Continuous Integration of Machine Learning Models with ease.ml/ci

Towards a Rigorous Yet Practical Treatment

Cedric Renggli (ETH Zurich), Bojan Karlaš (ETH Zurich), Bolin Ding (Data Analytics and Intelligence Lab, Alibaba Group), Feng Liu (Huawei Technologies), Kevin Schawinski (Modulos AG), Wentao Wu (Microsoft Research), Ce Zhang (ETH Zurich)
Past Work: Speed & Automation

The Tradeoffs of Large Scale Learning

Linus Butter
NEC Laboratories America
Princeton, NJ, USA
linus@nec-labs.org

Abstract

Since its creation, the ImageNet IBS benchmark set has played a significant role in benchmarking the accuracy of different deep neural net (DNN) models on the ImageNet dataset. In this paper, we describe our efforts towards creating an open-source tool to evaluate the performance of different approaches to DNN training.

Poster: "ImageNet in Minutes"

Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, Kurt Keutzer
UC Berkeley
ACM/IEEE

ImageNet Training in Minutes


c

Microsoft Azure ML

Our own small Prototypes
Observation

If some of our users are not careful, they are left with nothing else than a more powerful “overfitting machine”.

Let’s provide some guidelines for proper ML systems usage!
ease.ml/ci - Overview

What is hard about this?

1. **Rigorous** guaranties, but as **cheap** as possible.

2. Leaking information at every commit implies **Adaptive Analytics**.

Our results:

- **Statistically sound** estimators to reduce sample (and label) complexity of the testset by **1 - 2 order of magnitude**.
System Overview

1. Specify Requirements
   - e.g., all models checked in should have accuracy > 0.8 \((\epsilon, \delta)\)-approximation.

2. Commit a stream of \(T\) models

3. Receive Pass/Fail signal per commit

4. Ask for \(n\) test labels when it needs more

5. When test labels lose statistical power, downgrade to val set and let developers know

ML Repo (e.g., Github)

- Public
- Encryption - Protected

Manager

Developer
Managers Specify Requirements

R1: New model needs to be better than the old model by at least 1%, with probability 0.999.

\[ n - o > 0.01, \quad p > 0.999 \]

R2: New model cannot be different from the old model on more than 10% of predictions, with probability 0.999.

\[ d < 0.1, \quad p > 0.999 \]

R3: New model always have accuracy higher than 0.8, with probability 0.999.

\[ n > 0.8, \quad p > 0.999 \]

R4: Satisfy both R1 and R2, with probability 0.999.

\[ n - o > 0.01 \text{ and } d < 0.1, \quad p > 0.999 \]
Developers Task

Develop a ML model and **commit**.
Developers Task

Develop a new ML model and **recommit**.
Core Technical Component:

Adaptive Statistical Queries

We are inspired by the following seminal work:

- The ladder: A reliable leaderboard for machine learning competitions. Blum and Hardt, 2015
- The algorithmic foundations of differential privacy. Dwork et. al., 2014
- The reusable holdout: Preserving validity in adaptive data analysis. Dwork et. al., 2015
Background: Adaptive Analytics

Contract between System and User:
\[ \Pr[\exists t, |f_t(X_1, \ldots, X_n) - f_t(X)| > \epsilon] < \delta \]

Given \( \epsilon, \delta, T \), how large does \( n \) need to be?

How can we decrease the dependency of \( n \) on \( \epsilon, \delta, T \) as much as possible?

\( i.i.d \) samples: \( X_1, X_2, X_3, \ldots, X_n \sim X \)

[(un)Labeled Samples from Test]

Encryption

Developer

\[ g(f_1(\{X_i\})) \]

\[ f_1 \]

\[ g(f_2(\{X_i\})) \]

\[ f_2 \]

\[ \cdots \]

\( \cdots \)

\[ g(f_T(\{X_i\})) \]

\[ f_T \]
Background: Single Steps – Hoeffding’s Inequality

Theorem (Hoeffding, 1963):
Let $X_1, X_2, \ldots, X_n$ be i.i.d random variables with
\[ \forall X_i \ 0 \leq X_i \leq 1 \text{ and } \overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i : \]
Then $\forall \epsilon$
\[ \Pr \left[ \overline{X} - \mathbb{E}[X] \geq \epsilon \right] \leq \exp(-2n\epsilon^2). \]

\[ \delta \leq \exp(-2n\epsilon^2) \quad \rightarrow \quad n \geq \frac{\ln \frac{1}{\delta}}{2\epsilon^2} \]
Background: Multiple Steps – Existing Solutions

\[ f_2(\{X_i\}) = h_g(f_1(\{X_1, X_2, \ldots, X_n\})) (\{X_i\}) \]

Baseline Approach: Resampling

Require a new sample for each step.

Ladder (Blum and Hardt, 2015)

Constrains how \( g(\cdot) \) evolves over time.

