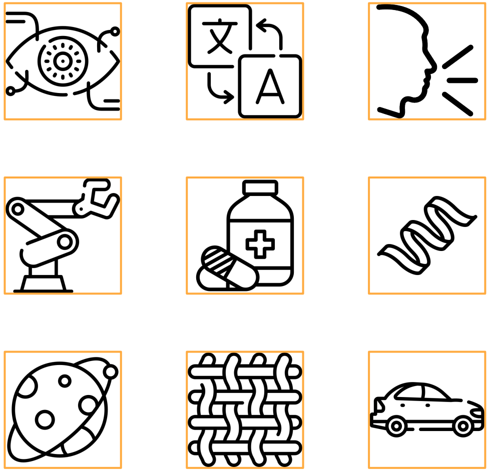
Honouring the 2018 ACM Turing Award Laureates

Geoffrey E. Hinton, Yann LeCun, Yoshua Bengio

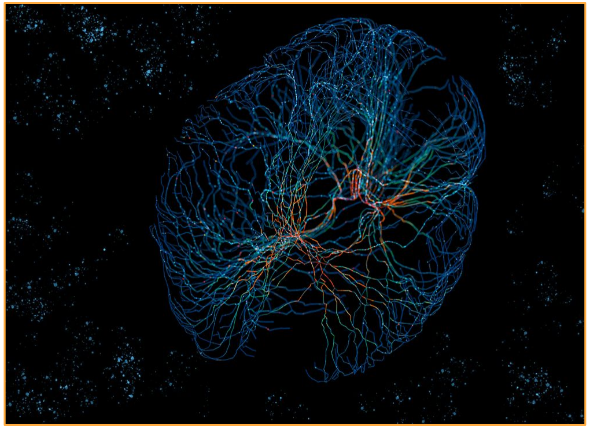
"for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing"

Artificial intelligence
is transforming nearly everything.



AI is the most rapidly growing area in all of science, and certainly one of the most talked-about topics in society.

The success of AI is driven by deep learning.



The incredible advances in AI would not have been possible without some of the foundations, namely, deep learning.

> Deep learning is a subset of machine learning that uses **artificial neural networks** to learn from data. Artificial neural networks are inspired by the human brain, and they are able to learn complex patterns from large amounts of data. Deep learning has been used ...

▲ the response from Google Bard

G. Hinton, Y. LeCun, and Y. Bengio pioneered deep learning.

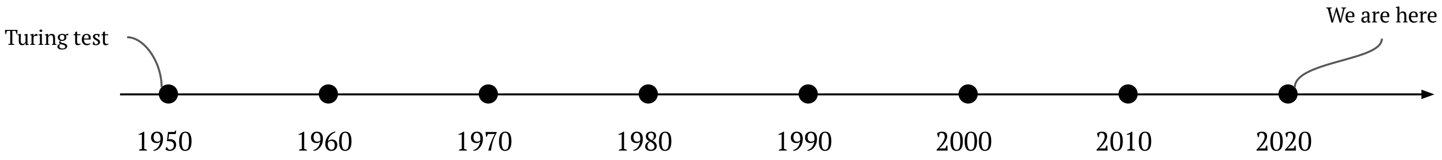
  

They collectively and independently worked over 30 years.

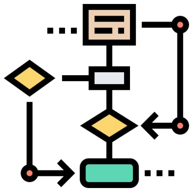
- establish conceptual foundations for deep networks
- identify a lot of interesting phenomena
- develop engineering advances in practice

Apparently, it wasn't always easy.

Two paradigms since 1950s

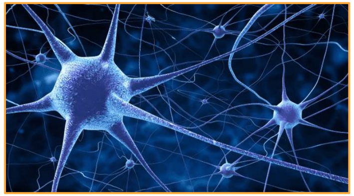


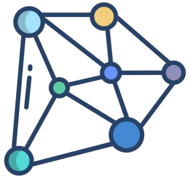
The two paradigms:
Symbolic AI and Connectionist AI



Symbolic AI

- the logic-inspired approach
- use human-readable symbolic rules
- focus on reasoning





Connectionist AI

- the biological-inspired approach
- learn strengths of the connections in a neural network (vectors)
- focus on learning

While the learning approach can solve it, the symbolic approach can't.

