Power-efficient Deep Learning

Des maths pour le climat

Andrew Chee [1] Sébastien Loustau [2]

[1]Cornell ORIE, Cornell University
Ithaca, New York, 14853, USA

[2]Laboratoire de Recherche en Mathématiques Appliquées de Pau (LMAP)
Université de Pau et des Pays de l'Adour
Pau, France

October 15, 2020

Motivations

- We love (theoretical) compromise,

Motivations

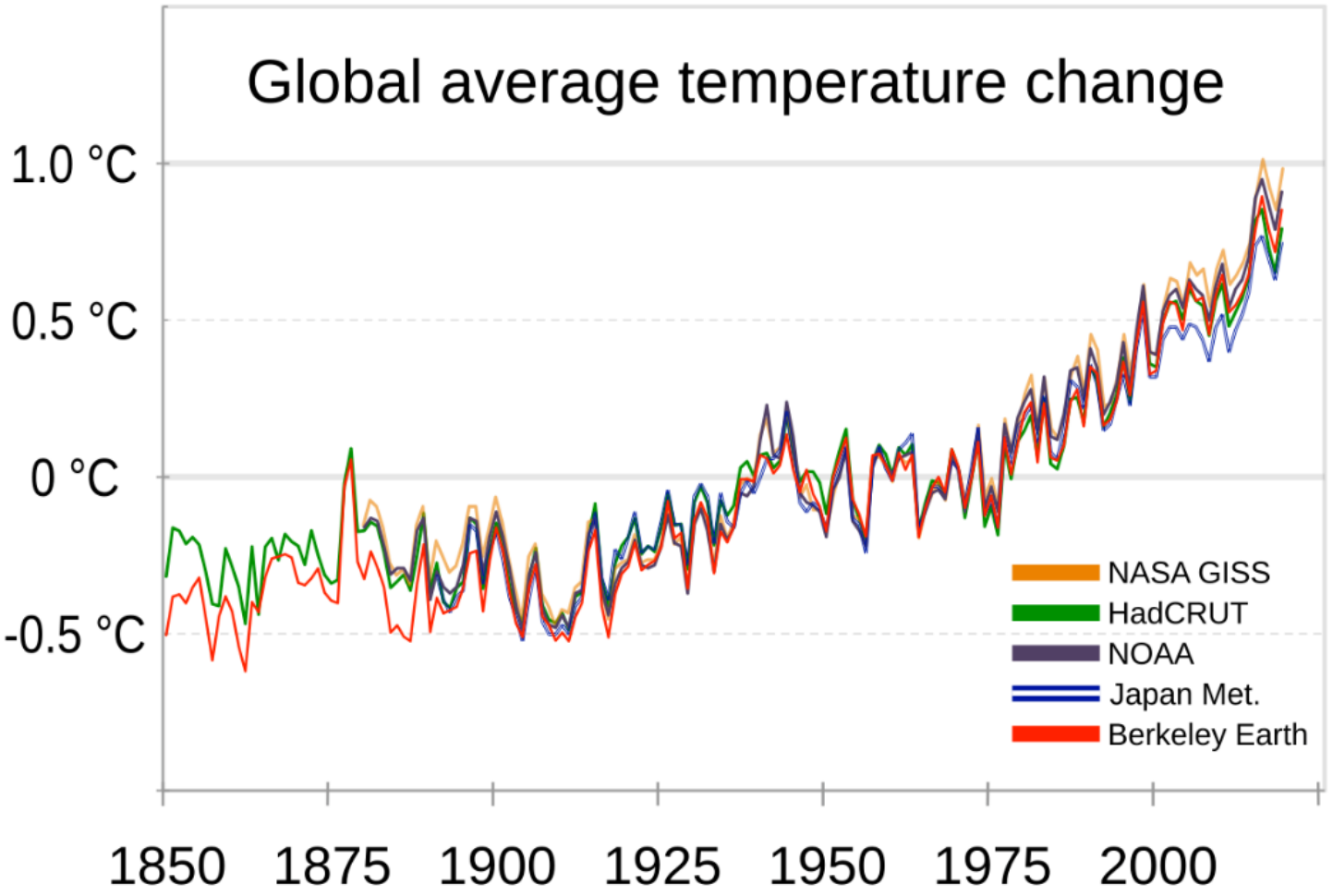
- We love (theoretical) compromise,
- Simple solutions are often sufficient (and sometimes even better),

Motivations

- We love (theoretical) compromise,
- Simple solutions are often sufficient (and sometimes even better),
- Deep Learning is popular and powerful,

