NIPS 2017 **Workshop: Deep Learning Theory, bridging theory and practice (12/09/17)**

Kiran Vodrahalli

These are notes from the panel at the end of the workshop. **Sanjeev Arora** is the moderator and asks the questions. The panelists are **Sham Kakade, Percy Liang, Peter Bartlett, Yoshua Bengio, and Ruslan Salakhutdinov**.

1 What aspect of deep learning can theory make a big impact in the next few years?

Sham Kakade: Model-based solution concepts.

Percy Liang: Inductive bias. You get all these complicated models with attention etc.

Peter Bartlett: Interaction between optimization and statistical properties, are there principled ways of doing this?

Yoshua Bengio: One place I'm interested in is how do we formalize questions about uniqueness of representations. A couple of papers exist where under some conditions, you can get unique solutions to the problem of nonlinear independent component analysis. Here we are trying to look at factorized distributions, under mild assumptions you can come up with unique solutions. I suppose this is far from statistical and optimization questions.

Sanjeev Arora: You mean casese where non-convex problems behave like convex ones?

Yoshua: More subtle than that. Up to say linear transformations, you get uniqueness.

Ruslan Salakhutdinov: Understanding optimization and generalization is most interesting – basically what Peter said.

2 Is adversarial examples just a puzzle or is it fundamental?

Sham: There was a workshop at Stanford talking about adversarial learning in RL, one of the difficulties is there is a sense where errors in RL always look like worst case errors. We don't understand the behavior of this. The concern with RL is that making progress on robustness may be closely related to adversarial learning, no understanding or very little understanding of how errors in RL compound. I kind of agree it's important.

Percy: At first I thought it was cute, but I think over the years it triggered other examples. Understanding how to adversarially attack is related to interpretability, and models which generalize to new distributions (distribution-robust learning). Adversarial examples are like a stress test.

Peter: Defending against adversarial attacks in a critical setting is hugely important. I don't know if you should be worried about reverse engineering with adversarial attacks, it might be that you can always do that.

Yoshua: I think adversarial examples are not specific to neural networks. The intuition from the beginning has a lot to do with the fact that humans use extra info about the distribution of images, which influences our classifiers. Humans only learn the joint distribution of x and y. We can use adversarial examples as a kind of regularization for unsupervised

learning. There are no convincing strategies of combining these methods with classifiers; this has been somewhat forgotten, would like to see work in this area come back.

Russ: Yes, learning $p(x)$ is great, I basically agree with Yoshua. It is hard to build good generative models and evaluate densities. If we could build density estimators, adversarial attacks would not work as well.

Yoshua: Small perturbations should then change the category.

Percy: What about attacking generative models? If you try to estimate a generative model with a ton of parameters, you can still attack it.

Sanjeev: Let me interupt there, this anticipates my next question.

3 Are generative models the correct approach to unsupervised learning? Are there other ways?

Sanjeev: For instance, why shouldn't there be problems there? You aren't getting the distribution.

Sham: Density estimation might be able to do well, but the stuff you care about is in the last bit of error. You might miss fine-grained things and basically show that some concepts you get wrong most of the time. Your model might be wrong in most senses, things you care about may have little weight until you learn the distribution completely.

Percy: In some sense this is a way forward. If you treat it as density estimation, the model tries to explain local details. On the other hand, if you set the objective function and then don't care about it, that has issues too. Your goal is to explain high-level things: If you know what the high-level things you care about are, you can set the objective function accordingly. If you want to learn what they are, you don't know how to set the objective function. It's like a chicken-and-egg problem.

Peter: The central issue is what is the appropriate objective. One direction might be: Other kinds of interaction can be used to pin that down, essentially re-writing the problem.

Yoshua: I agree with both of you, it's important to think about other training objectives centered on the representation level. I have this notion I've been talking about called the consciousness prior. For instance, if you try to model acoustics, it's still hard to model speech: There are only a few relevant speech sounds. Information-theoretic objectives make sense. We have all the ideas already; think of PCA and autoencoders, minimize reconstruction, maximize independence. It's possible to define training objectives in the representation space. RL is there to help find out what matters to the objective.

Russ: I believe in generative models. But unsupervised learning is ill-defined problem. I see research in RL; this is taking things in an interactive direction. This is where the field will move, in settings where you can interact with the environment.

Yoshua: It's more convenient to predict the future in the representation space, not pixel space.

Sham: For language models, I believe in generative models there cause you want to generate language. But we still miss concepts and things also depend on the specific objective

function.

