TrimTuner: Efficient Optimization of Machine Learning Jobs in the Cloud via Sub-Sampling

Pedro Mendes[1], Maria Casimiro[1,2], Paolo Romano[1], David Garlan[2]

[1] INESC-ID and Instituto Superior Técnico

[2] Institute of Software Research, Carnegie Mellon University

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

Contents

- Motivation
- TrimTuner
- Evaluation
- Ongoing Work
- Final Remarks

Training large models

- In recent years:
  - Machine Learning (ML) is ubiquitous
  - **ML models** are increasing in **complexity**
  - **Data-set grow larger and larger**
- Training process involves **enormous** amount of **computational resources**

Training large models on the cloud

- **Cloud** provides access to:
  - "unlimited" **on-demand resources**
  - a **large spectrum of Virtual Machines** (VMs) with different types, sizes, and prices

...that can be used to train ML models

(it represents the standard approach nowadays)

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

Cloud Cost & Ecological Footprint

	Energy Consumption (kWh)	CO2 Emissions (Kg of CO2e)	Cloud Cost (USD)
Transformer (65M parameters) (Jun,17)	27	12	$41-$140
Transformer (213M parameters) (Jun,17)	201	87	$289-$981
ELMo (Feb,18)	275	119	$433-$1,472
BERT (110M parameters) (Oct,18)	1507	652	$3,751-$12,571
Transformer (213M parameters) (NAS) (Jan, 19)	656347	284019	$942,973-$3,201,722
GPT-2 (Feb, 19)	-	-	$12,902-$43,008

Source: Energy and Policy Considerations for Deep Learning in NLP, Strubell et al. (2019)

Which cloud & model configuration to use?

Large spectrum of VMs

Several hyper-parameters defined over large domains

Testing even a single **configuration** can be very **expensive** with current **large data-sets**

Joint optimization of cloud & model parameters

- **Disjoint optimization** is unable to identify optimal configurations:
  - up to 3.7x more expensive configurations with Neural Network [ICDCS20]

- The wrong configuration selection can significantly **reduce model's accuracy** and **amplify operational costs**

Fundamental to **joint optimize** cloud and application parameter

Yielding vast search spaces

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

TrimTuner

System to optimize the training of **ML jobs** in the **cloud**

Exploits **sub-sampling** techniques to enhance the efficiency of the optimization process

Jointly optimizes the cloud configuration and hyper-parameters of ML models

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

TrimTuner - Optimization Problem

$$\underset{\mathbf{x}\in\mathbb{X}}{\text{maximize}} \quad A(\mathbf{x}, s=1)$$
$$\text{subject to} \quad q_1(\mathbf{x}, s=1) \geq 0, \ldots, q_m(\mathbf{x}, s=1) \geq 0$$

x: configuration s: data-set size [%] $\mathbb{X}$: set of configurations to test

A(**x**,s): accuracy of the ML model

q_i (**x**,s): Quality of Service (QoS) constraints (e.g., cost or execution time)

TrimTuner

- Deploy the job using **sub-sampled data-sets to reduce the cost** of testing individual configurations

- **Build predictive models** that keep into account how shifts of the data-set size affect the accuracy and cost of the target job

TrimTuner

Leverages **Bayesian Optimization** (BO) techniques to update the models and select the next configuration to evaluate

Does not assume **any *a priori* knowledge** about the target job and the training platform

Bayesian Optimization: base idea

- Aims at identifying the optimum of an unknown function *f*
- Does not assume any *a priori* knowledge about the target job
- Builds black-box model of *f*

BO: Algorithm

1. Test a set of initial configurations
2. Build a model of f
3. $\mathbf{x}_i = \arg\max_{\mathbf{x} \in \mathbb{X}} \{\alpha(\mathbf{x})\}$
   $\mathbb{X}$: untested set
   α: acquisition function
4. Test the configuration $\mathbf{x}_i$
5. Update the model of f

BO: Model

Typically, Gaussian Processes are used in BO:

- Provides not only information about a configuration, but also its uncertainty (the prediction follows a normal distribution)
- Analytical tractability and flexibility (the covariance function can be chosen)
- Continuous and discrete

As we are going to see next,
GPs have a high computational cost

BO: Model

Alternatively, it is possible to use an ensemble of learners:

- Predicted value - determines the average of the prediction of each learner
- Uncertainty - calculate the standard deviation using the predictions of each learner

TrimTuner uses Ensemble of Decision Trees (DTs)

BO: Acquisition Function

Exploits the model's knowledge and uncertainty to determine which configuration to evaluate next by balancing:

- **Exploitation** - recommend configs that the model deems to be optimal
- **Exploration** - recommend configs whose knowledge can reduce the model's uncertainty and enhance its accuracy

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

BO: Acquisition Function

- **Expected Improvement:**
  - Estimates the expected amount by which evaluating f at x can improve over the current best value (or incumbent)

$$\alpha_{EI}(\mathbf{x}) = \int max(0, f(\mathbf{x}) - \eta) p(f(\mathbf{x})|\mathcal{S}) df(\mathbf{x})$$

- **Entropy Search:**
  - Estimates the Information that can be gained about the optimum by evaluating f at x

BO Problem:

Testing even a single configuration can be **very costly** with large data sets

BO using sub-sampling: FABOLAS [AISTATS17]

- Aims to maximize ML model accuracy by optimizing its hyper-parameters
- Leverages BO to solve the optimization problem
- Trains the model with sub-sampled data-sets to reduce the cost of testing configurations

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

BO + sub-sampling: base idea

Additional dimension in the search space that corresponds to the data-set size

1

Test a set of initial configs x in all the sub-sampled data-set sizes

2

Build a model of *f*

3

$(\mathbf{x}_i, s_i) = \arg\max_{\mathbf{x} \in \mathbb{X},\, s \in S} \{a(x, s)\}$

$\mathbb{X}$: untested set

S: sub-sampled data-set sizes

a: acquisition function

4

Test the config $(\mathbf{x}_i, s_i)$

5

Update the model of *f*

FABOLAS Acquisition Function:

Information gain per unit cost

- **Entropy Search** [JMLR12]: estimates the information on the optimum using the full data-set (s=1) that can be gained by testing a sub-sampled configuration (x,s), where s<1.
- ...normalized by the cost of testing (x,s)

$$\alpha_F(\mathbf{x}, s) = \frac{1}{C(\mathbf{x},s)} \mathbb{E}_{p(y|\mathbf{x},s,\mathcal{S})} \left[\int p_{opt}^{s=1}(\mathbf{x}'|\mathcal{S} \cup \{\mathbf{x}, s, y\}) \log \frac{p_{opt}^{s=1}(\mathbf{x}'|\mathcal{S} \cup \{\mathbf{x}, s, y\})}{u(\mathbf{x}')} d\mathbf{x}' \right]$$

Normalize by the cost of testing (x,s)

Entropy Search

FABOLAS: Limitations

Does not keep into account any QoS related constraints

e.g., on model's training cost or latency.

Only optimizes the hyper-parameters of the ML model

Underlooks the problem of **jointly optimizing** cloud parameters

Smaller search spaces & globally sub-optimal recommendations

The acquisition function is very expensive to compute

Problem exacerbated with **large search** spaces (up to 13 minutes to recommend a configuration (Search space: 1440 configs))

TrimTuner

Novel acquisition function to keep into account QoS constraints

Optimizes both the hyper-parameters and cloud resources

Lightweight models based on Decision Trees (vs GPs)

Novel filtering heuristic to reduce the number of configurations to compute the acquisition function

TrimTuner: acquisition function

- Determines the probability that the new optimum is feasible
- Estimates the information gain of evaluating (x,s) about the optimum
- Normalizes by the cost of evaluating (x,s).

