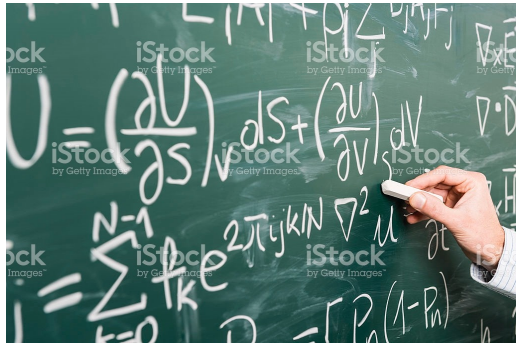

CIFAR
Université de Montréal
Mila
Introduction à
l'apprentissage profond
Yoshua Bengio
École Langlands du CRM
23 août 2021

Lecture 1

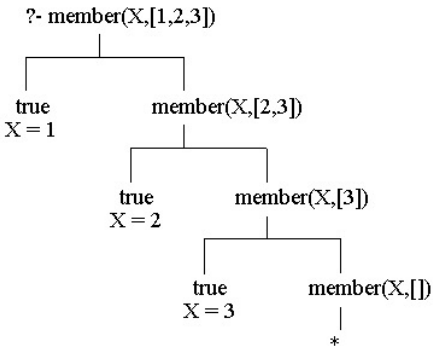
Neural Networks & AI: Underlying Assumption

- There are principles giving rise to intelligence (machine, human or animal) via learning, simple enough that they can be described compactly, similarly to the laws of physics, i.e., our intelligence is not just the result of a huge bag of tricks and pieces of knowledge, but of general mechanisms to acquire knowledge.



Mila

The Machine Learning approach to AI



- **Classical AI, rule-based, symbolic**
  - knowledge is provided by humans
    - but intuitive knowledge (e.g. much of common sense) not communicable
  - machines only do inference
  - no strong learning, adaptation
  - insufficient handling of uncertainty
  - not grounded in low-level perception and action
- **Machine learning tries to fix these problems**
  - succeeded to a great extent
  - higher-level (conscious) cognition not achieved yet

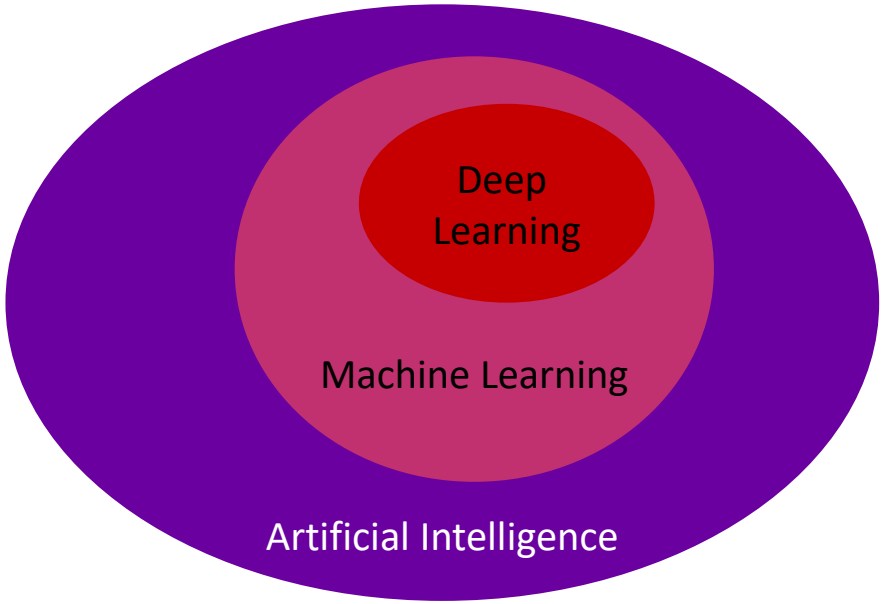
Mila

Intelligence Needs Knowledge

- Learning:

  powerful way to transfer knowledge to intelligent agents

  - Failure of classical AI: a lot of knowledge is **intuitive**
  - Solution: get knowledge from data & experience

