CIS 522: Lecture 2

Linear Deep Theory/ Intro to theory

Penn Engineering

What we learned about regularization

ANNs are efficient universal approximators

ANN's memorize some but generalize well

Regularization as shrinkage

Regularization by Dropout

Regularization by Data Augmentation

Perils of Hyper-Parameter Tuning

Rethinking generalization

Continued

Pruning

Lottery tickets

Distillation

Adversarial attacks

Regularization is surprisingly complex

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

Oriol Vinyals
Google DeepMind
vinyals@google.com

Everything is a Regularizer and Not a regularizer

SGD: The Role of Implicit Regularization, Batch-size and Multiple-epochs

Satyen Kale
Google Research, NY
satyen@google.com

Ayush Sekhari
Cornell University
as3663@cornell.edu

Karthik Sridharan
Cornell University
ks999@cornell.edu

Abstract

Multi-epoch, small-batch, Stochastic Gradient Descent (SGD) has been the method of choice for learning with large over-parameterized models. A popular theory for explaining why SGD works well in practice is that the algorithm has an implicit regularization that biases its output towards a good solution. Perhaps the theoretically most well understood learning setting for SGD is that of Stochastic Convex Optimization (SCO), where it is well known that SGD learns at a rate of $O(1/\sqrt{n})$, where n is the number of samples. In this paper, we consider the problem of SCO and explore the role of implicit regularization, batch size and multiple epochs for SGD. Our main contributions are threefold:

1. We show that for any regularizer, there is an SCO problem for which Regularized Empirical Risk Minimzation fails to learn. This automatically rules out any implicit regularization based explanation for the success of SGD.
2. We provide a separation between SGD and learning via Gradient Descent on empirical loss (GD) in terms of sample complexity. We show that there is an SCO problem such that GD with any step size and number of iterations can only learn at a suboptimal rate: at least $\widetilde{\Omega}(1/n^{5/12})$.
3. We present a multi-epoch variant of SGD commonly used in practice. We prove that this algorithm is at least as good as single pass SGD in the worst case. However, for certain SCO problems, taking multiple passes over the dataset can significantly outperform single pass SGD.

We extend our results to the general learning setting by showing a problem which is learnable for any data distribution, and for this problem, SGD is strictly better than RERM for any regularization function. We conclude by discussing the implications of our results for deep learning, and show a separation between SGD and ERM for two layer diagonal neural networks.

Looking forward towards projects

What makes for a good deep learning class project

The papers you read / we teach

Experiments with huge datasets (imagenet is tiny)

Competitions on popular datasets

Theories on “why deep learning works”

New general purpose architectures

Projects that start from zero

And everything on the list would be horrible class projects

Let us talk about the data space where DL class projects work

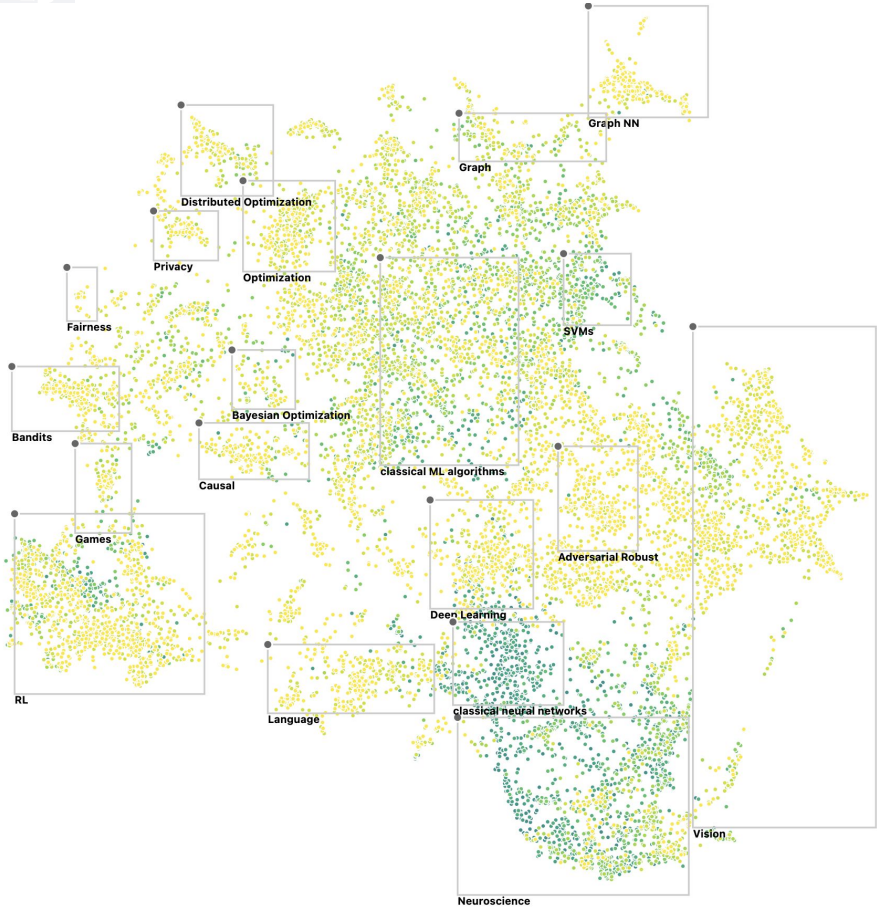
DL usually requires relatively large amounts of data (often not competitive on N=100)