What are shown in the photo?

A man and a chicken.

What does the man feel and why?

He is scared of the chicken because it is flying at him.

Image captioning task:

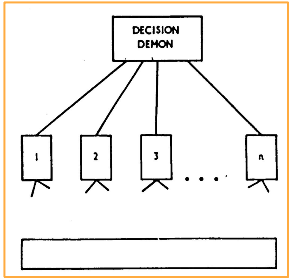
Symbolic AI people tried this for a long time, but they failed.

- It's not obvious how to write that program.

It became fairly easy for connectionist AI to solve this task.

- Current methods can capture subtleties.

The central question: can they learn to do it?



Pandemonium (1959)

trial and error

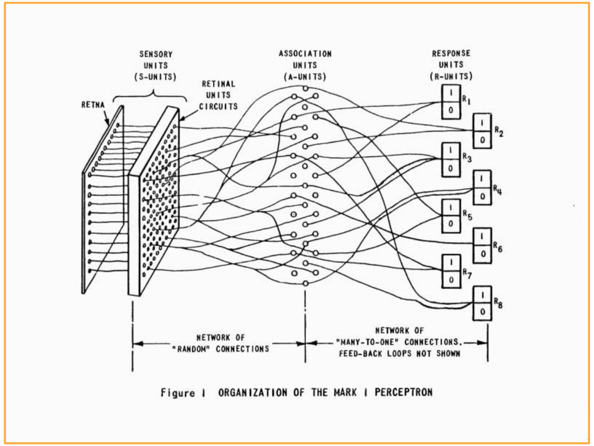
Large neural networks are very powerful computing devices.

- But can a neural network learn a difficult tasks?

Early researchers like Turing and Selfridge proposed that neural networks with initially random connections could be trained by reinforcement learning.

- This is extremely inefficient.

Perceptrons:
A simple learning procedure.

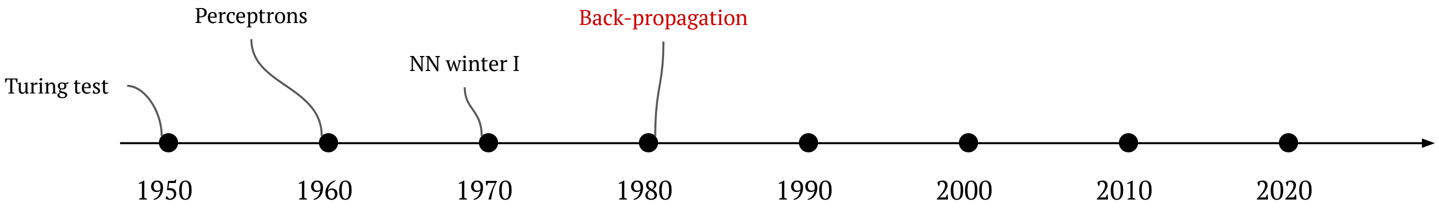


~1960: Rosenblatt introduced a simple, efficient learning procedure that could figure out how to weight features of the input in order to classify inputs correctly.

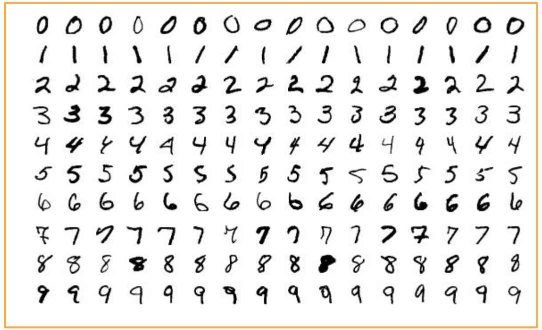
1969: Minsky and Papert showed that perceptrons had some very strong limitations on what they could do.

1970s: The first neural net winter has begun.

