial services company, or a government agency and want to work for a software company, that's a switch in industries.

A product manager at a tech startup who becomes a data scientist at the same company (or a different one) has switched roles. A marketer at a manufacturing firm who becomes a marketer in a tech company has switched industries. An analyst in a financial services company who becomes a machine learning engineer in a tech company has switched both roles and industries.

If you're looking for your first job in AI, you'll probably find switching either roles or industries easier than doing both at the same time. Let's say you're the analyst working in financial services:

- If you find a data science or machine learning job in financial services, you can continue to use your domain-specific knowledge while gaining knowledge and expertise in AI. After working in this role for a while, you'll be better positioned to switch to a tech company (if that's still your goal).
- Alternatively, if you become an analyst in a tech company, you can continue to use your skills as an analyst but apply them to a different industry. Being part of a tech company also makes it much easier to learn from colleagues about practical challenges of AI, key skills to be successful in AI, and so on.

If you're considering a role switch, a startup can be an easier place to do it than a big company. While there are exceptions, startups usually don't have enough people to do all the desired work. If you're able to help with AI tasks — even if it's not your official job — your work is likely to be appreciated. This lays the groundwork for a possible role switch without needing to leave the company. In contrast, in a big company, a rigid reward system is more likely to reward you for doing your job well (and your manager for supporting you in doing the job for which you were hired), but it's not as likely to reward contributions outside your job's scope.

After working for a while in your desired role and industry (for example, a machine learning engineer in a tech company), you'll have a good sense of the requirements for that role in that industry at a more senior level. You'll also have a network within that industry to help you along. So future job searches — if you choose to stick with the role and industry — likely will be easier.

When changing jobs, you're taking a step into the unknown, particularly if you're switching either roles or industries. One of the most underused tools for becoming more familiar with a new role and/or industry is the informational interview. I'll share more about that in the next chapter.

I'm grateful to Salwa Nur Muhammad, CEO of FourthBrain (a DeepLearning.AI affiliate), for providing some of the ideas presented in this chapter.

Overcoming Adversity

There's a lot we don't know about the future: When will we cure Alzheimer's disease? When will a Covid-19 vaccine be available? Who will win the next election? Or in a business context, how many customers will we have next year?

With so many changes going on in the world, many people are feeling stressed about the future. I have a practice that helps me regain a sense of control. Faced with uncertainty, I try to:

1. Make a list of plausible scenarios, acknowledging that I don't know which will come to pass.
2. Create a plan of action for each scenario.
3. Start executing actions that seem reasonable.
4. Review scenarios and plans periodically as the future comes into focus.

For example, during the Covid-19 pandemic back in March, I did this scenario planning exercise. I imagined quick (three months), medium (one year), and slow (two years) recoveries from Covid-19 and made plans for managing each case. These plans have helped me prioritize where I can.

The same method can apply to personal life, too. If you're not sure you'll pass an exam, get a job offer, or be granted a visa — all of which can be stressful — you can write out what you'd do in each of the likely scenarios. Thinking through the possibilities and following through on plans can help you navigate the future effectively no matter what it brings.

Bonus: With training in AI and statistics, you can calculate a probability for each scenario. I'm a fan of the Superforecasting methodology, in which the judgments of many experts are synthesized into a probability estimate. I refer to this site as a source of probability estimates as well.

CHAPTER 10

Using Informational Interviews to Find the Right Job

Lorenzo Ostano

Software Engineer, VX Fiber

"I tried to interview with several consulting companies before I landed my current job."

Last chapter, I wrote about switching roles, industries, or both as a framework for considering a job search. If you're preparing to switch roles (say, taking a job as a machine learning engineer for the first time) or industries (say, working in an AI tech company for the first time), there's a lot about your target job that you probably don't know. A technique known as informational interviewing is a great way to learn.

An informational interview involves finding someone in a company or role you'd like to know more about and informally interviewing them about their work. Such conversations are separate from searching for a job. In fact, it's helpful to interview people who hold positions that align with your interests well before you're ready to kick off a job search.

- Informational interviews are particularly relevant to AI. Because the field is evolving, many companies use job titles in inconsistent ways. In one company, data scientists might be expected mainly to analyze business data and present conclusions on a slide deck. In another, they might write and maintain production code. An informational interview can help you sort out what the AI people in a particular company actually do.

