A MULTI-OBJECTIVE APPROACH FOR SUSTAINABLE DEEP LEARNING

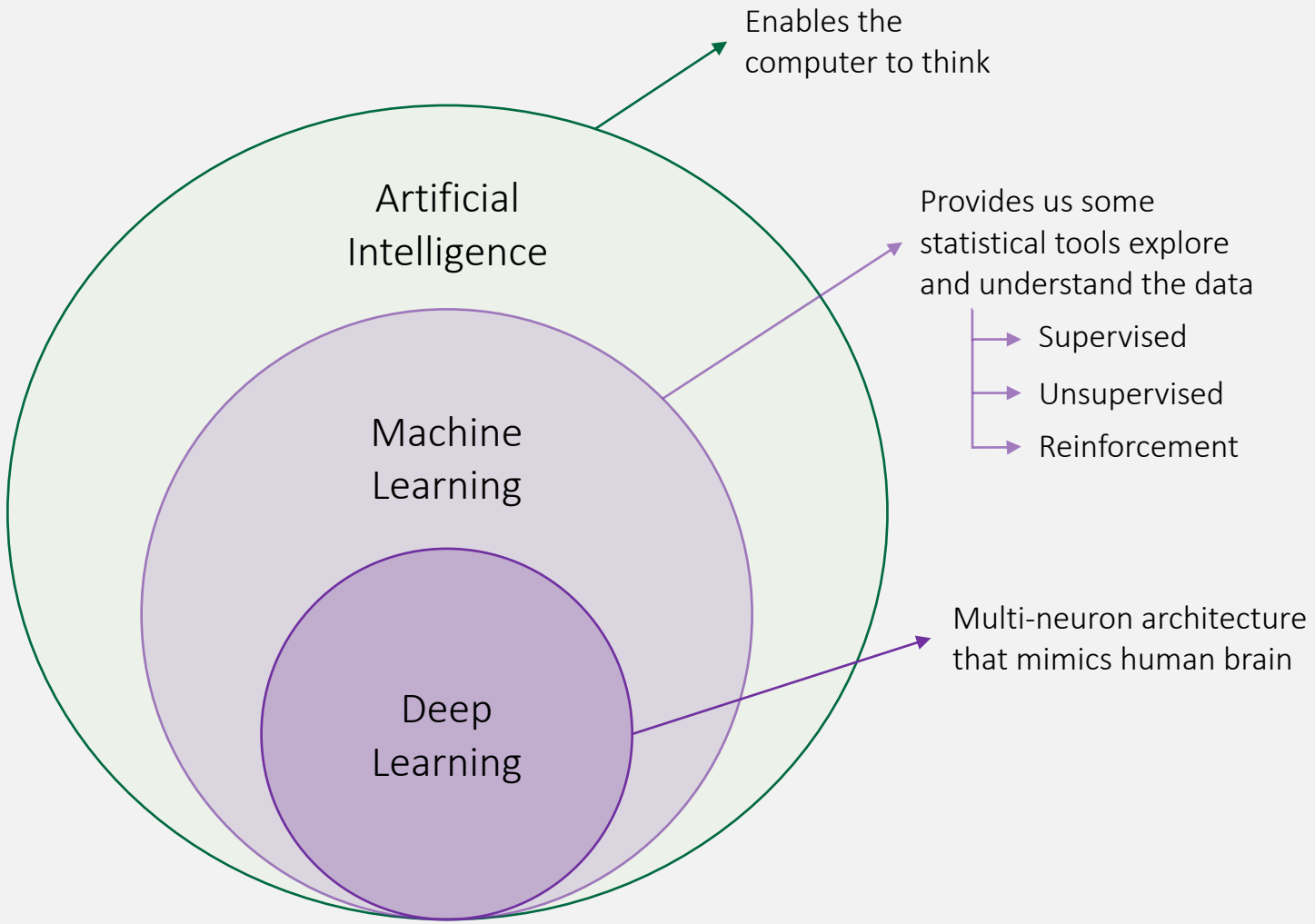
SIF 2021

Constance Douwes[1] and Philippe Esling[1]

