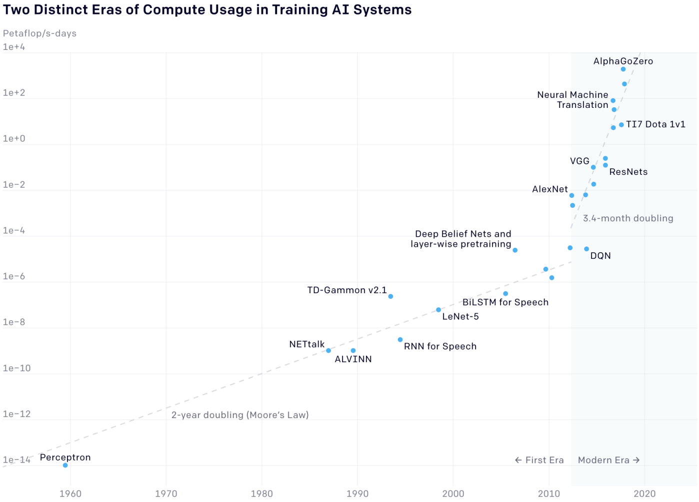
Power-efficient Deep Learning Workshop Introduction

Sébastien Loustau

Team leader of GreenAI UPPA
Researcher at Université de Pau et des Pays de l'Adour

Workshop Introduction
November, 17th 2021

AI and Compute



Carbon footprint of AI

Carbon footprint of AI

"The climate impact of digital technology is bigger than aviation."

Carbon footprint of AI

"The climate impact of digital technology is bigger than aviation."

- AI vs digital,

Carbon footprint of AI

"The climate impact of digital technology is bigger than aviation."

- AI vs digital,
- Compute vs LCA.

Carbon footprint of AI

"The climate impact of digital technology is bigger than aviation."

- AI vs digital,
- Compute vs LCA.

3h of training AI is equiv. to one month of battery charges of a laptop

Training GPT3 would cost around 200,000kWh [2]

Carbon footprint of AI

"The climate impact of digital technology is bigger than aviation."

- AI vs digital,
- Compute vs LCA.

3h of training AI is equiv. to one month of battery charges of a laptop

Training GPT3 would cost around 200,000kWh [2]

Deep Learning Zoology

Low bitwidth approaches

- Originally from [7] - BinaryConnect
- Weights and I/O in [15, 5] - XNOR-nets $++$
- Quantization in [24, 20]

Deep Learning Zoology

Low bitwidth approaches

- Originally from [7] - BinaryConnect
- Weights and I/O in [15, 5] - XNOR-nets ++
- Quantization in [24, 20]

Pruning approaches

- Originally after training in [14] (see also [23]),
- At initalization in [9],
- During the training and using sparsity in [3, 21],

Deep Learning Zoology

Low bitwidth approaches

- Originally from [7] - BinaryConnect
- Weights and I/O in [15, 5] - XNOR-nets ++
- Quantization in [24, 20]

Pruning approaches

- Originally after training in [14] (see also [23]),
- At initalization in [9],
- During the training and using sparsity in [3, 21],
- Related with EDropout and architecture search - [22, 8, 17].

Deep Learning Zoology

Low bitwidth approaches

- Originally from [7] - BinaryConnect
- Weights and I/O in [15, 5] - XNOR-nets ++
- Quantization in [24, 20]

Pruning approaches

- Originally after training in [14] (see also [23]),
- At initalization in [9],
- During the training and using sparsity in [3, 21],
- Related with EDropout and architecture search - [22, 8, 17].

Bio-inspired energy efficiency

Deep Learning Zoology

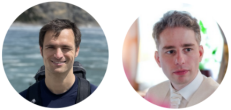
Low bitwidth approaches

- Originally from [7] - BinaryConnect
- Weights and I/O in [15, 5] - XNOR-nets ++
- Quantization in [24, 20]

Pruning approaches

- Originally after training in [14] (see also [23]),
- At initalization in [9],
- During the training and using sparsity in [3, 21],
- Related with EDropout and architecture search - [22, 8, 17].

Bio-inspired energy efficiency



Power meters and Hardwares

Power measurements

- [4, 10, 2] uses **RAPL** and **nvidia-smi** specific performance counters,
- [16, 10] proposes a layer by layer energy measurements comparing Mobilenets [18] to standard Inception-V3 on a ARM Cortex-A57,
- [6, 16] proposes direct power meter measurements,
- [12, 11] quantifies the carbon emission of compute here.

