

Machine Learning with Engineering Applications - CPEN 355

Lecture Notes

**Souradeep Dutta
(pronounced as Show-Ro-Deep Duh-tah)**

Contents

1	What is intelligence ?	3
1.1	Key Components of Intelligence	5
1.2	Intelligence : The Beginning (1942-50)	6
1.2.1	Representation Learning	7
1.3	Intelligence : Reloaded(1960-2000)	9
1.4	Intelligence: Revolutions(2006-)	10
1.5	A summary of our goals in this course	11

Chapter 1

What is intelligence ?

Reading :

1. "A logical calculus of the ideas immanent in nervous activity " by [McCulloch and Pitts \(1943\)](#)
2. "Computing machinery and intelligence" by Alan Turing in 1950 ([Turing, 2009](#)).

What is intelligence? It is hard to define, I don't know a good definition. We certainly know it when we see it. All humans are intelligent. Dogs are plenty intelligent. Most of us would agree that a house fly or an ant is less intelligent than a dog. What are the common features of these species? They all can gather food, search for mates and reproduce, adapt to changing environments and, in general, the ability to survive.

Are plants intelligent? Plants have sensors, they can measure light, temperature, pressure etc. They possess reflexes, e.g., sunflowers follow the sun. This is an indication of "reactive/automatic intelligence". The mere existence of a sensory and actuation mechanism is not an indicator of intelligence. Plants cannot perform planned movements, e.g., they cannot travel to new places.

A Tunicate in Fig. 1.1 is an interesting plant however. Tunicates are invertebrates. When they are young they roam around the ocean floor in search of





Figure 1.1: A Tunicate on the ocean floor

nutrients, and they also have a nervous system (ganglion cells) at this point of time that helps them do so. Once they find a nutritious rock, they attach themselves to it and then eat and digest their own brain. They do not need it anymore. They are called “tunicates” because they develop a thick covering (shown above) or a “tunic” to protect themselves.

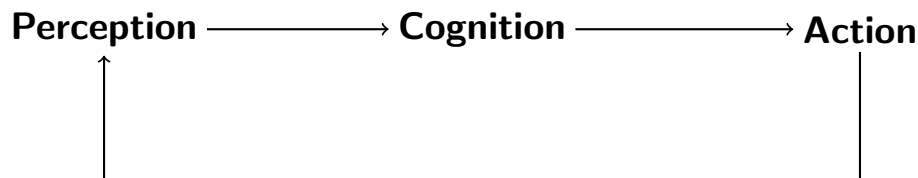
Is a program like AlphaGo intelligent? There is a very nice movie on Netflix on the development of AlphaGo and here’s an excerpt from the [movie](#). The commentator in this video is wondering how Lee Se-dol, who was one of the most accomplished Go players in the world then, might defeat this very powerful program; this was I believe after AlphaGo was up 3-0 in the match already. The commentator says so very nonchalantly: if you want to defeat AlphaGo all you have to do is pull the plug. A key indicator of intelligence (and this is just my opinion) is the ability to take actions upon the world. With this comes the ability to affect your environment, preempt antagonistic agents in the environment and take actions that achieve your desired outcomes. You should not think of intelligence (artificial or otherwise) as something that takes a dataset and learns how to make predictions using this dataset. For example, if I dropped my keys at the back of the class, I cannot possibly find them without moving around, using priors of where keys typically hide (which is akin to learning from a dataset) only helps us search more efficiently.

Is an LLM based software tool like ChatGPT/Gemini intelligent ? The short answer is no. ChatGPT does not understand meaning. It produces language by detecting and reproducing statistical patterns in data, not by forming beliefs, intentions, or comprehension about the world. Intelligence requires understanding ChatGPT has none. Philosopher [John Searle](#), (who recently passed away in 2025 !) proposed the [Chinese room argument](#). Consider the following thought experiment : A person who does not understand Chinese sits in a room. He receives Chinese symbols and follows a rulebook to produce correct responses. To outsiders it looks like the person understands Chinese. But

inside, there is no understanding - only *rule-following*. No matter how sophisticated this rule gets, it is still a rule. In my opinion we should not mistake this for intelligence. Just like no matter how good a store mannequin gets at mimicking human gestures, we can always tell the difference between a real human and a robot.

