
Costis Daskalakis
Aleksander Mądry
Course Logistics

• Website: https://stellar.mit.edu/S/course/6/sp18/6.883/index.html
• Mailing list: 6883-all@lists.csail.mit.edu [Make sure to fill out the form]
• Prerequisites: algorithms (6.046); probability (6.042/6.041/6.008); ML (6.867)
• Format: Five modules (five lectures each)

1. Optimization and Generalization in Deep Learning
2. (Deep) Generative Models
3. Robust/Secure Machine Learning
4. Deep Reinforcement Learning
5. Societal Impact of Machine Learning

• Scribe notes [45%]
• Crucial aspect: Class discussion [10%]
• Class projects: Explores questions raised in discussion (experiments and theory); done in 2-3 person student teams [45%]
  [We will run a team matching process soon]
What will this class be about?

Goal: Build a principled and crisp overview of what deep learning can and cannot do, and what we do and do not know about it.

Science = theoretical models + empirical evaluation
What this class is NOT?

- Intro to machine learning/deep learning/Tensorflow/PyTorch/…
  → 6.867, 6.S198
  → http://www.coursera.org/learn/machine-learning
  → http://www.fast.ai/
  → http://neuralnetworksanddeeplearning.com/ (Book)
  → http://www.deeplearningbook.org/ (Book*)

A survey of state of the art deep learning techniques
  → Impossible (10s of papers uploaded every day)

- Tips on how to make your AI/deep learning startup cooler

Key skill we want you to develop:
  “Critical thinking” about deep learning (and ML/AI, in general)
Humble beginnings

- Perceptron [Rosenblatt ‘58]
- Criticism of Perceptrons (XOR affair) [Minsky Papert ‘69]
  → Effectively causes a “deep learning winter”
(Early) Spring

- **Back-propagation** [Rumelhart et al. ‘86, LeCun ‘85, Parker ‘85]

  ![Back-propagation Diagram](image)

- **Convolutional layers** [LeCun et al. ‘90]

  ![Convolutional Layers Diagram](image)

- **Recurrent Neural Networks/Long Short-Term Memory (LSTM)** [Hochreiter Schmidhuber ‘97]

  ![LSTM Diagram](image)
Summer

• **2006:** First big success: speech recognition

• **2012:** Breakthrough in computer vision: AlexNet [Krizhevsky et al. ‘12]

• **2015:** Deep learning-based vision models outperform humans
What enabled this success?

- Better architectures (e.g., ReLUs) and regularization techniques (e.g. Dropout)
- Sufficiently large datasets
- Enough computational power
Geist of deep learning
Module I: Optimization and Generalization in Deep Learning
Supervised Machine Learning

\[ f^* = \text{concept to learn} \]
Supervised Machine Learning

\[ f^* = \text{concept to learn} \]
Supervised Machine Learning

\[ f^* = \text{concept to learn} \]

**Training:** Recover (approx. of) \( f^* \) by finding parameters \( \theta^* \) s.t. \( f(\theta^*) \) fits the training data

\[ f(\theta) = \text{classifier (parametrized by } \theta) \]

Choice of (the family) \( f(\cdot) \) is crucial

Too simple \( \rightarrow \) underfitting
Supervised Machine Learning

\[ f^* = \text{concept to learn} \]

**Training:** Recover (approx. of) \( f^* \) by finding parameters \( \theta^* \) s.t. \( f(\theta^*) \) fits the training data

\[ f(\theta) = \text{classifier (parametrized by } \theta) \]

Choice of (the family) \( f(\cdot) \) is crucial

Too simple \( \rightarrow \) underfitting

Too flexible \( \rightarrow \) overfitting
Supervised Machine Learning

Training: Recover (approx. of) $f^*$ by finding parameters $\theta^*$ s.t. $f(\theta^*)$ fits the training data