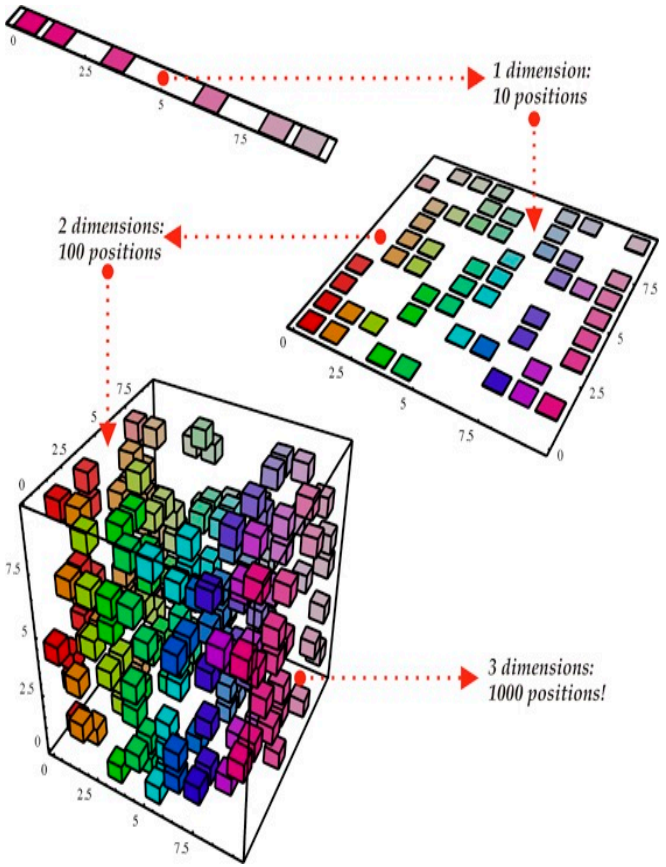


Mila

ML 101. What We Are Fighting Against: The Curse of Dimensionality

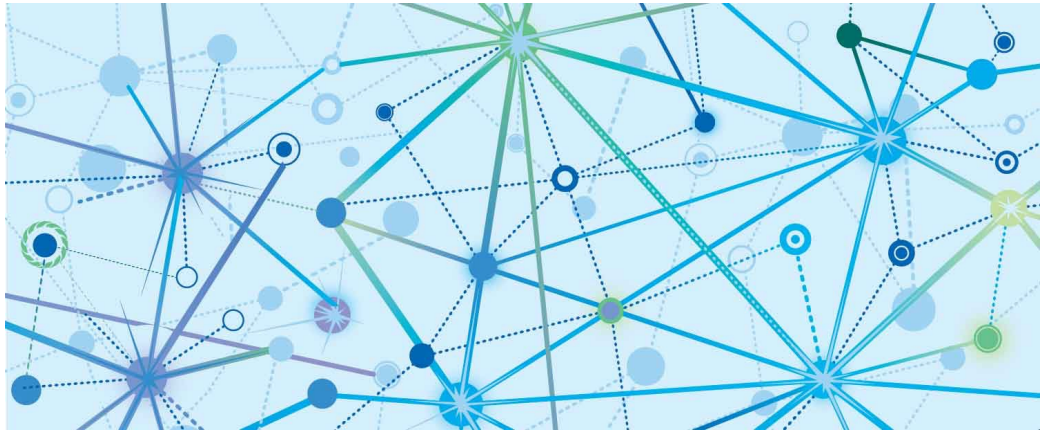
To generalize
locally, need representative examples for all relevant variations!

Classical solution:
hope for a smooth enough target function, or make it smooth by handcrafting good features / kernel



Mila

The Neural Net Approach to AI

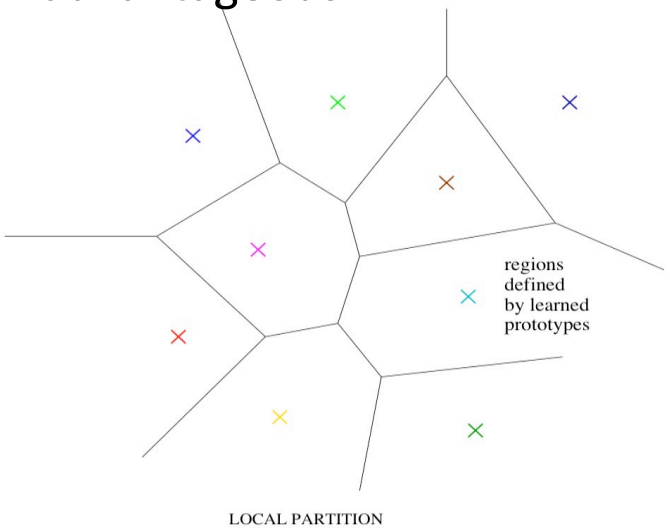


- **Brain-inspired**
- Synergy of a large number of simple adaptive computational units
- Focus on **distributed representations**
  - **E.g. word representations** (Bengio et al NIPS'2000)
- View intelligence as arising of combining
  - an objective or reward function
  - an approximate optimizer (learning rule)
  - an initial architecture / parametrization
- End-to-end learning (all the pieces of the puzzle adapt to help each other)

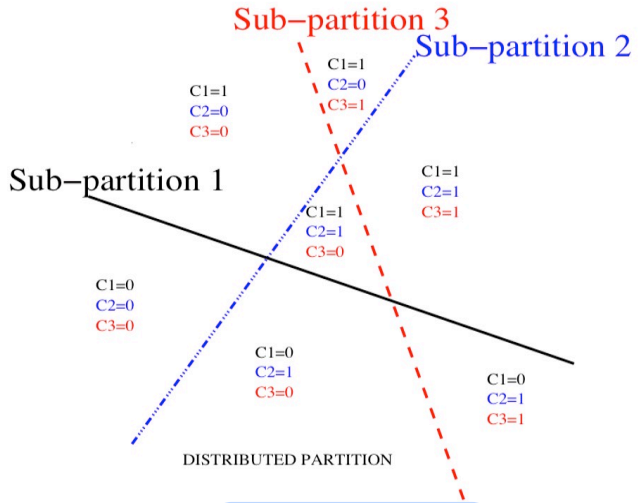
Mila

Distributed Representations: The Power of Compositionality

- Distributed (possibly sparse) representations, learned from data, can capture the **meaning** of the data and state
- Parallel composition of features: can be exponentially advantageous



Not Distributed

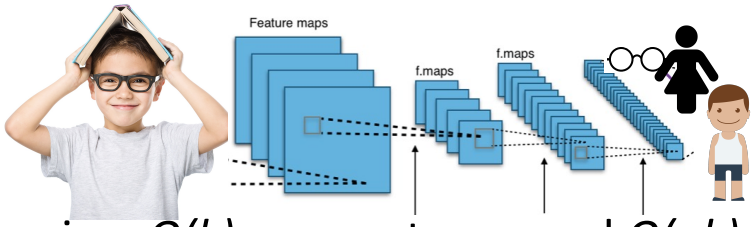


Distributed

Mila

Each feature can be discovered without the need for seeing the exponentially large number of configurations of the other features

- Consider a network whose hidden units discover the following features:
  - Person wears glasses
  - Person is female
  - Person is a child
  - Etc.

