ease.ml ci

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*)

ETH Systems@ETH Zürich Alibaba Group HUAWEI modulos.ai Microsoft

Past Work: Speed & Automation

ease.ml ci

The Tradeoffs of Large Scale Learning

Léon Bottou
NEC laboratories of Americ
Princeton, NJ 08540, USA
leon@bottou.org

Accurate, Large Minibatch SGD:
Training ImageNet in 1 Hour

Priya Goyal Piotr Dollár Ross Girshick Pieter Noordhuis
ulloch Yangqing Jia Kaiming He

ImageNet Training in Minutes

Yang You[1], Zhao Zhang[2], Cho-Jui Hsieh[3], James Demmel[1], Kurt Keutzer[1]
UC Berkeley[1], TACC[2], UC Davis[3]
{youyang, demmel, keutzer}@cs.berkeley.e

[cs.CV] 31 Jan 2018

Abstract

Since its creation, the ImageNet-1k benchmark set has played a significant role as a benchmark for ascertaining the accuracy of different deep neural net (DNN) models on the classification problem. Moreover, in recent years it has also served as the principal benchmark for assessing different approaches to DNN training. Finishing a 90-epoch ImageNet-1k training with ResNet-50 on a NVIDIA M40 GPU takes 14 days. This training requires 10^{18} single precision operations in total. On the other hand, the world's current fastest supercomputer can finish 2×10^{17} single precision operations per second. If we can make full use of the computing capability of the fastest supercomputer for DNN training, we should be able to finish the 90-epoch ResNet-50 training in five seconds. Over the last two years, a number of researchers have focused on closing this significant performance gap through scaling DNN training to larger numbers of processors. Most successful ap-

Highly Scalable Deep Learning Training System with Mixed-Precision: Training ImageNet in Four Minutes

Xianyan Jia[*1], Shutao Song[*1], Wei He[1], Yangzihao Wang[1], Haidong Rong[1], Feihu Zhou[1], Liqiang Xie[1], Zhenyu Guo[1], Yuanzhou Yang[1], Liwei Yu[1], Tiegang Chen[1], Guangxiao Hu[1], Shaohuai Shi[*2], Xiaowen Chu[2]
Tencent Inc.[1], Hong Kong Baptist University[2]
{xianyanjia, sampsonsong, winsonwhe, slashwang, hudsonrong, hopezhou, felixxie, alexguo, jokeyang, leolwyu, steelchen, gethinhu}@tencent.com;
{csshshi, chxw}@comp.hkbu.edu.hk
*Authors contributed equally

[cs.LG] 30 Jul 2018

ABSTRACT

Synchronized stochastic gradient descent (SGD) optimizers with data parallelism are widely used in training large-scale deep neural networks. Although using larger mini-batch sizes can improve the system scalability by reducing the communication-to-computation ratio, it may hurt the generalization ability of the models. To this end, we build a highly scalable deep learning training system for dense GPU clusters with three main contributions: (1) We propose a mixed-precision training method that significantly improves the training throughput of a single GPU without losing accuracy. (2) We propose an optimization approach for extremely large mini-batch size (up to 64k) that can train CNN models on the ImageNet dataset without losing accuracy. (3) We propose highly optimized

to use large batch to achieve weak scaling [2, 4, 5, 12, 23, 27]. In this way, the speedup is obtained by utilizing overall throughput of the system and fewer updates of the model.

However, there are two challenges when using large batch across large clusters:

- **Challenge 1**: Larger mini-batch size often leads to lower test accuracy, as there exists a generalization gap [15].
- **Challenge 2**: When using large clusters, it is harder to achieve near-linear scalability as the number of machines increases, especially for models with the high communication-to-computation ratio.

Challenge 1. Larger mini-batch size reduces the variance of

Alibaba Cloud

AUTO KERAS

Azure ML

ML.NET

Amazon SageMaker

Cloud AutoML

Our own small Prototypes

ZipML ease.ml

Are ML Systems "Usable"?