Power meters and Hardwares

Power measurements

- [4, 10, 2] uses **RAPL** and **nvidia-smi** specific performance counters,
- [16, 10] proposes a layer by layer energy measurements comparing Mobilenets [18] to standard Inception-V3 on a ARM Cortex-A57,
- [6, 16] proposes direct power meter measurements,
- [12, 11] quantifies the carbon emission of compute here.

Hardwares

- CPU to **GPU**,
- CNNs accelerator on **FPGA** or **ASIC**, see [19, 1],
- GPU to nano-computers (**Jetson** Nano / Xavier).

Power meters and Hardwares

Power measurements

- [4, 10, 2] uses **RAPL** and **nvidia-smi** specific performance counters,
- [16, 10] proposes a layer by layer energy measurements comparing Mobilenets [18] to standard Inception-V3 on a ARM Cortex-A57,
- [6, 16] proposes direct power meter measurements,
- [12, 11] quantifies the carbon emission of compute here.

Hardwares

- CPU to **GPU**,
- CNNs accelerator on **FPGA** or **ASIC**, see [19, 1],
- GPU to nano-computers (**Jetson** Nano / Xavier).

Our team approach

- Website
- AIPowerMeter
- Github

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach
$\min_\theta \ell(\theta)$	$\min_\rho \{\mathbb{E}_{\theta\sim\rho}\ell(\theta) + \mathcal{D}(\rho,\pi)\}$

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach
$\min_\theta \ell(\theta)$	$\min_\rho \{\mathbb{E}_{\theta\sim\rho}\ell(\theta) + \mathcal{D}(\rho,\pi)\}$
SGD	MCMC

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach
$\min_\theta \ell(\theta)$	$\min_\rho \{\mathbb{E}_{\theta\sim\rho}\ell(\theta) + \mathcal{D}(\rho,\pi)\}$
SGD	MCMC
differentiable, global and continuous	layerwise discrete hybrid

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach
$\min_\theta \ell(\theta)$	$\min_\rho \{\mathbb{E}_{\theta\sim\rho}\ell(\theta) + \mathcal{D}(\rho,\pi)\}$
SGD	MCMC
differentiable, global and continuous	layerwise discrete hybrid
statistical learning	sequential learning

Our team approach

- Website
- AIPowerMeter
- Github

Classical DL	Our approach
$\min_\theta \ell(\theta)$	$\min_\rho \{\mathbb{E}_{\theta\sim\rho}\ell(\theta) + \mathcal{D}(\rho,\pi)\}$
SGD	MCMC
differentiable, global and continuous	layerwise discrete hybrid
statistical learning	sequential learning
signal-processing based	math / stats / game theory based

- a sequence of deterministic (or i.i.d.) **inputs** $z_1, \ldots, z_T \in \mathcal{Z}$,
- a set of **weak learners** $g \in \mathcal{G}$,
- a **loss** function $\ell(\cdot, \cdot) : \mathcal{G} \times \mathcal{Z} \to \mathbb{R}_+$.

$\boxed{\text{Goal}}$ Find a sequence of distributions such that:

$$\sum_{t=1}^{T} \mathbb{E}_{g' \sim \tilde{\rho}_t} \ell(g', z_t) - \inf_{g \in \mathcal{G}} \left\{ \sum_{t=1}^{T} \ell(g, z_t) + \lambda \text{pen}(g) \right\}$$

is minimum, where pen(g) measure the **'complexity'** of the decision $g \in \mathcal{G}$.

- $z = (x, y)$, $x \in \mathcal{X}$ input space of images, time series, network,
- $\mathcal{G} := \{g_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}, \mathbf{w} \in \mathcal{W}\}$, where $\mathbf{w}$ are the weights of a given standard or XNOR-nets architecture. For XNOR-nets convolutions are approximated by bitwise operations:

$$x_k = \left(\mathbf{w}_k^{\mathrm{bin}} \bigoplus \mathrm{sign} \circ \mathrm{BNorm}\left(x_{k-1}\right)\right) \bigodot \mathbf{w}_k^{\mathrm{scale}},$$

- the **cross-entropy** loss function $\ell(\hat{y}, y)$.