[1] *IRCAM CNRS – UMR 9912, 1 Place Igor Stravinsky, F-75004 Paris, France*

douwes@ircam.fr

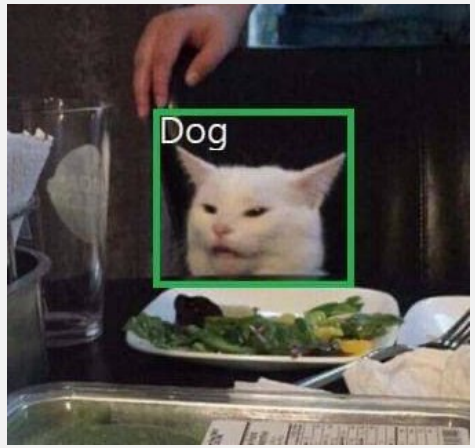
Artificial Intelligence



Modern (deep) learning

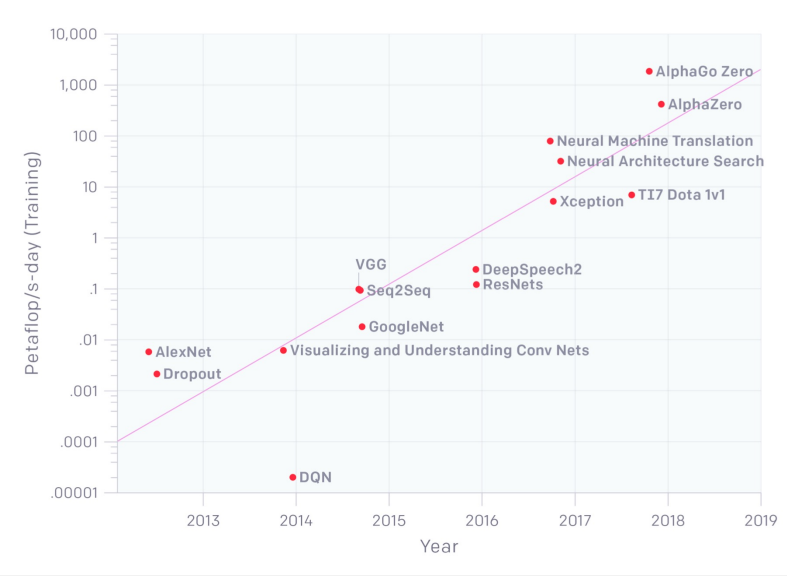
Deep learning holds most state-of-the-arts in various tasks :

- Image recognition, object detection, colorization, pixelization
- Music classification, generation, text-to-speech synthesis
- Language translation, data analysis



However deep learning suffers from several problems

- Networks can have up to billions of parameters
- Extremely demanding in computation, energy and memory
- Gains in accuracy now appear always linked to increased size

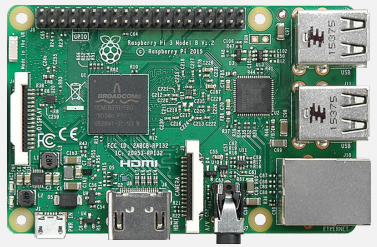


Dario Amodei and Danny Hernandez. AI and compute, 2018. Blog post.

Modern issues - Consequences

Direct consequences of this accuracy race :

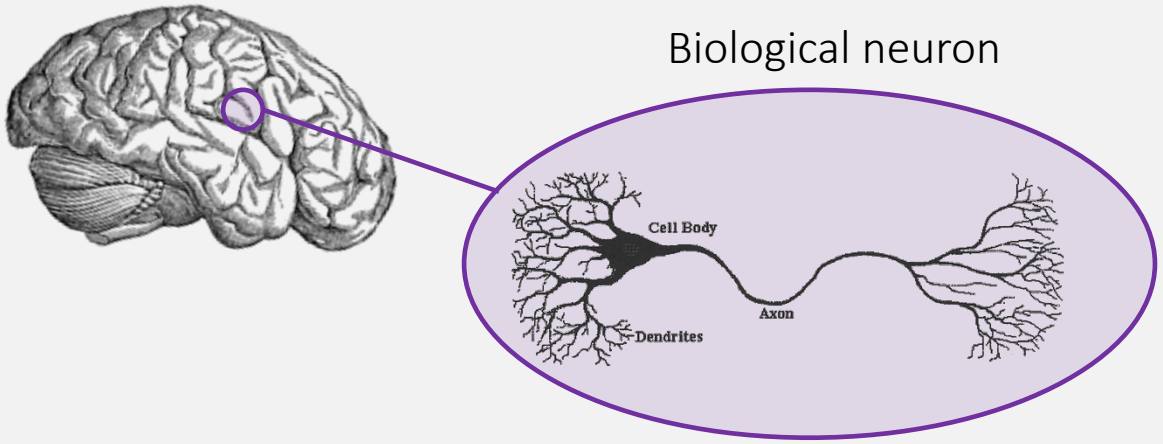
- Models are overparameterized and heavy computationally
- Huge environmental issue
- Precludes the use in *non-specialized* (user-side) hardware
- Even less possible for embedded systems

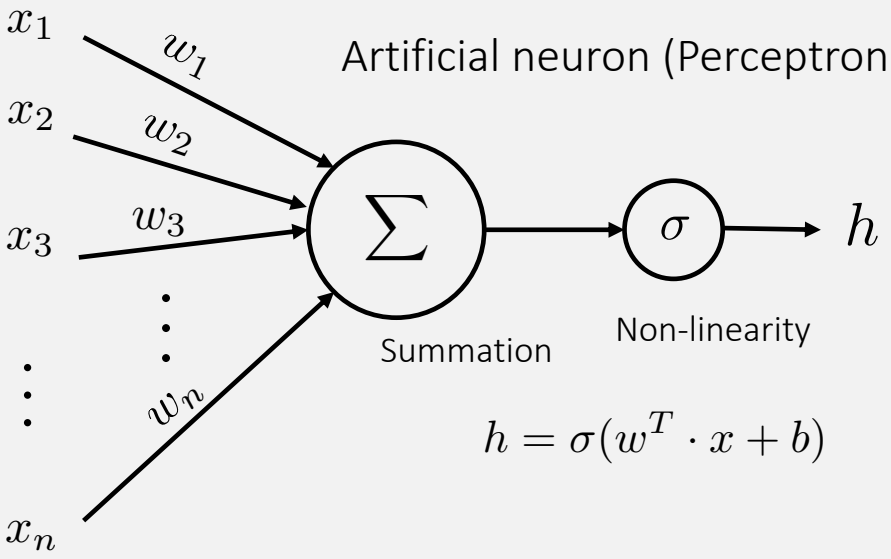


Example of GPT-3 model (NLP)

- 175 billon parameters and take 355 years on a single GPU to train
- Carbon footprint for training equivalent to driving to the moon and back

