

Neural Architectures: from the Mc-Culloch Pitts Model to GPT-3

Overview of Neural Models

Mirco Musolesi

mircomusolesi@acm.org

Computers and Brains

	Supercomputer	Personal Computer	Human Brain
Computational units	10^6 GPUs + CPUs 10^{15} transistors	8 CPU cores 10^{10} transistors	10^6 columns 10^{11} neurons
Storage units	10^{16} bytes RAM 10^{17} bytes disk	10^{10} bytes RAM 10^{12} bytes disk	10^{11} neurons 10^{14} synapses
Cycle time	10^{-9} sec	10^{-9} sec	10^{-3} sec
Operations/sec	10^{18}	10^{10}	10^{17}

From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

From Theories of Biological Learning to Deep Learning

- ▶ Three waves:
 - ▶ Cybernetics (1940s-1960s)
 - ▶ Connectionism (1980s-1990s)
 - ▶ Deep learning (2006-today)
- ▶ Some of the earliest learning algorithms were intended to be computational models of the brain. As a result, one of the names used for deep learning is *artificial neural networks (ANNs)*.

Cybernetics

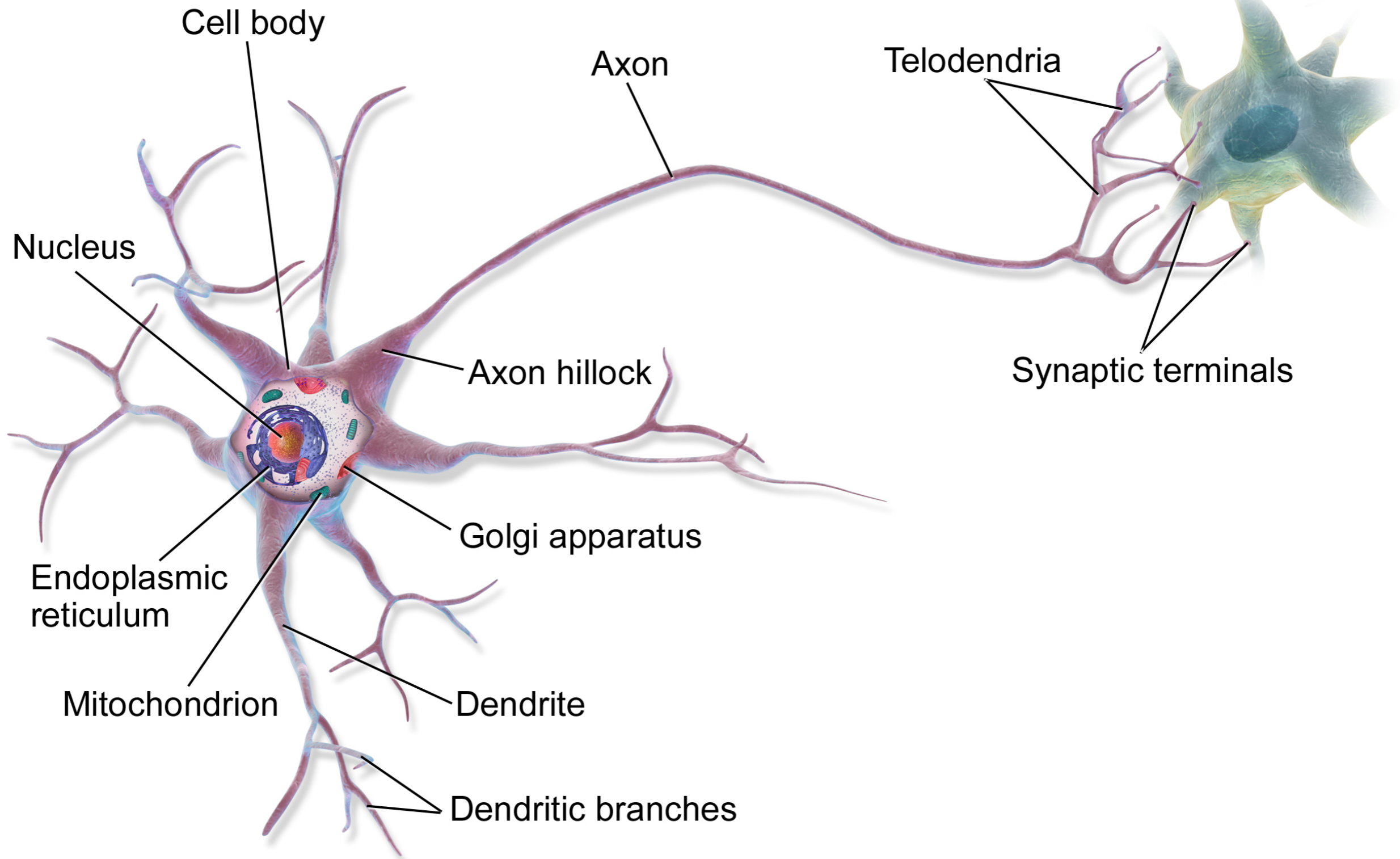
or CONTROL and COMMUNICATION
in THE ANIMAL and THE MACHINE

By NORBERT WIENER

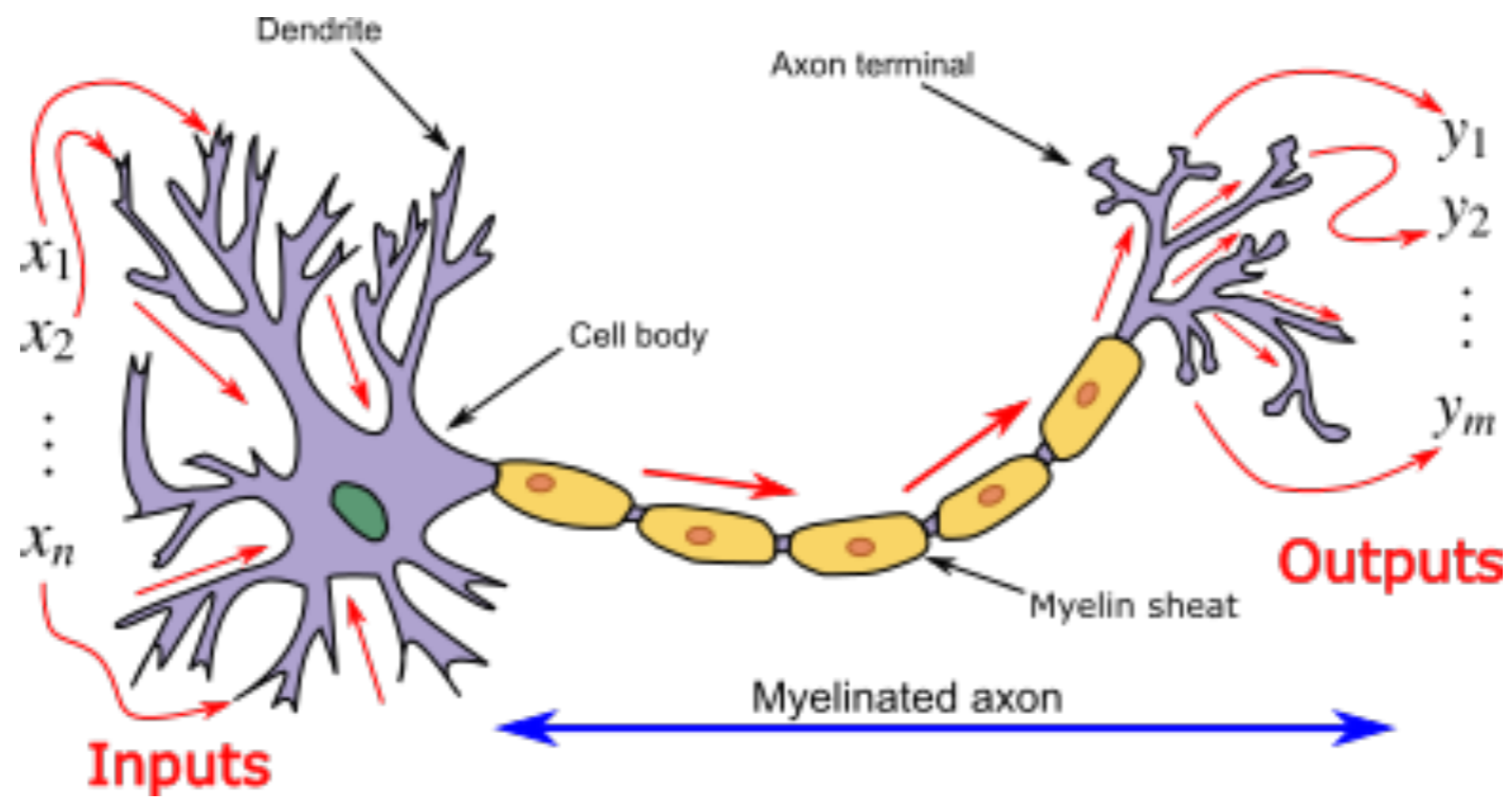
A study of vital importance to psychologists,
physiologists, electrical engineers, radio engineers,
sociologists, philosophers, mathematicians, anthro-
pologists, psychiatrists, and physicists.

Artificial Neural Networks and Neuroscience

- ▶ The earliest predecessors of modern deep learning were simple linear models motivated from a neuroscience perspective.
- ▶ These models were designed to take a series of n input values x_1, x_2, \dots, x_n and associate them to an output y .
- ▶ These models would be based or learn a set of weights:
$$y = f(\mathbf{x}, \mathbf{w}) = w_1x_1 + \dots + w_nx_n$$



Credit: Wikimedia



Credit: Wikimedia

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

I. Introduction

Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurrence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from less than one meter per second in thin axons, which are usually short, to more than 150 meters per second in thick axons, which are usually long. The time for axonal conduction is consequently of little impor-

get the other and gives the same results, although perhaps not at the same time. Various applications of the calculus are discussed.

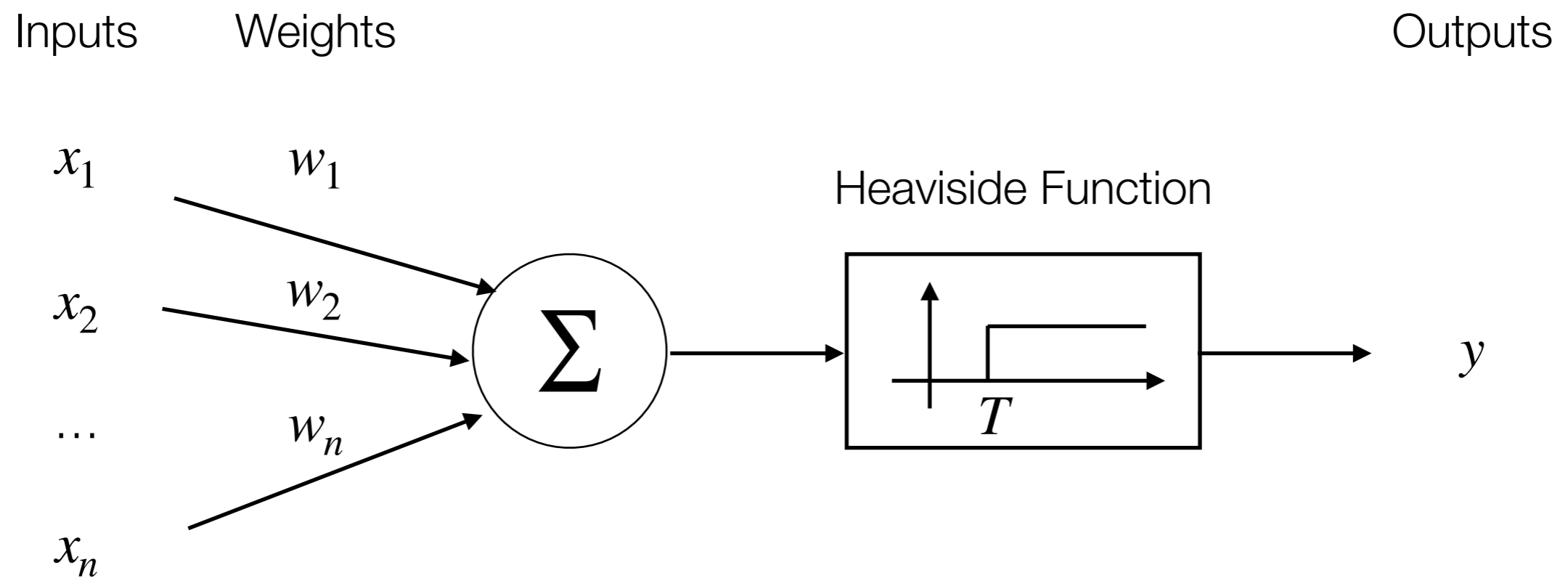
I. Introduction

Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurrence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from less than one meter per second in thin axons, which are usually short, to more than 150 meters per second in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any

The McCulloch-Pitts Neural Model

- ▶ In “A Logical Calculus of the Ideas Imminent in Nervous Activity” (1943), McCulloch and Pitts suggested a model about how thought executes.
- ▶ This is the original inspiration of current deep learning models.
- ▶ The set of operations is defined over two values:
 - ▶ True (1)
 - ▶ False (0)
- ▶ The calculus contained NOT, AND, OR. By changing the (fixed) values of the weights, you can obtain different functions.

McCulloch-Pitts Model of Neuron



In the McCulloch-Pitts model, the values of the weights are fixed.

*The Organization
of Behavior*

A NEUROPSYCHOLOGICAL THEORY

D. O. HEBB

McGill University

1949

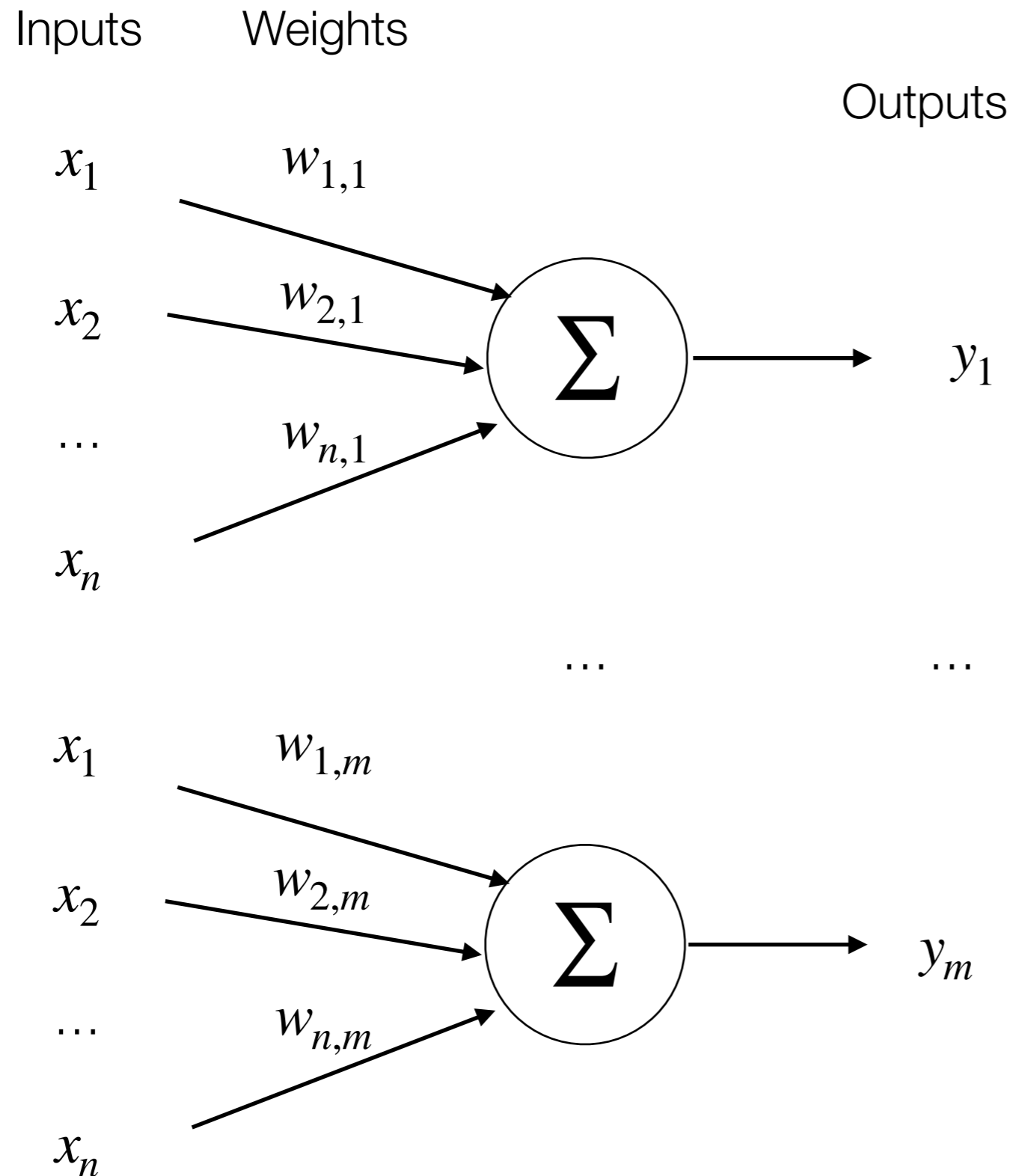
New York · JOHN WILEY & SONS, Inc.