1.1 Key Components of Intelligence

With this definition, we can write down the three key parts that an intelligent, autonomous agent possesses as follows.



Perception refers to the sensory mechanisms to gain information about the environment (eyes, ears, smell, tactile input etc.). Action refers to your hands, legs, or motors/engines in machines that help you move on the basis of this information. Cognition is kind of the glue in between. It is in charge of crunching the information of your sensors, creating a good “representation” of the world around you and then undertaking actions based on this representation. The three facets of intelligence are not sequential and intelligence is not merely a feed-forward process. Your sensory inputs depend on the previous action you took. While searching for something you take actions that are explicitly designed to give you different sensory inputs than what you are getting at the moment.

This class will focus on learning. It is a component, not the entirety, of cognition.

Learning is in charge of looking at past data and predicting what future data may look like.

Cognition also involves handling situations when the current data does not match past data, etc. To give you an example, arithmetic problems you solved in elementary school are akin to learning. Whereas figuring out that taking a standard deduction when you file your income tax versus itemized deduction is like cognition. [The objective of the learning process is really to crunch past data and learn a prior.](#)

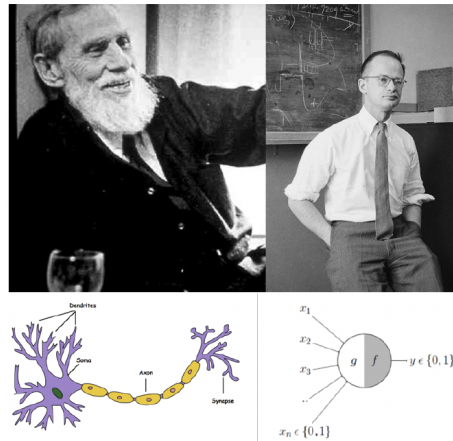
Imagine a supreme agent which is infinitely fast, clever, and can interpret its sensory data and compute the best actions for any task, say driving. Learning from past data is not essential for such an agent; effectively the supreme agent can simulate every physical process around it quickly and so decide upon the

best action it should take. Past data helps if you are not as fast as the supreme agent or if you want to save some compute time/energy during decision making.

A deep network or a machine learning model is not a mechanism that directly undertakes the actions. It is rather a prior on the possible actions to take. Other algorithms that rely on real-time sensory data will be in charge of picking one action out of these predictions. This is very easy to appreciate in robotics: how a car should move depends more upon the real-time data than any amount of past data. This aspect is less often appreciated in non-robotics applications but it holds there as well. Even for something like a recommendation engine that recommends movies in Netflix, the output of a prediction model will typically be modified by a number of algorithms before it is actually recommended to the user, e.g., filters for sensitive information, or toxicity in a chatbot.

1.2 Intelligence : The Beginning (1942-50)

Let us give a short account of how our ideas about intelligence have evolved.



A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

The story begins roughly in 1942 in Chicago. These are Warren McCulloch who was a neuroscientist and Walter Pitts who studied mathematical logic. They built the first model of a mechanical neuron and propounded the idea that simple elemental computational blocks in your brain work together to per-

form complex functions. Their paper (McCulloch and Pitts, 1943) is an assigned reading for this lecture.

VOL. LIX. No. 236.]

[October, 1950]

M I N D

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND
INTELLIGENCE

By A. M. TURING

1. *The Imitation Game.*

Around the same time in England, Alan Turing was forming his initial ideas on computation and neurons. He had already published his paper on computability by then. This paper (Turing, 2009) is the second assigned reading for this lecture.