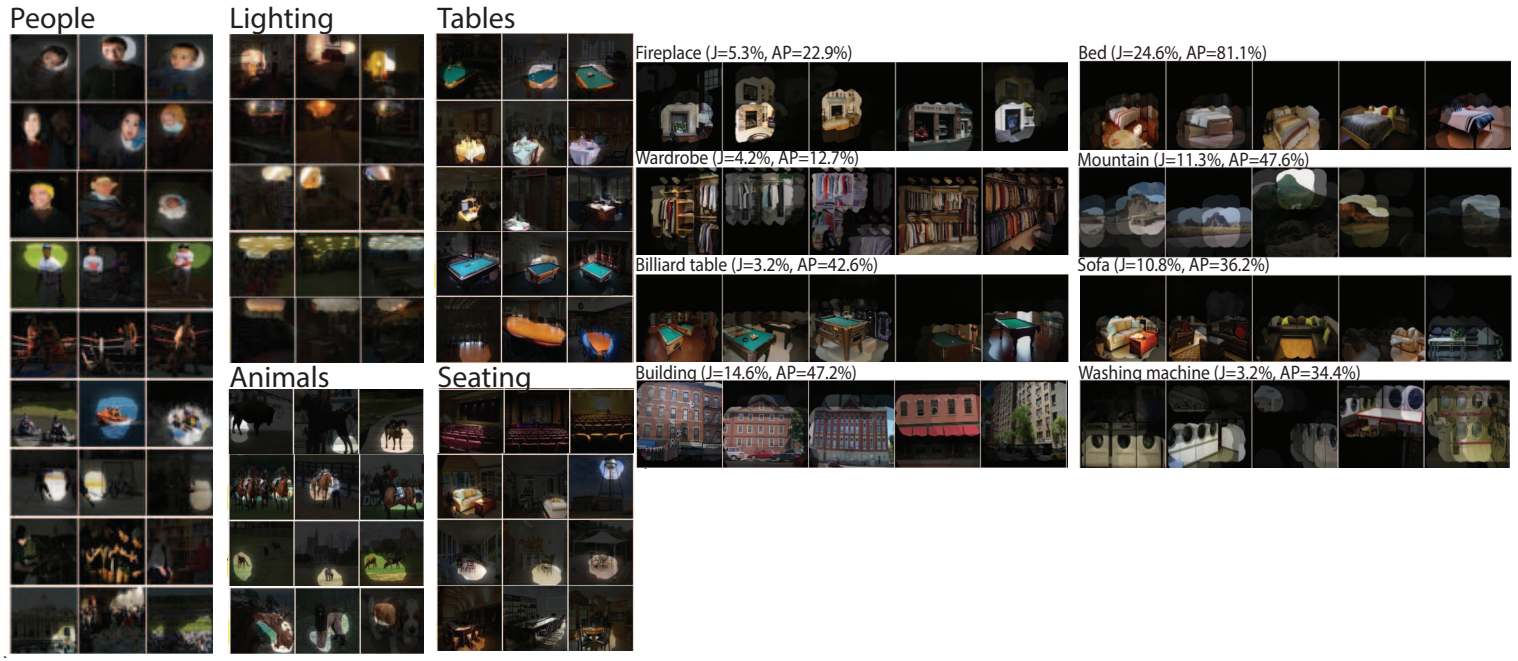


- If each of *n* feature requires *O(k)* parameters, need *O(nk)* examples
- Parallel composition of features: can be exponentially advantageous
- Non-distributed non-parametric methods would require $O(n^d)$ examples

Mila

Hidden Units Can Discover Semantically Meaningful Concepts

- *Zhou et al & Torralba, arXiv1412.6856 ,* ICLR 2015
- *Network trained to recognize places, not objects*

