AdaptiLab

Complete Guide to

Hiring for Machine Learning Roles

Introduction

Hiring is hard. From generalist software engineers to systems architects, there are very few roles where hiring is a walk in the park. This is especially true for roles in machine learning, which is one of the fastest-growing and most in-demand fields in today's industry. Whether you're hiring your company's first ML employee, or expanding your ML team from 2 to 20, it is going to be a difficult task with many open-ended questions. In this **Guide to Hiring for Machine Learning Roles**, we'll be answering many of the questions you might have, as well as providing tips and direction on the best ways to build out and optimize your machine learning hiring process.

Table of Contents

Identifying the Type of Role You’re Trying to Fill

- ## Understand what you’re trying to achieve with machine learning

Like software engineering, machine learning is an incredibly broad field with many different subdomains. For the most part, what you want to accomplish with machine learning really depends on your overall business application and the type of data you work with. If you work with text datasets and your business application involves extracting useful information from the text data, then you would require natural language processing (NLP) specialists. On the other hand, if your business application is to categorize objects based on different images, you would require computer vision (CV) specialists.

In general, most use cases of machine learning in industry aren’t too specialized. A common use case is to train standard feed-forward neural networks on large tabular datasets (e.g. CSV or Excel spreadsheets). The type of role required for this use case would involve a mix of general machine learning conceptual understanding and solid software engineering fundamentals.

Figure out exactly what you're looking for in machine learning roles

After taking a high-level look at your overall machine learning needs, it's time to figure out what you want to fill in specific roles, and how many different roles you're looking for. If you're building an ML team from the ground up, you likely want to hire more experienced people who can help build out the rest of the team and act as a lead. However, people with significant industry experience in ML are difficult to find and expensive to hire, so make sure that you **absolutely need** that type of person before trying to hire them. Many companies make the mistake of targeting highly experienced candidates when in reality their ML initiatives are relatively simple.

Furthermore, many industry initiatives involving ML are still heavily focused around software engineering. Tasks such as large-scale data processing, setting up model training pipelines, and deploying models to production require both machine learning conceptual knowledge and strong engineering ability. When hiring for ML roles, it is important that you remember **the engineering aspects** of the job as well. It is all too common in industry for companies to hire a candidate based solely on the fact that they have a PhD in ML, only for that candidate to come in and not know how to write any quality production-level code.

Creating the Right Job Description

- ## Clearly state the job requirements and logistics

It is not an exaggeration to say that many job descriptions currently published online, whether they're on LinkedIn, Indeed, AngelList, etc., are absolutely horrific. It is common for software engineering job descriptions to puff up the requirements in the role (in an attempt to weed out inexperienced candidates) and ultimately end up requiring expertise in 5 different web frameworks and 8 different database engines.

Since machine learning is a relatively novel field in industry, the job descriptions for ML hiring can end up being ridiculous. There are job descriptions out there that require 5+ years of experience using TensorFlow in industry applications, when the actual open-source version of TensorFlow was released fewer than 5 years ago. It is hard enough to attract talented candidates in the ML space, and having an inaccurate or laughable job description will only make things worse.

The best plan of attack when making a job description is to **focus on the skills necessary to succeed in the role**, rather than knowledge or expertise in a particular technology. Give a high-level description of the type of work and projects that come with the role. Make the job description as accurate as possible, so that candidates will not be surprised with what's expected of them when they actually start the job. Employee churn is a major issue in software engineering and machine learning, in large part due to the lack of accurate job descriptions.

Finally, make sure your job description is clear about the logistics and requirements of the position. For example, there are many incredibly talented candidates out there who would do very well in the interview process, but who require visa sponsorship to work in the US. If your company is unable to provide visa sponsorship, stating that directly in the job description will save both your time and the candidate's time. Being clear on other aspects of the position, such as compensation range and relocation requirements, will also help to avoid any potential misunderstandings and wasted time.