Sparsity regret bound

XNOR-Nets case

Theorem (L. Chee and Gay, 2021)

Considering inputs $\{(x_t, y_t), t = 1, \ldots, T\}$, *the decision space* $\mathcal{G}$, *and cross-entropy loss, there exists a* ***sequence of distributions*** $(\tilde{\rho}_t)_{t=1}^T$ *on* $\mathcal{G}$ *such that:*

$$\sum_{t=1}^{T} \mathbb{E}_{g' \sim \tilde{\rho}_t} \ell(y_t, g'(x_t)) \leq \inf_{\mathbf{w} \in \mathcal{W}_{\mathrm{XNOR}}} \left\{ \sum_{t=1}^{T} \ell(y_t, g_{\mathbf{w}}(x_t)) + pen(g_{\mathbf{w}}) \right\} + \Delta_T,$$

where $\Delta_T > 0$ *is defined in [13] and* $pen(g_{\mathbf{w}})$ *measure the complexity of the network as follows:*

$$pen(g_{\mathbf{w}}) = 4 \sum_{\mathbf{w} \in \{\mathbf{w}^{\mathrm{real}}, \mathbf{w}^{\mathrm{scale}}\}} \|\mathbf{w}\|_0 \log\left(1 + \frac{\|\mathbf{w}\|_1}{\tau \|\mathbf{w}\|_0}\right) + p_{\mathrm{bin}} \log 2$$

Accept-Reject algorithm

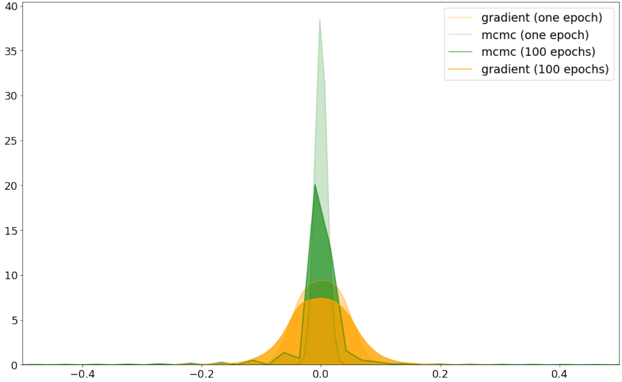
Pseudo-code

init. $\lambda > 0$, sparsity prior π, $k = 1$.

1. Draw $\hat{\mathbf{w}}_1$ from π.
2. **Repeat** for $k = 1, \dots, K$:
   - Generate a proposal $\tilde{\mathbf{w}} \sim \tilde{p}_{\hat{\mathbf{w}}_k, \sigma}$, where $\tilde{p}$ has mean $\hat{\mathbf{w}}_k$.
   - Compute:
   $$\rho_j(\tilde{\mathbf{w}}, \hat{\mathbf{w}}_k) = \frac{\exp\{-\lambda \ell(\tilde{\mathbf{w}}, B_j)\}\, \pi(\tilde{\mathbf{w}})}{\exp\{-\lambda \ell(\hat{\mathbf{w}}_k, B_j)\}\, \pi(\hat{\mathbf{w}}_k)}.$$
   - Update
   $$\hat{\mathbf{w}}_{k+1} = \begin{cases} \tilde{\mathbf{w}} & \text{with proba } \rho_j(\tilde{\mathbf{w}}, \hat{\mathbf{w}}_k) \\ \hat{\mathbf{w}}_k & \text{otherwise.} \end{cases}$$

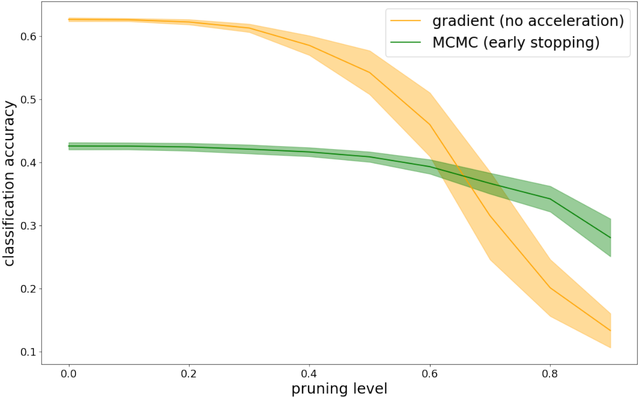
Weight distributions

SGD vs MCMC



Robustness to pruning

SGD vs MCMC



Future works

- Structural sparsity and more specific priors,
- Asynchronous and decentralized algorithms,
- Forward accelerations,
- Extensions to non-diff losses and other divergences, link with MTS problem,
- Workshop page see - the dedicated page -

References I

Renzo Andri, Lukas Cavigelli, Davide Rossi, and Luca Benini.
Yodann: An architecture for ultralow power binary-weight cnn acceleration.
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(1):48–60, 2017.

