Machine Learning in Physics

Course plan:

Part one – fundamentals

1. **Machine Learning fundamentals**: definitions, machine learning approaches, typical machine learning tasks, learning scheme, cross validation, performance metrics, bias-variance tradeoff, overfitting, domain/covariate shift, no free lunch theorems, precision-recall tradeoff.
2. **Neural network architectures (part one):** deep neural networks (NNs), convolutional NNs. Universal approximation theorem, neural network training schemes, activation functions, hyperparameters, regularization, dropout, augmentation, transfer learning.
3. **Classification problem revisited**: decision trees, trees growing algorithms, ensemble methods: bootstrapping, bagging, boosting. Examples: random forests, AdaBoost algorithms.
4. **Neural network architectures (part two, advanced ideas):** autoencoders, VAEs, recurrent NNs. Generative models: GANs, diffusion models (Stable Diffusion).
5. **Attention mechanism and Transformers**, Large language models.

Machine Learning in Physics

Course plan:

Part two – learning physics (by neural networks)

6. **Imposing physics through learning**. Physics-informed Neural Networks, example applications: PDEs, e.g. fluids, dynamical systems. Siamese Networks and contrastive learning, self- and semi-supervised learning.
7. **Graph neural networks** and their applications: social networks, quantum chemistry; Neural network as variational principle: many-body ground state estimation.
8. **Geometric deep learning**: imposing physics through network architecture, Group-equivariant Convolutional Neural Networks. Gauge Equivariant CNNs.
9. **Bayesian Learning,** estimating models' uncertainty. **Neural tangent kernel** and Kernel Methods.
10. **Inverse problems** from deep learning perspective.

Machine Learning in Physics

General literature:

- I. Goodfellow , Y. Bengio, and A. Courville, *Deep Learning,* 2015. https://www.deeplearningbook.org
- K. Murphy, *Probabilistic Machine Learning: An Introduction*, 2022. https://probml.github.io/pml-book/book1.html
- K. Murphy, *Probabilistic Machine Learning: Advanced Topics*, 2022. https://probml.github.io/pml-book/book2.html

Machine learning in physics (or science):

- A. Tanaka, A. Tomiya, K. Hashimoto, *Deep Learning and Physics*, 2022.
- D. A. Roberts, S. Yaida, B. Hanin, *The Principles of Deep Learning Theory*, 2022.
- P. Grohs, G. Kutyniok, *Mathematical Aspects of Deep Learning*, 2022.
- M. M. Bronstein, J. Bruna, T. Cohen, P. Veličković, *Geometric Deep Learning*, 2024. https://geometricdeeplearning.com/book/

Machine Learning fundamentals

Materials for this section:

- I. Goodfellow , Y. Bengio, and A. Courville, *Deep Learning, Chapter 5: Machine Learning Basics*, https://www.deeplearningbook.org
- Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature **521,** 436 (2015).
- https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf (2012).

General reading:

- Kevin Murphy, *Probabilistic Machine Learning: An Introduction*, 2022.
- https://lilianweng.github.io

What is machine learning?

Family of computer methods that improve **automatically** through **experience**
- without being *explicitly* programmed to
- in opposition to standard computer science approach – direct programing

<u>What kind of problems we would like (to be able) to solve?</u>
- the early days of AI: problems that are intellectually difficult for humans but straightforward for computers
- nowadays, the *true* challenge to AI are tasks easy for people to perform but **hard to be described formally**
  - like recognizing spoken words or faces on images

The **machine learning** (ML) solution to these problems:
- allow computers to learn from experience and understand the world in terms of a hierarchy of concepts

this approach avoids the need for human to specify all the knowledge that the computer needs.

The hierarchy of concepts enables the computer to learn complicated concepts by building them out of simpler ones
- representations of data with multiple (**deep**) levels or *layers* of abstraction

What is deep learning?

If we draw a graph showing how these concepts are built on top of each other, the graph is *deep*, with many layers. For this reason, we call this approach to AI **deep learning**

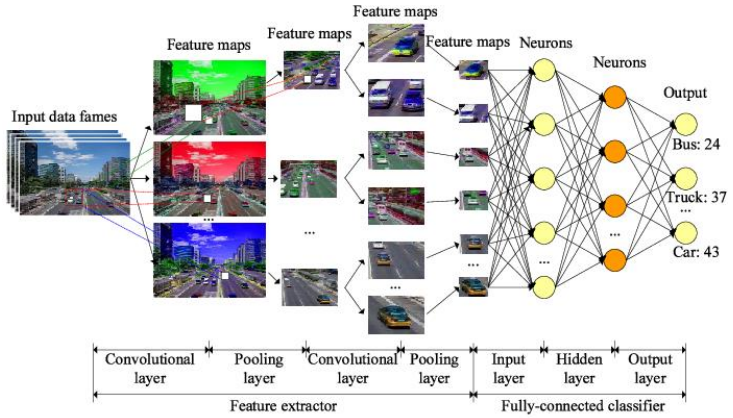
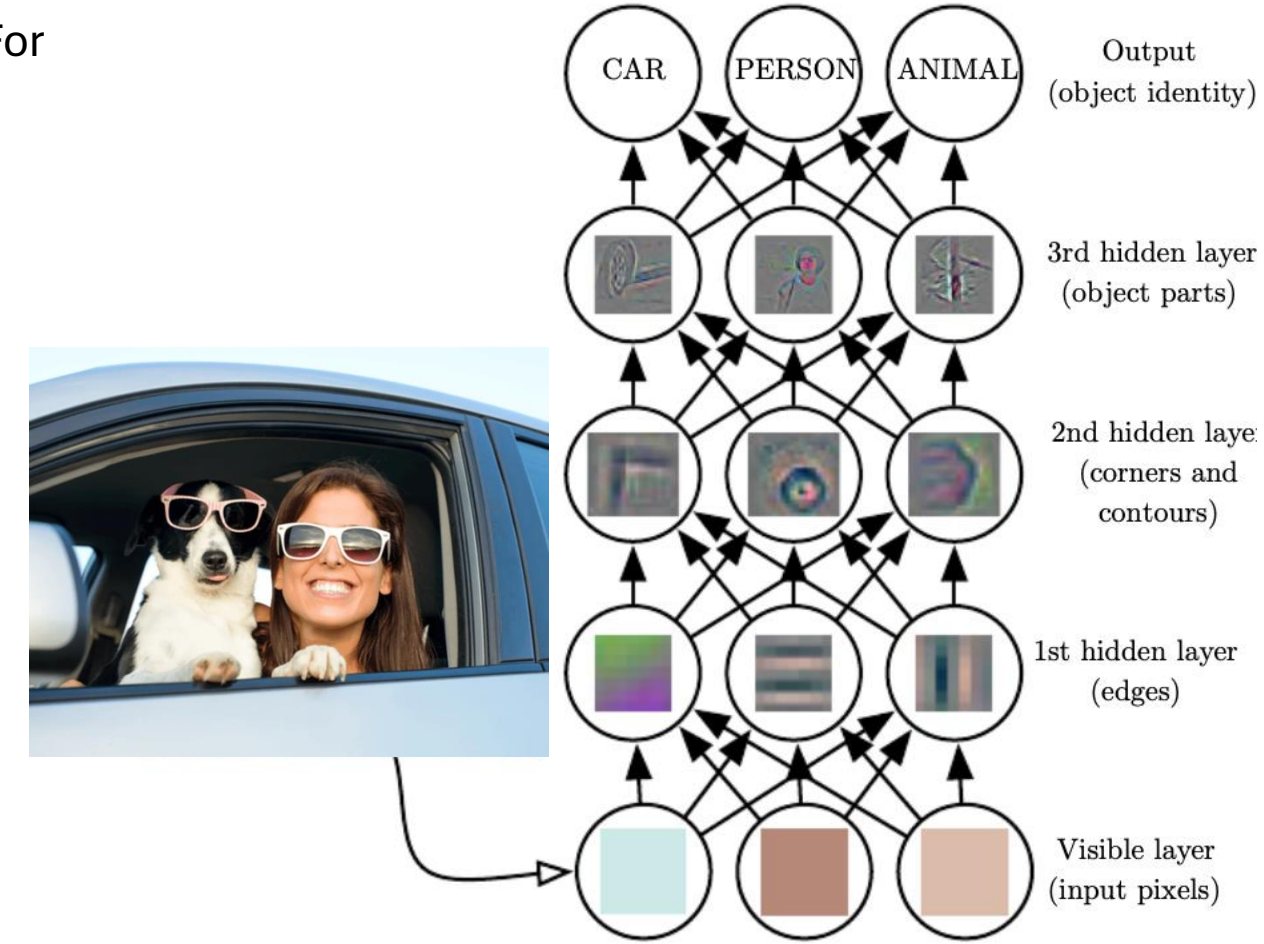


Fig. 3. Example of the CNN model for vehicle classification.

The model efficiently deconstruct the problem (image classification) into multiple levels/layers of abstraction **by itself.**



https://www.deeplearningbook.org/contents/intro.html

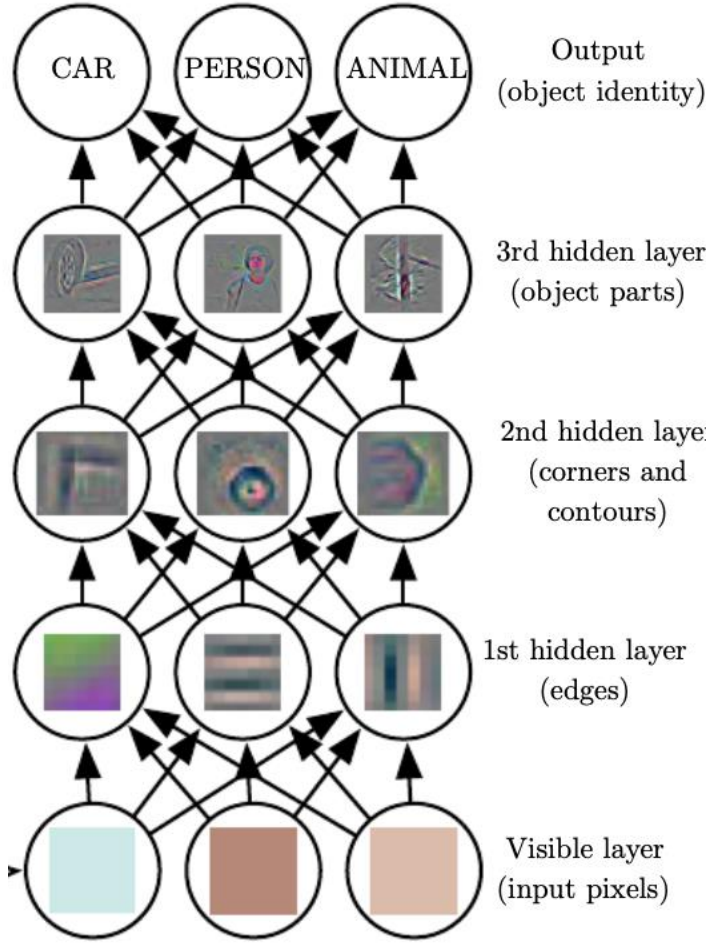
What is deep learning?

How to efficiently solve these problems – *automatically* learn from data

A **machine learning** algorithm is an algorithm that is able to efficiently learn from data and gradually *improve* its performance by **extracting patterns** from raw *data*.

Deep learning is a type of ML

- **deep learning** solves problems by introducing representations that are expressed in terms of other, simpler representations
- deep learning enables the computer to automatically learn complex concepts out of simpler ones



https://www.deeplearningbook.org/contents/intro.html

What is machine learning?

Typical ML tasks:

- classification $f : \mathbb{R}^n \to \{1, \ldots, k\}$.
- regression $f : \mathbb{R}^n \to \mathbb{R}$

- transcription (OCR/ASR), translation
- extract word and whole sentences from written or spoken data

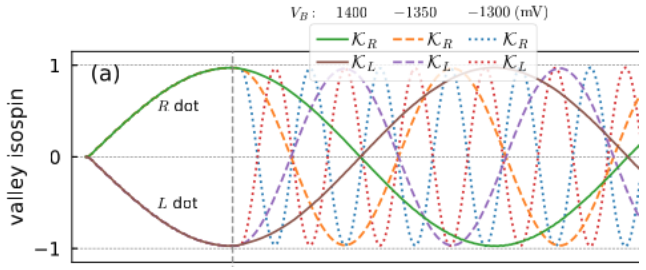
- anomality detection
- detect events or objects that are unusual in a given context

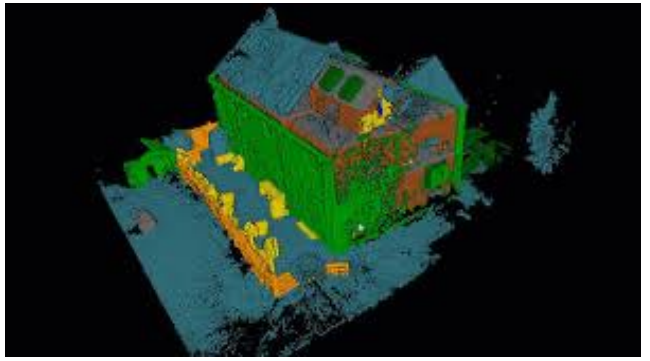
- object detection
- detect objects/events on images / in signals. Expl: fraud detection.