$f(\theta) = \text{classifier (parametrized by } \theta)$

Choice of (the family) $f(\cdot)$ is crucial

Too simple $\rightarrow$ underfitting
Too flexible $\rightarrow$ overfitting

"Classic" ML developed a rich and successful theory to understand this phenomenon
Deep neural networks are very expressive, why don’t they overfit?
Our true goal: To minimize (wrt $\theta$) the population risk

$$E_{(x,y) \sim D} \left[ \text{loss}(f(\theta,x),y) \right]$$

What we actually do: Minimize (wrt $\theta$) the empirical risk

$$\sum_i \text{loss}(f(\theta,x_i),y_i)$$

where $\{(x_i,y_i)\}_i$ are the training data points

→ In case of neural networks, empirical risk is a continuous and (mostly) differentiable function
→ Can use gradient descent method (back-propagation) to solve it!
Optimization in Deep Learning

$$\min_\theta \sum_i \text{loss}(f(\theta,x_i),y_i)$$

→ **Issue 1:** There is a lot of terms in this sum  
→ Use **stochastic** gradient descent (SGD) instead of grad. descent  
  (SGD = the workhorse of deep learning)

→ **Issue 2:** This problem is very non-convex  
→ Still, we seem to reliably* converge to good solutions. Why?

**In fact:** Stochasticity of SGD seems to be a “feature”, not a deficiency. (Hypothesis: “Implicit regularization.”)
Module II: Deep Generative Models
Unsupervised Machine Learning

- **Goal:** Learn from *unlabeled* data by understanding its structure

  Popular approach: Try to fit the data to some *generative model*

- **Example:** Fit the distribution to a mixture of Gaussians
Deep Generative Models

• Neural networks constitute (parametric) models too!

• Variational Autoencoders (VAEs) [Kingma Welling ’13, Rezende et al. ’14]

• Generative Adversarial Networks (GANs) [Goodfellow et al. ’14]

Questions:
• What are/should be the guarantees these models aim to satisfy?
• Do existing constructions work? Can they ever?
• How would we measure their success?
Module III: Robust/Secure ML
Recent Progress in ML

Have we *really* achieved human-level performance?
Adversarial Examples

Translations + rotations
(shifts by <10% pixels, <30° rotations)

CIFAR10: 93% → 8% accuracy
ImageNet: 76% → 31% accuracy

Too fragile?

Too contrived?
[Engstrom, Tsipras, Schmidt, M., 2017]
Why Does It Matter?

- **Security** (currently, everything is “broken”)

- **Safety** (“benign” noise can be a problem too)

- Understanding “failure modes” of current vision models
  (they are not as “human-like” as we might have expected)

[Sharif, Bhagavatula, Bauer, Reiter, 2016]

**Crucial question:**
Can you really rely on your (deep) ML model?
What Do We Do Now?

• **Problem:** Adversarial examples are **not** at odds with our current notion of generalization

• Time to re-think what we mean by generalization?

• There is a number of other problems/questions, such as data poisoning, model theft,…

• **Again:** This is not only about security/safety but also about understanding how ML/deep learning works (and fails!)
Module IV: (Deep) Reinforcement Learning
Reinforcement Learning (RL)

Questions:

• How to train such agent (exploration vs. exploitation)?
• What are the fundamental limits on efficiency of this approach?
• How to ensure that the agent does what we really intend it to do?

What if the Agent was a (deep) neural network?
Module V: Societal Impacts of ML
Machine learning is entering (and taking control of) every aspects of our life

• Should we be worried?

• Potential concerns:
  → Interpretability (Can we understand ML models “reasoning”?)
  → Reliability (Can I trust the prediction of an ML model?)
  → Fairness (Is the ML model behaving in a “fair” way?)
  → Privacy (Is the ML model protecting our privacy?)
  → AI Safety (If we build a super-human AI, will it destroy us?)
  → (Your suggestion here)

Costis Daskalakis
Aleksander Mądry
Learning Theory

what, when, how do deep NNs learn?
e.g. Classification

- Basic learning task: design function $h: \mathcal{X} \rightarrow \mathcal{C}$, mapping objects from some set $\mathcal{X}$ to their class label in $\mathcal{C}$

- e.g. $\mathcal{X}$: images of cats and dogs, $\mathcal{C} = \{0, 1\}$

- How to do this?
  