- ### Highlight interesting aspects of your company and the role

Apart from looking for well-written and accurate job requirements, candidates also look for any interesting tidbits in a job description. This varies on a candidate-to-candidate basis, since candidates all have different values. The best way to attract candidates who are likely to have a strong fit with your company is to highlight any **unique** or **interesting aspects** of your company or the role you're specifically looking to fill. For example, if your company is fully remote, that's something you would want to highlight to attract candidates who love remote work. You could also highlight lighter aspects of the company, such as dog-friendly offices, lakefront views, or even well-known investors.

For the role itself, it’s wise to highlight any unique or state-of-the-art technology that’s being worked on. Talented engineering and machine learning candidates absolutely love working on the most cutting-edge technologies, so it would be a good selling point and could even make up for potential negatives in location or compensation.

Finding and Attracting Talented Candidates

- ### Establish company brand in machine learning

Companies that attract many talented candidates have already established a brand that engineers gravitate towards. Dropbox, Quora, and MemSQL were known as companies with extremely difficult interview processes, which allowed them to attract smart engineers looking for a challenge. This obviously would not work for every company, since having a difficult interview process means turning away many solid engineers. However, the concept around the idea is sound: create a brand around your company or organization within the company that can attract top talent.

The research divisions of top tech companies like Google, Microsoft, and Facebook have built their brand around releasing widely-used open source packages and beating state-of-the-art results in several machine learning fields.

Uber and Cruise built brands around autonomous vehicles, one of the hottest and coolest applications of computer vision in recent years. If you have an interesting application of machine learning or technologically advanced model architectures, use that to build a **machine learning brand**. If your applications of machine learning aren't necessarily the most fascinating, it's useful to leverage conferences, speaking engagements, and online blogging to make your machine learning use cases seem more interesting for potential candidates. Sponsoring online competitions and hosting hackathons are also great ways to build brand, as showcased by Walmart Labs and Netflix.

- ### Build an interview process that candidates don't hate

Most people really don't enjoy interviewing. It can be stressful, irritating, time-consuming, and ultimately lead to stacks of rejection emails. Some candidates will go through terrible interview processes if they don't have any better options, but most talented or experienced candidates will not go through an interview process that seems dull or overly time-consuming.

When it comes to machine learning, this is especially true. There are far fewer talented machine learning candidates than software engineers, so they can afford to be pickier with interview processes. It is normal for these candidates to drop off the interview process after the first technical phone call if they find the interview to be irrelevant to actual machine learning skills. To these candidates, if a company can’t even properly gauge ML in an interview, then it’s likely that their internal ML teams are not very sophisticated.

Evaluating Technical Ability

- ## Assess relevant machine learning skills

For many engineers, interviewing is a chore that takes away precious time from their actual job. While it would behoove engineers to be thoughtful and thorough during interviews, given that they are effectively choosing their future teammates, this is often **not** the case. It is common for engineers to find a question on sites like Leetcode or GeeksforGeeks and judge a candidate based on how closely their answer matches the online solution.

These types of questions can be adequate for evaluating general software engineers, where the focal point of interviews is assessing a candidate’s problem solving and algorithmic ability. However, when hiring for machine learning positions it is vital to **actually test for relevant machine learning skills**, and to test for these skills early on in the interview process. If a candidate’s machine learning skills are not up to par for the position, it is better to figure this out as soon as possible, in order to save interviewing time.

Don’t forget about the code

Recruiters and hiring managers will often become entranced by candidates who have a PhD in machine learning. A PhD necessarily means a candidate has gone through years of research and studying in the field of machine learning, so it would make sense that their skills in machine learning are top-notch. However, while it is true that candidates with a PhD have advanced *conceptual knowledge* of machine learning, this does not mean that they can produce industry-standard code for machine learning projects. In fact, it is often the case that candidates who have just finished their PhD are rusty in their actual coding ability, given that they’ve spent the past few years focusing on writing research papers and fine-tuning models.

