

Plan for Collaborations - Fall 2018

This write up is pretty much inspired by a bunch of things. Firstly, [Long term oriented thinking by Jeff Bezos](#): I believe it is very practical to have a delayed gratification for a sustained long term growth at the cost of short term inconvenience.

In the context of research, it translates to planning at least a couple of years ahead and not just worry about the task at hand. It helps us strategize, spend our time better and look around the corners more intelligently. At the other end of spectrum, one of the problems with academics is that there is a lot of planning but little implementation. I personally have experienced this too. This might lead to overachievers' dilemma and [imposter syndrome](#). To address the same, this writeup is also based on the [rapid experimentation approach by Marissa Mayer](#). In any such endeavours, it is imperative to expect failure and shortcomings. But I think it is important to have [the right kind of failure](#). Finally, let's scope this write up to a small goal. For this writeup, it is going to be deep learning, because it seems to stay relevant a decade from now.

I subscribe to the notion that going forward, success in deep learning will be dependent on the [trinity](#) of data, algorithms and infrastructure. This is an easy problem to solve for the industry. But what about academia? In this write up I will try to pen down some of the ideas I have with respect to the 'algorithms' leg.

Algorithms: Let's start with algorithms because I believe this is the easiest component of trinity for academicians. Definitely seems so at the surface. It appears that the best way to build algorithmic prowess is to keep working on them. In a university setting, this is very practical since we usually perform exhaustive experimentation and vet our hypotheses. But something tells me that this is not enough. Specifically when it comes to scaling and testing, industry clearly has an upper hand today. Competing in this aspect is a zero sum game. However, this does not imply we give up.

A viable approach to handle this is to break it down into two chunks: (1) Develop algorithmic prowess and (2) Scale up the application of the proposed algorithms. With this breakup, we can focus on the individual components more clearly. Developing algorithms can be viewed as a more serious task requiring a lot of focus while there are fun ways to innovate in the scaling up counterpart. One way I see accomplishing (1) is by participating in numerous challenges being organized today. For example, in the current semester, I have participated and built around 90% of our submissions to 5 [speech related challenges](#). (Yes it was incredibly hard.) Similarly one way I see accomplishing (2) is by collaborating. Important thing is to do this organically. We need to collaborate with the objective of (a) strengthening the areas where we are weak and (b) fostering the collaboration medium to long term. For example in speech, we are very strong in Automatic Speech Recognition and Text to Speech Synthesis. Moreover, these domains are sufficiently complex with a non trivial amount of entry level knowledge requirement. On the other hand, we do not have exposure in areas such as voice conversion, language/speaker

recognition, etc. These areas also seem cool to beginners. Therefore, I think the biggest ROI(for both parties involved) is in collaborating in these domains and not in the core areas.

I have done a variant of this in current semester's offering of 11-791 by proposing as task human emotion recognition from speech. We had 90% people doing this project. This was pretty surprising to me as I had not anticipated such huge participation. However, I do not think this is without reason: CMU is blessed with some of the smartest people around the planet. Also, most of them are from a different country. Therefore, they in general take risks. In other words, when faced with the scenario of choosing a safe path vs an adventurous one, students typically chose the latter. However, I believe that there is a third component to algorithms leg in the trinity. It has to do with rapid experimentation. To truly accomplish fruitful results, we need to have a strong infrastructure that helps us experiment freely. For example, I have created proxy accounts for 11-791 in three of my clusters so that we can experiment without break. This leads us to the second leg in the trinity: infrastructure. I will talk about this in a different post.