ON THE POWER OF CURRICULUM LEARNING IN TRAINING DEEP NETWORKS

Daphna Weinshall
School of Computer Science and Engineering
The Hebrew University of Jerusalem
NOT MY FIRST JIGSAW PUZZLE
My first JIGSAW PUZZLE
LEARNING COGNITIVE TASKS (CURRICULUM):
NOT MY FIRST CHAIR
学习关于物体外观的知识

SUPERVISED MACHINE LEARNING

- Data is sampled randomly
- We expect the train and test data to be sampled from the same distribution

Exceptions:
- Boosting
- Active learning
- Hard data mining

but these methods focus on the more difficult examples...
CURRICULUM LEARNING

- **Curriculum Learning (CL):** instead of randomly selecting training points, select easier examples first, slowly exposing the more difficult examples from easiest to the most difficult

- **Previous work:** empirical evidence (only), with mostly simple classifiers or sequential tasks
  - $\Rightarrow$ CL speeds up learning and improves final performance

- **Q:** since curriculum learning is intuitively a good idea, why is it rarely used in practice in machine learning?
  - **A:** maybe because it requires additional labeling...

- **Our contribution:** curriculum by-transfer & by-bootstrapping
PREVIOUS EMPIRICAL WORK: DEEP LEARNING

- (Bengio et al, 2009): setup of paradigm, object recognition of geometric shapes using a perceptron; *difficulty is determined by user from geometric shape*.

- (Zaremba 2014): LSTMs used to evaluate short computer programs; *difficulty is automatically evaluated from data – nesting level of program*.

- (Amodei et al, 2016): End-to-end speech recognition in English and Mandarin; *difficulty is automatically evaluated from utterance length*.

- (Jesson et al, 2017): deep learning segmentation and detection; *human teacher (user/programmer) determines difficulty*.
Outline

1. Empirical study: curriculum learning in deep networks
   - Source of supervision: by-transfer, by-bootstrapping
   - Benefits: speeds up learning, improves generalization

2. Theoretical analysis: 2 simple convex loss functions, linear regression and binary classification by hinge loss minimization
   - Definition of “difficulty”
   - Main result: faster convergence to global minimum

3. Theoretical analysis: general effect on optimization landscape
   - Optimization function gets steeper
   - Global minimum, which induces the curriculum, remains the/a global minimum

⇒ theoretical results vs. empirical results, some surprises
**Definitions**

- *Ideal Difficulty Score (IDS)*: the loss of a point with respect to the optimal hypothesis $L(X, h_{opt})$.

- *Stochastic Curriculum Learning (SCL)*: variation on SGD. The learner is exposed to the data gradually based on the *IDS* of the training points, from the easiest to the most difficult.

- SCL algorithm should solve two problems:
  - Score the training points by difficulty.
  - Define the scheduling procedure – the subsets of the training data (or the highest difficulty score) from which mini-batches are sampled at each time step.
Curriculum Learning: Algorithm

- Data, $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$
- Scoring function, $f : \mathbf{X} \rightarrow \mathbb{R}$
- Pacing function, $g_\theta : [M] \rightarrow [N]. \Rightarrow \mathbf{X}_1', ..., \mathbf{X}_M' \subseteq \mathbf{X}$

```
Algorithm Curriculum learning method

Input: pacing function $g_\theta$, scoring function $f$, data $\mathbf{X}$.
Output: sequence of mini-batches $[B'_1, ..., B'_M]$.

sort $\mathbf{X}$ according to $f$, in ascending order
result $\leftarrow []$
for all $i = 1, ..., M$ do
    size $\leftarrow g_\theta(i)$
    $\mathbf{X}'_i \leftarrow \mathbf{X}[1, ..., size]$
    uniformly sample $B'_i$ from $\mathbf{X}'$
    append $B'_i$ to result
end for
return result
```
RESULTS

- Vanilla – no curriculum

- Curriculum learning by-transfer
  - Ranking by Inception, a big public domain network pre-trained on ImageNet
  - Similar results with other pre-trained networks

- Basic control conditions
  - Random ranking (benefits from the ordering protocol per se)
  - Anti-curriculum (ranking from most difficult to easiest)
RESULTS: LEARNING CURVE

Subset of CIFAR-100, with 5 sub-classes
RESULTS: different architectures and datasets, transfer curriculum always helps

Small CNN trained from scratch

Pre-trained competitive VGG

cats (from imagenet)
CURRICULUM HELPS MORE FOR HARDER PROBLEMS

3 subsets of CIFAR-100, which differ by difficulty
ADDITIONAL RESULTS

- Curriculum learning by-bootstrapping
  - Train current network (vanilla protocol)
  - Rank training data by final loss using trained network
  - Re-train network from scratch with CL