McCulloch was inspired by Turing's idea of building a machine that could compute any function in finitely-many steps. In his mind, the neuron in a human brain, which either fires or does not fire depending upon the stimuli of the other neurons connected to it, was a binary object; rules of logic 10 where a natural way to link such neurons, just like the Pitt's hero Bertrand Russell rebuilt modern mathematics using logic. Together, McCulloch & Pitts' and Turing's work already had all the terms of neural networks as we know them today: nonlinearities, networks of a large number of neurons, training the weights in situ etc. Let's now move to Cambridge, Massachusetts. Norbert Wiener, who was a famous professor at MIT, had created a little club of enthusiasts around 1942. They would coin the term "Cybernetics" to study exactly the perception-cognition-action loop we talked about. You can read more in the original book titled "Cybernetics: or control and communication in the animal and the machine" (Wiener, 1965). You can also look at the book "The Cybernetic Brain" (Pickering, 2010) to read more.

1.2.1 Representation Learning

Perceptual agents, from plants to humans, perform measurements of physical processes ("signals") at a level of granularity that is essentially continuous. They also perform actions in the physical space, which is again continuous. Cognitive science on the other hand thinks in terms of discrete entities like concepts, ideas, objects, or categories. These can be manipulated with tools of logic and inference. It is useful to ask what information is transferred from the perception system to the cognition system to create such symbols from signals,

or from cognition to control which creates back signals from the symbols? We will often call these symbols the “internal representation” of an agent.

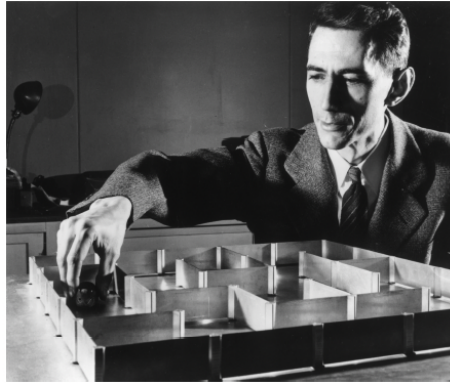


Figure 1.2: Claude Shannon studied information theory. This is a picture of a maze solving mouse that he made around 1950, among the world’s first examples of machine learning; read more [here](#)

Claude Shannon formulated information theory which is one way to study these kind of ideas. Shannon devised a representation learning scheme for compressing (e.g., taking the intensities at each pixel of the camera and encoding them into something less redundant like JPEG), coding (adding redundancy into the representation to gain resilience to noise before transmitting it across some physical medium such as a wireless channel), decoding (using the redundancy to guess the parts of the data 5 packet that were corrupted during transmission) and finally decompressing the data (getting the original signal back, e.g., pixel intensities from JPEG). Information theory as described above is a tool to transmit data correctly between a sender and a receiver. We will use this theory for a different purpose. Compression, decompression etc. care about never losing information from the data; machine learning necessarily requires you forget some of the data. If the model focuses too much on the grass next to the dogs in the dataset, it will “over-fit” to the data and next time when you see grass, it will end up predicting a dog. It not easy to determine which parts of the data one should forget and which parts one should remember.

The study of artificial intelligence has always had this diverse flavor. Computer scientists trying to understand perception, electrical engineers trying to understand representations and mechanical and control engineers building actuation mechanisms.