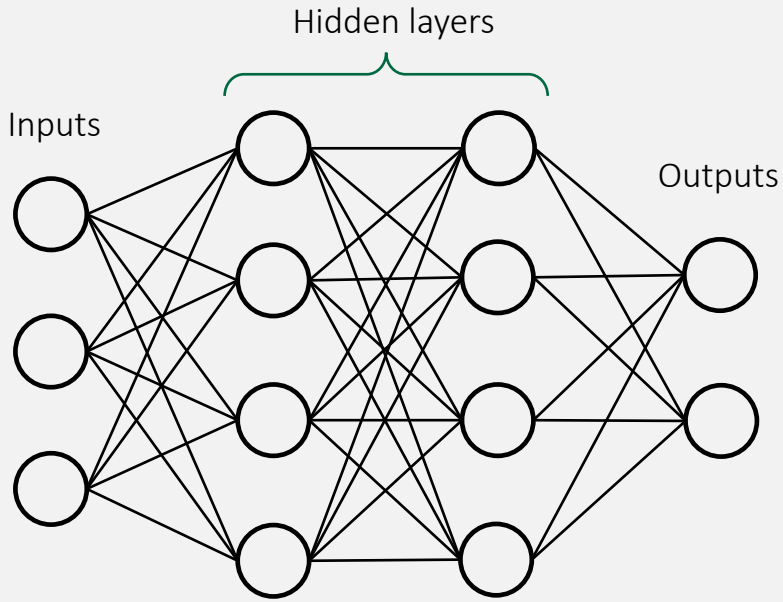
Deep learning - Architectures





Artificial neuron (Perceptron)

$$h = \sigma(w^T \cdot x + b)$$



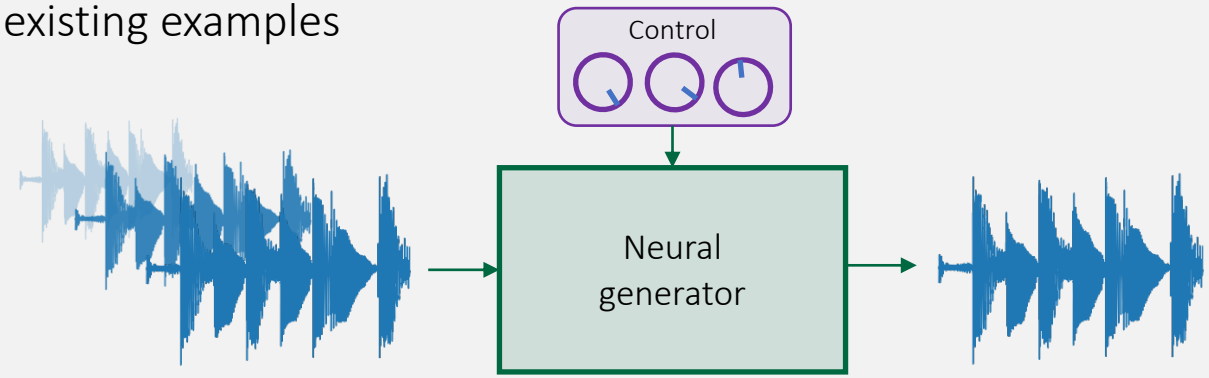
Fully-connected neural network

- More complex architectures : CNN, RNN, LSTM

Generative models for Audio

Generative models are a flourishing class of deep learning approaches

- Deal to generate novel data based on existing examples



Plurality of architectures :

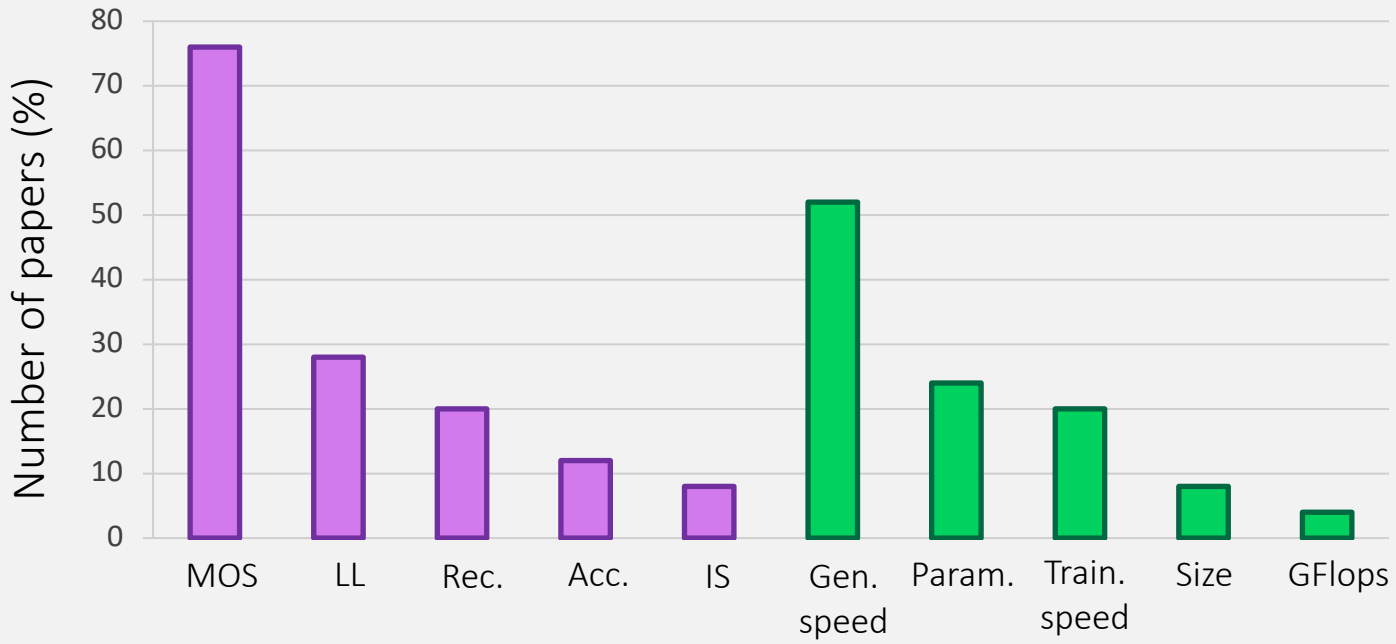
- Auto-Regressive: Heavy architectures, no direct control
- VAE: low-dimensional representation, blurry generation
- GAN: lack latent expressivity, difficult to optimize
- Normalizing Flows: complex distributions, no input reduction

?