![Accuracy Bar Chart]

- vanilla
- curriculum
- anti
- random
- self-taught
1. Empirical study: curriculum learning in deep networks
   • Source of supervision: by-transfer, by-bootstrapping
   • Benefits: speeds up learning, improves generalization

2. Theoretical analysis: 2 simple convex loss functions, linear regression and binary classification by hinge loss minimization
   • Definition of “difficulty”
   • Main result: faster convergence to global minimum

3. Theoretical analysis: general effect on optimization landscape
   • Optimization function gets steeper
   • Global minimum, which induces the curriculum, remains the/a global minimum

⇒ theoretical results vs. empirical results, some mysteries
Theoretical analysis: linear regression loss, binary classification & hinge loss minimization

- **Theorem**: convergence rate is monotonically decreasing with the Difficulty Score of a point.

- **Theorem**: convergence rate is monotonically increasing with the loss of a point with respect to the current hypothesis*.

- **Corollary**: expect faster convergence at the beginning of training.

* when Difficulty Score is fixed
**Definitions**

- **ERM loss** $L_D(h) = \mathbb{E}_{X_t \sim D}(L(X_t, h))$

- **Definition:** point difficulty $\Leftrightarrow$ loss with respect to optimal hypothesis $\bar{h}$

  $$\Psi(X) = g(L(X, \bar{h}))$$

- **Definition:** transient point difficulty $\Leftrightarrow$ loss with respect to current hypothesis $h_t$

  $$\Upsilon(X) = g(L(X, h_t))$$

- $\lambda = \|\bar{h} - h_t\|_2$ \hspace{1cm} $\lambda_t = \|\bar{h} - h_{t+1}\|_2 = f(x)$

- $\Delta(\Psi, \Upsilon) = \mathbb{E}[\lambda^2 - \lambda_t^2]$
THEORETICAL ANALYSIS: LINEAR REGRESSION LOSS

- **Theorem**: convergence rate is **monotonically decreasing** with the Difficulty Score of a point $\Psi$
  
  Proof: \[ \frac{\partial \Delta(\Psi)}{\partial \Psi} \leq 0 \]

- **Theorem**: convergence rate is **monotonically increasing** with the loss of a point with respect to the current hypothesis $\gamma$
  
  Proof: \[ \frac{\partial \Delta(\Psi_0, \gamma)}{\partial \gamma} + O(\eta^2) \geq 0 \quad \forall \Psi_0 \]

- **Corollary**: expect faster convergence at the beginning of training (only true for regression loss)
  
  Proof: \[ \frac{\partial \Delta(\Psi)}{\partial \lambda} \geq 0 \quad \text{when} \quad \eta \leq \frac{\mathbb{E}[r^2 \cos^2 \vartheta]}{\mathbb{E}[r^4 \cos^2 \vartheta]} \]
**Matching Empirical Results**

- Setup: image recognition with deep CNN
- Still, average distance of gradients from optimal direction shows agreement with Theorem 1 and its corollaries
SELF-PACED LEARNING

- Self-paced is similar to CL, preferring easier examples, but ranking is based on loss with respect to the current hypothesis (not optimal).

- The 2 theorems imply that one should prefer easier points with respect to the optimal hypothesis, and more difficult points with respect to the current hypothesis.

⇒ Prediction: self-paced learning should decrease performance.
All conditions

- **Vanilla**: no curriculum
- **Curriculum**: transfer, ranking by inception
- **Controls**:
  - anti-curriculum
  - random
- **Self taught**: bootstrapping curriculum:
  - training data sorted after vanilla training
  - subsequently, re-training from scratch with curriculum
- **Self-Paced Learning**: ranking based on local hypothesis
OUTLINE

1. Empirical study: curriculum learning in deep networks
   - Source of supervision: by-transfer, by-bootstrapping
   - Benefits: speeds up learning, improves generalization

2. Theoretical analysis: 2 simple convex loss functions, linear regression and binary classification by hinge loss minimization
   - Definition of “difficulty”
   - Main result: faster convergence to global minimum

3. Theoretical analysis: general effect on optimization landscape
   - Optimization function gets steeper
   - Global minimum, which induces the curriculum, remains the/a global minimum

⇒ theoretical results vs. empirical results, some mysteries
**Effect of CL on Optimization Landscape**

- **Corollary 1**: with an ideal curriculum, under very mild conditions, the modified optimization landscape has the same global minimum as the original one.

