The Emergence of Machine Learning as a Rupture technology for Artificial Intelligence

James L. Crowley
Professor Emeritus, Grenoble INP
Grenoble Informatics Laboratory (LIG)
INRIA Grenoble Rhone-Alpes
Univ. Grenoble Alpes

The Emergence of Machine Learning as a Rupture technology for Artificial Intelligence

A scientific community devoted to Artificial Intelligence (AI) was created in the 1950s.

After a euphoric period in the 1980s, AI was declared “dead”.

Since 2010, the popular media increasingly claim that we are in an AI revolution.

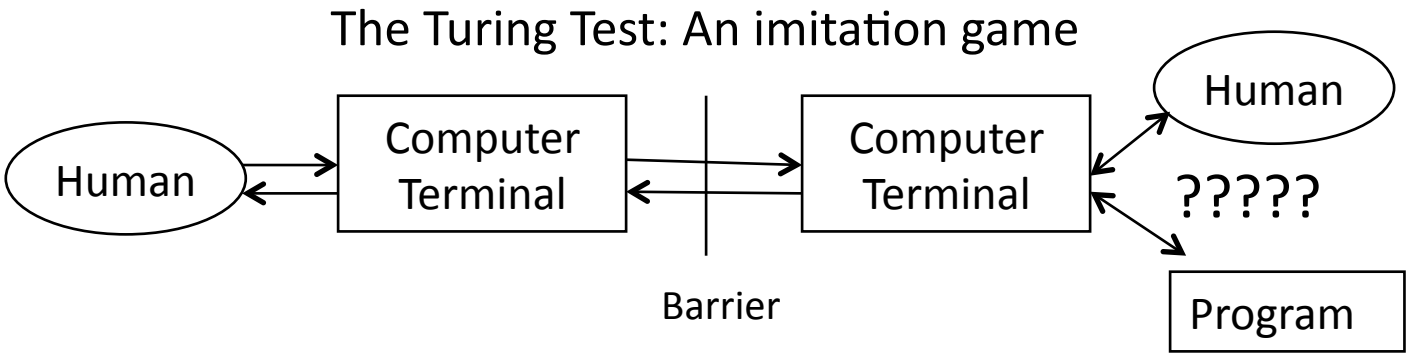
What changed between 1980 and 2010?

The Emergence of Machine Learning as a Rupture technology for Artificial Intelligence

Outline:

- History of Paradigms for Artificial Intelligence
- Perceptrons, Neural Networks and Back-Propagation
- Convolutional Networks and Deep Learning
- Generative Networks and auto-encoders
- Transformers and Self-Supervised Learning
- What happens next?

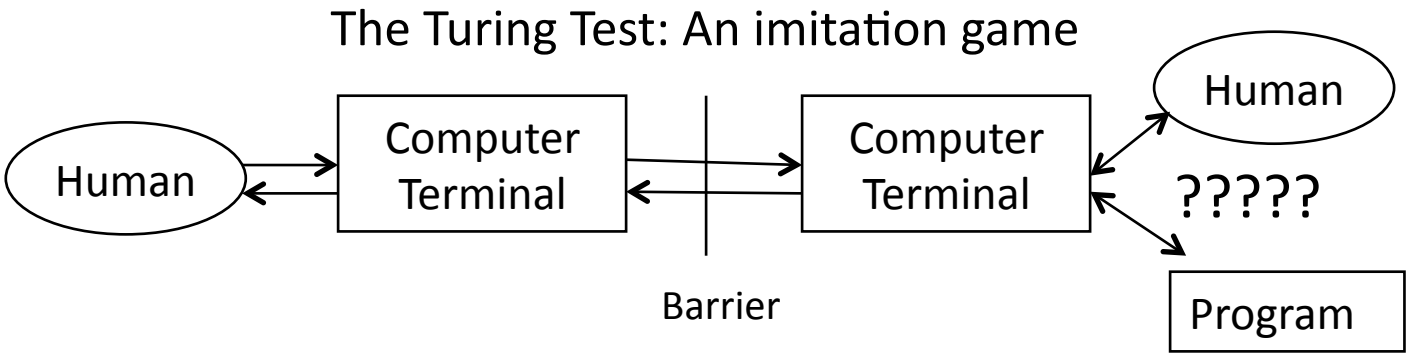
Artificial Intelligence (AI)



Intelligence according to Turing:
Human-level performance at (text-based) interaction.

The Turing Test: If a human cannot reliably discriminate between a machine and a human using text-based interaction then the machine is said to to be intelligent.

Artificial Intelligence (AI)



Modern technologies allow us to extend Turing's definition to tasks requiring perception, action, communication or interaction.

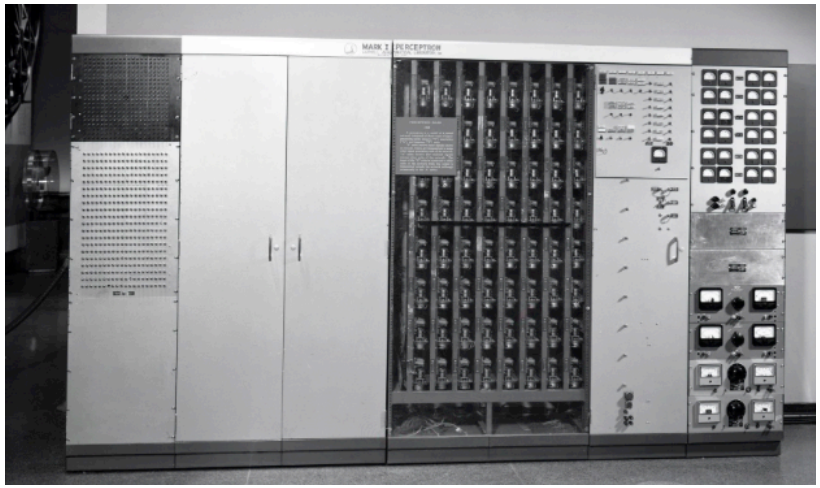
Intelligence: Human-level performance at tasks requiring perception, action, communication or interaction.

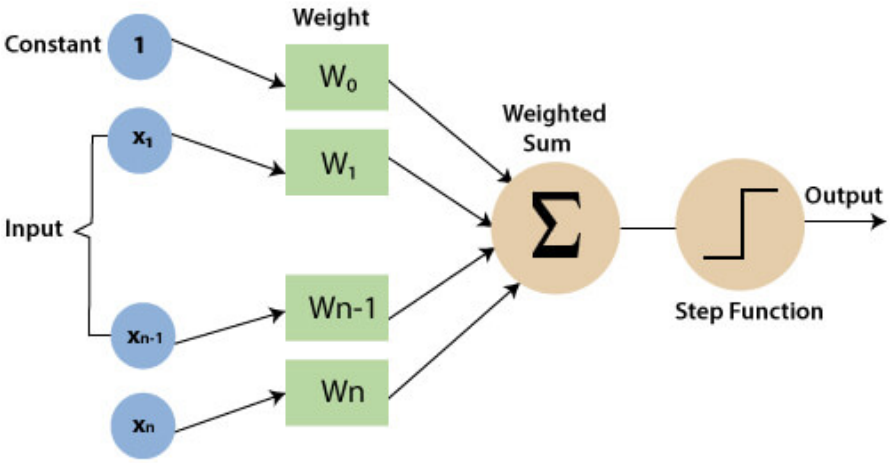
AI as a Modern Scientific Discipline

AI Pioneers at the Dartmouth Symposium (1956)

The modern scientific domain emerged in the 1960s as a convergence of Cognitive Science, Logic, Planning, Pattern Recognition, Image Processing and other fields, driven by the emergence of Computer Science.

Rosenblatt's Perceptron (1958)





Perceptron: Learning algorithm for a linear decision surface.

Problems: (1) Could only classify patterns
(2) Required labeled training data for supervised learning.
(3) Required linearly separable properties for classes.

If the training data was not linearly separable, the algorithm would not terminate

Evolution of Artificial Intelligence

From Pattern Recognition to Deep Learning

Dominant Paradigms for Artificial Intelligence:

- Pre-1960: Automata and Pattern Recognition
- 1960-1985: Planning, problem solving
- 1975-1990: Expert systems, symbolic reasoning
- 1985-2000: Logic programming, theorem proving
- 1995-2010: Bayesian methods, Semantic Web

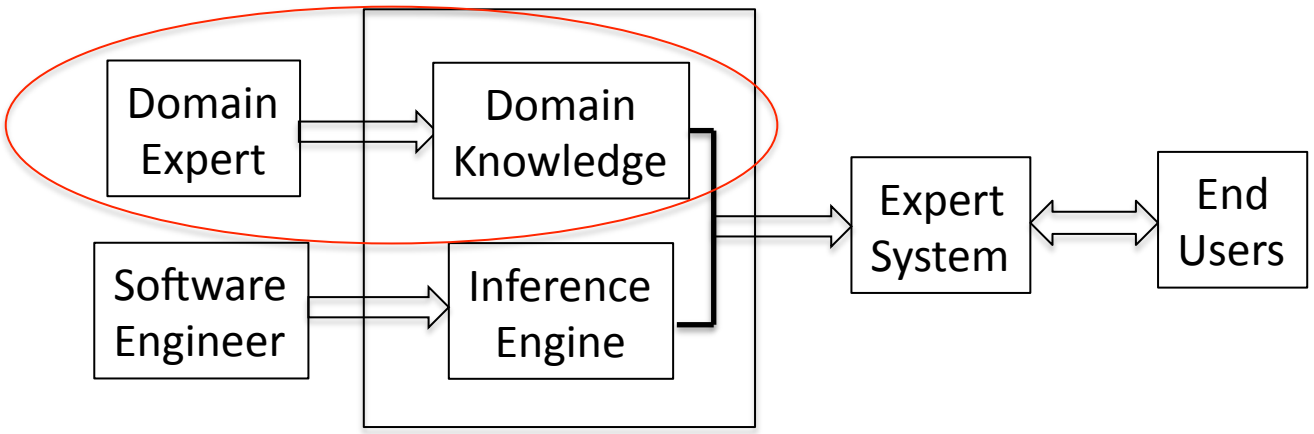
Three Fundamental Barriers to AI:

(1) Insufficient Labeled Data for Supervised Learning.

(2) Insufficient Computing Power.

(3) Prohibitive Cost of Encoding Domain Knowledge.

Expert System Design Process (1980)

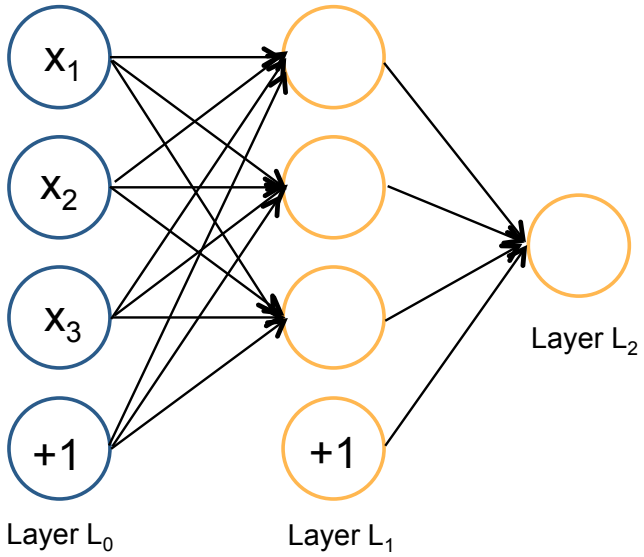


Example: MYCIN – Antibiotic Therapy Advisor (Feigenbaum et al 1980).
Domain expert worked with Software Engineer to build system.

Fundamental Problem:

Prohibitive cost of generating domain knowledge.