London · CHAPMAN & HALL, Limited

Hebb's Law

- ▶ From Hebb's "The Organization of Behavior" (1949): "When an axon cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such as that A 's efficiency, as one of the cells firing B is increased".
- ▶ This is usually referred to as "Hebb's Law".
- ▶ First simulations of artificial neural networks in 1950s based on Hebb's model.
- ▶ Weights of the models are also called synaptic connectivity.

Hebb's Model of Neuron



The Hebbian network model has n -node input layer:

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$

and an m -node output layer

$$\mathbf{y} = [y_1, y_2, \dots, y_m]^T$$

Each output is connected to all input as follow:

$$y_i = \sum_{j=1}^n w_{j,i} x_j$$

The learning rule is the following:

$$w_{j,i}^{new} \leftarrow w_{j,i}^{old} + \eta x_j y_i$$

η is the learning rate.

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN¹

F. ROSENBLATT

Cornell Aeronautical Laboratory

If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental questions:

1. How is information about the physical world sensed, or detected, by the biological system?

2. In what form is information stored, or remembered?

3. How does information contained in storage, or in memory, influence

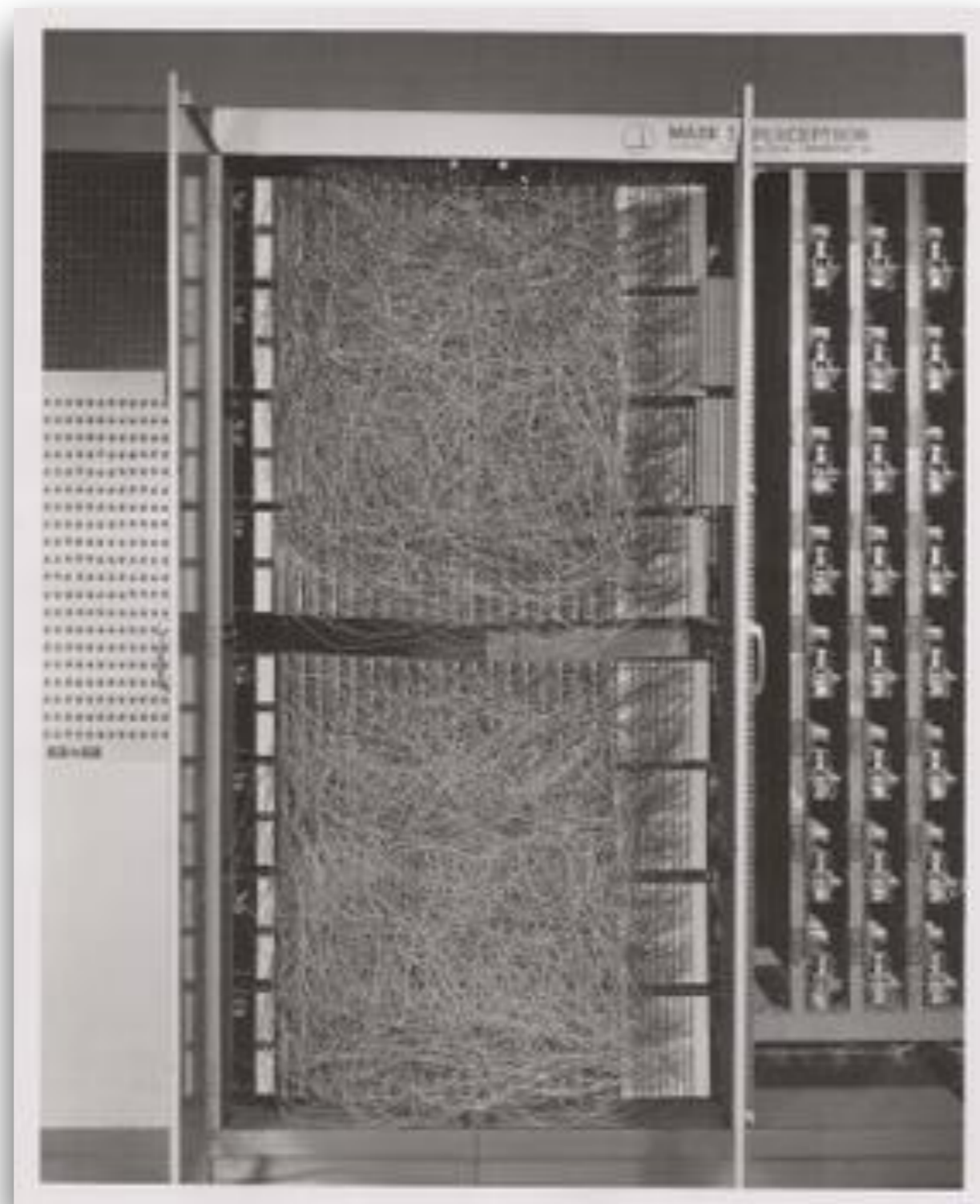
and the stored pattern. According to this hypothesis, if one understood the code or "wiring diagram" of the nervous system, one should, in principle, be able to discover exactly what an organism remembers by reconstructing the original sensory patterns from the "memory traces" which they have left, much as we might develop a photographic negative, or translate the pattern of electrical charges in the "memory" of a digital computer. This hypothesis is appealing in its simplicity and ready intelligibility,

Rosenblatt's Perceptron Model

- ▶ Frank Rosenblatt's perceptron model is the first one with variable weights that are learned from examples:
 - ▶ Learning the weights of categories given examples of these categories.
- ▶ The perceptron was intended to be a machine rather than a program.
 - ▶ First implementation was actually for IBM 704.

IBM 704





Mark I Perceptron

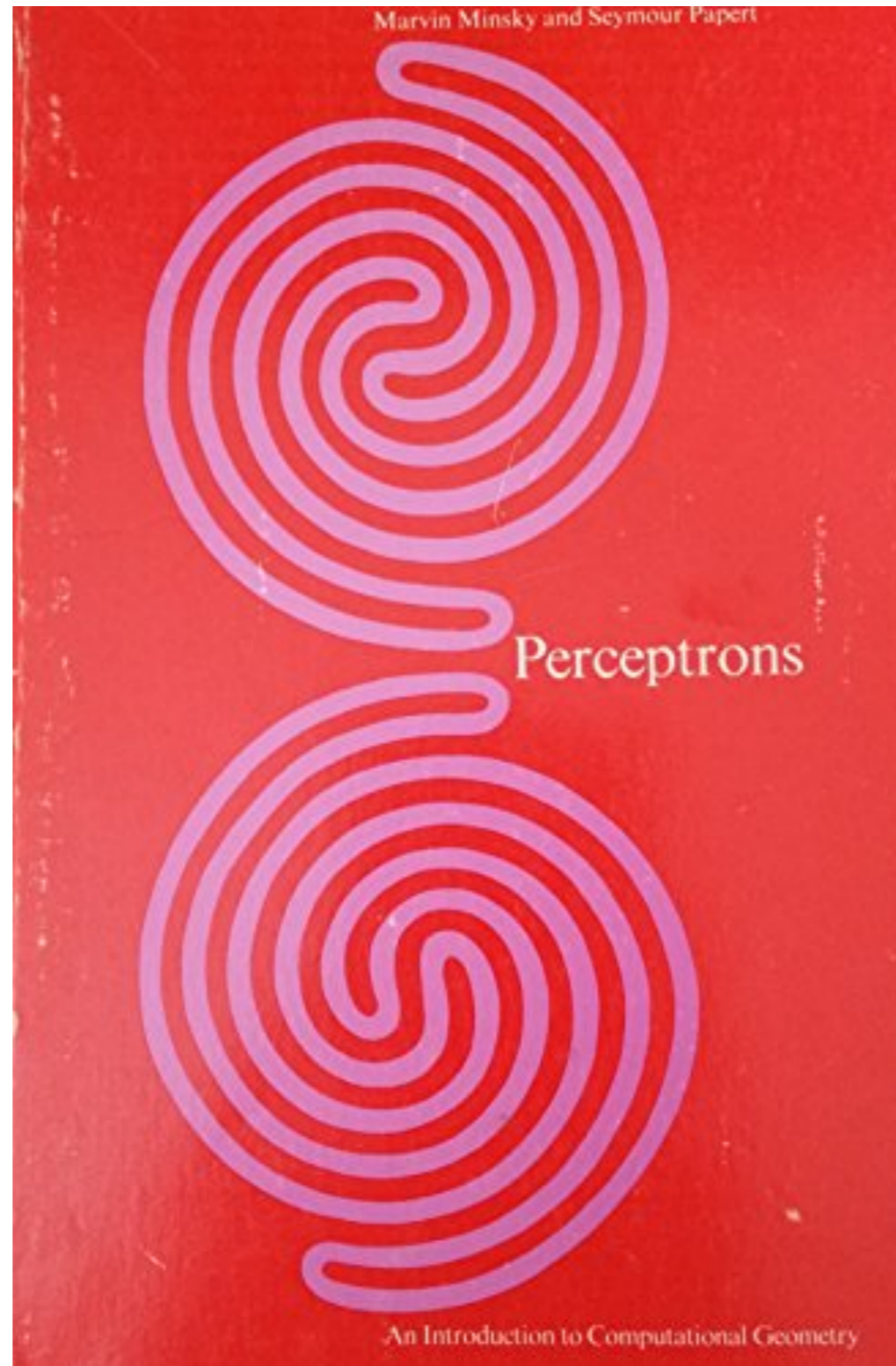
Built for image
recognition

400 photocells
randomly connected
to the neurons

Weights encoded in
potentiometers

Limitations of Perceptrons

- ▶ Linear models have many limitations.
- ▶ Most famously, they cannot learn the XOR function where $f([0,1], \mathbf{w}) = 1$ and $f([1,0], \mathbf{w}) = 1$ but $f([1,1], \mathbf{w}) = 0$ and $f([0,0], \mathbf{w}) = 0$.
- ▶ This was observed by Minsky and Papert in 1969 in *Perceptrons*.
- ▶ This was the first major dip in the popularity of neural networks.



Neurocognitron and Convolutional Neural Networks

- ▶ Neuroscience can be an inspiration for the design of novel architectures and solutions.
- ▶ The basic idea of having multiple computational units that become intelligent via their interactions with each others is inspired by the brain.
- ▶ The *neurocognitron* introduced by Fukushima can be considered as a basis for the modern convolutional networks architectures.
- ▶ The neurocognitron was the basis of the modern convolutional network architectures (see Yann LeCun et al.'s LeNet architecture).

Cognitron: A Self-organizing Multilayered Neural Network

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

Received: February 4, 1975

Abstract

A new hypothesis for the organization of synapses between neurons is proposed: "The synapse from neuron x to neuron y is reinforced when x fires provided that no neuron in the vicinity of y is firing stronger than y ". By introducing this hypothesis, a new algorithm with which a multilayered neural network is effectively organized can be deduced. A self-organizing multilayered neural network, which is named "cognitron", is constructed following this algorithm, and is simulated on a digital computer. Unlike the organization of a usual brain models such as a three-layered perceptron, the self-organization of a cognitron progresses favorably without having a "teacher" which instructs in all particulars how the individual cells respond. After repetitive presentations of several stimulus patterns, the cognitron is self-organized in such a way that the receptive fields of the cells become relatively larger in a deeper layer. Each cell in the final layer integrates the information from

At present, however, the algorithm with which a neural network is self-organized is not known. Although several hypothesis for it have been proposed, none of them has been physiologically substantiated.

The three-layered perceptron proposed by Rosenblatt (1962) is one of the examples of the brain models based on such hypotheses. For a while after the perceptron was proposed, its capability for information processing was greatly expected, and many research works on it have been made. With the progress of the researches, however, it was gradually revealed that the capability of the perceptron is not so large as it had been expected at the beginning.