How to evaluate/compare them?

Evaluation of models

Among the 28 surveyed papers (2016-2020) :



MOS	Mean Opinion Score
LL	Log-Likelihood
Rec.	Reconstruction
Acc.	Accuracy
IS	Inception Score

Gen. speed	Generation speed
Param.	Number of parameters
Train. speed	Training speed
Size	Memory size
Gflops	Gigaflops

- Most of used metrics are on “quality” either than “performance”
- No real energy-based criterion
- Best *trade-off* : quality or energy efficiency?

Pareto efficiency - Theory

Optimization problems involving conflicting objectives to be optimized simultaneously :

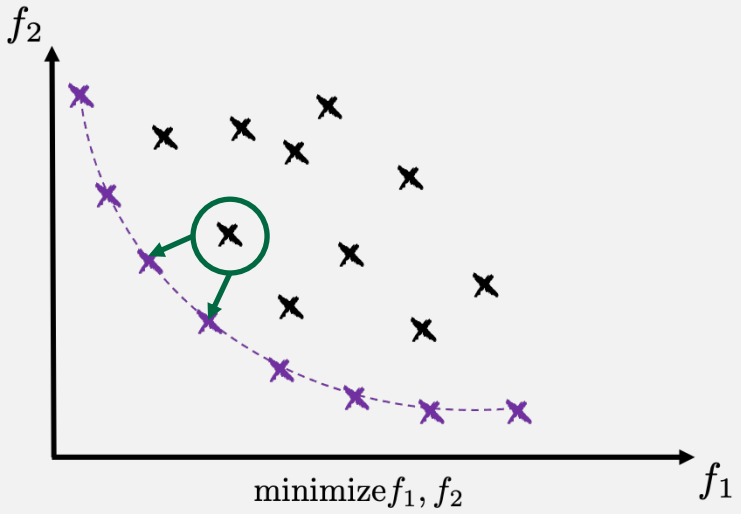
$$\min_{x \in X}(f_1(x), f_2(x), \ldots, f_k(x))$$

Let $\{x_a, x_b\} \in X \times X$. x_a is said to dominate x_b ($x_a \prec x_b$) ,if :

- $\forall i \in \{1, \ldots, k\}, f_i(x_a) \leq f_i(x_b)$
- $\exists j \in \{1, \ldots, k\}, f_j(x_a) < f_j(x_b)$

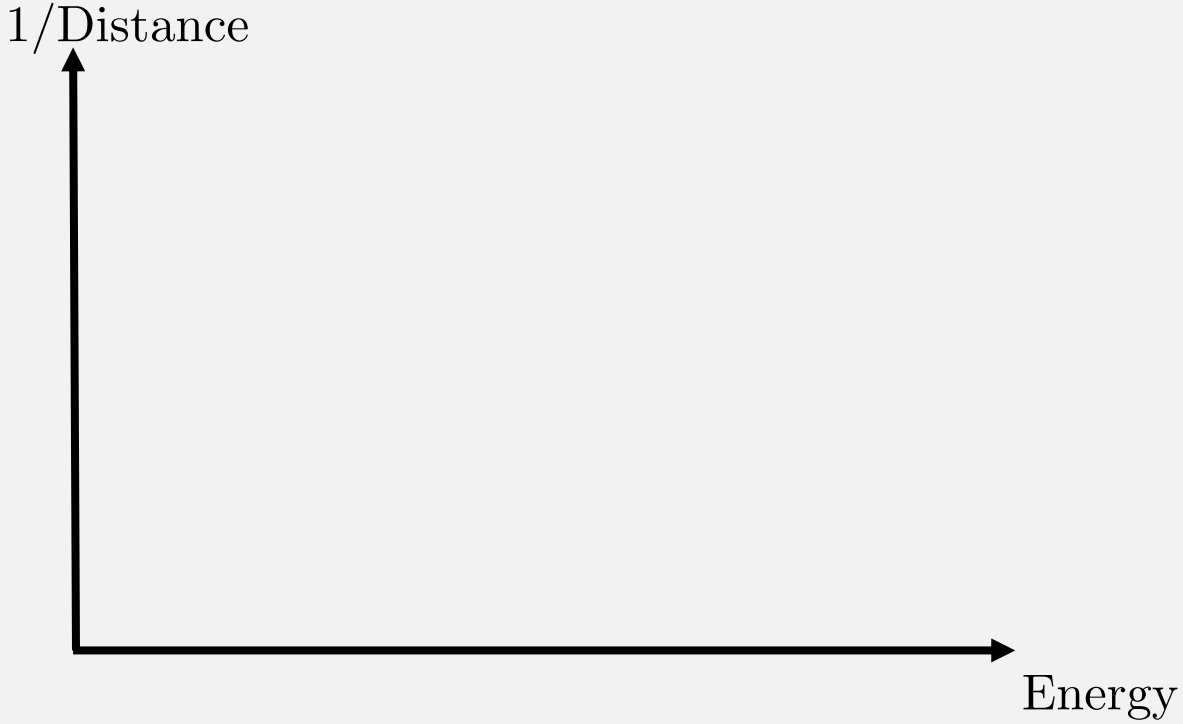
A solution $x^* \in X$ is a Pareto optimal point and $f(x^*)$ is a Pareto optimal objective vector if there does not exist $\hat{x}$ such that $\hat{x} \prec x^*$.