Russ: How do you define the space you're working in though? Computer vision friends a while ago used to say that dealing with pixels is a disaster (this is before deep nets). They said you have to work in a higher space, but at the end of the day, the input is just pixels.

Percy: There's a difference between the generation task and generative modeling. From an RL perspective, if you care about max reward or something you can just go to the mode of the distribution. The other purpose is to have methods for using generative models to learn representations.

4 Does RL need deep learning? (Excluding using deep nets for sensing, say, images)

Sham: That's a great question. I'm not convinced in the vanilla control setting. So maybe not. I'm not convinced current methods are doing anything extra. Logic and so on is more difficult. I wonder if people think there is a current impressive demo where we are truly seeing the fruits of using RL in deep learning. I think the deep learning part is just better representations of the environment.

Percy: Long term you might need it, where you do sensing, but also need to hold a large memory about the world. Then the policy may be complicated. The complexity might necessitate using deep nets.

Peter: It'd be surprising to me if you could solve very complex control problems without having rich nonparametric function classes. There are of course examples where you can get by with linear systems, but it'd be surprising if you couldn't get something out of them also.

Russ: I guess I agree, deep learning is going to go past sensors. You can have a parametrized policy with a deep net, architectures with memory, how to read and write with attention. I suspect we will go beyond just using deep learning for representations.

Percy: We need to have baselines. Some control policy which is very simple (for instance, for language) using some inductive bias is much better than a joint network. We need to be careful how we measure success.

5 Generalization theory got popular recently. Is it just a puzzle or a fundamental insight?

Sanjeev: Practically, we could just have a held-out set. What better insight do we get by giving generalization bounds?

Peter: There's more to understand I think. The part I think is most interesting is the interaction between optimization and generalization. If we could have better measures of complexity of deep nets and have lower bounds that would be great. There are implications for regularization. For instance why is SGD effective in finding solutions which generalize well. This could also lead to more effective algorithms, particularly for noisy cases.

Percy: It would be interesting to understand that story. I'm trying to see how it would change my behavior in the sense that there's approximation error and estimation error. There's an estimation error for how you choose inductive biases for your problem. We have collapsed optimization error and estimation error together into something more end-to-end regarding estimating functions and generalizing at the same time.

Russ: I think the current state of affairs is very frustrating: SGD with momentum plus batch norm plus dropout is very hard to beat. This is essentially a bag of tricks.

Sham: Generalization theory is a proxy for thinking about algorithms which generalize. The hope is the theory of generalization can enlighten algorithms which generalize. On any fixed problem it's hard to beat SGD, but we want to identify a richer class of problems where we want to generalize between problems. This can be thought of as multi-task learning. There are many ways you can drive the error down to zero. Some algorithms may not work on other problems, there's a lot to do here.

Sanjeev: Generalization can also be construed more broadly, in terms of transfer learning, domain adaptation. Thoughts?

Percy: Notions of extrapolation are important – have you learned the right concepts? For instance, can you generalize sequences of length 5 to length 10? Images with three cows to ten cows? Does it generalize beyond the dataset?

6 Audience question 1: One interesting point, there's a question of neural nets learning sensible abstractions. They're not: We have adversarial examples to demonstrate that. Do we need to rethink the architectures we are using? For instance, there's Geoff Hinton's capsules...

Russ: New architectures with invariance and geometry is great. It's good to also incorporate prior knowledge. A face with three eyes is still a face. How do you incorporate these new priors? There are notions of memory, neural machine translation, and so on. It's not claer how well capsules will work but it is still interesting. How to design new architectures which have inductive biases or can learn inductive biases is a great research question. Learning the whole spectrum is difficult. If I take a big net that is not convolutional, it is very hard to get these models to the same levels of performance. CNNs have a lot of prior knowledge which work really well for images.

Percy: Inductive bias is definitely important for sample efficiency. It's more important if you're trying to learn with adversarial examples. Yoshua showed in some paper a while ago I think that if you change the Fourier spectrum of images, the CNN does not work at all. Why would a particular architecture match the human visual system? I'm pessimistic you can just pick an architecture that will match human intuition for what is right on a lot of

problems. Can you make machines like humans? I'm pessimistic about that too.

7 Audience question 2: Optimization moving forward has been mostly first order, taking the approach "convex until proven guilty" – how much can we squeeze out of just first order methods?