Although the perceptron consists of only three

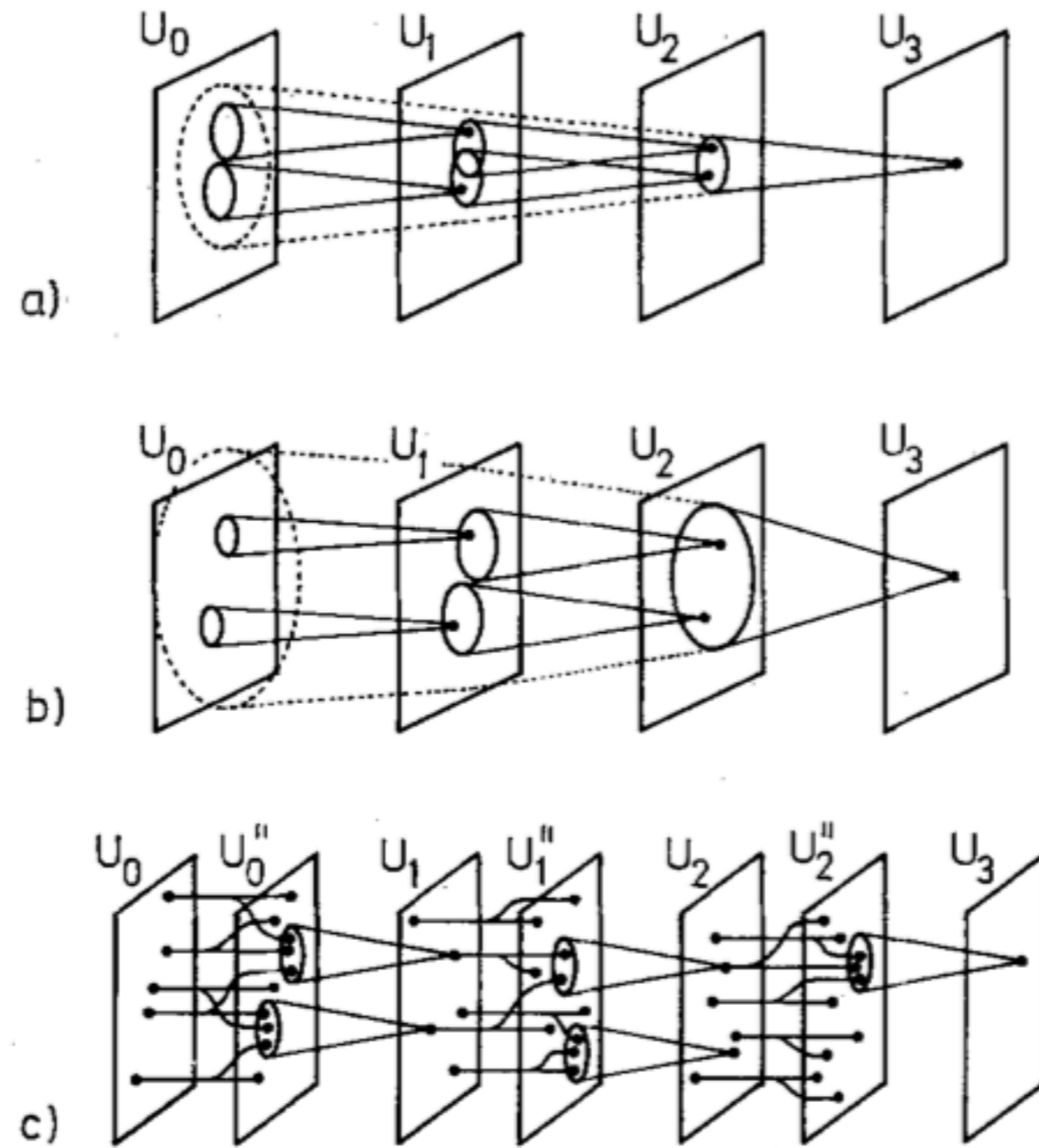


Fig. 4a-c. Three possible methods for interconnecting layers. The connectable area of each cell is differently chosen in these three methods. Method c is adopted for the cognitron discussed in this paper

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

Abstract—

Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called Graph Transformer Networks (GTN), allows such multi-module systems to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks.

A Graph Transformer Network for reading bank check is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provide record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

I. INTRODUCTION

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed heuristics. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on pixel images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually integrating individually designed modules can be replaced by a unified and well-principled design paradigm, called *Graph Transformer Networks*, that allows training

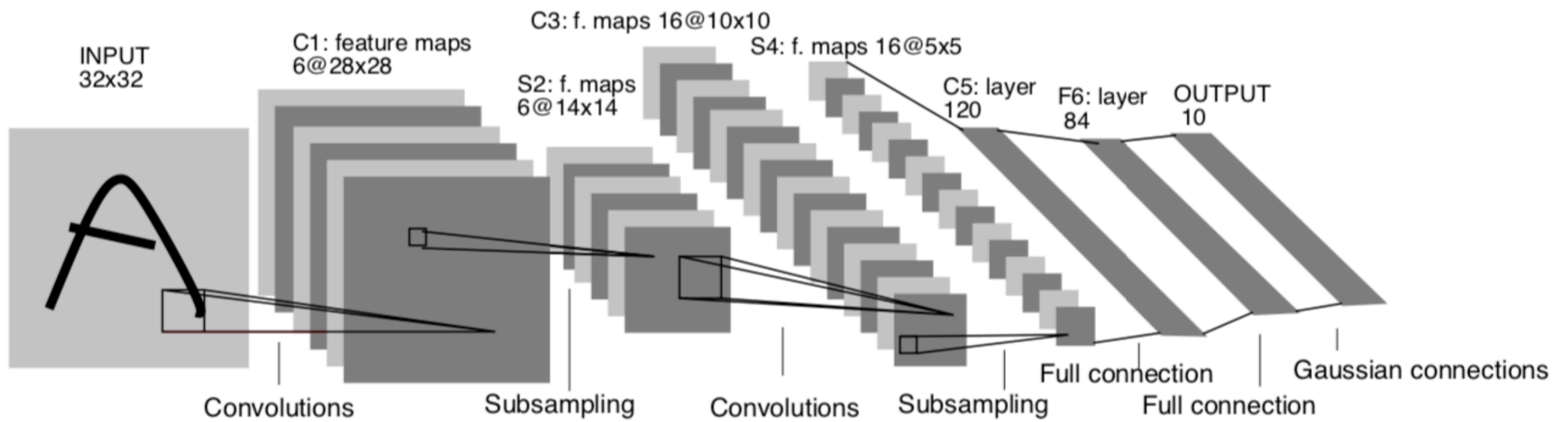
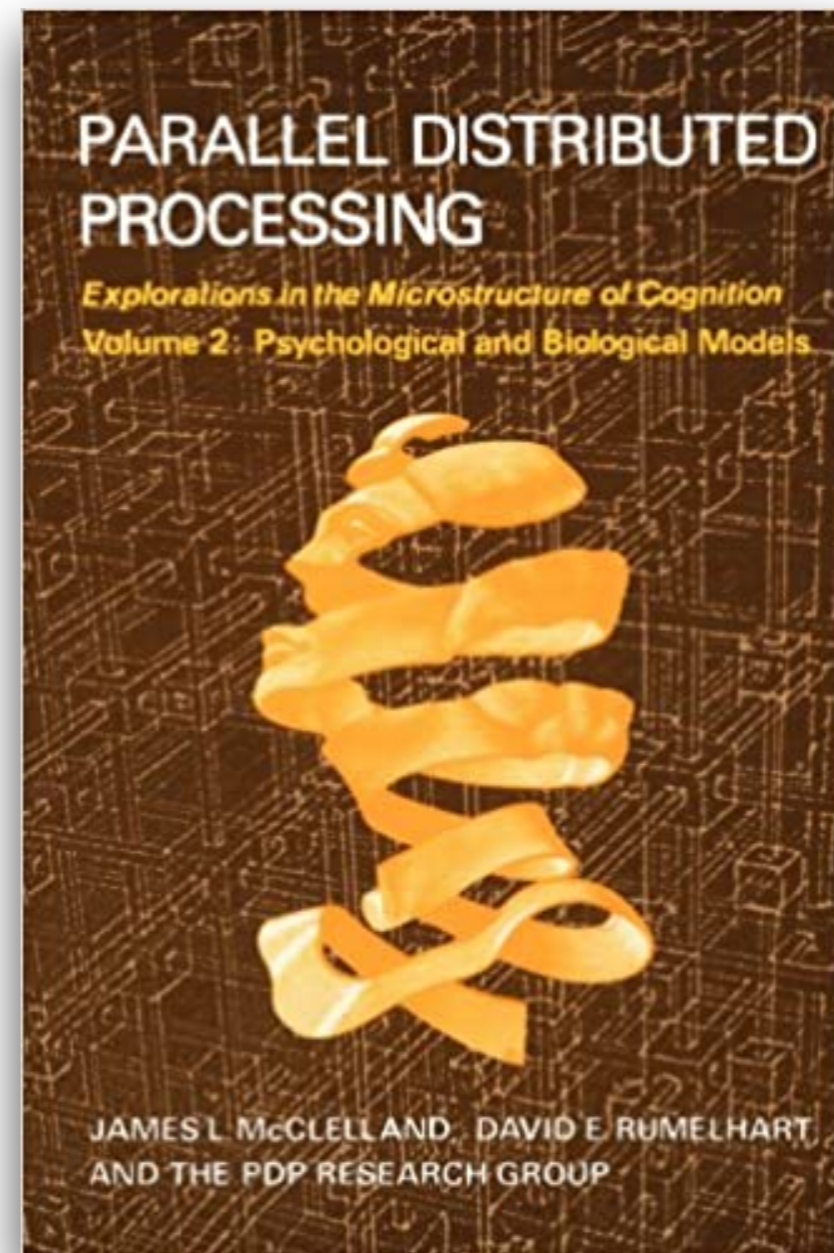
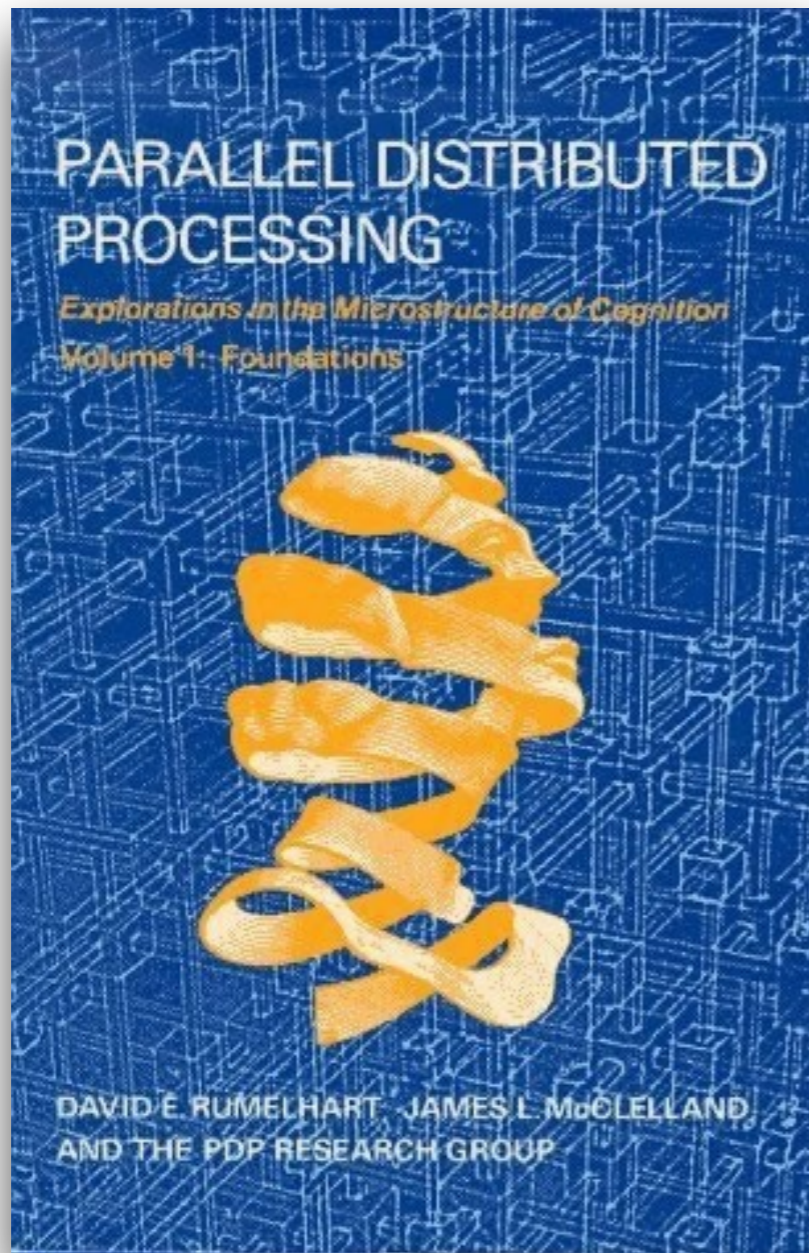


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Connectionism

- ▶ The second wave of neural network research was in 1980s and started in the cognitive science. It was called connectionism or parallel distributed processing.
 - ▶ This followed the first winter (mid 70s-1980).
- ▶ The focus was on devising models of cognition combining symbolic reasoning and artificial neural network models.
- ▶ Many ideas are inspired by Hebb's models.
- ▶ The idea of *distributed representation*, i.e., using the raw data without devising features or pre-categorisation of the inputs was introduced by this research movement.
- ▶ The other key contribution of connectionism was the development of the *back-propagation algorithm* for training neural networks, which is central in deep learning.



delineating the absolute indigeneity of amino acids in fossils. As AMS techniques are refined to handle smaller samples, it may also become possible to date individual amino acid enantiomers by the ^{14}C method. If one enantiomer is entirely derived from the other by racemization during diagenesis, the individual D- and L-enantiomers for a given amino acid should have identical ^{14}C ages.

Older, more poorly preserved fossils may not always prove amenable to the determination of amino acid indigeneity by the stable isotope method, as the prospects for complete replacement of indigenous amino acids with non-indigenous amino acids increases with time. As non-indigenous amino acids undergo racemization, the enantiomers may have identical isotopic compositions and still not be related to the original organisms. Such a circumstance may, however, become easier to recognize as more information becomes available concerning the distribution and stable isotopic composition of the amino acid constituents of modern representatives of fossil organisms. Also, AMS dates on individual amino acid enantiomers may, in some cases, help to clarify indigeneity problems, in particular when stratigraphic controls can be used to estimate a general age range for the fossil in question.

Finally, the development of techniques for determining the stable isotopic composition of amino acid enantiomers may enable us to establish whether non-racemic amino acids in some carbonaceous meteorites²⁷ are indigenous, or result in part from terrestrial contamination.

M.H.E. thanks the NSF, Division of Earth Sciences (grant EAR-8352055) and the following contributors to his Presidential Young Investigator Award for partial support of this research:

Arco, Exxon, Phillips Petroleum, Texaco Inc., The Upjohn Co. We also acknowledge the donors of the Petroleum Research Fund, administered by the American Chemical Society (grant 16144-AC2 to M.H.E., grant 14805-AC2 to S.A.M.) for support. S.A.M. acknowledges NSERC (grant A2644) for partial support.

Received 19 May; accepted 15 July 1986.

1. Bada, J. L. & Protisch, R. *Proc. natn. Acad. Sci. U.S.A.* **70**, 1331-1334 (1973).
2. Bada, J. L., Schroeder, R. A. & Carter, G. F. *Science* **184**, 791-793 (1974).
3. Boulton, G. S. *et al. Nature* **298**, 437-441 (1982).
4. Wehmiller, J. F. in *Quaternary Dating Methods* (ed. Mahaney, W. C.) 171-193 (Elsevier, Amsterdam, 1984).
5. Engel, M. H., Zumberge, J. E. & Nagy, B. *Analyt. Biochem.* **82**, 415-422 (1977).
6. Bada, J. L. *A. Rev. Earth planet. Sci.* **13**, 241-266 (1985).
7. Chisholm, B. S., Nelson, D. E. & Schwarz, H. P. *Science* **216**, 1131-1132 (1982).
8. Ambrose, S. H. & DeNiro, M. J. *Nature* **319**, 321-324 (1986).
9. Macko, S. A., Estep, M. L. F., Hare, P. E. & Hoering, T. C. *Yb. Carnegie Instn Wash* **82**, 404-410 (1983).
10. Hare, P. E. & Estep, M. L. F. *Yb. Carnegie Instn Wash* **82**, 410-414 (1983).
11. Engel, M. H. & Hare, P. E. in *Chemistry and Biochemistry of the Amino Acids* (ed. Barrett, G. C.) 462-479 (Chapman and Hall, London, 1985).
12. Johnstone, R. A. W. & Rose, M. E. in *Chemistry and Biochemistry of the Amino Acids* (ed. Barrett, G. C.) 480-524 (Chapman and Hall, London, 1985).
13. Weinstein, S., Engel, M. H. & Hare, P. E. in *Practical Protein Chemistry—A Handbook* (ed. Darbre, A.) 337-344 (Wiley, New York, 1986).
14. Bada, J. L., Gillespie, R., Gowlett, J. A. J. & Hedges, R. E. M. *Nature* **312**, 442-444 (1984).
15. Mitterer, R. M. & Kraussak, N. *Org. Geochem.* **7**, 91-98 (1984).
16. Williams, K. M. & Smith, G. G. *Origins Life* **8**, 91-144 (1977).
17. Engel, M. H. & Hare, P. E. *Yb. Carnegie Instn Wash* **81**, 425-430 (1982).
18. Hare, P. E. *Yb. Carnegie Instn Wash* **73**, 576-581 (1974).
19. Pillingier, C. T. *Nature* **296**, 802 (1982).
20. Neuberger, A. *Adv. Protein Chem.* **4**, 298-383 (1948).
21. Engel, M. H. & Macko, S. A. *Analyt. Chem.* **56**, 2598-2600 (1984).
22. Dungworth, G. *Chem. Geol.* **17**, 135-153 (1976).
23. Weinstein, S., Engel, M. H. & Hare, P. E. *Analyt. Biochem.* **121**, 370-377 (1982).
24. Macko, S. A., Lee, W. Y. & Parker, P. L. *J. exp. mar. Biol. Ecol.* **63**, 145-149 (1982).
25. Macko, S. A., Estep, M. L. F. & Hoering, T. C. *Yb. Carnegie Instn Wash* **81**, 413-417 (1982).
26. Valentyne, J. R. *Geochim. cosmochim. Acta* **28**, 157-188 (1964).
27. Engel, M. H. & Nagy, B. *Nature* **296**, 837-840 (1982).

