Deep Learning: Introduction

Machine Learning in der Medizin

Asan Agibetov, PhD

asan.agibetov@meduniwien.ac.at

Medical University of Vienna
Center for Medical Statistics, Informatics and Intelligent Systems
Section for Artificial Intelligence and Decision Support
Währinger Strasse 25A, 1090 Vienna, OG1.06

October 31, 2019

Introduction

- References (available online for free):
  - "Neural Networks and Deep Learning". Michael A. Nielsen, Determination Press, 2015
    - intuition first, math after
  - "Deep Learning". Ian Goodfellow, Youshua Bengio and Aaron Courville, MIT Press, 2016
    - formal with fair amount of intuition, more general than Nielsen
- These slides are based on DL courses:
  - NVIDIA Deep Learning Institute Deep Learning course
  - Course notes "CNN for Visual Recognition" (Stanford, Spring 2017)
  - Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato (Facebook AI Research), DeepLearn Summer School - Bilbao, 17-21 July 2017

Deep Learning powers all devices around us

Speech Recognition

Recommender Systems

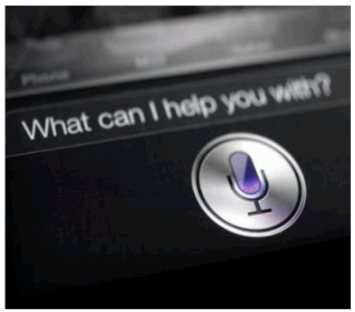
Autonomous Driving

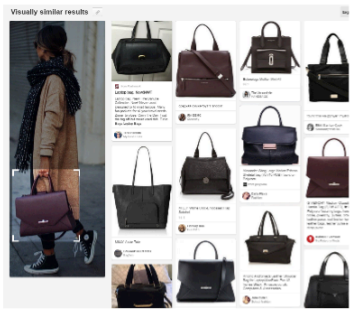
Real-time Object Recognition

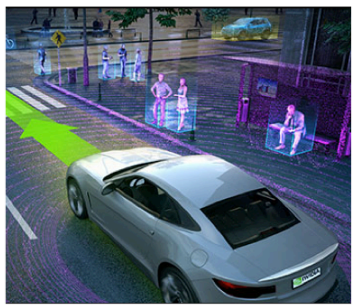
Robotics

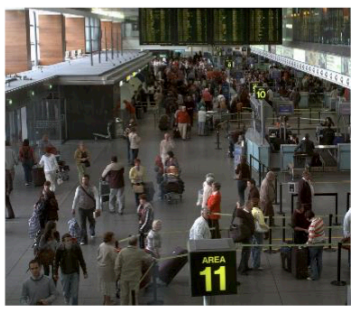
Real-time Language Translation

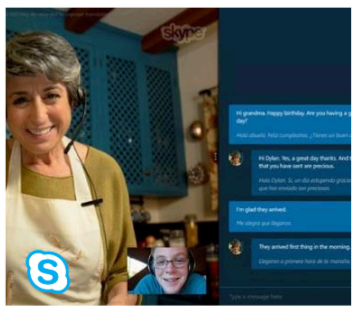
Many More...











(Image credit) NVIDIA Deep Learning Institute course

Deep Learning and Turing award

The New York Times

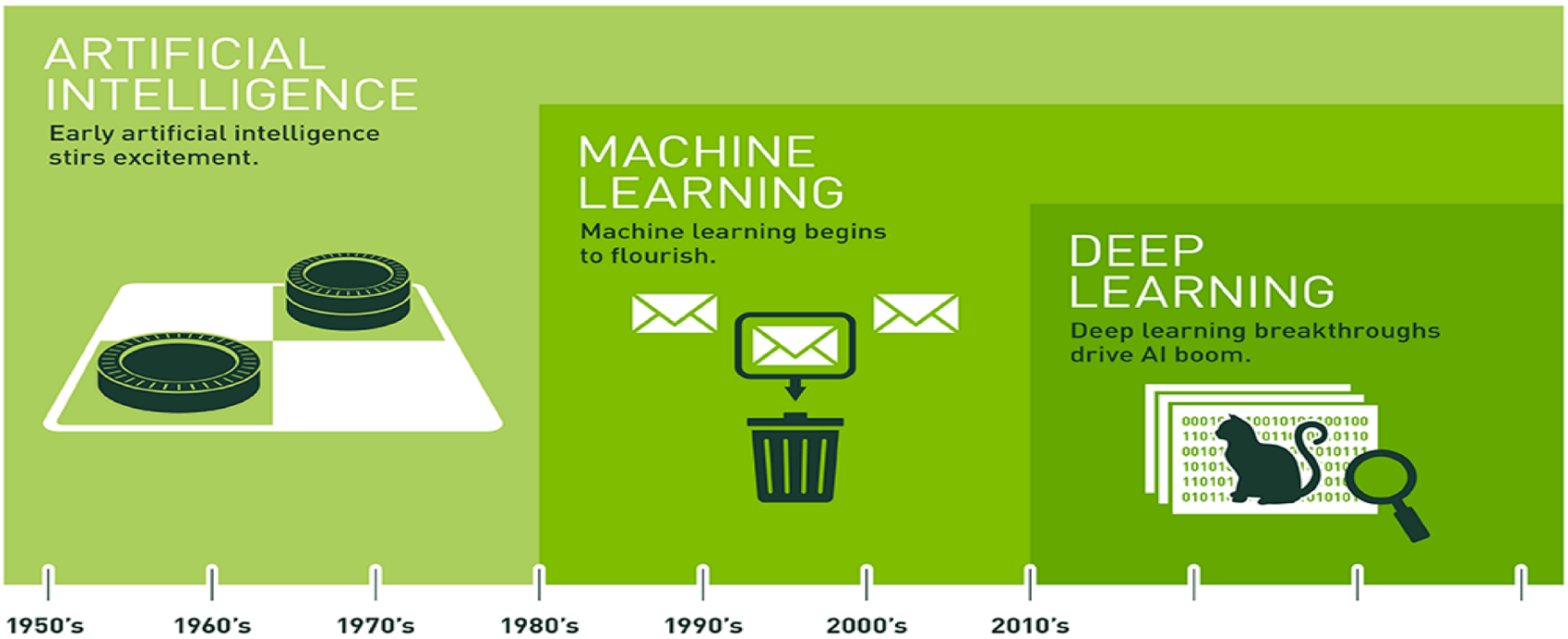
Turing Award Won by 3 Pioneers in Artificial Intelligence

From left, Yann LeCun, Geoffrey Hinton and Yoshua Bengio. The researchers worked on key developments for neural networks, which are reshaping how computer systems are built.

From left, Facebook, via Associated Press; Aaron Vincent Elkaim for The New York Times; Chad Buchanan/Getty Images

(Image credit) NY times website

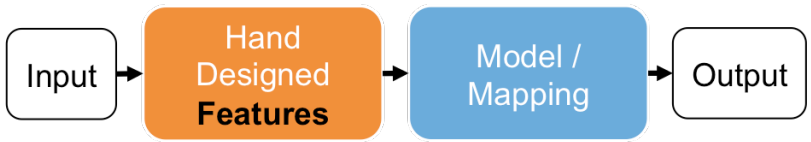
History of AI



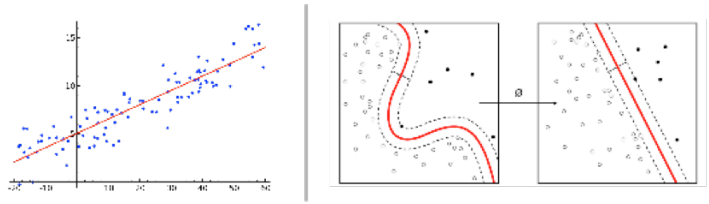
(Image credit) NVIDIA Deep Learning Institute course

Feature learning: Machine Learning vs. Deep Learning

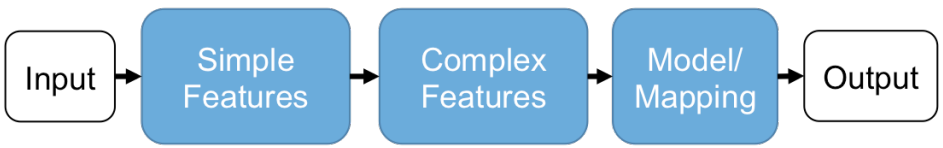
Classic Machine Learning [1990 : now]



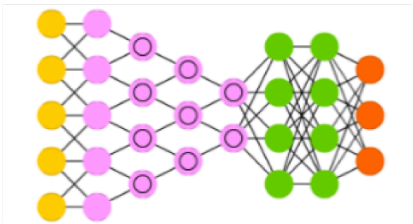
Examples [Regression and SVMs]



Deep/End-to-End Learning [2012 : now]

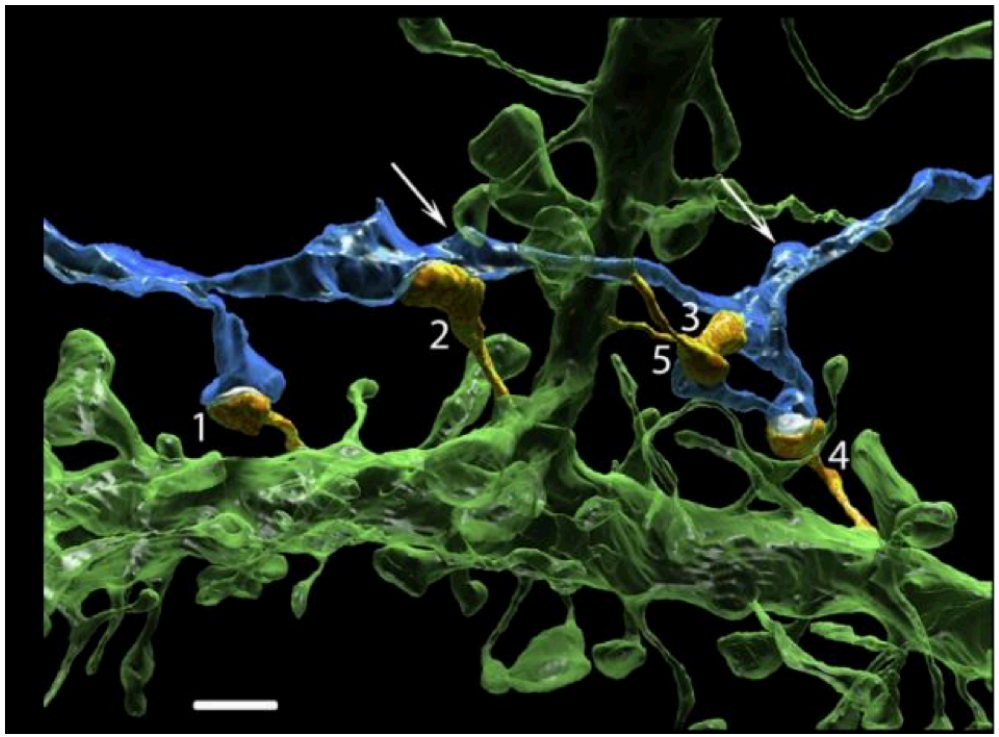


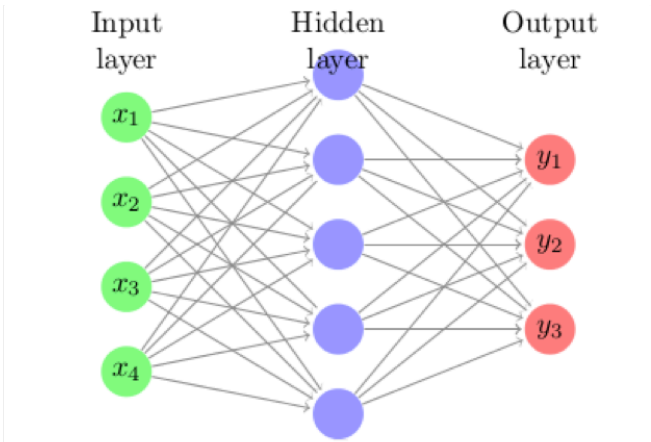
Example [Conv Net]