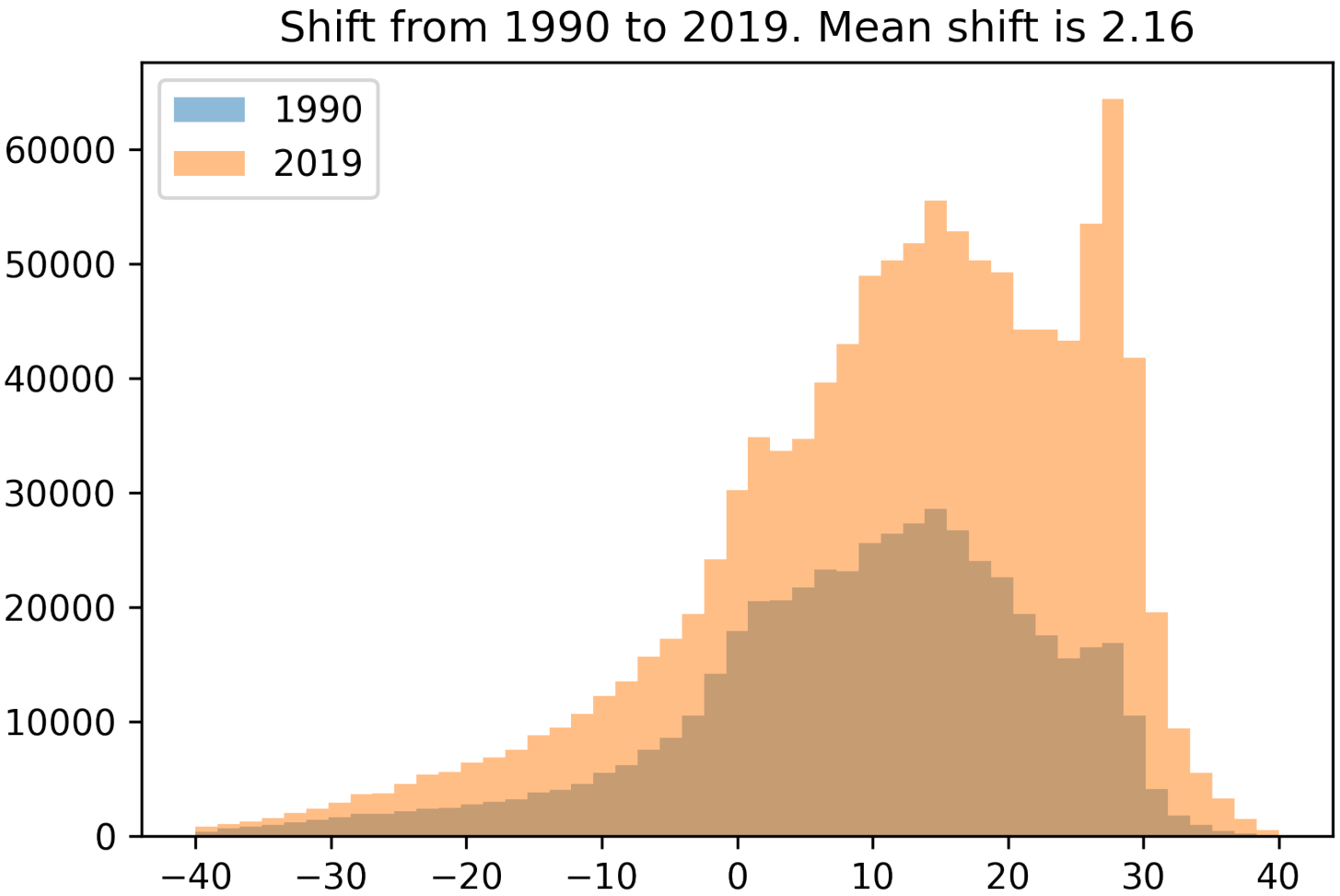
Motivations

- We love (theoretical) compromise,
- Simple solutions are often sufficient (and sometimes even better),
- Deep Learning is popular and powerful,
- Has the earth ever been this hot before?