1.3 Intelligence : Reloaded(1960-2000)

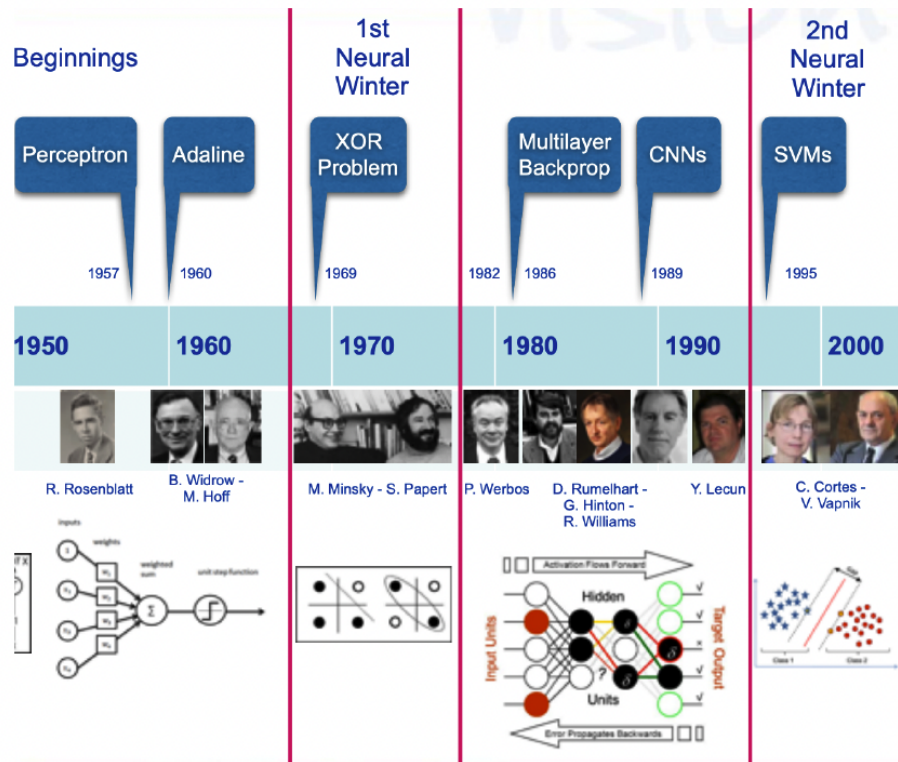
The early period created interest in intelligence and developed some basic ideas. The first major progress of what one would call the second era was made by Frank Rosenblatt in 1957 at Cornell University. Rosenblatt's model called the perceptron is a model with a single binary neuron. It was a machine designed to distinguish punch cards marked on the left from cards marked on the right, and it weighed 5 tons ([Link](#)). The input integration is implemented through the addition of the weighted inputs that have fixed weights obtained during the training stage. If the result of this operation is larger than a given threshold, the neuron fires. When the neuron fires its output is set to 1, otherwise it is set to 0. It looks like the function

$$f(x; w) = \text{sign}(w^\top x) = \text{sign}(w_1 x_1 + \dots + w_d x_d).$$

Rosenblatt's perceptron ([\(Rosenblatt, 1958\)](#)) had a single neuron so it could not handle complicated data. Marvin Minsky and Seymour Papert discussed this in a famous book titled *Perceptrons* ([\(Minsky and Papert, 2017\)](#)). But unfortunately this book was widely perceived as two very well established researchers being skeptical of artificial intelligence itself. Interest in building neuron-based artificial intelligence (also called the connectionist approach) waned as a result. The rise of symbolic reasoning and the rise of computer science as a field coincided with these events in the early 1970s and caused what one would call the "first AI winter".

There was resurgence of ideas around neural networks, mostly fueled by the (re)-discovery of back-propagation by [Rumelhart et al. \(1985\)](#); Shunichi Amari developed methods to train multi-layer neural networks using gradient descent all the way back in 1967 and this was also written up in a book but it was in Japanese ([Amari, 1967](#)). Multi-layer networks came back in vogue because they could now be trained reasonably well. This era also brought along the rise of convolutional neural networks built upon a large body of work starting from two neuroscientists Hubel and Wiesel who did very interesting experiments in the 60s to discover visual cell types ([Hubel and Wiesel, 1968](#)) and Fukushima who implemented convolutional and downsampling layers in his famous Neocognitron ([Fukushima, 1988](#)). Yann LeCun demonstrated classification of handwritten digits using CNNs in the early 1990s and used it to sort zipcodes ([LeCun et al., 1989, 1998](#)). Neural networks in the late 80s and early 90s was arguably, as popular a field as it is today.