Artificial Neural Networks (1975-1990)
Multi-layer Perceptrons with Learning using Back-propagation



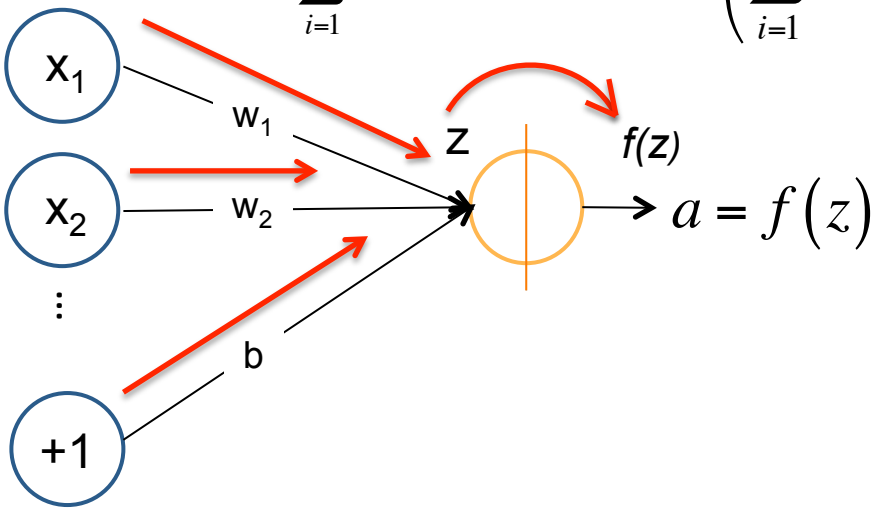
Artificial Neural Networks (1975-1990) – Two innovations

1) Multi-layer perceptrons with soft decision surface
2) Learning with Back-Propagation (distributed gradient descent).

Provided a simple alternative to symbolic computing

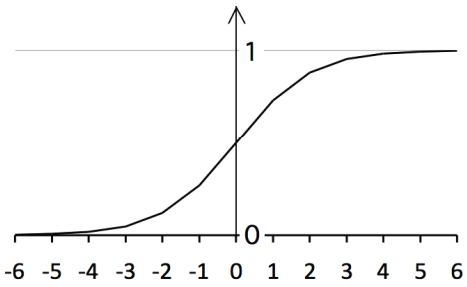
Artificial Neural Networks

$$z = \sum_{i=1}^{N} w_i x_i + b \qquad a = f\left(\sum_{i=1}^{N} w_i x_i + b\right)$$



Decision: Sigmoid function

$$f(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$



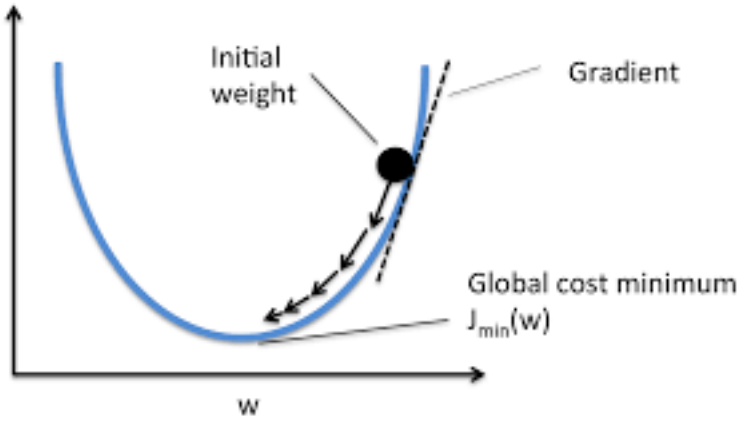
$$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z))$$

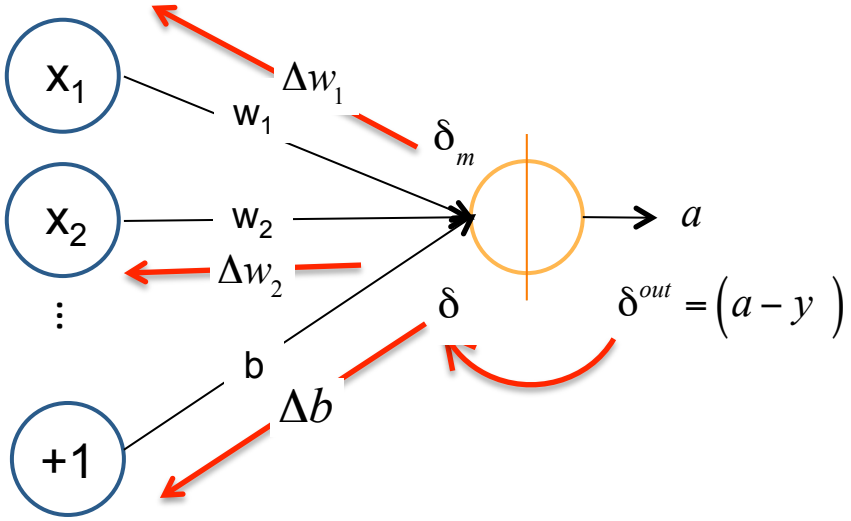
Important Innovation in the 1970's: Soft decision function.
A soft (differentiable) decision function makes it possible to learn from errors using Gradient Descent.

Back-propagation is Gradient Descent

Training Data: M samples $\{\vec{X}_m\}$ labeled with indicator Variables $\{y_m\}$

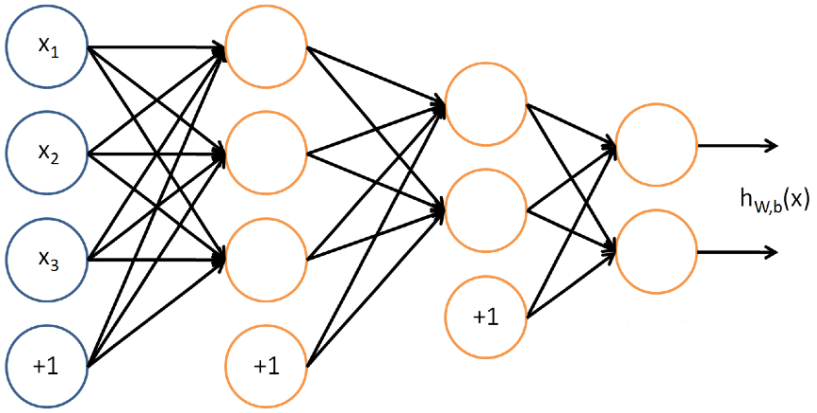
$$C(\vec{X}_m, \vec{w}) = \frac{1}{2}(a_m^{out} - y_m)^2$$





Gradient descent: A first-order iterative optimization algorithm for finding the minimum of a function.
Used to determine the best weights and bias.
Scalable to any quantity of training data.

Generalized to Multi-Layer Networks



Recursive Feed-Forward calculation

$$\vec{a}^{(3)} = f(f(....f(w_{ij}^{(1)}\vec{X}_i + b_j^{(1)})))$$

Hebbian representation:
propagation of activation energy

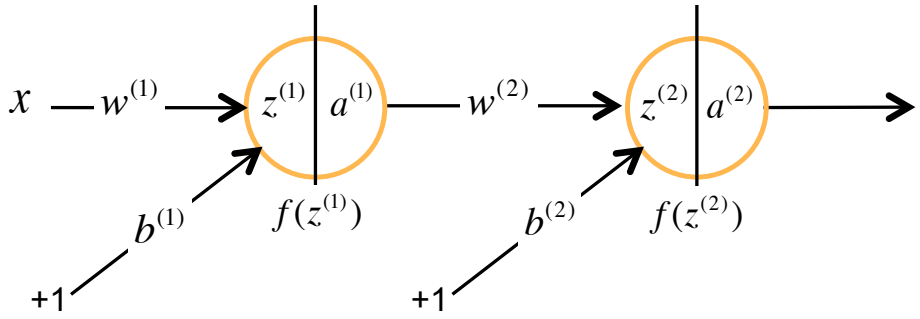
$$a_j^{(l)} = f\left(\sum_{i=1}^{N^{(l-1)}} w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$$

A Neural Network is a distributed algorithm using **propagation of activation energy**, enabling arbitrary scale networks using **SIMD** parallel computing.

Network Training with Gradient Descent

Consider a 2-layer network with one unit at each layer.

Network has 4 parameters: $\vec{w} = (w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)})$



$$a^{(2)} = f(w^{(2)} f(w^{(1)} X + b^{(1)}) + b^{(2)})$$

Cost of using *w* to estimate *y* from *X is* : $C = \frac{1}{2}\left(a^{(2)} - y\right)^2$

$$\vec{\nabla} C = \frac{\partial C}{\partial \vec{w}} = \begin{pmatrix} \frac{\partial C}{\partial w^{(1)}} \\ \frac{\partial C}{\partial b^{(1)}} \\ \frac{\partial C}{\partial w^{(2)}} \\ \frac{\partial C}{\partial b^{(2)}} \end{pmatrix} = \begin{pmatrix} \Delta w^{(1)} \\ \Delta b^{(1)} \\ \Delta w^{(2)} \\ \Delta b^{(2)} \end{pmatrix}$$

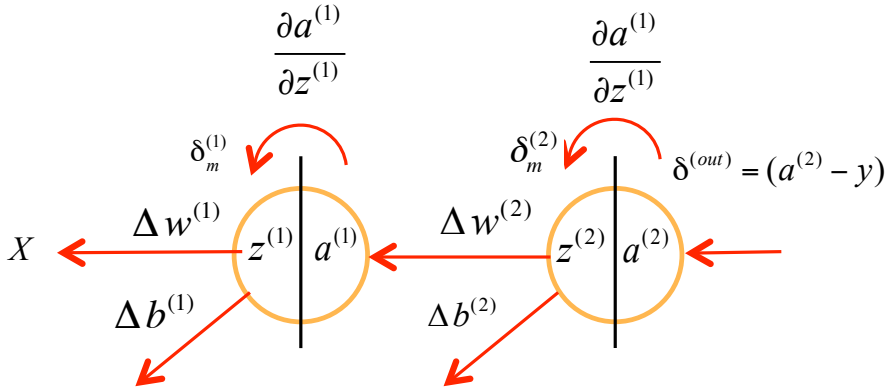
The gradient of the cost with respect to the parameters, ∇ C tells how much to correct each parameter.

The derivatives are computed with the chain rule:

$$\Delta w^{(1)} = \frac{\partial C}{\partial w^{(1)}} = \frac{\partial C}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial a^{(1)}} \cdot \frac{\partial a^{(1)}}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial w^{(1)}}$$

Back-propagation expresses the chain rule as a **backward flow** of **correction energy** (SIMD)

The Gradient tells how much to correct each parameter to minimize the cost.



$$\nabla C = \frac{\partial C}{\partial \vec{w}} = \begin{pmatrix} \frac{\partial C}{\partial w^{(1)}} \\ \frac{\partial C}{\partial b^{(1)}} \\ \frac{\partial C}{\partial w^{(2)}} \\ \frac{\partial C}{\partial b^{(2)}} \end{pmatrix} = \begin{pmatrix} \Delta w^{(1)} \\ \Delta b^{(1)} \\ \Delta w^{(2)} \\ \Delta b^{(2)} \end{pmatrix}$$

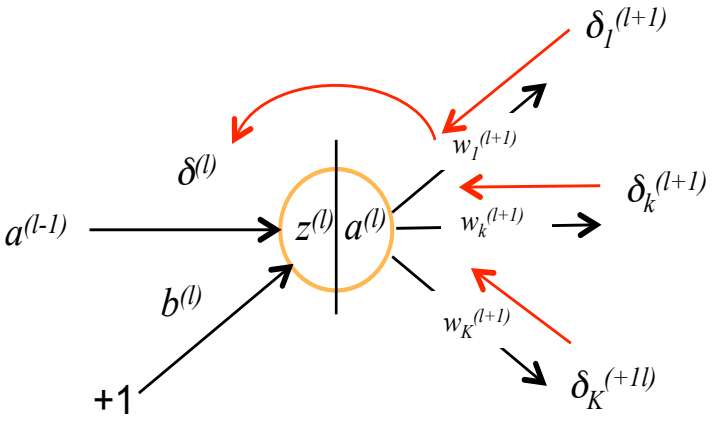
The derivatives are computed with the chain rule: $\Delta w^{(1)} = \frac{\partial C}{\partial w^{(1)}} = \frac{\partial C}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial a^{(1)}} \cdot \frac{\partial a^{(1)}}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial w^{(1)}}$

Back-propagation expresses the chain rule as backward flow of **correction energy, δ** :

$$\Delta w^{(1)} = \delta^{(1)} \cdot X = \left(\delta^{(2)} \cdot w^{(2)} \cdot \frac{\partial a^{(1)}}{\partial z^{(1)}} \right) \cdot \frac{\partial z^{(1)}}{\partial w^{(1)}} \qquad X$$

$$\Delta b^{(1)} = \delta^{(1)} = \left(\delta^{(2)} \cdot w^{(2)} \cdot \frac{\partial a^{(1)}}{\partial z^{(1)}} \right) \cdot \frac{\partial z^{(1)}}{\partial b^{(1)}} \qquad 1$$

Generalized to Multi-Layer Networks



Back-Propagation computes the correction terms as a backward flow of correction energy. This is a parallel (scalable) SIMD algorithm.