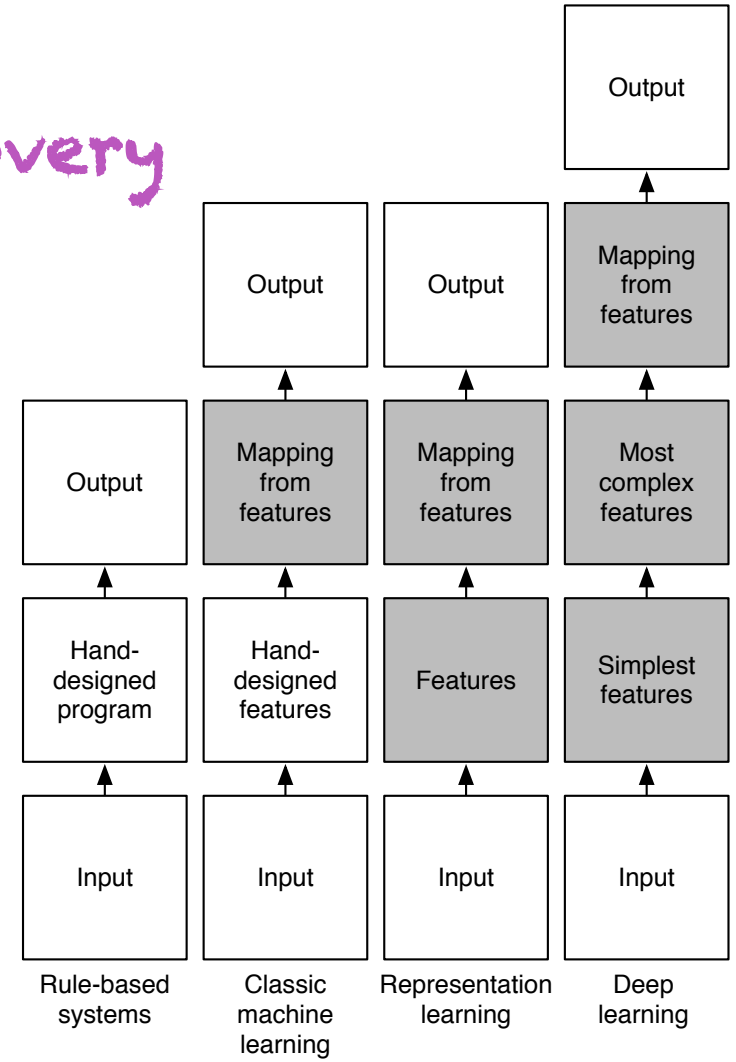


Mila

Deep Learning: Learning an Internal Representation

- Unlike other ML methods with either
  - no intermediate representation (linear)
  - or fixed (generally very high-dimensional) intermediate representations (SVMs, kernel machines)
- What is a good representation? Makes other tasks easier.

Automating Feature Discovery

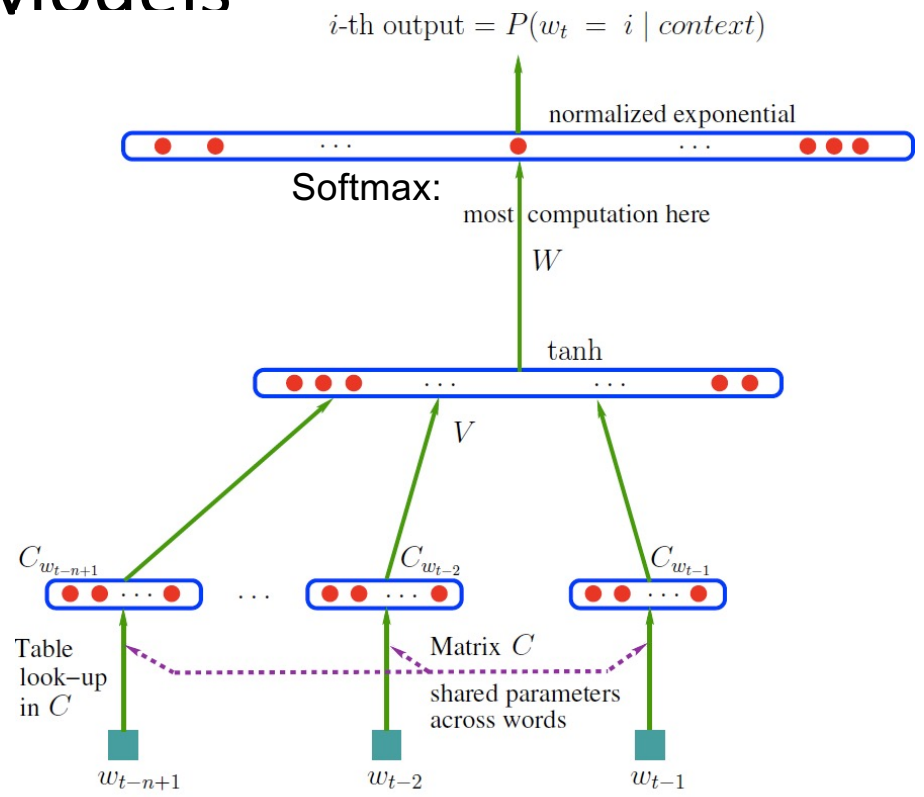


Mila

Neural Language Models



- *Bengio et al NIPS'2000 and JMLR 2003 "A Neural Probabilistic Language Model"*
  - Each word represented by a distributed continuous-valued code vector = embedding
  - Generalizes to sequences of words that are semantically similar to training sequences