- synthesis and sampling
- generate new examples that are similar to those in the training data (e.g. GANs)
- denoising
- predict clean sample x from the corrupted one $\tilde{x}$

Example data dimensionality:

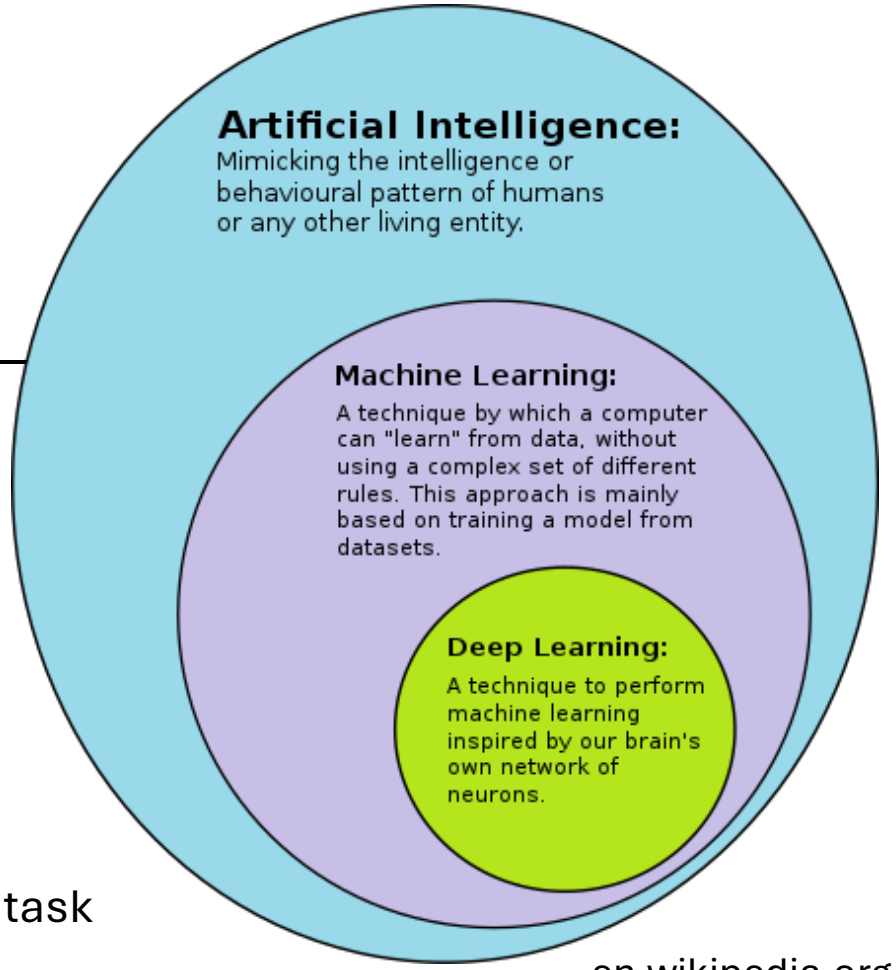
- 1d signal
- 2d image
- 3d point cloud

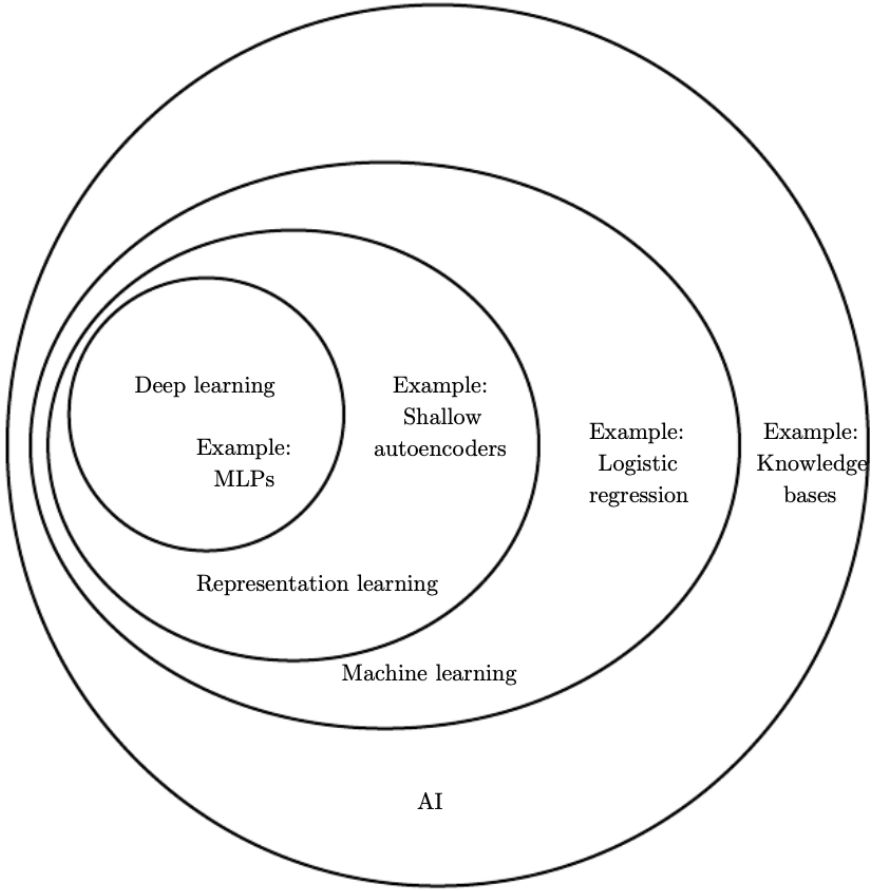




What is machine learning?

To summarize AI technologies



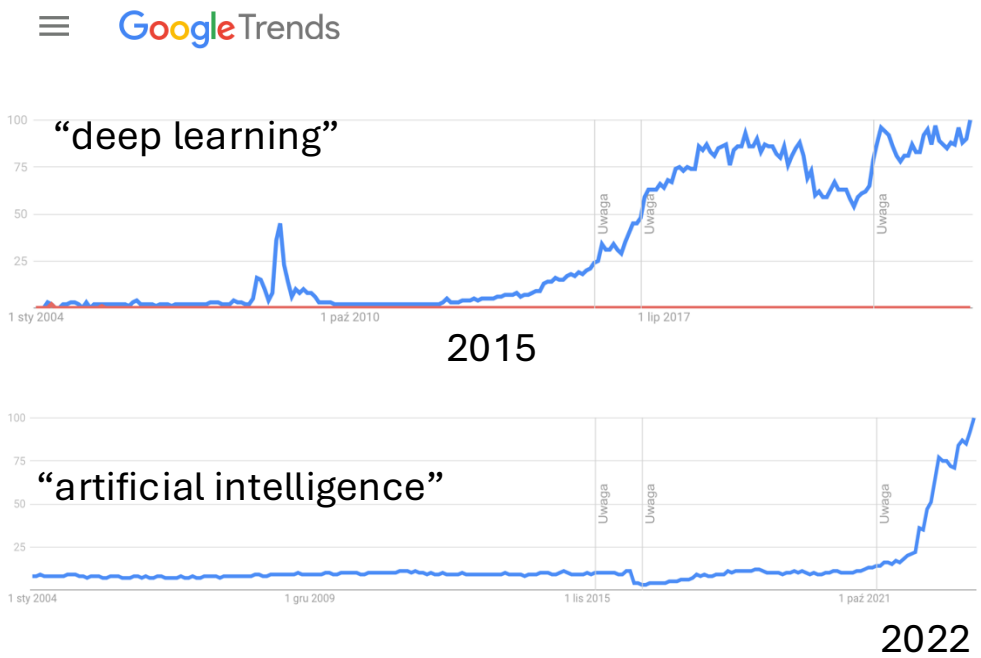


- AGI > AI
- weak AI = solve one specific task

en.wikipedia.org/wiki/Deep_learning

Why deep learning became so popular?

Why deep learning became a crucial technology in last ten years?



- Increasing dataset sizes
- a dumb algorithm with lots of data beats a clever algorithm with a modest amount of data

- is connected with increasing size of models

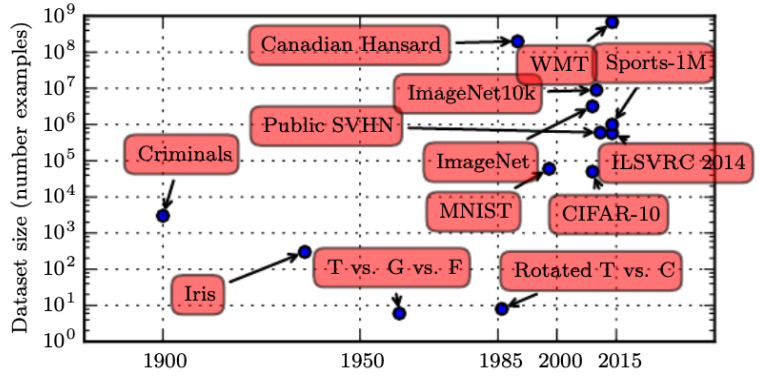


Figure 1.8: Increasing dataset size over time. In the early 1900s, statisticians studied datasets using hundreds or thousands of manually compiled measurements (Garson, 1900;

https://www.deeplearningbook.org/contents/intro.html

MNIST Dataset

- 60,000 examples, 10 classes
- features: 28x28x1
- http://yann.lecun.com/exdb/mnist/

ImageNet Dataset

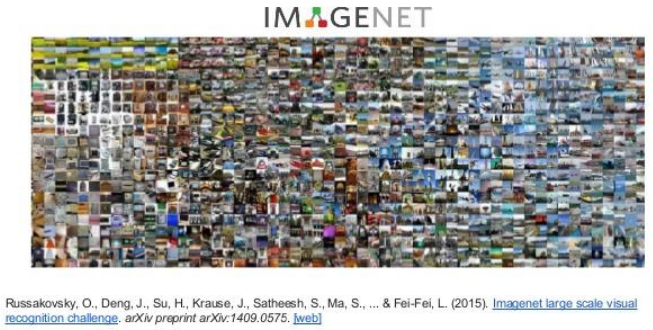


image-net.org: 14 M images, 1000 categories

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

cocodataset.org

Why deep learning became so popular?

How to train such a big models?

- big means that they can't be trained using CPU in a reasonable time

CPU vs GPU

	Cores	Clock Speed	Memory	Price	Speed
CPU (Intel Core i7-7700k)	4 (8 threads with hyperthreading)	4.2 GHz	System RAM	$339	~540 GFLOPs FP32
GPU (NVIDIA GTX 1080 Ti)	3584	1.6 GHz	11 GB GDDR5 X	$699	~11.4 TFLOPs FP32
TPU NVIDIA TITAN V	5120 CUDA, 640 Tensor	1.5 GHz	12GB HBM2	$2999	~14 TFLOPs FP32 ~112 TFLOP FP16
TPU Google Cloud TPU	?	?	64 GB HBM	$6.50 per hour	~180 TFLOP

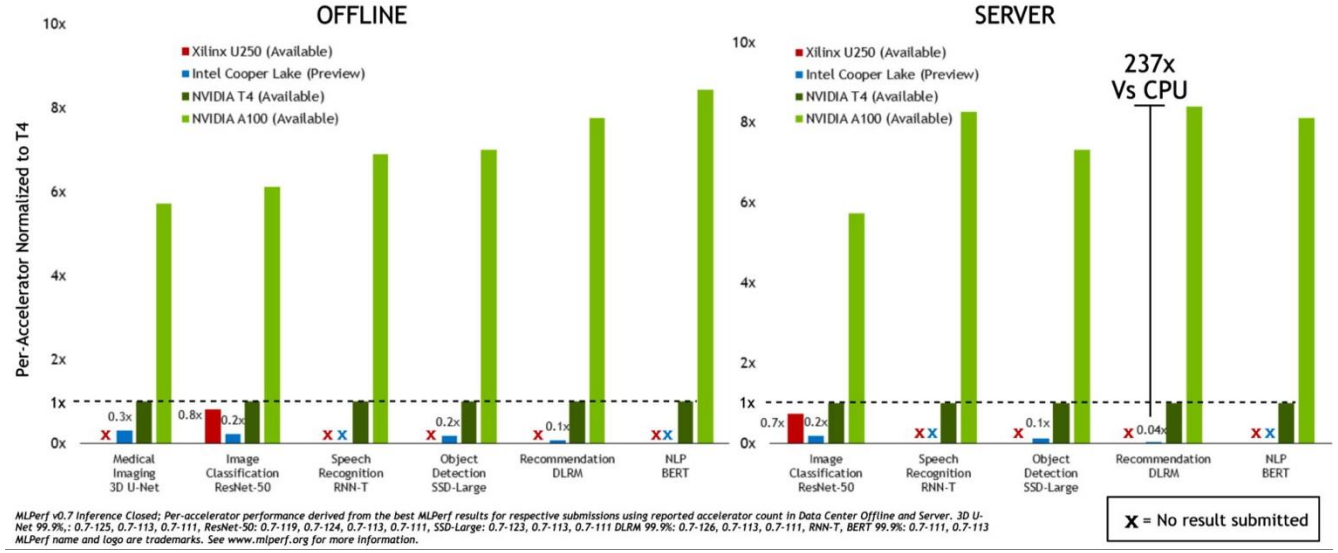
CPU: Fewer cores, but each core is much faster and much more capable; great at sequential tasks

GPU: More cores, but each core is much slower and "dumber"; great for parallel tasks