- With the rapid expansion of opportunities in AI, many people will be taking on an AI job for the first time. In this case, an informational interview can be invaluable for learning what happens and what skills are needed to do the job well. For example, you can learn what algorithms, deployment processes, and software stacks a particular company uses. You may be surprised — if you're not already familiar with the data-centric AI movement — to learn how much time most machine learning engineers spend iteratively cleaning datasets.

Prepare for informational interviews by researching the interviewee and company in advance, so you can arrive with thoughtful questions. You might ask:

- What do you do in a typical week or day?
- What are the most important tasks in this role?
- What skills are most important for success?
- How does your team work together to accomplish its goals?
- What is the hiring process?
- Considering candidates who stood out in the past, what enabled them to shine?

Finding someone to interview isn't always easy, but many people who are in senior positions today received help when they were new from those who had entered the field ahead of them, and many are eager to pay it forward. If you can reach out to someone who's already in your network — perhaps a friend who made the transition ahead of you or someone who attended the same school as you — that's great! Meetups such as Pie & AI can also help you build your network.

Finally, be polite and professional, and thank the people you've interviewed. And when you get a chance, please pay it forward as well and help someone coming up after you. If you receive a request for an informational interview from someone in the DeepLearning.AI community, I hope you'll lean in to help them take a step up! If you're interested in learning more about informational interviews, I recommend this article from the UC Berkeley Career Center.

I've mentioned a few times the importance of your network and community. People you've met, beyond providing valuable information, can play an invaluable role by referring you to potential employers.

CHAPTER 11

Finding the Right AI Job For You

In this chapter, I'd like to discuss some fine points of finding a job.

The typical job search follows a fairly predictable path.

- Research roles and companies online or by talking to friends.
- Optionally, arrange informal informational interviews with people in companies that appeal to you.
- Either apply directly or, if you can, get a referral from someone on the inside.
- Interview with companies that give you an invitation
- Receive one or more offers and pick one. Or, if you don't receive an offer, ask for feedback from the interviewers, human resources staff, online discussion boards, or anyone in your network who can help you plot your next move.

Although the process may be familiar, every job search is different. Here are some tips to increase the odds you'll find a position that supports your thriving and enables you to keep growing.

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 rise 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.

CHAPTER 12

Pathways for AI Careers

Companies in many industries are building AI teams, but it may not be obvious to people early in their career how to join one of them.

Different companies organize their teams differently and use different terms to describe the same job. Even more confusing, job titles don't correspond directly with common AI tasks like modeling and data engineering.
What positions are responsible for which tasks? What skills are recruiters looking for? Which opportunities are right for you?

Workera, a DeepLearning.AI affiliate, interviewed over 100 leaders in machine learning and data science to answer these questions. They summarized their findings in a report called "AI Career Pathways: Put Yourself on the Right Track."

> *"AI Career Pathways" is designed to guide aspiring AI engineers in finding jobs and building a career. You can read a summary of the report's findings below. You can also download it here*

The report defines two basic types of AI organizations.

- **Data science organizations** use AI to help a firm's leaders make scientific or data-driven decisions so they can run their business more effectively. Team members collect data, analyze datasets, and suggest hypotheses and actions.

- **Machine learning organizations** use AI to automate tasks, reduce costs, or scale products. The output is the automation itself, and is achieved by collecting data, training models, and deploying them.

The report identified six basic machine learning roles that exist at companies that perform machine learning. These are:

- **Data Scientists:** Determine the data needs of an ML project, collect the data, and ensure that it is in the correct format for analysis. Data scientists also play supporting roles in deploying models and building software infrastructure. Their skills include: Proficiency using data engineering packages in Python, PyTorch, and TensorFlow. Using tools like R, Tableau, and Excel for business analysis. They should also be familiar with version control tools like Git, Subversion, or Mercurial.

- **Machine Learning Engineers:** Train and deploy machine learning or deep learning models. They draw upon an understanding of different model architectures and proficiencies like tuning hyperparameters. Their skills include: Using query languages like Python or SQL to develop and deploy models. Familiarity with cloud services such as AWS, GCP, and Microsoft Azure. Version control and collaboration tools including Git, Jupyter Notebooks, and Jira.