Correction energy for unit *j* of layer *l* computed recursively from level *l+1* by back propagating a correction energy

$$\delta_j^{(l)} = \frac{\partial f(z_j^{(l)})}{\partial z_j^{(l)}} \sum_{k=1}^{N^{(l+1)}} w_{jk}^{(l+1)} \delta_k^{(l+1)}$$

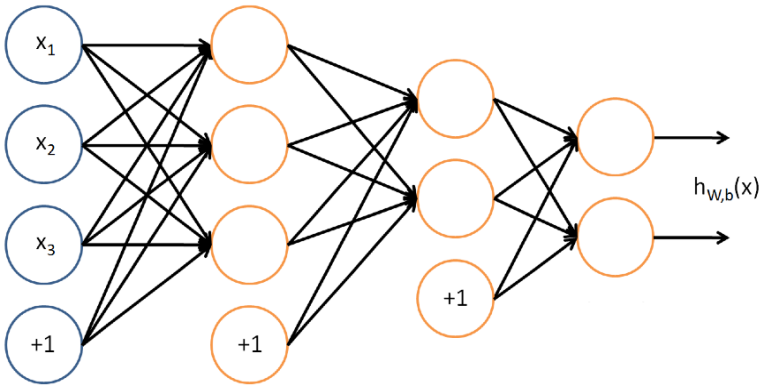
Correction of weight i of unit *j* at layer *(l)*

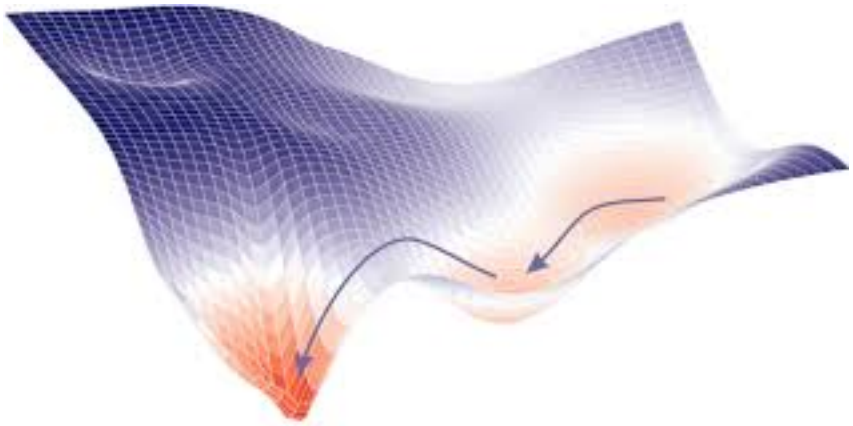
$$\Delta w_{ij}^{(l)} = a_i^{(l-1)} \delta_j^{(l)}$$

Correction for bias of unity *j* at layer *(l)*

$$\Delta b_j^{(l)} = \delta_j^{(l)}$$

Generalized to Multi-Layer Networks



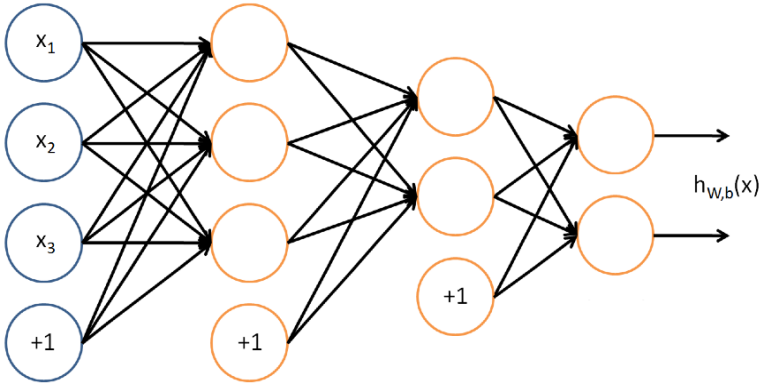


Training requires massive computing with massive data.

Difficulties: (1980's)

- Network has thousands of parameters
- Training data is very noisy.
- Loss function has local minima

Artificial Neural Networks (1975-1990)
Multi-layer Perceptrons with Back Propagation Learning



Problems:

1) Black Box (unexplainable, unpredictable behavior)
2) Difficult to reproduce
3) Cost of Learning (data and computation) grow exponentially with number of Layers

Neural networks were (mostly) abandoned in the 1990s in favor of mathematically sound Bayesian machine learning.

Three Fundamental Barriers to AI

(1) Insufficient training data
(2) Insufficient computing power
(3) Prohibitive cost of encoding domain knowledge

Enabling Technologies

Overcoming the three fundamental Barriers:

(1) Insufficient training data

$\Rightarrow$ Planetary scale data from the internet and the WWW

$\Rightarrow$ Data from realistic simulations

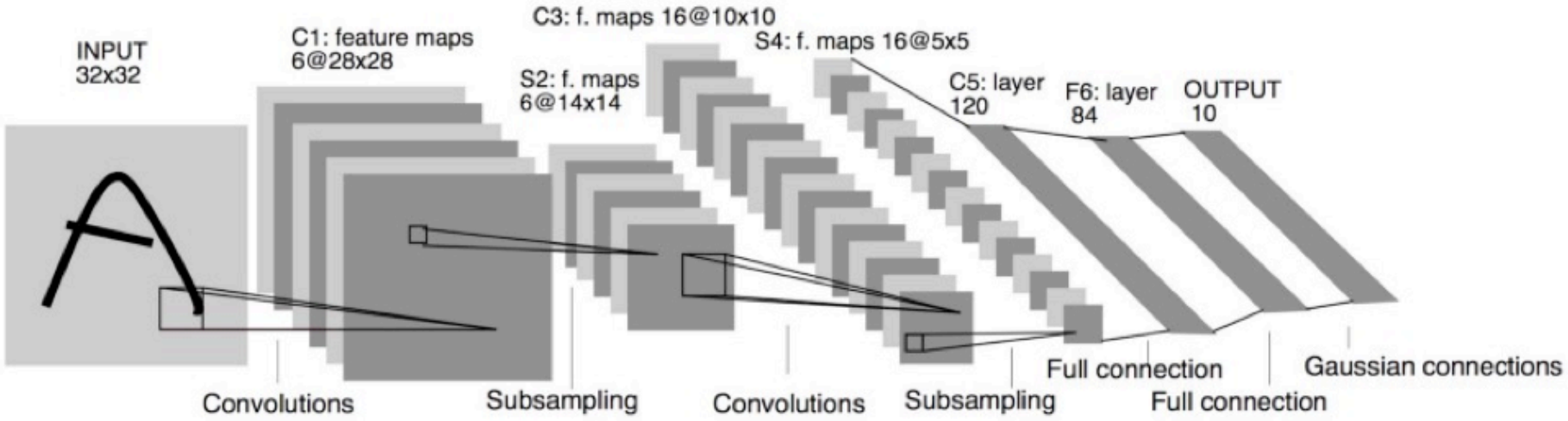
(2) Insufficient computing power

$\Rightarrow$ Moore's Law, GPUs, massively parallel computing

(3) Prohibitive cost of encoding knowledge

$\Rightarrow$ Generalized Deep Learning

Le Net5 - 1994

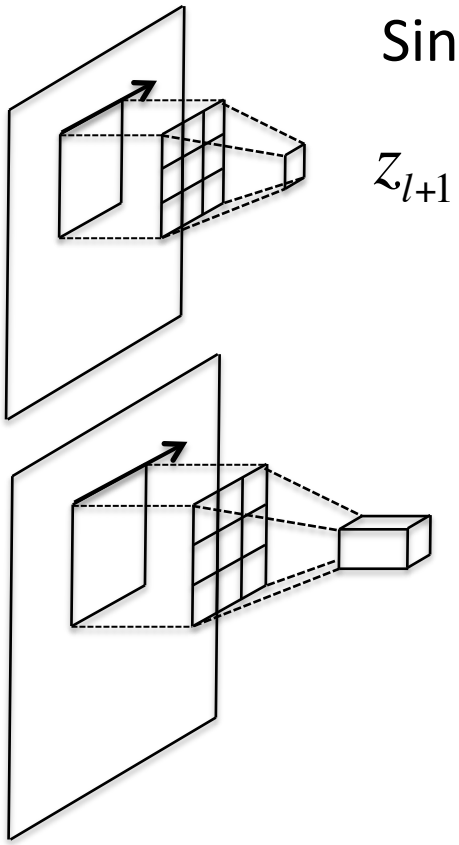


7-level convolutional network by Yann LeCun in 1998.
State of the art for recognizing hand-written numbers on checks.

Ignored by the Machine Learning and Computer Vision communities until around 2010.

Convolutional Neural Networks

Convolutional networks reduce the number of parameters to learn and increase the amount of training data.