TPU: Specialized hardware for deep learning

CPU GPU TPU

ASICs specifically designed for deep learning

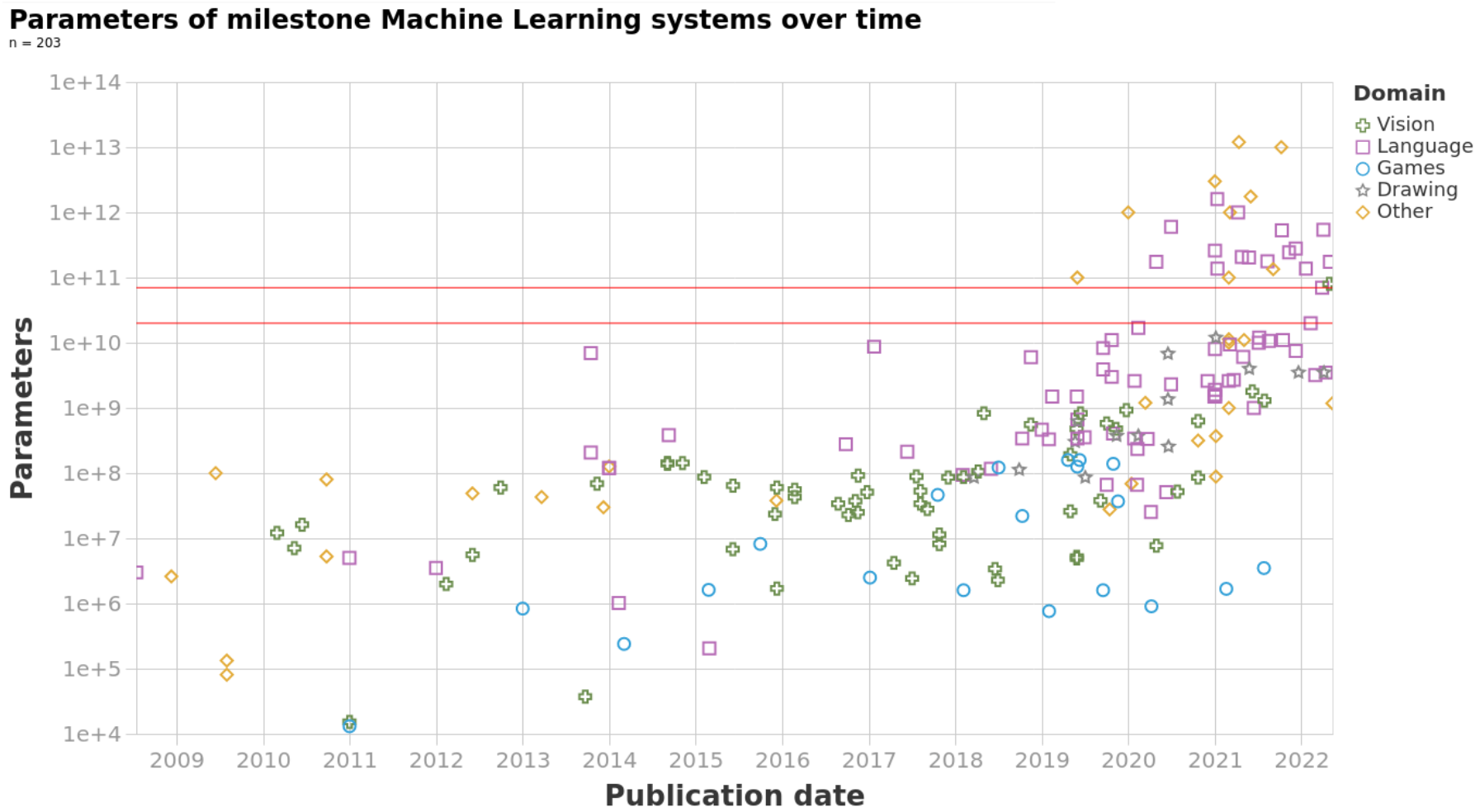


source: nvidia.com

- We can train big models on GPU dozens or hundreds times faster than on CPU
- GPUs have limitations: most of cores are processing in single or half-precision

https://medium.com/dataseries/cpu-gpu-and-tpu-machine-learning-5033e25b8a0f

Why deep learning became so popular?



- NLP models are the biggest: several billions of parameters
- largest models outside NLP: hundreds of millions of neural units

epochai.org/blog/machine-learning-model-sizes-and-the-parameter-gap

Why deep learning became so popular?

NLP models are the biggest

https://arxiv.org/pdf/2101.03961.pdf

OpenAI GPT-3 Architecture

The GPT-3 is not one single model but a family of models. Each model in the family has a different number of trainable parameters. The following table shows each model, architecture and its corresponding parameters:

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Sizes, architectures, and learning hyper-parameters of the GPT-3 models

GPT-4 ~ 1 T params

10^{12}

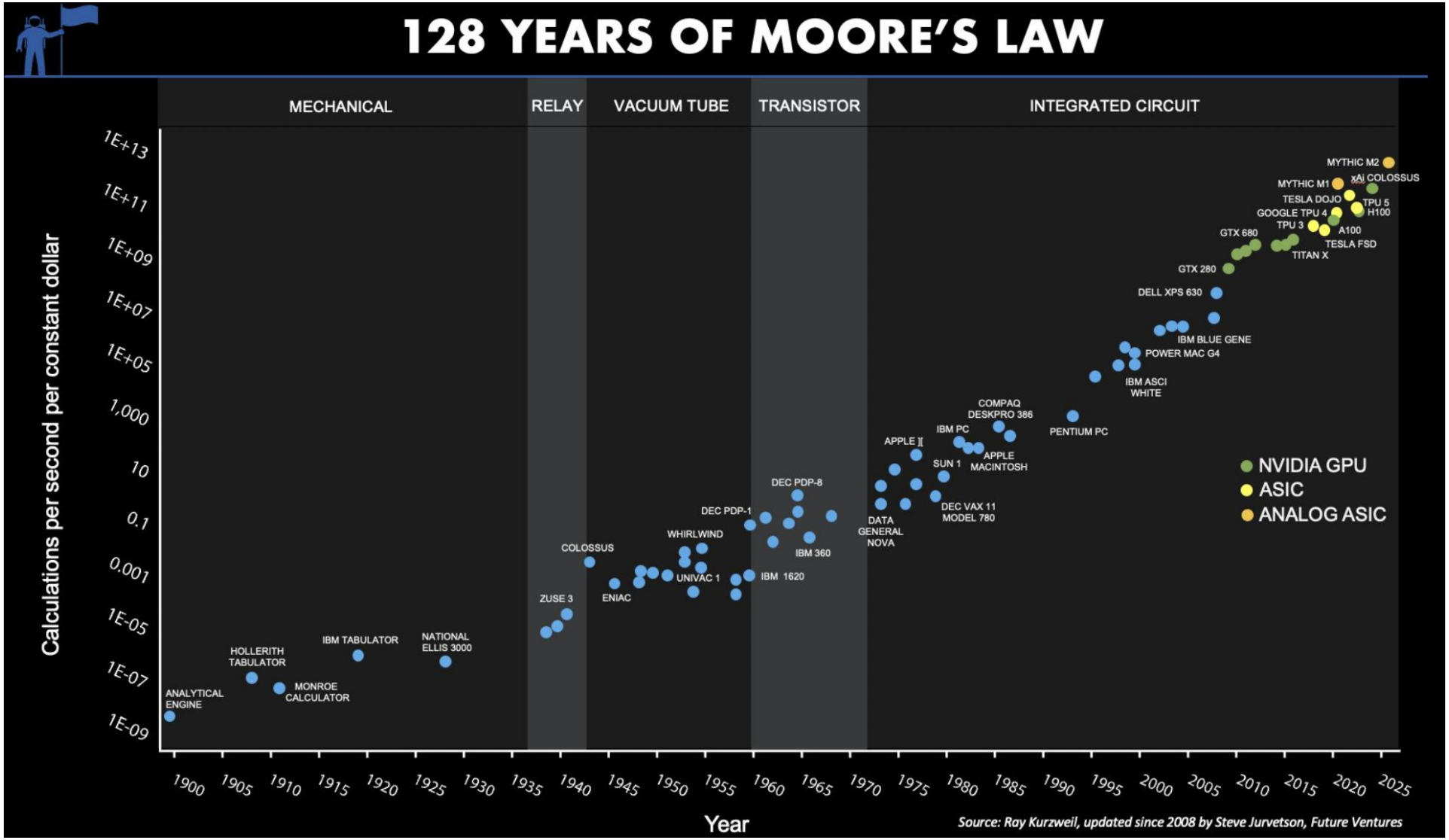
SWITCH TRANSFORMERS: SCALING TO TRILLION PARAMETER MODELS WITH SIMPLE AND EFFICIENT SPARSITY

William Fedus* Google Brain liamfedus@google.com
Barret Zoph* Google Brain barretzoph@google.com
Noam Shazeer Google Brain noam@google.com

ABSTRACT

In deep learning, models typically reuse the same parameters for all inputs. Mixture of Experts (MoE) models defy this and instead select *different* parameters for each incoming example. The result is a sparsely-activated model – with an outrageous number of parameters – but a constant computational cost. However, despite several notable successes of MoE, widespread adoption has been hindered by complexity, communication costs, and training instability. We address these with the Switch Transformer. We simplify the MoE routing algorithm and design intuitive improved models with reduced communication and computational costs. Our proposed training techniques mitigate the instabilities, and we show large sparse models may be trained, for the first time, with lower precision (bfloat16) formats. We design models based off T5-Base and T5-Large (Raffel et al., 2019) to obtain up to 7x increases in pre-training speed with the same computational resources. These improvements extend into multilingual settings where we measure gains over the mT5-Base version across all 101 languages. Finally, we advance the current scale of language models by pre-training up to trillion parameter models on the "Colossal Clean Crawled Corpus", and achieve a 4x speedup over the T5-XXL model.[1]

Future of deep learning?



Future of deep learning?

LLMs scaling laws?

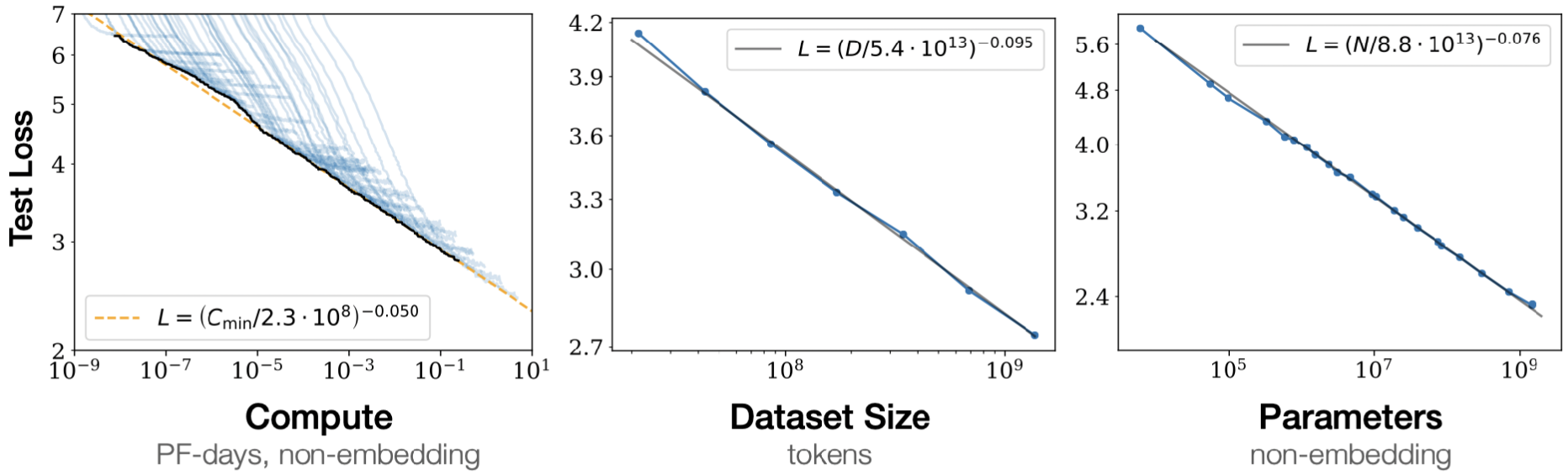
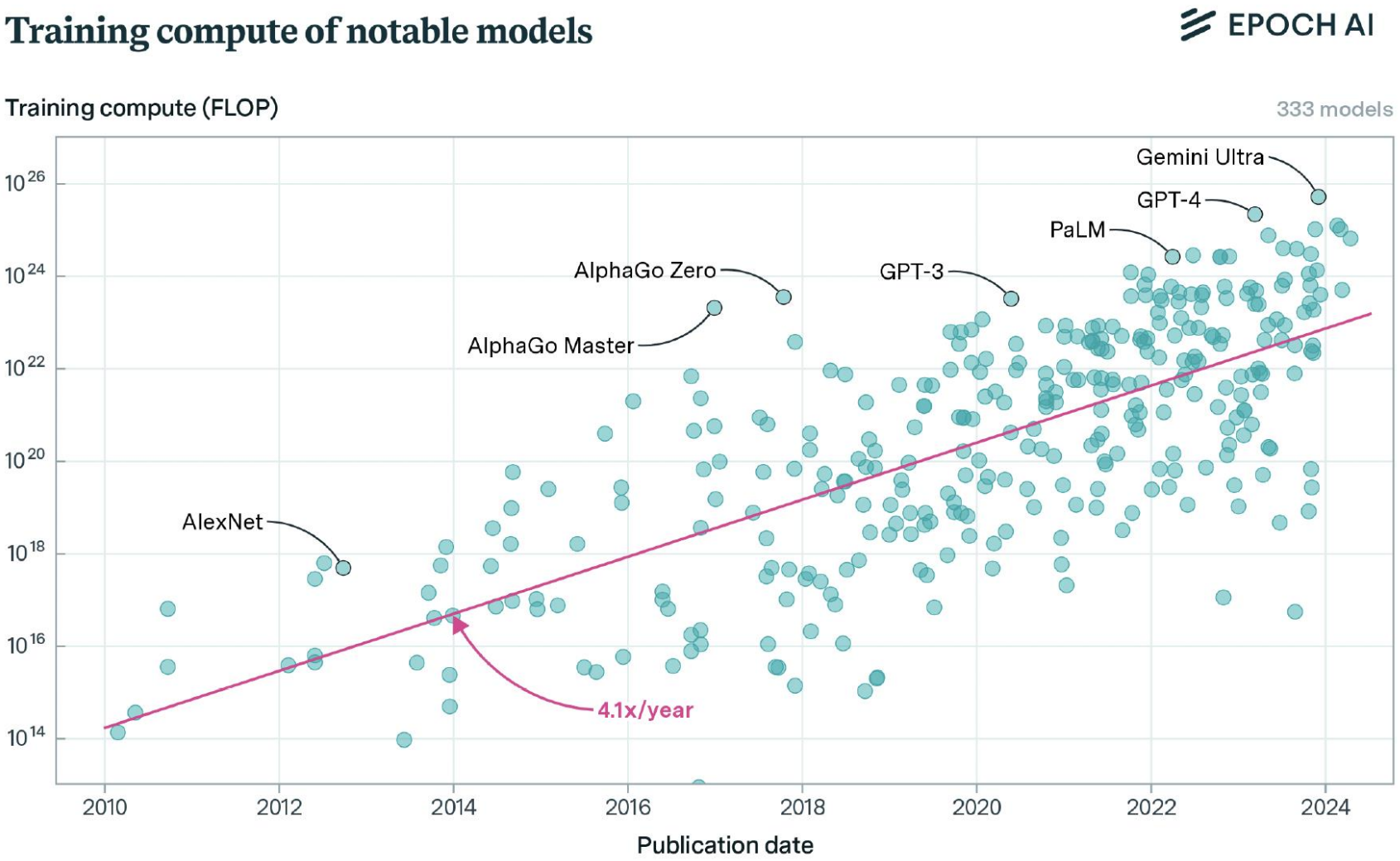