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA

† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure¹.

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors². Learning becomes more interesting but

more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input-output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j , to unit j is a linear function of the outputs, y_i , of the units that are connected to j and of the weights, w_{ji} , on these connections

$$x_j = \sum_i y_i w_{ji} \quad (1)$$

Units can be given biases by introducing an extra input to each unit which always has a value of 1. The weight on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights.

A unit has a real-valued output, y_j , which is a non-linear function of its total input

$$y_j = \frac{1}{1 + e^{-x_j}} \quad (2)$$

* To whom correspondence should be addressed.

Second AI Winter and Current AI Summer

- ▶ The second wave of neural networks lasted until mid 1990s.
 - ▶ Loss of interest and lot of disappointment due to unrealistic goals led to a new “winter”.
- ▶ During the second winter, a lot of work continued especially in Canada (and NYU).
- ▶ The summer returned in 2006 when Geoffrey Hinton showed that a particular neural network called a deep belief network could be very efficiently trained (the strategy is called *greedy layer-wise pre-training*).

A fast learning algorithm for deep belief nets *

Geoffrey E. Hinton and **Simon Osindero**

Department of Computer Science University of Toronto
10 Kings College Road
Toronto, Canada M5S 3G4
{hinton, osindero}@cs.toronto.edu

Yee-Whye Teh

Department of Computer Science
National University of Singapore
3 Science Drive 3, Singapore, 117543
tehyw@comp.nus.edu.sg

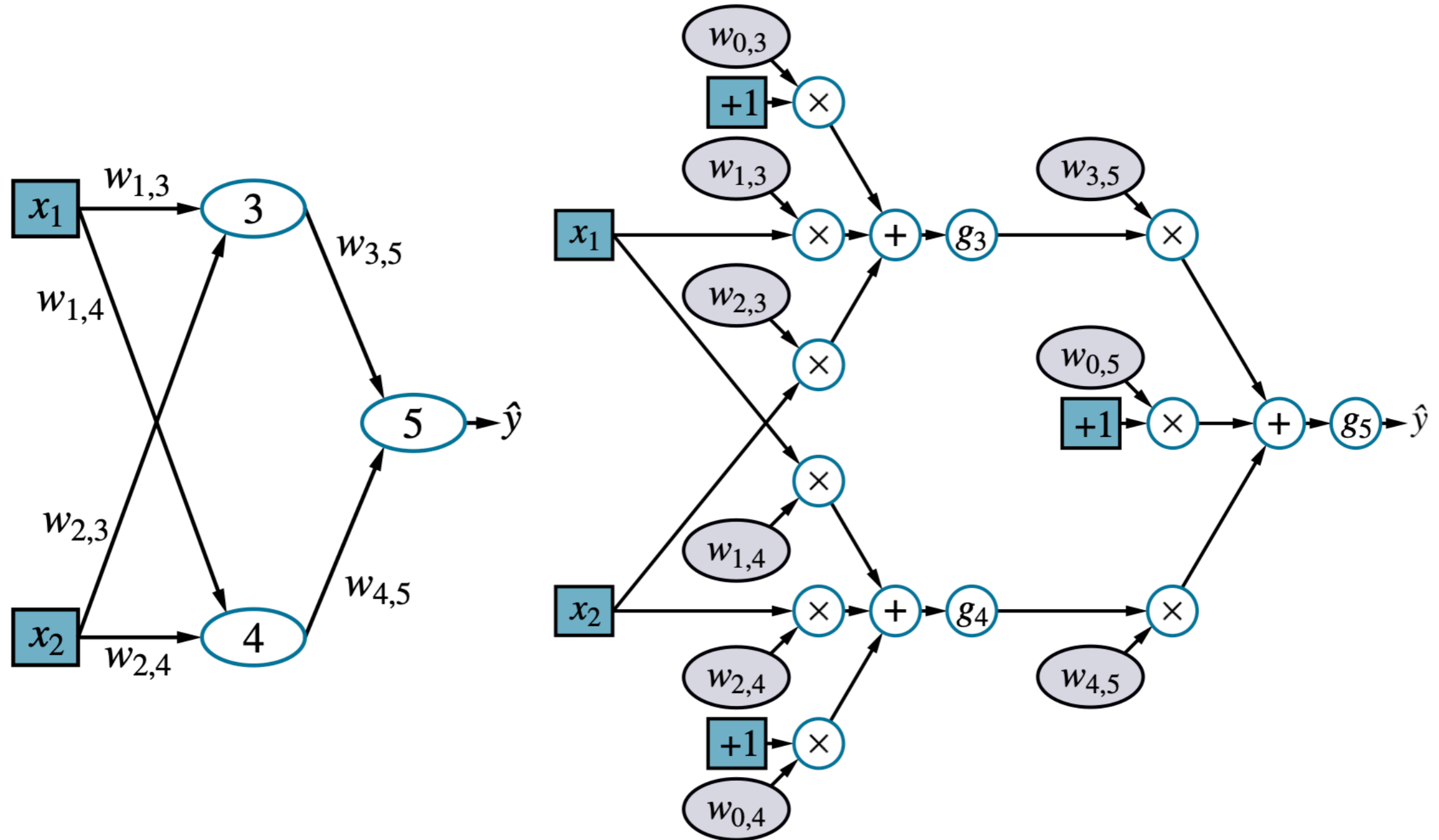
Abstract

We show how to use “complementary priors” to eliminate the explaining away effects that make inference difficult in densely-connected belief nets that have many hidden layers. Using complementary priors, we derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a very good generative model of the joint distribution of handwritten digit images and their labels. This generative model gives better digit classification than the best discriminative learning algorithms. The low-dimensional manifolds on which the digits lie are modelled by

remaining hidden layers form a directed acyclic graph that converts the representations in the associative memory into observable variables such as the pixels of an image. This hybrid model has some attractive features:

1. There is a fast, greedy learning algorithm that can find a fairly good set of parameters quickly, even in deep networks with millions of parameters and many hidden layers.
2. The learning algorithm is unsupervised but can be applied to labeled data by learning a model that generates both the label and the data.
3. There is a fine-tuning algorithm that learns an excellent generative model which outperforms discriminative methods on the MNIST database of hand-written digits.
4. The generative model makes it easy to interpret the distributed representations in the deep hidden layers.
5. The inference required for forming a percept is both fast

Networks and Computational Graphs



From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

Deep Learning Applications

- ▶ The number of application of deep learning is increasing everyday:
 - ▶ Image and video processing and vision;
 - ▶ Machine translation;
 - ▶ Speech generation;
 - ▶ Applications to many scientific fields (astronomy, biology, etc.).
 - ▶ See for example the problem of protein folding.
- ▶ One of the biggest achievement is the extension of the domain of reinforcement learning.
 - ▶ We refer to the convergence of deep learning and reinforcement learning as *deep reinforcement learning*.
 - ▶ Applications of deep reinforcement learning include games, robotics, etc.

Convolutional Networks

0	1	0
1	0	1
0	1	0

128	97	53	201	198
35	22	25	200	195
37	24	28	197	182
33	28	92	195	179
31	40	100	192	177

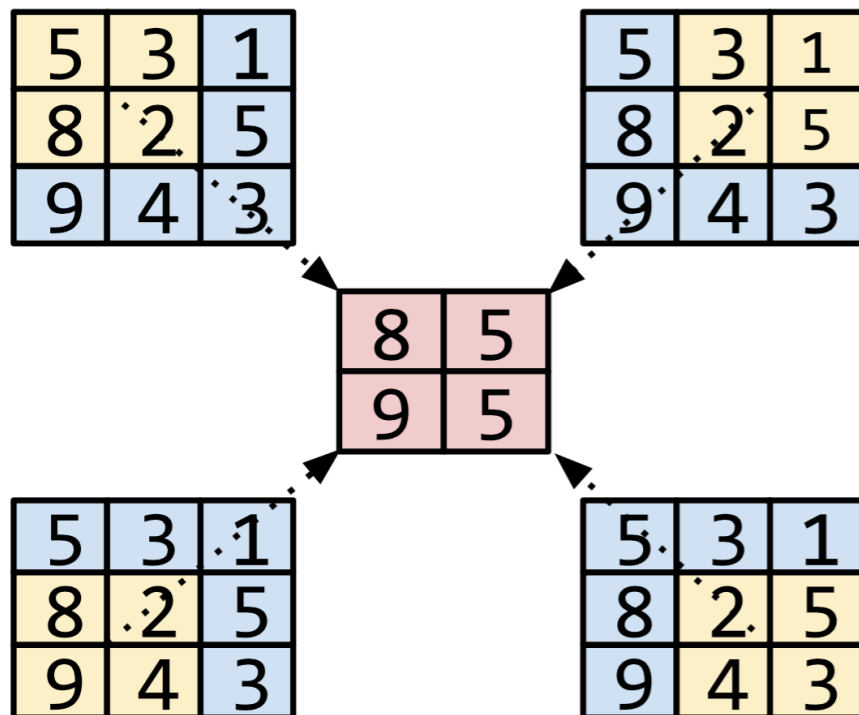
181		

- ▶ *Convolutional networks* are networks that contain a mix of convolutional layers, pooling layers and dense layers.
- ▶ A *convolutional layer* is a layer of a deep neural network, which contains a convolutional filter.
- ▶ A convolutional filter is a matrix having the same rank as the input matrix but a smaller shape.

Convolutional Networks

5	3	1
8	2	5
9	4	3

- ▶ A *pooling layer* reduces a matrix (or matrices created by an earlier convolutional layer) to a smaller matrix. Pooling usually involves taking either maximum or average value across the pooled area.

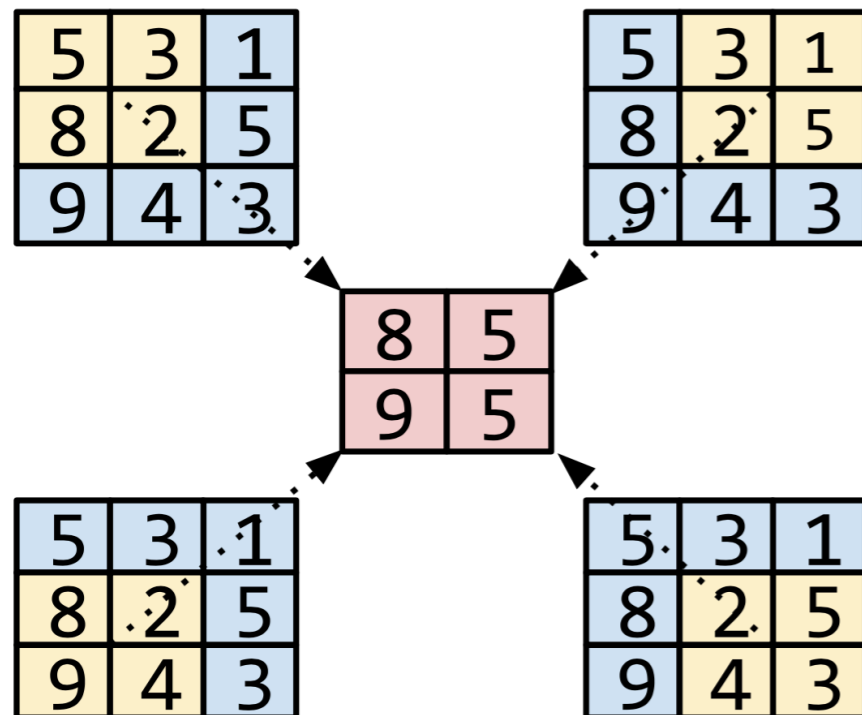


- ▶ A *pooling operation* divides the matrix into slices and then slides that convolutional operation by strides.
- ▶ A *stride* is the delta in each dimension of the convolutional operation.

Convolutional Networks

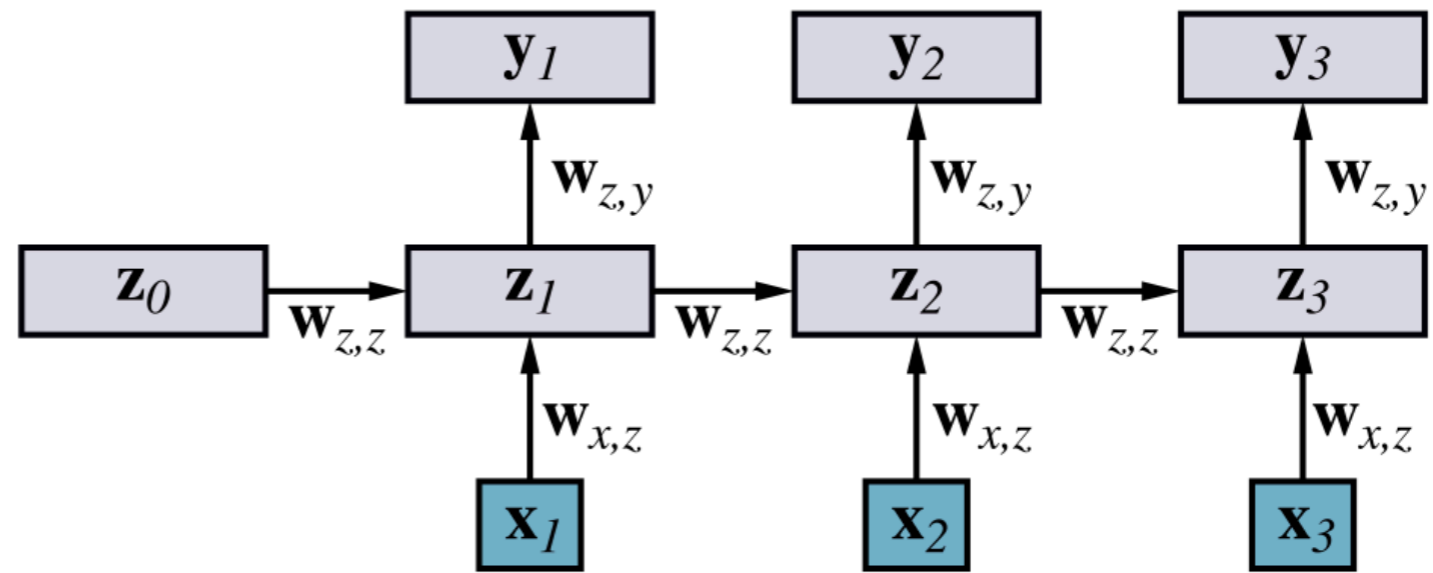
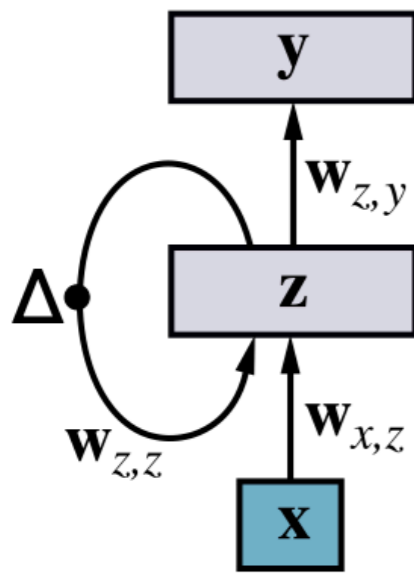
5	3	1
8	2	5
9	4	3

- ▶ Pooling helps enforce translational invariance, which allows algorithms to classify images when the position of the objects within the images change, in the input matrix.



