

On computer science—a turbo in the algorithm

July 4 2017, by Serge Abiteboul And Christine Froidevaux



rbg h j. Credit: Lauri Rantala/Flickr

A new ["Interview on Computer Science"](#). Serge Abiteboul and Christine Froidevaux interview [Claude Berrou](#), computer engineer and electronics engineer, and a member of the French Academy of Sciences. Claude Berrou is a professor at [IMT Atlantique](#). He is best known for his work on turbo codes, which has been used extensively in mobile telephony. His current research focus is on informational neuroscience. This article is

published in collaboration with the [blog Binaire](#).

Binaire: You started out as an electronics engineer, how did you get into computer science?

Claude Berrou: I am a science Rambler. After my initial training at a graduate school that today is called [Phelma](#), I studied a little bit of everything: electronics, signal processing, circuit architecture. Then I got into computer science... by chance, through correction codes and information theory.

What is your definition of computer science?

CB: I have an aphorism: computer science is to the sciences what [natural language](#) is to intelligence. Before computer science, there were equations, formulas and theorems. Computer science allowed sequences of operations, processes, and procedures to be developed to process complex problems. This makes it almost synonymous with language, and it is very similar to natural language, which also requires structure. Just like when we have a common language, computer science offers languages that everyone can understand.

You worked with correction codes. Can you tell us what they are used for?

CB: When we transmit information, we want to retrieve the full message that was sent. Even if we have a lot of users and limited bandwidth. If the message is binary, due to noise and interference disturbing the line, some of the transmitted 0s will be received as 1s, and some of the 1s will become 0s. The greater the noise compared to the signal, the more frequent these kinds of errors happen. The [signal-to-noise ratio](#) can be decreased by poor weather conditions, for example, or disturbances

caused by other communications taking place at the same time. With all these errors, the quality becomes very poor. To prevent this, we encode the transmitted information by adding redundancy. The challenge is to be able to retrieve the message relatively well without adding too much redundancy, without making the message too big. We have a similar problem in mass storage. Bits can switch, sometimes due to wear to the disk. We also introduce redundancy into these systems to be able to retrieve the information.

Talk to us about your wonderful invention, turbo codes.

CB: Turbo codes were born thanks to the Titanic, when we needed to achieve the transmission for viewing the wreck (work by [Alain Glavieux](#)). I played around with ways of reducing the effect of the noise in the transmission, and to deal with the errors, and I thought of introducing the principle of negative feedback in the decoding process, a classic concept in electronics.

For me, the interdisciplinary aspect is fundamental; innovation is often found at the interface of different disciplines. You take an idea that has been proven to work in one area of science, and you try to adapt it to an entirely different context. The original idea behind the turbo codes was to import an electronics technique into computer science.

When we want to create a high-gain amplifier, we put in 2 or 3 of them in a series. But this creates unstable behaviour. To stabilize the arrangement, we implement a negative feedback principle: send a fraction of the amplifier's output back to its input with the "-" sign; this reduces unwanted variations.

I started with a known algorithm: the [Viterbi algorithm](#). It makes it

possible to correct (if there is not too much noise) the errors that occur during transmission through a noisy channel, and can therefore be considered to be a signal-to-noise ratio amplifier. The Viterbi decoder exploits the algebraic law used to design the redundancy of the encoded message and uses it by means of a trellis (the deterministic equivalent of a [Markov chain](#)), thereby delivering the most probable original message. Therefore, I put two Viterbi algorithms in a series. I then tried to integrate the [negative feedback](#) concept into the decoding process. It's a difficult task, and I was not a coding expert.

One problem was that the Viterbi algorithm makes binary choices: the bit was either switched, or it wasn't. Along with a colleague, [Patrick Adde](#) we adapted it so that it would produce probabilistic decisions, which significantly improves the subsequent performance of the decoder.

How does it work?

CB: Like I mentioned, to protect a message, we add redundancy. The turbo code performs the coding in two dimensions. A good analogy is the grid of a crossword puzzle, with vertical and horizontal dimensions. If the definitions were perfect, only one dimension would be enough. We could rebuild the grid, for example, with only horizontal definitions. But since we do not always know what the definitions refer to, and since there can be ambiguities (due to noise, deletions, etc.), we also provide vertical definitions.

