Hybrid AI & Machine Learning Systems

Logan Kearsley
TJHSST Computer Systems Lab 2006-2007

Abstract:

The purpose of this project is to design a set of libraries and demonstration programs that combine the capabilities of multiple AI and machine learning models to produce a more flexible and versatile hybrid model.

I will primarily be working with back-propagation neural networks and subsumption architectures.

The primary demonstration program will be a highly extensible character recognition system with the capacity to add new characters simply and rapidly.

Background:

Most AI and Machine Learning research to this point has consisted of developing separate individual models to maximize utility for a single problem type or to duplicate biological models.

Comparatively little research has been done in AI systems combining the best aspects of multiple models to produce a more versatile hybrid system. One example, however, is 'hybrid neural networks', which were proposed in the late 1980s to incorporate symbolic processing and neural processing into a single system for speed and ease of control while retaining the flexibility of pure neural nets.

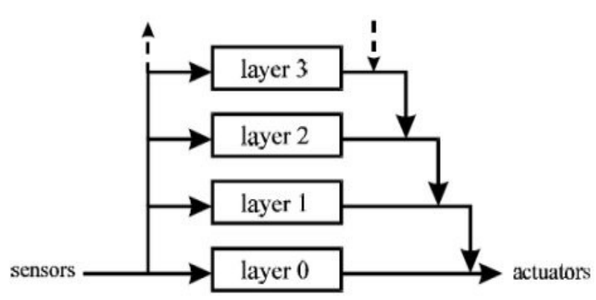
The subsumption architecture model was first used in 1984 and invented by Alexandre Parodi. Subsumption architectures make use of multiple simple rules layered over each other to produce complex emergent behaviors. In this project, I am using the competitive subsumption model.

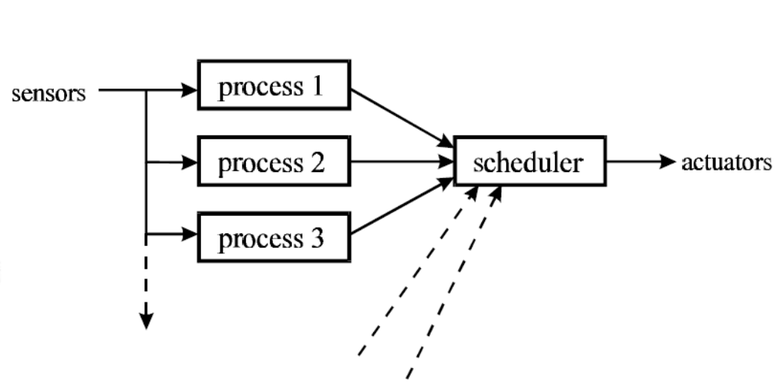
Procedures & Methods

The programming for this project will be done in C, using the gcc compiler. For the neural network programming, I will be using a back-propagation learning algorithm with weight matrices to simulate neuron layers. I have devised a new data type, neuro_t, to store all of the relevant information for a single net. The project is composed of 3 major stages:

- Model Development- creating the code for the neural networks, including a back-propagation learning algorithm.
- Algorithmic Testing- evaluating and improving the effectiveness of the AI algorithms mostly by feeding test programs random input/output associations to ensure that they learn properly.
- Application Testing- writing a character recognition program and testing with data from a custom bitmap font.

Permanent Priority vs. Competitive Subsumption:






```
Load/Teach/Query/Exit? : Q

Input Vector:
1
0

Layer 1          1.000000                  0.000000

0.994611                  2.956922                  3.066278

0.485401                  0.530026                  0.538494
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
Layer 2          0.994611                  0.485401

0.920221                  4.082325                  -5.088287
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```

Testing Interface Screenshot: Weight matrices for computing boolean XOR.

```
Accuracy: 88.524590%
(6.557373 Improvement)
Rerun [y/n]?

"et's test oLt mu ascii recognition abilitiesi ^`:'/?.><u*"-=
```

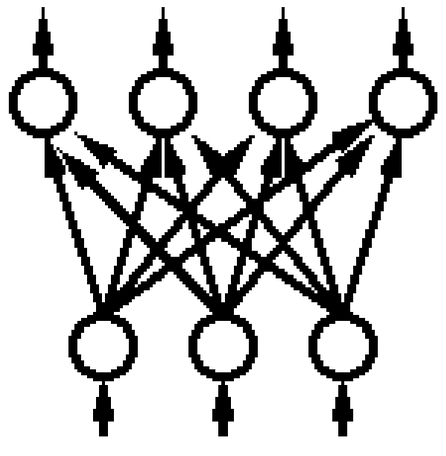
Training Program Screenshot: The program compares its transcription to a user-provided answer file and uses the data to re-train its component networks.

Expected Results

The results of the project will be a set of libraries and architecture descriptions for the manipulation and combination of multiple AI models, and a set of demonstration programs. The programs will be evaluated in terms of speed of learning and accuracy of character identification, and compared to a single-network model.

It is intended that these results will provide a starting point for future researchers to work with the same algorithms, and provide inspiration for investigating more combinations of different AI models.

Neural Connectivity & the Associated Weight Matrix:



In1	In2	In3	
W1,1	W1,2	W1,3	Out1
W2,1	W2,2	W2,3	Out2
W3,1	W3,2	W3,3	Out3
W4,1	W4,2	W4,3	Out4