- **Corollary 2**: when using any curriculum which is positively correlated with the ideal curriculum, gradients in the modified landscape are steeper than the original one.

![Optimization Function](image)
THEORETICAL ANALYSIS: OPTIMIZATION LANDSCAPE

Definitions:

- **ERM optimization:**
  \[ \mathcal{L}(\vartheta) = \mathbb{E}[L_{\vartheta}] = \frac{1}{N} \sum_{i=1}^{N} L_{\vartheta}(X_i) \]
  \[ \tilde{\vartheta} = \arg \min_{\vartheta} \mathcal{L}(\vartheta) = \arg \max_{\vartheta} \prod_{i=1}^{N} e^{-L_{\vartheta}(X_i)} \]

- **Empirical Utility/Gain Maximization:**
  \[ \mathcal{U}(\vartheta) = \mathbb{E}[U_{\vartheta}] = \frac{1}{N} \sum_{i=1}^{N} U_{\vartheta}(X_i) \triangleq \frac{1}{N} \sum_{i=1}^{N} e^{-L_{\vartheta}(X_i)} \]

- **Curriculum learning:**
  \[ \mathcal{V}(\vartheta) = \mathbb{E}_{p}[U_{\vartheta}] = \sum_{i=1}^{N} U_{\vartheta}(X_i)p(X_i) = \sum_{i=1}^{N} e^{-L_{\vartheta}(X_i)}p(X_i) \]

- **Ideal curriculum:**
  \[ p(X_i) = P(X_i|\tilde{\vartheta}) \propto P(\tilde{\vartheta}|X_i) \]
**Some results**

For any prior:

\[ \mathcal{V}(\vartheta) = \mathcal{U}(\vartheta) + \hat{\text{Cov}}[U_{\vartheta}, p] \]

For the ideal curriculum:

\[ \mathcal{V}(\vartheta) = \mathcal{U}(\vartheta) + \frac{1}{C} \text{Cov}[U_{\vartheta}, U_{\tilde{\vartheta}}] \]

which implies

\[ \mathcal{V}(\tilde{\vartheta}) - \mathcal{V}(\vartheta) \geq \mathcal{U}(\tilde{\vartheta}) - \mathcal{U}(\vartheta) \quad \forall \vartheta : \text{Cov}[U_{\vartheta}, U_{\tilde{\vartheta}}] \leq 0 \]

and generally

\[ \mathcal{V}(\tilde{\vartheta}) - \mathcal{V}(\vartheta) \geq \mathcal{U}(\tilde{\vartheta}) - \mathcal{U}(\vartheta) \quad \forall \vartheta : \text{Cov}[U_{\vartheta}, U_{\tilde{\vartheta}}] \leq \text{Var}[U_{\tilde{\vartheta}}] \]
**Remaining Unclear Issues**, when matching the theoretical and empirical results...

**Empirical findings**
- CL steers optimization to better local minimum
- Curriculum helps mostly at the beginning (one step pacing function)

**Theoretical results**
- Steeper landscape
- Predicts faster convergence at the end, anywhere in final basin of attraction
NO PROBLEM... IF LOSS LANDSCAPE IS CONVEX

Densenet121 (Tom Goldstein)
Back to the regression loss…

\[ L(\omega, (x, y)) = (\omega \cdot x - y)^2 \]

\[ s = \frac{\partial L(\omega)}{\partial \omega} \bigg|_{\omega=\omega_t} = 2 \ (\omega_t \cdot x - y) \ x \]

\[ \Delta = E[\|\omega_t - \bar{\omega}\|^2 - \|\omega_{t+1} - \bar{\omega}\|^2] \]
Computing the gradient step

difficulty score $\Psi/r^2$

\[ \Psi(X) = g(L(X, \bar{\omega})) \]

\[ \frac{1}{4} \Delta(\Psi) = \eta \mathbb{E}[r^2 \lambda^2 \cos^2 \vartheta] - \eta^2 \mathbb{E}[r^4 \lambda^2 \cos^2 \vartheta] - \eta^2 \Psi^2 \mathbb{E}[r^2] \]
**Theoretical Analysis: Linear Regression Loss**

- **Theorem**: convergence rate is monotonically decreasing with the *Difficulty Score* of a point.
  
  Proof: \( \frac{\partial \Delta(\Psi)}{\partial \Psi} \leq 0 \)

- **Theorem**: convergence rate is monotonically increasing with the *loss* of a point with respect to the *current hypothesis*.