Other DP - inspired approaches

\[ \epsilon = 0.01 \quad \delta = 0.001 \quad T = 32 \]

Expensive: \(~53K / Day\)

\[ n \geq T - \frac{\ln \delta}{2e^2} \approx 1.7M \]

\[ n \geq 69K \]

\( g(\cdot) \) is non-monotonic

Unclear how to add noise to \( g(\cdot) \) in CI

Goal: Optimizing Sample Complexity for the specific regime that our system cares about.
Overview of Optimizations

Goal: Optimizing Sample Complexity for the specific regime that our system cares about.

1) General Optimization
2) Stable Signal
3) Conditional Variance
4) Active Labeling
Adaptive Analytics - Observation 1

Observation 1: The Most Trivial Approach is Not That Bad

- We know \( g(\cdot) \) returns a binary signal.
- \# of possible functions for \( T \) binary signals \( \leq 2^T \)
- Apply union bound on all possible functions.

\[
\frac{\delta}{2^T} \leq \exp \left( -2n\epsilon^2 \right) \quad \Rightarrow \quad n \geq \frac{T \ln \frac{2 - \ln \delta}{2 \epsilon^2}}{2 \epsilon^2}
\]

Still order \( O(T) \)

\[
\begin{align*}
\epsilon &= 0.01 \\
\delta &= 0.001 \\
T &= 32 \\
n \geq T \frac{-\ln \frac{\delta}{T}}{2 \epsilon^2} &\approx 1.7M \\
n \geq \frac{T \ln(2) - \ln \delta}{2 \epsilon^2} &\approx 145K
\end{align*}
\]
Adaptive Analytics - Observation 2

Observation 2: Conditional Variance Bound

The most popular condition used in ease.ml/ci:

\[ n - o > 0.01 \quad \text{and} \quad d < 0.1, \quad p > 0.999 \]

The new model only makes different predictions on at most 10% of data points compared to the old model.

The new model is better than the old model by at least 1 percentage point.

Observation 2.1: \( d < 0.1 \) does not need labels.

Observation 2.2: Conditioned on \( d < 0.1 \), \( n - o \) has small variance.
Adaptive Analytics - Observation 2

Observation 2: Conditional Variance Bound

Theorem (Bennett, 1962):

Let $X_1, X_2, \ldots, X_n$ be i.i.d random variables with
\[
\forall X_i \mid X_i \leq 1, \quad \sum_{i=1}^{n} \mathbb{E}[X_i^2] = \sigma^2 \quad \text{and} \quad S_n = \sum_{i=1}^{n} X_i : 
\]

Then $\forall \epsilon$
\[
\Pr \left[ \frac{S_n - \mathbb{E}[X_i]}{n} \geq \epsilon \right] \leq \exp \left( -\sigma^2 h \left( \frac{n\epsilon}{\sigma^2} \right) \right),
\]

with $h(u) = (1 + u) \ln(1 + u) - u$ for $u > 0$.

\[
\begin{align*}
\epsilon &= 0.01 \\
\delta &= 0.001 \\
T &= 32
\end{align*}
\]

\[
\begin{align*}
\text{Baseline} & \quad \sim 7.5 \text{ M} \\
\text{Union Bound} & \quad \sim 609 \text{ K} \\
\text{Benett} & \quad \sim 63 \text{ K}
\end{align*}
\]
Adaptive Analytics - Observation 3

Observation 3: Not all labels are useful

Focus: \( n - o > 0.01, \ p > 0.999 \)

Old Model: 0 1 1 1 0
New Model: 0 1 1 0 1

Same predictions – Not useful to estimate the difference

If new models and old models are only different in their prediction with probability \( \nu \), how many savings can we have in terms of labels (NOT SAMPLES) that we need to provide?

If the probability of two models being different is \( \nu \sim O(\sqrt{\epsilon}) \), than the amount of labels we need is \( n \geq O(1/\epsilon) \).

Hoeffding

\( \nu = 0.1 \)

15K samples/signal

2.2K samples/signal

(Assuming unlabeled data points are free)
Ease.ml/ci in Action

$ git commit -m newmodel

Popular Use Cases: \((\varepsilon = 0.0125)\)

- \(n - o > 0.01\) and \(d < 0.1\)
- \(n > 0.8\)

Cheap Mode: \((\varepsilon = 0.025)\)

- \(n - o > 0.01\) and \(d < 0.1\)
- \(n > 0.8\)

# of Labels/32 Models

- **Baseline**
  - 4.8M (150K / Day)
  - 1.1M (35K / Day)
  - 1.2M (38K / Day)
  - 283K (8.9K / Day)

- **Ease.ml/ci**
  - 41K (1.3K / Day)
  - 95K (3K / Day)
  - 11K (330 / Day)
  - 24K (745 / Day)

Baseline 10s / Label

300 Labels / Day => < 1 Hour / Day
Ongoing Projects

```bash
$ git commit -m newmodel
```

If ML is “Software 2.0”, what are the missing principles in “Software Engineering 2.0”?

Release of both Systems planned this Summer