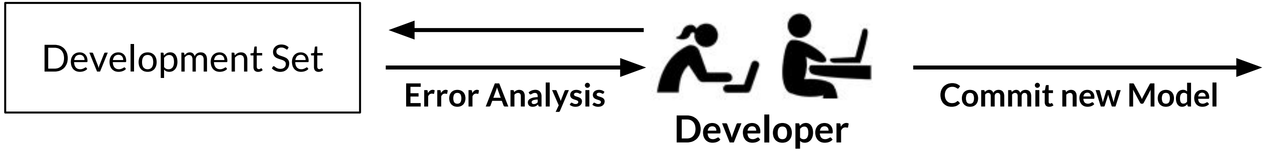
ease.ml ci

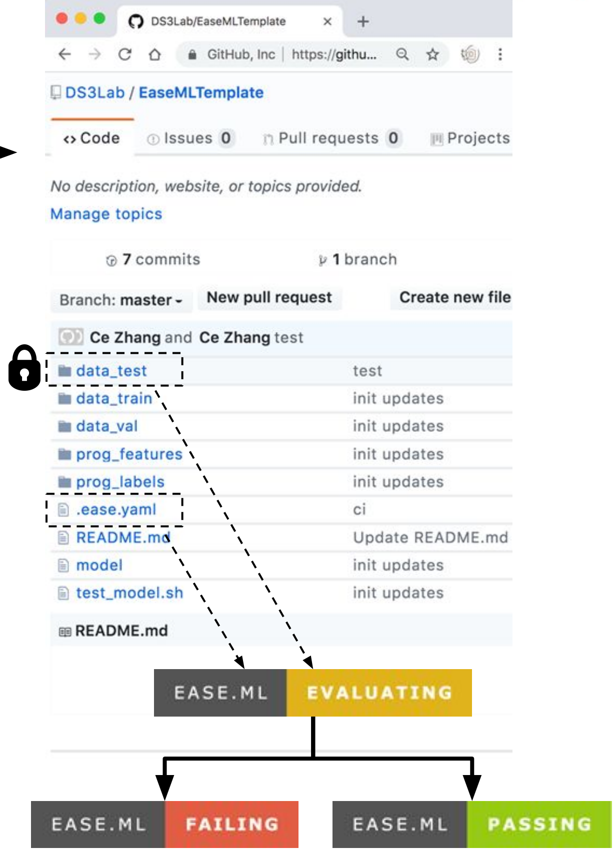
Observation

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





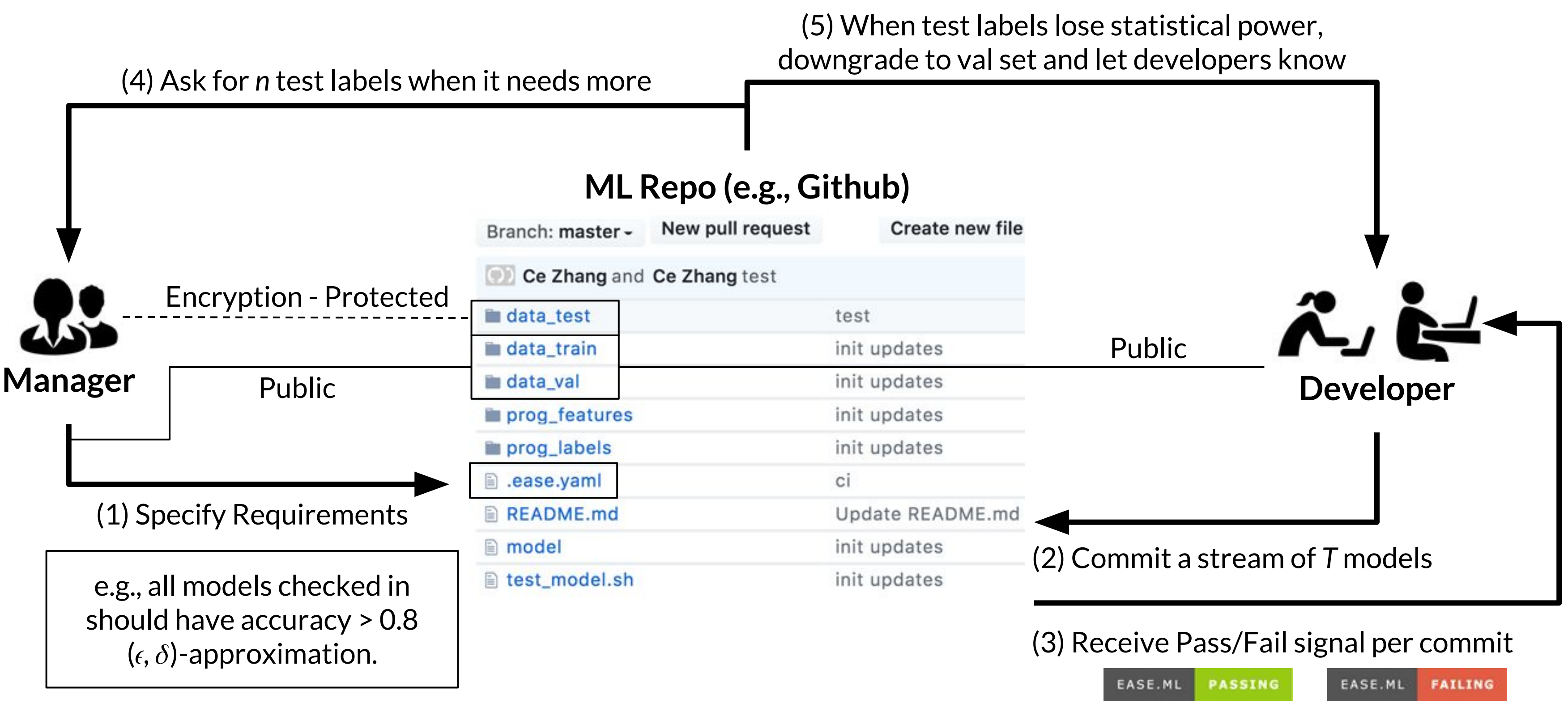
What is hard about this?

