

Build your first system quickly, then iterate

(SPEECH)

If

(DESCRIPTION)

Logo, Deep Learning.ai. Error Analysis, Build your first system quickly, then iterate

(SPEECH)

you're working on a brand new machine learning application, one of the piece of advice I often give people is that, I think you should build your first system quickly and then iterate.

Let me show you what

(DESCRIPTION)

Speech Recognition Example. Noisy background, cafe noise, car noise, accented speech, far from microphone, young children's speech, stuttering

(SPEECH)

I mean. I've worked on speech recognition for many years.

And if you're thinking of building a new speech recognition system, there's actually a lot of directions you could go and a lot of things you could prioritize.

For example, there are specific techniques for making speech recognition systems more robust to noisy background.

And noisy background could mean cafe noise, like a lot of people talking in the background or car noise, the sounds of cars and highways or other types of noise.

There are ways to make a speech recognition system more robust to accented speech.

There are specific problems associated with speakers that are far from the microphone, this is called far-field speech recognition.

Young children speech poses special challenges, both in terms of how they pronounce individual words as well as their choice of words and the vocabulary they tend to use.

And if sometimes the speaker stutters or if they use nonsensical phrases like oh, ah, um, there are different choices and different techniques for making the transcript that you output, still read more fluently.

So, there are these and many other things you could do to improve a speech recognition system.

And more generally, for almost any machine learning application, there could be 50 different directions you could go in and each of these directions is reasonable and would make your system better.

But the challenge is, how do you pick which of these to focus on.

And even though I've worked in speech recognition for many years, if I'm building a new system for a new application domain, I would still find it maybe a little bit difficult to pick without spending some time thinking about the problem.

So what we recommend you do, if you're starting on building a brand new machine learning application, is to build your first system quickly and then iterate.

(DESCRIPTION)

Set up dev test set and metric

(SPEECH)

What I mean by that is I recommend that you first quickly set up a dev/test set and metric.

So this is really deciding where to place your target.

And if you get it wrong, you can always move it later, but just set up a target

(DESCRIPTION)

Build initial system quickly

(SPEECH)

somewhere.

And then I recommend you build an initial machine learning system quickly.

Find the training set, train it and see.

Start to see and understand how well you're doing against your dev/test set and your values and metric.

When

(DESCRIPTION)

Use bias variance analysis and error analysis to prioritize next steps

(SPEECH)

you build your initial system, you then be able to use bias/variance analysis which we talked about earlier as well as error analysis which we talked about just in the last several videos, to prioritize the next steps.

In particular, if error analysis causes you to realize that a lot of the errors are from the speaker being very far from the microphone, which causes special challenges to speech recognition, then that will give you a good reason to focus on techniques to address this called far-field speech recognition which basically means handling when the speaker is very far from the microphone.

Of all the value of building this initial system, it can be a quick and dirty implementation, you know, don't overthink it, but all the value of the initial system is having some learned system, having some trained system allows you to localize bias/variance, to try to prioritize what to do next, allows you to do error analysis, look at some mistakes, to figure out all the different directions you can go in, which ones are actually the most worthwhile.

(DESCRIPTION)

Guideline, build your first system quickly, then iterate

(SPEECH)

So to recap, what I recommend you do is build your first system quickly, then iterate.

This advice applies less strongly if you're working on an application area in which you have significant prior experience.

It also implies to build less strongly if there's a significant body of academic literature that you can draw on for pretty much the exact same problem you're building.

So, for example, there's a large academic literature on face recognition.

And if you're trying to build a face recognizer, it might be okay to build a more complex system from the get-go by building on this large body of academic literature.

But if you are tackling a new problem for the first time, then I would encourage you to really not overthink or not make your first system too complicated.

Well, just build something quick and dirty and then use that to help you prioritize how to improve your system.

So I've seen a lot of machine learning projects and I've seen some teams overthink the solution and build something too complicated.

I've also seen some teams underthink and then build something maybe too simple.

Well on average, I've seen a lot more teams overthink and build something too complicated.

And I've seen teams build something too simple.

So I hope this helps, and if you are applying to your machine learning algorithms to a new application, and if your main goal is to build something that works, as opposed to if your main goal is to invent a new machine learning algorithm which is a different goal, then your main goal is to get something that works really well.

I'd encourage you to build something quick and dirty.

Use that to do bias/variance analysis, use that to do error analysis and use the results of those analysis to help you prioritize where to go next.