- ▶ Pooling for vision applications is usually called *spatial pooling*.
- ▶ Pooling for time-series applications is usually referred to as *temporal pooling*.
- ▶ You can also hear the expressions *subsampling* and *downsampling*.

Recurrent Neural Networks



Recurrent Neural Networks and NLP

- ▶ In a recurrent neural network language model, each input is encoded as a word embedding vector \mathbf{x}_i .
- ▶ The network is based on a hidden layer with output \mathbf{z}_t , which gets passed as input from one step to the next.
- ▶ We are interested in doing multi-class classification and, therefore, the output \mathbf{y}_t will be a softmax probability distribution over the possible values of the next word in the sentence.

LSTMs

- ▶ It is a kind of RNN that does not suffer from the problem of vanishing gradients.
- ▶ In fact, an LSTM can choose to remember part of the input, copying it over to the next time step and to forget other parts.
- ▶ LSTM stands for long short-term memory. LSTM is a kind of RNN with gating units that does not suffer

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

Sepp Hochreiter

Fakultät für Informatik

Technische Universität München

80290 München, Germany

hochreit@informatik.tu-muenchen.de

<http://www7.informatik.tu-muenchen.de/~hochreit>

Jürgen Schmidhuber

IDSIA

Corso Elvezia 36

6900 Lugano, Switzerland

juergen@idsia.ch

<http://www.idsia.ch/~juergen>

Abstract

Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. We briefly review Hochreiter's 1991 analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called "Long Short-Term Memory" (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing *constant* error flow through "constant error carousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with RTRL, BPTT, Recurrent Cascade-Correlation, Elman nets, and Neural Sequence Chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers— $8\times$ deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis

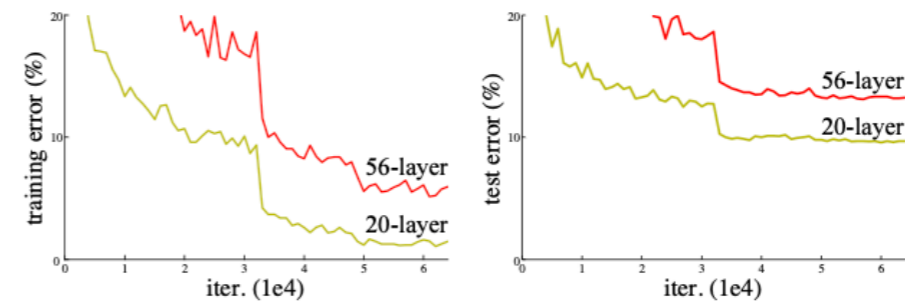
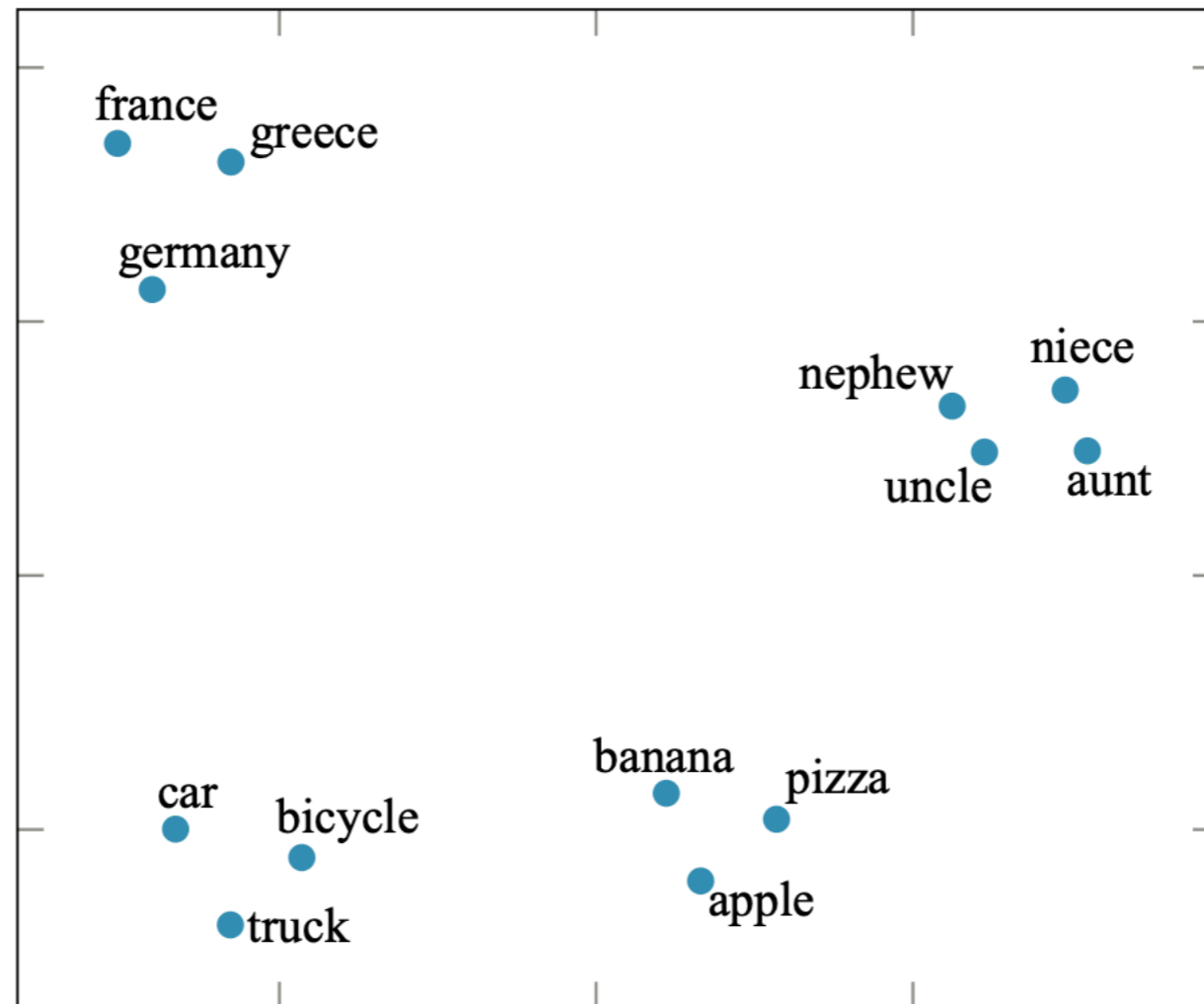


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?* An obstacle to answering this question was the notorious

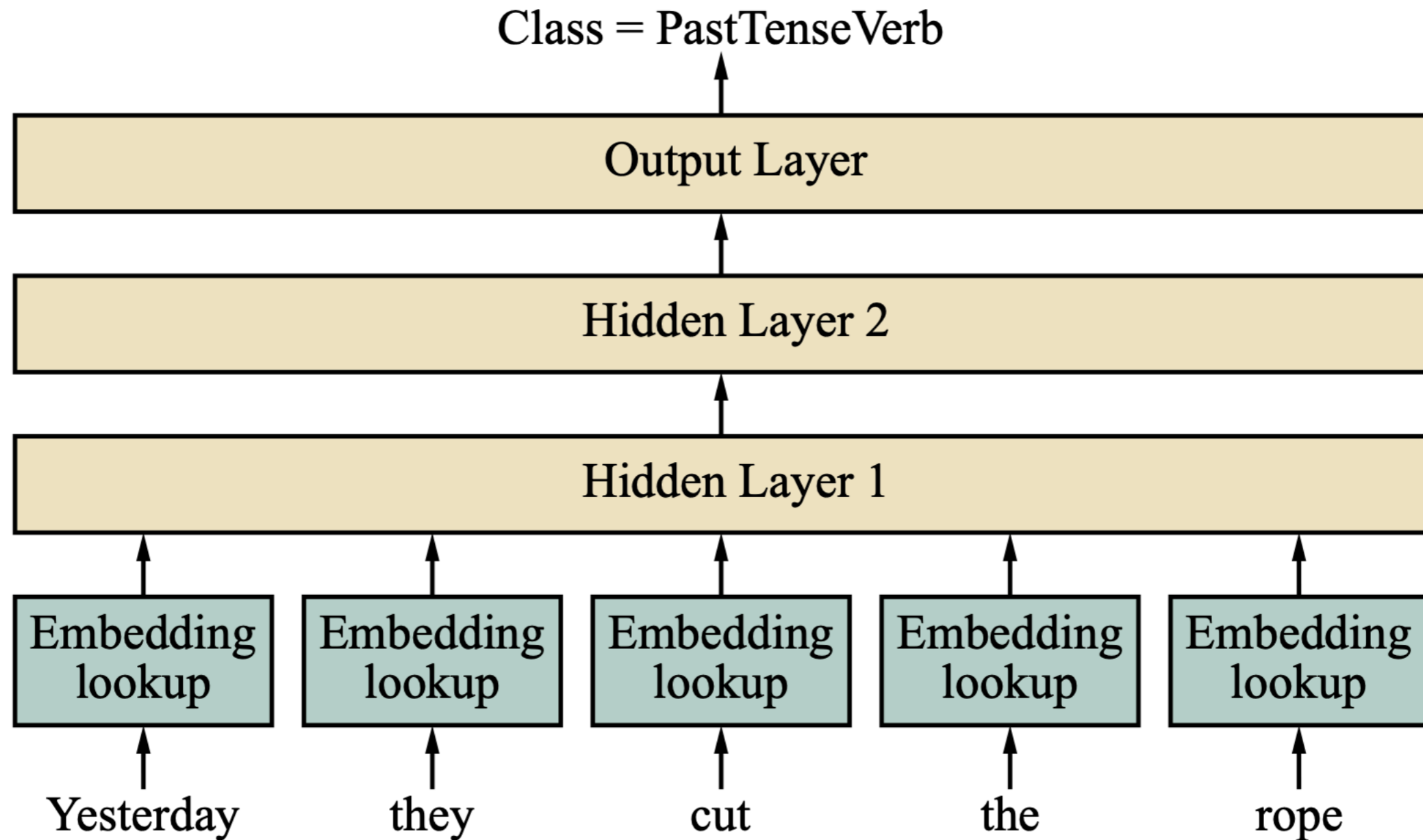
Word Embeddings



France = [0.2, 2.7]
Greece = [0.8, 2.6]
...

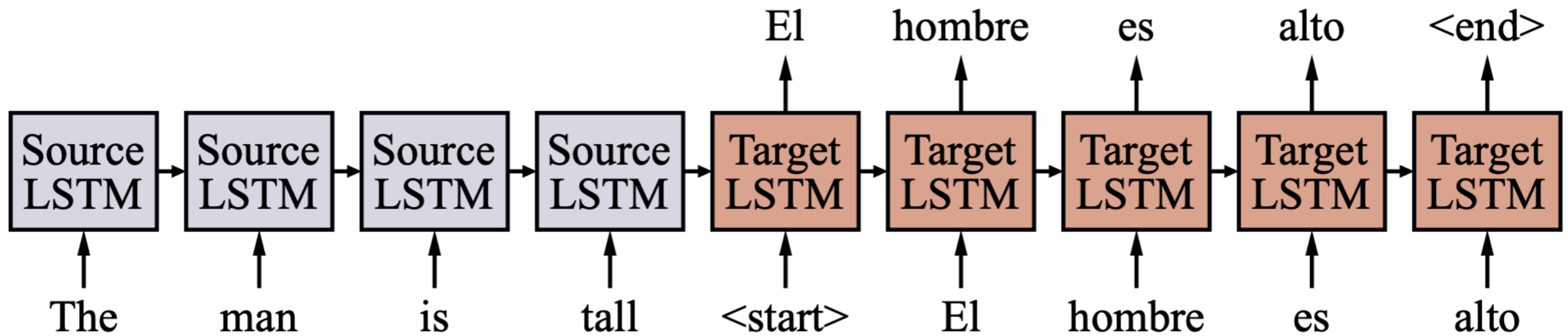
From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

Feedforward Part-of-Speech Tagging Model



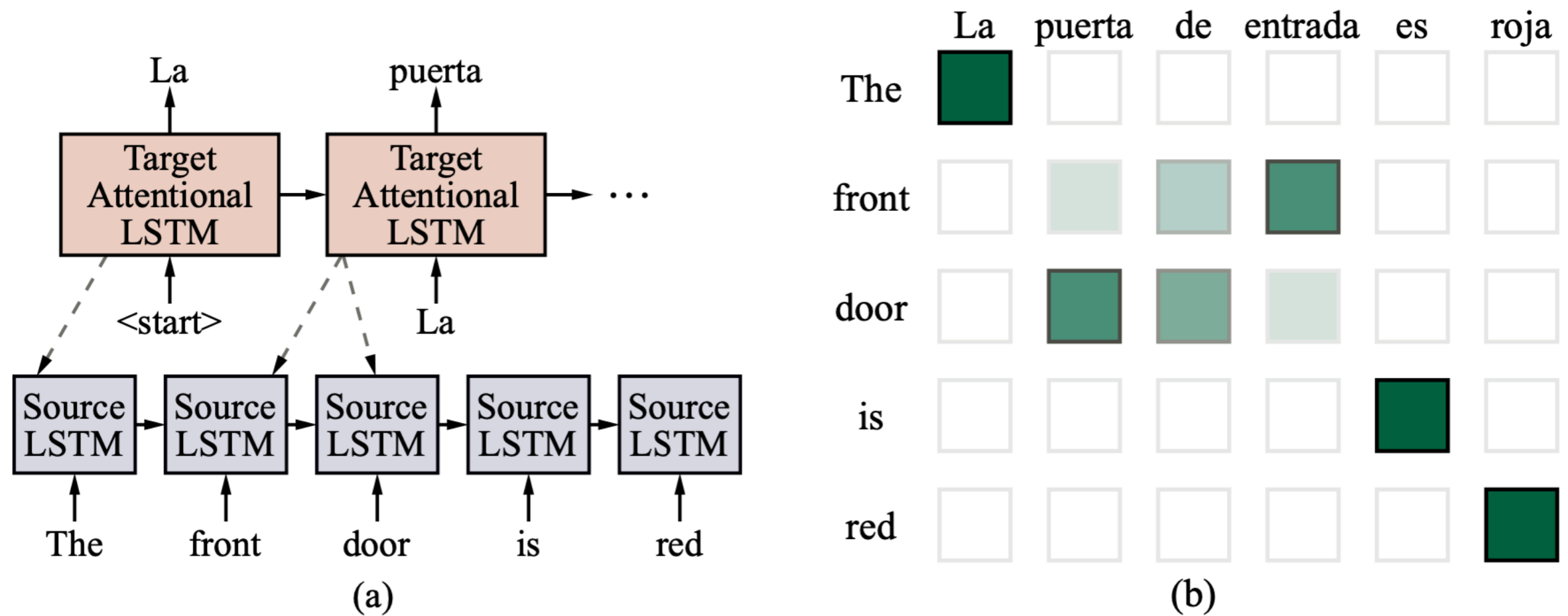
From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