(Image credit) NVIDIA Deep Learning Institute course

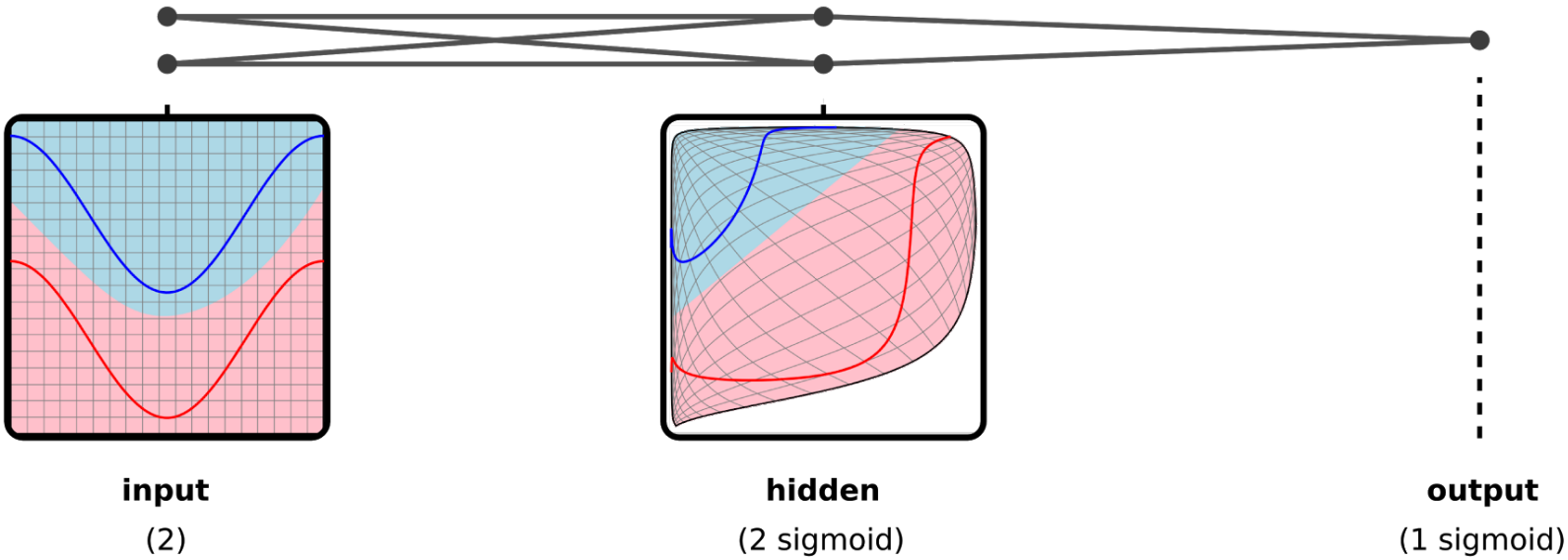
Deep Learning: biological inspiration





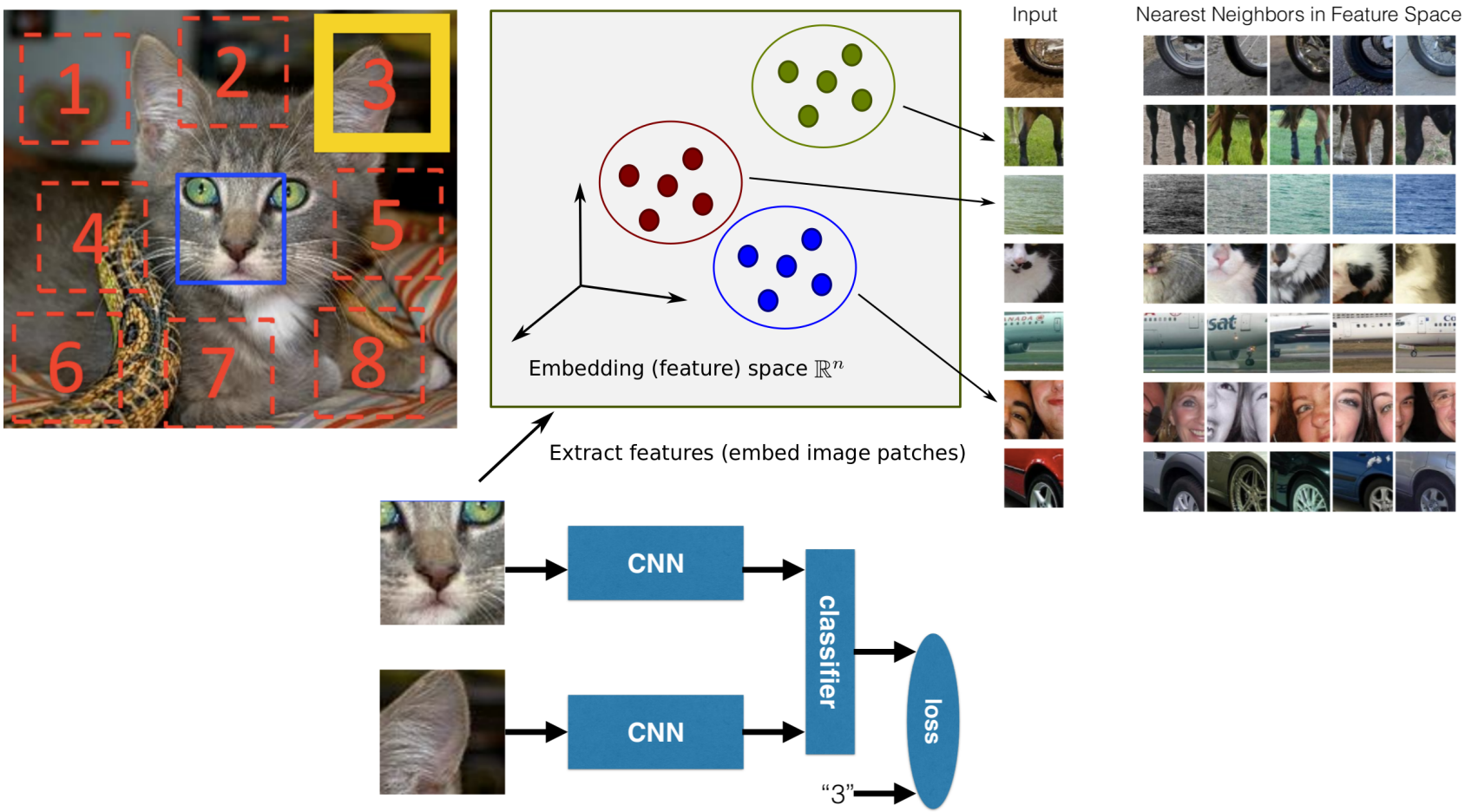
(Image credit) NVIDIA Deep Learning Institute course

Deep Learning: Representation learning and manifold hypothesis



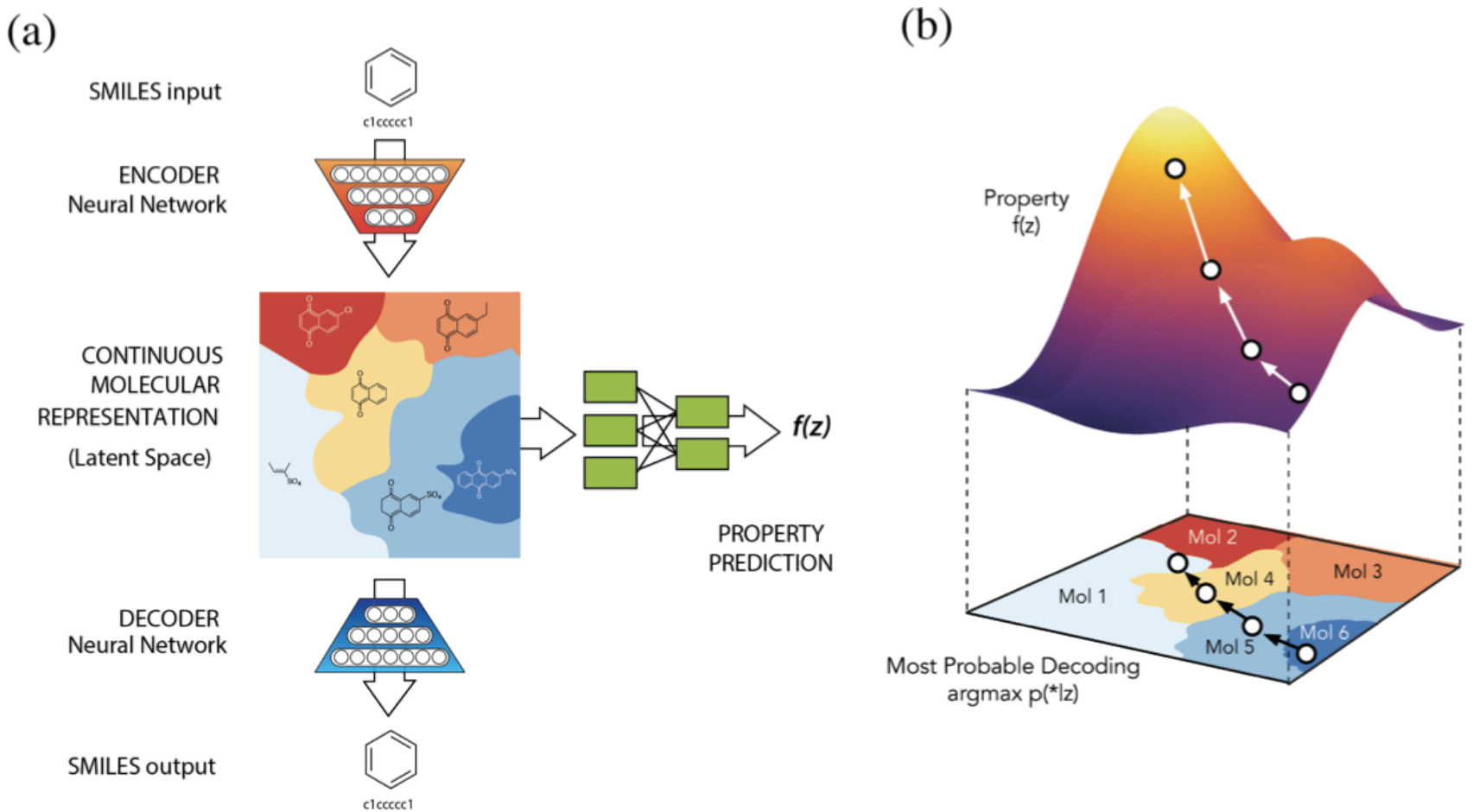
(Image credit) Chris Olah's (OpenAI) blogpost on "Neural Networks, Types, and Functional Programming" (http://colah.github.io/posts/2015-09-NN-Types-FP/

Deep Learning: Manifold hypothesis



(Image credit) Adapted from C. Doersch et al. "Unsupervized Visual Representations Learning by Context Prediction". ICCV 2015

Deep Learning: Latent variables (probabilistic models)

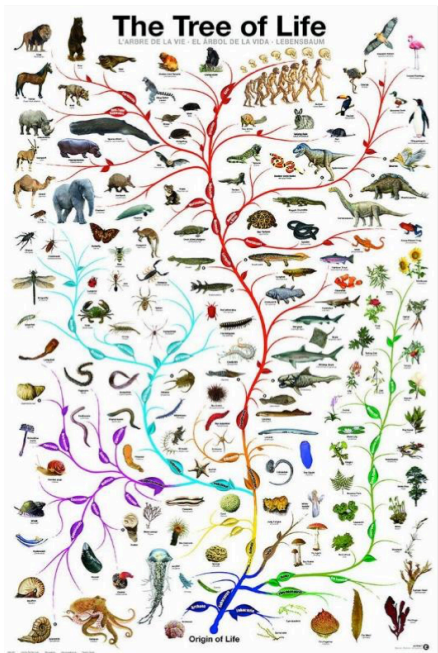


(Image credit) Gomez-Bombarelli "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules" arxiv. 2017

Knowledge acquisition and machine intelligence

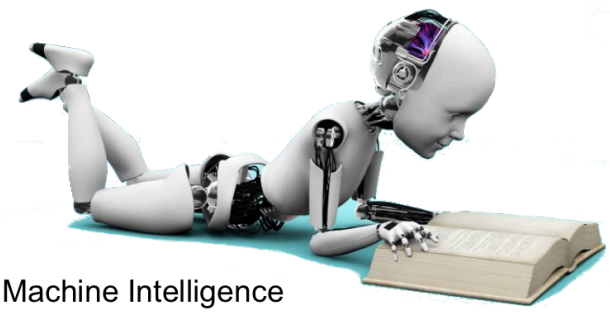
4th revolution in knowledge acquisition

1st - Evolution



2nd - Experience/Brain

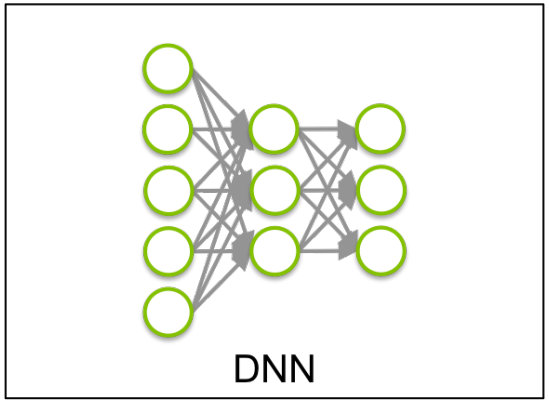
3rd - Culture

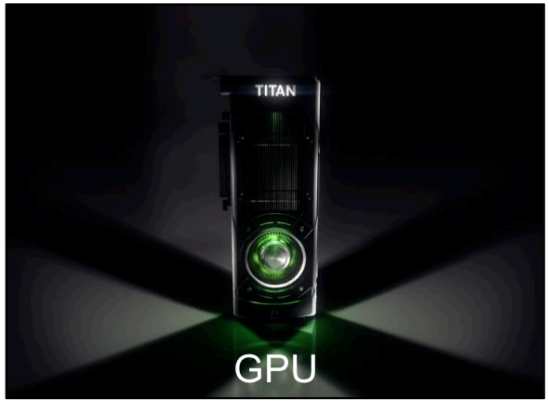


4th – Machine Intelligence

(Image credit) NVIDIA Deep Learning Institute course

Why Deep Learning took off only recently?