Back-propagation in 1980s

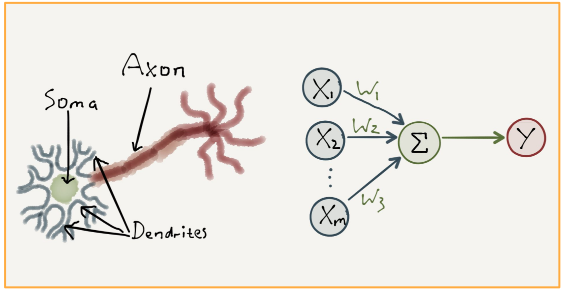


Back-propagation created a lot of excitement.



1980s: The back-propagation procedure allows neural networks to design their own features and have multiple layers of features.

What is an artificial neuron?

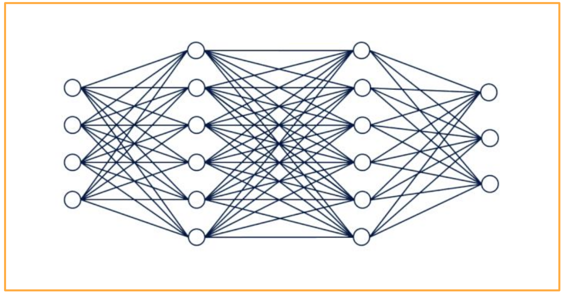


McCulloch-Pitts neuron (1943)

We make a gross idealization of a real neuron so that we can investigate how neurons can collaborate to do computations that are too difficult to program such as:

- Convert the pixel intensity values of an image into a string of words that describe the image.

What is an artificial neural network?



A feed-forward neural network

You have weights on the incoming weights for each of these neurons, and as you change those incoming weights, you're changing what feature that neuron will respond to. So by learning these weights, you're learning the features.

The question is how are we gonna train it?

- Supervised (or unsupervised) training
- "Mutation" methods?

How do we train artificial neural networks?
A “mutation” method that is easy to understand?

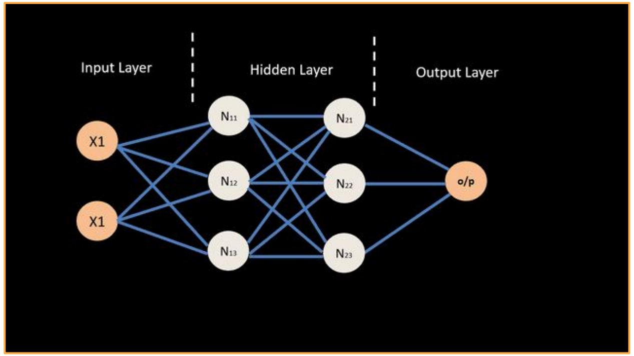
Supervised training: Show the network an input vector and tell it to the correct output.

- Adjust the weights to reduce the discrepancy between the correct output and the actual output.

A mutation method:

- Pick one weight. Increase or decrease the weight slightly and measure how well the network now does. Keep it if it helped.

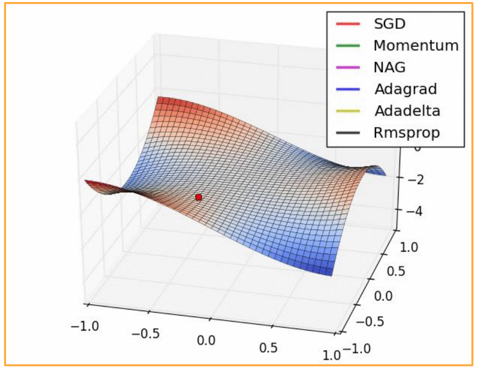
The backpropagation algorithm



Instead of perturbing the weights one at a time and measuring the effect, use calculus to compute the error gradients for all of the weights at the same time.

- With a million weights, this is more efficient than the mutation method by a factor of a million.
- Once the gradient is computed, perform stochastic gradient descent, which works really well at scale.

How to learn many layers of features (~1985)



1. Forward pass through the network (for a small batch).

2. Calculate the difference between what you got and what you wanted.

3. Backward pass with the chain rule.

4. Take a stochastic optimisation step (e.g., SGD).

- It works really well at scale.