Basic Sequence to Sequence Model



From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

Attentional Sequence-to-Sequence Model



From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

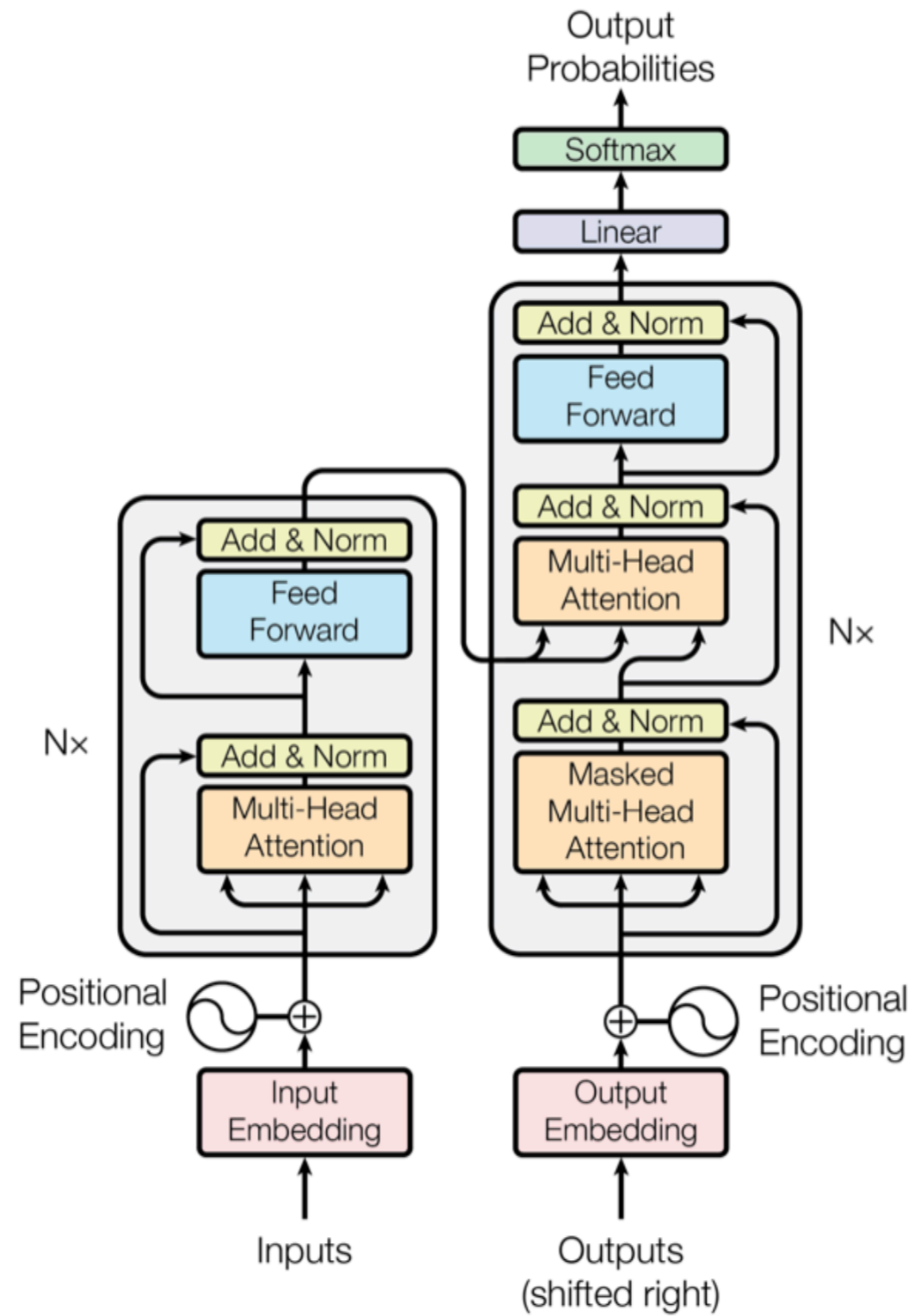
Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Transformers



Technical Preview

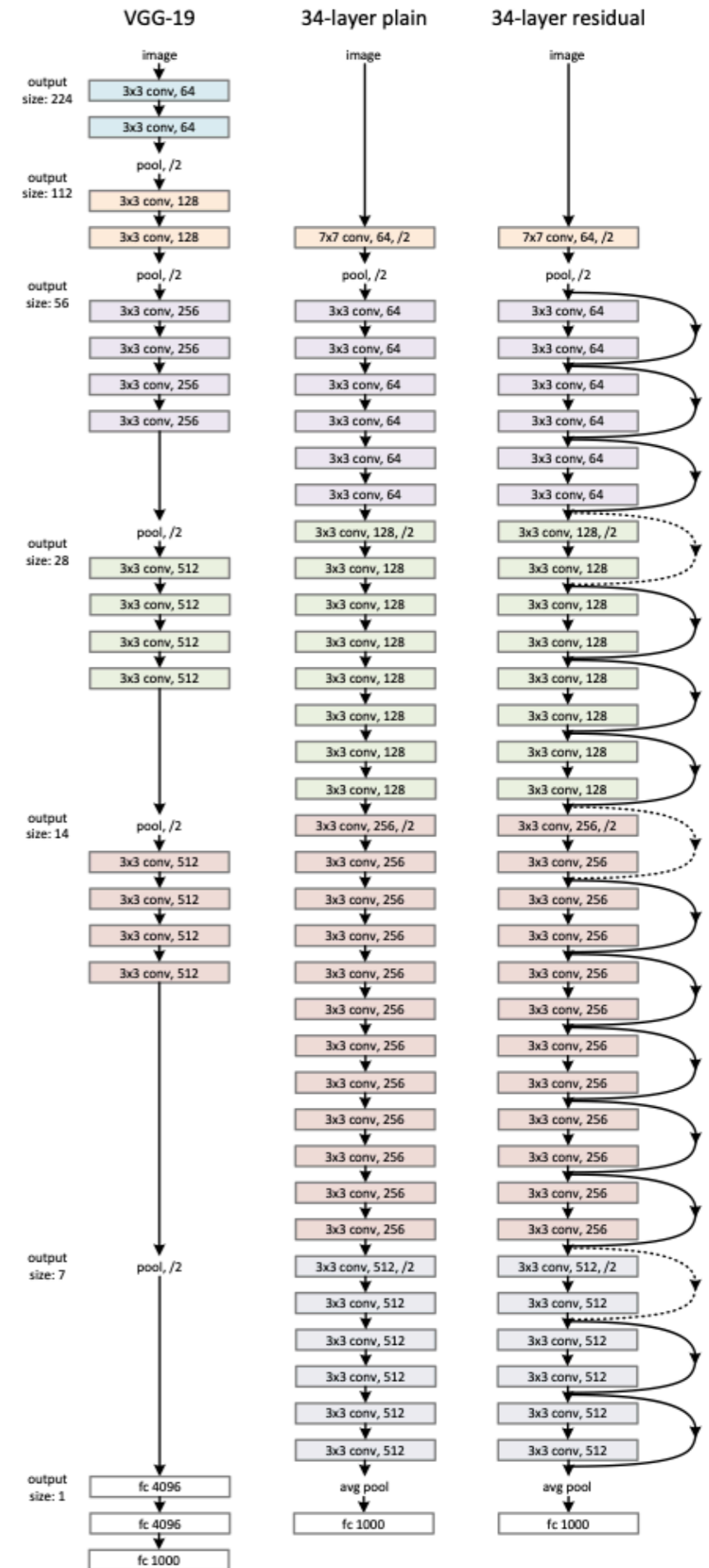
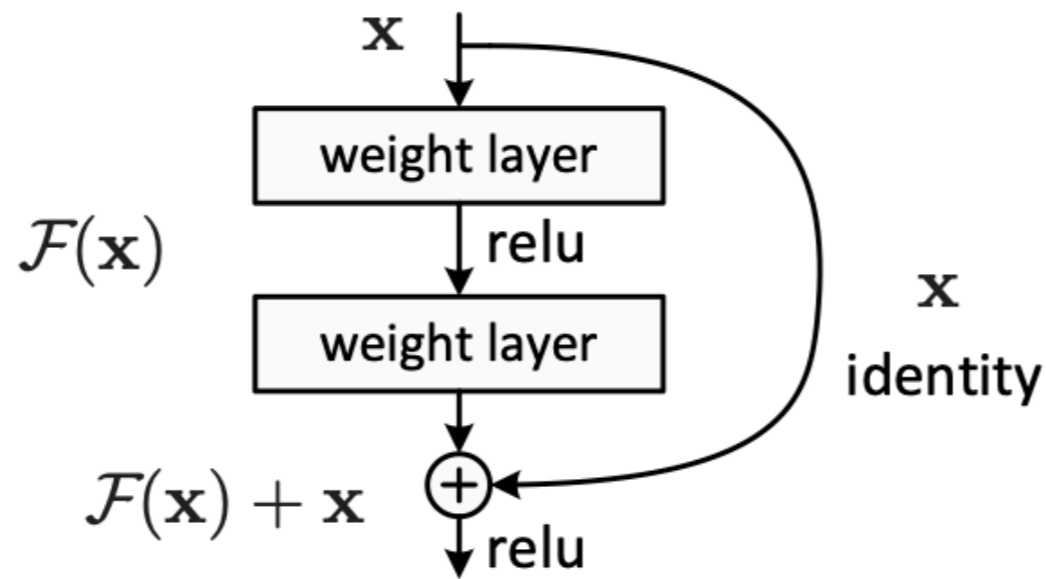
Your AI pair programmer

With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor.

Sign up >

```
sentiment.ts write_sql.go parse_expenses.py addresses.rb
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/` {
```

Residual Networks



GPT-3

- ▶ GPT-3 is an autoregressive language model based on deep learning (transformers).
- ▶ It was introduced in June 2020.
- ▶ GPT stands for Generative Pre-trained Transformer 3.
- ▶ Based on 175 million parameters.
- ▶ Microsoft has licensed exclusive use of GPT-3 in terms of underlying model. There is still a public API.
- ▶ Other large language models exist such as BERT, RoBERTa, etc.

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***
Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**
Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**
Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**
Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**
Benjamin Chess **Jack Clark** **Christopher Berner**
Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kudithipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*¹

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) that are trained on broad data at scale and are adaptable to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotics, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based

Neuroscience and Deep Learning: Some Caveats

- ▶ Neuroscience can be an inspiration, but we should remember that we are trying to “engineer” a system.
- ▶ Actual neurons are not based on the simple functions that we use in our systems.
 - ▶ At the moment, more complex functions haven’t led to improve performance yet.
- ▶ Neuroscience has inspired the design of several neural architectures, but our knowledge is limited in terms of how the brain actually learn.
- ▶ For this reason, neuroscience is of limited help for improving the design of the learning algorithms themselves.
- ▶ Deep learning is not an attempt to simulate the brain!

Deep Learning and Computational Neuroscience

- ▶ At the same time, it is worth noting that there is an entire field of neuroscience devoted to understanding the brain using mathematical and computational models. The area is called *computational neuroscience*.
- ▶ AI and neuroscience are strictly linked and indeed understanding brain biology will lead to improvement in the design of AI systems.
- ▶ This is currently an area of intense research.

Neuroscience-Inspired Artificial Intelligence

Demis Hassabis,^{1,2,*} Dharshan Kumaran,^{1,3} Christopher Summerfield,^{1,4} and Matthew Botvinick^{1,2}

¹DeepMind, 5 New Street Square, London, UK

²Gatsby Computational Neuroscience Unit, 25 Howland Street, London, UK

³Institute of Cognitive Neuroscience, University College London, 17 Queen Square, London, UK

⁴Department of Experimental Psychology, University of Oxford, Oxford, UK

*Correspondence: dhcontact@google.com

<http://dx.doi.org/10.1016/j.neuron.2017.06.011>

The fields of neuroscience and artificial intelligence (AI) have a long and intertwined history. In more recent times, however, communication and collaboration between the two fields has become less commonplace. In this article, we argue that better understanding biological brains could play a vital role in building intelligent machines. We survey historical interactions between the AI and neuroscience fields and emphasize current advances in AI that have been inspired by the study of neural computation in humans and other animals. We conclude by highlighting shared themes that may be key for advancing future research in both fields.

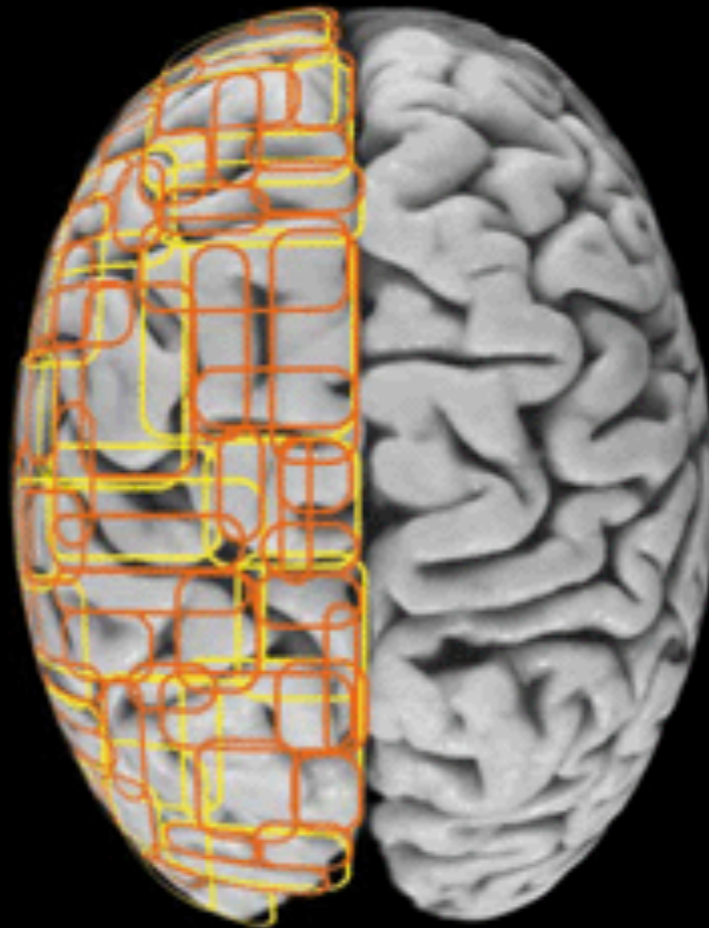
In recent years, rapid progress has been made in the related fields of neuroscience and artificial intelligence (AI). At the dawn of the computer age, work on AI was inextricably intertwined with neuroscience and psychology, and many of the early pioneers straddled both fields, with collaborations between these disciplines proving highly productive (Churchland and Sejnowski, 1988; Hebb, 1949; Hinton et al., 1986; Hopfield, 1982; McCulloch and Pitts, 1943; Turing, 1950). However, more recently, the interaction has become much less commonplace, as both subjects have grown enormously in complexity and disciplinary boundaries have solidified. In this review, we argue for the critical and ongoing importance of neuroscience in generating ideas that will accelerate and guide AI research

tively. For example, if an algorithm is not quite attaining the level of performance required or expected, but we observe it is core to the functioning of the brain, then we can surmise that redoubled engineering efforts geared to making it work in artificial systems are likely to pay off.