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



Managers Specify Requirements

Manager

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

$n - o > 0.01,\ p > 0.999$

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

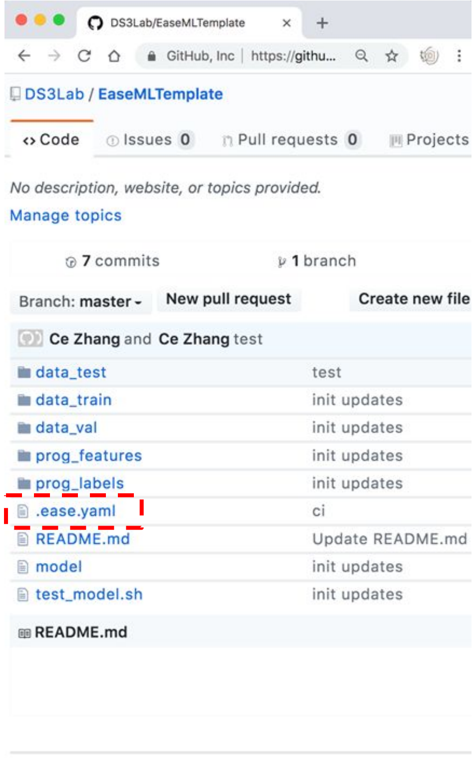
$d < 0.1,\ p > 0.999$

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

$n > 0.8,\ p > 0.999$

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

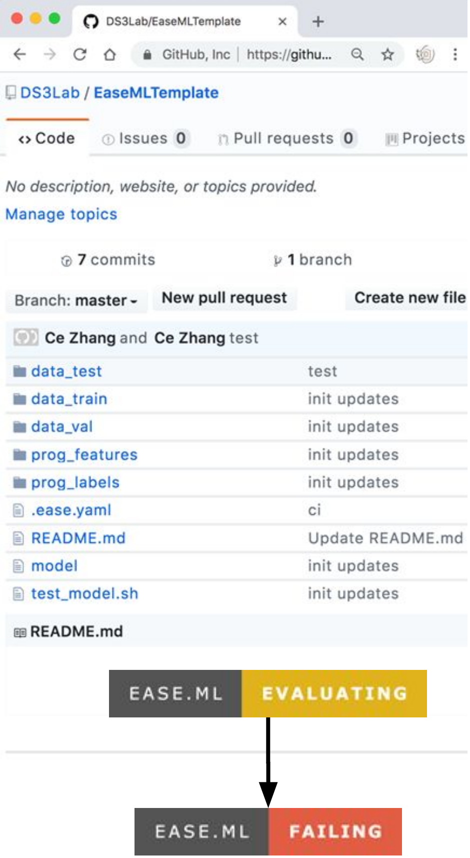
$n - o > 0.01$ **and** $d < 0.1,\ p > 0.999$



Developers Task

Developer

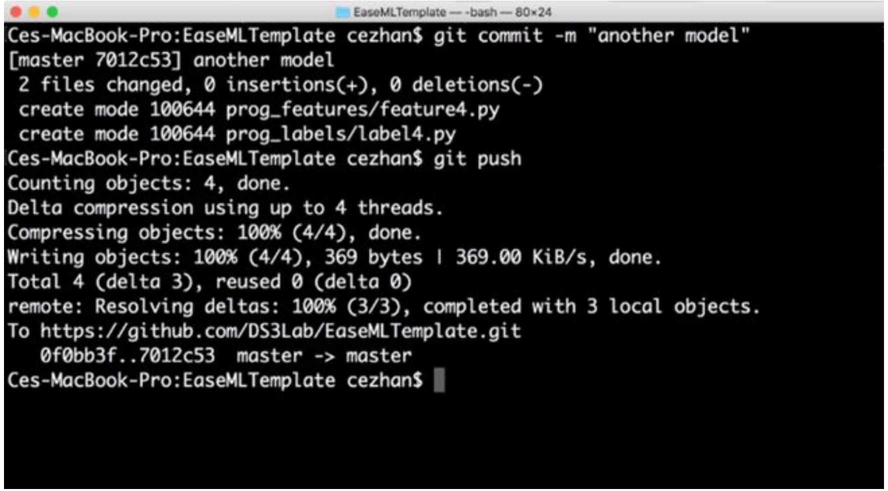
Develop a ML model and commit.