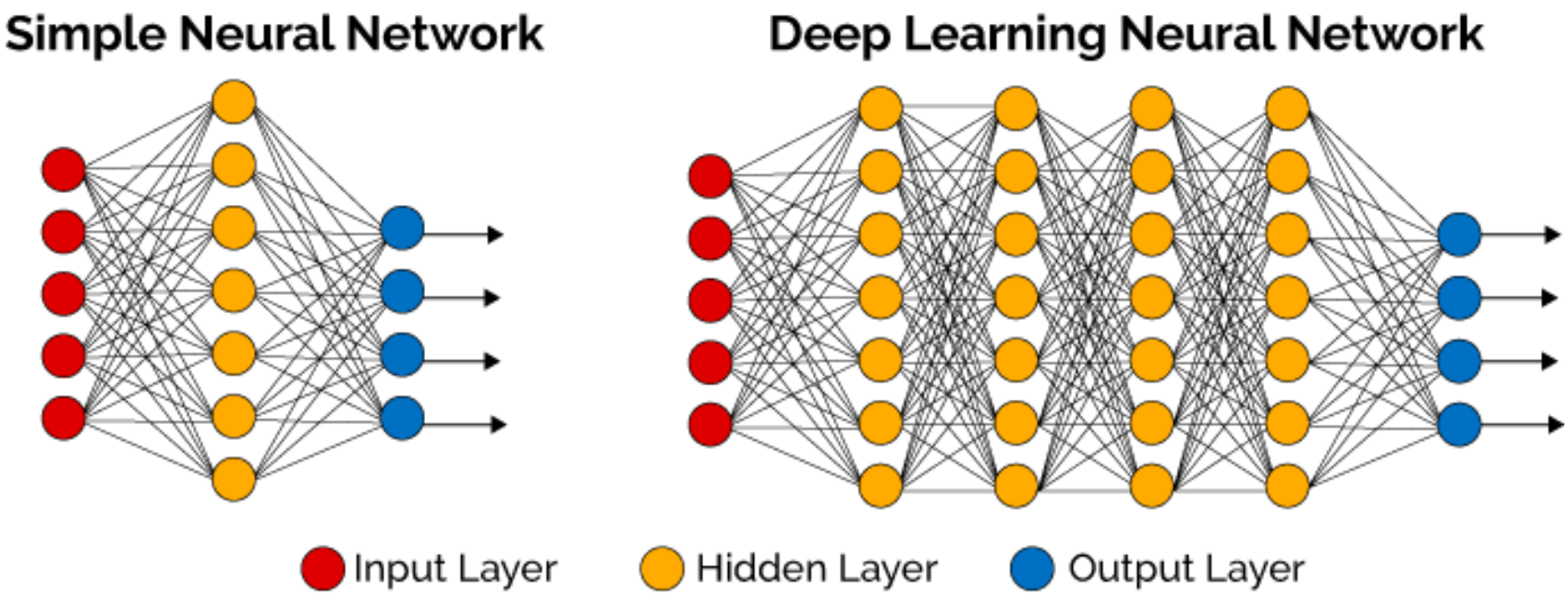


WIRED

(Image credit) NVIDIA Deep Learning Institute course

Deep Learning - what's in the name?

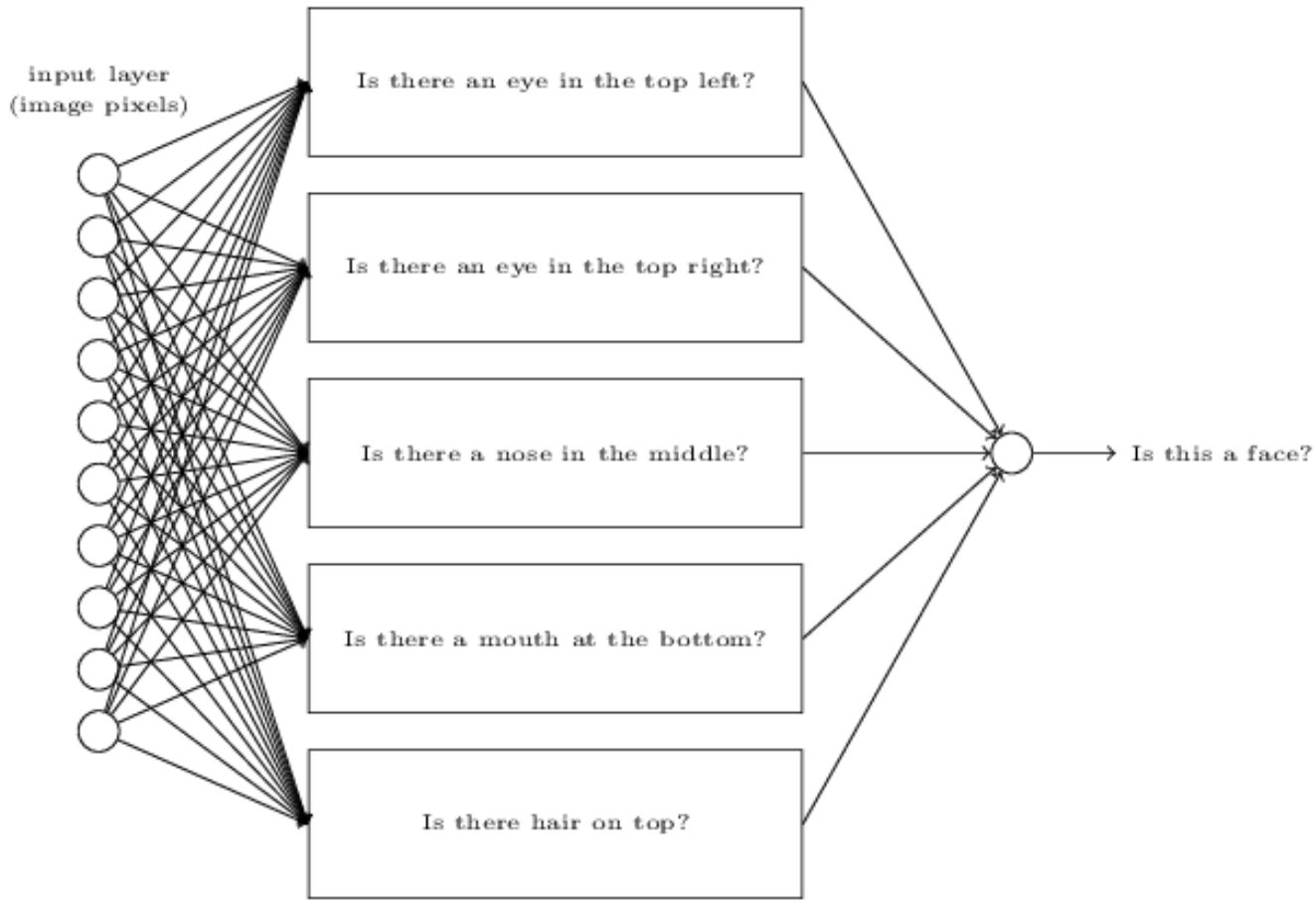
- DL, roughly speaking, is NN with many layers and many neurons in each layer
  - not true in all cases though (e.g., embeddings are often shallow)



(Image credit) https://hackernoon.com/log-analytics-with-deep-learning-and-machine-learning-20a1891ff70e

Hierarchical feature learning

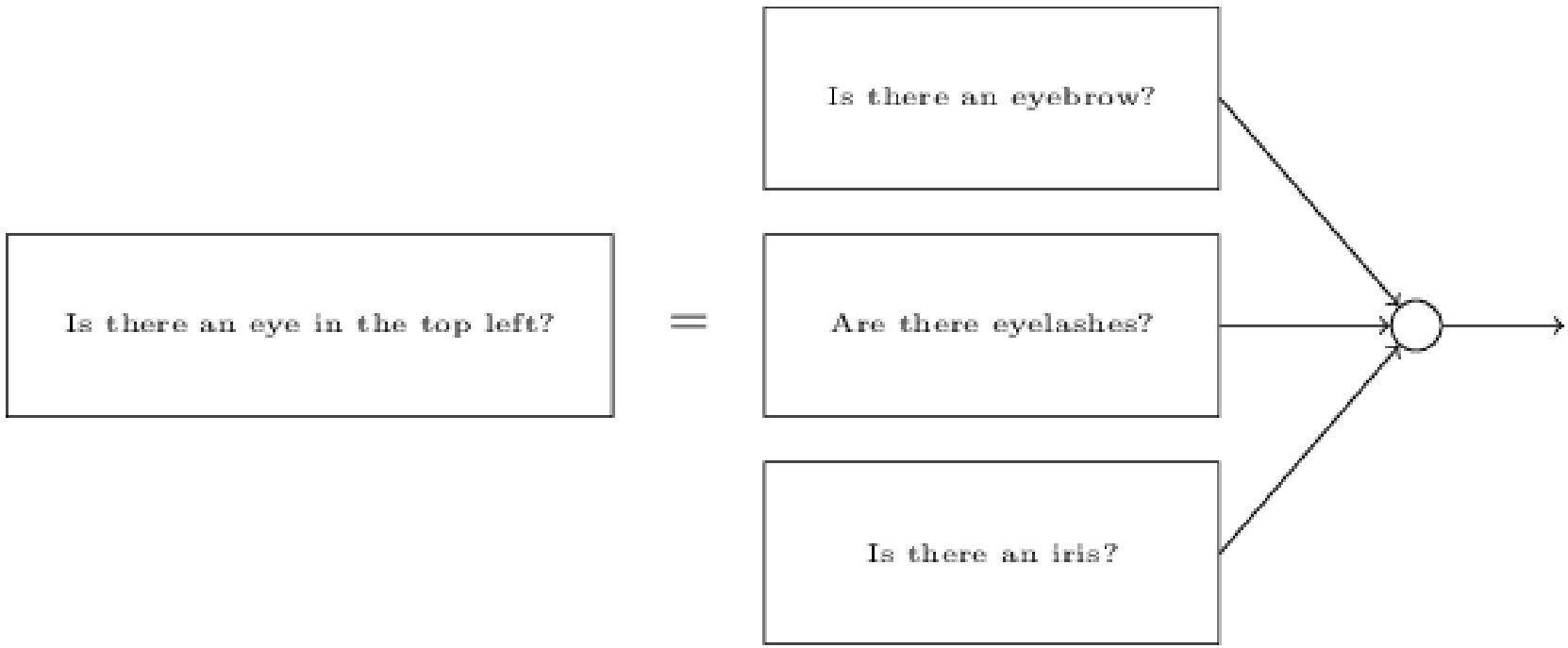
- DL learns features automatically, and hierarchically



