1 Experiments

1.1 Setup

With the background knowledge of neural oscillations and their possibly important role in doing modular tasks, it could be interesting to see how new methods that use these observations perform against other known modular methods.

In this section I therefore introduce two new methods that is based on neural oscillations, MiO-HyperNEAT and MaO-HyperNEAT which is explained in the beginning of this sections. Their results will be compared against a modular method by Schrum[ref] that is based in Hyper-NEAT and another simple network that doesn't have any modularity.

1.2 MB-Hyper-NEAT [ref]

1.2.1 Microscopic Oscilliations HyperNEAT (MiO-HyperNEAT)

The general idea with this method is to try and mimic the oscillation that happens on the microscopic level of the brain and use that to modularize the network. As mentioned earlier, the microscopic oscillations (MiO) is the firing patterns that take place for each neuron. To mimic this, MiO-HyperNEAT uses HyperNEAT together with SUPG to evolve oscillations for each hidden neuron placed on the substrate.

Along with the oscillation evolved for each hidden neuron the network on the substrate has an extra output that generate a *Master Frequency*. The master frequency, isn't evolved using a SUPG, but is instead generated as a normal output (that if necessary can be modulated onto some pre-defined function - ex. sine function). The idea with the master frequency is to generate a frequency that is dependent on the environment, whereas the neuron oscillations only are determined by their position on the substrate.

Each hidden neuron then compare their oscillations with the master frequency, and if they are in synchrony they are enabled, and if not - they are disabled. One reason for this, is to let Hyper-NEAT try to discover frequencies that make part of the network be active at different times. Another reasons is that research (as mentioned earlier) points to certain oscillations be in synchrony for different task.

Since the total number of ways the hidden neurons can be active/disabled is 2^n it could in theory mean that Hyper-NEAT could discover 2^n modules - where n is the number of hidden neurons. A setup of the method can be seen in figure (x).

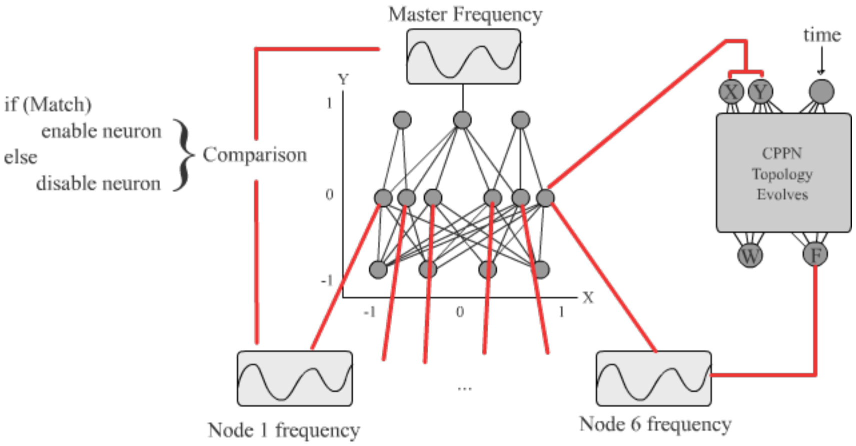
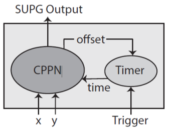
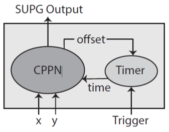


Figure 1:

1.2.2 MaO-HyperNEAT

1.3 Experiment 1 - Recognizing time patterns

Function	Image1	Image2	Error term (r^2)
$f(x) = \begin{cases} 0, & \text{if } x < 0.5 \\ 1, & \text{otherwise} \end{cases}$	 SUPG Output offset CPPN Timer time x y Trigger	 SUPG Output offset CPPN Timer time x y Trigger	0.9
$f(x) = sin(x)$			
$f(x) = 3 \cdot sin(x+2)$			

1.4 Experiment 2 - Timely split task (Simple domain)

1.4.1 Domain

1.4.2 Results

1.5 Experiment 3 - Environment split task

1.5.1 Domain

1.5.2 Results