Global average temperature change
1.0 °C
0.5 °C
0 °C
-0.5 °C
1850
1875
1900
1925
1950
1975
2000
NASA GISS
HadCRUT
NOAA
Japan Met.
Berkeley Earth

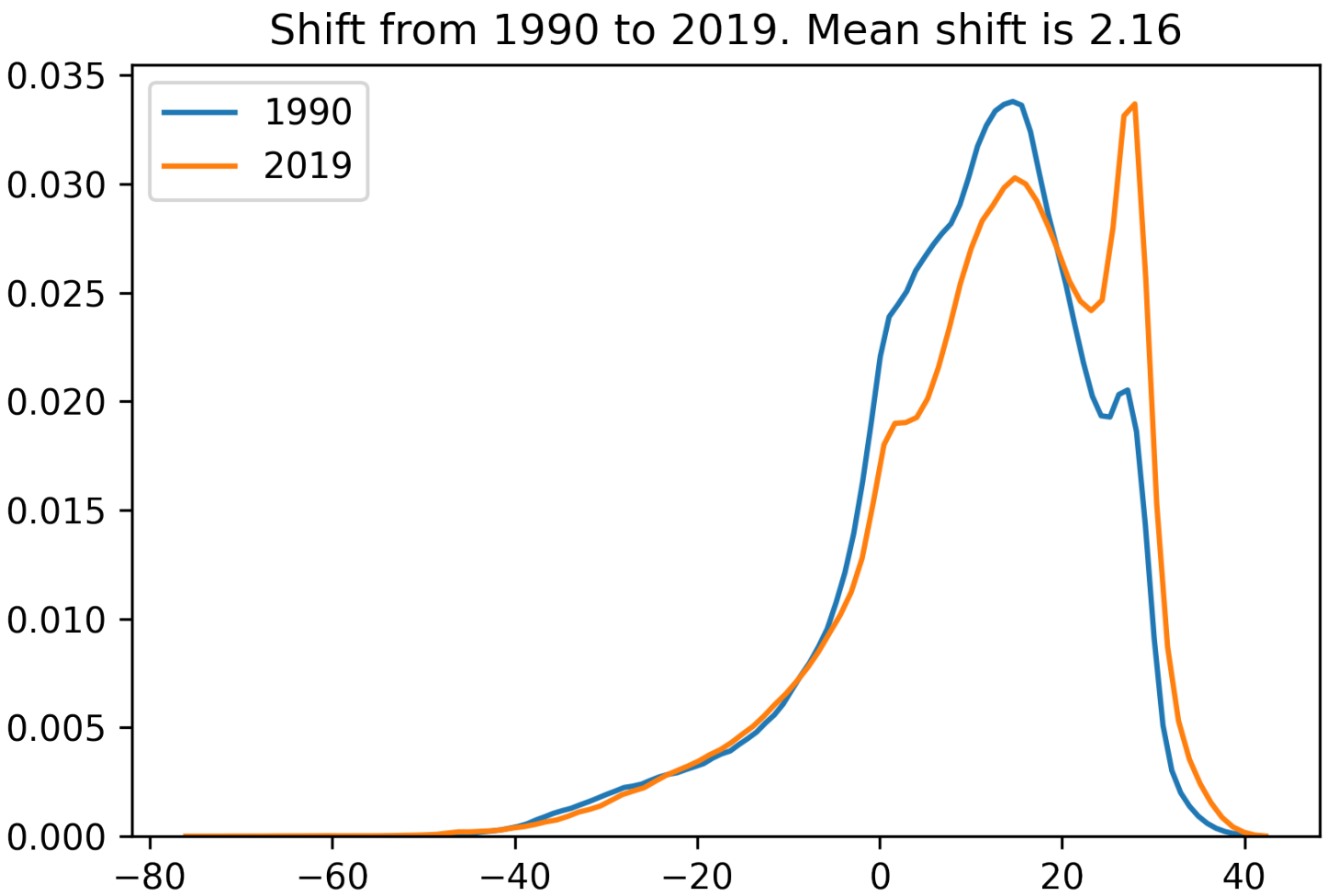
Source here

Mean shift VS Distribution shift (NOAA Dataset)