- **Corollary**: expect faster convergence at the beginning of training (only true for regression loss)
  
  Proof: \( \frac{\partial \Delta(\Psi)}{\partial \lambda} \geq 0 \) when \( \eta \leq \frac{\mathbb{E}[r^2 \cos^2 \vartheta]}{\mathbb{E}[r^4 \cos^2 \vartheta]} \)
Theoretical analysis: Linear regression loss

- **Theorem**: convergence rate is monotonically decreasing with the *Difficulty Score* of a point.

- **Theorem**: convergence rate is *monotonically increasing* with the *loss* of a point with respect to the *current hypothesis*.

- **Corollary**: expect faster convergence at the beginning of training (only true for regression loss)
LOSS WITH RESPECT TO CURRENT HYPOTHESIS

\[
\gamma(X) = g(L(X, \omega_t))
\]

\[
\frac{1}{4\eta} \Delta(\Psi_0, \gamma) = \Psi_0^2 + \gamma^2 + 2\Psi_0 \gamma \nabla
\]

\[
\nabla = \frac{f(\frac{\Psi+\gamma}{\lambda}) - f(\frac{\Psi-\gamma}{\lambda}) - f(\frac{-\Psi+\gamma}{\lambda}) + f(\frac{-\Psi-\gamma}{\lambda})}{f(\frac{\Psi+\gamma}{\lambda}) + f(\frac{\Psi-\gamma}{\lambda}) + f(\frac{-\Psi+\gamma}{\lambda}) + f(\frac{-\Psi-\gamma}{\lambda})}
\]

**Theorem** Assume that the gradient step size is small enough so that we can neglect second order terms \(O(\eta^2)\), and that \(\frac{\partial \nabla}{\partial \gamma} \geq \frac{\Psi}{\Psi} - \frac{\gamma}{\Psi} \forall \gamma\). Fix the difficulty score at \(\Psi_0\). At time \(t\) the expected convergence rate is monotonically increasing with the local difficulty \(\gamma(x)\).

**Corollary** For any \(c \in \mathbb{R}^+\), if \(\nabla\) is \((c - \frac{1}{c})\)-Lipschitz then \(\frac{\partial \Delta(\Psi, \gamma)}{\partial \gamma} \geq 0\) for any \(\gamma \geq c \Psi\).
Hinge loss

\[ L(X, w) = \max(1 - (x \cdot w)y, 0) \]

\[ \Delta(\Psi) = \mathbb{E} \left[ \frac{w_{t+1} \cdot \bar{w}}{\|w_{t+1}\|\|\bar{w}\|} - \frac{w_t \cdot \bar{w}}{\|w_t\|\|\bar{w}\|} \right] \Psi \]

\[ = \int_{-\infty}^{B(\Psi)} \eta[(1 - \Psi) \sin^2 \vartheta - x_2 \sin \vartheta \cos \vartheta] \cdot f(x_2) dx_2 + O(\eta^2) \]

**Theorem** Assume that the gradient step size is small enough so that we can neglect second order terms \(O(\eta^2)\). The expected convergence rate decreases monotonically as a function of \(\Psi\) for every \(\Psi > (1 - \cos \vartheta)\) when \(\cos \vartheta > 0\) (\(\bar{w}, w_t\) are positively correlated), and for every \(\Psi < (1 - \cos \vartheta)\) when \(\cos \vartheta < 0\). Monotonicity holds \(\forall \Psi\) when \(\cos \vartheta = 0\).

**Theorem** Assume that the gradient step size is small enough so that we can neglect second order terms \(O(\eta^2)\). Assume further that \(\cos \vartheta \geq 0\). Fixing \(\Psi\) and \(\forall \Psi\), the expected convergence rate is monotonically increasing with \(\gamma\) for every \(\gamma > 0\).
SUMMARY AND DISCUSSION

1. First theoretical demonstration that curriculum learning indeed helps, speeding up convergence during training. Previous related results have relied mostly on empirical evidence.

2. The literature is confusing, with 2 apparently conflicting methods:
   - Curriculum learning, giving preference to easier examples
   - Methods like hard example mining and boosting, which focus on the more difficult examples

   Resolution: results are consistent, it’s all in how one measures difficulty:
   - Curriculum: Easy, with respect to final hypothesis.
   - Hard example mining: Difficult, with respect to current hypothesis.

3. Curriculum learning made practical:
   - CL by transfer: source network, which is bigger and more powerful, is used to sort the examples for the weaker network.
   - CL by bootstrapping: same pre-trained network is used to sort the examples
This Research is supported by the Israeli Science Foundation, the Gatsby Charitable Foundation, and Mafat Center for Deep Learning.