Of course from a practical standpoint of building an AI system, we need not slavishly enforce adherence to biological plausibility. From an engineering perspective, what works is ultimately all that matters. For our purposes then, biological plausibility is a guide, not a strict requirement. What we are interested in is a systems neuroscience-level understanding of the brain, namely the algorithms, architectures, functions, and representations it utilizes. This roughly corresponds to

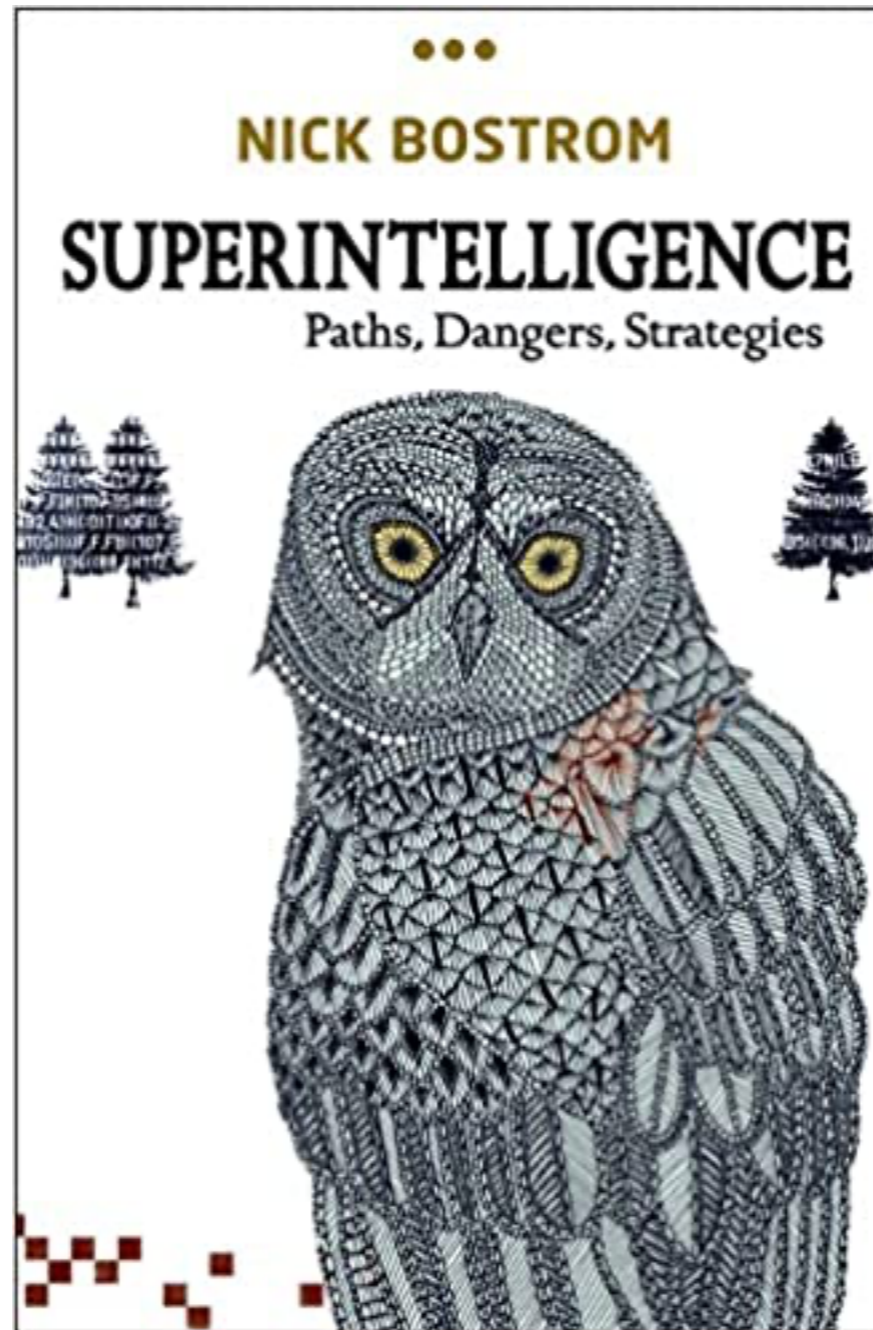
THEORETICAL NEUROSCIENCE

Computational and Mathematical
Modeling of Neural Systems



Peter Dayan and L. F. Abbott

Superintelligence



ARE YOU LIVING IN A COMPUTER SIMULATION?

BY NICK BOSTROM

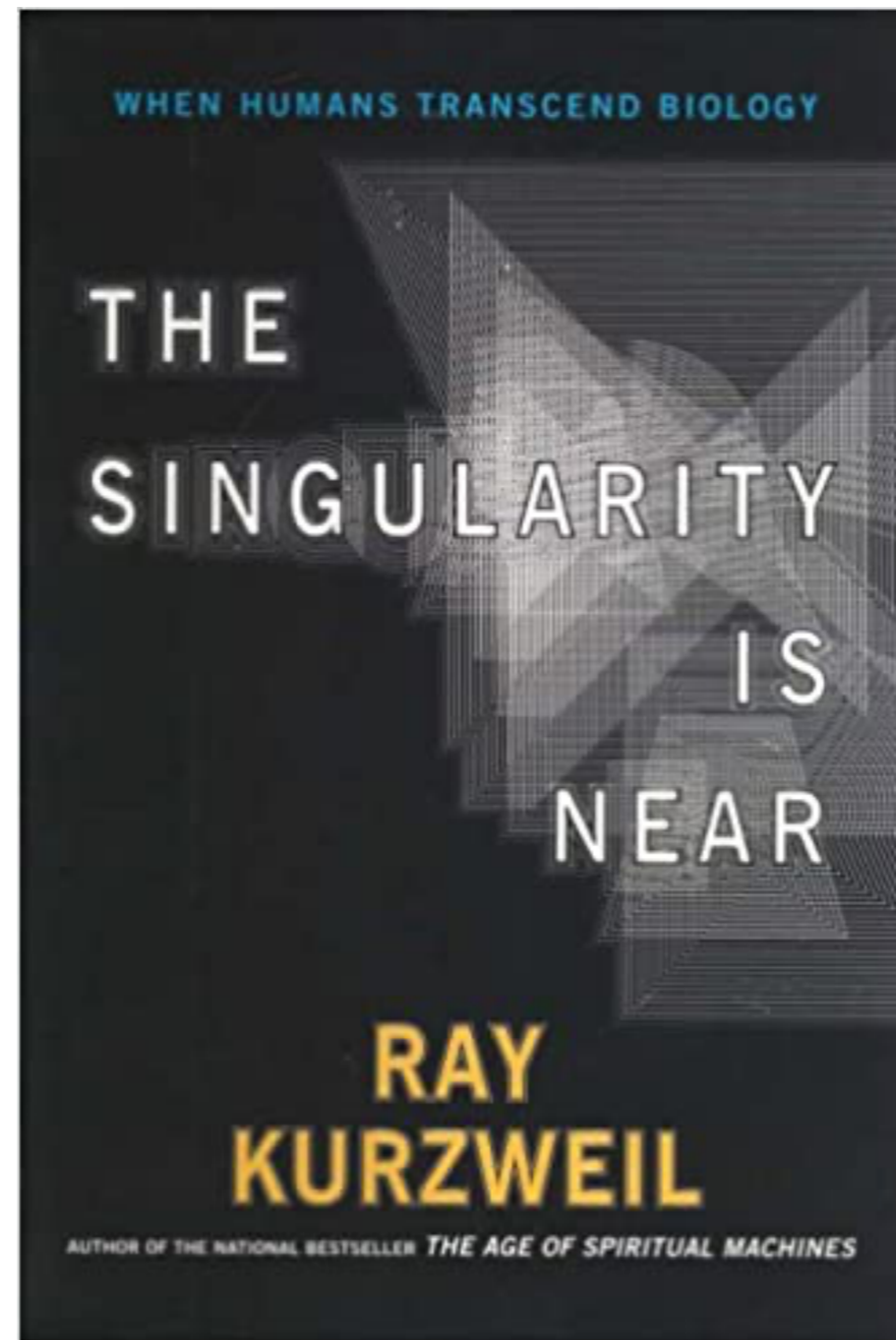
[Published in *Philosophical Quarterly* (2003) Vol. 53, No. 211, pp. 243-255. (First version: 2001)]

This paper argues that *at least one* of the following propositions is true: (1) the human species is very likely to go extinct before reaching a “posthuman” stage; (2) any posthuman civilization is extremely unlikely to run a significant number of simulations of their evolutionary history (or variations thereof); (3) we are almost certainly living in a computer simulation. It follows that the belief that there is a significant chance that we will one day become posthumans who run ancestor-simulations is false, unless we are currently living in a simulation. A number of other consequences of this result are also discussed.

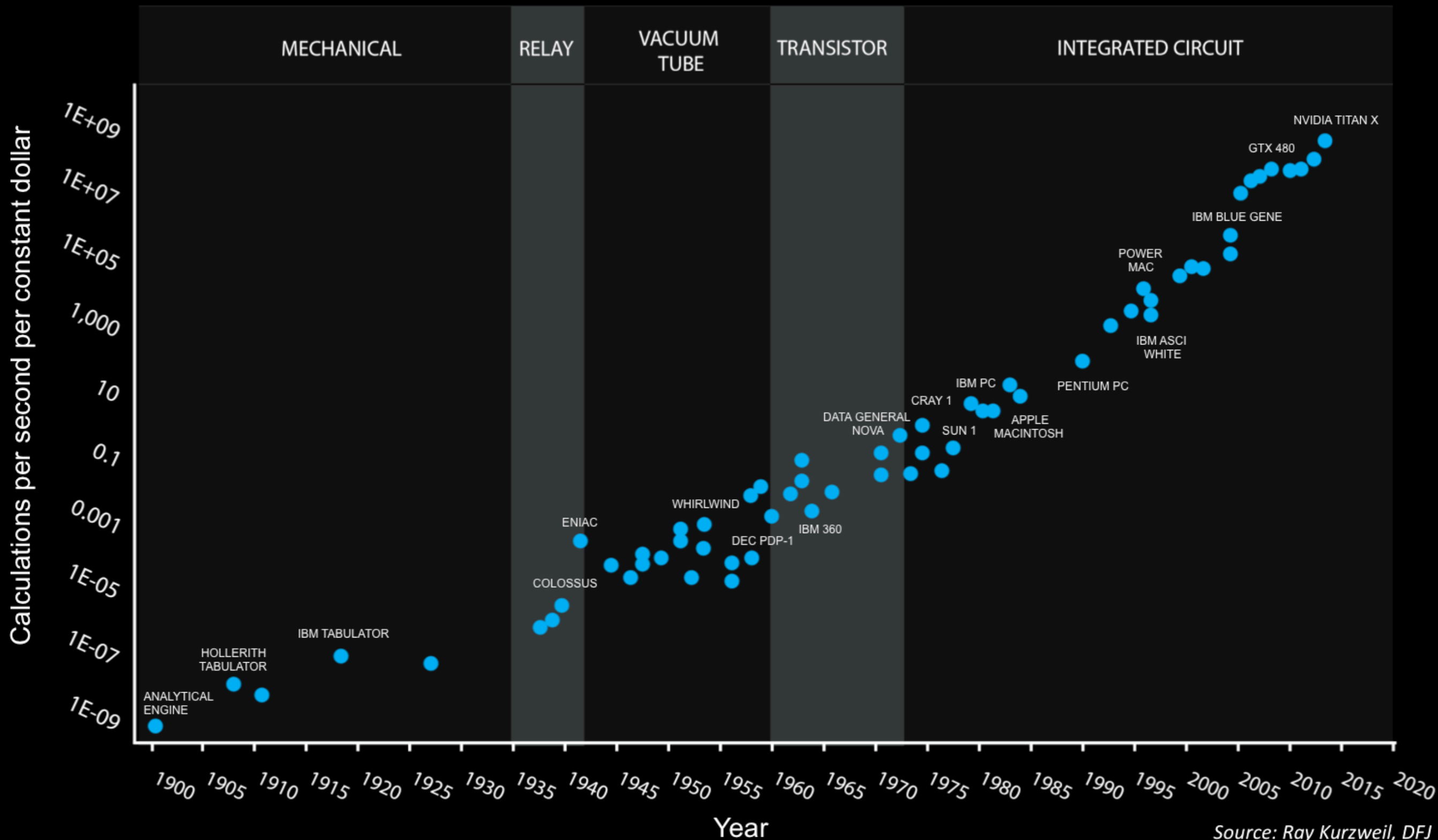
I. INTRODUCTION

Many works of science fiction as well as some forecasts by serious technologists and futurologists predict that enormous amounts of computing power will be

Is the Singularity near?