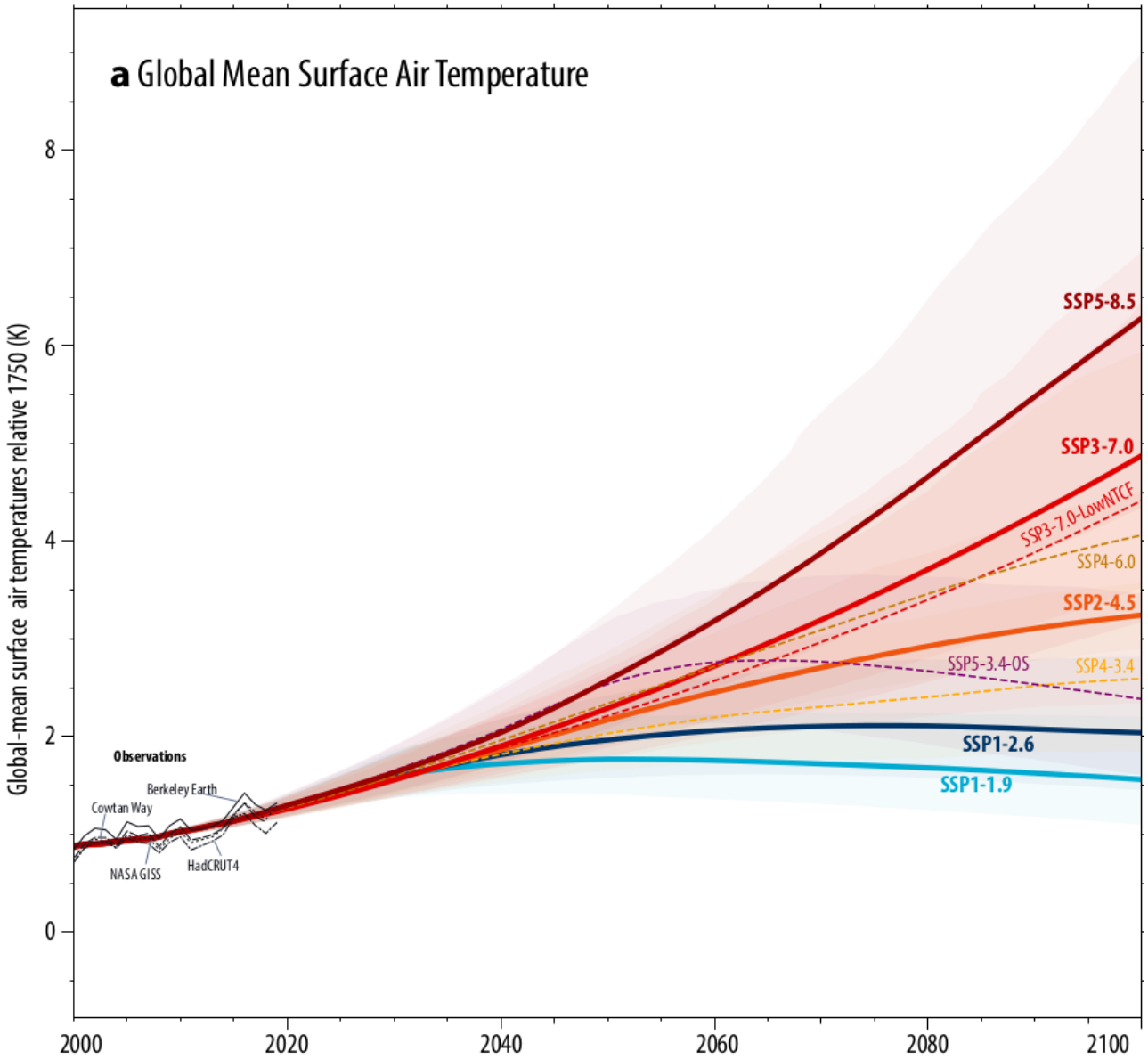


data

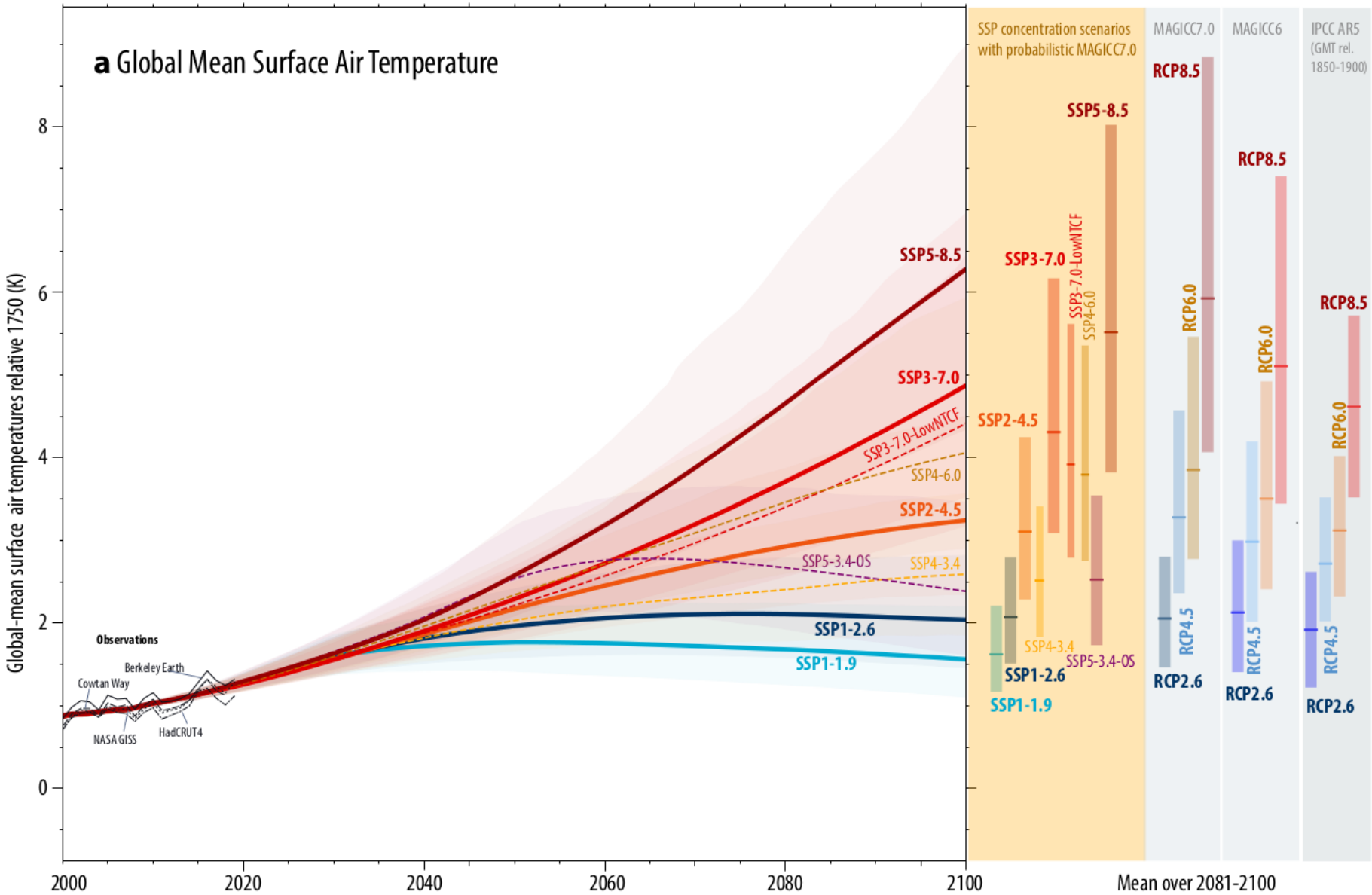
Mean shift VS Distribution shift (NOAA Dataset)