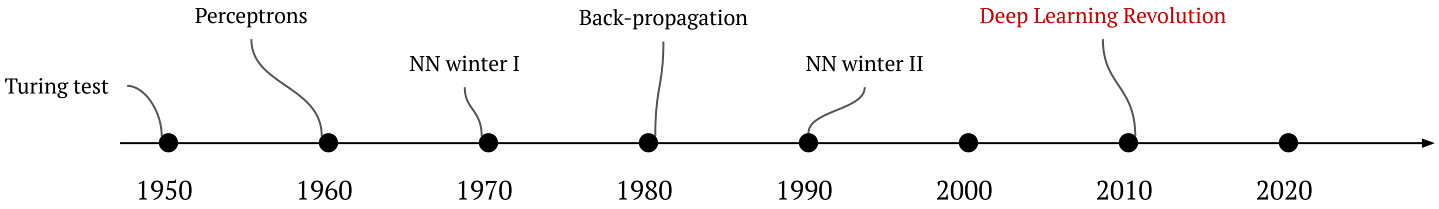
A big disappointment, and the 2nd neural net winter.

1990s: BP works pretty well, but underperforms the expectations of its proponents on modest-sized datasets.

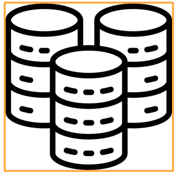
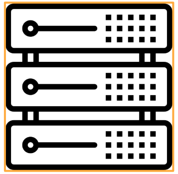
- Symbolic AI: it is silly to expect to learn difficult tasks in big deep nets that start with random connections and no prior knowledge.
- A series of rejections from NIPS, ICML, and CVPR.

The second neural network winter begins.

the Deep learning revolution



The return of backpropagation, largely due to a lot of data and compute.

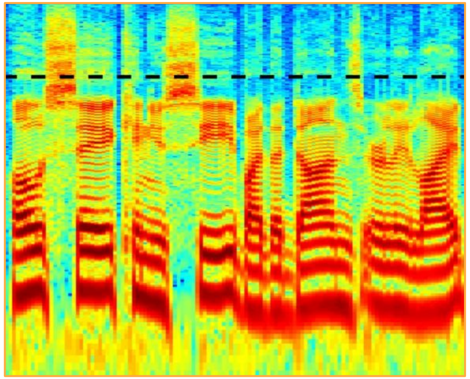
Between 2005 and 2009, several technical advances enabled back-propagation to work better in feed-forward nets.

- Unsupervised pre-training; random dropout of units; rectified linear units.

Back-propagation works amazingly well if you have two things:

- a lot of labeled data
- a lot of convenient compute power (e.g. GPUs)

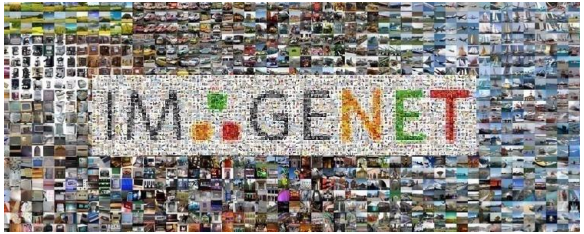
Acoustic modeling:
The killer App (Mohamed, Dahl & Hinton 2009)

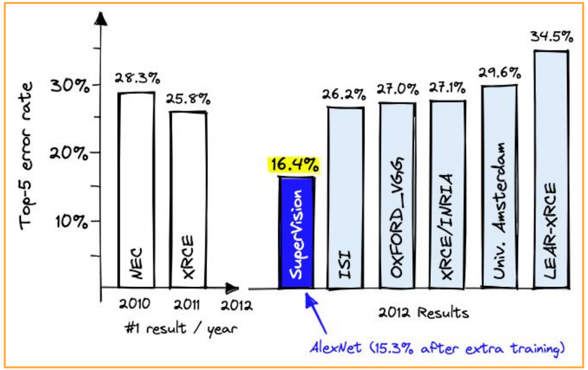


Acoustic modeling: for the middle frame of the spectrogram, which piece of which phone in the speaker is trying to express?