The set of all these pareto optimal solution is called the *Pareto front :*



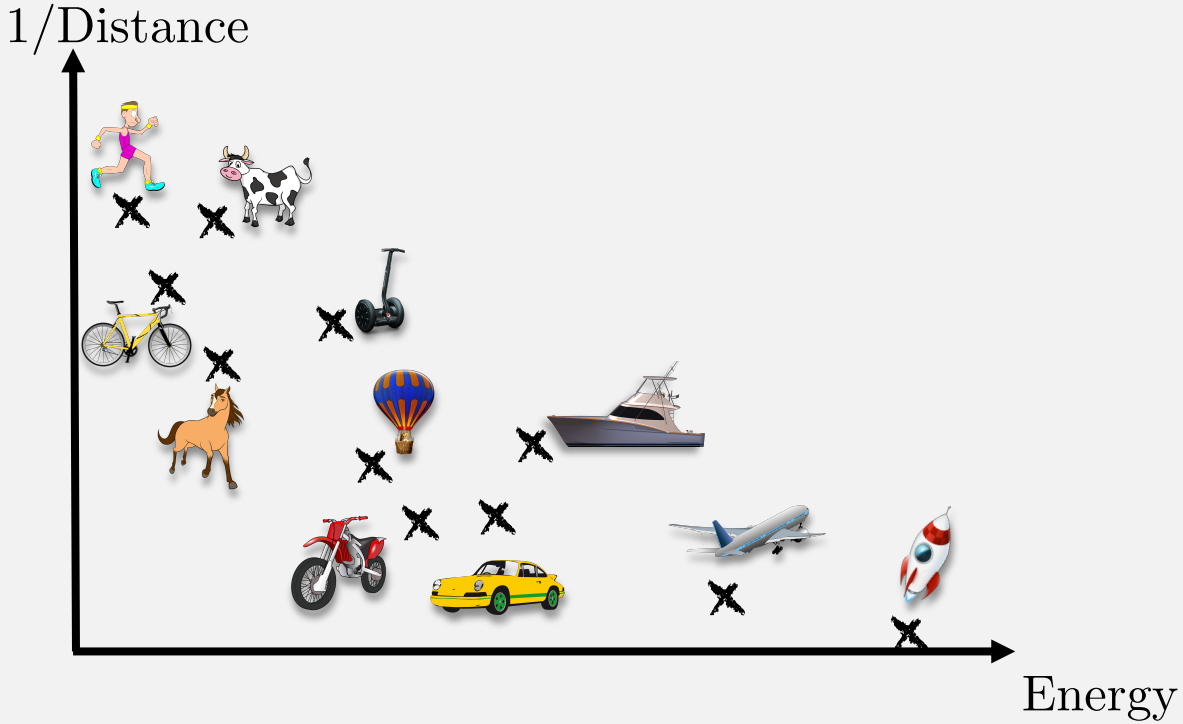
Pareto efficiency - Application

Energy efficiency of transportation modes according to the distance



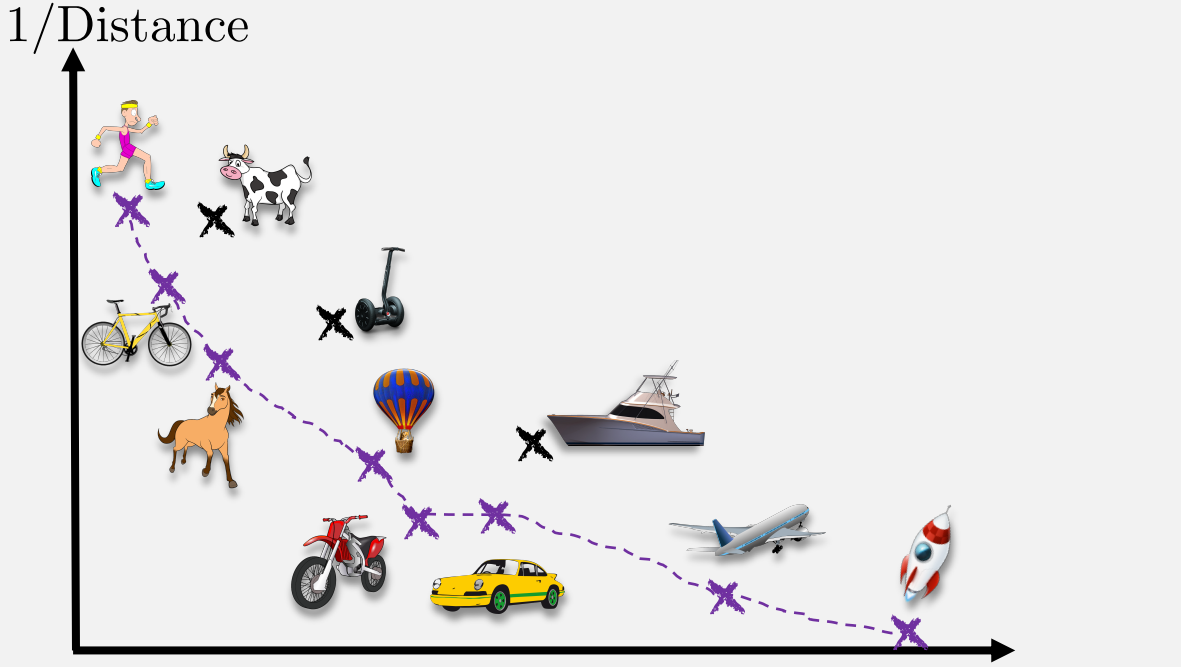
Pareto efficiency - Application

Energy efficiency of transportation modes according to the distance



Pareto efficiency - Application

Energy efficiency of transportation modes according to the distance



C-S. Sheih, Bi-objective optimization of distance and energy. Spring 2013

EXPERIMENTS & RESULTS

Training cost

- Training Time : depends on the model's implementation number & performance of GPU
- Electricity usage : still hardware - dependent but location- agnostic
- Carbon Emissions : real carbon footprint impact local electricity infrastructure

Carbon emissions estimation (in $kgC0_2eq$) per training can be expressed as :

$$\mathrm{CO_2e} = \alpha \times n \times p_{max} \times t$$

Lacoste et al. 2019. Quantifying the Carbon Emissions of Machine Learning

α	Electricity emission factor ($kgC0_2eq/kWh$)
n	Number of GPUs
p_{max}	Maximum Power of the GPU (kWatt)
t	Training time (Hours)

Training cost

- Training Time : depends on the model's implementation number & performance of GPU
- Electricity usage : still hardware - dependent but location- agnostic