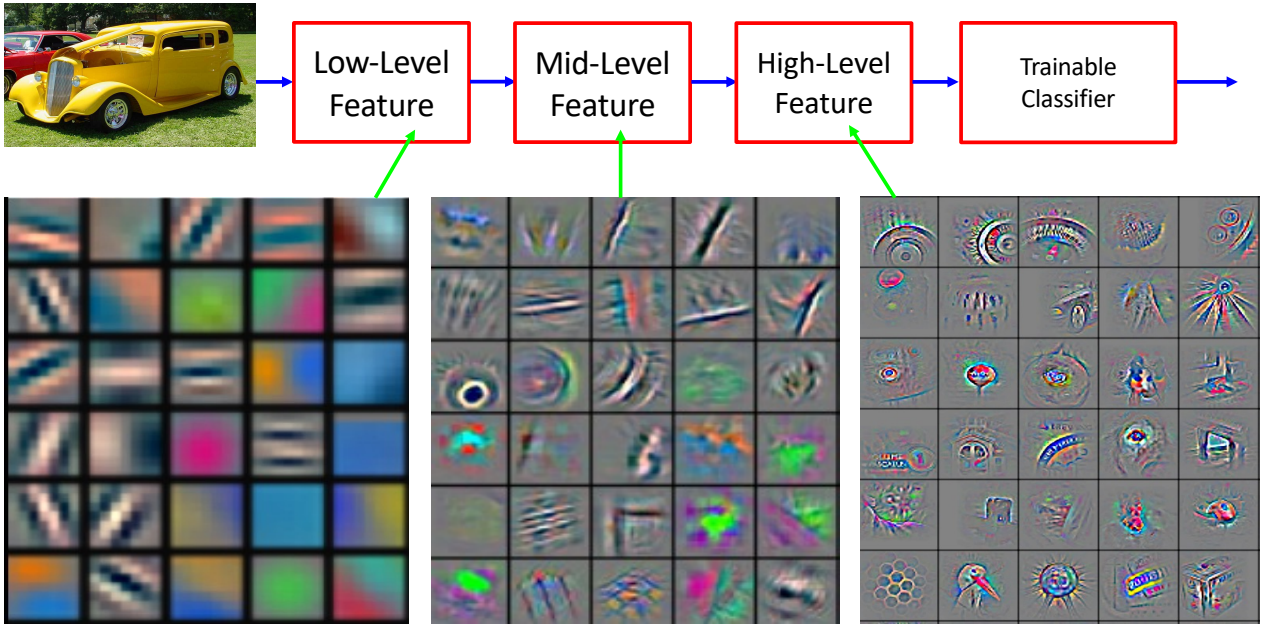
$$P(w_1, w_2, w_3, \ldots, w_T) = \prod_t P(w_t | w_{t-1}, w_{t-2}, \ldots, w_1)$$

Mila

Neural word embeddings - visualization

Why Multiple Layers? The World is Compositional

- Hierarchy of representations with increasing level of abstraction
- Each stage is a kind of trainable feature transform
- Image recognition: Pixel → edge → texton → motif → part → object
- Text: Character → word → word group → clause → sentence → story
- Speech: Sample → spectral band → sound → … → phone → phoneme → word



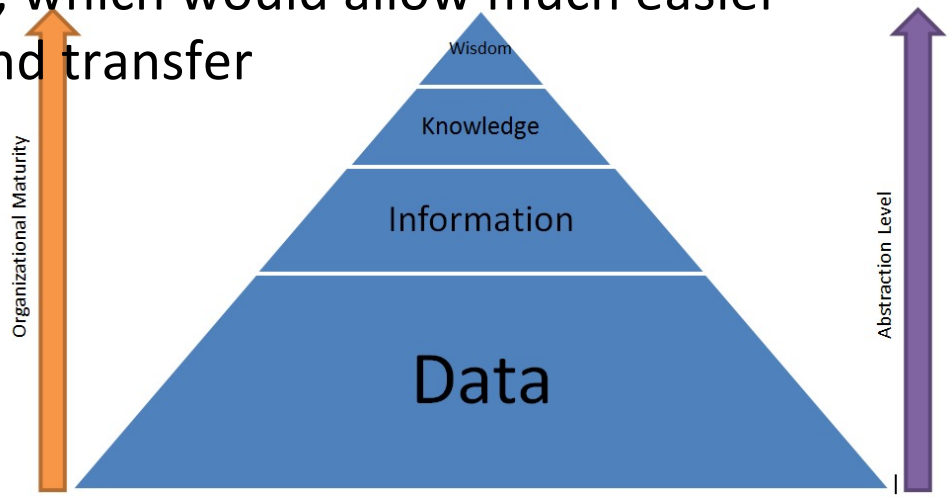
Credits: Yann LeCun

Mila

Learning Higher Levels of Abstraction

(Bengio & LeCun 2007)

- The big payoff of deep learning is to allow learning higher levels of abstraction
- Higher-level abstractions **disentangle the explanatory variables and their causal mechanisms**, which would allow much easier generalization and transfer



Mila

Bypassing the curse of dimensionality

We need to build compositionality into our ML models

- Just as human languages exploit compositionality to give representations and meanings to complex ideas

Exploiting compositionality can give an **exponential** gain in representational power

- Distributed representations / embeddings: feature learning
- Deep architecture: multiple levels of feature learning

Prior assumption: compositionality is useful to describe the world around us efficiently

Mila

Exponential advantage of depth

- Expressiveness of deep networks with piecewise linear activation functions: up to exponential advantage for depth

  (Montufar et al & Bengio, NIPS 2014)

- Number of pieces distinguished for a network with depth L and n_i units per layer is at least

$$\left(\prod_{i=1}^{L-1}\left\lfloor\frac{n_i}{n_0}\right\rfloor^{n_0}\right)\sum_{j=0}^{n_0}\binom{n_L}{j}$$

  or, if hidden layers have width n and input has size n_0

$$\Omega\left(\left(n/n_0\right)^{(L-1)n_0}\, n^{n_0}\right)$$

Hidden units

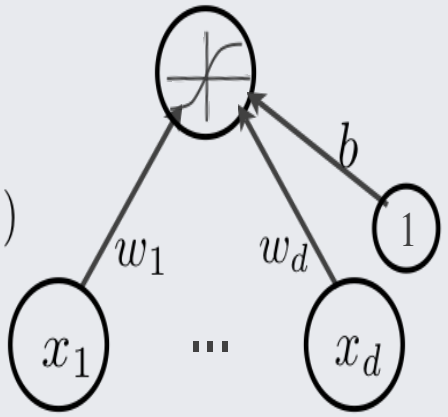
(from Hugo Larochelle)