$$\alpha_T(\mathbf{x}, s) = \mathbb{E}_{p(\mathbf{q},a|\mathbf{x},s,\mathcal{S})}\left[\prod^{q_i \in \mathbf{Q}} p(q_i(\mathbf{x}^*, s=1) \geq 0 | \mathcal{S} \cup \{\mathbf{x}, s, \mathbf{q}, a\}\right] \alpha_F(\mathbf{x}, s)$$

Probability that the new predicted incumbent meets the constraints

Need to **simulate the models** using the predicted values

Speeding up the recommendation process

The computation of FABOLAS acquisition function is **very expensive**

TrimTuner tackles this issue by:

Decreasing the number of configurations for which the acquisition function is evaluated:

key idea: filtering non-promising configuration via a fast heuristic

Relying on **ensembles of Decision Trees**

Unlike FABOLAS, which exploits Gaussian Processes

TrimTuner: filtering out non-promising configurations

Introduces the **Constrained Expected Accuracy (CEA)** as the filtering heuristic

$$CEA(\mathbf{x}, s) = A(\mathbf{x}, s) \cdot \prod^{q_i \in \mathbf{Q}} p(q_i(\mathbf{x}, s) \geq 0 | \mathcal{S})$$

CEA can be seen as a rough, but cheap, approximation of the acquisition function

TrimTuner: filtering heuristic

- CEA can be efficiently evaluated for all the untested configurations
- TrimTuner ranks the untested configurations via the CEA

TrimTuner: Models

Allows both modelling techniques:

- **Gaussian Processes**
- Ensemble of **Decision Trees** (DTs)

Evaluation

- Considered baselines:
  - FABOLAS
  - BO using Constrained Expected Improvement (EIc)
  - BO using EIc/USD
  - Random search
- Search space composed of **1440 configurations**
- Target jobs: Distributed training of **3 Neural Networks** (NNs) deployed in AWS EC2

Configurations

Our data-set is publicly available:
github.com/pedrogbmendes/TrimTuner

Cloud parameters

VM Type	VM Characteristics	#VMs
t2.small	1 vCPU, 2GB RAM	8, 16, 32, 48, 64, 80
t2.medium	2 vCPU, 4GB RAM	4, 8, 16, 24, 32, 40
t2.xlarge	4 vCPU, 16GB RAM	2, 4, 8, 12, 16, 20
t2.2xlarge	8 vCPU, 32GB RAM	1, 2, 4, 6, 8, 10

Parameter	Values
Learning Rate	10^{-3}, 10^{-4}, 10^{-5}
Batch size	16, 256
Training mode	synchronous, asynchronous
Data-set size [%]	1/60, 1/10, 1/4, 1/2, 1

Hyper-parameters

Constraints

Only ≈**10%** **of configurations are feasible and have a high accuracy**

Neural Network	Feasible Configurations	Feasible configurations with high accuracy
RNN	178 (61.8%)	28 (9.72%)
MLP	161 (55.8%)	29 (10.07%)
CNN	111(38.5%)	39 (13.54%)

Evaluation Metrics

Constrained Accuracy:

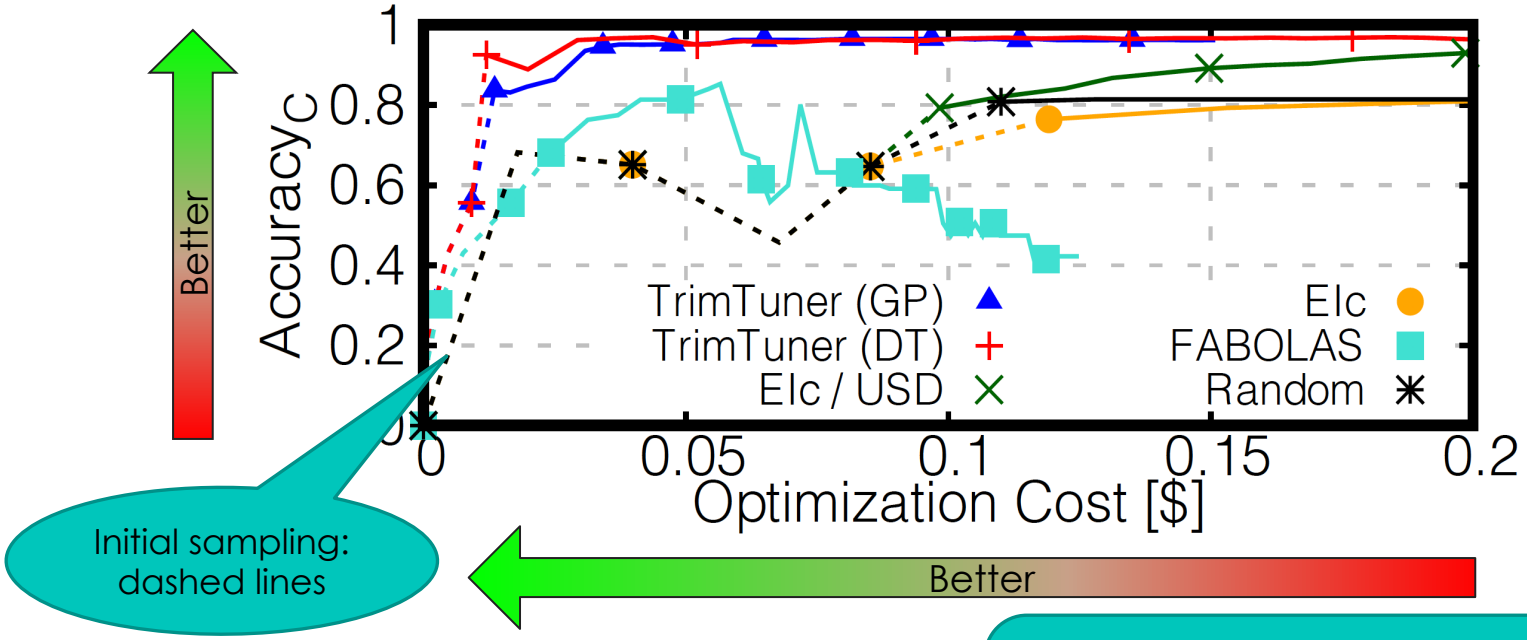
$$Accuracy_C = \begin{cases} A(x,s), & \text{if } (x,s) \text{ is feasible} \\ A(x,s) \cdot \frac{C_{max}}{C(x,s)}, & \text{otherwise} \end{cases}$$

We report the average value of 10 runs.

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University



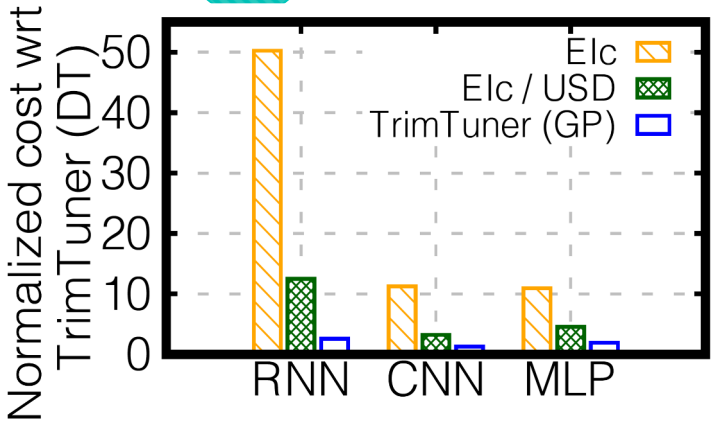
Initial sampling: dashed lines

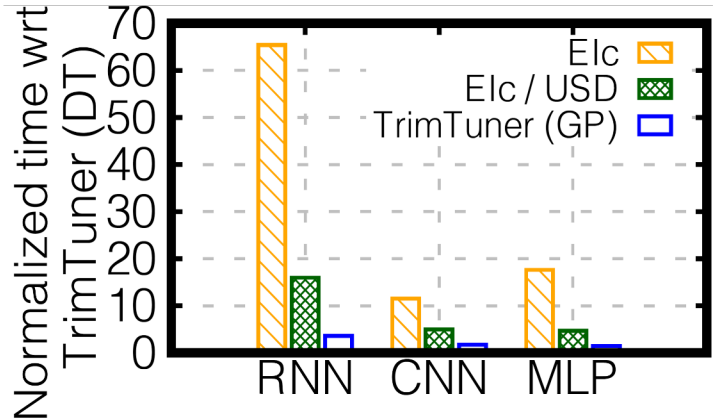
Optimization Process - RNN

TrimTuner can recommend **better configurations** while **reducing** the **cost**