Soon after, leading speech groups at MSR, IBM & Google developed them further.

Object Recognition:
the 2012 ImageNet object recognition challenge.





The challenge:

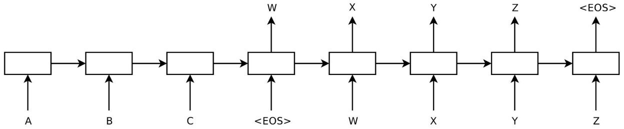
- Given a million high-resolution training images of 1000 different classes of object, get the “correct” class in your top 5 bets.

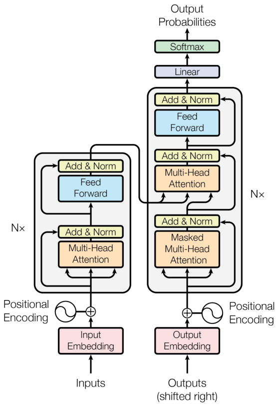
Error rates:

- While all previous / standard ones asymptote at about 25% error, the AlexNet got 16% error.

- By 2015 it was down to 5%. And now it’s down to considerably below that.

A radically new way to do machine translation (Sutskever, Vinayals and Le, 2014)





The sequence-to-sequence model:

- The encoder read reads in the sequence of words, and the decoder expresses the thought in the target language (“thought vector”).

A lot of advances since 2014:

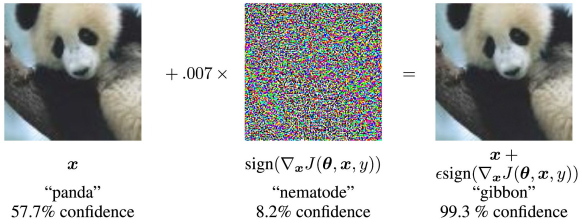
- Soft attention, pre-training, and the transformer networks.

The final nail in the coffin of symbolic AI:

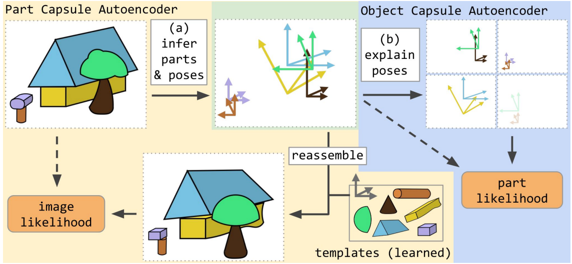
- Machine translation is an idea task for symbolic AI because the input is symbols and the output is symbols (“vectors inside”).

Continuing advances

The future of neural network vision



Adversarial example (2015)



Stacked Capsule Autoencoder (2019)

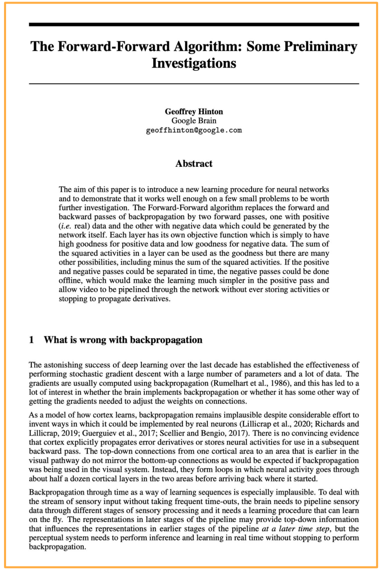
Convolutional nets get a huge win by wiring in the idea that if a feature is useful in one location it is useful everywhere.

- But they do not recognize objects the same way as us, hence adversarial examples.

People recognize objects by using the viewpoint invariant geometrical relationships between an object and its parts.

- We can make neural networks do this by using transformers.