- Neuron pre-activation (or input activation):

$$a(\mathbf{x}) = b + \sum_i w_i x_i = b + \mathbf{w}^\top \mathbf{x}$$

- Neuron (output) activation

$$h(\mathbf{x}) = g(a(\mathbf{x})) = g(b + \sum_i w_i x_i)$$



- $\mathbf{w}$ are the connection weights
- b is the neuron bias
- $g(\cdot)$ is called the activation function

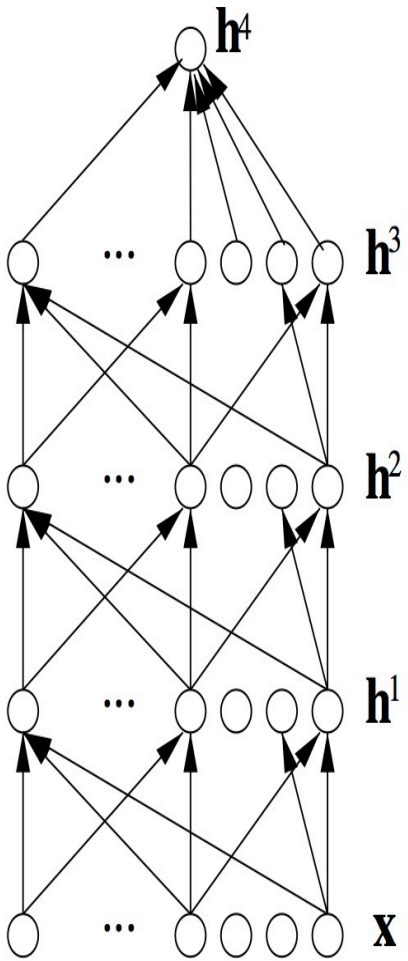
Mila

Multilayer network as universal approximator

A series of non-linear transformations of the same type but different parameters

A single but large enough hidden layer yields a **universal approximator**

More layers allow representing more complex functions with less parameters



Universal approximator property does not guarantee

1. easy optimization (low training error is found)
2. good generalization

Mila

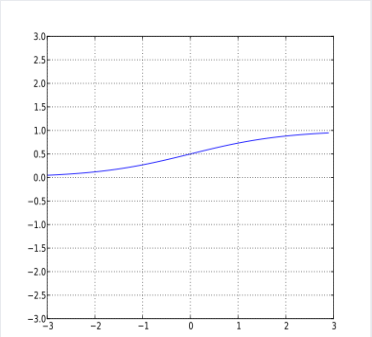
Non-linearity = activation function

- Stacking linear layers: like one (factorized) linear layer
- Universal approximator : stack linear+nonlinear transformations
- Many types of non-linearities are possible: activation function
  - E.g. linear, sigmoid, tanh, rectifier (ReLU), softmax

- *Breakthrough in 2011: it is much easier to train a deep multilayer network with rectifiers (ReLU) than with sigmoid or tanh, making it possible to train deep nets in a purely supervised way for the first first time (Glorot & Bengio AISTATS 2011)*

Topics: sigmoid activation function

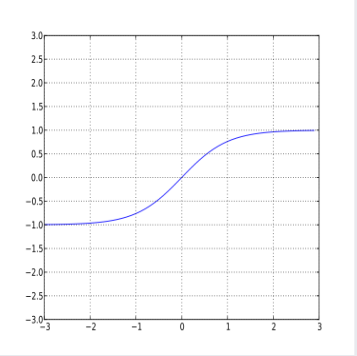
- Squashes the neuron's pre-activation between 0 and 1
- Always positive
- Bounded
- Strictly increasing



$$g(a) = \text{sigm}(a) = \frac{1}{1+\exp(-a)}$$

Topics: hyperbolic tangent ("tanh") activation function

- Squashes the neuron's pre-activation between -1 and 1
- Can be positive or negative
- Bounded
- Strictly increasing



$$g(a) = \tanh(a) = \frac{\exp(a)-\exp(-a)}{\exp(a)+\exp(-a)} = \frac{\exp(2a)-1}{\exp(2a)+1}$$

Topics: softmax activation function

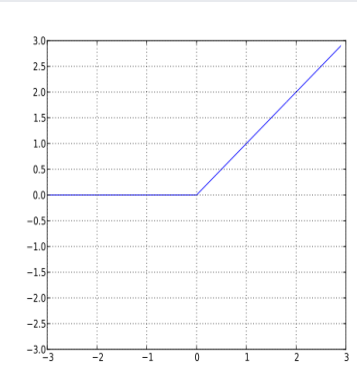
- For multi-class classification:
  - we need multiple outputs (1 output per class)
  - we would like to estimate the conditional probability $p(y = c|\mathbf{x})$
- We use the softmax activation function at the output:

$$\mathbf{o}(\mathbf{a}) = \text{softmax}(\mathbf{a}) = \left[\frac{\exp(a_1)}{\sum_c \exp(a_c)} \cdots \frac{\exp(a_C)}{\sum_c \exp(a_c)}\right]^\top$$

  - strictly positive
  - sums to one
- Predicted class is the one with highest estimated probability