TrimTuner Gains

To recommend configs close to optimum (within 5%)





Cost reduced by up to:

- **50x** w.r.t. EIc
- ≈**10x** w.r.t EIc/USD

Time accelerated by up to:

- **65x** w.r.t. EIc
- **15x** w.r.t EIc/USD

Recommendation Times

Optimizer	Avg. time to recommend a configuration [min]
TrimTuner (GPs)	18.65
TrimTuner (DTs)	**1.36**
Fabolas	13.96
Elc (Elc/USD)	1.17

Recommendation time reduced by **13.7x**

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

Efficiency of the CEA

- CEA compared with other filtering heuristics:
  - Direct [JGO01]
  - CMAES [EC03]

  Two state-of-the-art black-box optimizers
  - Random approach

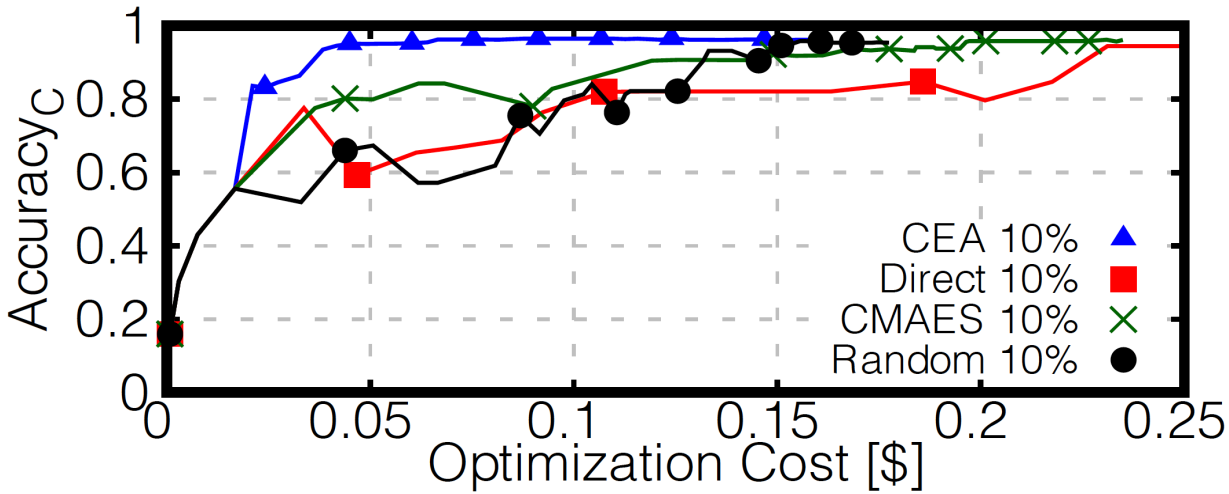
TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

Efficiency of the CEA - RNN

CEA reduces the recommendation cost by up to **7x/3.6x** w.r.t. Direct/CMAES



Efficiency of the CEA - RNN

Filtering Heuristic	Rec. Time TT DTs [min]	Rec. Time TT GPs [min]
CEA	**1.72**	16.85
Direct	2.63	36.18
CMAES	2.26	30.87
Random	1.62	16.53

Ongoing Work

Computation of acquisition functions based on ES is **very expensive**

(particularly, when GPs are used)

New acquisition function

(TrimTuner 2.0)

TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

TrimTuner 2.0: acquisition function

Determines the probability that the new optimum is feasible

Estimates the expected accuracy of the new optimum

Normalizes by the cost of evaluating (x,s)

