Part 1 (brief):
Is optimization the right language for understanding deep learning?

Min $f(x)$ for some objective $f$; solve as fast as possible

"need to perform well on training data..."
[Zhang et al.'17] Current deep nets are so overparametrized that they can fit even randomly labeled data (regularization doesn’t help..)

Emerging view: overparametrization enables fast optimization, and on “proper” data doesn’t hurt generalization…
My view: Most of the magic happens due to **dynamics** of gradient descent trajectory (not captured by objective *per se*)

“Implicit bias of gradient descent” [Neyshabur et al’17, Gunasekar et al’17]

(NB: Not to be confused with beneficial effects of noise in the process due to **Stochastic GD**, which is probably a different mechanism.)
Vignette 1: Infinitely wide deep nets.

- Consider any standard architecture, eg AlexNet
- Let its “width” (= # channels in conv. Filters, or # nodes in FC layers) go to infinity. (Initialize w/ tiny gaussians so layer norms bounded.)
- Train via GD on finite dataset, e.g. CIFAR10.

Thm (emerged from seq. of recent papers): GD trajectory approaches a limit, which is equiv. to regression via ”Neural Tangent Kernel” (NTK)

Vignette 2: Solving matrix completion via deep linear nets


[Srebro et al’05] Nuclear norm minimization (convex)

$$\sum_{ij \in \Omega} (M_{ij} - b_{ij})^2 + \lambda |M|_*$$

[Gunasekar et al’17] Find $M$ as a product of two matrices (depth 2 linear net)

$$\sum_{ij \in \Omega} ((M_2M_1)_{ij} \in \Omega - b_{ij})^2$$

Conjecture: GD minimizes nuclear norm...

[A, Cohen, Hu, Luo’19]: Conjecture seems false. Beneficial effect arises from dynamics along GD path. Performs better than nuclear norm minimization as depth rises...
Paper: A theoretical analysis of contrastive unsupervised representation learning”
[A., Hrishikesh Khandeparkar, Mikhail Khodak (CMU), Orestis Plevrakis, Nikunj Saunshi ARXIV’2019)

Theory for representation learning

Sanjeev Arora
Princeton University and Institute for Advanced Study

http://www.cs.princeton.edu/~arora/
Group website: unsupervised.cs.princeton.edu
Blog: www.offconvex.org
Twitter: @prfsanjeevarora

Support: NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Resarch, Mozilla Research. DARPA/SRC
Why does learning to do A help you do B later on?

Example:
A = major in math
B = start a hedge fund

Surprisingly, this is hard to capture* for Machine Learning Theory

(*except if you go hardcore, full Bayesian.
But, even then unclear how to interpret the phenomena we’ll see in next few slides.)
Talk Overview

- Part 1: Representation learning. Desired goals and need for new theoretical framework
- Part 2: The Lore of representations/embeddings
- Part 3: Our new framework; minimalistic yet surprisingly powerful.
- Part 4: Some experiments
PART 1: REPRESENTATION LEARNING AND ITS GOALS
(AND NEED FOR NEW THEORETICAL FRAMEWORK…)

(“Solving Task A later helpful in doing Task B. “ )
Data Representation

Solve other tasks using these representations.

Often called “embeddings” (e.g., text embedding).

(Neuroscience example: x = image
  f(x) = Pattern of visual cortex activations it leads to...)
Powerful Data Representation

Data \[ x \]

Representation \[ f(x) \]

Allows new classification tasks to be solved well using linear classifier on \( f(x) \) and small amounts of labeled data.

(f = representation function)

method to bring semantic content “to the surface”.

Classic example: Kernel SVMs. “Lift” data to kernel representation, classify using linear classifier.
With lots of labeled data, deep nets learn powerful representations

Trained on labeled dataset ImageNet: (10^3 classes, 10^3 examples ea.)