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

https://arxiv.org/pdf/2005.14165

1 PF-days = PetaFLOP/s machine working through 1 day (number of compute operations)

Future of deep learning?



No sign of stopping (scaling up) development (yet?)

Future of deep learning?

MATH dataset: high-school level mathematical problems

Hendrycks MATH – 12.5k (total, 5k in test set)
middle/high-school level competition-style problems

Problems scraped from art-of-problem-solving forum; range in difficulty from level 1 (trivial) to level 5 (easy AIME level)

Categories:
Geometry, Number theory, Pre-algebra, Algebra, Intermediate algebra, Counting, Precalculus

Question: What are the eigenvalues of a 2-by-2 matrix $M = \begin{pmatrix} 1 & 2 \\ -1 & 5 \end{pmatrix}$?

Question: Let

$$f(n) = \begin{cases} n^3 + 2n - 1 & \text{if } n > 1, \\ n - 1 & \text{if } n \leq 1. \end{cases}$$

Find $f(0) + f(1) + f(2)$.

Question: Mary has 6 identical basil plants, and three different window sills she can put them on. How many ways are there for Mary to put the plants on the window sills?

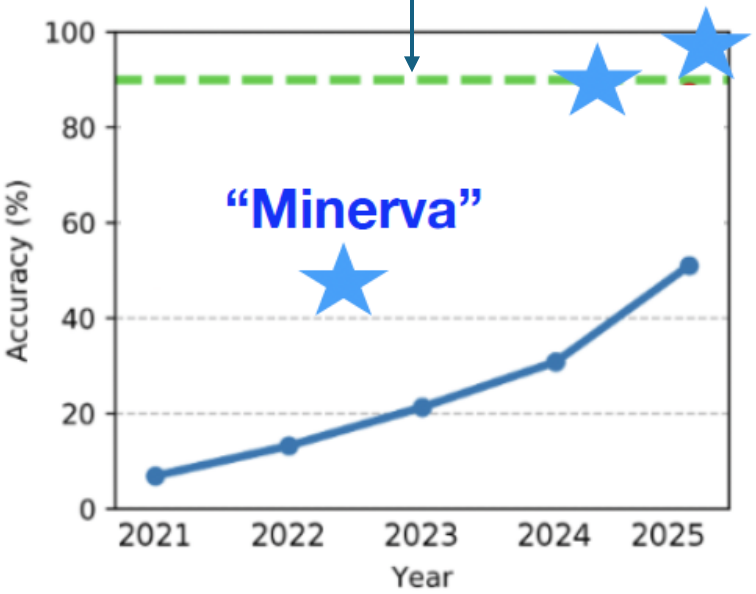
source: Vinay Ramasesh

Measuring Mathematical Problem Solving With the MATH Dataset

5 Mar 2021

Dan Hendrycks UC Berkeley, Collin Burns UC Berkeley, Saurav Kadavath UC Berkeley, Akul Arora UC Berkeley, Steven Basart UChicago
Eric Tang UC Berkeley, Dawn Song UC Berkeley, Jacob Steinhardt UC Berkeley

"We evaluated humans on MATH, and found that a computer science PhD student who does not especially like mathematics attained approximately 40% on MATH, while a three-time IMO gold medalist attained 90%, indicating that MATH can be challenging for humans"



Minerva = LLM trained in mathematical reasoning

Future of deep learning?

GPQA Diamond (graduate science questions):

The GPQA (Graduate-Level Google-Proof Q&A) Diamond dataset is a collection of challenging multiple-choice questions in biology, physics, and chemistry. Questions are written by domain experts (people with or pursuing PhDs in the relevant fields), and they are designed to be very difficult for non-experts to answer, even with unrestricted internet access.

It contains 198 questions, for which both domain expert annotators got the correct answers, but which the majority of non-domain experts answered incorrectly.

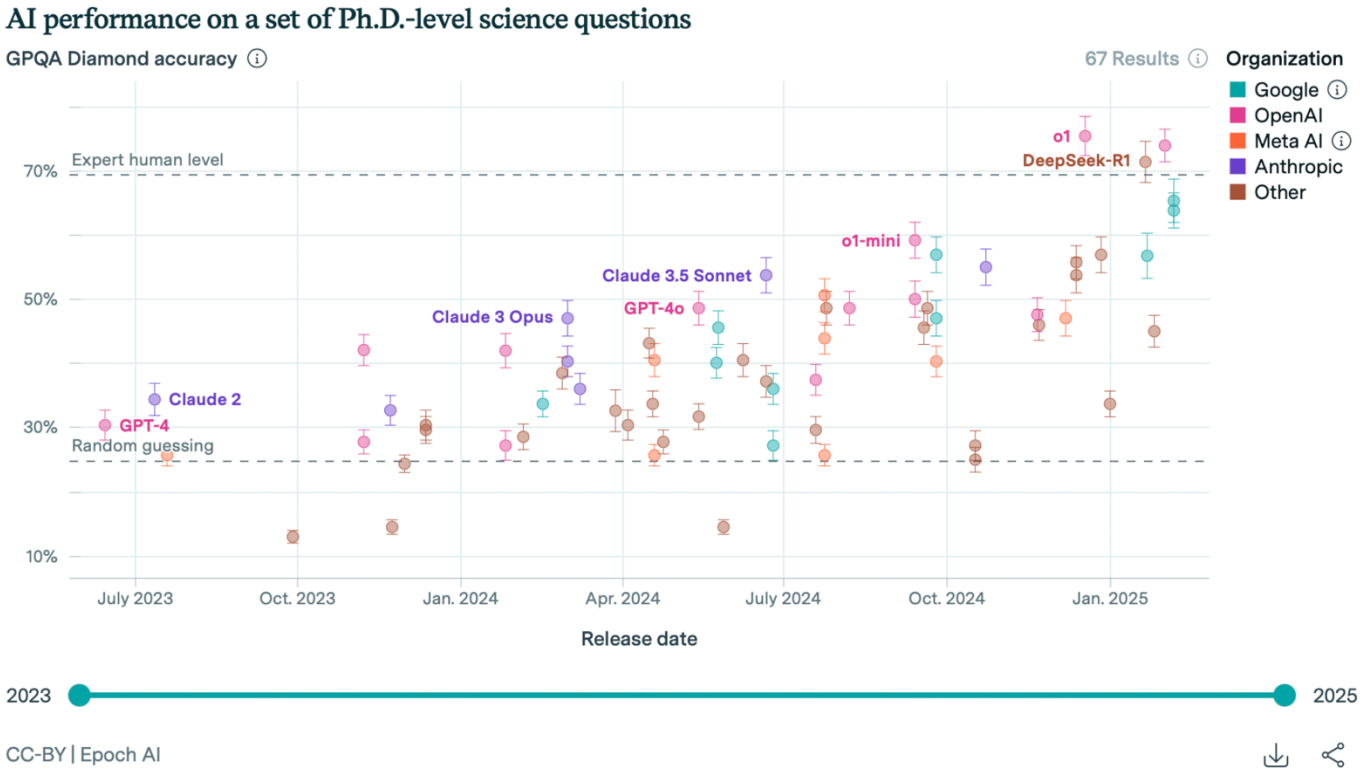
Since GPQA questions are multiple choice questions with four options, the random guessing baseline accuracy on GPQA Diamond is 25%

PhD-level experts… scored 69.7%.

Example question:

The universe is filled with the Cosmic Microwave Background. Consider the annihilation of high energy γ-rays with a photon from the CMB Radiation into electron-positron, i.e. $\gamma\gamma \rightarrow e^{+}e^{-}$. From what energy γ-rays would have their lifetimes in the universe limited by this process? Knowing that the average photon energy of the CMB is $10^{-3}eV$.

(A) $3.9 \times 10^{5} GeV$
(B) $9.5 \times 10^{4} GeV$
(C) $2.6 \times 10^{5} GeV$
(D) $1.8 \times 10^{5} GeV$



Future of deep learning?

2019 preschool

2020 elementary school

2022 high school

today? undergrad

? grad ?

professor

einstein → 10^6 einsteins

Source: Adam Brown (Google Deep Mind)

Analogy with (human) brain

Artificial neural network vs (human) brain
- differences:

- number of connections: human brain has 10^5 times more connections in total
- topology: connections are randomly distributed, artificial networks for layered feedforward structure
- human brain is much faster in generating outputs
- brain structure is much more stable and resilient to errors (artificial networks are susceptible to attacks)
- brains are being in training mode continuously

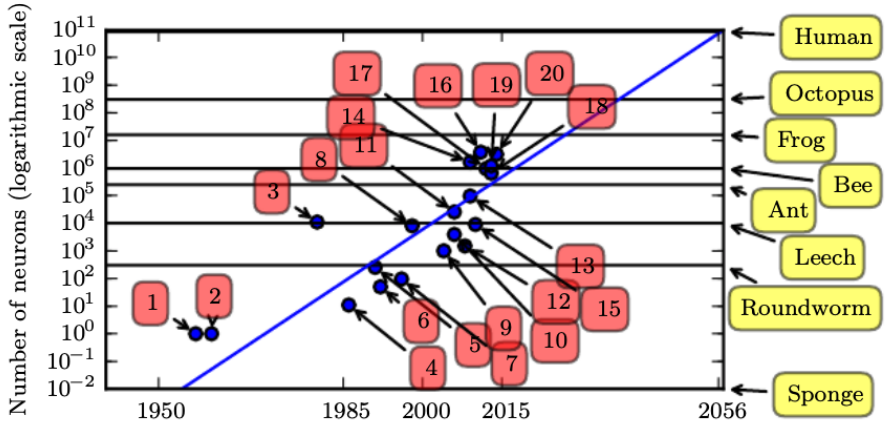


Figure 1.11: Increasing neural network size over time. Since the introduction of hidden units, artificial neural networks have doubled in size roughly every 2.4 years. Biological neural network sizes from Wikipedia (2015).