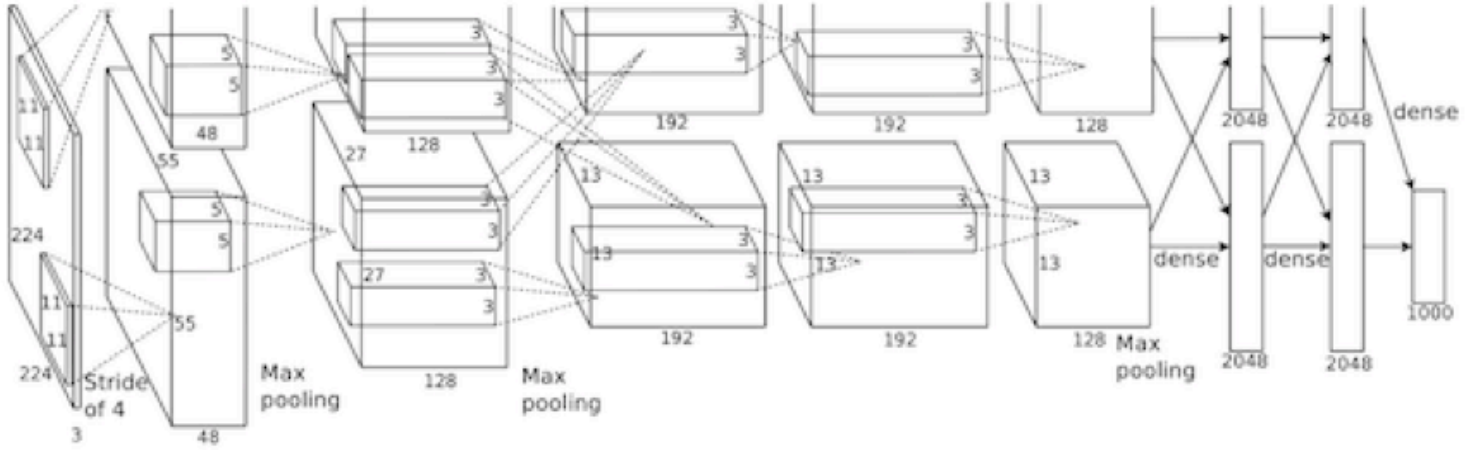
Single Receptive Field per layer

$$z_{l+1}(x,y) = \sum_{u,v} a_l(u-x, v-y) w_l(u,v)$$

Multiple Receptive Fields per layer

$$\vec{z}_{l+1}(x,y) = \sum_{u,v} \vec{a}_l(u-x, v-y) w_l(u,v)$$

AlexNet 2012

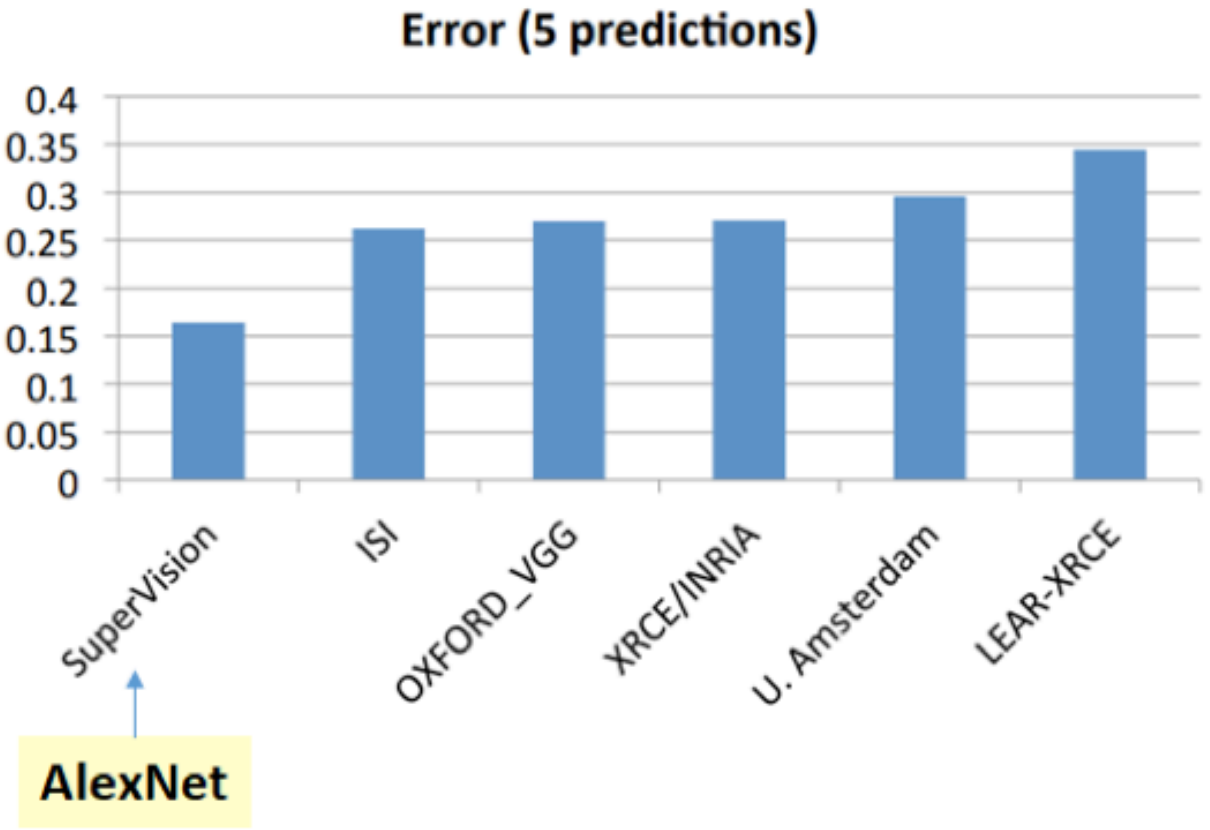


Created by Alex Krizhevsky and Geoff Hinton (based on LeNet)
Won the ImageNet Large Scale Visual Recognition Challenge in 2012 by a large margin with an error of around 15%

Triggered a paradigm shift for Computer Vision, Speech Recognition, Machine Learning and (more recently) Artificial Intelligence.

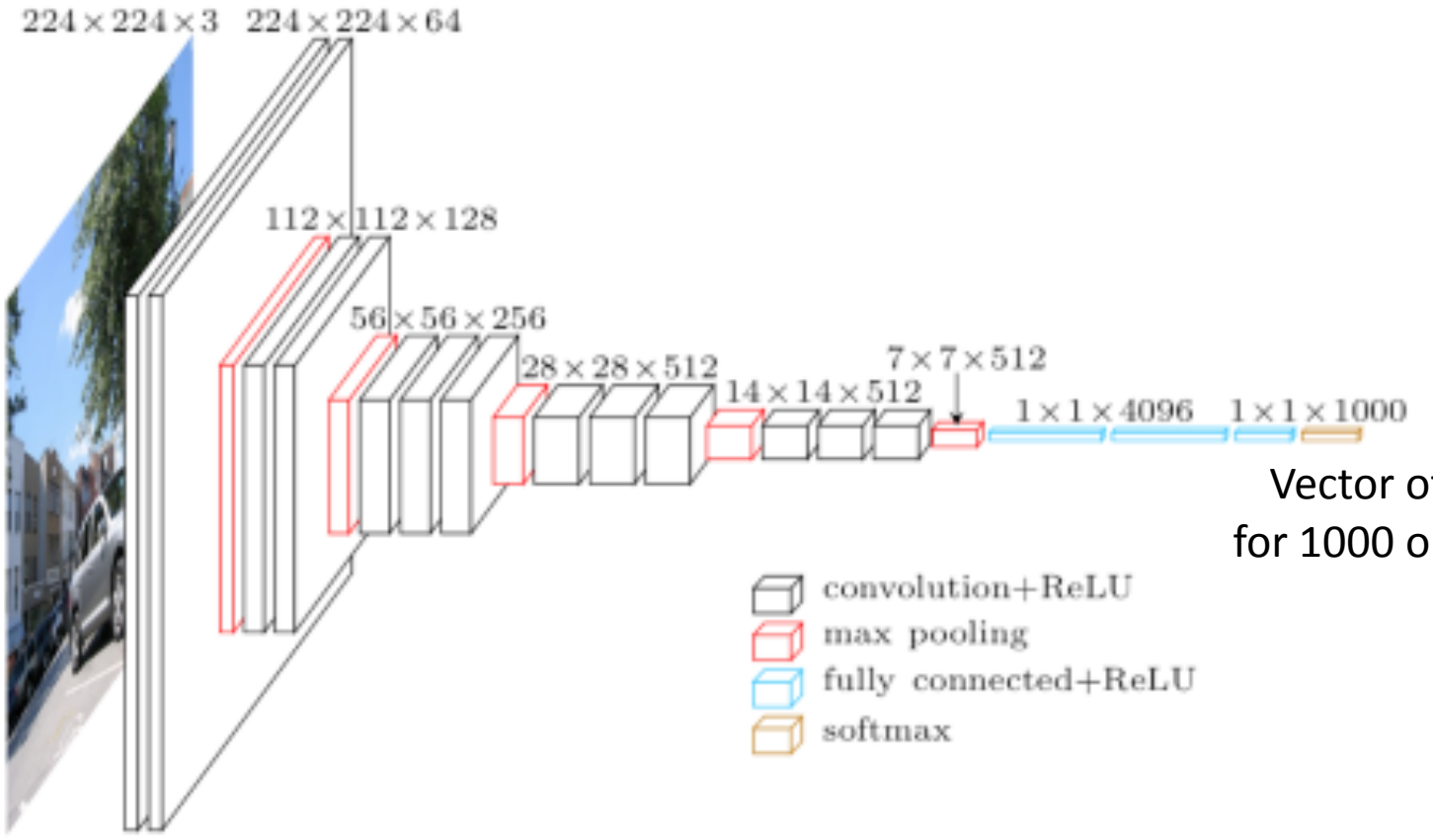
ImageNet Large Scale Visual Recognition Challenge in 2012

Ranking of the best results from each team



VGG 2015

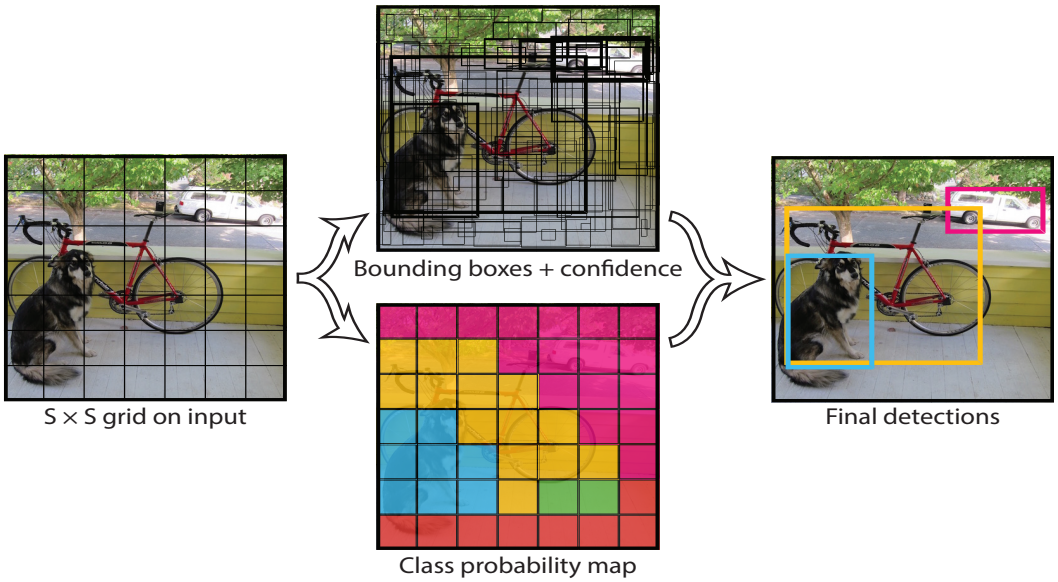
Karen Simonyan and Andrew Zisserman, Oxford **Visual Geometry Group**
Published at ICLR 2015, Available in Github, Tensorflow, Keras
Simple and effective workhorse for Transfer Learning



Output:
Vector of Likelihoods
for 1000 object classes

Yolo: You only look once (2016)

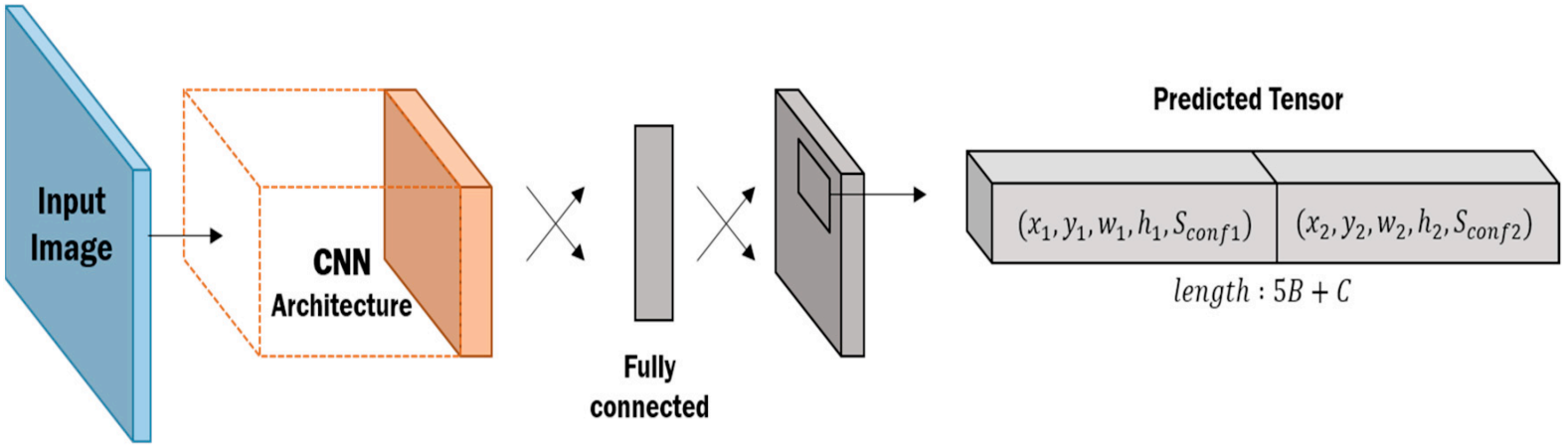
YOLO poses object detection as a single regression problem that estimates bounding box coordinates and class probabilities at the same time directly from image pixels.



Redmon, J., Divvala, , S., Girshick, R., et al. You only look once: Unified, real-time object detection. In : CVPR 2016. p. 779-788.