- Carbon Emissions : real carbon footprint impact local electricity infrastructure

Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

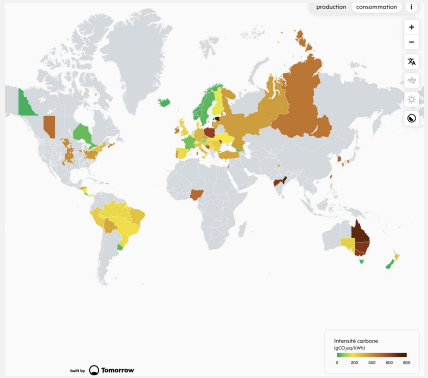
Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Strubell et al. 2019. Energy and Policy Considerations for Deep Learning in NLP

Carbon emissions estimation (in $kgCO_2eq$) per training can be expressed as :

$$\mathrm{CO_2e} = \alpha \times n \times p_{max} \times t$$

Lacoste et al. 2019. Quantifying the Carbon Emissions of Machine Learning



https://www.electricitymap.org/map

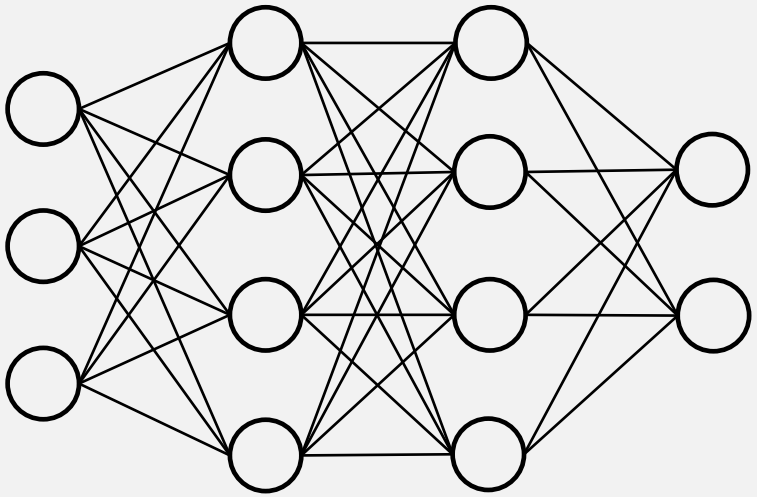
α Electricity emission factor ($kgCO_2eq/kWh$)
n Number of GPUs
p_{max} Maximum Power of the GPU (kWatt)
t Training time (Hours)

We took $\alpha = 0,437$ $\mathrm{kgCO_2eq/kWh}$ (2018 global average)

Model	Hardware	p_{max}	t	$\mathrm{CO_2e}$
SampleRNN	GTX TITAN X	0.25	168	18.4
SING	4 NVIDIA P100	1	52	22.7
WaveGAN	NVIDIA P100	0.25	96	10.5
GANSynth	NVIDIA V100	0.3	108	15.45
FloWaveNet	NVIDIA V100	0.3	272	35.7

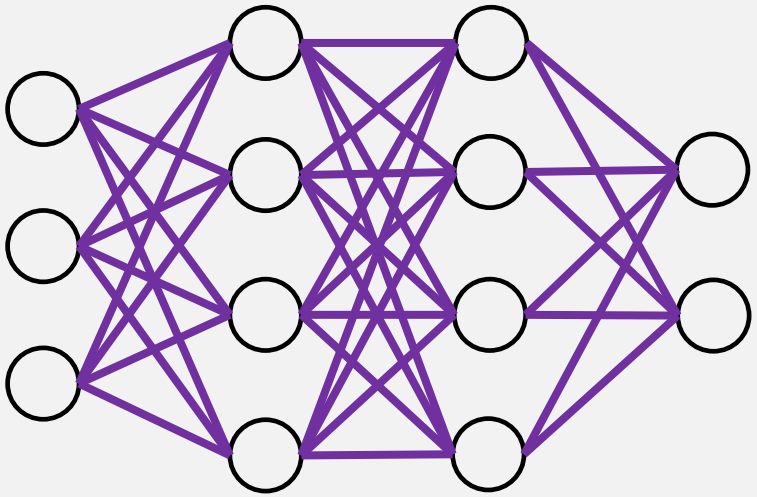
Inference cost

- Elapsed real time (sample/sec) : Other jobs running on the same device, number of cores
- Number of Floating Points Operations (FPOs) : location - independent but not straightforward
- Number of Parameters : Correlated with computational complexity different operations costs