Image Source: https://timedotcom.files.wordpress.com/2014/05/brain.jpg?w=1100&quality=85

Brain with 1.6 x 10^{10} neurons

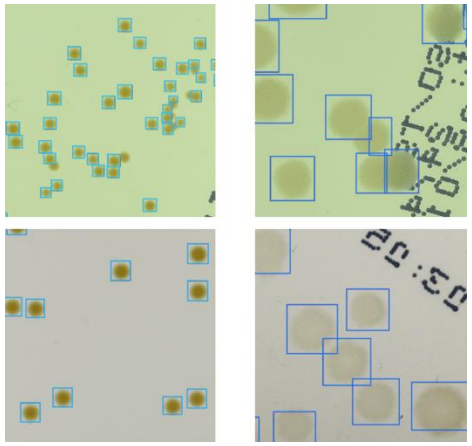
10^4 -10^5 connections per neuron

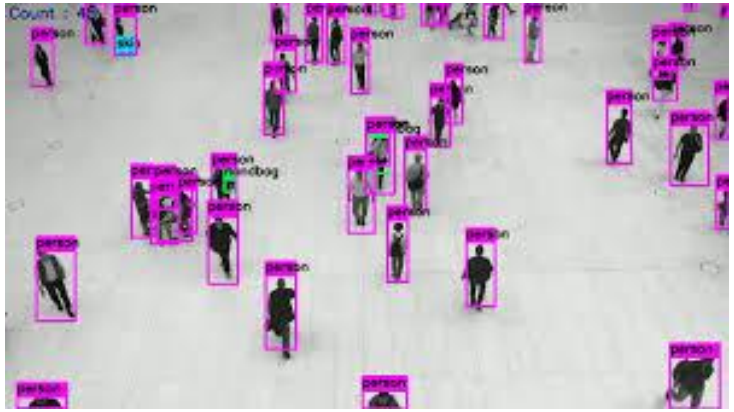
approx. 10^{15} connections in total

Big datasets are everywhere

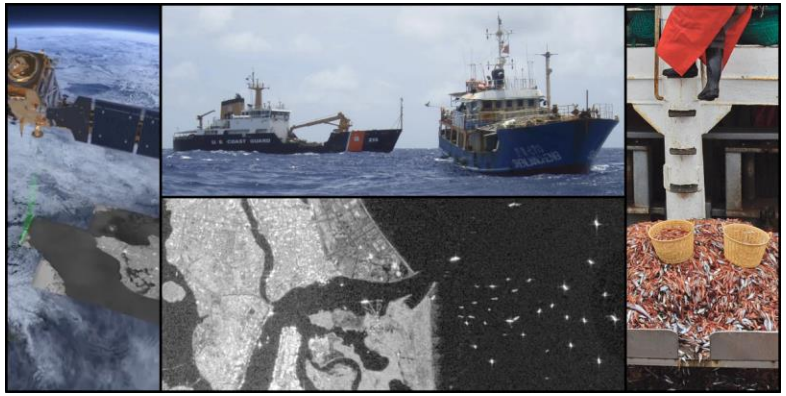
- Waste detection and classification

- Objects counting





- Catching illegal fishing



- Driving images segmentation

- There are companies specialized in labeling any data you want

Popular data sources

1. https://huggingface.co/
- popular web service with interesting datatsets and ready-to-use models (SOTA in many cases)
2. https://www.kaggle.com/
- machine-learning competitions supported by plenty of datasets,
- e.g. butterfly classification https://www.kaggle.com/datasets/phucthaiv02/butterfly-image-classification
3. https://paperswithcode.com/
- ML scientific papers supported by models + datasets
4. Classical datasets, https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research
- MNIST, Fashion-MNIST
- Iris
- CIPHAR
- ImageNet
- other somewhat older: https://github.com/jbrownlee/Datasets

Machine learning approaches

Supervised learning:

- data is organized in pairs (input, expected output)
- problems: classification, regression
- SVM, decision trees, neural networks
- gaussian discriminant analysis, naive Bayes

Unsupervised learning:

Reinforcement learning

"training examples"

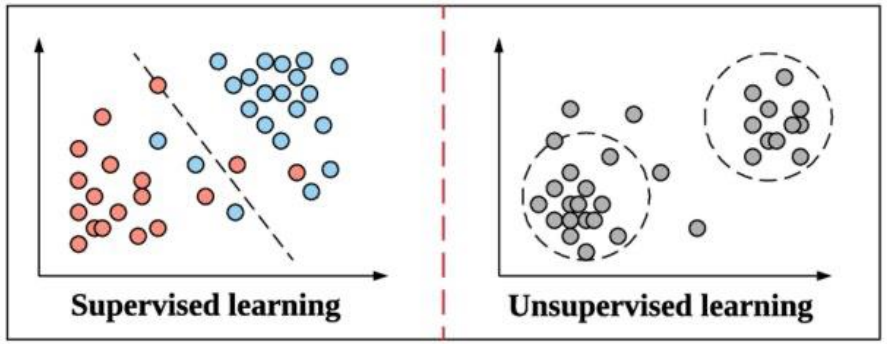
Training set: $\mathscr{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \ldots, n\}$,

Unknown function: $f(\mathbf{x}) = y$

Hypothesis: $h(\mathbf{x}) = \hat{y}$ sometimes t or o

Classification: $h : \mathbb{R}^m \to \mathscr{Y}, \quad \mathscr{Y} = \{1, \ldots, k\}$

Regression: $h : \mathbb{R}^m \to \mathbb{R}$



www.researchgate.net/project/software-defined-network-3

Learning – abstract definition

P. Grohs, G. Kutyniok, *Mathematical Aspects of Deep Learning*

Definition 1.1 (Learning – informal)**.** Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be measurable spaces. In a learning task, one is given data in $\mathcal{Z}$ and a loss function $\mathcal{L}\colon \mathcal{M}(\mathcal{X},\mathcal{Y}) \times \mathcal{Z} \to \mathbb{R}$. The goal is to choose a hypothesis set $\mathcal{F} \subset \mathcal{M}(\mathcal{X},\mathcal{Y})$ and to construct a learning algorithm, i.e., a mapping

$$\mathcal{A}\colon \bigcup_{m\in\mathbb{N}} \mathcal{Z}^m \to \mathcal{F},$$

that uses training data $s = (z^{(i)})_{i=1}^m \in \mathcal{Z}^m$ to find a model $f_s = \mathcal{A}(s) \in \mathcal{F}$ that performs well on the training data s and also generalizes to unseen data $z \in \mathcal{Z}$. Here, performance is measured via the loss function $\mathcal{L}$ and the corresponding loss $\mathcal{L}(f_s, z)$ and, informally speaking, generalization means that the out-of-sample performance of f_s at z behaves similarly to the in-sample performance on s.

The model has to be parametrized:

$\mathcal{M}(\mathcal{X},\mathcal{Y})$ to be the set of measurable functions from $\mathcal{X}$ to $\mathcal{Y}$.

Let $a = (N, \varrho)$ be a NN architecture

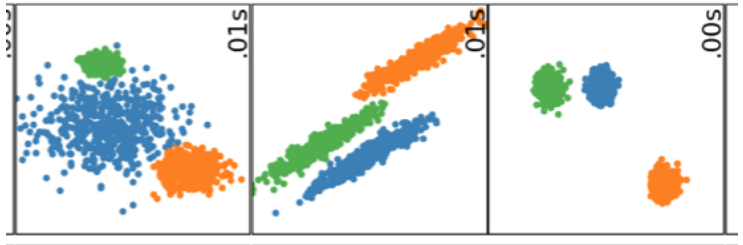
Definition 1.2 (Prediction task)**.** In a prediction task, we have that $\mathcal{Z} := \mathcal{X}\times\mathcal{Y}$, i.e., we are given training data $s = ((x^{(i)}, y^{(i)}))_{i=1}^m$ that consist of input features $x^{(i)} \in \mathcal{X}$ and corresponding labels $y^{(i)} \in \mathcal{Y}$. For one-dimensional regression tasks with

$$\mathcal{F}_a = \left\{\Phi_a(\cdot,\theta)\colon \theta \in \mathbb{R}^{P(N)}\right\}$$

Machine learning approaches

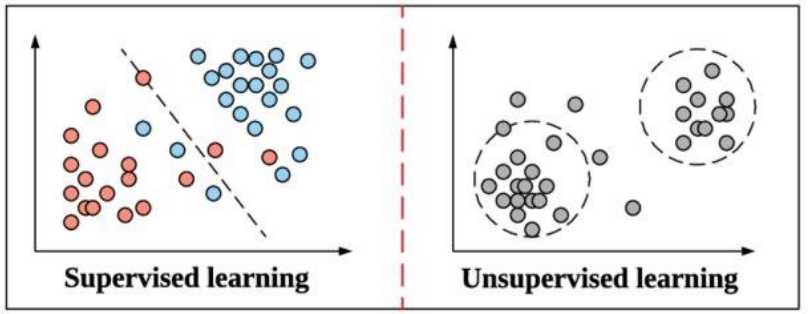
Unsupervised learning:

- in opposite to supervised learning, data is not labeled
- problems: clustering, dimensionality reduction
- PCA, LDA, k-means clustering
- anomaly detection



https://scikit-learn.org

Reinforcement learning



Example: Supervised vs Unsupervised

- Having N photos of different animals
- Supervised task (requires labeled data)

> Train an algorithm to recognise given species on a photo.
>
> Output: There is X on a photo.

- Unsupervised task

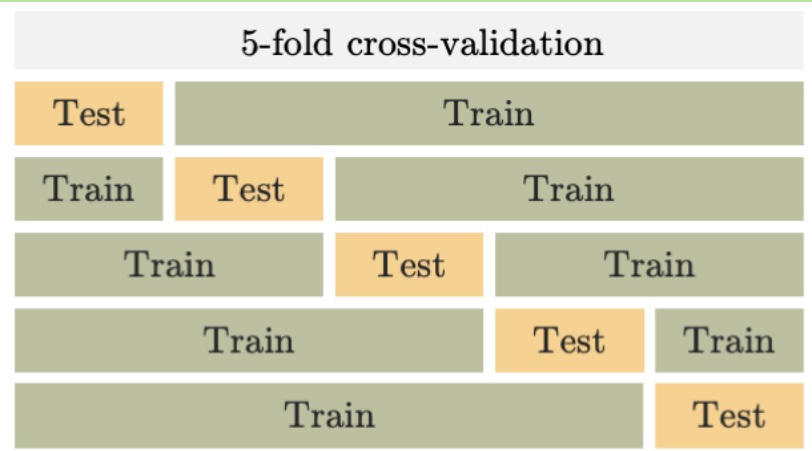
> Train an algorithm to group animals with similar features.
>
> Output: No idea what it is, but it looks similar to these animals.

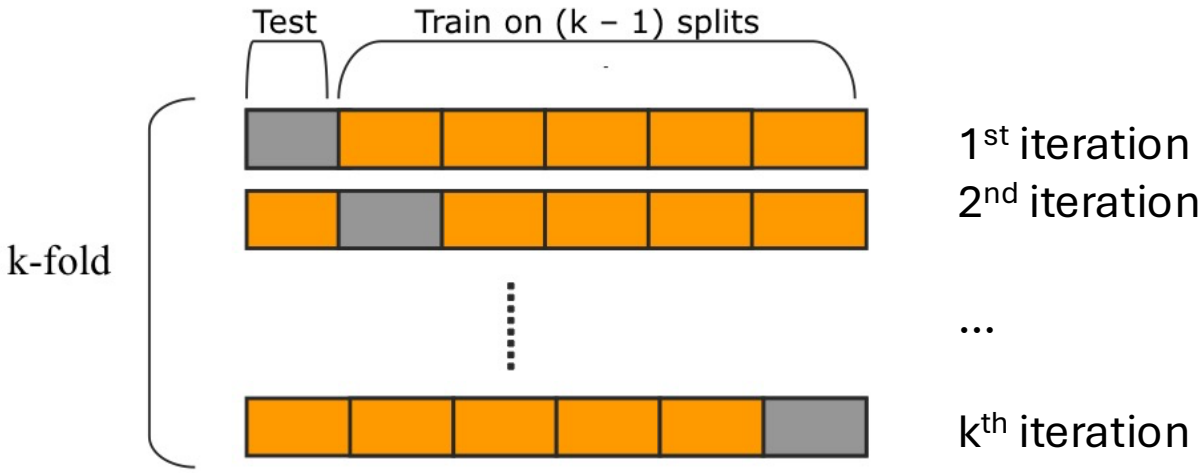
Learning scheme

How to perform learning and evaluate model performance

- divide dataset into **train** and **test** subsets
  - test should be evaluated on data *unseen* during training

- *k*-fold cross-validation





1st iteration
2nd iteration

...

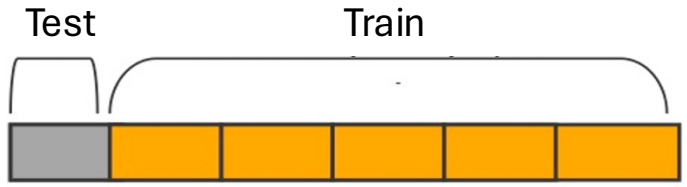
kth iteration

https://aszokalski.github.io/AI/Cross-Validation.html

- in practice, while testing big models,
  we limit ourselves to only one *k*-fold iteration

Learning scheme

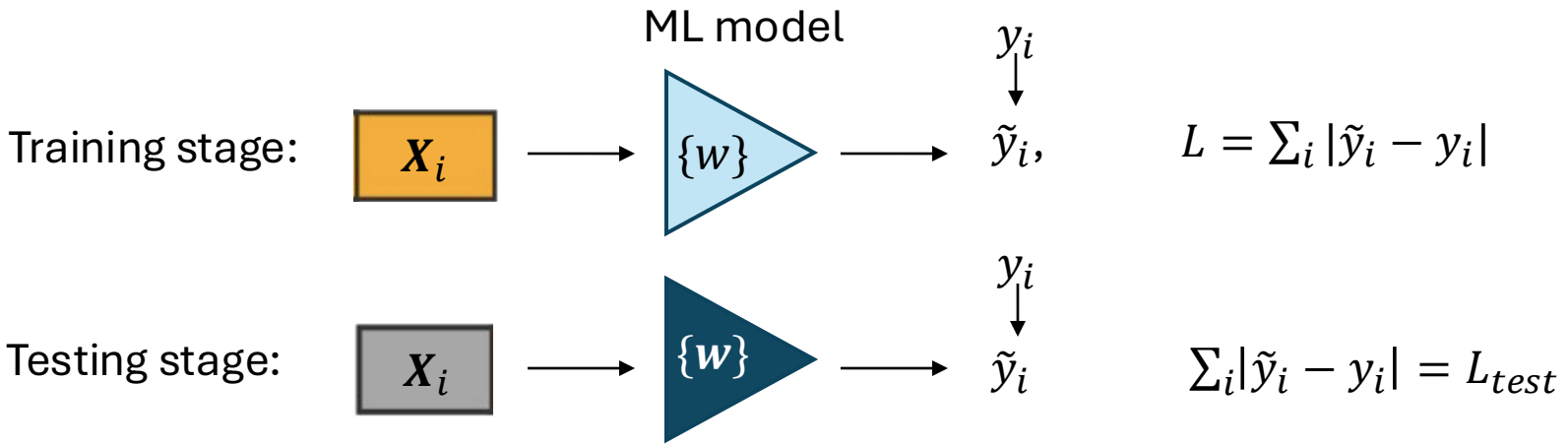
Supervised learning



Where each data sample is labeled:

$$\{< \boldsymbol{X}_i, y_i >, i = 1 \ldots, N\}$$

- Data organized in pairs.
- The training aims to minimize L by optimizing the model weights.
- But the real goal is to have L_{test} small.
- Testing simulate evaluation of the model in real life (inference).