Lasse Anthony, Benjamin Kanding, and Raghavendra Selvan.
Carbontracker: Tracking and predicting the carbon footprint of training deep learning models.
page arXiv preprint https://arxiv.org/abs/2007.03051, 07 2020.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein.
Deep rewiring: Training very sparse deep networks.
In *International Conference on Learning Representations*, 2018.

References II

Aurélien Bourdon, Adel Noureddine, Romain Rouvoy, and Lionel Seinturier.
Powerapi: A software library to monitor the energy consumed at the process-level.
ERCIM News, 2013(92).

Adrian Bulat and Georgios Tzimiropoulos.
Xnor-net++: Improved binary neural networks.
arXiv preprint arXiv:1909.13863, 2019.

Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian.
Towards accurate and reliable energy measurement of NLP models.
In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 141–148, Online, November 2020. Association for Computational Linguistics.

References III

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David.
Binaryconnect: Training deep neural networks with binary weights during propagations.
In *Advances in neural information processing systems*, pages 3123–3131, 2015.

Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, and Joseph E. Gonzalez.
Fbnetv3: Joint architecture-recipe search using predictor pretraining.
arXiv preprint arXiv:2006.02049, 2021.

References IV

Pau de Jorge, Amartya Sanyal, Harkirat S Behl, Philip HS Torr, Gregory Rogez, and Puneet K Dokania.
Progressive skeletonization: Trimming more fat from a network at initialization.
arXiv preprint arXiv:2006.09081, 2020.

Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn.
Estimation of energy consumption in machine learning.
Journal of Parallel and Distributed Computing, 134:75–88, 2019.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres.
Quantifying the carbon emissions of machine learning.
arXiv preprint arXiv:1910.09700, 2019.

References V

Loïc Lannelongue, Jason Grealey, and Michael Inouye.
Green algorithms: Quantifying the carbon emissions of computation.
Advanced science, 05 2021.

Chee Andrew Loustau, Sébastien and Paul Gay.
Sparsity regret bounds for xnor-nets++.
https://hal.archives-ouvertes.fr/hal-03262679, 2021.

Michael C Mozer and Paul Smolensky.
Skeletonization: A technique for trimming the fat from a network via relevance assessment.
In *Proceedings of the 1st International Conference on Neural Information Processing Systems*, pages 107–115, 1988.

References VI

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi.
Xnor-net: Imagenet classification using binary convolutional neural networks.
In *European conference on computer vision*, pages 525–542. Springer, 2016.

Crefeda Faviola Rodrigues, Graham Riley, and Mikel Luján.
Synergy: An energy measurement and prediction framework for convolutional neural networks on jetson tx1.
In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pages 375–382. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018.

Hojjat Salehinejad and Shahrokh Valaee.
Edropout: Energy-based dropout and pruning of deep neural networks.
CoRR, abs/2006.04270, 2020.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen.
Mobilenetv2: Inverted residuals and linear bottlenecks, 2018.
cite arxiv:1801.04381.

Ahmad Shawahna, Sadiq M Sait, and Aiman El-Maleh.
Fpga-based accelerators of deep learning networks for learning and classification: A review.
IEEE Access, 7:7823–7859, 2018.

References VIII

Wonyong Sung, Sungho Shin, and Kyuyeon Hwang.
Resiliency of deep neural networks under quantization.
arXiv preprint arXiv:1511.06488, 2015.

Masha Taheri, Eang Xie, and Johannes Lederer.
Statistical guarantees for regularized neural networks.
Neural Networks, 142:148–161, 2021.

Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer.
Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.

References IX

Zichao Yang, Marcin Moczulski, Misha Denil, Nando De Freitas, Alex Smola, Le Song, and Ziyu Wang.
Deep fried convnets.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1483, 2015.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou.
Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients.
arXiv preprint arXiv:1606.06160, 2016.