Inference cost

- Elapsed real time (sample/sec) : Other jobs running on the same device, number of cores
- Number of Floating Points Operations (FPOs) : location - independent but not straightforward
- Number of Parameters : Correlated with computational complexity different operations costs



Model	Number of parameters
SampleRNN	52M
SING	64M
WaveGAN	89M
GANSynth	15M
FloWaveNet	183M

```
PyTorch : sum(p.numel() for p in model.parameters())
Tensorflow & Keras : model.summary()
```

Quality score

Generation quality measurements are plural:

- We rely on the MOS as it is the most popular measure
- This score is highly dependent on each experimental setup
- We compute : $\%\text{MOS} = \dfrac{\text{MOS}_{Model}}{\text{MOS}_{GroundThruth}}$

The goal is to maximize this ratio, and thus to minimize $1 - \%\text{MOS}$

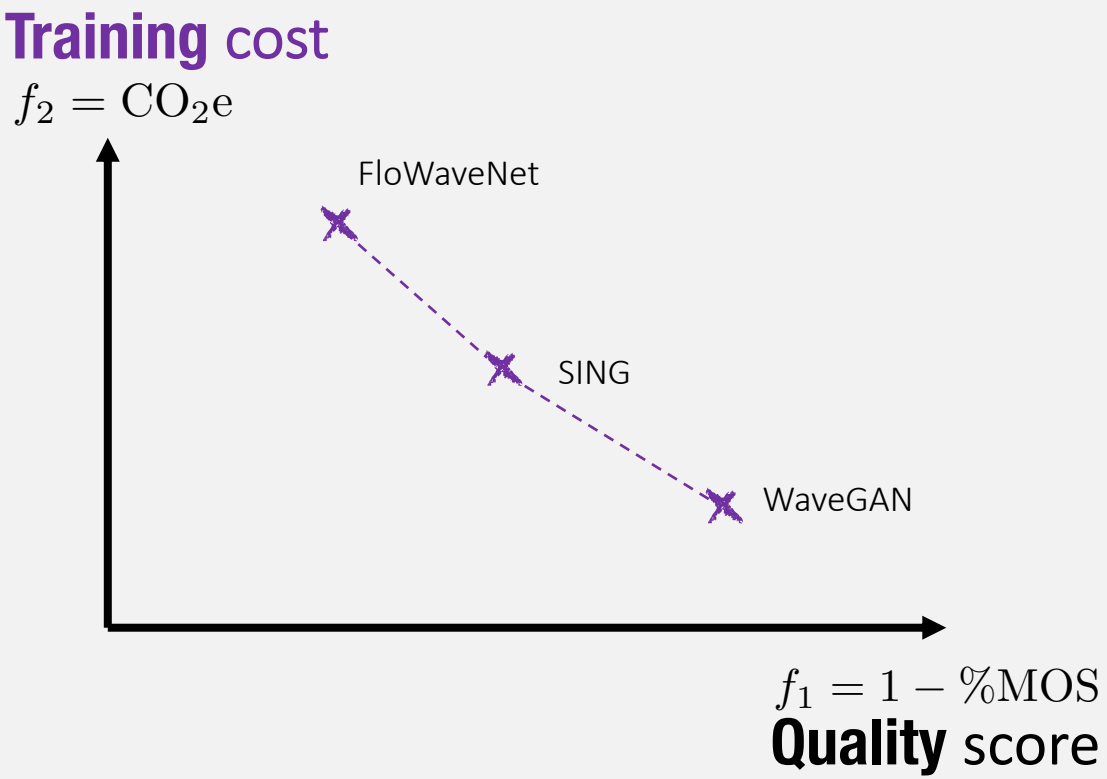
Model	MOS_{Model}	$\text{MOS}_{GroundTruth}$	1-%MOS
SampleRNN	-	-	-
SING	$2,8 \pm 0,24$	$3,86 \pm 0,24$	0,26
WaveGAN	$2,3 \pm 0,9$	$3,9 \pm 0,9$	0,41
GANSynth	-	-	-
FloWaveNet	$3,95 \pm 0,15$	$4,67 \pm 0,08$	0,15

METHODS	5-SCALE MOS
GROUND TRUTH	4.67± 0.076
MOL WAVENET	4.30± 0.110
GAUSSIAN WAVENET	4.46± 0.100
GAUSSIAN IAF	3.75± 0.159
FLOWAVENET	3.95± 0.154