- Classification accuracy abysmal if trained with 2 classes (⇒ other 998 classes important for learning the right representation!)
- Penultimate layer gives powerful representation!
With lots of labeled data, deep nets learn powerful representations

“Headless well-trained deep net” (Gold Standard in this field)

Trained on labeled dataset ImageNet: $(10^3 \text{ classes}, 10^3 \text{ examples ea.})$

- Classification accuracy abysmal if trained with 2 classes ($\Rightarrow$ other 998 classes important for learning the right representation!)

Can we learn equally powerful representations using only unlabeled data?

Vector on penultimate layer useful: solves new unrelated tasks via linear classifier!
Powerful Data Representation

Data $x$  Representation $f(x)$

Allows new classification tasks to be solved well using linear classifier on $f(x)$.

(As good as headless well-trained deep net??)

What theory can analyse such things?
**Powerful Data Representation**

Data \( x \) → Representation \( f(x) \)

Allows new classification tasks to be solved well using **linear** classifier on \( f(x) \).

As good as headless well-trained deep net??

(f = representation function)

What theory can analyse such things?

* test and train involve different objectives…
* don’t know test tasks while training…
PART 2: THE LORE OF SEMANTIC EMBEDDINGS…

(Created via solving Task A, helpful in doing Task B.
Not much theoretical analysis…)
Ex1: Word embeddings via language models

Idea: Using large corpus (eg, Wikipedia), train a model to predict part of text from adjacent text.

Example: “I went to a café and ordered a…. “

(In learning to do this, model implicitly picks up on grammar rules, common sense etc. )
Ex1: Word embeddings via language models

Baby word2vec [Mikolov et al’13]

"I went to a café and ordered a…. “

\[ \theta = \{ v_w \in \mathbb{R}^{300} : \text{w an English word} \} \]

(“word embeddings”)

Preceding words \( w_1, w_2, \ldots, w_5 \)

Distribution on all English words

\[ \Pr[w|w_1 \ldots w_5] \propto \exp\left( \frac{1}{5} \sum_i \langle v_w, v_{w_i} \rangle \right) \]

(implicit normalizing term; will ignore for simplicity)

Loss \( \ell(\theta) \) : Reciprocal of Probability assigned by model to Wikipedia = \( w_1 w_2 w_3 \ldots w_N \)

Training method: negative sampling. Tries to give high inner product to word pairs occurring nearby, and low inner product to random pairs of words.
Ex 3: Sentence embeddings via QuickThoughts

[Logeswaran & Lee, ICLR’18] “like word2vec..”

Using text corpus (eg Wikipedia) train deep representation function $f$ to minimize

$$
\mathbb{E} \left[ \log \left( 1 + e^{f(x)^T f(x^{-}) - f(x)^T f(x^+)} \right) \right]
$$

$x, x^+$ are adjacent sentences, $x^-$ is random sentence from corpus

(“Make adjacent sentences have high inner product, while random pairs of sentences have low inner product.”)

We call such word2vec-like methods

“Contrastive Learning”

(Also work for molecules, genes, social nets,..).

[For image embeddings, Wang-Gupta’15 use video...]
Past ideas that seem related:

1) Semi-supervised methods: but training needs labeled data from classification tasks (e.g., kernel learning)
2) Generative models (e.g. topic models, language models, VAE, etc.)

Why we don’t use generative models

• Not known for complicated data types (e.g. images, text).
• Contrastive learning works for radically different data types, which are unlikely to share the same generative model.
PART 3: NEW FRAMEWORK FOR CONTRASTIVE UNSUPERVISED LEARNING: THE PARTS

"Why do representations learnt via contrastive learning help in downstream classification tasks?"
Goal: **Powerful Data Representations**

**Part 1** of theory: Available data consists of:

Pairs $(x, x^+)$ of "similar" inputs.

Random pairs of inputs $(x, x^-)$ treated as "dissimilar."

Data

Representation

$x$  \quad f(x)$

Allows new classification tasks to be solved well using linear classifier on $f(x)$.