ML is not just an optimization!

Bias-variance tradeoff

The ability to perform well on previously unobserved inputs is called **generalization**:

- during learning we optimize model to have lowest **training error**,
- we *also* want the generalization error, also called the **test error** to be low as well (this separates ML from optimization).

Training and testing error correspond to the two central challenges in machine learning: (model) **underfitting** and **overfitting.**

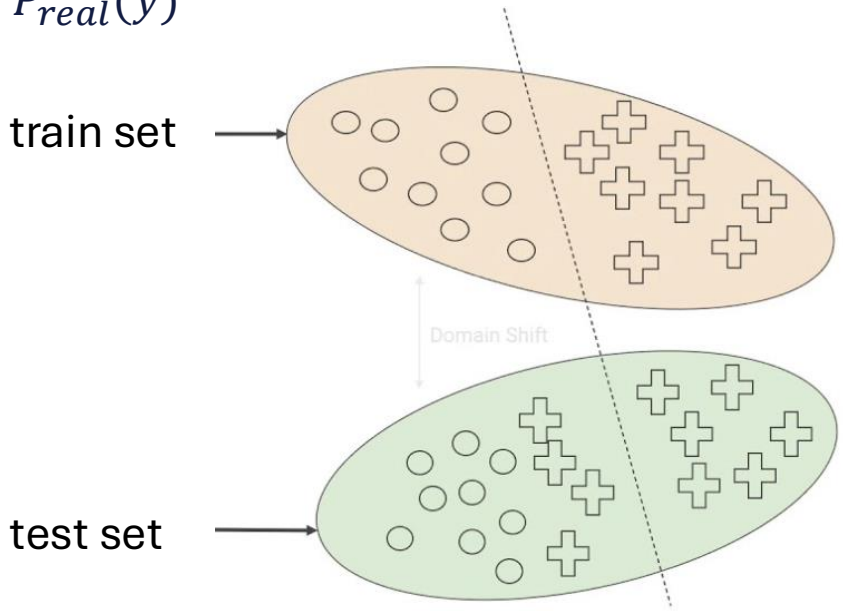
strongly recommended paper: https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

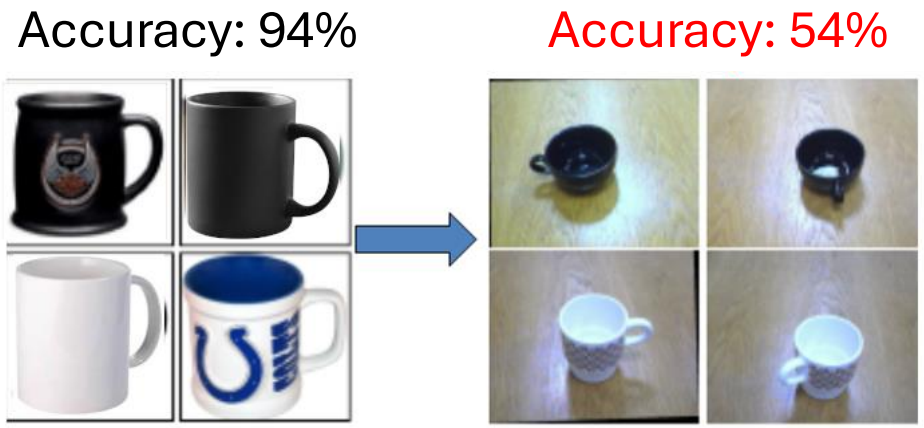
Domain and covariate shift

Other fundamental problems:

domain shift – related to *bias* in statistics, data distributions shifts between the training domain and live domain.

$P_{train}(y) \neq P_{real}(y)$





covariate shift – ...

Domain and covariate shift

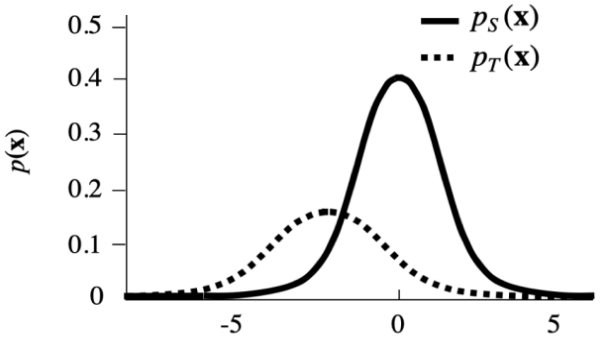
Other fundamental problems:

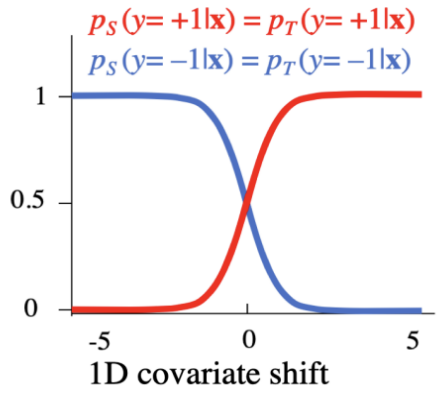
domain shift

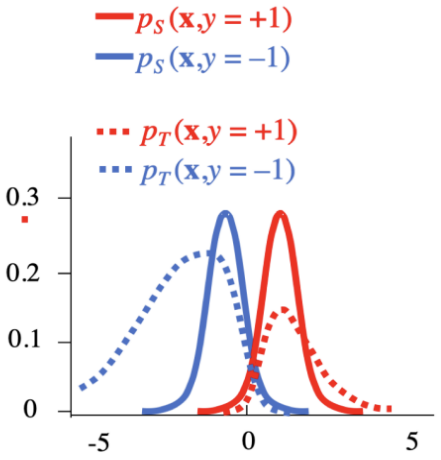
covariate shift – *specific* to supervised learning,
the input $\boldsymbol{X}_i$ distribution may change, the output (labels) y_i distribution remain the same:

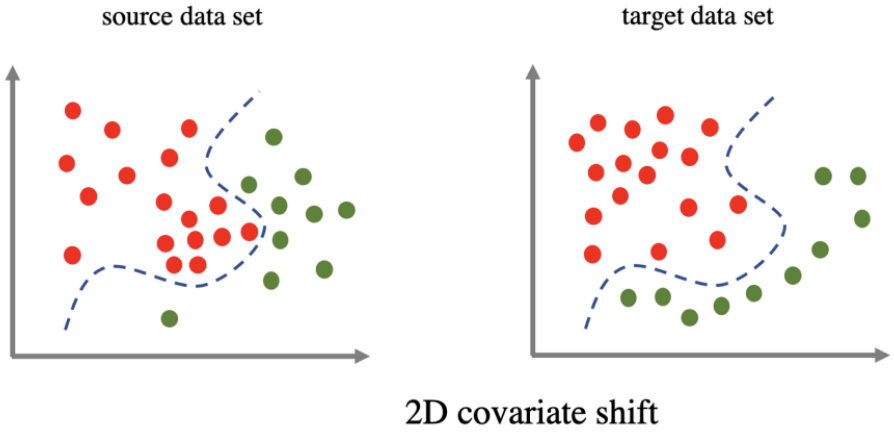
$P_{train}(y|\boldsymbol{X}) = P_{real}(y|\boldsymbol{X})$, while $P_{train}(\boldsymbol{X}) \neq P_{real}(\boldsymbol{X})$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$









https://arxiv.org/pdf/2001.09994.pdf

Domain and covariate shift

What type of shift do we have here?

Accuracy: 94%

Accuracy: 54%

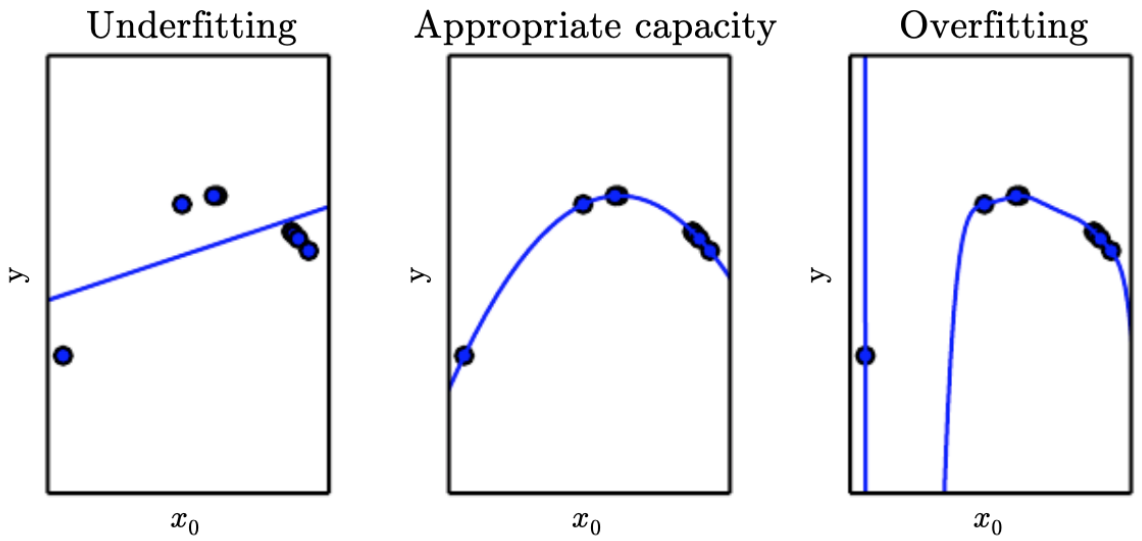
Bias-variance tradeoff

Underfitting: model is not able to obtain a sufficiently low error on the training set

Overfitting occurs when the distance (gap) between the training and test error is too large

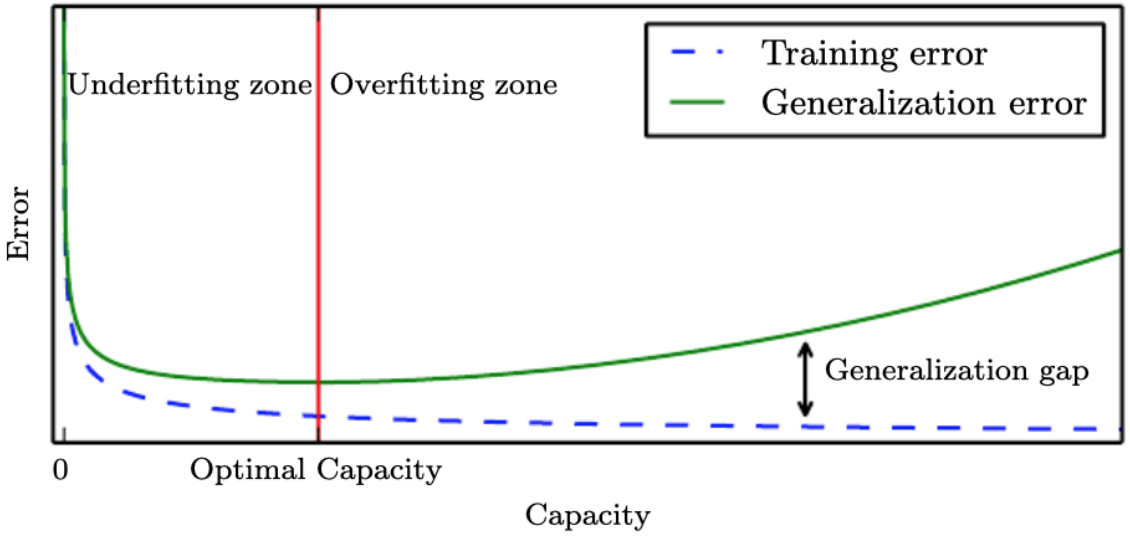
We can control whether a model is more likely to overfit or underfit by altering its **capacity**

Here capacity means *degree* of the fitted polynomial:



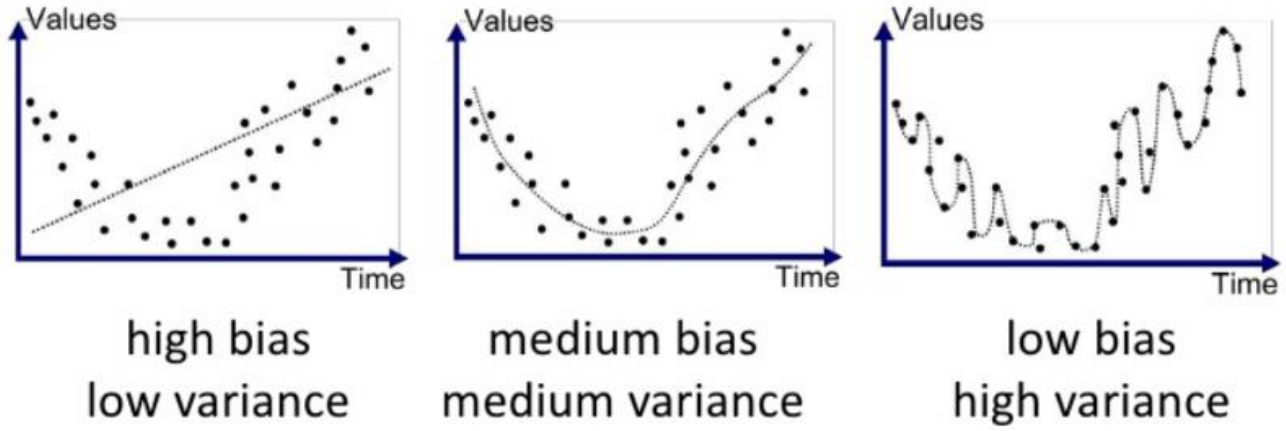
https://www.deeplearningbook.org/contents/ml.html

Bias-variance tradeoff



- To simple to capture relations in data

- To much attention towards training data (and noise within)



$$\mathrm{MSE} = \mathbb{E}\left[\left(\hat{f} - y\right)^2\right] = \mathrm{Bias}\left(\hat{f}\right)^2 + \mathrm{Var}\left(\hat{f}\right),$$
$$\text{with } \mathrm{Bias}\left(\hat{f}\right) = \mathbb{E}\left(\hat{f}\right) - y$$

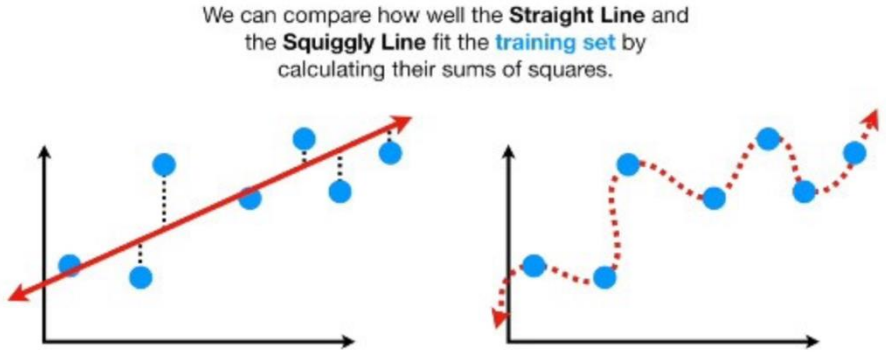
$f(x) = y$ – ground truth,
$\hat{f} = \hat{f}(x, D)$ – model trained on D (bootstrap)

averaged over "statistics" of many models

https://www.deeplearningbook.org/contents/ml.html
https://towardsdatascience.com/tradeoffs-how-to-aim-for-the-sweet-spot-c20b40d5e6b6

Bias-variance tradeoff

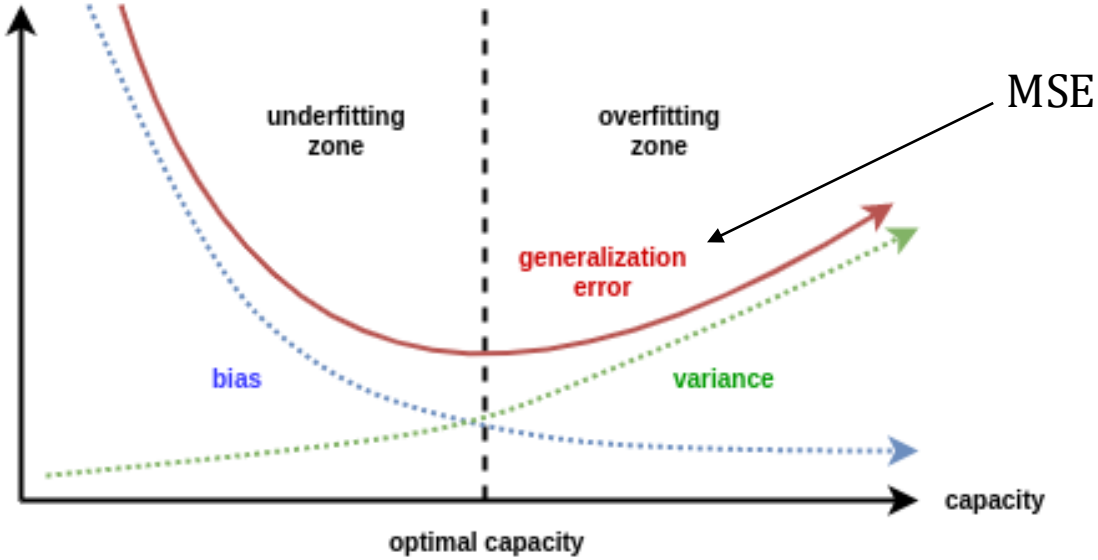
- High bias can cause an algorithm to miss the relevant relations in data (underfitting).
- High variance can cause an algorithm to model the random noise in the training data (overfitting).

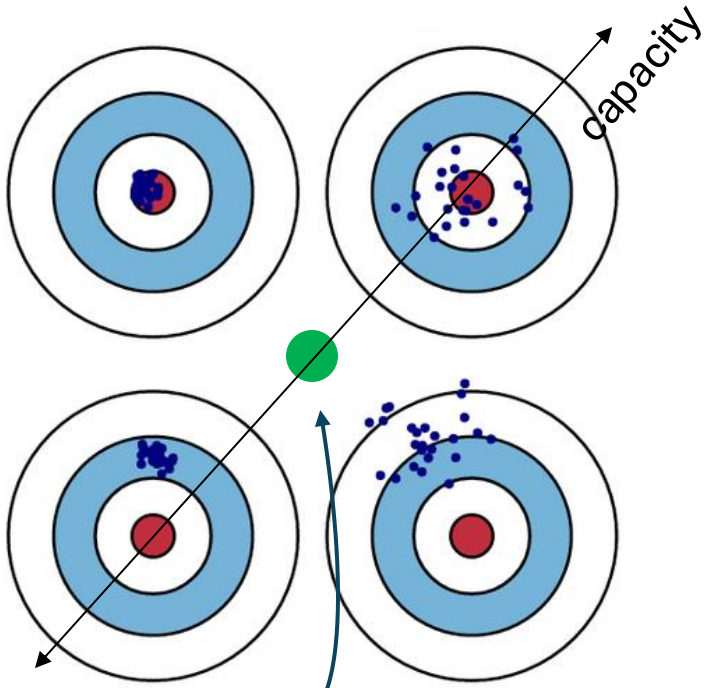


https://www.deeplearningbook.org/contents/ml.html
https://towardsdatascience.com/tradeoffs-how-to-aim-for-the-sweet-spot-c20b40d5e6b6

Bias-variance tradeoff

- we aim to minimize generalization error
- to low bias will result into high variance and vice versa:
  bias-variance tradeoff



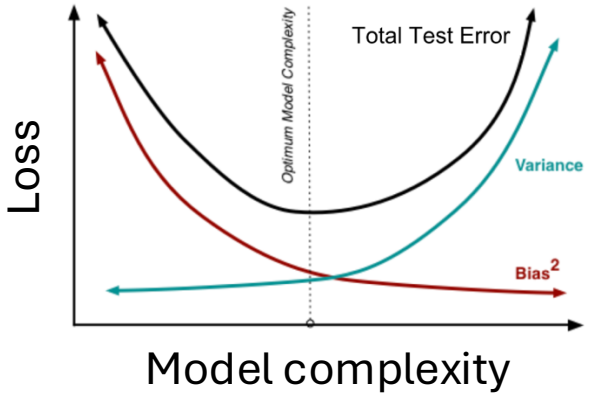


optimization
“sweet spot”

https://towardsdatascience.com/tradeoffs-how-to-aim-for-the-sweet-spot-c20b40d5e6b6

Bias-variance in deep learning

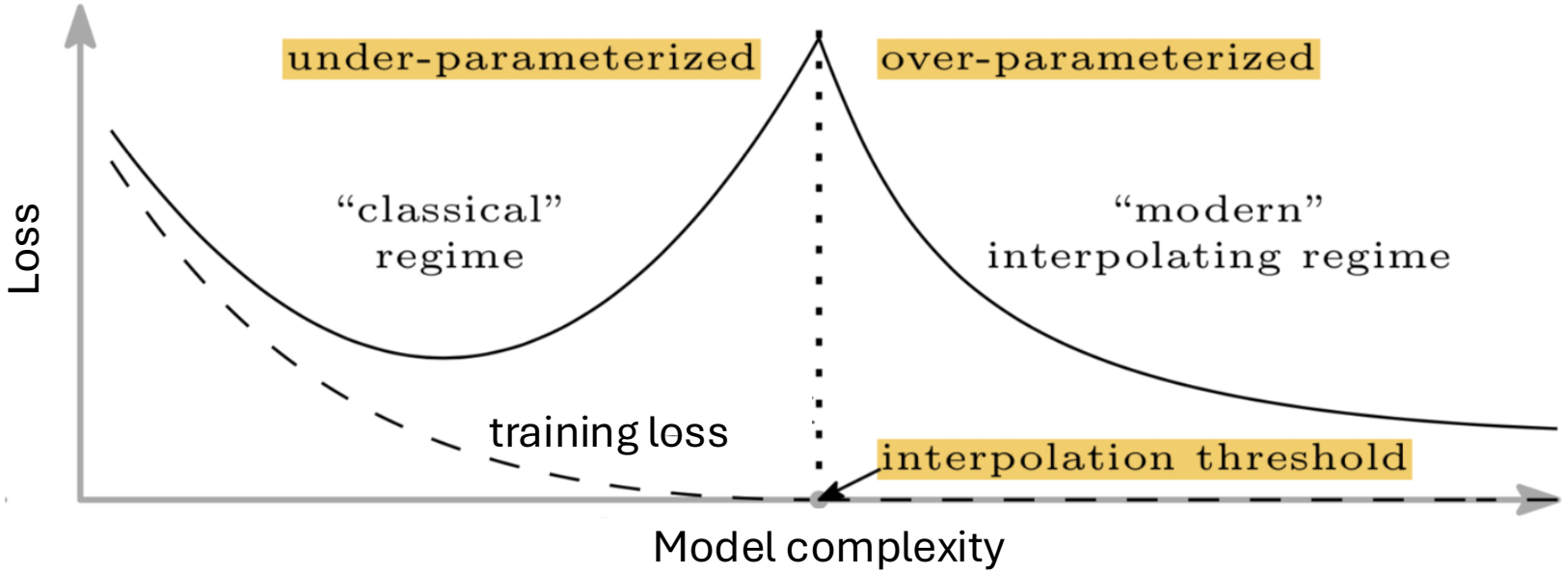
classical ML



https://lilianweng.github.io/posts/2019-03-14-overfit

https://arxiv.org/pdf/1812.11118.pdf

deep learning



Overparametrized model:

- The number of parameters is not a good measure of inductive bias, defined as the set of assumptions of a learning algorithm used to predict for unknown samples.
- Equipped with a larger model, we might be able to discover larger function classes and further find interpolating functions that have smaller norm and are thus "simpler".

Bias-variance in deep learning

A. Tanaka, A. Tomiya, K. Hashimoto, *Deep Learning and Physics*

VC-dimension generalization bounds (errors normalized to 1)

Classical ML, $d_{VC} \ll m$

$$\text{(Generalization error)} \leq \text{(Training error)} + \mathcal{O}\left(\sqrt{\frac{\log m/d_{VC}}{m/d_{VC}-1}}\right)$$

- it is worth having m/d_{VC} large

m = size of dataset
d_{VC} = **Vapnik-Chervonenkis** dimension**:**

$$d_{\mathrm{VC}}(\mathcal{F}_N) = \sup\{\#\text{: numer of classification patterns captured by model } \mathcal{F}_N = 2^{\mathrm{m}}\}$$

d_{VC} describes complexity (capacity) of a model $\mathcal{F}_J$
d_{VC} depends on the number of model parameters N

Deep Learning, $d_{VC} \sim N \log N \sim m$

$$\text{(Generalization error)} \leq \text{(Training error)} + \mathcal{O}_{DL} \leq \text{(Training error)} + \mathcal{O}\left(\sqrt{\frac{\log m/d}{m/d-1}}\right)$$

- different bounds: it is worth having large model

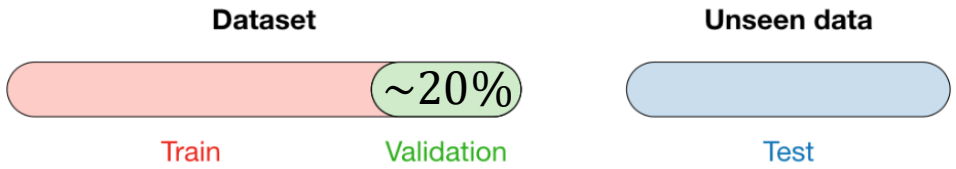
Avoiding overfitting: validation set

How to avoid overfitting

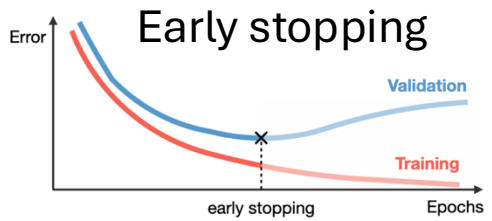
- The model should chosen/tuned (*hyperparameters*) to be as simple as possible but not simpler (Occam's razor), i.e., with appropriate capacity.