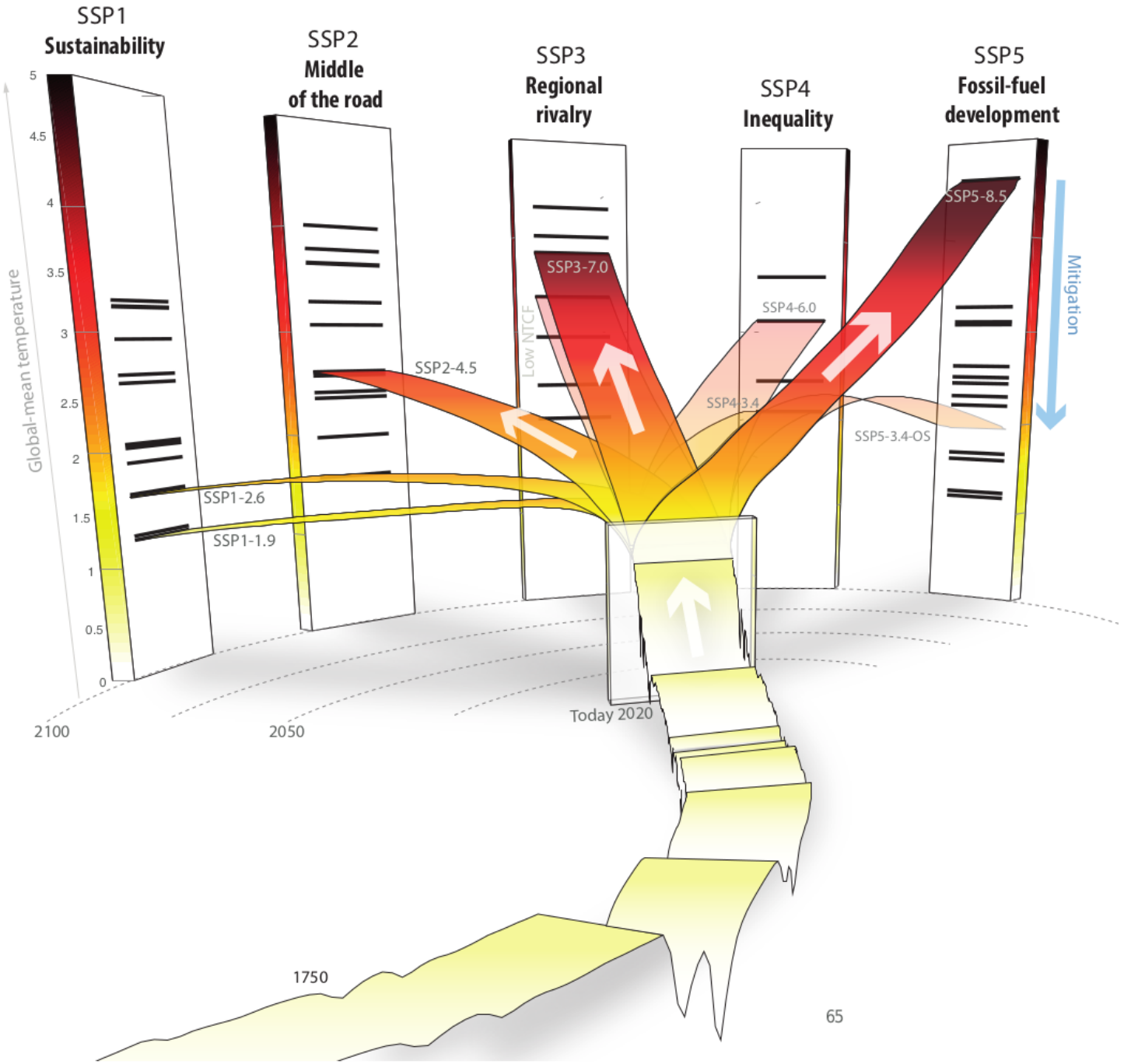
data


a Global Mean Surface Air Temperature
Global-mean surface air temperatures relative 1750 (K)
8
6
4
2
0
2000
2020
2040
2060
2080
2100
SSP5-8.5
SSP3-7.0
SSP3-7.0-LowNTCF
SSP4-6.0
SSP2-4.5
SSP5-3.4-OS
SSP4-3.4
SSP1-2.6
SSP1-1.9
Observations
Berkeley Earth
Cowtan Way
NASA GISS
HadCRUT4

Source here


a Global Mean Surface Air Temperature
Global-mean surface air temperatures relative 1750 (K)
0
2
4
6
8
2000
2020
2040
2060
2080
2100
Observations
Berkeley Earth
Cowtan Way
NASA GISS
HadCRUT4
SSP5-8.5
SSP3-7.0
SSP3-7.0-LowNTCF
SSP4-6.0
SSP2-4.5
SSP5-3.4-OS
SSP4-3.4
SSP1-2.6
SSP1-1.9
SSP concentration scenarios with probabilistic MAGICC7.0
MAGICC7.0
MAGICC6
IPCC AR5 (GMT rel. 1850-1900)
RCP8.5
RCP6.0
RCP4.5
RCP2.6
Mean over 2081-2100

Source here


SSP1
Sustainability
SSP2
Middle
of the road
SSP3
Regional
rivalry
SSP4
Inequality
SSP5
Fossil-fuel
development
Global-mean temperature
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
SSP5-8.5
SSP3-7.0
SSP4-6.0
SSP2-4.5
Low NTCF
SSP4-3.4
SSP5-3.4-OS
Mitigation
SSP1-2.6
SSP1-1.9
2100
2050
Today 2020
1750
65

Source here

Motivations

- We love (theoretical) compromise,
- Simple solutions are often sufficient (and sometimes even better),
- Deep Learning is popular and powerful,
- Has the Earth Ever Been This Hot Before? **It sucks.**

Binary Nets, Quantization

A general way of creating new Power-efficient Deep Learning is to reduce the number and the cost of MAC (multiply-and-accumulates) operations:

Binary Nets, Quantization

A general way of creating new Power-efficient Deep Learning is to reduce the number and the cost of MAC (multiply-and-accumulates) operations:

- In [1] **Binaryconnect**, Binarized NNs are trained with SGD (binary weights are used during the forward and backward propagations but real-valued weights are stored for SGD updates) on MNIST, CIFAR-10 and SVHN,

Binary Nets, Quantization

A general way of creating new Power-efficient Deep Learning is to reduce the number and the cost of MAC (multiply-and-accumulates) operations:

- In [1] **Binaryconnect**, Binarized NNs are trained with SGD (binary weights are used during the forward and backward propagations but real-valued weights are stored for SGD updates) on MNIST, CIFAR-10 and SVHN,
- In [2, 3] **XNOR-nets** are trained (binary weights and activations for propagations) with the same SGD real-valued updates,

Binary Nets, Quantization

A general way of creating new Power-efficient Deep Learning is to reduce the number and the cost of MAC (multiply-and-accumulates) operations:

- In [1] **Binaryconnect**, Binarized NNs are trained with SGD (binary weights are used during the forward and backward propagations but real-valued weights are stored for SGD updates) on MNIST, CIFAR-10 and SVHN,
- In [2, 3] **XNOR-nets** are trained (binary weights and activations for propagations) with the same SGD real-valued updates,
- **Quantization** of weights (see [4] for references and a study on the compromise between precision and large size network).