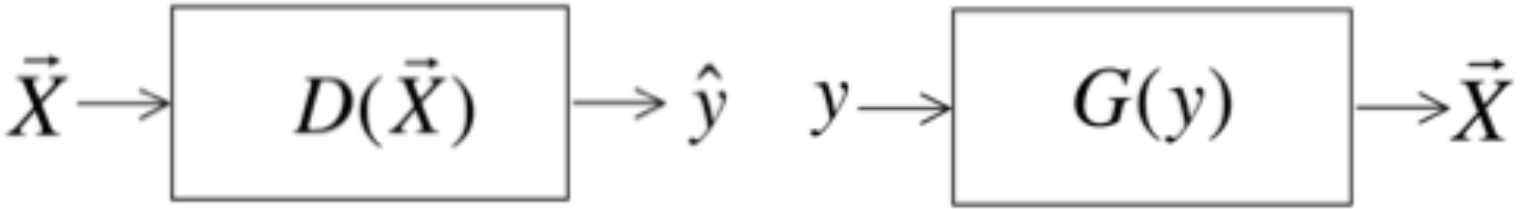
Yolo: You only look once (2016)

A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for each box.



From Kim, J. and Cho, J. Exploring a Multimodal Mixture-Of-YOLOs Framework for Advanced Real-Time Object Detection. Applied Sciences, 2020, vol. 10, no 2, p. 612.

Generative and Discriminative Networks



Discriminative Networks:
Does data X contain class y?

Generative Networks:
Generate pattern X for class y

Deep learning was originally invented for recognition.

The same technology can be used for generation.

Examples: Natural sounding speech

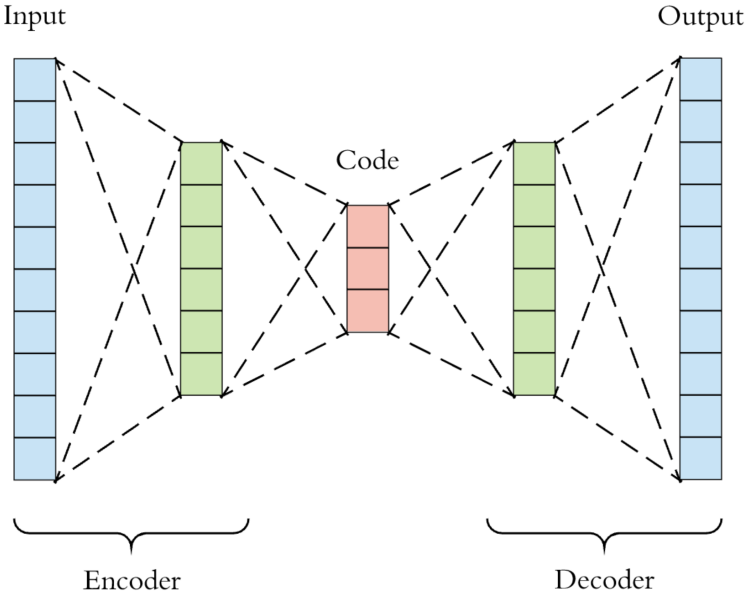
Natural Language

Synthetic images

Robot animation

Realistic talking heads (Deep Fake!)

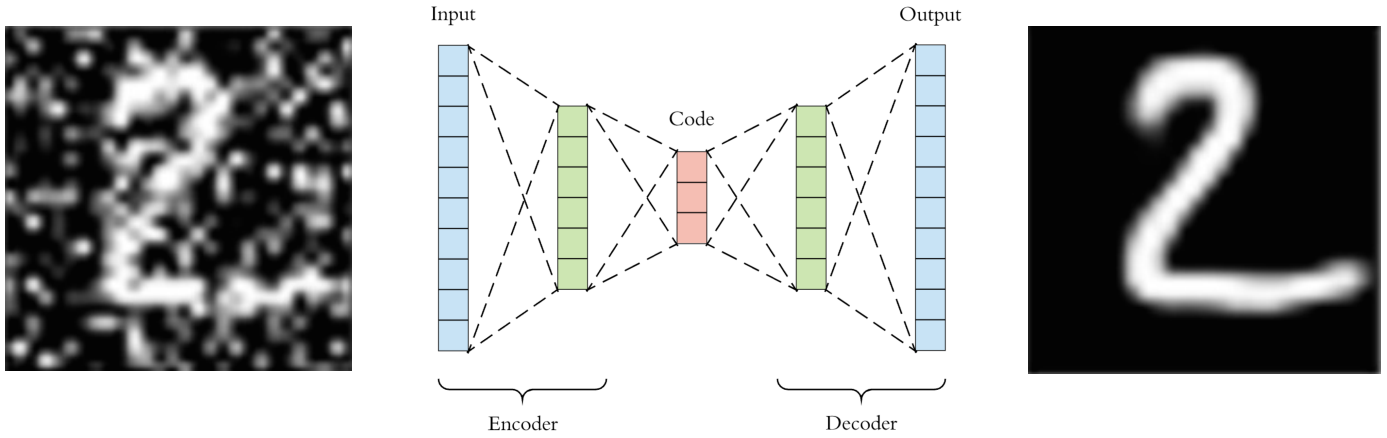
Autoencoder



An Autocoder learns to reconstruct (generate) clean copies of data without noise. Key concepts:

1) Training data is target. Error is difference between input and output
2) Scalable to any quantity of data using Back Propagation
3) Compresses the training data into a minimum number of independent hidden units (Code vector)

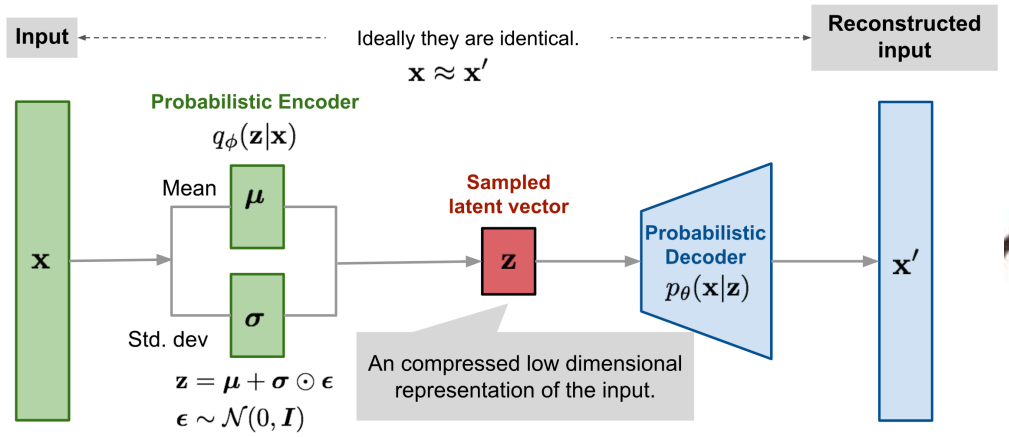
AutoEncoders



Autoencoders were originally used to compute **principal component analysis**, using least squares reconstruction error.

Adding a information theoretic “sparsity term” to the cost function provides **independent components analysis**, providing **unsupervised learning** of classes from data.

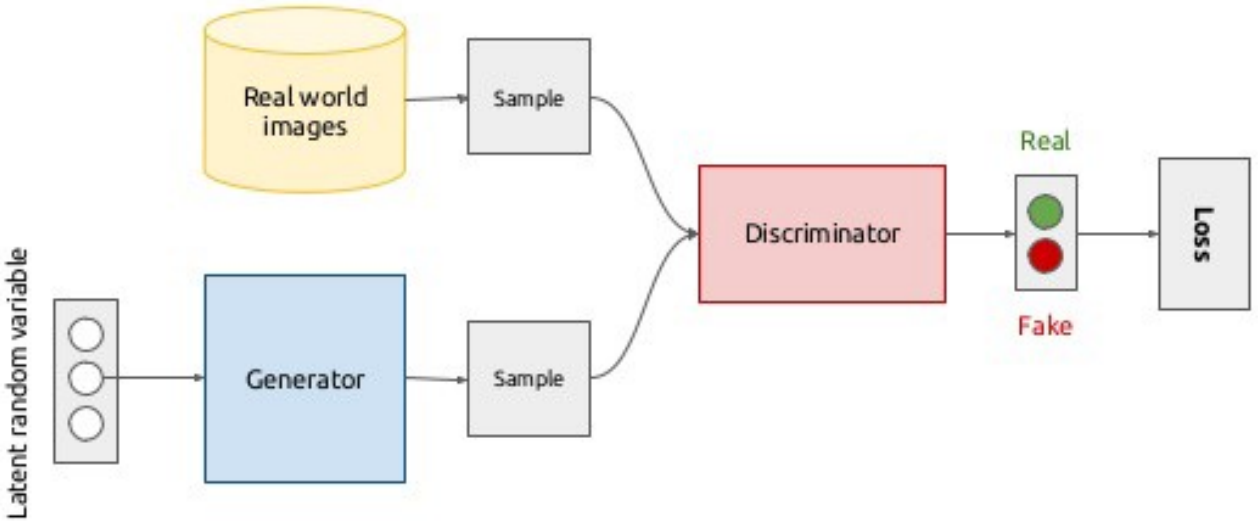
Variational Autoencoder



A VAE can be used to generate synthetic output.
Example:

1) Train VAE on dancers doing the same dance.
   => Code represents posture
2) Drive decoder of a dancer from encoder of another.

Generative Adversarial Networks

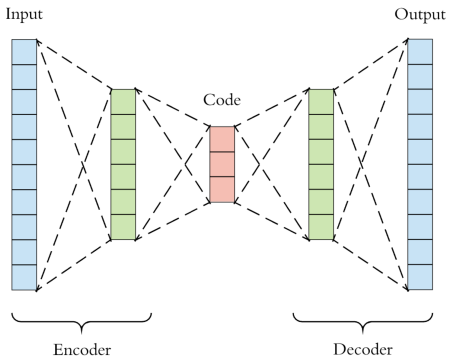


5

Unsupervised competitive learning between a Generative and a Discriminative network

Can be used to generate DeepFake, Realistic Speech synthesis, photo Realistic images (Hot topic at the IJCAI 2018)

Auto-encoders enable Self Supervised learning

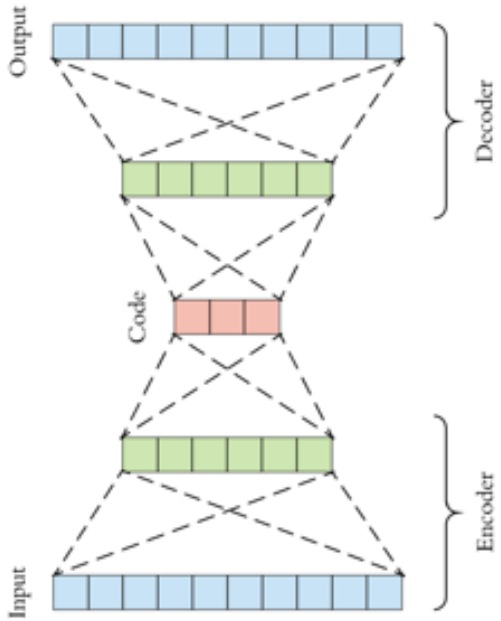


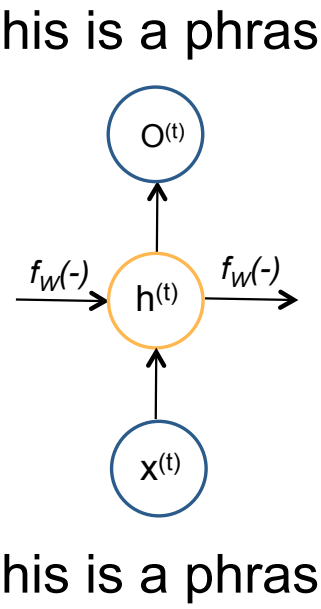
Self-supervised learning is a form of unsupervised learning reconstructing data. The data its the ground truth.

The system learns to reconstruct missing parts in the data (**missing token replacement**), and to predict the next data (**next token prediction**).

Self supervised learning can potentially unlock all recorded human knowledge to machine learning.