Expected accuracy of the new predicted inc.

$$\alpha_{T2}(\mathbf{x}, s) = \boxed{\frac{1}{C(\mathbf{x},s)}} \cdot \boxed{\mathbb{E}_{p(\mathbf{q},a|\mathbf{x},s,\mathcal{S})}\left[A(\mathbf{x}^*, s=1|\mathcal{S}\cup\{\mathbf{x}, s, \mathbf{q}, a\})\right]} \cdot \boxed{\mathbb{E}_{p(\mathbf{q},a|\mathbf{x},s,\mathcal{S})}\left[\prod^{q_i\in\mathbf{Q}} p(q_i(\mathbf{x}^*, s=1)\geq 0|\mathcal{S}\cup\{\mathbf{x}, s, \mathbf{q}, a\}\right]}$$

Probability that the new predicted inc. meets the constraints

Ongoing Work

New filtering heuristic

- The filtering heuristic can be seen as a rough, but cheap approximation of the acquisition function
- The acquisition function is normalized by the cost (in order to sample cheaper configs)

Constrained Expected Accuracy per unit cost CEA/USD

$$\text{CEA/USD}(\mathbf{x}, s) = \frac{A(\mathbf{x},s)}{C(\mathbf{x},s)} \cdot \prod^{q_i \in \mathbf{Q}} p(q_i(\mathbf{x}, s) \geq 0 | \mathcal{S})$$

Advantages: selects **cheaper configs** using smaller data-set and thus **reduces the optimization cost**

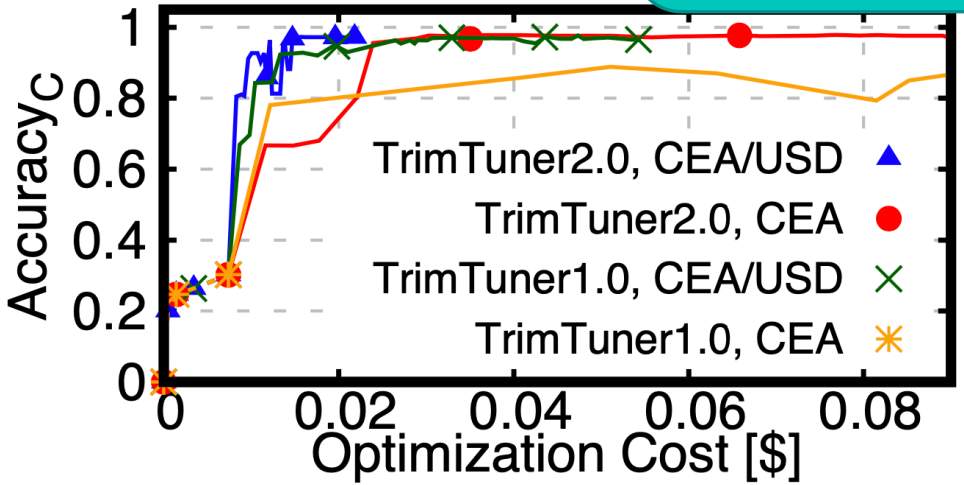
TÉCNICO LISBOA

inesc id lisboa

Carnegie Mellon University

New Acquisition Function & Filtering Heuristic - RNN

The joint use of the new acq. func. and new filtering heuristic reduces the optimization cost by up to **5.6x**



Final Remarks

TrimTuner: novel system to **optimize ML model training in the cloud**

- Maximizes model accuracy subject to QoS constraint
- Exploits sub-sampling data to reduce the costs

TrimTuner relies on subsampling techniques to reduce the:

- Training cost by up to **50x**
- Latency of the exploration process by up to **65x** (compared with BO using EIc);

It accelerates the recommendation process by up to **117x** via

- Novel filtering heuristic: CEA
- Lightweight predictive models based on DTs

Code and data-sets are available online

github.com/pedrogbmendes/TrimTuner

Check our paper for more details

Questions?