Support Vector Machines (SVMs) were invented in [Cortes and Vapnik \(1995\)](#). These were (are) brilliant machine learning models with extremely good performance. They were much easier to train than neural networks. They also had a nice theoretical foundation and, in general were a delight to use as compared to neural networks. It was famously said in the 90s that only the neural network researchers were able to get good performance with neural networks and no one else could train them well. This was largely true even until 2015 or so before the rise of libraries like PyTorch and TensorFlow. So we should



give credit to these libraries for popularizing deep learning in addition to all the researchers in deep learning. Kernel methods, although known much before in the context of the perceptron (Aizerman, 1964; Schölkopf and Smola, 2018), made SVMs very powerful. The rise of Internet commerce in the late 90s meant that a number of these algorithms found widespread and impactful applications. Others such as random forests (Breiman, 2001) further led the progress in machine learning. Neural networks, which worked well when they did but required a lot of tuning and expertise to get to work, lost out to this competition. However, there were other neural network-based models in the natural language processing (NLP) community such as LSTMs (Hochreiter and Schmidhuber, 1997) which were discovered in this period and have remained very popular and performant all through.

1.4 Intelligence: Revolutions(2006-)

The growing quantity of data and computation came together in late 2000s to create ideas like deep Belief Networks (Hinton et al., 2006), deep Boltz-

mann machines (Salakhutdinov and Larochelle, 2010), large-scale training using GPUs (Raina et al., 2009) etc. The watershed moment that got everyone's attention was when Krizhevsky et al. (2012) trained a convolutional neural network to show dramatic improvement in the classification performance on a large dataset called ImageNet. This is a dataset with 1.4 million images collected across 1000 different categories. Performing well on this dataset was considered very difficult, the best approaches in 2011 (ImageNet challenge used to be an annual competition 30 until 2016) achieved about 25% error. Krizhevsky et al. (2012) managed to obtain an error of 15.3%. Many significant results in the world of neural networks have been achieved since 2012. Today, deep networks in their various forms run a large number of applications in computer vision, natural language processing, speech processing, robotics, physical sciences such as physics, chemistry and biology, medical sciences, and many many others (LeCun et al., 2015).

1.5 A summary of our goals in this course

This course will take off from around late 1990s (kernel methods) and develop ideas in deep learning that bring us to today. Our goals are to

1. become good at using modern machine learning tools, i.e., implementing them, training them, modeling specific problems using ideas in ML;
2. understanding why the many quixotic-looking ideas in machine learning works.

After taking this course, we expect to be able to not only develop methods that use machine learning, but more importantly improve existing ideas using foundational understanding of the mathematics behind these ideas and develop new ways of improving machine learning theory and practice.

Bibliography

- Aizerman, M. A. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837.
- Amari, S. (1967). A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119–130.
- Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- Minsky, M. and Papert, S. A. (1974). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA.
- Pickering, A. (2010). *The Cybernetic Brain: Sketches of Another Future*. University of Chicago Press, Chicago.
- Raina, R., Madhavan, A., and Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 873–880.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, Institute for Cognitive Science, University of California, San Diego, La Jolla, CA.
- Salakhutdinov, R. and Larochelle, H. (2010). Efficient learning of deep boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, pages 693–700.
- Schölkopf, B. and Smola, A. J. (2018). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.
- Turing, A. M. (2009). Computing machinery and intelligence. In Epstein, R., Roberts, G., and Beber, G., editors, *Parsing the Turing Test*, pages 23–65. Springer, Dordrecht.
- Wiener, N. (1965). *Cybernetics: Or Control and Communication in the Animal and the Machine*, volume 25. MIT Press, Cambridge, MA.