**MS or Startup Job — Which way to go to build a career in Deep Learning?**

**读硕士还是工作-哪条路更适合深度学习？**

Today I met two college friends. Both of them currently work as software engineers in one of the most successful technology companies of India. They have been reading about Deep Learning from popular online sources like [cs231n.stanford.edu](http://cs231n.stanford.edu/), [cs224d.stanford.edu](http://cs224d.stanford.edu/), [3Blue1Brown](https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AJtAw) and [Siraj Raval](https://www.youtube.com/channel/UCWN3xxRkmTPmbKwht9FuE5A) for quite some time and they regularly implement CNNs and RNNs in TensorFlow and PyTorch for practice. Fascinated by the potential and market-hype, now they are looking for an opportunity to pursue Deep Learning full-time and they wanted to discuss what should be the right entry-point — is it a job in one of the startups or is it pursuing an MS degree.

今天，我碰见了两个大学时的朋友。他们现在都是软件工程师，工作在印度最成功的技术公司之一。他们已经从在线资源，像[cs231n.stanford.edu](http://cs231n.stanford.edu/), [cs224d.stanford.edu](http://cs224d.stanford.edu/), [3Blue1Brown](https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AJtAw) 和 [Siraj Raval](https://www.youtube.com/channel/UCWN3xxRkmTPmbKwht9FuE5A) 学习机器学习一段时间了。作为练习，他们经常用TensorFlow和PyTorch实现CNNs和RNNs。因为对人工智能潜力和概念炒作的心驰神往，他们现在想全身心的投入深度学习里，但他们想知道正确的切入点是什么-工作还是去读一个研究生学位。

Numerous people ask me this question every other day. Unfortunately, I could never give a structured answer, until today. Today’s answer was a result of an extensive discussion of possibilities with these close friends of mine who are extremely intelligent and I am proud that we could come up with such a comprehensive answer. I invite you all, the readers of this article, to express your opinions in the comments section below so that we can refine the answer together.

基本上，每隔几天就有人问我这个问题。遗憾的是，直到今天，我本人也不能给出一个建设性的答案。今天的答案是和我的亲密朋友们对各种可能性广泛讨论的结果，他们都是极其睿智的人。我很高兴大家一起提出这个全面的答案。我也诚邀本文的读者，积极地表达你的观点，好让我们一起改善答案。

In today’s scenario, there are two broad streams of activity in the field of Deep Learning. It is pretty much the same as that in any other field of scientific research.

Stream-1: Implementing the state-of-the-art theory in vertical applications.

第一类：将最先进的理论应用到实践中

Stream-2: Pushing the theoretical state-of-the-art.

第二类：推动理论的前瞻性发展

Both of these streams are equally rewarding but they call for two distinct sets of skills/instincts. Let’s call them S¹ and S². (Disclaimer: S¹ ∩ S² ≠ ∅). Depending on whether you have S¹ or S², you fall in Category-1 or Category-2.

Next we make non-exhaustive but sufficiently comprehensive enumerations of S¹ and S².

Let’s start with S¹. We are talking about Stream-1 which is implementing the state-of-the-art theory in vertical applications. The goal is to solve a problem at hand with the state-of-the-art Deep Learning algorithm. Say for example, the problem at hand is to build a conversational AI agent to help the millions of people who suffer from depression and related mental-health issues. You do a quick literature survey and expert consultation and locate the state-of-the-art approach to making such an AI that was maybe published in the most recent iteration of NIPS. With this paper in hand, you come back to the original idea of your system and analyze the paper in the light of your problem statement. You ask the following questions:

How much does this paper cater to the problem statement you are trying to solve?

What is the scale of deployment of your system? Is the solution proposed in the paper scalable enough?

What is the maximum allowable latency of your system? If you are handling personal/private data, its recommended that you do not ship any of that data to the cloud and your algorithm learns and infers in the user’s device. Is the solution in the paper suitable for handling such a case? If not, can you make it suitable?

Add to this list the N number of deployment issues that are relevant to the domain of your application. Your job is to focus on implementing, optimizing and adapting the NIPS algorithm for your application. For this you need the following non-exhaustive set of skills which we collectively call S¹.

Superlative programming skills and preferably, a background in software engineering.

The zeal of optimizing an algorithm to the core.

Knowledge of different hardware platforms — like CPU (x86 vs ARM), GPU, FPGA, VPU/ASIC etc and how to write optimum code for each of these. Also, the ability to judge the suitability of a given platform for your application.

Knowledge of High Performance Computing and Big Data handling.

Prior experience in full-stack software development.