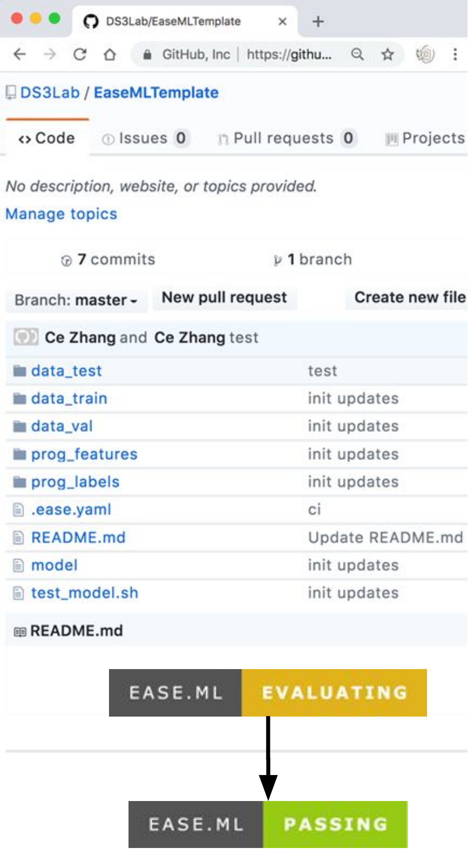


Developers Task

Developer

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

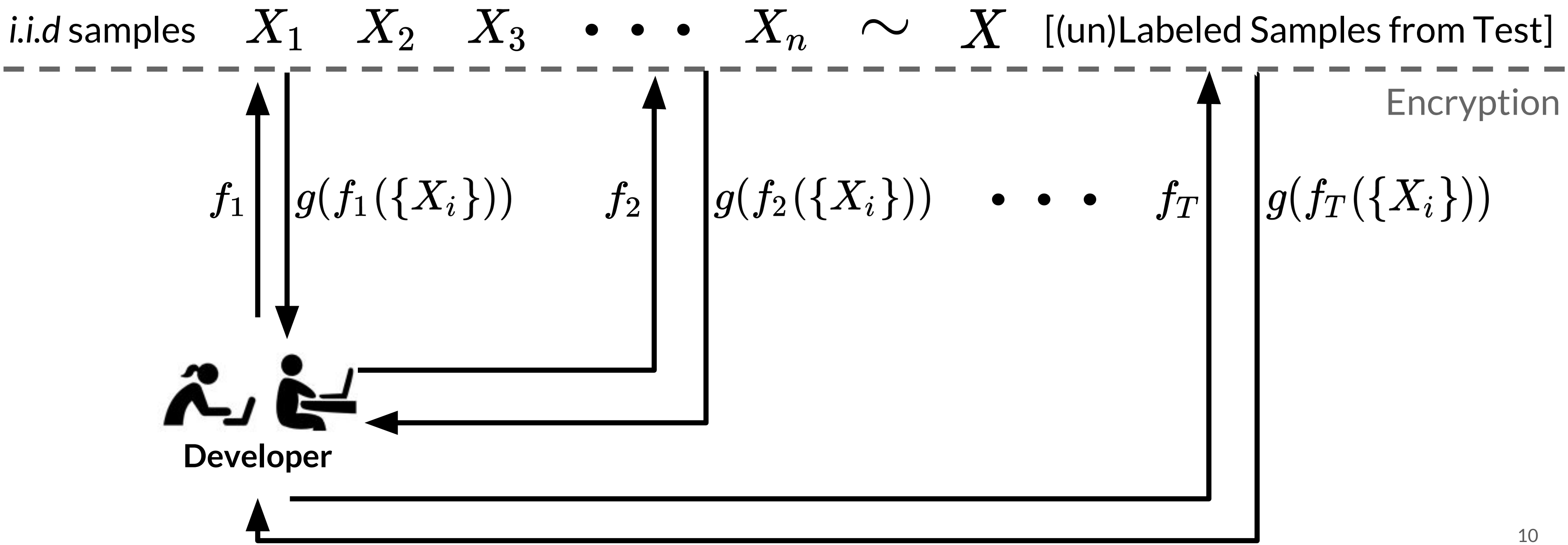
Background: Adaptive Analytics

Contract between System and User:

$\Pr\left[\exists t, |f_t(X_1, \ldots, X_n) - f_t(X)| > \epsilon\right] < \delta$

Given ε, δ, *T*, how large does *n* need to be?

How can we decrease the dependency of *n* on ε, δ, *T* as much as possible?



Background: Single Steps – Hoeffding's Inequality

ease.ml ci

Theorem (Hoeffding, 1963):

Let $X_1, X_2, \ldots, X_n$ be i.i.d random variables with

$\forall X_i \; 0 \leq X_i \leq 1$ and $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$:

Then $\forall \epsilon$

$$\Pr\left[\overline{X} - \mathbb{E}[\overline{X}] \geq \epsilon\right] \leq \exp(-2n\epsilon^2).$$

$$\delta \leq \exp(-2n\epsilon^2) \longrightarrow n \geq \frac{\ln\frac{1}{\delta}}{2\epsilon^2}$$