“Contrastive Data.”
Part 2: Learn best representation from function class $\mathcal{F}$

Available data: Pairs $(x, x^+)$ of "similar" inputs. Random pairs of inputs $(x, x^-)$ treated as "dissimilar."

Fix particular deep net architecture (eg., ResNet 50 of certain size)

Find $f$ in this class that minimizes the loss

$$L_{un}(f) = \mathbb{E}_{(x, x^+) \sim D_{sim}} \left[ \log \left( 1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)} \right) \right]$$

Goal for Today: Understand why low unsupervised loss helps for supervised tasks later.

NB: Theory also suggests some new loss fns as we see later...
Part 3.1: Assumption about “Semantically similar” pairs

Collection of latent classes $C$

$c_1 \sim \rho(c)$

$c_2 \sim \rho(c)$

$\rho(c) = \text{prob. assoc. with class } c$

$x \sim D_{c_1}(x)$

$x \sim D_{c_2}(x)$

Semantically similar $(x, x^+)$

Negative sample $x^-$

(Reminiscent of co-training and Multiview assumptions...)

- Classes sampled from $C$ according to $\rho(c)$
- Samples $x$ from $x \sim D_c(x)$
Part 3.1: Assumption about “Semantically similar” pairs

• A ‘class’ defines distrib. $D_c$ on datapoints; $D_c(x) = \text{Prob. of seeing datapoint } x \text{ in } c$ (note: $x$ may lie in many classes, which can overlap arbitrarily)

Key assumptions
• “Similar pairs”: Pick $c_1$ according to $\rho$ and then two indep. samples $x, x^+$ from $c_1$ according to $D_{c_1}()$
• “Negative Sample/Dissimilar pairs”: Pick $c_2$ according to $\rho$ and then $x^-$ according to $D_{c_2}()$
Part 3.2) What downstream classification tasks are of interest?
(For now, restrict to 2-way classification)

• Nature picks random pair of classes \((c_1, c_2) \propto \rho(c_1)\rho(c_2)\)

• Pick \(k_1\) i.i.d. samples from \(D_{c_1}()\), and \(k_2\) iid samples from \(D_{c_2}\), where \(k_1/k_2\) can depend on pair \((c_1, c_2)\).

Part 3.3) Evaluation of representation: Pick random binary task as above. Solve by training logistic classifier on the representations. (Theory extends to hinge loss...)

\[
L_{sup}(task, f) = \inf_w \mathbb{E}_{(x,c) \sim task} \log(1 + \sum_{c' \neq c} e^{f(x)^T(w_{c'} - w_c)})
\]

\[
L_{sup}(f) = E_{task}[L_{sup}(task, f)]
\]
(Reminder) Logistic classifier on binary task. *

Given: Data labeled with 0/1

Trains vectors $w_1, w_2$.

Output on input $x$ is the following:

$$P(y = 1) = \frac{e^{\langle w_1, x \rangle}}{e^{\langle w_1, x \rangle} + e^{\langle w_2, x \rangle}}$$

$$P(y = 2) = \frac{e^{\langle w_2, x \rangle}}{e^{\langle w_1, x \rangle} + e^{\langle w_2, x \rangle}}$$

* Aka “softmax,” usually used as the top layer of deep nets
THE ANALYSIS...

(NB: Will ignore computational cost, and just analyse quality of representations that have low training loss.)
A) Sample complexity of training

Unsupervised loss (minimize over f: $\mathbb{R}^n \to \mathbb{R}^d$ that lie in class $\mathcal{F}$):

$$L_{un}(f) = \mathbb{E}_{(x,x^+) \sim D_{sim}, x^- \sim D_{neg}} \left[ \log \left( 1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)} \right) \right]$$

Empirical Loss (for $M$ unlabeled samples, i.e., similar pairs):

$$\hat{L}_{un}(f) = \frac{1}{M} \sum_{i=1}^{M} \left[ \log \left( 1 + e^{f(x_i)^T f(x_i^-) - f(x_i)^T f(x_i^+)} \right) \right]$$