(Image credit) "Neural Networks and Deep Learning". Michael A. Nielsen, Determination Press, 2015

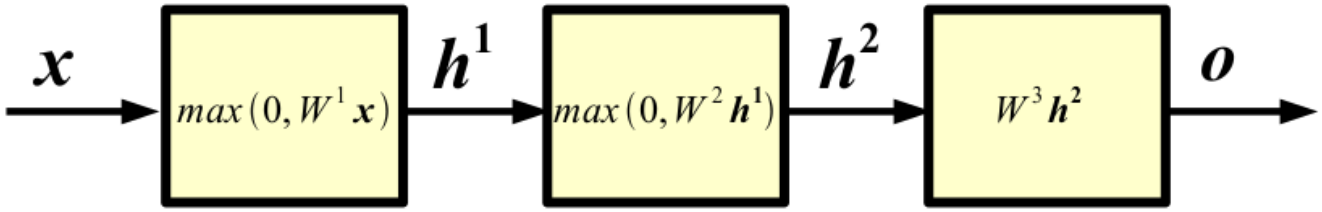
Hierarchical feature learning

- Learned features can be combined



(Image credit) "Neural Networks and Deep Learning". Michael A. Nielsen, Determination Press, 2015

Neural networks



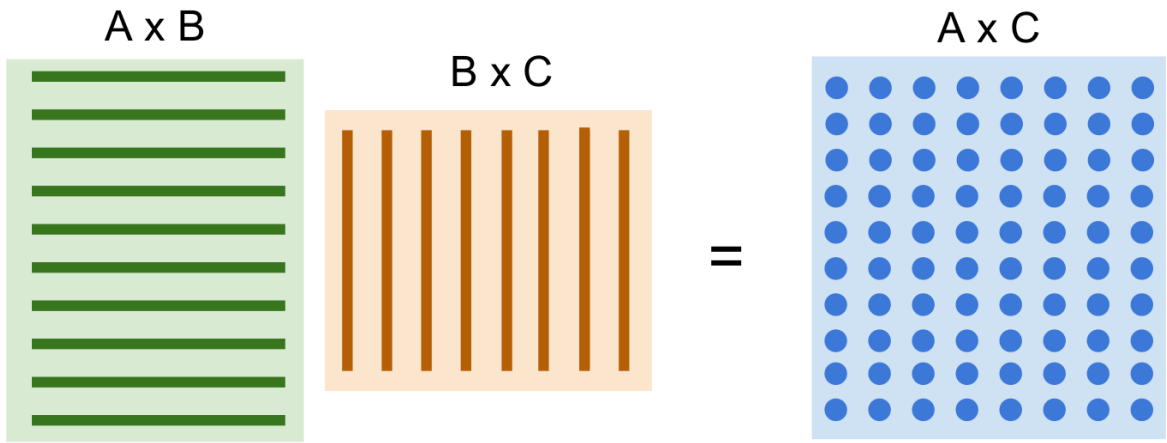
- Universal function approximation that maps input to output
  - $f : \mathbb{R}^n \mapsto \mathbb{R}^m$
- Class of functions considered to map input to output
  - composition of simpler (including non-linear [1]) functions
    - h^1 is non-linear $max(0, \mathbf{W} \cdot \vec{x} + \vec{b})$ aka ReLU
  - $f = o \circ h^2 \circ h^1 \circ x$

[1] composition of only linear function would be equivalent to one linear function

(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Computation: speed up and parallelization with GPU

In a nutshell DL is all about matrix multiplication



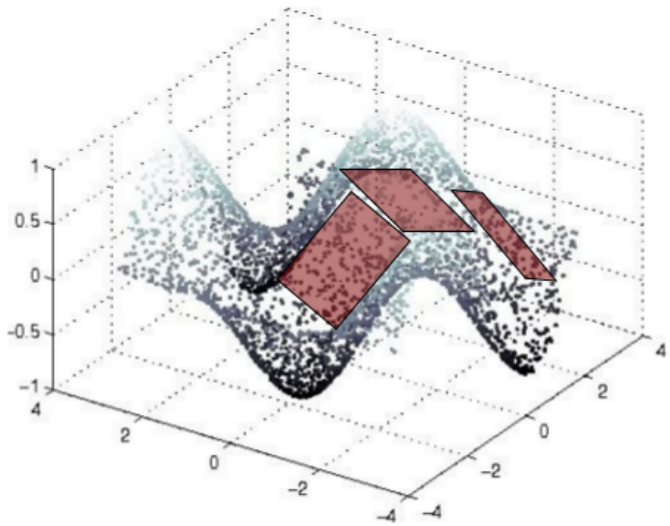
- Entries of the $A \times C$ matrix can be computed in parallel with GPU
- $A \times B$ rows and $B \times C$ cols loaded in the shared memory

---

(Image credit) Course notes "CNN for Visual Recognition" (Stanford, Spring 2017)

Why ReLU (rectified linear unit)

- ReLU layers provide *piece-wise* linear tiling
  - # planes grows exponentially w. # hidden units
- Reduce probability of vanishing gradients
  - when $\mathbf{W}x + b > 0$, derivative stays constant (faster learning)
- Sparse representations
  - $\mathbf{W}x + b \leq 0$ many neurons are zero



(Image credit) Montufar et al. "On the number of linear regions of DNNs", arXiv, 2014

Optimization: finding the best f

Typical setup for optimization

- f can be parameterized with Θ ($f = \Theta \cdot x$ linear case)
- minimizing (learning) the loss function V over all training examples $1 \ldots n$
- plus regularizations on:
  - $\lambda_2(f)$ - controls complexity of the function (usually norm $\|f\|$)
  - $\lambda_1(f, \Theta)$ - sparsity of the solution, where Θ parameters of f

$$f^* = \underset{f}{argmin} = \sum_{1}^{n} V(y, f(x)) + \lambda_2(f) + \lambda_1(f, \Theta)$$

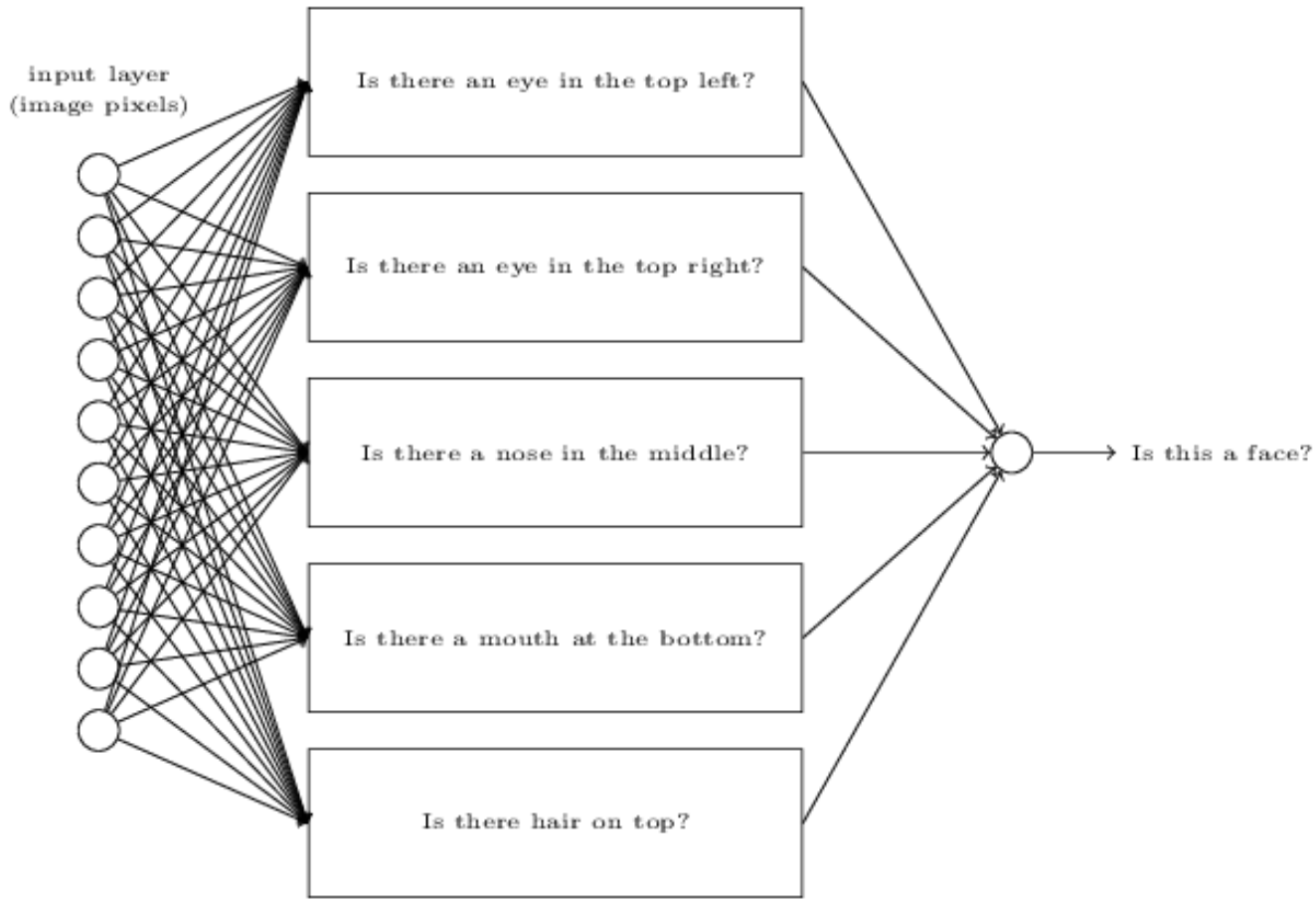
- to find f^* you need to minimize complicated function
- **backpropagation** gives the gradients of that complicated function

Recap

- Neural nets - chain (composition) of non-linear operations, implementing highly non-linear functions
- Forward pass computes error between the currently learnt mapping function and the actual output
- Backward pass computes gradients w.r.t. inputs at each layer and parameters
- Optimization (minimization of the loss error) done by stochastic gradient descent (or variants of it)

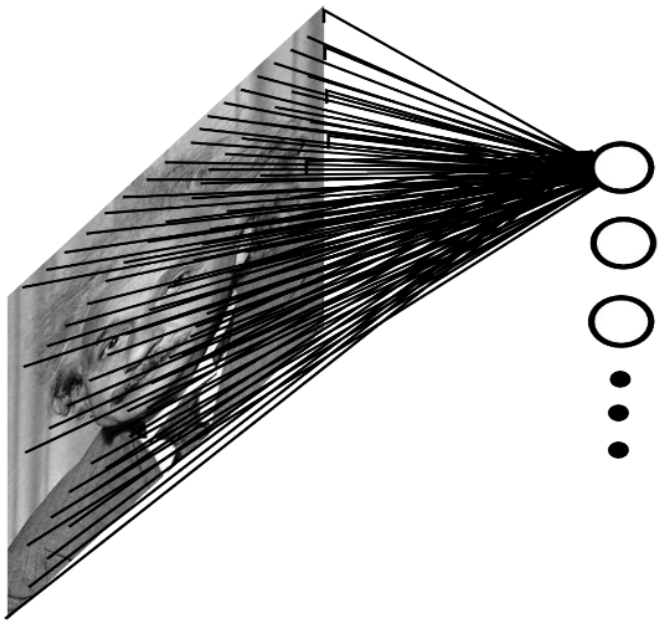
Hierarchical feature learning

- DL learns features automatically, and hierarchically
  - (Convolutional) Neural Network to detect a face



(Image credit) "Neural Networks and Deep Learning". Michael A. Nielsen, Determination Press, 2015