Binary Nets, Quantization

A general way of creating new Power-efficient Deep Learning is to reduce the number and the cost of MAC (multiply-and-accumulates) operations:

- In [1] **Binaryconnect**, Binarized NNs are trained with SGD (binary weights are used during the forward and backward propagations but real-valued weights are stored for SGD updates) on MNIST, CIFAR-10 and SVHN,
- In [2, 3] **XNOR-nets** are trained (binary weights and activations for propagations) with the same SGD real-valued updates,
- **Quantization** of weights (see [4] for references and a study on the compromise between precision and large size network).

From a **Machine Learning perspective**, both training and inference of well-known architectures can be lightened by acting on the weights and I/O spaces.

CNNs Accelerators on hardware

Many papers implements **CNNs accelerators** on field-programmable gate arrays (**FPGAs**) or on Application Specific Integrated Circuits (**ASICs**) leading to ultra-low power CNNs acceleration:

CNNs Accelerators on hardware

Many papers implements **CNNs accelerators** on field-programmable gate arrays (**FPGAs**) or on Application Specific Integrated Circuits (**ASICs**) leading to ultra-low power CNNs acceleration:

- Binaryconnect on ASIC, see [5],

CNNs Accelerators on hardware

Many papers implements **CNNs accelerators** on field-programmable gate arrays (**FPGAs**) or on Application Specific Integrated Circuits (**ASICs**) leading to ultra-low power CNNs acceleration:

- Binaryconnect on ASIC, see [5],
- Hardware accelerators on FPGAs such as XNOR-Nets, see [6],

CNNs Accelerators on hardware

Many papers implements **CNNs accelerators** on field-programmable gate arrays (**FPGAs**) or on Application Specific Integrated Circuits (**ASICs**) leading to ultra-low power CNNs acceleration:

- Binaryconnect on ASIC, see [5],
- Hardware accelerators on FPGAs such as XNOR-Nets, see [6],
- Efficient inference processing of CNNs on many different platforms with bit-wise identical results (ASIC and FPGA-based designs, but also kernel computation on CPU and GPU), see [7].

CNNs Accelerators on hardware

Many papers implements **CNNs accelerators** on field-programmable gate arrays (**FPGAs**) or on Application Specific Integrated Circuits (**ASICs**) leading to ultra-low power CNNs acceleration:

- Binaryconnect on ASIC, see [5],
- Hardware accelerators on FPGAs such as XNOR-Nets, see [6],
- Efficient inference processing of CNNs on many different platforms with bit-wise identical results (ASIC and FPGA-based designs, but also kernel computation on CPU and GPU), see [7].

Many trade-offs between **many criterion** are possible to have a complete picture of the proposed designs and techniques (accuracy, number of MAC operations, throughput/latency and power/energy consumption).

Automation and Mobile phone

The problem of designing efficient neural networks lead finally to search for **device-specific CNNs** and finally automated machine learning (www.auto-ml.org) and **Neural Architecture Search** (NAS, see [8]).

Automation and Mobile phone

The problem of designing efficient neural networks lead finally to search for **device-specific CNNs** and finally automated machine learning (www.auto-ml.org) and **Neural Architecture Search** (NAS, see [8]). Several papers apply these considerations to design Mobile nets (see also chamnet for CPU/GPU:

- [9, 10] propose to use NAS neural architecture to construct hardware efficient CNNs for **mobile phones**,

Automation and Mobile phone

The problem of designing efficient neural networks lead finally to search for **device-specific CNNs** and finally automated machine learning (www.auto-ml.org) and **Neural Architecture Search** (NAS, see [8]). Several papers apply these considerations to design Mobile nets (see also chamnet for CPU/GPU:

- [9, 10] propose to use NAS neural architecture to construct hardware efficient CNNs for **mobile phones**,
- [11] uses several budgeted super networks to predict well in less than 100 milliseconds or to learn efficient models in terms of memory (for instance models that fit in a 50Mb memory),

Automation and Mobile phone

The problem of designing efficient neural networks lead finally to search for **device-specific CNNs** and finally automated machine learning (www.auto-ml.org) and **Neural Architecture Search** (NAS, see [8]). Several papers apply these considerations to design Mobile nets (see also chamnet for CPU/GPU:

- [9, 10] propose to use NAS neural architecture to construct hardware efficient CNNs for **mobile phones**,
- [11] uses several budgeted super networks to predict well in less than 100 milliseconds or to learn efficient models in terms of memory (for instance models that fit in a 50Mb memory),
- [12] uses hierarchical neural ensemble to reduce FLOPS and control dynamically the **inference latency**.