Topics: rectified linear activation function

- Bounded below by 0 (always non-negative)
- Not upper bounded
- Strictly increasing
- Tends to give neurons with sparse activities



$$g(a) = \text{reclin}(a) = \max(0, a)$$

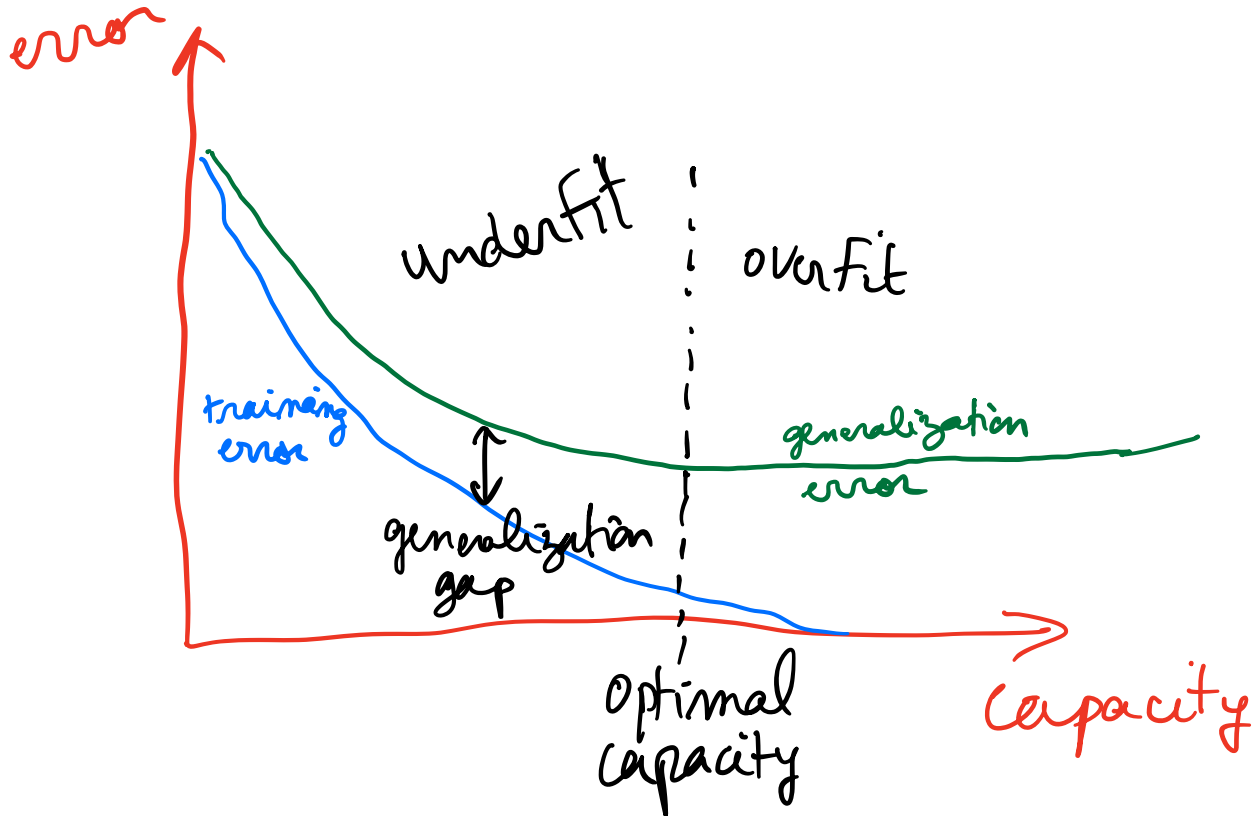
Statistical Learning Theory 101

- Training set → empirical risk = average loss
- Empirical risk minimization: choose a function in some family with low avg training loss
- Typically set up as an approximate iterative optimization problem
- **Parametric**: family of functions spanned by choice of parameter, $f_\theta(x)$
- **Non-parametric**: complexity of solution can grow with amount of data
- **Capacity**: number of training examples one can always learn perfectly (more precisely, VC-dimension, see Vapnik's book "The nature of statistical learning theory" , 2000)

Statistical Learning Theory 101

- What we care about is **generalization error**: expected loss under the unknown data distribution
- Estimate it empirically with **out-of-sample performance**: use a separate **test set** Statistical **confidence intervals around average test error** can be computed
- **IID** or **exchangeability** assumption, obtained by shuffling data before train/test split
- **Optimal capacity**: gives lowest generalization error
- **Underfitting**: capacity below optimal capacity
- **Overfitting**: capacity above optimal capacity
- **Effective capacity** includes effect of finite compute, may be smaller

