

Critical introduction to a neuronal riddle, programmed in Max/MSP

This Max project explores the roots of artificial neural networks (ANNs). By doing so, the known order of the human as programmer and the computer as being programmed is inverted. The aim is that the user tries to think like a computer while the patch is inspired by the biological neural networks that constitute human brains. The interaction between user and programme is mediated by musical and visual feedback, a language both systems can understand.

But why did I create this loop of human and computer communication? Technology has become indispensable in today's everyday life. However, most people are users and not programmers. My intention is therefore twofold: (i) to reflect on one's personal relationship between oneself and technology and (ii) to illuminate our understanding of how computers' think. By doing so, I aim to make programming more accessible and fun to people without any knowledge in this field.

The process of the project started by conducting research on the following four areas: ANNs, sampling, creativity and Max/MSP. ANNs are computing systems which abstract the human brain neural network in regard to information processing (Wu & Feng, 2018). The ANNs operates in three units. There is the input unit which collects data from the outside world, this is represented by the matrix in my patch. The output unit is generated by the system processing result. And finally the hidden unit, which operates between the input and output units (Wu & Feng, 2018). In 1943, Warren McCulloch, a psychologist and Walter Pitts, a mathematician, developed the first conceptual model of an ANN. In their paper, "A logical calculus of the ideas immanent in nervous activity," they describe the M-P model, where neural events and the relations among them can be treated by means of propositional logic, because of the "all-or-none" character of nervous activity¹. Based on the M-P model, Rosenblatt invented in 1957 the Perceptron: "This is an MP neural network model with continuously adjustable weights. After training, it can achieve the purpose of classifying and recognizing a certain input vector mode." (Wu & Feng, 2018, p. 1647).

A sample can be various things; often it is a sound sampling of a continuous waveform recontextualized via digital techniques. I picked the track Girlfriend (feat. Dâm-Funk) from Palms Trax. Sampling records movement of air over time and stores the amplitude of the wave at a particular time, expressed as numbers. A phenomenon called

¹ A neuron either fires or there is no reaction at all, it will not be a graded response.

aliasing can happen, which means that a high frequency signal appears as low frequency after sampling at a too low sampling rate. The Nyquist theorem prevents aliasing by stating that the sampling rate needs to be at least twice the highest frequency contained in the sound of the signal (Burk, 2011). A sound sample is an array of floating point numbers, and needs to be stored in the form of a buffer, so that the computer can have access to it at any point. In 1988, the release of the first Akai MPC-60, a hardware sampler, made sampling accessible to a wider audience. While sampling has influenced all genres of music, with it more copyright problems surfaced.

Another main influence, determining my approach of working with Max, was the documentary “The Creative Brain” from David Eagleman, a neuroscientist. By trying to unravel human creativity, the take home message was that true creativity takes the familiar and combines it in a way, how it was never combined before. Thus, it creates something new and original. I find this approach extremely helpful for working with Max. You can use and combine many great inspirations in new ways, so that it suits and enriches your concept. When it comes to Max/MSP, I used the programme itself, YouTube tutorials and internet forums for inspiration, help and troubleshooting.

After my search for inspiration and the contextual background of the elements I decided to work with (ANNs and sampling), I started off my patch by programming a perceptron. My main reference for this was a YouTube tutorial by Samuel Pearce-Davies², where he models a single neuron with three inputs and one output. By using a matrix, we can train the neuron to recognize patterns. Each input gets a random weight between 1 and -1. The weights reflect the connection strength between cells in the human brain. The neuron’s output is calculated from a training set example, and with it the error, which is the difference between the neuron’s output and the desired output in the training set example. The error determines the direction of the weight adjustment. The weights are adjusted slightly in over 10,000 trials in order to reach an optimum. In a new situation, following the same pattern, the neuron should have learned to make a good prediction (Spencer-Harper, 2015).

Since one of my aims was to make programming more accessible and fun, the patch desperately needed some more sparkle. Thinking in programming language usually is not something that comes natural to a person. Therefore, music is the perfect mediator, a language the programme and the user can easily understand. The “groove~” object in Max

² <https://www.youtube.com/watch?v=yGWMSkaCoS0&t=1516s>

offers good control over the playback speed of the sample. Ideal musical communication would imply a direct transmission from programme to listener. Believing that “a given aesthetic tendency can be seen as confronting its opposite, and examining this tension can sometimes lead to insight” (Roads, 2015, p. 20), I decided to choose the biggest difference for the playback speed, namely normal and reverse speed. Therefore, the focus is on learning and not differentiating between similar playback speeds.

After some critical evaluation of the patch, I decided to increase the experience of cognitive response to a perceptual reaction by adding a video to the patch. Hence, I am trying to aim for bigger learning outcomes by making it more exciting.

Finally, by taking the familiar, a perceptron, a sample and a video, I committed myself to the creative process by combining them through the neurons output and logic arguments to something new: a riddle. The user has to try and think like a computer in order to find a logic behind possible matrix patterns, while the patch is inspired by a human neuron. With the help of musical and visual mediation, both instances can learn from each other.

Bibliography

- Wu, Y. chen, & Feng, J. wen. (2018). Development and Application of Artificial Neural Network. *Wireless Personal Communications*, 102(2), 1645–1656.
<https://doi.org/10.1007/s11277-017-5224-x>
- Burk, P., Polansky, L., Repetto, D., Roberts, M., & Rockmore, D. (2011). *Music and Computers. A Theoretical and Historical Approach*. Chapter 2.3 and 5.1. Retrieved from <http://sites.music.columbia.edu/cmc/MusicAndComputers/>
- Spencer-Harper, M. (2015). *How to build a simple neural network in 9 lines of Python code*. Retrieved from <https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1>
- Roads, C. (2015). *Composing Electronic Music: A New Aesthetic*. Oxford, New York: Oxford University Press. Chapter 2.