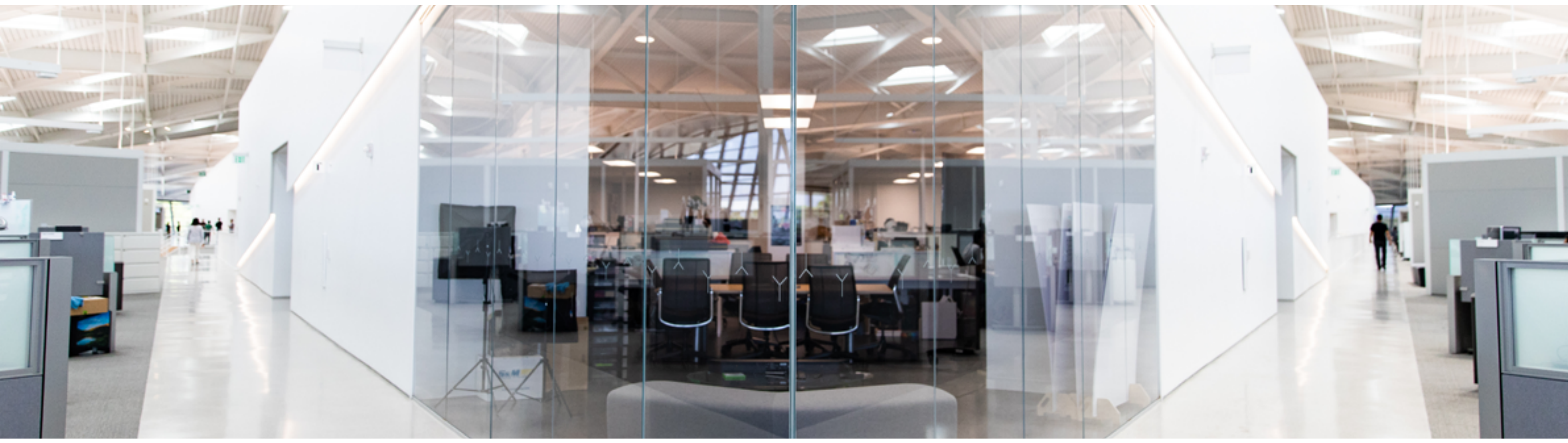
- **Data Analysts:** Analyze the output and performance of machine learning models using a combination of analytical skills and business acumen. They primarily use spreadsheets and query languages, and may not be required to know algorithmic coding. Skills include: Python or SQL for data engineering and business analysis. Tableau and/or Excel for data analysis and visualization. External software for A/B testing.

- **Software Engineer-ML:** This hybrid role performs many of the same tasks as data scientists, machine learning engineers, and data analysts, often acting as a one-person machine learning team. They are common at startups, or in larger companies where they work on improving machine learning systems that are already in production. Skills include: Python, PyTorch, TensorFlow for data engineering and model development. Familiarity with cloud services such as AWS, GCP, and Microsoft Azure. Version control and collaboration tools including Git, Jupyter Notebooks, and Jira.

- **Machine Learning Researcher:** As their name suggests, these specialists research new machine learning architectures and applications. They focus primarily on data engineering and model development, but are supported by teams in charge of deployment, business analysis, and AI infrastructure. Skills include: Using query languages like Python or SQL to develop and deploy models. Familiarity with cloud services such as AWS, GCP, and Microsoft Azure. Version control and collaboration tools including Git, Jupyter Notebooks, and Jira.

- **Software Engineer:** Carry out data engineering and AI infrastructure tasks, including building and maintaining fast, secure, and scalable software systems for all steps of the machine learning pipeline. They should be able to write production code, understand cloud technologies, and build distributed storage and database systems. Skills include: Skills include: Python and SQL for data engineering. Python, Java, C++ or other object oriented programming languages for software development. Familiarity with cloud services such as AWS, GCP, and Microsoft Azure. Version control and collaboration tools including Git, Jupyter Notebooks, and Jira.

It is important to note that most companies do not define these roles in exactly the same way, and there is often overlap between them. For example, data scientists often train and deploy models and machine learning engineers are often called upon to maintain AI infrastructure.



Conclusion

Make Every Day Count

Every year on my birthday I get to thinking about the days behind and those that may lie ahead.

Maybe you're good at math; I'm sure you'll be able to answer the following question via a quick calculation. But let me ask you a question, and please answer from your gut, without calculating.

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?