Deep Learning for Vision

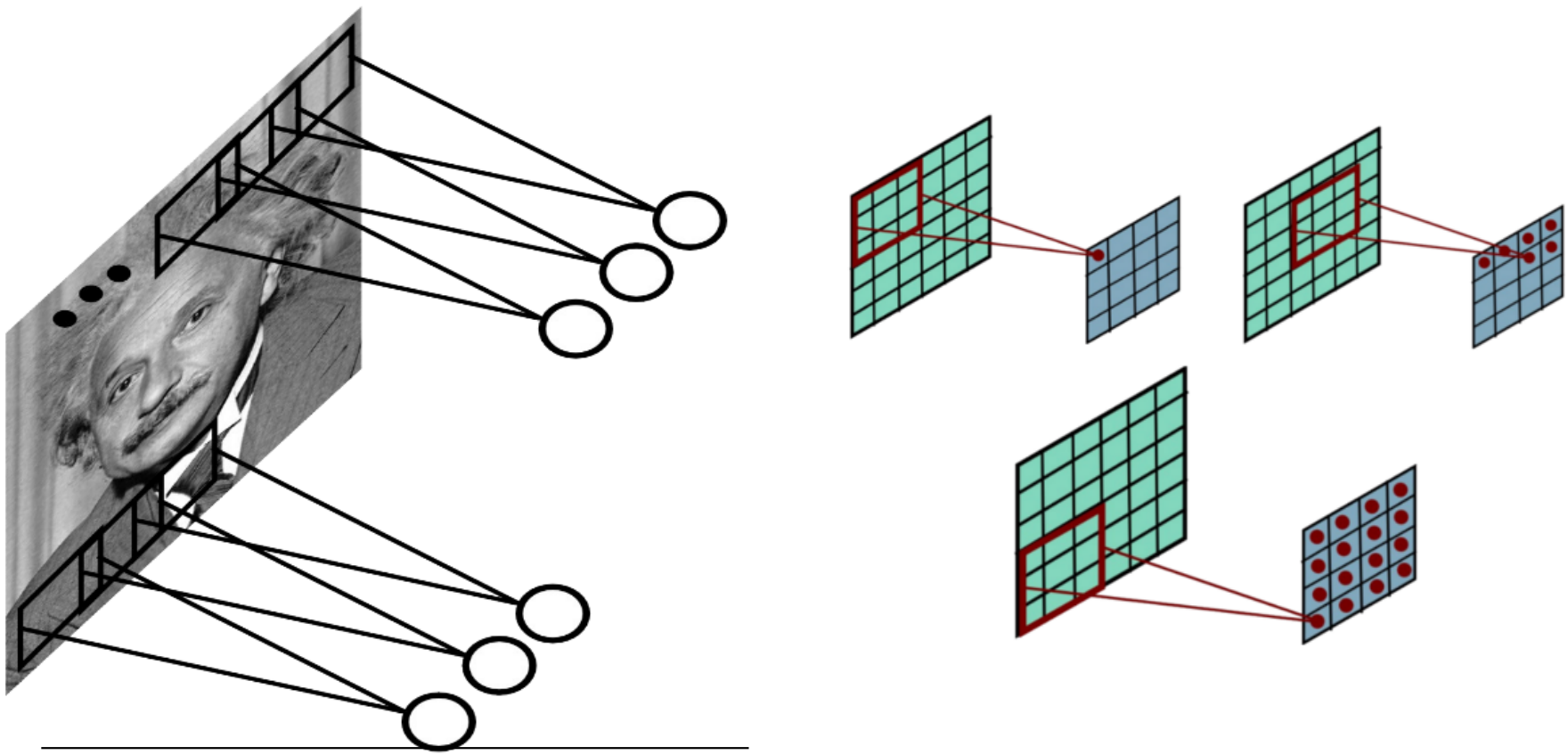


- Idea:
  - unwrap images (2d matrices) into 1d vectors
    - $\mathbb{R}^{200 \times 200} \mapsto \mathbb{R}^{40000}$
  - feed them into Neural Networks (fully connected layers)
- Problem:
  - spatial correlation is local
  - waste of resources
  - not robust to transformations (scale, rotation, translation)
  - depends on the original image size

(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Convolutional Layer

- shared weights across the whole image
- convolution takes advantage of
  - *stationarity* (similar statistics at different locations)
  - local spatial correlation

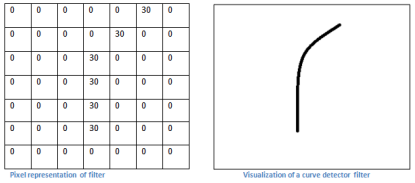


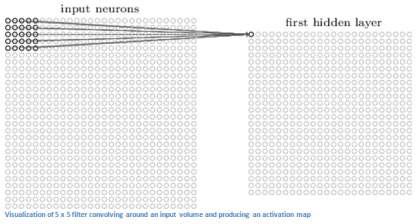
(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Convolutional layer activations

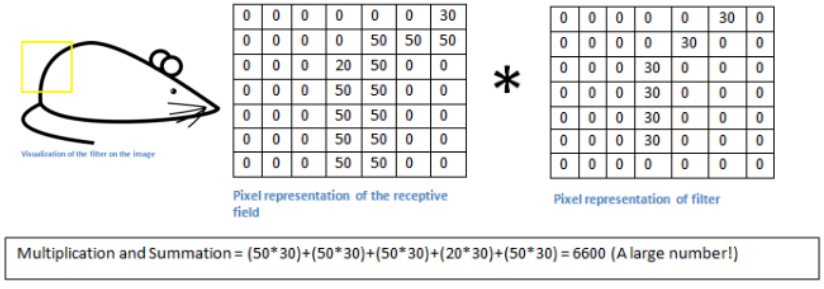
Receptive field (aka. filter, kernel)

Swipes through input and outputs activation maps

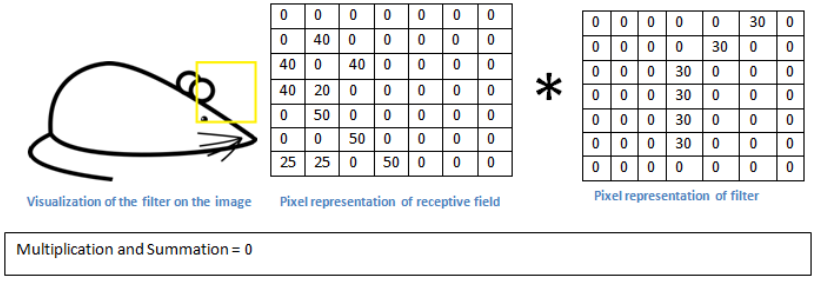




activation maps - results of convolutions (sum of element-wise multiplications)



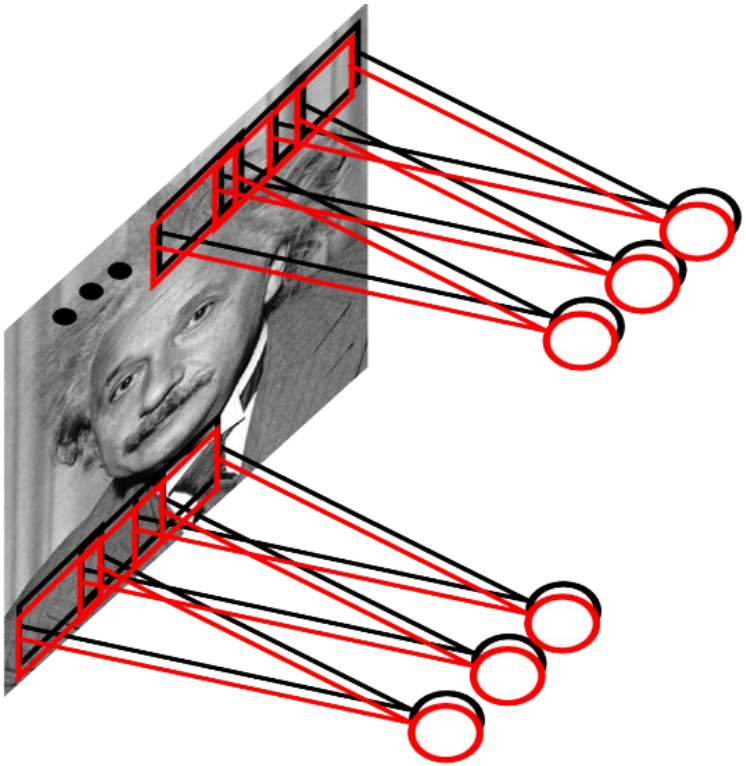
positive activation

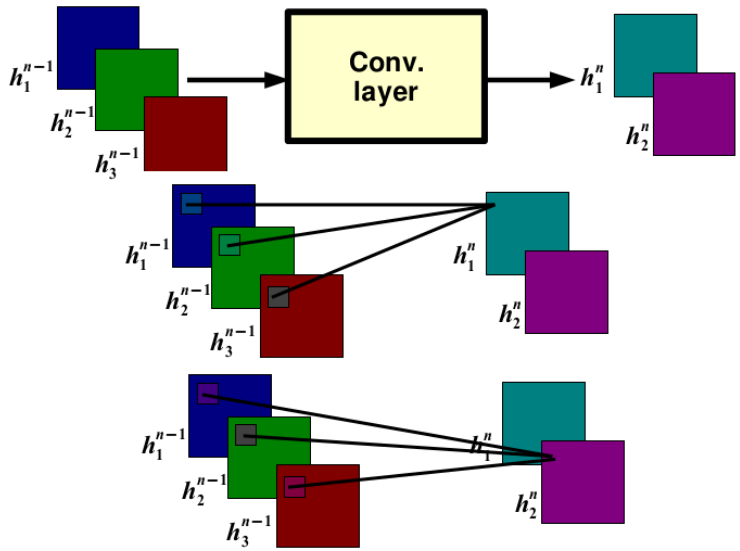


null activation

Multiple convolutional filters

$$h_j^n = max(0, \sum_{k=1}^{K} h_k^{n-1} * w_{kj}^n)$$

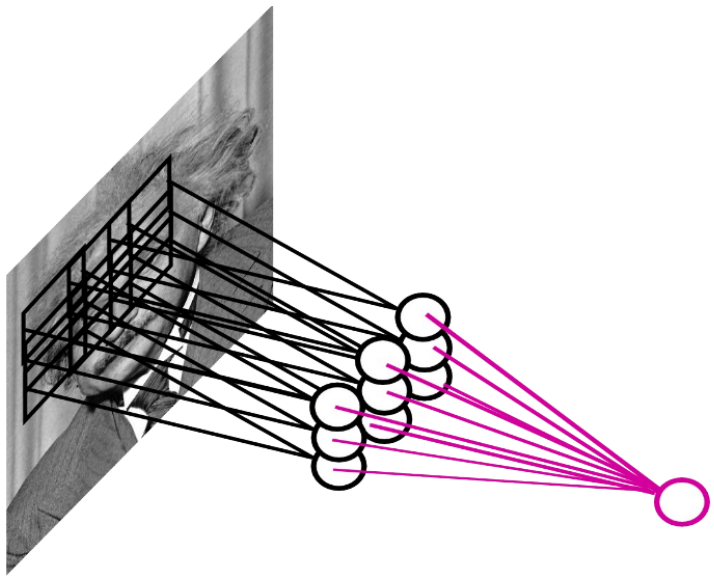




(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Pooling layer

$$h_j^n(x, y) = max_{\bar{x} \in N(x), \bar{y} \in N(y)} h_j^{n-1}(\bar{x}, \bar{y})$$

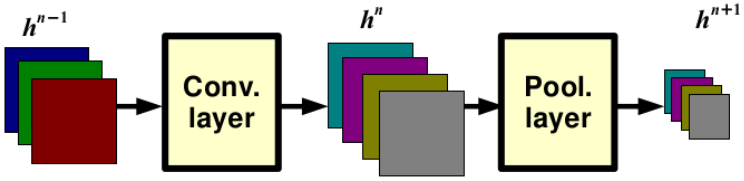


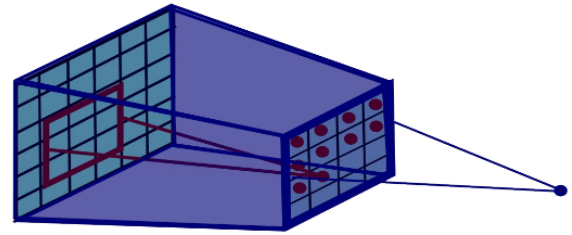
- Pooling layer goal: spatial robustness for feature extraction
- Assume our filter is *eye* dectector
  - Pooling layer makes *eye* detector robust to exact location of eye
- by *pooling* (e.g., taking max) filter responses at different locations
  - we gain robustness to the exact spatial location of features