  1. identify “expressive enough” family of functions $\mathcal{H}$
  
  2. use examples to choose some “good” $h \in \mathcal{H}$
e.g. Classification

- Basic learning task: design function $h: \mathcal{X} \rightarrow \mathcal{C}$, mapping objects from some set $\mathcal{X}$ to their class label in $\mathcal{C}$

- e.g. $\mathcal{X}$: images of cats and dogs, $\mathcal{C} = \{0,1\}$

- How to do this?

  1. identify “expressive enough” family of functions $\mathcal{H}$
     - e.g. $\mathcal{H}$ all convolutional nets of certain width and depth

  2. use examples to choose some “good” $h \in \mathcal{H}$
     - each example is a pair $(x, y)$ of an image and its label

     - output empirical risk minimizer:
       $$\hat{h} \in \operatorname{argmax}_{h \in \mathcal{H}} \sum_{\text{examples } (x_i, y_i)} 1_{h(x_i) = y_i}$$
e.g. Classification

- identify “expressive enough” family of functions $\mathcal{H}$
  - e.g. $\mathcal{H}$ all convolutional nets of certain width and depth
- use examples to choose some “good” $h \in \mathcal{H}$
  - output empirical risk minimizer $\hat{h} \in \arg\max_{h \in \mathcal{H}} \sum_{(x_i,y_i) \in \mathcal{E}} 1_{h(x_i) = y_i}$
- hope: $\mathbb{E}_{(X,Y) \sim F} [1_{\hat{h}(X) = Y}] \geq \max_{h \in \mathcal{H}} \mathbb{E}_{(X,Y) \sim F} [1_{h(X) = Y}] - \epsilon$

- $F$: true distribution of (image, class label) pairs to be encountered in the future
  - presumably training set of examples were drawn from $F$

Two questions:

1. How close is $\max_{h \in \mathcal{H}} \mathbb{E}_{(X,Y) \sim F} [1_{h(X) = Y}]$ to $\max_{h: \text{unrestricted}} \mathbb{E}_{(X,Y) \sim F} [1_{h(X) = Y}]$?
2. How fast does $\epsilon$ decay in the number of examples $N$?

- Rich $\mathcal{H}$ $\Rightarrow$ 1 good, 2 bad
- Poor $\mathcal{H}$ $\Rightarrow$ 1 bad, 2 maybe good
- For 1, use a rich enough family $\mathcal{H}$
- For 2, bound the “dimensionality” of $\mathcal{H}$, get generalization bounds
Generalization Bounds

• How to prove?
  • Many ways, central topic in ML theory
  • **Here:** Vapnik–Chervonenkis (VC) theory

• Consider a class of Boolean functions $\mathcal{H} = \{ h: \mathcal{X} \rightarrow \{0,1\} \}$

• **Def:** VC dimension of $\mathcal{H} = \max \text{ #points } \mathcal{H} \text{ can shatter}

  • points $x_1, ..., x_k \in \mathcal{X}$ are **shattered** by $\mathcal{H}$ iff $\forall$ 0/1 patterns $\sigma \in \{0,1\}^k$ $\exists$ a function $h \in \mathcal{H}$ whose values on the points $x_1, ..., x_k$ equal $\sigma$, i.e. $h(x_i) = \sigma_i, \forall i$

  • e.g. say $\mathcal{H} = \{ \text{halfplanes in } \mathbb{R}^2 \}$

  • $VC(\mathcal{H}) = 3$
Generalization Bounds

• How to prove?