Certain machine learning positions, such as research scientists, are more focused on the theoretical aspects of machine learning and do not require much coding outside of creating model architectures. However, most machine learning engineer positions actually focus on the data pipelining and deployment cycles of machine learning processes, which require a significantly greater amount of coding and engineering ability. For these positions, just having a PhD in ML will not be enough if the candidate cannot write efficient and structured code. Therefore, it is **crucial** to test a candidate's ability to complete code-based tasks such as creating a data pipeline. The best way to test this is to have candidates write code for these tasks, similar to what they'd be doing on the job, and then to run the code on actual datasets and models. This not only provides a better way to evaluate candidates, but also provides candidates with a better understanding of what's to be expected of them for the given role.

- ### Assess managerial ability, if necessary

If you're in the situation of hiring someone who will end up building out and leading the rest of the machine learning team, you'll need to make sure the person you hire is able to take on those managerial responsibilities. If this person is one of the first machine learning hires made at the company, they should have a large role in molding the company's machine learning foundation. This means that they should have ample experience and knowledge in leading new machine learning projects, as well as hiring for new machine learning roles.

Evaluating Company Fit

- ## Make sure candidates can work well with you and the team

Strong technical ability is important in candidates, but just as important is how well a candidate can mesh with the rest of the team. You want to make sure that the person you're hiring is someone who can cooperate well with others, incorporate feedback to improve, and discuss ideas cordially. An evaluation strategy that has become increasingly popular is having candidates partake in a "trial period," where they get paid to work with the company for some period of time (e.g. a week or so) before receiving an offer. This allows both the candidate and company to assess whether the fit is right.

Of course, not every company is able to do the "trial period," given time and resource constraints. An alternative is to have onsite interviews focus more on a candidate's intangibles rather than continued technical and coding assessments. Companies that prefer this route usually will start their interview process with take home projects or automated technical challenges, to make up for the less technically focused onsite.

Better company fit means lower possibility of churn

While hiring good candidates is difficult for most companies, a problem just as irritating is employee churn. While there is churn that cannot be avoided or anticipated, a lot of it is simply due to a lack of company fit. Make sure that the candidate is given a clear picture of what to expect on the job while they’re going through the interview process. A common mistake that recruiters or hiring managers make is to butter up some aspects of the work or company to make the candidate more likely to accept an offer. While this may help persuade candidates to join the company, it will increase the chances of churn if their expectations for the job are not met.

The best course of action is to **be honest** with the candidate while still trying to sell them on the company’s highlights. It is completely fine to pass on a talented candidate if you’re unsure whether they’re a good fit for the company. It is much better to take a bit longer in the hiring process to find the right person for the job than to hire someone earlier who will just leave the company after a year.

Conclusion

While hiring for many roles is difficult, it is particularly challenging to hire for machine learning roles given the novelty and rapid growth of the field. In this **Guide to Hiring for Machine Learning Roles**, we've provided tips and guidelines for best practices in creating job descriptions and evaluating candidates. We hope that after going through this guide you've gained the necessary knowledge to improve your own machine learning hiring processes, which will ultimately enhance the overall machine learning capability at your company.

About the Author

Allen Lu, Co-Founder & **CTO of AdaptiLab**

Allen is a cofounder and CTO at AdaptiLab. He spent time working on machine learning teams at Google and Microsoft, where he experienced both sides of the hiring process. He has helped several companies improve their machine learning and data science hiring processes, and hopes that he can continue doing so with AdaptiLab.

About AdaptiLab

AdaptiLab is a Seattle-based startup that graduated from the Techstars accelerator and is backed by Trilogy Equity Partners. AdaptiLab has helped several companies, from large corporations like Experian to fast-growing startups like Highspot, optimize and improve their machine learning and data science hiring processes.

If you are interested in learning how AdaptiLab can **help you with your machine learning and data science hiring**, click below to request a demo!

Request a Demo