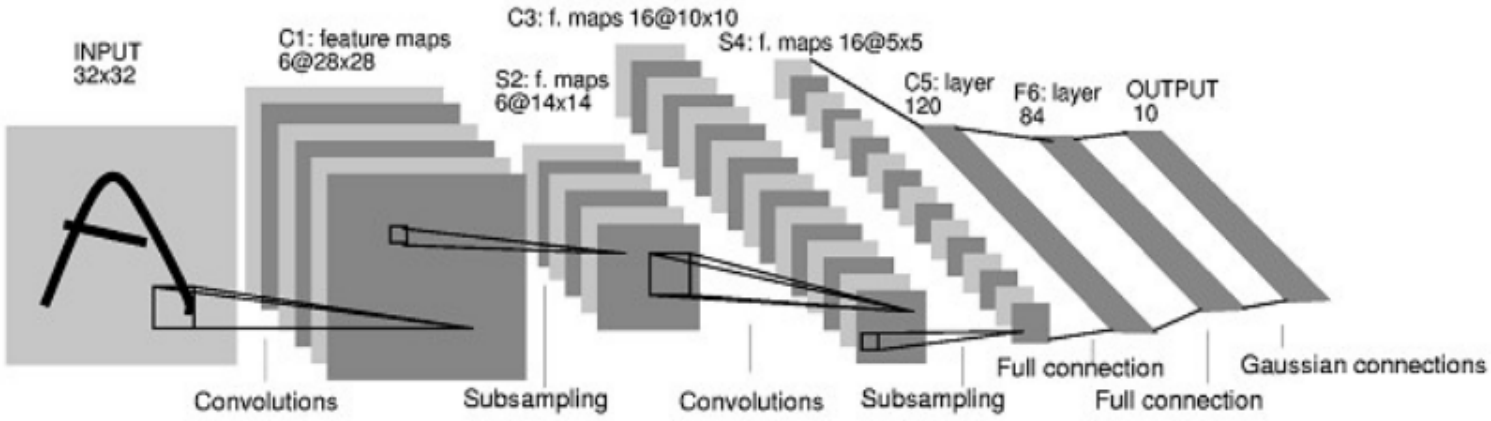
---

(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

ConvNets architecture



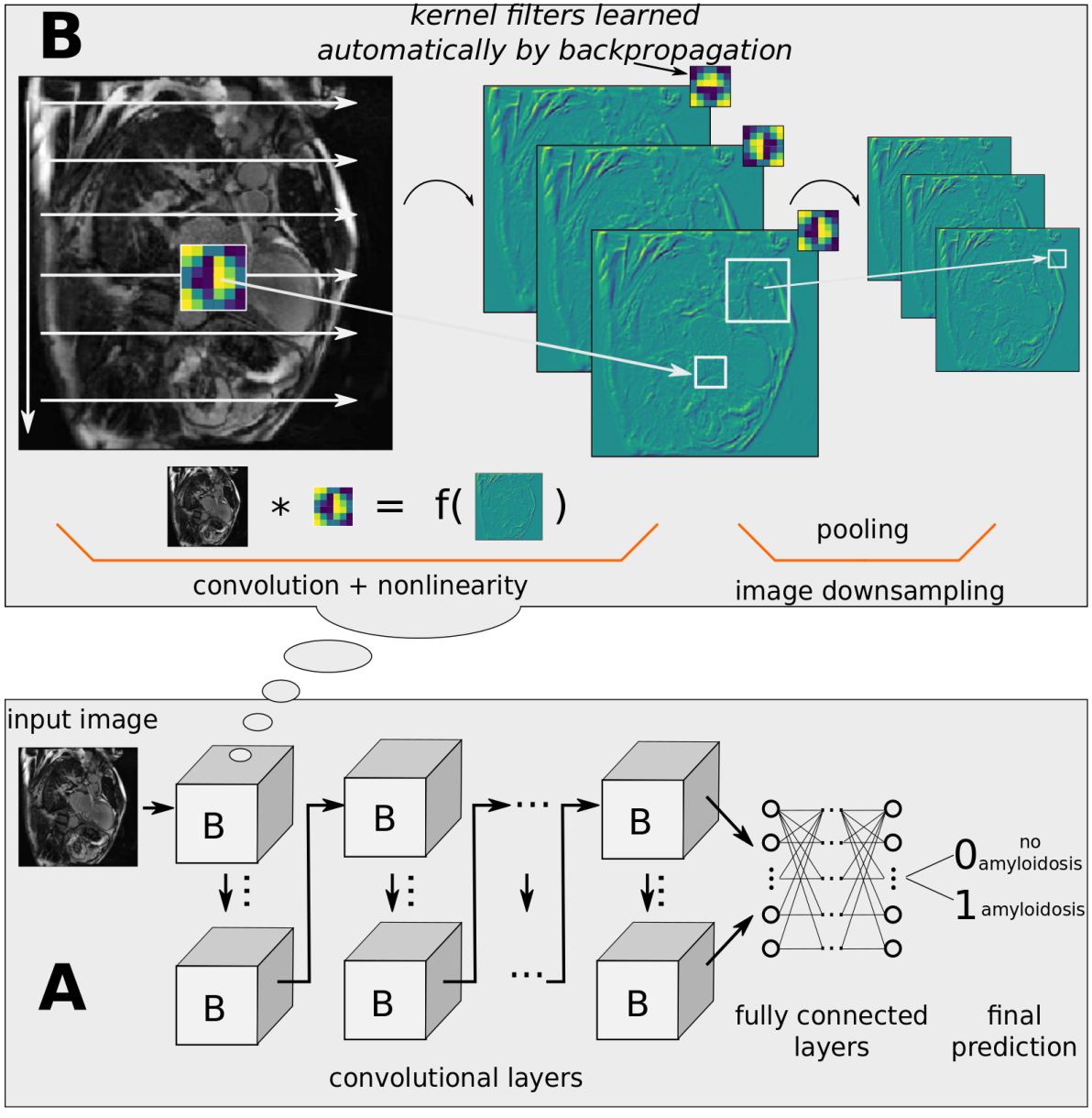




(Image credit (top)) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

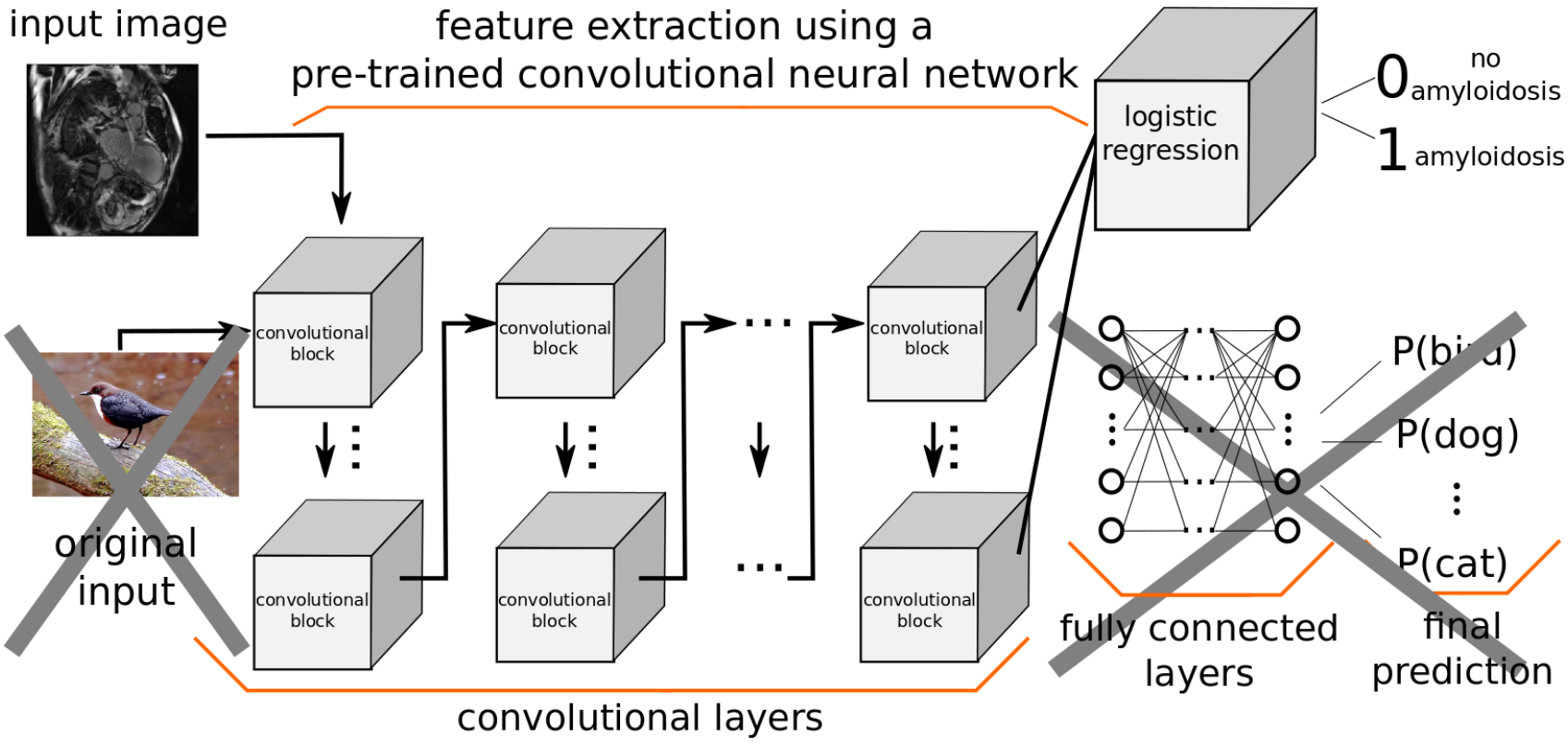
(Image credit (bottom)) LeCun et al. "Gradient based learning applied to document recognition" IEEE 1988

Diagnosis of medical images

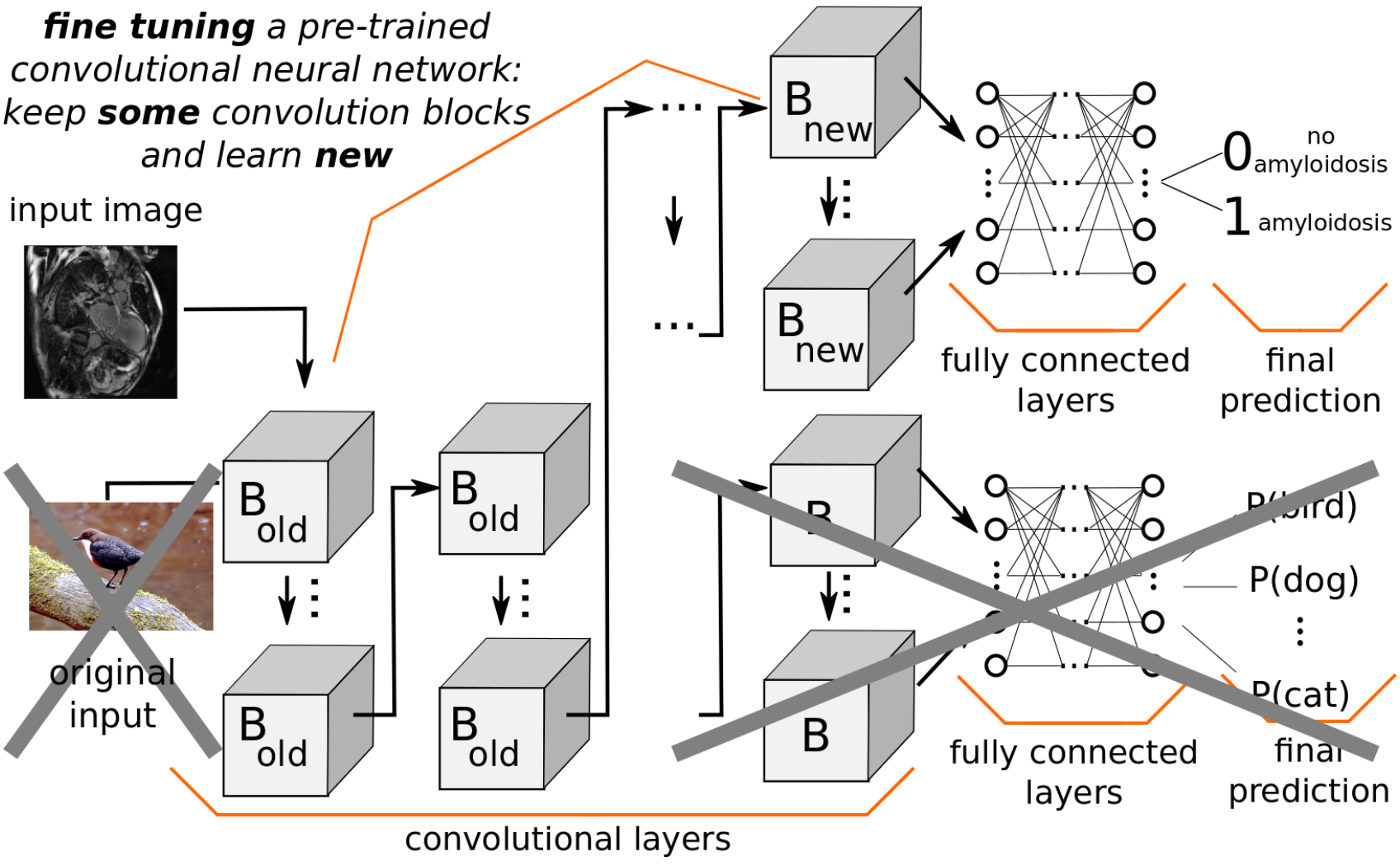


Transfer learning

Feature extraction



Fine tuning


fine tuning *a pre-trained convolutional neural network: keep* ***some*** *convolution blocks and learn* ***new***
input image
original input
B old
B old
B old
B old
B new
B new
B
B
fully connected layers
fully connected layers
0 no amyloidosis
1 amyloidosis
final prediction
P(dog)
P(cat)
convolutional layers