Sham: In RL a lot of solution concepts are not just first order methods. There are trust region methods for instance. RL is also tricky, we have things like Monte Carlo Tree Search as a part of the optimization procecure. What are the boundaries between optimization and learning? In the sense of raw end-to-end supervised learning problems, SGD might be the right thing to do. It is tied to a natural form of regularization and generalization. In broader settings this is less clear.

Russ: In my experience things like KFAC (Kronecker-Factored Approximate Curvature) work well in terms of optimizing a training objective. It helps only a little in terms of generalization though. There should be something that goes beyond SGD. This is going back to the optimization-generalization question.

8 Audience question 3: Followup – is it helpful to find local minima? There is a lot of interest in global optima but do you think there are good things about local optima?

Peter: It is plausible that they are in fact finding global optima and then the question is why are these particular global optima generalizing well? Are optimization methods finding interpolating solutions? First order methods seem to be good at finding predictive rules with low complexity.

Sham: Methods which solve the task of finding global optima are guaranteed to get to at least local optima. It might be that a lot of methods are just getting us to local optima – in control, this is what works.

Peter: I think you're using a slightly different notion of local here.

Sham: In the way you are parametrizing that's essentially equivalent to "local in parameter space" for some models.

9 Concluding Thoughts

Sanjeev: Any last thoughts before we wrap up?

Sham: I have a question for you Sanjeev: What do *you* think are the most interesting problems in machine learning and deep learning theory?

Sanjeev: What's the role of theory in RL and deep learning? One of my favorite research questions is "How can differentiable techniques combine with old AI and logic?"

10 Audience question 4: Percy has recent work on generating text. You can add a natural language sentence to a document so that a machine comprehension system performance drops a lot. This can defeat generative models, is there an answer for that?

Sanjeev: If we had a good model for reality as Russ said earlier, we could escape. I think the speaker was pointing out that this can trick the model for reality defense to adversarial attacks.

Percy: Well it was a human who added the sentence, so that's not fair. I think it would look a fair bit different though (if we had a good internal model of reality, that added sentence would be strange).

Russ: This kind of result shouldn't be surprising for physical adversaries. If you had a good generative model, you could detect outside the distribution.

Percy: We have to stop thinking statistically in these adversarial settings. ℓ_∞ is just the tip of the iceberg. How can you be robust against everything?

Sham: Robustness theory might have some contributions to make. Statistics is trying to do this.

Sanjeev: It seems to me like the information-theoretic approach of thinking about unsupervised learning seems off. It is not clear that generating a view is coming out of the distributions.

Percy: Humans don't have a distribution over all sentences except when there's noise in the environment.

Peter: Good example, we generate sentences in a particular way. We consider the previous sentence in a dialogue for instance.

Sanjeev: So what about the question, "When is the probability of this particular image 10^{-51}?" That seems like a flaw of this way of thinking about things.

Percy: How do you fit a policy? Well, maximum likelihood will give you generic responses. There are multiple solutions. That is perhaps a much easier problem to solve than the task of actually generating sensible langauge according to a dialogue.

Sanjeev: There is some logic involved, it is not the same as just having probabilities. We reason in terms of logic and not just probabilities.

11 Audience question 5: What about fuzzy logic?

Percy: I work in semantic parsing, mapping logic to programs. Logic lets you move big pieces of things around at once and you get extrapolation for free, so it is powerful. On the other hand it has sharp curves. Ideally you might have some sort of relaxed version. People work on this. It is a very hard problem to have logical operations. Pragmatically, we just encode these primitives in and add inductive bias, back off to something more soft which has a logical backbone. For hard reasoning tasks, suppose you are trying to grow a vine. You need a trellis, then the vine can grow up to the top. This is filling in the gaps. The trellis here is logic.

Russ: With respect to logic, the question for me is how you can combine it with deep learning? So that if it makes sense, you can pick it up? You want to have a prior which gets rid of moves/rules which don't work. Maybe there is a way of coming up with logic rules. You need to adaptively decide what makes sense. Rules can be used as a kind of prior.

Sham: The aspect of planning in learning seems logic-based. You are fitting a value function in space, there is a different system for rules. Alpha Go is leaning heavily off of Monte-Carlo Tree Search for instance. There is an aspect of continuously approximating the world. Planning itself seems like a separate sub piece of the problem.

Percy: I want to decouple logic as a representation as an alternative for learning. Logic is more useful as a way of encoding structure I think. This is like a digital versus analogue point of view almost. Maybe one needs error correction to make sure the agent does not forget. Humans can build systems which are more reliable.

Sanjeev: So do you think we will ever have a synthesis of 1960s AI and deep learning?

Percy: I don't quite agree with that.