Automation and Mobile phone

The problem of designing efficient neural networks lead finally to search for **device-specific CNNs** and finally automated machine learning (www.auto-ml.org) and **Neural Architecture Search** (NAS, see [8]). Several papers apply these considerations to design Mobile nets (see also chamnet for CPU/GPU:

- [9, 10] propose to use NAS neural architecture to construct hardware efficient CNNs for **mobile phones**,
- [11] uses several budgeted super networks to predict well in less than 100 milliseconds or to learn efficient models in terms of memory (for instance models that fit in a 50Mb memory),
- [12] uses hierarchical neural ensemble to reduce FLOPS and control dynamically the **inference latency**.

In what follows, we intent to build **a new theoretical-based approach** that reaches these kind of trade-offs.

Energy consumption of Deep Learning

Performance counters (PMCs) are used to estimate power or energy consumptions of Deep Learning:

- [13] provides a layer by layer energy measurements and estimation for usual convnets for imagenet,
- [14] proposes a survey on energy estimation models for machine/deep learning at three levels (application and instruction level for software) and hardware level, and practical study comparing mobilenet to standard Inception-V3 on a ARM Cortex-A57 thanks to `SyNERGY`,
- [15] proposes `PowerAPI` a software to monitore in real time the energy consumption of a program.
- [16] quantifies the carbon emission of many GP-GPU.

Energy consumption of Deep Learning

Performance counters (PMCs) are used to estimate power or energy consumptions of Deep Learning:

- [13] provides a layer by layer energy measurements and estimation for usual convnets for imagenet,
- [14] proposes a survey on energy estimation models for machine/deep learning at three levels (application and instruction level for software) and hardware level, and practical study comparing mobilenet to standard Inception-V3 on a ARM Cortex-A57 thanks to `SyNERGY`,
- [15] proposes `PowerAPI` a software to monitore in real time the energy consumption of a program.
- [16] quantifies the carbon emission of many GP-GPU.

Energy consumption of Deep Learning

AlexNet to AlphaGo Zero: A 300,000x Increase in Compute

$$1860 * 10^{15} * 86400 = 1.607e + 23$$

References cited above I

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.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio.
Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1.
arXiv preprint arXiv:1602.02830, 2016.

References cited above II


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.


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


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.

References cited above III


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.


Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer.
Efficient processing of deep neural networks: A tutorial and survey.
Proceedings of the IEEE, 105(12):2295–2329, 2017.


Barret Zoph and Quoc V Le.
Neural architecture search with reinforcement learning.
arXiv preprint arXiv:1611.01578, 2016.

References cited above IV

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.

Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al.
Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions.
arXiv preprint arXiv:2004.05565, 2020.

References cited above V


Tom Veniat and Ludovic Denoyer.
Learning time/memory-efficient deep architectures with budgeted super networks.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3500, 2018.


Adrià Ruiz and Jakob Verbeek.
Distilled hierarchical neural ensembles with adaptive inference cost.
arXiv preprint arXiv:2003.01474, 2020.


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

References cited above VI

375–382. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018.


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.


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).

References cited above VII


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


Pierre Alquier and Benjamin Guedj.
Simpler pac-bayesian bounds for hostile data.
Machine Learning, 107(5):887–902, 2018.


Yingzhen Li and Richard E Turner.
Rényi divergence variational inference.
In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

References cited above VIII


Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas.
Generalized variational inference: Three arguments for deriving new posteriors.
arXiv preprint arXiv:1904.02063, 2019.


Pierre Alquier.
Non-exponentially weighted aggregation: regret bounds for unbounded loss functions.
arXiv preprint arXiv:2009.03017, 2020.


Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota.
Practical deep learning with bayesian principles.
In *Advances in neural information processing systems*, pages 4287–4299, 2019.

References cited above IX


Jérémie Bigot, Raúl Gouet, Thierry Klein, and Alfredo López.
Geodesic pca in the wasserstein space.
arXiv preprint arXiv:1307.7721, 2013.


Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy.
Joint distribution optimal transportation for domain adaptation.
In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.


Aude Genevay, Gabriel Peyré, and Marco Cuturi.
Learning generative models with sinkhorn divergences.
In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617, 2018.

References cited above X


Gabriel Peyré, Marco Cuturi, et al.
Computational optimal transport.
Foundations and Trends® in Machine Learning, 11(5-6):355–607, 2019.