Embeddings: representing symbolic data

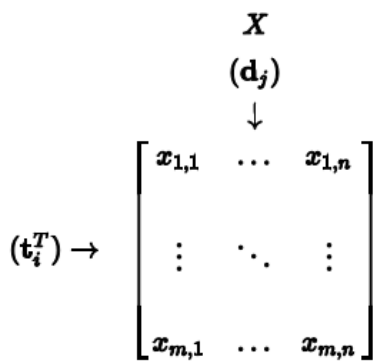
- Goal: represent symbolic data in a continuous space
  - easily measure relatedness (distance measures on vectors)
  - Linear Algebra and DL to perform complex reasonng
- Challenges
  - discrete nature, easy to count, but not obvious how to represent
  - cannot use backprop (no gradients) on discrete units
  - embedded continuous space can be potentially very large (high-dimensional vector spaces)
  - data not associated to a regular grid structure like image (e.g., text)

Learning word representations

- learning word representations from raw text (without any supervision)
- applications:
  - text classification
  - ranking (e.g., google search)
  - machine translation
  - chatbot

Latent Semantic Analysis [2]

- Problem: Find similar documents in a corpus
- Solution:
  - construct term/document matrix - (normalized) occurrence counts



term-document matrix

Example
doc1: the cat is furry
doc2: dogs are furry

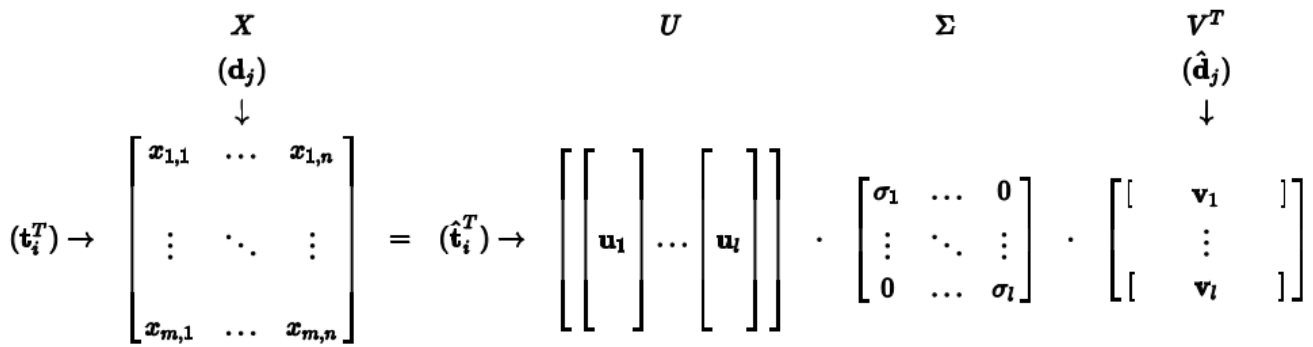
	doc1	doc2
are	0	1
cat	1	0
dogs	0	1
furry	1	1
is	1	0
the	1	0

(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

[2] Deerwester et al. "Indexing by Latent Semantic Analysis", JASIS, 1990

Latent Semantic Analysis (contd.)

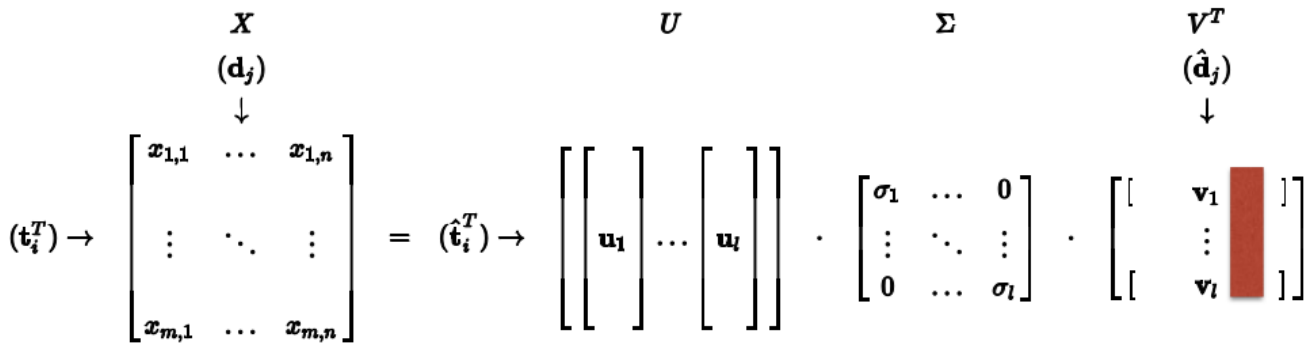
- Problem: Find similar documents in a corpus
- Solution:
  - construct term/document matrix - (normalized) occurrence counts
  - perform SVD (singular value decomposition)
  - $x_{i,j}$ # times word i appears in document j



(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Latent Semantic Analysis (contd.)

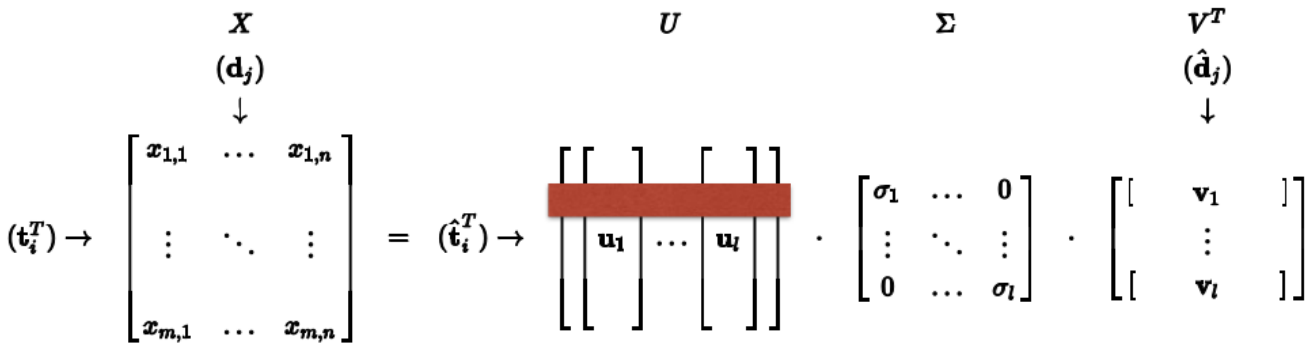
- Each column of V^T - representation of a document in the corpus
- Each column is D-dimensional vector
  - we can use it to compare and retrieve documents
  - $x_{i,j}$ # times word i appears in document j



(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Latent Semantic Analysis (contd.)

- Each row of U - vectorial representation of a word in the dictionary
  - aka **embedding**
  - $x_{i,j}$ # times word i appears in document j



Word embeddings

- Convert words (symbols) into a D dimensional vector
  - D becomes a hyperparameter
- In the embedded space
  - compare words (compare vectors)
  - apply machine learning (DL) to represent sequence of words
  - use weighted sum of embeddings (linear combination of vectors) for document retrieval

Bi-gram

- Bi-gram models probability of a word given the preceding one

$$p(w_k|w_{k-1}), w_k \in V$$

- Bi-gram model builds a matrix of counts

$$c(w_k|w_{k-1}) = \begin{bmatrix} c_{1,1} \dots c_{1,|V|} \\ \dots c_{i,j} \dots \\ c_{|V|,1} \dots c_{|V|,|V|} \end{bmatrix}$$

- $c_{i,j}$ number of times word i is preceded by word j

Factorized bi-gram

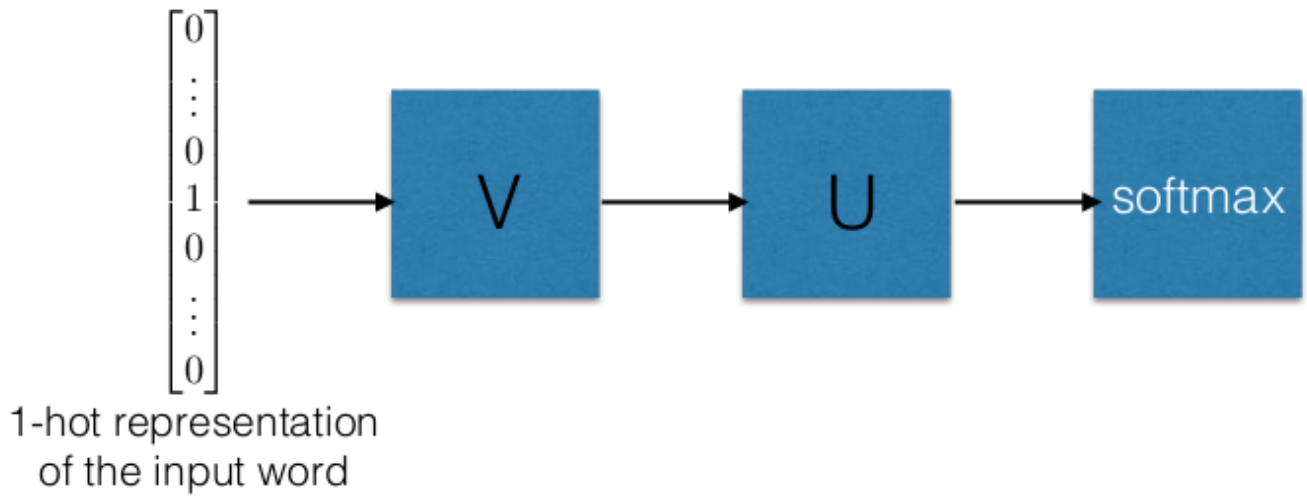
- factorize (via SVD) the bigram matrix
  - reduce # of parameters to learn
  - become more robus to noise (entries with low counts)

$$c(w_k|w_{k-1}) = \begin{bmatrix} c_{1,1} \dots c_{1,|V|} \\ \dots c_{i,j} \dots \\ c_{|V|,1} \dots c_{|V|,|V|} \end{bmatrix} = UV$$

- rows of $U \in \mathbb{R}^{|V| \times D}$ - "output" word embeddings
- columns of $V \in \mathbb{R}^{D \times |V|}$ - "input" word embeddings