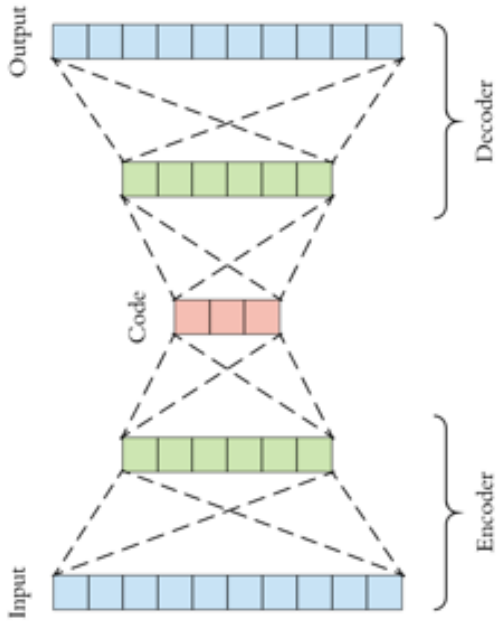
Auto-encoders enable Self Supervised learning

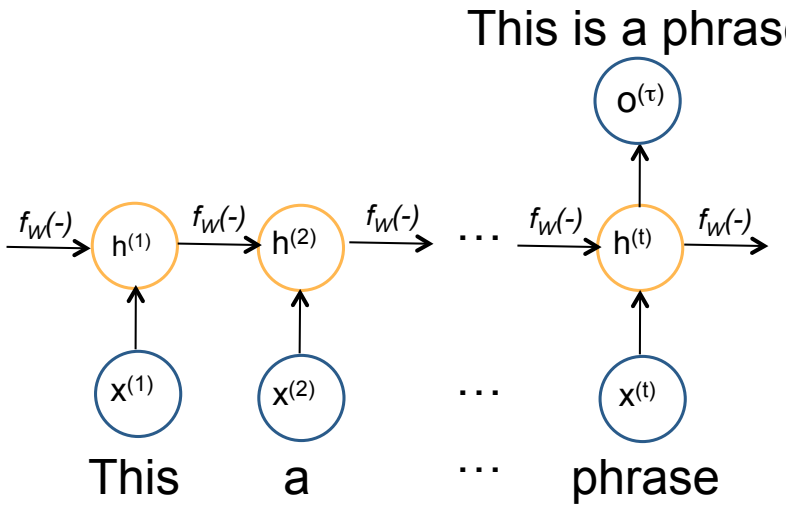




Auto-encoders can trained with missing token replacement be used to correct spelling errors, replace missing words, and improve grammar or style.

Recurrent Networks:

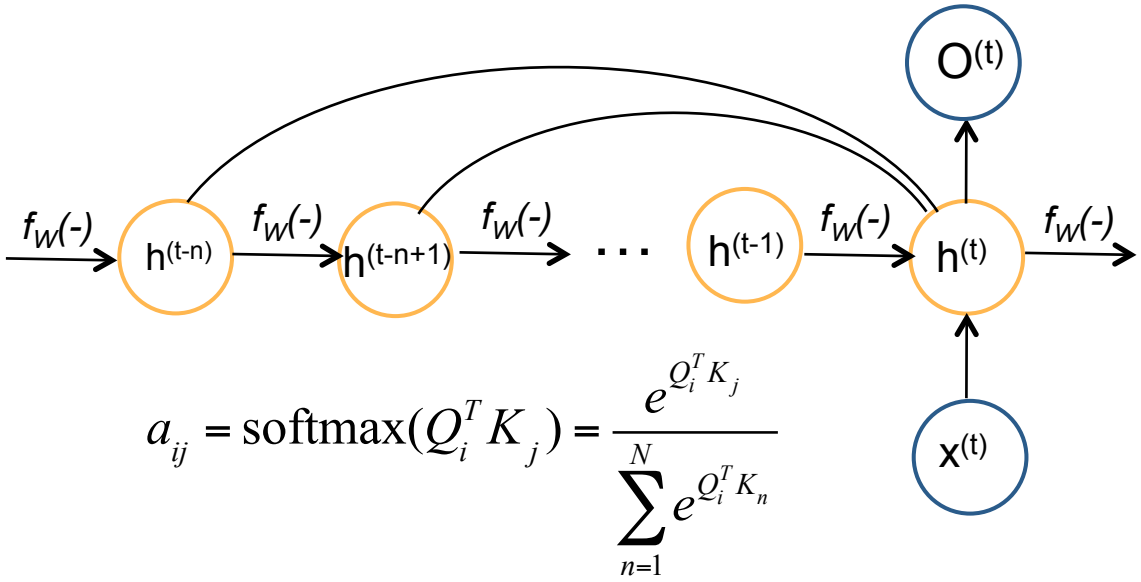




Auto encoders can also be used for natural language processing, by capturing the “meaning” of a word or sentence for example, decoding the meaning in different language.

Attention Extends Time for Recurrent Networks

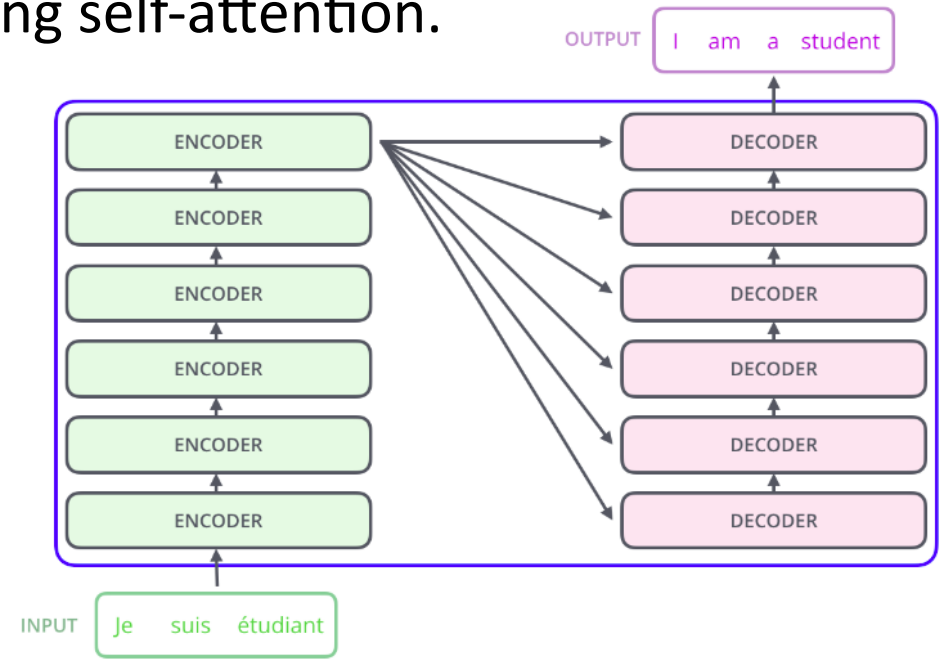
Attention was originally proposed as a soft search mechanism to extend the temporal range of Recurrent Networks (Bahdanau et al 2015).



D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In 3rd International Conference on Learning Representations, 2015

Self Supervised Learning with Transformers

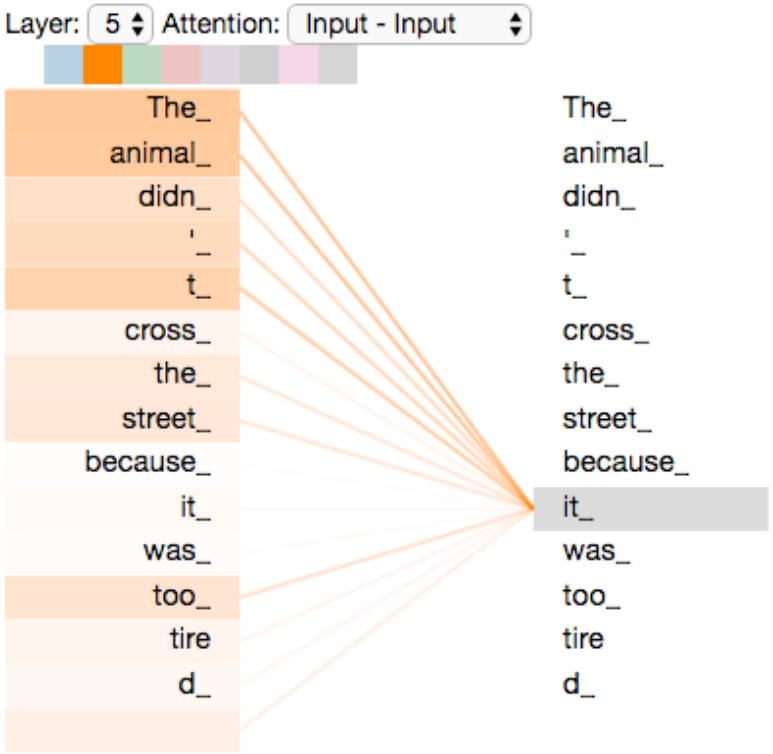
In 2017, a revolutionary paper by Vaswani et al [1] from Google showed that the deep convolutional and recurrent networks using layers of could be completely replaced with stacked auto-encoders using self-attention.



From Jay Alamar, The Illustrated Transformer: http://jalammar.github.io/illustrated-transformer/

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. . Attention is all you need. 2017

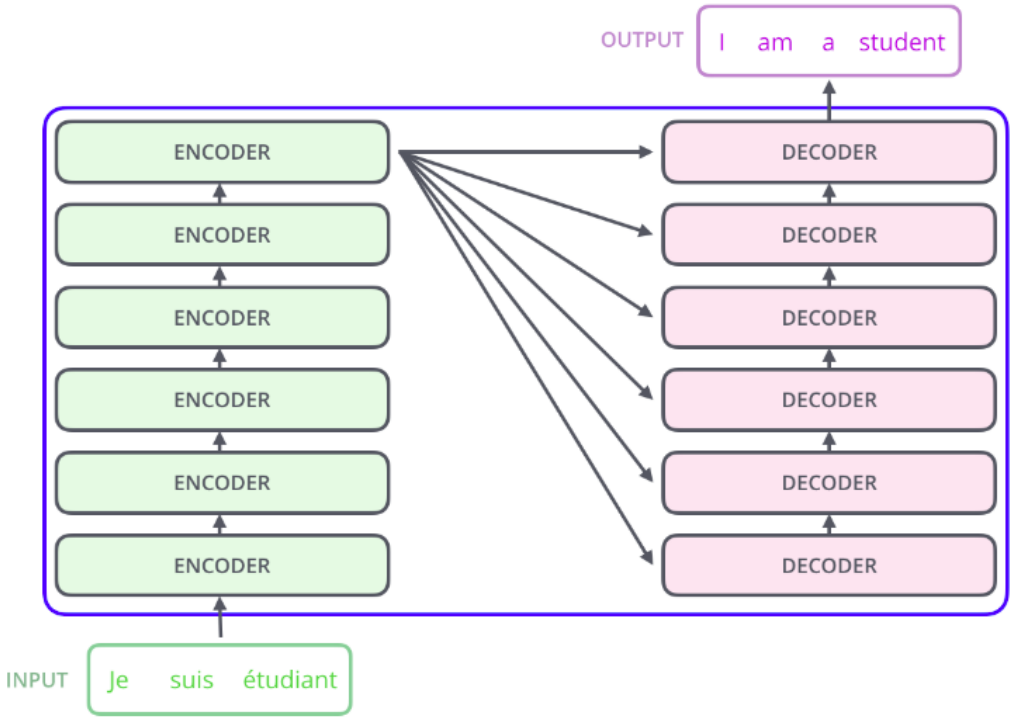
Transformers use attention to associate mutually relevant entities



Self-attention associates words in a sentence or paragraph in order to provide context for a more abstract representation and establish meaning.