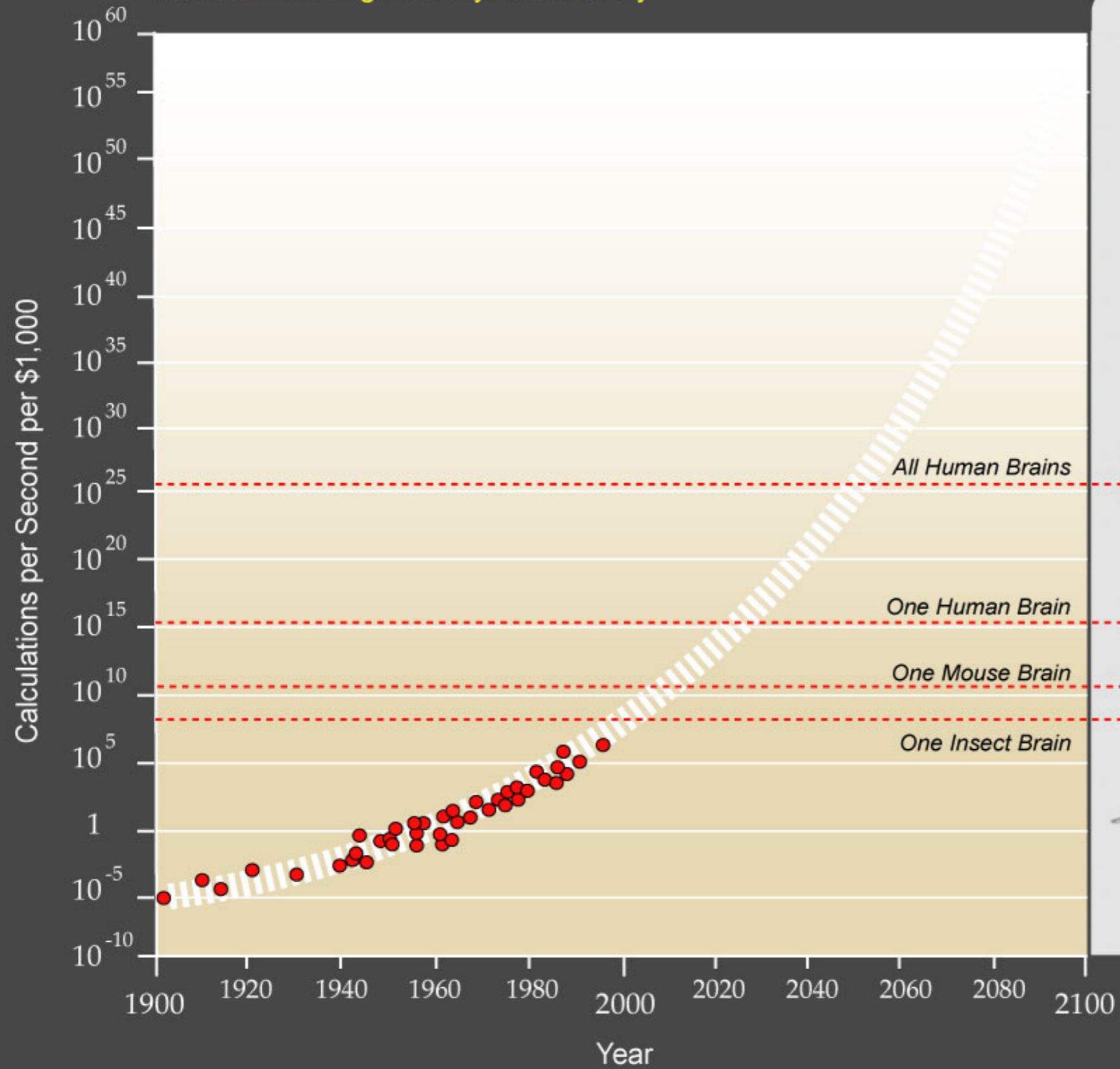
120 Years of Moore's Law



Source: Ray Kurzweil, DFJ

Exponential Growth of Computing

Twentieth through twenty first century



Logarithmic Plot



THE NEW YORK TIMES BESTSELLER



HOW TO
CREATE
A
MIND

THE SECRET OF
HUMAN THOUGHT REVEALED



RAY KURZWEIL

AUTHOR OF *THE SINGULARITY IS NEAR*

Conscious Machines

REVIEWS

Neural correlates of consciousness: progress and problems

Christof Koch¹, Marcello Massimini^{2,3}, Melanie Boly^{4,5} and Giulio Tononi⁵

Abstract | There have been a number of advances in the search for the neural correlates of consciousness — the minimum neural mechanisms sufficient for any one specific conscious percept. In this Review, we describe recent findings showing that the anatomical neural correlates of consciousness are primarily localized to a posterior cortical hot zone that includes sensory areas, rather than to a fronto-parietal network involved in task monitoring and reporting. We also discuss some candidate neurophysiological markers of consciousness that have proved illusory, and measures of differentiation and integration of neural activity that offer more promising quantitative indices of consciousness.

Neural correlates of consciousness (NCC). The minimum neural mechanisms jointly sufficient for any one specific conscious experience.

Being conscious means that one is having an experience — the subjective, phenomenal ‘what it is like’ to see an image, hear a sound, think a thought or feel an emotion. Although our waking experiences usually refer to the external world, we continue to be conscious when we day-dream and during those periods of sleep when we dream¹. Consciousness only vanishes during dreamless sleep or

This Review focuses on visual and auditory studies; for accounts of the NCC for metacognition, body, tactile and olfactory experiences, see REFS 4–7.

Behavioural correlates of consciousness

Although experiences are private, we can usually infer that people are conscious if they are awake and act purpose-

Conscious Machines

Trends in Cognitive Sciences

CellPress

Review

No-Report Paradigms: Extracting the True Neural Correlates of Consciousness

Naotsugu Tsuchiya,^{1,2,*} Melanie Wilke,^{3,4,5} Stefan Frässle,⁶ and Victor A.F. Lamme⁷

The goal of consciousness research is to reveal the neural basis of phenomenal experience. To study phenomenology, experimenters seem obliged to ask reports from the subjects to ascertain what they experience. However, we argue that the requirement of reports has biased the search for the neural correlates of consciousness over the past decades. More recent studies attempt to dissociate neural activity that gives rise to consciousness from the activity that enables the report; in particular, no-report paradigms have been utilized to study conscious experience in the full absence of any report. We discuss the advantages and disadvantages of report-based and no-report paradigms, and ask how these jointly bring us closer to understanding the true neural basis of consciousness.

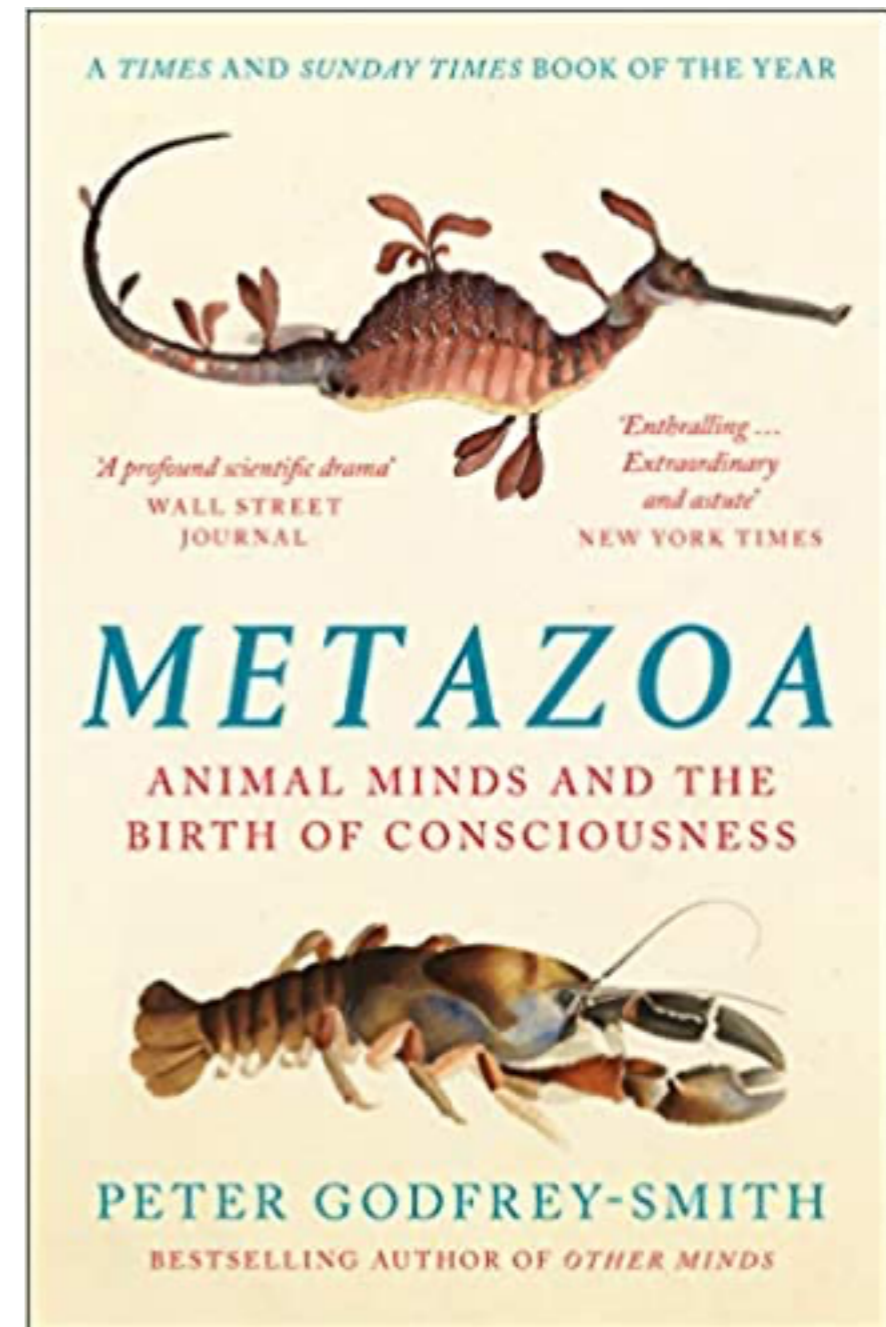
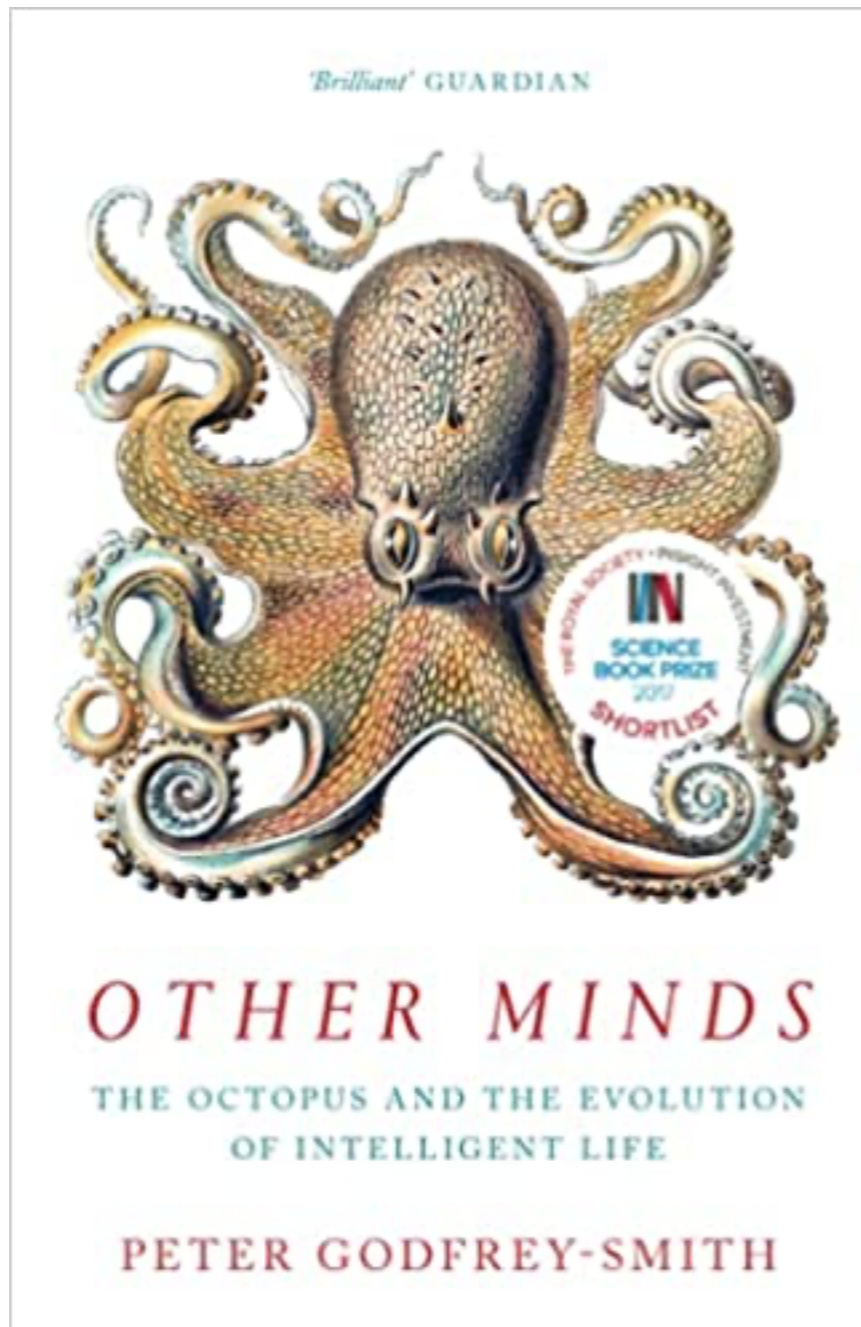
Trends

To study the neural correlates of consciousness (NCC), some forms of behavioral reports from subjects may seem absolutely necessary. However, strong reliance on reports has biased much of the NCC research towards the search for the neural correlates of perceptual reports.

Stringent requirement of behavioral reports not only overestimates the true NCC, owing to the inclusion of the neural correlates of reports, but also underestimates it because there are some aspects of real conscious experience that are fundamentally difficult to

Looking for Consciousness in All The Wrong Places?

Alternative Minds



References

- ▶ Chapters 1, 16 and 20 of Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press. 2016.
- ▶ Google Developers Tutorials: <https://developers.google.com/machine-learning/crash-course>

Attribution Notice

- ▶ Portion of the material in the slides about convolutional networks, recurrent networks are modifications based on work created and [shared by Google](#) and used according to terms described in the [Creative Commons 4.0 Attribution License](#).
- ▶ Source: <https://developers.google.com/machine-learning/glossary/>
- ▶ Attribution license: <https://creativecommons.org/licenses/by/4.0/>

Extra slides

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov

Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen

Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado

Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean

Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words

Part of Speech Tagging

Tag	Word	Description	Tag	Word	Description
CC	<i>and</i>	Coordinating conjunction	PRP\$	<i>your</i>	Possessive pronoun
CD	<i>three</i>	Cardinal number	RB	<i>quickly</i>	Adverb
DT	<i>the</i>	Determiner	RBR	<i>quicker</i>	Adverb, comparative
EX	<i>there</i>	Existential there	RBS	<i>quickest</i>	Adverb, superlative
FW	<i>per se</i>	Foreign word	RP	<i>off</i>	Particle
IN	<i>of</i>	Preposition	SYM	<i>+</i>	Symbol
JJ	<i>purple</i>	Adjective	TO	<i>to</i>	to
JJR	<i>better</i>	Adjective, comparative	UH	<i>eureka</i>	Interjection
JJS	<i>best</i>	Adjective, superlative	VB	<i>talk</i>	Verb, base form
LS	<i>l</i>	List item marker	VBD	<i>talked</i>	Verb, past tense
MD	<i>should</i>	Modal	VBG	<i>talking</i>	Verb, gerund
NN	<i>kitten</i>	Noun, singular or mass	VBN	<i>talked</i>	Verb, past participle
NNS	<i>kittens</i>	Noun, plural	VBP	<i>talk</i>	Verb, non-3rd-sing
NNP	<i>Ali</i>	Proper noun, singular	VBZ	<i>talks</i>	Verb, 3rd-sing
NNPS	<i>Fords</i>	Proper noun, plural	WDT	<i>which</i>	Wh-determiner
PDT	<i>all</i>	Predeterminer	WP	<i>who</i>	Wh-pronoun
POS	<i>'s</i>	Possessive ending	WP\$	<i>whose</i>	Possessive wh-pronoun
PRP	<i>you</i>	Personal pronoun	WRB	<i>where</i>	Wh-adverb
\$	<i>\$</i>	Dollar sign	#	<i>#</i>	Pound sign
“	<i>‘</i>	Left quote	”	<i>’</i>	Right quote
(<i>[</i>	Left parenthesis)	<i>]</i>	Right parenthesis
,	<i>,</i>	Comma	.	<i>!</i>	Sentence end
:	<i>;</i>	Mid-sentence punctuation			

From: Stuart Russell and Peter Norvig. Introduction to Artificial Intelligence. 4th Edition. 2020.