Background: Multiple Steps – Existing Solutions

ease.ml ci

$$f_2(\{X_i\}) = h_{g(f_1(\{X_1, X_2, \ldots, X_n\}))}(\{X_i\})$$

Baseline Approach: Resampling	Ladder (Blum and Hardt, 2015)	Other DP - inspired approaches
Require a new sample for each step.	*Constrains how g(−) evolves over time.*	

$$\epsilon = 0.01$$
$$\delta = 0.001$$
$$T = 32$$

$$n \geq T\frac{-\ln\frac{\delta}{T}}{2\epsilon^2} \approx 1.7M$$

Expensive: ~53K / Day

$$n \geq 69K$$

g(-) is non-monotonic

Unclear how to add noise to g(-) in CI

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

Overview of Optimizations

ease.ml ci

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(−)* returns a binary signal.
- # of possible functions for *T* binary signals $\leq 2^T$
- Apply union bound on all possible functions.

$$\frac{\delta}{2^T} \leq \exp\left(-2n\epsilon^2\right) \longrightarrow n \geq \frac{T\ln 2 - \ln\delta}{2\epsilon^2}$$

Still order O(T)

Baseline **Union Bound**

$$\epsilon = 0.01 \quad \delta = 0.001 \quad T = 32$$

$$n \geq T\frac{-\ln\frac{\delta}{T}}{2\epsilon^2} \approx 1.7M \qquad n \geq \frac{T\ln(2) - \ln\delta}{2\epsilon^2} \approx 145K$$