The future of neural networks


The Forward-Forward Algorithm: Some Preliminary Investigations

Geoffrey Hinton
Google Brain
geoffhinton@google.com

Abstract

The aim of this paper is to introduce a new learning procedure for neural networks and to demonstrate that it works well enough on a few small problems to be worth further investigation. The Forward-Forward algorithm replaces the forward and backward passes of backpropagation by two forward passes, one with positive (*i.e.* real) data and the other with negative data which could be generated by the network itself. Each layer has its own objective function which is simply to have high goodness for positive data and low goodness for negative data. The sum of the squared activities in a layer can be used as the goodness but there are many other possibilities, including minus the sum of the squared activities. If the positive and negative passes could be separated in time, the negative passes could be done offline, which would make the learning much simpler in the positive pass and allow video to be pipelined through the network without ever storing activities or stopping to propagate derivatives.

1 What is wrong with backpropagation

The astonishing success of deep learning over the last decade has established the effectiveness of performing stochastic gradient descent with a large number of parameters and a lot of data. The gradients are usually computed using backpropagation (Rumelhart et al., 1986), and this has led to a lot of interest in whether the brain implements backpropagation or whether it has some other way of getting the gradients needed to adjust the weights on connections.

As a model of how cortex learns, backpropagation remains implausible despite considerable effort to invent ways in which it could be implemented by real neurons (Lillicrap et al., 2020; Richards and Lillicrap, 2019; Guerguiev et al., 2017; Scellier and Bengio, 2017). There is no convincing evidence that cortex explicitly propagates error derivatives or stores neural activities for use in a subsequent backward pass. The top-down connections from one cortical area to an area that is earlier in the visual pathway do not mirror the bottom-up connections as would be expected if backpropagation was being used in the visual system. Instead, they form loops in which neural activity goes through about half a dozen cortical layers in the two areas before arriving back where it started.

Backpropagation through time as a way of learning sequences is especially implausible. To deal with the stream of sensory input without taking frequent time-outs, the brain needs to pipeline sensory data through different stages of sensory processing and it needs a learning procedure that can learn on the fly. The representations in later stages of the pipeline may provide top-down information that influences the representations in earlier stages of the pipeline *at a later time step*, but the perceptual system needs to perform inference and learning in real time without stopping to perform backpropagation.

Forward-forward, Hinton (2022)

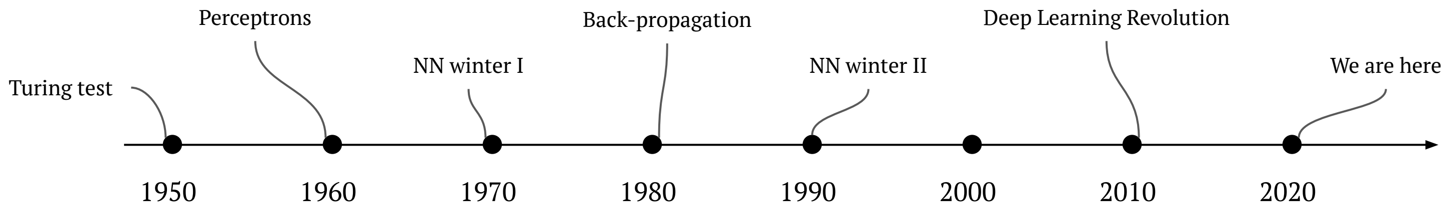
Nearly all artificial neural nets use only two time scales: slow adaptation of weights and fast changes in neural activity.

But synapses adapt at multiple different time scales.

- Using fast weights for short-term memory will make neural networks different and better.
- It can improve optimization.
- It allows true recursion (1973, unpublished).

Closing

Summary &
The message



We've seen the *history* of deep learning, and its transformative and revolutionary *impact* on society.

Deep learning with large neural networks is becoming *critical* for nearly everything.

Credits

"a baby is laying on a seat with a remote",
generated using 🤗 Hugging Face API

G. Hinton and Y. LeCun's Turing lectures at 2019 ACM FCRC,

J. Doumont's Trees, maps, and theorems,

various sources on the Internet for visual materials,

the audience who just came to this event (on Friday),

and, of course, my family.

Thank you

-Namhoon Lee, POSTECH