You can not train on really large amounts of data (N>10,000 is *hard* on colab)

You often need to use more

You will almost always start with pre-trained models that others trained on big data - inductive biases

First, you gotta be excited about it!



What's your thing?

- Vision
- Text
- Reinforcement Learning
- Graphs
- Theory
- None of the above

Second you need to have a question

So what makes for a good question?

What makes a question a good question?

Criteria for a good question

Must be a question

Must have an answer

Answer must be useful

Answer must come from data

DL must be useful to answer question

You must be able to pull it off

Must not be unethical

Question should not already be answered

Must be a question

We apply convnets to a really large dataset of pictures of alligators

We use GANs and make them bigger

Must have an answer

We will ask how conscious neural networks are

We will ask how learning works in brains

Must be useful

We will ask if Alexnet is better with ReLU or with leaky ReLU

We will distinguish cats from dogs

Answer must come from data

We will analyze what things make people happy

We will analyze why helicopters fly

DL must be useful for the problem

Predict mood based on phone accelerometry K~10 from N~100 students

Decode user intent from brain data from K~200 brain regions from N=80 users

Remark: You will always compare against standard ML

You must be able to pull it off

Data must be small enough

Little enough compute

Not too many steps.

Data must be available. To you.

Must not be unethical

Recognize faces of people that wear a mask

Build a system that automatically writes class project reports

Build a system that finds the right insult for the right person

Novelty

It is a class project.

No need to be really novel.

But, it is very much better if it is.

Rather valiantly fail on a great problem then recook some boring stuff.

What matters for project quality

Find a good question

Show DL skills

Show data thinking

Show DL skills

Do not use complex stuff because it is complex

Simplicity always wins

Still, it is fun to revisit a range of skills you obtained in CIS 522

Show Data skills!

A good data scientists

- Understands the domain.
- Uses DL to build in knowledge of the domain.

Examples of usable domain knowledge

Electrodes in brains have physics. We know physics is low-d. So lets do preprocessing shared across electrodes.

Music generation. Western music tends to have a tonic key. The patterns of the rhythm section are often conserved. Build both in.

More

Neuron connection probability depends on neuron type. But it also depends on distance. So if I want to learn the type I want distance to be removed.

Modeling hand movement in space is highly stereotypical. So better decoding by having a good model of movements.

DL makes it easy to build in knowledge

- Architecture
- Cost functions
- Data interfaces
- Domain adaptation

Name a question that you could imagine answering in a class project

So in summary

Find a topic you love.

A good question.

In a domain that you understand.

A dataset that you really get.

Good data science = thinking about data.

Shoe Leather approach to data science

This week

Learn about convnets

Convolutions and why they are useful

Pooling

Needing fewer parameters

Train faster and better

Data augmentation (more data is better)

Not only for vision