Adaptive Analytics - Observation 2

Observation 2: Conditional Variance Bound

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

```
n - o > 0.01 and d < 0.1, 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 $\forall X_i \; |X_i| \leq 1, \sum_{i=1}^{n} \mathbb{E}[X^2] = \sigma^2$ and $S_n = \sum_{i=1}^{n} X_i$:

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 $u > 0$.

	Baseline	Union Bound	Benett
$\epsilon = 0.01$ $\delta = 0.001$ $T = 32$	~7.5 M	~609 K	~63 K

Adaptive Analytics - Observation 3

ease.ml ci

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 $\boldsymbol{\nu}$, how many savings can we have in terms of labels (NOT SAMPLES) that we need to provide?

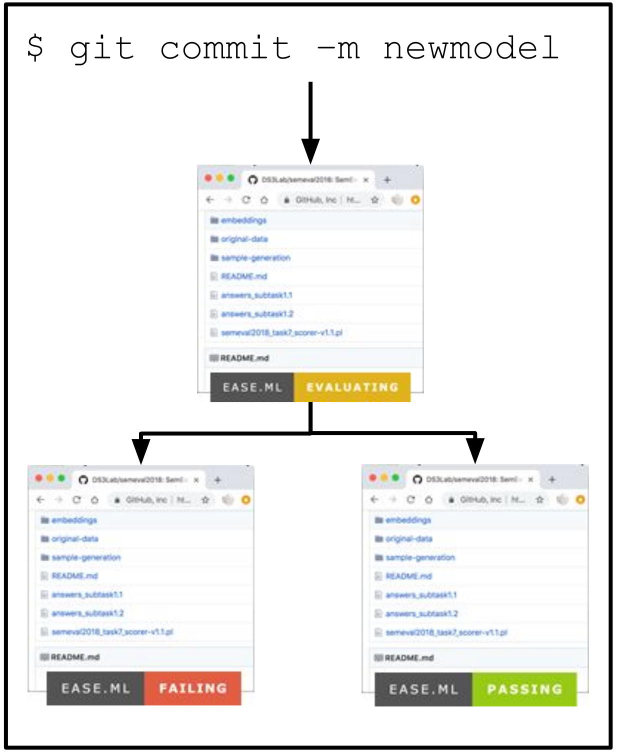
If the probability of two models being different is $\boldsymbol{\nu} \sim O(\sqrt{\varepsilon})$, than the amount of labels we need is $n \geq O(1/\varepsilon)$.