Model	MOS
Ground Truth	3.86 ± 0.24
Wavenet	2.85 ± 0.24
SING	3.55 ± 0.23

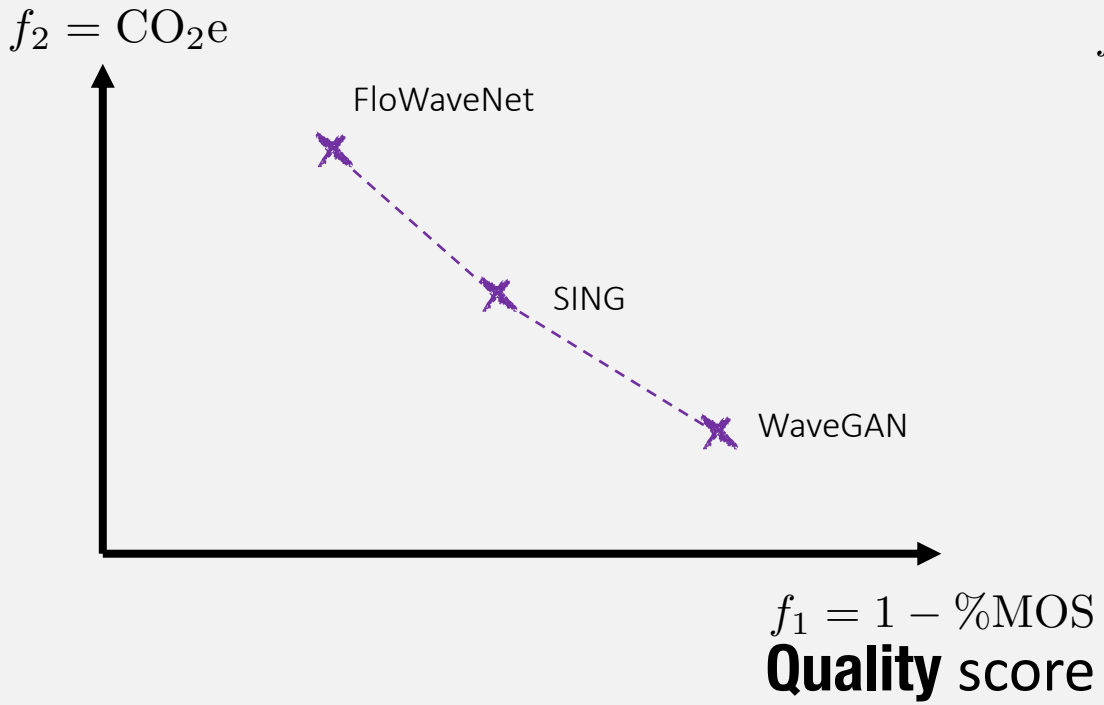
Experiment	Quality
Real (train) Real (test) Parametric	3.9 ± 0.8
WaveGAN + Phase shuffle $n = 2$ + Phase shuffle $n = 4$	2.3 ± 0.9

Results

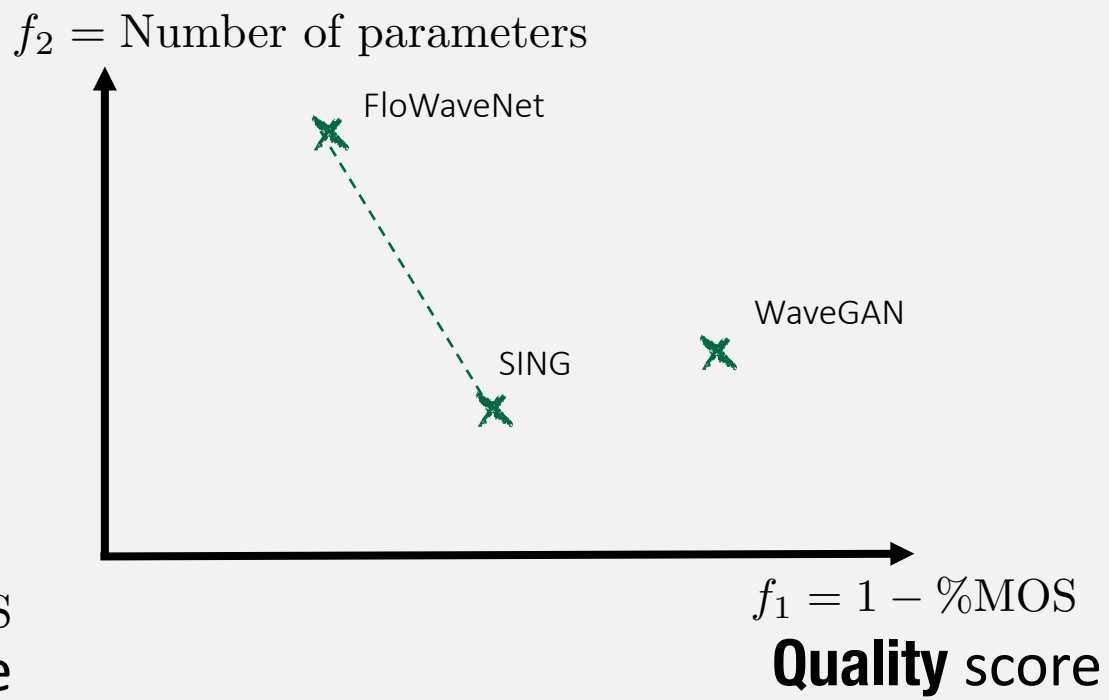

Training cost
$f_2 = \mathrm{CO_2e}$
FloWaveNet
SING
WaveGAN
$f_1 = 1 - \%\mathrm{MOS}$
Quality score

Results

Training cost



Inference cost



CONCLUSIONS & PERSPECTIVES

Conclusions

- The lack of training details affected our work : authors must report the training time & hardware or use online tool[1] to report actual CO_2
- Models that are sub-optimal should be discredited from publications
- Our approach is generic, and could be applied to any type of model or input data

Perspectives

- Automatic implementation to count FPOs
- Exhibit a training/inference ratio
- Run experiments in another field of AI

[1] https://mlco2.github.io/impact/

THANKS!