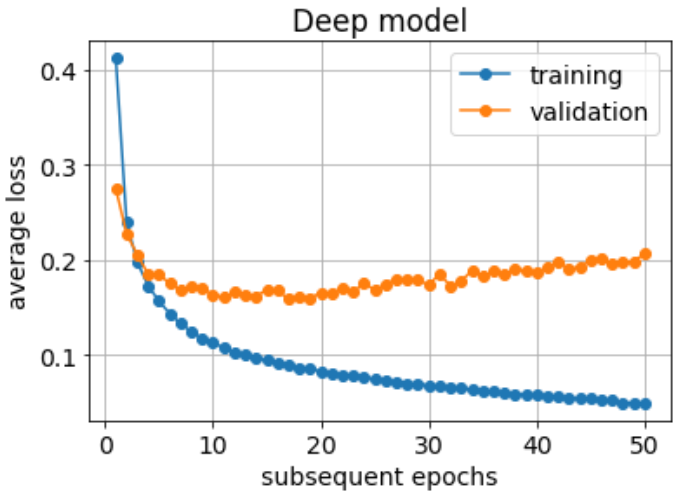
Hyperparameters are all the parameters that should be set up before learning starts.

We need another subset (**validation** set) to estimate the generalization error during training, allowing for the model hyperparameters to be updated accordingly.



❐ **Early stopping** — This regularization technique stops the training process as soon as the validation loss reaches a plateau or starts to increase.





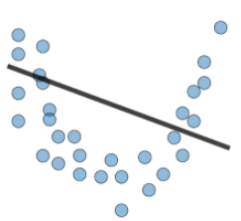
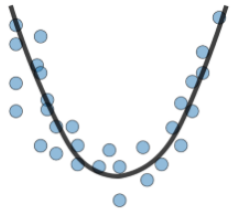
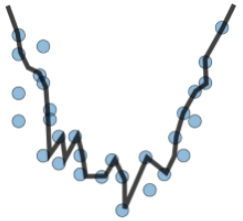
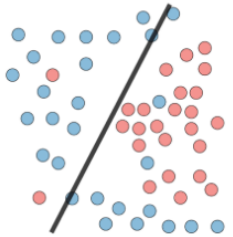
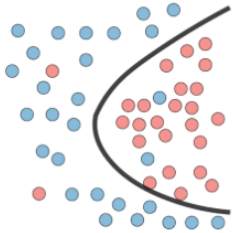
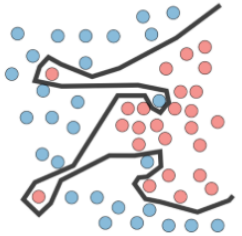
training stopping moment is one of the hyperparameters to be tuned

https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks

Avoiding overfitting

How to avoid overfitting

- Do not train to long!
- **Regularization**: adding weight to loss function to prefer one types of solutions over other (e.g. with lower polynomial degree).
- **Augmentation:** adding slightly modified copies of already existing data or synthetize new samples from existing one.
- Increase dataset size (but new data should be coherent)
  …a dumb algorithm with lots of data beats a clever algorithm with a modest amount of data…

	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks

No free lunch theorem

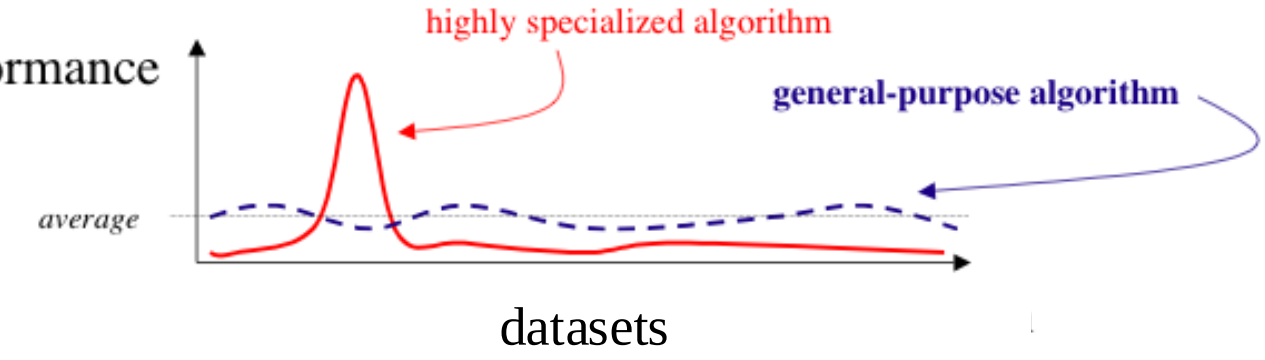
Suppose that we have all possible domains (datasets) for a given problem (e.g. image classification) and need to compare two methods A and B:

No free lunch theorem states that:

- performance for algorithms A and B averaged over the all possible datasets is the same,
- this holds true even when one of the algorithms is just random guessing.

There is no such thing as a single, universally-best machine learning algorithm, and there are no *a priori* reasons to favor one algorithm over all others.

- we can't get good machine learning for *free*.
- we must use knowledge about out data and the context, to select an appropriate machine learning model.



medium.com/@LeonFedden/the-no-free-lunch-theorem-62ae2c3ed10c

No free lunch theorem

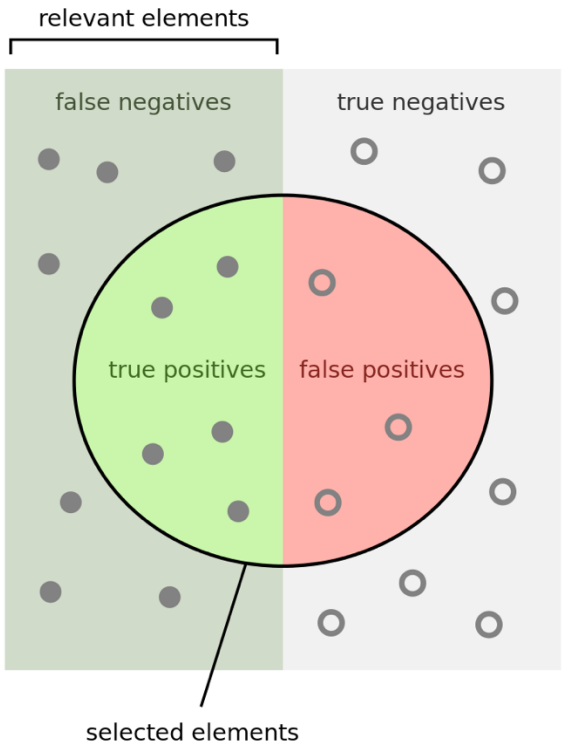
A *family* of **theorems**:

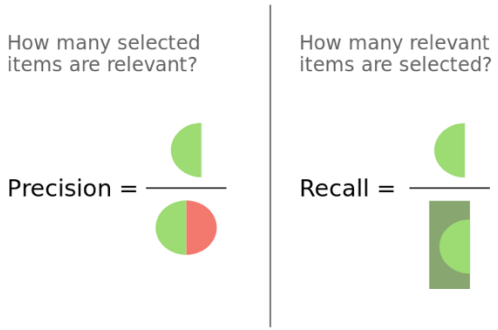
- Wolpert, D H., 1996. The lack of *a priori* distinctions between learning algorithms.
  Neural Computation, **8**(7), 1341–1390.
- Wolpert, D H., 2001. The supervised learning no-free-lunch theorems.
  In: *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*.
- Wolpert, D H., and W G. MacReady, 1995. No free lunch theorems for search.
  Technical Report SFI-TR-95-02-010. Sante Fe, NM, USA: Santa Fe Institute.
- Wolpert, D H., and W G. MacReady, 1997. No free lunch theorems for optimization.
  IEEE Transactions on Evolutionary Computation, **1**(1), 67–82.
- Wolpert, D H., and W G. MacReady, 2005. Coevolutionary free lunches.
  IEEE Transactions on Evolutionary Computation, **9**(6), 721–735

- good review paper:
  arxiv.org/pdf/2007.10928.pdf

- also visit: http://www.no-free-lunch.org

Precision-recall tradeoff





		Predicted class	
		+	-
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	-	**FP** False Positives Type I error	**TN** True Negatives

classification and detection metrics:

- information stored in *confusion matrix* are used to calculate:
- accuracy,
- precision, recall,
- F1 score
- …

Metric	Formula	Interpretation
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$	Overall performance of model
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	Coverage of actual positive sample
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$	Coverage of actual negative sample
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$	Hybrid metric useful for unbalanced classes

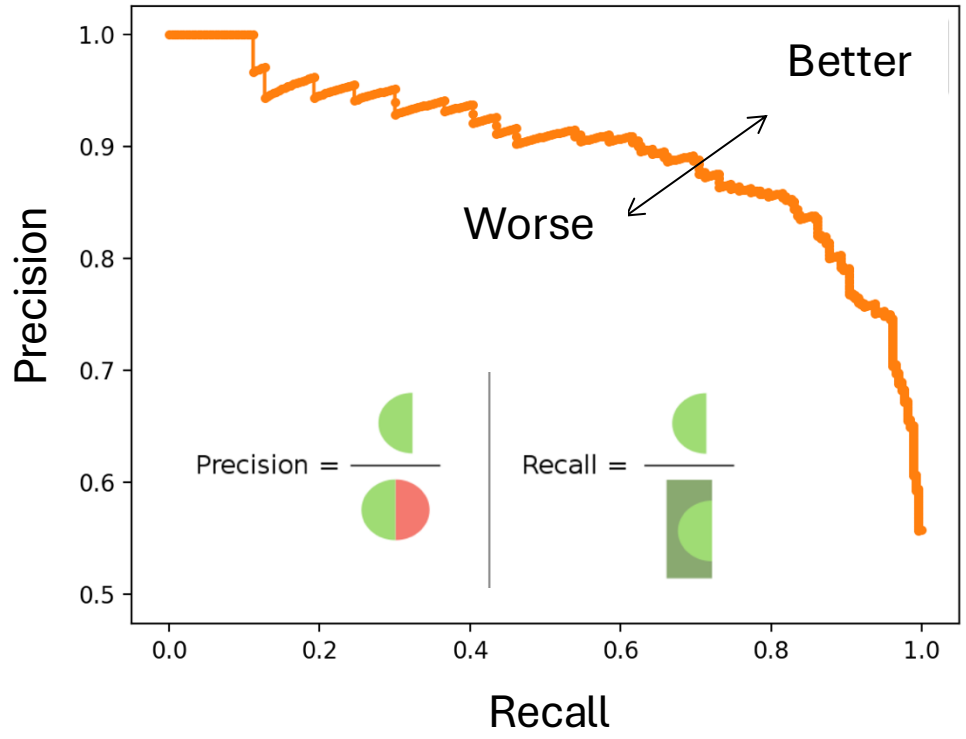
https://en.wikipedia.org/wiki/Precision_and_recall

https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks

Precision-recall tradeoff

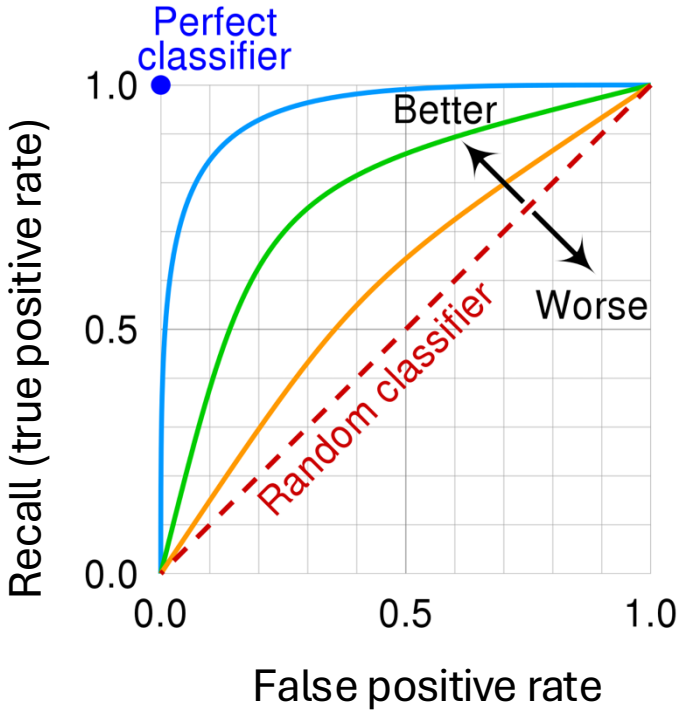
precision-recall tradeoff:

- if you increase precision (by model tuning),
  it will reduce recall, and vice versa.



We select classifier to have highest *area* under the precision-recall curve.

A similar concept: **ROC**
(receiver operating characteristic)



$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

en.wikipedia.org/wiki/Receiver_operating_characteristic