The decoding process is a little like what someone does when working on a crossword puzzle. The decoder works in a line (it uses the horizontal definitions), and then moves onto the vertical dimension. Like the crossword fan, the decoder requires several passes to reconstruct the message.

With all of these aspects, the turbo codes are effective.

We believe you. Billions of objects use this technology!

CB: Yes. All media data on 3G and 4G are protected by turbo codes.

This brings us to another Claude: [Claude Shannon](#) and the information theory?

CB: Yes, with this algorithm we clearly enter the realm of the [information theory](#). In fact, I recently helped organize the [symposium at IHP celebrating the centenary of Claude Shannon's birth](#), which was a fascinating symposium.

Shannon demonstrated that all ideal transmission (or storage) should be accomplished using two fundamental operations. First, to reduce the message size, it is compressed to remove the maximum amount of unnecessary redundancy. Next, to protect against errors, intelligent redundancy is added.

Shannon demonstrated the limits of correction codes in 1948! Turbo codes reach Shannon's theoretical limit, to within a few tenths of a decibel!

And now. You have moved on to neuroscience...

CB: My current research is related to informational neuroscience. You recently interviewed Olivier Faugeras, who talked to you about [computational neuroscience](#), a fairly different approach.

My starting point is still information, but this time in the brain. The

human cerebral cortex can be compared to a graph, with billions of nodes and thousands of billions of edges. There are specific modules, and between the modules are lines of communication. I am convinced that the mental information, carried by the cortex, is binary.

Conventional theories hypothesise that information is stored by the synaptic weights, the weights on the edges of the graph. I propose a different hypothesis. In my opinion, there is too much noise in the brain; it is too fragile, inconsistent, and unstable; pieces of information cannot be carried by weights, but rather by assemblies of nodes. These nodes form a clique, in the geometric sense of the word, meaning they are all connected two by two. This becomes digital information.

Is this where we will see coding and redundancy? To prevent information from getting lost in the brain, do redundancies also exist?

CB: Yes. For the traditional, analog school of thought, information is carried by the synapses. In this case, [redundancy](#) could only be achieved using repetitions: several edges would carry the same information.

According to our approach, information is encoded in the connections of a grouping of nodes. Redundancy is naturally present in this type of coding. Take a clique made up of 10 nodes on a graph. You have 45 connections in the clique. This is a large number of connections compared to the number of nodes. I base this on the [Hebbian theory](#) (1949): when neuron A sends spikes and neuron B activates systematically, the connection between A and B will be reinforced if it exists, and if it doesn't exist it will form. Because the clique is redundant, it will resonate, and a modified connection will be reinforced: using Hebbian theory we obtain a reconstruction in the event of deterioration. We have established an entire theory based on this.

You lost us. A clique carries a piece of information. And the fact that the clique features so much redundancy ensures the information will be lasting?

CB: Yes. And furthermore, the clique can be the building block for an associative memory. I will be able to find the complete information based on certain content values. And this is due to the cliques' highly redundant structure.

What does your work involve?

CB: I have set up a multidisciplinary team made up of neuropsychologists, neurolinguists, computer scientists, etc. We are trying to design a demonstrator, a machine based on the model of the brain as we see it, on an informational level. In a traditional computer, the memory is on one side and the processor on the other. In our machine, and in the brain, everything is interlinked.

Based on the theory we are developing (not yet fully published), mental information relies on little pieces of knowledge that are stored in the cliques. The cliques are chosen randomly. But once it has been done, they become permanent. This varies from one person to another; the same cliques do not carry the same [information](#) in different individuals. I would like to develop [artificial intelligence](#) using this machine model.

How do you see artificial intelligence?

CB: There are, in fact, two types of artificial intelligence. First, there is the kind concerned with the senses, with vision and speech recognition, for example. We are starting to be able to do this using deep learning. And then, there is the type that allows us to imagine and create, and know how to answer new questions. For now, we are not able to do this.

In my opinion, the only way to make progress in this strong AI is to base it on the human cerebral cortex.

I am passionate about this subject. I would like to see it advance and continue my research for a long time to come.

This article was originally published on [The Conversation](#). Read the [original article](#).

Provided by The Conversation

Citation: On computer science—a turbo in the algorithm (2017, July 4) retrieved 25 April 2024 from <https://phys.org/news/2017-07-sciencea-turbo-algorithm.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.