• Many ways, central topic in ML theory

• **Here:** Vapnik–Chervonenkis (VC) theory

• Consider a class of Boolean functions \( \mathcal{H} = \{ h: \mathcal{X} \to \{0,1\} \} \)

• **Def:** VC dimension of \( \mathcal{H} = \max \) #points \( \mathcal{H} \) can **shatter**

• points \( x_1, \ldots, x_k \in \mathcal{X} \) are **shattered** by \( \mathcal{H} \) iff \( \forall \) 0/1 patterns \( \sigma \in \{0,1\}^k \) \( \exists \) a function \( h \in \mathcal{H} \) whose values on the points \( x_1, \ldots, x_k \) equal \( \sigma \), i.e. \( h(x_i) = \sigma_i, \forall i \)

• e.g. say \( \mathcal{H} = \{ \text{halfplanes in } \mathbb{R}^2 \} \)

• \( \text{VC}(\mathcal{H}) = 3 \)

• **VC Theorem:** Suppose \( \mathcal{H} \) is a class of Boolean functions and VC-dimension \( d \). Then given:
\[
N = \frac{(d \cdot \ln(1/\epsilon) + \ln(1/\delta))}{\epsilon^2}
\]
samples \( (X_1, Y_1), \ldots, (X_N, Y_N) \sim F \) we have that, w/ prob \( \geq 1 - \delta \),
\[
\forall h \in \mathcal{H}: \left| \mathbb{E}_{(X,Y) \sim F} [1_{h(X)=Y}] - \frac{1}{N} \sum_{i} 1_{h(X_i)=Y_i} \right| \leq \epsilon
\]
Generalization Bounds

• How to prove?
  • Many ways, central topic in ML theory
  • **Here**: Vapnik–Chervonenkis (VC) theory
  • Similar theorems for real-valued functions via Rademacher complexity, pseudo-dimension, …
  • also for different access to examples
  • Well-developed theory
• Disconnect with practical performance of Deep NNs:
  • VC/Rademacher complexity of Deep NNs too large compared to sample size: is there overfitting?
  • Finding ERM is sort of hopeless; maybe SGD finds local optimum:
    • maybe a good thing?
    • Is there an optimality vs overfitting tradeoff?
    • Is stochasticity in GD also a good thing?
• Role of optimization method, max pooling, dropout?
• Training set: attacks because training set non-representative or because of overfitting?
Statistics
Generative Adversarial Networks

• Algorithms mapping white noise to high-dimensional objects with structure

\[ z \sim N(0, I_{100\times100}) \]

• If you want, what human imagination does (presumably)

• Trained using samples (e.g. faces) from true high-dimensional distribution with structure (e.g. natural face images)

• *Statistical Question:* after GAN has been trained, did it really learn the underlying structured high-dimensional distribution?

• Or did it “memorize” the training set?
A Hypothesis Testing Problem

- Sample access to $F$: distribution of true faces
- Sample + white-box access to $Q$: GAN, and its output
- **Goal**: distinguish $d(F, Q) \leq \varepsilon_1$ vs $d(F, Q) \geq \varepsilon_2$
- Really well-studied problem in Statistics, Information Theory, TCS
- **Trouble is**:
  - what is the right distance $d$ to use?
  - $F, Q$: high-dimensional (e.g. face image distributions)
    - Statistical tests commonly require exponentially many samples in the dimension, unless one has deeper understanding of structure in both $F$ and $Q$
    - e.g. even if $Q$ is trivial (product measure), and $d$ is total variation distance, answering above question requires exponentially many samples in the dimension.

- What is the right statistical lens via which to approach this question?
Game Theory
Think $F$: true high-dimensional distribution (e.g. faces) in $\mathbb{R}^n$

$Q$: output of a Deep NN $G$, of certain architecture, with parameters $\theta$
  - i.e. $G_\theta(z)$, where $z \sim N(0, I)$

Suppose interested in Wasserstein distance:

$$W(F, Q) = \sup_{D: \mathbb{R}^n \to \mathbb{R}, \text{1-Lipschitz}} (\mathbb{E}_{X \sim F}[D(X)] - \mathbb{E}_{X \sim Q}[D(X)])$$

In a perfect world, $G_\theta$ should minimize:

$$\inf_{\theta} \sup_{D: \mathbb{R}^n \to \mathbb{R}, \text{1-Lipschitz}} (\mathbb{E}_{X \sim F}[D(X)] - \mathbb{E}_{Z \sim N(0,I)}[D(G_\theta(Z))])$$

