Sharp-Brain

Tom Rochette <tom.rochette@coreteks.org>

November 2, 2024 — 36c8eb68

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

The goal of this article is to detail my process of developing an initial implementation of a "brain" in C#. As a initial version, I want to lay down the tools that will help me build such thing so that it may be decomposed into libraries of reusable components.

The first thing I did was to build a complete body in an object oriented fashion. Body, head, brain, eyes, ears, nerves and so on. The head had eyes, which were attached to their own optic nerve, which would then connect to the brain. The brain was connected to a set of nerves, such as the optic nerve, which could be used either to receive or transmit data/signal.

Since I was using a multithreaded language, I made each input organ a thread of its own. Thus, there would be 2 threads for the eyes (one per eye), 2 threads for the ears (one per ear), a thread for the brain, and so on. This would be the model of a human internal communication and computation system.

However, it is possible to generalize the idea to a robot. If we replace the eyes with a single webcam, then the architecture would be similar in the sense that the eye, now a webcam, would take a frame/image and transmit it over the optic nerve to the brain. The brain would receive this array of bytes (the quantized signal) and process it (whatever that would mean, at this point no processing was done within the brain other than receive the data).

The following depicts the embodiment structure as of its latest implementation (2016-05-04).

```
Embodiment
    Body.cs
    Brain.cs
    Ear.cs
    Environment.cs
    Eye.cs
    Head.cs
    Nerve
        AbducensNerve.cs
        AccessoryNerve.cs
        FacialNerve.cs
        GlossopharyngealNerve.cs
        HypoglossalNerve.cs
        MandibularNerve.cs
        MaxillaryNerve.cs
        Nerve.cs
```

```
OculomotorNerve.cs
OlfactoryNerve.cs
OphthalmicNerve.cs
OpticNerve.cs
TrigeminalNerve.cs
TrochlearNerve.cs
VagusNerve.cs
VestibulocochlearNerve.cs
```

Once the initial components were created, the main task was to find how they would all communicate with one another, or more specifically, who would be responsible of wiring the components. What I decided to do at this point was simply to have the `Head` class be responsible of instantiating all the related brain nerves and head organs (eyes and ears). Then, each nerve would be added to the brain as part of its set of nerves (all nerves implemented the same interface and were not treated in any special manner).

The idea is that an input organ (such as the eyes, the ears, the tongue and so on) produces a `Signal` that is pushed into the nerves queue (a `Nerve` is considered as a simple queue) for the brain (or any receiving end) to process at the desired time. For instance, the eye, being simulated by a webcam, receives 24 x (640px * 480px) frames per second of data that it converts into an array of bytes (a bitmap) which is then encapsulated into a `Signal` object and pushed onto the `OpticNerve`.

At the other end, the `Brain` processes its `Nerve` set and dequeues signals. This new signal is then converted into an `InputTask` which encapsulates the data that was received. The idea in this case is that we assume the brain to be an homogeneous processor, such that given a stream of data (vector of integers), it will be able to make "sense" of it (in other words, this is where the magic is supposed to happen).

After `InputTask` have been created, the `Brain` moves into its `Process` procedure where it will treat any type of tasks (`InputTask` being a specialization of a `Task`). In the case of the `InputTask`, this is where we would have some program attempt to make sense of the signal. One good candidate for this would be a neural network. We could furthermore create specialized `InputTask` that would deal with image/audio/video/taste/smell/temperature/etc. if it is necessary.

Once the `Process` procedure is complete, we move onto the last procedure, `Output`. At this point, if we have any `OutputTask` in the task queue, we will process them. At this point in time (2016-05-04), nothing was implemented. One would have to consider competing and conflicting output tasks or use some sort of attention model which would only allow a specific set of tasks to output during a certain period of time.

The next step after all this was put in place was to introduce the capability to interact with the `Brain` through a web service. It would then be possible to fetch the state of the brain (number of queued tasks, number of processed tasks since the start of the executable, how long it had been alive and how much tasks were processed per second).

2 Post-mortem

Overall the Sharp-Brain project was an interesting way to get familiar with the various nerves within the brain. It was also possible to discover certain particular aspects of real-time AI such as the requirement for the creation of `Thread` with a `Wait` method to prevent it from executing for too long. It was already assessed in the PHP-Brain project that this would be an important aspect of processing real-time information.

The processing model implemented in Sharp-Brain considered that the brain was a monolithic unit that would process all incoming signal as a single step. However, current cognition science separates the brain into 4 regions (called lobes): frontal, parietal, temporal and occipital. Each of these lobes are responsible for different things, such as the occipital lobe for processing visual information. This could point out to two potential models: processing multiple input at once (multithreaded signal processing instead of sequential) or splitting the brain into components which would then process information in parallel between them but sequentially internally.

Although the idea of replicating the human body in a virtual environment is interesting, I think it's distracting us from the issue of figuring out how processing really is done. The part where I said "this is where the magic is supposed to happen" is the part that needs to be worked out. It is however definitely worth trying to understand the context in which this information is received by the brain.

3 Follow-up

Nerves are currently only information transmission queues, but they are also supposed to be part of the information processing sequence. It would thus be appropriate to determine the processes/tasks they accomplish individually before being received by the brain for further treatment.

An initial procedure for processing visual signal would have to be implemented. The difficulty here is that visual processing is currently assumed as being a multi-stage process[1].

Various web service endpoint can be added to control the brain internal state, such as resetting is task queue, flushing the nerves queues, saving and restoring a brain state, etc.

This prototype did not attempt to think of ways it would be possible to interact with the agent in order to provide it with tasks. It assumed that given an environment with inputs (webcam that would stream video and audio), something interesting might happen. With some additional work it would be able to output actions to act in the environment it was provided, namely transmit some output signals. The brain would thus be connected to some output organs (such as a video/audio output or a text output) and would be able to produce output in order to interact with that environment. It would also have as a task to understand what is the result its output signals so that it may reuse them appropriately in the future (for example, how to generate a specific sequence of sounds as to create the speech "I am hungry").

4 See also

5 References

[1] https://en.wikipedia.org/wiki/David_Marr_(neuroscientist)#Stages_of_vision