Lemma (rough statement): If $M > d \frac{R(\mathcal{F})}{\varepsilon}$, then these differ by $\leq \varepsilon$. $R(\ ) = \text{Rademacher Complexity.}$
B) Relating classification accuracy to $L_{un}(f)$ (simplest version)

**Theorem: Average Binary Task Guarantee**

With probability at least $1 - \delta$, for all $f \in \mathcal{F}$

$$L_{sup}(\hat{f}) \leq \frac{1}{1 - \tau} \left[ L_{un}(f) - \tau + \varepsilon \right]$$

$\tau =$ collision probability for pair of random classes (usually small)

Translation: Every $f$ with low $\text{unsup.}$. Loss gives low classification loss on avg binary task $c_1, c_2$

(Recall: we solve classification by training logistic classifier on d-dim representations of data using $f$.)
Instead of training best $w_1, w_2$ to minimize logistic loss, just set $w_i$ to be the mean representation of labeled samples from $c_i$

$$\mu_c = \mathbb{E}_{x \sim D_c} f(x)$$

$$L_{\text{sup}}^{\mu}(\text{task}, f) = \mathbb{E}_{(x,c) \sim \text{task}} \log(1 + \sum_{c' \neq c} e^{f(x)^T (\mu_{c'} - \mu_c)})$$

$$L_{\text{sup}}^{\mu}(f) = \mathbb{E}_{\text{task}} L_{\text{sup}}^{\mu}(\text{task}, f)$$

**Theorem: Average Binary Task Guarantee**

With probability at least $1 - \delta$, for all $f \in \mathcal{F}$

$$L_{\text{sup}}^{\mu}(\hat{f}) \leq \frac{1}{1 - \tau} [L_{\text{un}}(f) - \tau + \varepsilon]$$

**Pf idea 1: mean classifiers for 2-way classifications**

Instead of training best $w_1, w_2$ to minimize logistic loss, just set $w_i$ to be the mean representation of labeled samples from $c_i$
Theorem: Average Binary Task Guarantee

With probability at least $1 - \delta$, for all $f \in \mathcal{F}$

$$L^\mu_{sup}(\hat{f}) \leq \frac{1}{1 - \tau} [L_{un}(f) - \tau + \varepsilon]$$

Pf Idea 2

Key step: Jensen’s inequality ( $\varphi(E[X]) \leq E[\varphi(X)]$. )

$$\log \left( 1 + e^{f(x)^T \mu_c- - f(x)^T \mu_c+} \right) \leq \mathbb{E}_{x^+ \sim D_{c+}, \ x^- \sim D_{c-}} \log \left( 1 + e^{f(x)^T f(x-) - f(x)^T f(x+)} \right)$$

Sup loss of mean classifier

Unsup loss

NB: # of labeled samples needed is sample complexity of linear classification (can be made precise; see paper)
More fine grained analysis when $L_{un}(\cdot)$ is not small.

\[ L_{sup}^\mu(\hat{f}) \leq L_{un}^\mu(f) + \frac{2\tau}{1-\tau} s(f) + \frac{1}{1-\tau} \text{Gen}_M \quad \forall f \]

Term for $c^+ \neq c^-$

$s(f)$ is a notion of geometric variance among representations within classes.

Type equation here.

Let $\Sigma(f, c)$ be the covariance matrix of $f(x)$ when $x \sim D_c$ and

\[ s(f) = \mathbb{E}_{c \sim \rho} \left[ \sqrt{\|\Sigma(f, c)\|_2} \mathbb{E}_{x \sim D_c} \| f(x) \|_2 \right] \]

Guarantee is strong if we have

- Contrastive $f$
- Small collision probability
- Concentrated $f$
- More unlabeled data

(Empirically, we find representations are concentrated, so above bound can be stronger)
Dream result for analysis?