Hoeffding	**15K** samples/signal
$\boldsymbol{\nu} = 0.1$	**2.2K** samples/signal

(Assuming unlabeled data points are free)

`ease.ml/ci` in Action

ease.ml/ci



of Labels/32 Models

	Baseline	`ease.ml/ci`
Popular Use Cases: ($\epsilon = 0.0125$)		
`n - o > 0.01 and d < 0.1`	**4.8M** (150K / Day)	**41K** (1.3K / Day)
`n > 0.8`	**1.1M** (35K / Day)	**95K** (3K / Day)
Cheap Mode: ($\epsilon = 0.025$)		
`n - o > 0.01 and d < 0.1`	**1.2M** (38K / Day)	**11K** (330 / Day)
`n > 0.8`	**283K** (8.9K / Day)	**24K** (745 / Day)

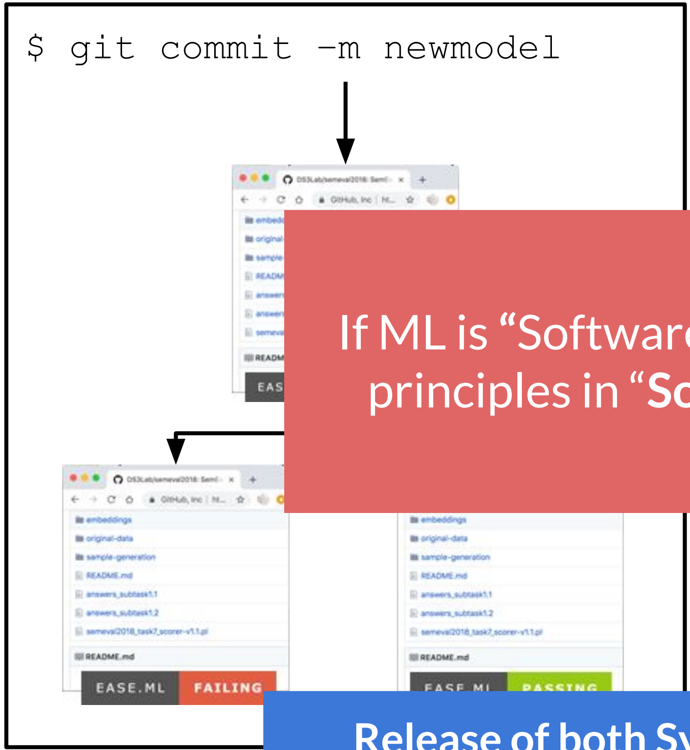
10s / Label

300 Labels / Day => < 1 Hour / Day

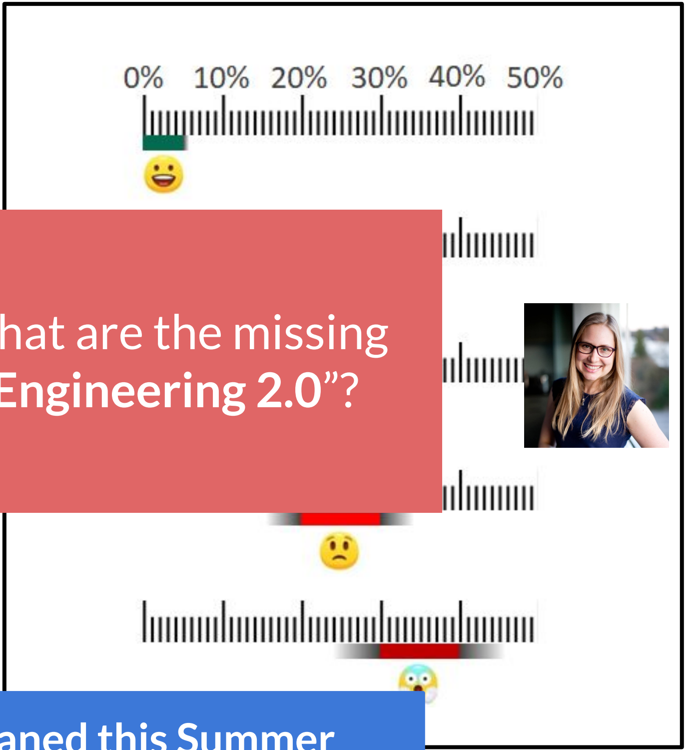
Ongoing Projects

ease.ml ci

ease.ml/ci



ease.ml/meter



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

Release of both Systems planed this Summer