Jay Alammar, The Illustrated Transformer (http://jalammar.github.io/illustrated-transformer/)

The Transformer Architecture



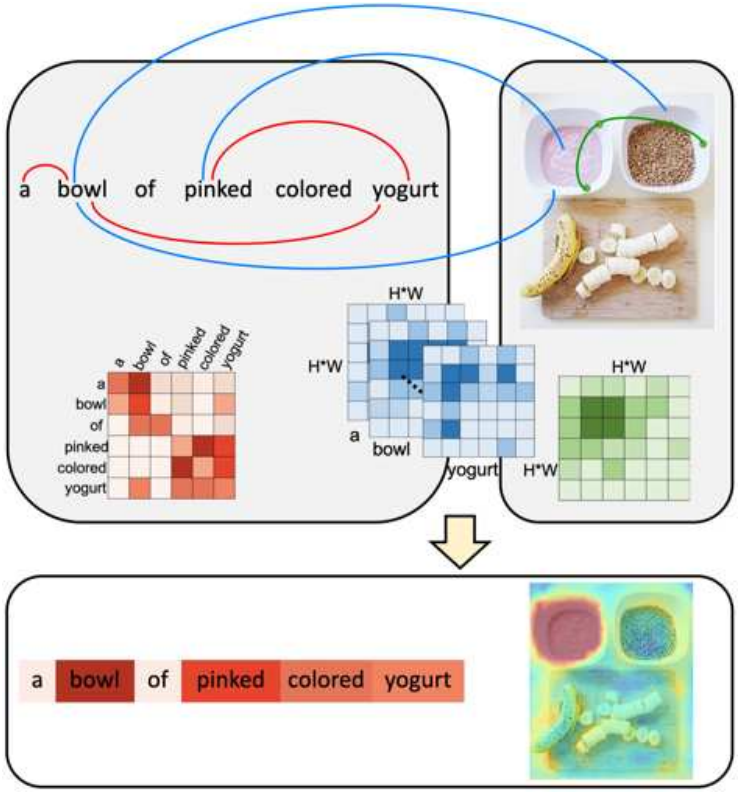
Transformers Stack multiple layers of Encoder-Decoders to describe a signal a multiple levels of abstraction.

At each layer, tokens are associated using **Self-Attention**.

Transformers have become the dominant approach for natural language processing (NLP).

Jay Alamar, The Illustrated Transformer: http://jalammar.github.io/illustrated-transformer/

Cross-Modal Self-Attention



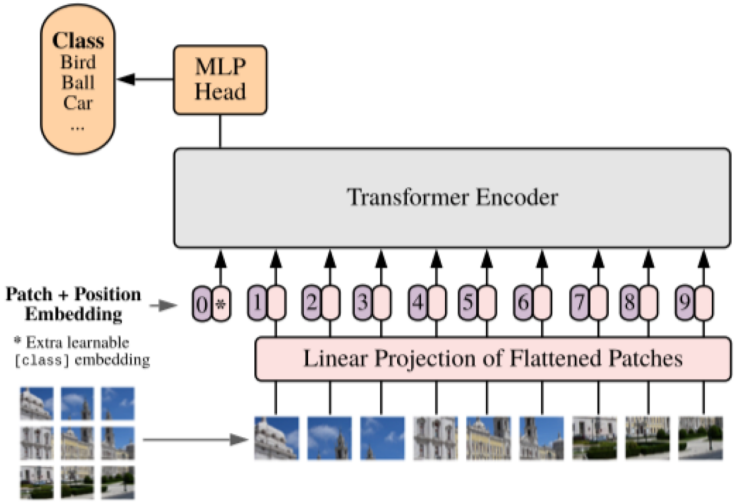
When used with multiple modalities, **self-attention** determines mutually relevant information.

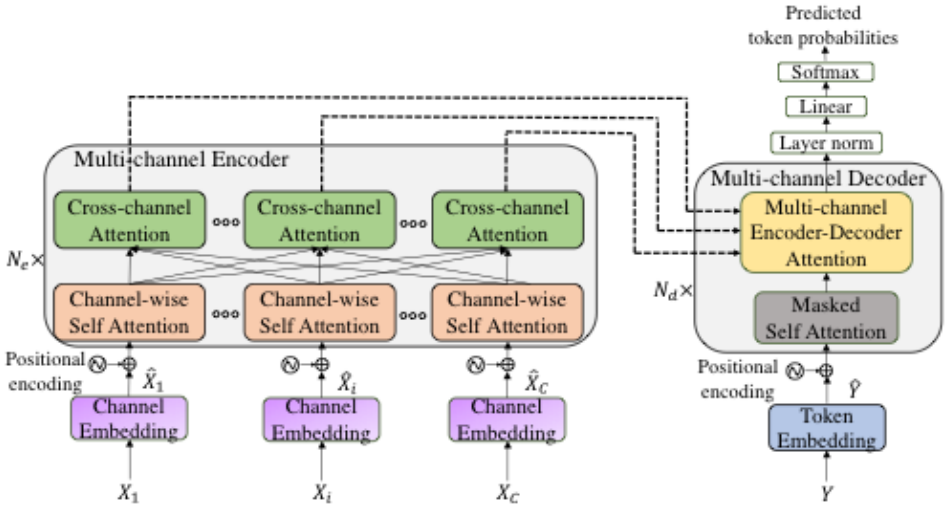
Self-attention can be used to relate words to image patches as well as other words.

From: Ye et. Al. "Cross-Modal Self-Attention Network for Referring Image Segmentation", cvpr 2019, IEEE Conf. on Computer Vision and Pattern Recognition, June, 2019.

Extensions to Vision and Speech

Transformers are rapidly replacing Deep Recurrent Networks and Convolutional networks for **Speech Recognition** and **Computer Vision.**



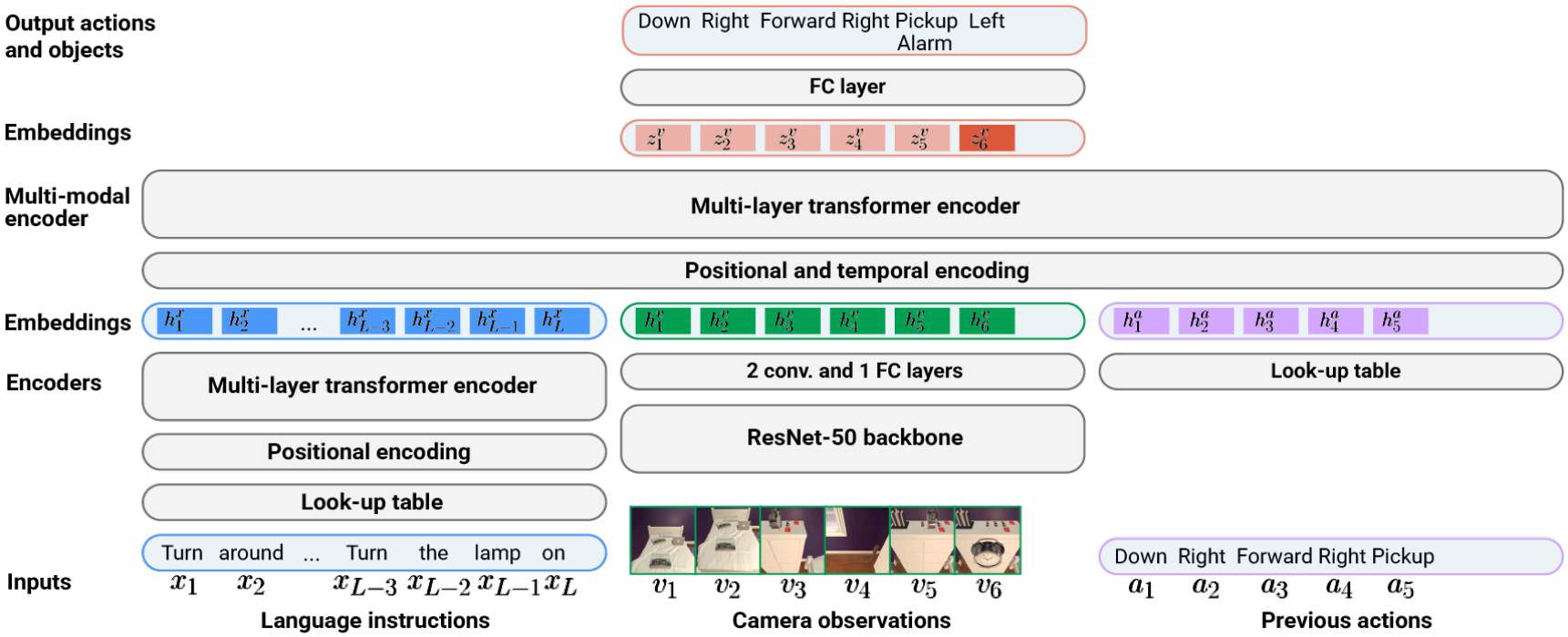


Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J. An image is worth 16x16 words: Transformers for image recognition at scale. ICLR, 2021

Chang, F. J., Radfar, M., Mouchtaris, A., King, B., & Kunzmann, S. (2021, June). End-to-End Multi-Channel Transformer for Speech Recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5884-5888). IEEE, 2021

Multimodal Perception with Transformers

Recent results indicate that Transformers are well adapted for **multi-modal Perception, Robotics and Human-Computer Interaction**



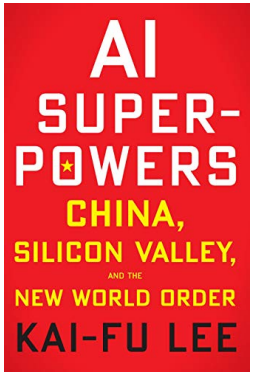
Pashevich, A., Schmid, C. and Sun, C., Episodic Transformer for Vision-and-Language Navigation, Int. Conf. on Computer Vision, ICCV 2021, Oct. 2021.

What happens next?*

What domains are most suitable for economic and societal rupture from AI technologies?

*“predictions are always difficult, especially about the future”
- Niels Bohr (or was it Yogi Berra?, or Mark Twain?)

AI is the fire. Data is the fuel.



To predict AI innovation, look for the data (Kai Fu Lee).

Five Waves of rupture from innovation through AI

1. Internet AI and "AI as a Service" (2015 – 2025) (US and China)
2. Enterprise AI (2015 – 2025) (US leads)
3. Mobile AI using Smart Phones (2015 – 2025) (China leads)
4. Ubiquitous Perception and Interaction (2020 – 2030)
5. Autonomous AI Systems. (2025 – 2035)

USA, China, and Europe are unevenly positioned to profit or suffer from each wave.

Potential Innovations from AI

AI: Human level ability at interaction

Interaction with People:

=> Education, Entertainment, Healthy living,...

Interaction with the Physical World:

=> Robotics, Transportation, Manufacturing, …

Interaction with Systems:

=> Smart Buildings, Smart City, Smart Roads,…

Interaction with Information:

=> Virtual Personal Assistant, travel planning, …

New Categories of interactive Systems

Affectors

Inspire affection.
Compensate for a loss of social contact.
Examples: Aibo, Nao, Paro, ...

Media

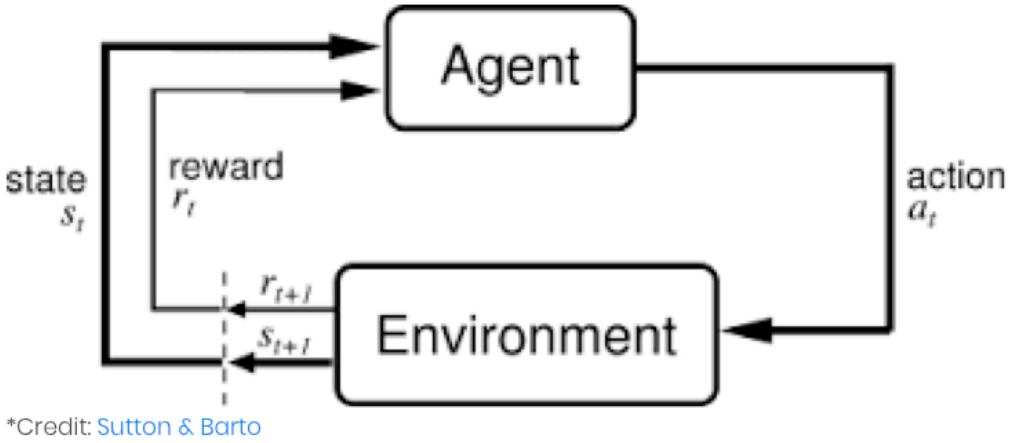
Extend human perception and experience.
Can be interactive or peripheral
Provide a sense of immersion.
Examples: Ambient Orb (Rose 14)

Advisors

Propose courses of actions.
Completely obedient. Do not act.
Avoid unwanted distractions.
Example: GPS Navigation system

Affectors

Affectors: Objects that interact to inspire affection



*Credit: Sutton & Barto

Multimodal perception of affect and **Deep Reinforcement Learning** can be used to learn actions for stimulating Affection.

Used with affective computing, can be adapted to any interaction.