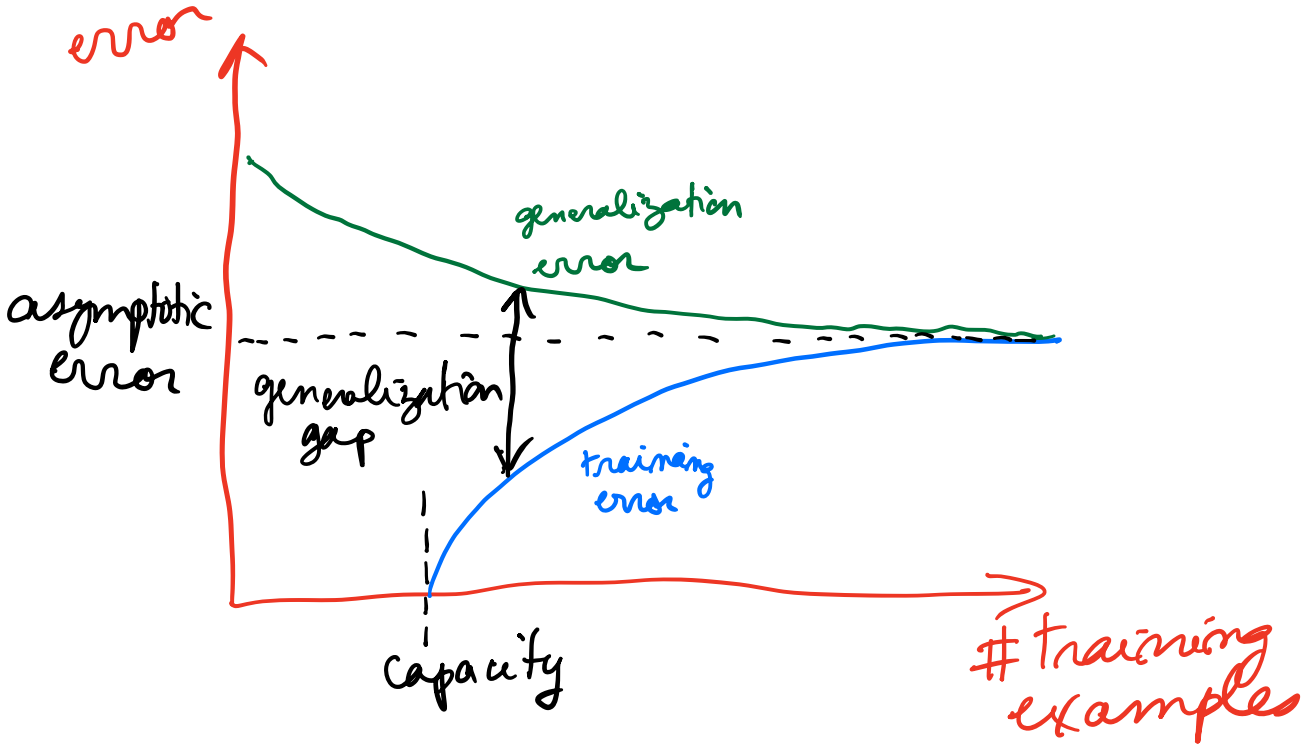
Statistical Learning Theory 101



Fixed capacity case
(e.g. parametric)

Large neural net case may behave differently (double descent scenario)

Statistical Learning Theory 101

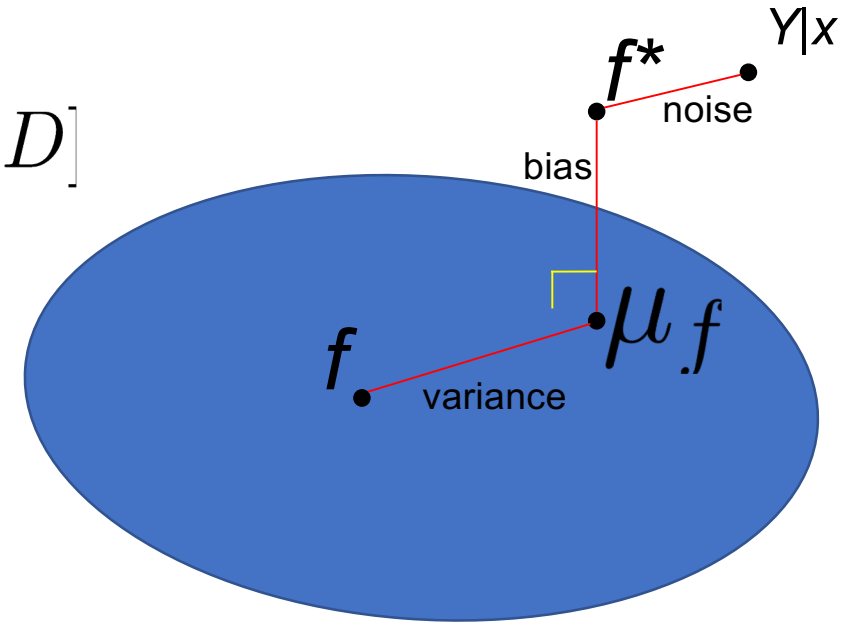


Fixed capacity case
(e.g. parametric)

In non-parametric case, capacity increases with #training examples and error may approach Bayes error

Total Expected Error = Bias + Variance + Noise

$$\mu_f(x) = E_\theta[f(x,\theta)|D]$$



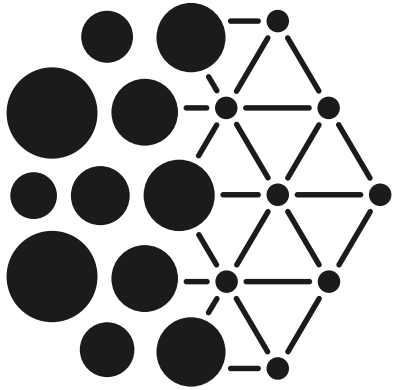
Total expected error:

$$E[(f(x;\theta) - Y)^2|D] = E[(f^*(x) - Y)]^2 + E[(f^*(x) - \mu_f(x;\theta))^2|D] + E[(f(x,\theta) - \mu_f(x))^2|D]$$

noise bias variance

Mila

MERCI!



Mila

Université de Montréal

McGill

Québec

CIFAR