Factorize bi-gram with NN

- we could express UV factorization with a two layer (linear) neural network



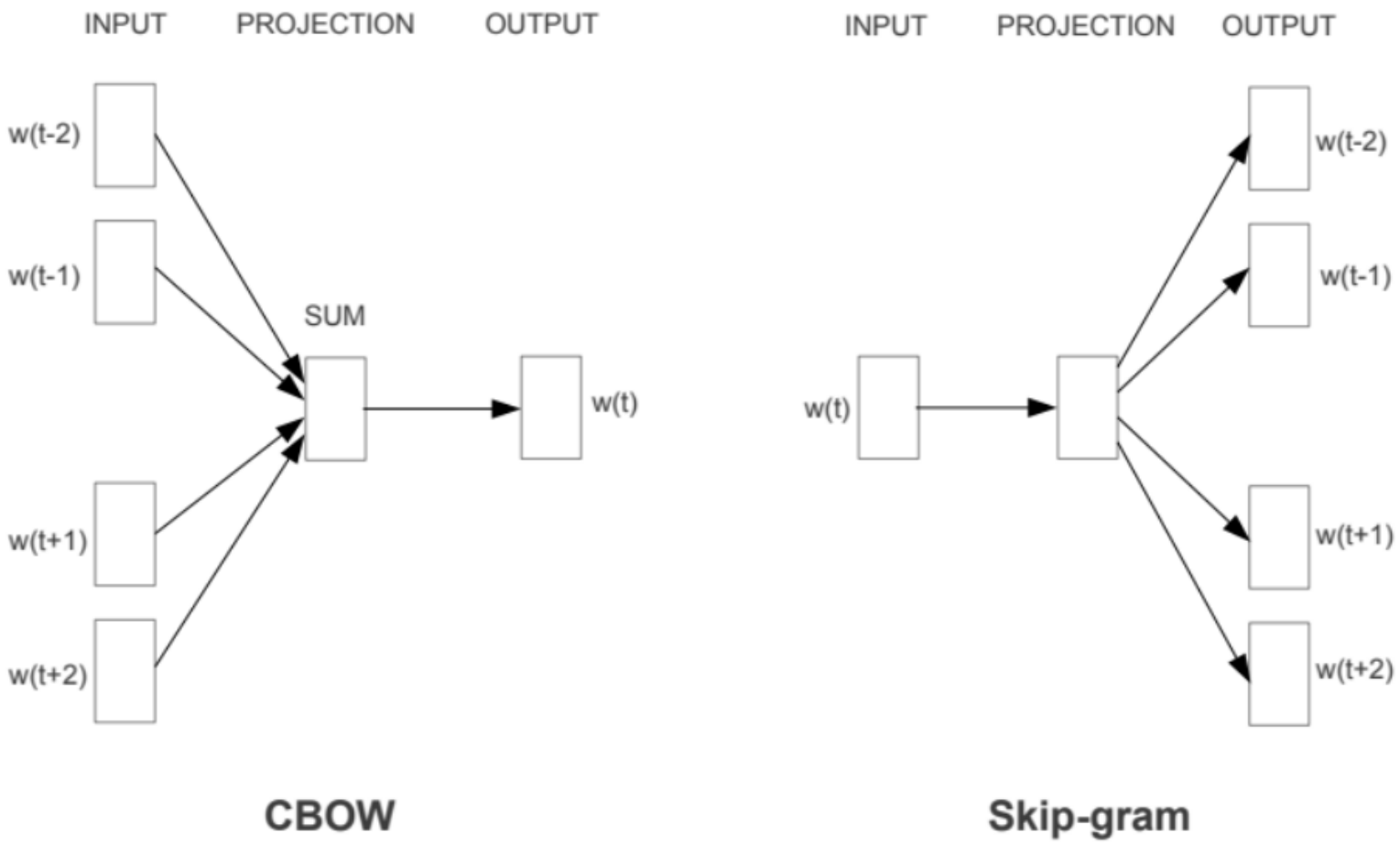
(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Recap

- LSA learns word embeddings via co-occurences across documents
- bi-gram learns word embeddings that only take into account next word
- if you combine the two you get **word2vec** [3]
  - use more context around the word ($n \geq 2$)
  - but look for context only around the word of interest

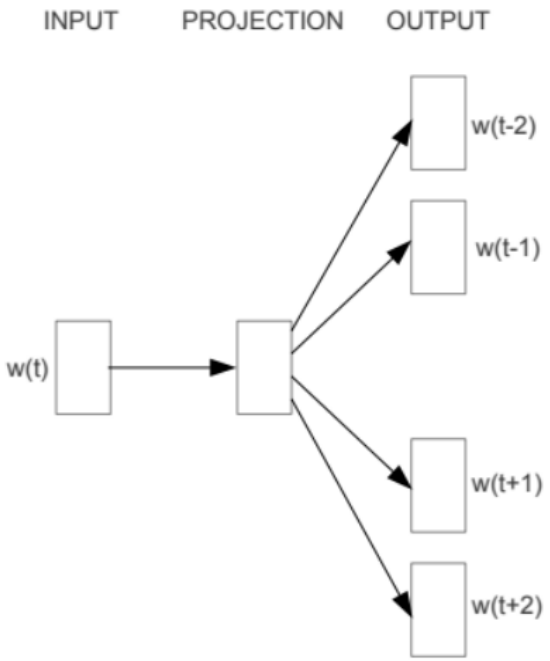
[3] "Efficient estimation of word representations", Mikolov et al. NIPS 2013

Word2Vec: Continuous bag of words



(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Word2Vec: Skip-gram



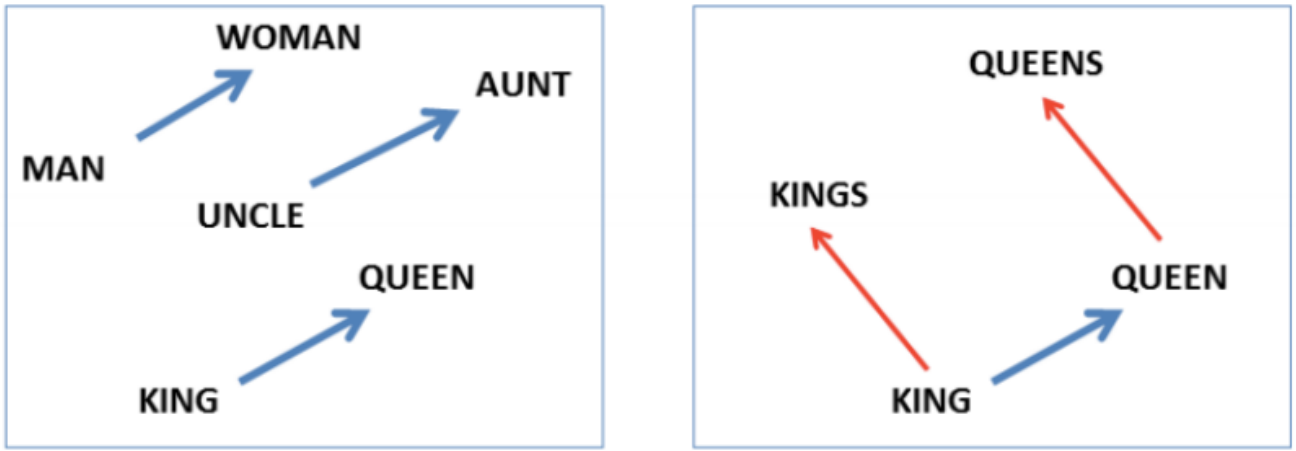
Skip-gram

- similar to *bi-gram*, but predict N preceding and N following words
- words with same context will have similar embedding (e.g., cat & kitty)
- input projection - look-up table
- bulk of computation - prediction of words in context
- learning by cross-entropy minimization via SGD

(Image credit) Course notes "An introduction to Deep Learning", Marc'Aurelio Ranzato

Word2Vec: Linguistic regularities in Word Vector Space

- word vector space implictly encodes linguistic regularities



- distributed word representations implicitly contain syntactic and semantic information
  - **KING** is similar to **QUEEN** as **MAN** is similar to **WOMAN**
  - **KINGS** is similar to **KINGS** as **MAN** is similar to **MEN**

(Image credit) "Efficient estimation of word representations", Mikolov et al. NIPS 2013

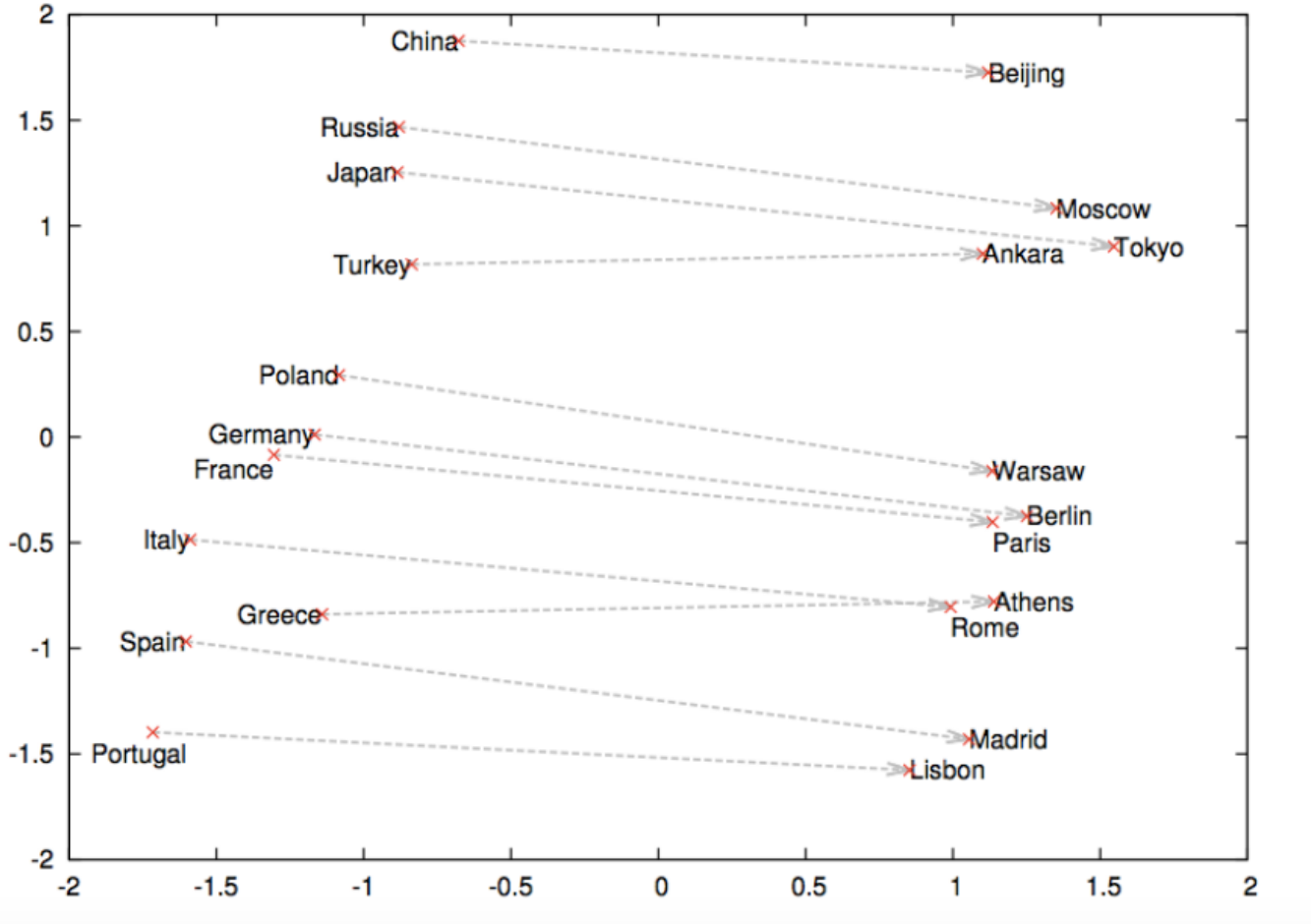
Word2Vec: Vector operations in Word Vector Space

- vector operations (addition/subtraction) give intuitive results

Expression	*Nearest token*
Paris - France + Italy	Rome
bigger - big + cold	colder
sushi - Japan + Germany	bratwurst
Cu - copper + gold	Au
Windows - Microsoft + Google	Android
Montreal Canadiens - Montreal + Toronto	Toronto Maple Leafs

(Image credit) "Efficient estimation of word representations", Mikolov et al. NIPS 2013

Word2Vec: Linguistic regularities in Word Vector Space (contd.)



(Image credit) "Efficient estimation of word representations", Mikolov et al. NIPS 2013

Recap

- Embeddings of words (from 1-hot to distributed representation):
  - understand similarity between words
  - plug them within any parametric ML model
- Several ways to learn word embeddings
  - **word2vec** among the most popular ones
- **word2vec** leverages large amounts of *unlabeled* data