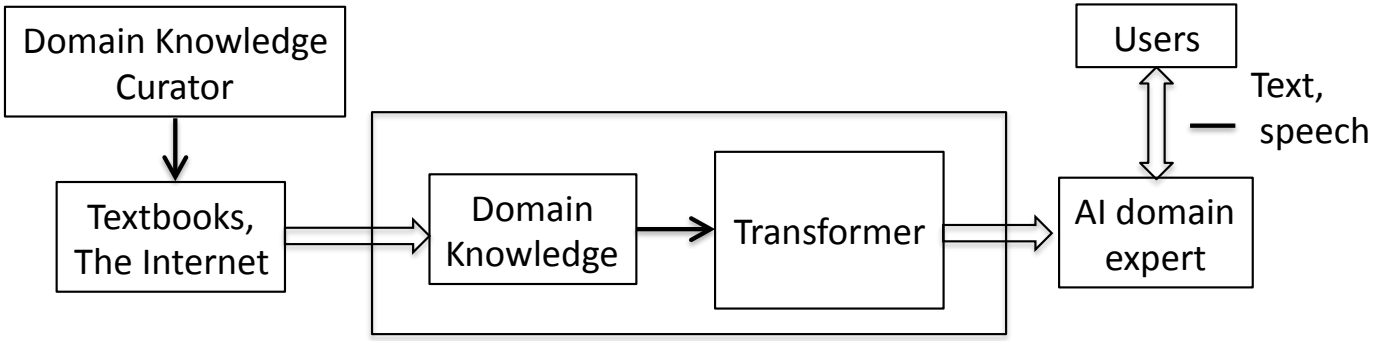
Media: Augmented Reality

Current AR technology: Displayed information is “pre-programmed”

With AI and Machine learning:

- Computer vision can learn to recognize new phenomena
- Multi-modal interaction and Reinforcement Learning can be used to learn the appropriate information to display

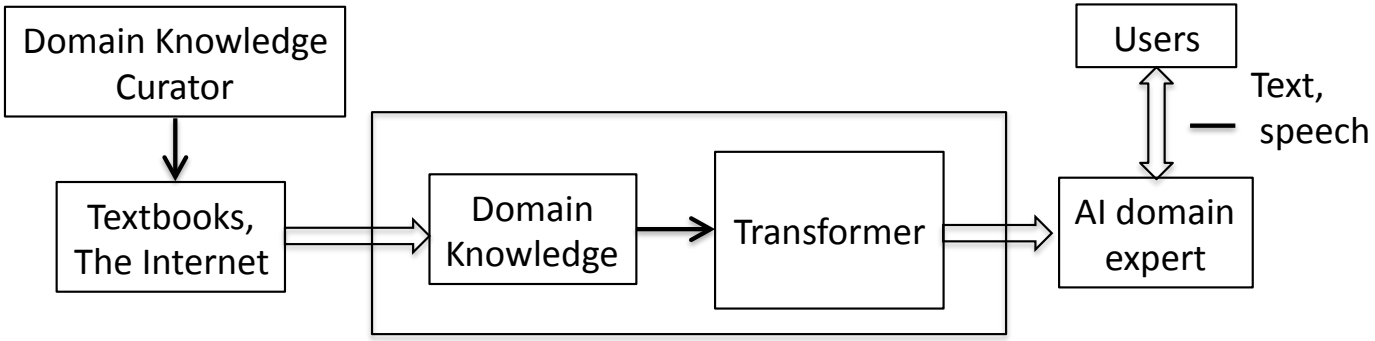
Advisors: Cognitive Computing



Transformers can be trained with domain knowledge to serve as expert advisors about a scientific literature.

What are key references to read about <x>?
Has anyone published data on this phenomena?

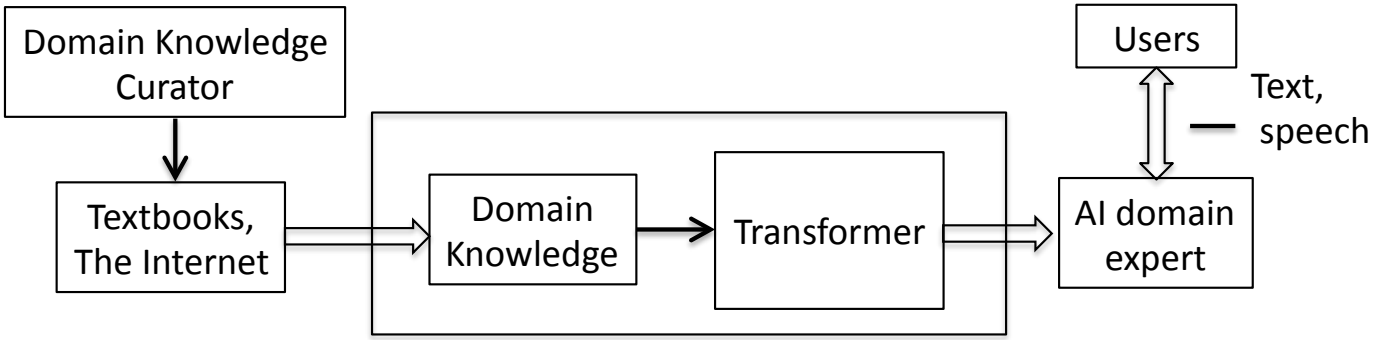
AI Chatbots with Transformers



Intelligent Chatbots: Can be trained on any recorded literature.

Example: Replika. Based on OpenAI's GPT-3 transformer, Replika was trained on emails and text chats from a programmers' deceased friend, and learned a realistic imitation of the language patterns.

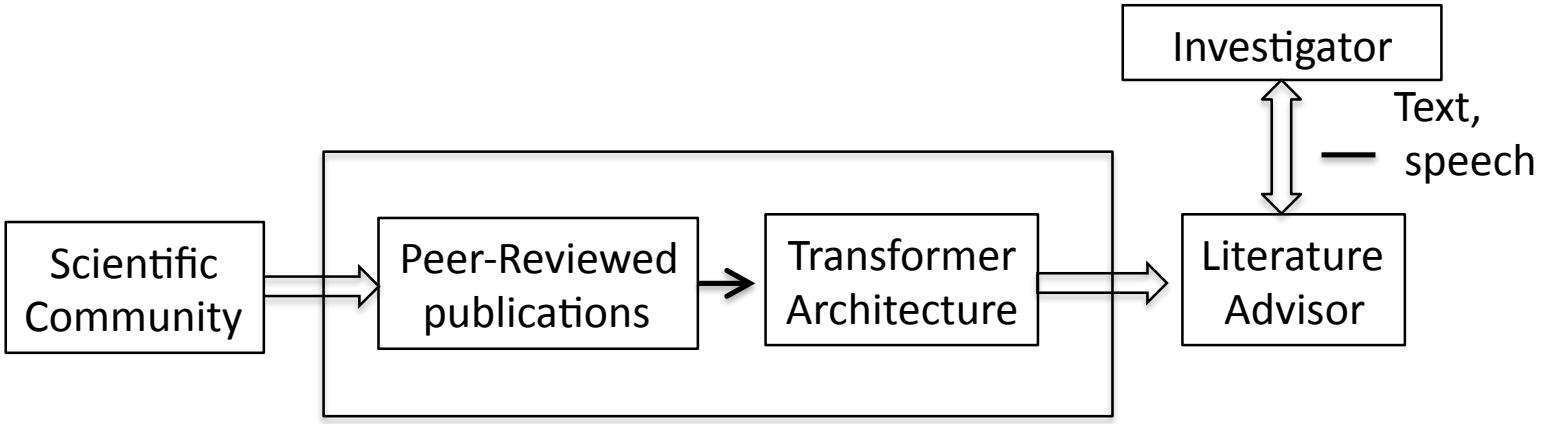
Transformers Can be used for Intelligent Advisors



Transformers can encodes knowledge from any written source (textbooks, literature, the internet) to generate a domain expert advisor program (an expert system!)

Example domains: Medical, Legal, Financial, Scientific, Programming, ….

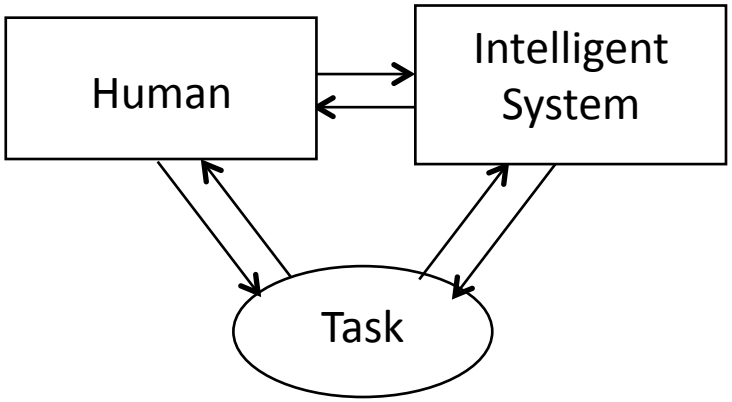
Intelligent Advisors:
A Tool for Monitoring the Scientific Literature



Transformers can potentially provide a tool to provide advice and guidance in monitoring the scientific literature.

Intelligent advisors would NOT discover new concepts, but COULD provide a tool to augment human intelligence.

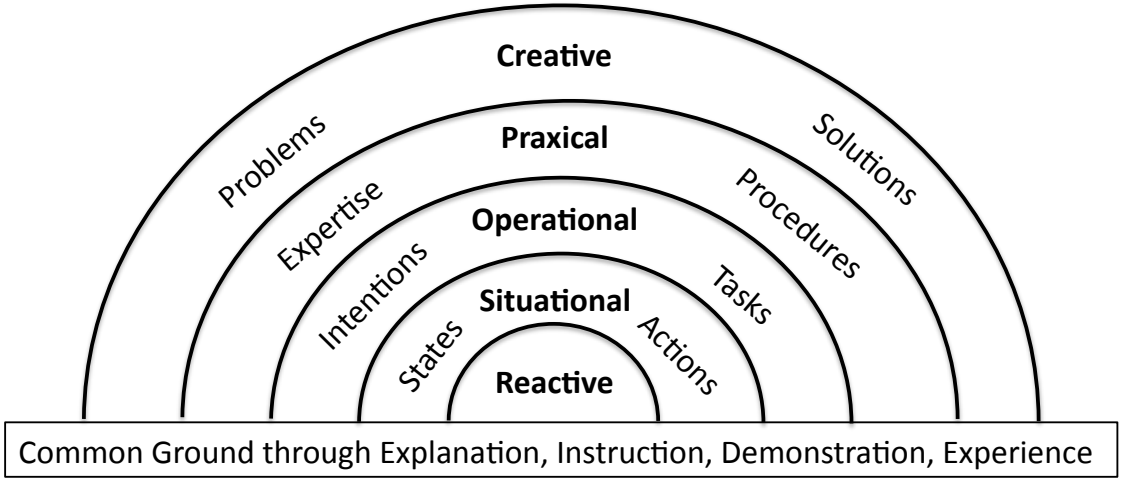
Beyond Advisors:
Collaborative Intelligent Systems



Collaboration is a process where two or more actors (agents) **work together** in order **to achieve** some shared **goals**.

Collaborative Intelligent Systems are intelligent systems that work with humans as partners to achieve a common goal, sharing a **mutual understanding** of the abilities and respective roles of each other.

Empowering humans with AI.
Collaborative Intelligent Systems



Challenge: Build collaborative intelligent systems that work synergistically with human user. Human and system each play roles suited for their abilities:

System: Collect, analyze and display massive volumes of data
Detect anomalous phenomena, advise about literature.

Human: Determine challenges where theories and models fail.
Generate conceptual understanding.

Conclusions:

- Intelligence is human-level performance at interaction.
- Machine Learning is a rupture technology for Artificial Intelligence.
- Machine Learning is made possible by planetary scale data, and massive scale computing using GPUs, TPUs and ASICs
- Transformers (Stacked auto-encoders trained with self-supervised learning) open all recorded literature to Artificial Intelligence
- Artificial Intelligence can provide Affectors, Media and Advisors
- For Innovation: if AI is the fire, data is the fuel.

*TPU: Tensor Processing Units. ASIC: Application Specific integrated Circuits

The Emergence of Machine Learning as a Rupture technology for Artificial Intelligence

James L. Crowley
Professor Emeritus, Grenoble INP
Grenoble Informatics Laboratory (LIG)
INRIA Grenoble Rhone-Alpes
Univ. Grenoble Alpes