In practice, hard to compute sup over all Lipschitz functions, so only take sup over all Deep NNs $D$, of certain architecture, w/ parameters $w$:

$$\inf_{\theta} \sup_{w} (\mathbb{E}_{X \sim F}[D_w(X)] - \mathbb{E}_{Z \sim N(0,I)}[D_w(G_\theta(Z))])$$

In other words, set up a game between a Generator deep NN, and a Discriminator deep NN
GAN Training

- A game between a Generator deep NN, w/ parameters \( \theta \) and a Discriminator deep NN, w/ parameters \( w \):

\[
\inf_{\theta} \sup_{w} \left( \mathbb{E}_{X \sim F} [D_{w}(X)] - \mathbb{E}_{Z \sim N(0,I)} [D_{w}(G_{\theta}(Z))] \right)
\]

- Training: generator and discriminator run some variant of gradient descent each to update their parameters \( \theta, w \); expectations are approximated by sample averages.
GAN Training

- A game between a Generator deep NN, w/ parameters $\theta$ and a Discriminator deep NN, w/ parameters $w$:
  \[
  \inf_{\theta} \sup_{w} (\mathbb{E}_{x \sim F} [D_{w} (x)] - \mathbb{E}_{z \sim N(0, I)} [D_{w} (G_{\theta}(z))])
  \]

- **Training**: generator and discriminator run some variant of gradient descent each to update their parameters $\theta, w$; expectations are approximated by sample averages.

- Will gradient descent converge?

- If yes, to what?
The Min-Max Theorem

• **[von Neumann 1928]**: If \( X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m \) are compact and convex, and \( f: X \times Y \rightarrow \mathbb{R} \) is convex-concave (i.e. \( f(x, y) \) is convex in \( x \) for all \( y \) and is concave in \( y \) for all \( x \)), then
  \[
  \min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y)
  \]

• Min-max optimal \((x, y)\) is essentially unique (unique if \( f \) is strictly convex-concave, o.w. a convex set of solutions)

• von Neumann: "As far as I can see, there could be no theory of games … without that theorem … I thought there was nothing worth publishing until the Minimax Theorem was proved“

• Equivalent to strong LP duality

• **[Blackwell,…]**: A host of uncoupled update-rules (dynamics) applied by the min and the max players “converge” to min-max equilibrium

• *no-regret learning dynamics*: e.g. Multiplicative-weights-update, follow-the-regularized-leader, follow-the-perturbed-leader, etc.

• Follow-the-regularized-leader with \( \ell_2 \)-regularization \( \equiv \) gradient descent
Challenges

- “Convergence” of online learning to min-max solutions for convex-concave functions $f(x, y)$ only happens in an average sense
  - E.g. gradient descent for $f(x, y) = x \cdot y$

Objective function in Wasserstein GAN training isn’t convex-concave

Questions:
- Stability: how to converge to local saddles?
- Generalization: Effects of approximation of expectation with sample averages?
Game Theory
Game Playing

Deep I Teache Hours,

DeepMind pro not DQN was only otherwise left
Atari games. The as IBM’s Watson
information the

It’s been almost 20 ye
champion, Gary Kas
chess-playing compu
cChance even against

Cofounder and WIRE2014 speaker Demis Hassabis called the
move, detailed in a paper published in Nature, “the first significant
rung on the ladder to proving general learning systems can work”.
“it’s the first time that anyone has built a single general learning
Deep Mind

- Stated Mission: Solve intelligence, use it to make the world a better place.
- ...
- We’ll take a look at the guts of AlphaGo, and AlphaGo Zero
- Connection to Reinforcement Learning, Policy and Value Iteration, and the Min-Max Theorem
6.883 Statement of Purpose:
- to entice the practically-minded into theory as a means to understand and improve practice
- to entice the theoretically-minded into the deep questions motivated by practical experience
Outlook

• Really small sample regime: health data
• Robust Statistics
• Causality + Counterfactuals
• Privacy concerns
• Fairness
• Ethical Considerations
• Philosophical ramifications of unreasonable practical success of Deep Learning