If \( \hat{f} \in \arg \min_{f \in \mathcal{F}} \hat{L}_{un}(f) \), “Learnt representation”

then would like “Competitive with BEST representation”

\[
L_{sup}(\hat{f}) \leq \alpha L_{sup}(f) + \gamma \text{Gen}_M \quad \forall f
\]

(2nd term \( \Rightarrow 0 \) as unlabeled data is cheap. Unsup. representation would compete with best representation function \( f \) in the same class of circuits/deep nets)

Easy Thm: This is impossible for an arbitrary class of functions and arbitrary tasks...

Implication: Need more assumptions
We can compete against headless well-trained deep net that produces representations “tightly concentrated” within classes and have high margin using mean classifier.

Thm: for some $f$, $\sigma^2$ sub-gaussian in each class + low $(1 + \tilde{\Omega}(\sigma R))$-margin loss using mean classifier

$\Rightarrow$ low 1-margin loss for our representations.

$(R : \text{max norm of representations})$
Extensions
(briefly…)

• Downstream classification involves $k$-way task. Unsup. learning uses one similar pair and $k-1$ negative samples.

$$L_{un}(f) := \mathbb{E} \left[ \log \left( 1 + \sum_{i=1}^{k-1} \exp \left( f(x)^T f(x_i^-) - f(x)^T f(x^+) \right) \right) \right]$$

however increasing #neg. samples can hurt performance…

• A new unsup. objective if given b-tuples of similar points.

$$L^{\text{block}}_{un}(f) := \mathbb{E} \left[ \log \left( 1 + \exp \left( f(x)^T \frac{\sum_i f(x_i^-)}{b} - f(x)^T \frac{\sum_i f(x_i^+)}{b-1} \right) \right) \right]$$

Gives a tighter bound, so we tried it in the wild and it slightly improves on Quick-thoughts!

Averages within blocks used as proxies of the classifier
Some experiments
Experiments/Test of Theory

- $\mathcal{F} = \text{GRU, VGG-16}$
- Controlled setting, where distributional assumptions hold.
- WIKI-3029: classes are the articles datapoints are sentences.
- CIFAR 100

Representations trained on the full multiclass problem, using labeled data

<table>
<thead>
<tr>
<th></th>
<th>Supervised</th>
<th>Unsupervised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tr</td>
<td>$\mu$</td>
</tr>
<tr>
<td>WIKI-3029</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG-2</td>
<td>97.8</td>
<td>97.7</td>
</tr>
<tr>
<td>AVG-10</td>
<td>89.1</td>
<td>87.2</td>
</tr>
<tr>
<td>TOP-10</td>
<td>67.4</td>
<td>59.0</td>
</tr>
<tr>
<td>TOP-1</td>
<td>43.2</td>
<td>33.2</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG-2</td>
<td>97.2</td>
<td>95.9</td>
</tr>
<tr>
<td>AVG-5</td>
<td>92.7</td>
<td>89.8</td>
</tr>
<tr>
<td>TOP-5</td>
<td>88.9</td>
<td>83.5</td>
</tr>
<tr>
<td>TOP-1</td>
<td>72.1</td>
<td>69.9</td>
</tr>
</tbody>
</table>

Quick-Thoughts

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>b=2</th>
<th>b=5</th>
<th>b=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB</td>
<td>CURL</td>
<td>89.2</td>
<td>89.6</td>
<td>89.7</td>
</tr>
</tbody>
</table>

Blocks can help “in the wild”
Conclusions

• **A first cut** theory for formalization of representation learning; minimalistic assumptions!

• **Future work:** Extensions to more intricate settings (e.g., lattice structure or metric structure among classes)?

• **More empirical and theoretical development?** Transfer learning/meta learning etc.?

Resources [www.offconvex.org](http://www.offconvex.org)

Grad lec. notes on theory of deep learning fall’17 and fall’18