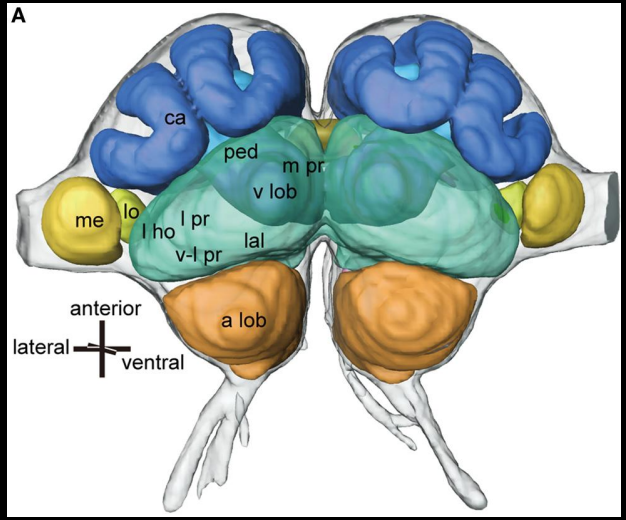
Ant Colony Optimization of Neural Network Structure

or

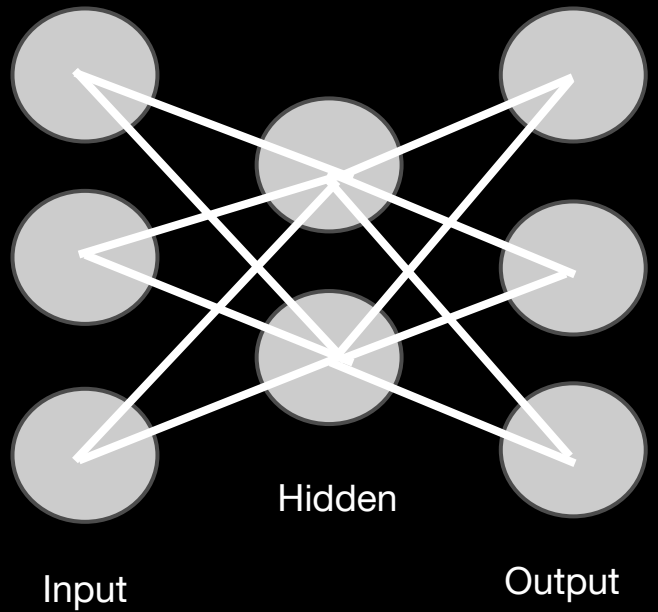
Other Fancy Title



http://www.frontiersin.org/files/Articles/1544/fnbeh-04-00028/image_m/fnbeh-04-00028-g001.jpg

Daniel “Hot Like Shakira” Cohen
Josh “Ginger Bunny” Imhoff
Liam “Baby Boy” Taylor
May 11, 2015
CS 3445
Bowdoin College

Neural Network (w/ hidden layer)



Backpropagation:

- Hidden-to-Output weights updates like perceptron:

$$W_{j,i} \leftarrow W_{j,i} + (\alpha \times a_j \times Err_i \times g'(in_i))$$

- Input-to-Hidden weight updates require error to be propagated from output units back to hidden units:

$$W_{k,j} \leftarrow W_{k,j} + (\alpha \times I_k \times \sum_i W_{j,i} Err_i \; g'(in_i) \; g'(in_j))$$

Majercik, 2015. CS 3445

Do you want ants? Because that's how you get ants.

ACO

Probabilistic Choice --->

- Heuristic = weight changes

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in allowed\ k} \tau_{il}^{\alpha}\eta_{il}^{\beta}} & if\ j \in \text{allowed } k \\ 0 & \text{otherwise} \end{cases}$$

Pheromone Update --->

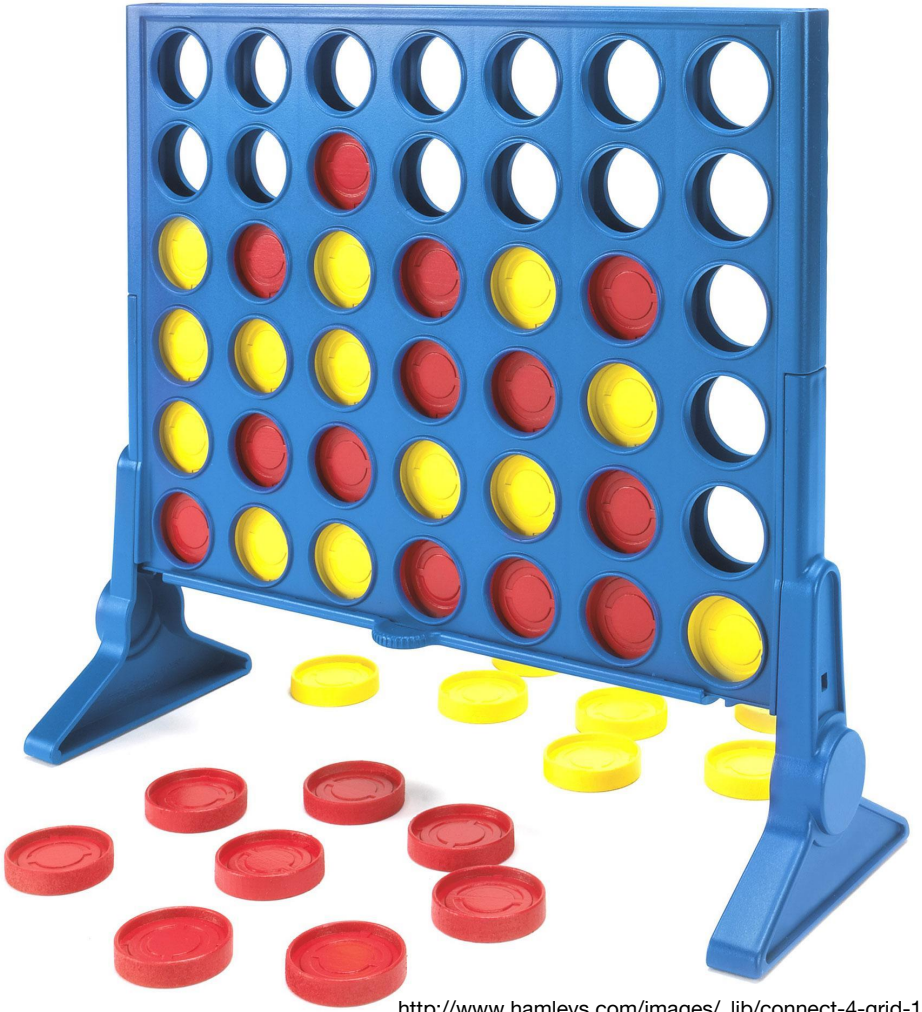
$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bsf}$$

Best Tour Update --->

- 1/error

$$\Delta\tau_{ij}^{bsf} = \begin{cases} \dfrac{1}{L^{bsf}} & if\ (i,j) \in \text{tour ant } bsf \\ 0 & \text{otherwise} \end{cases}$$

Connect 4

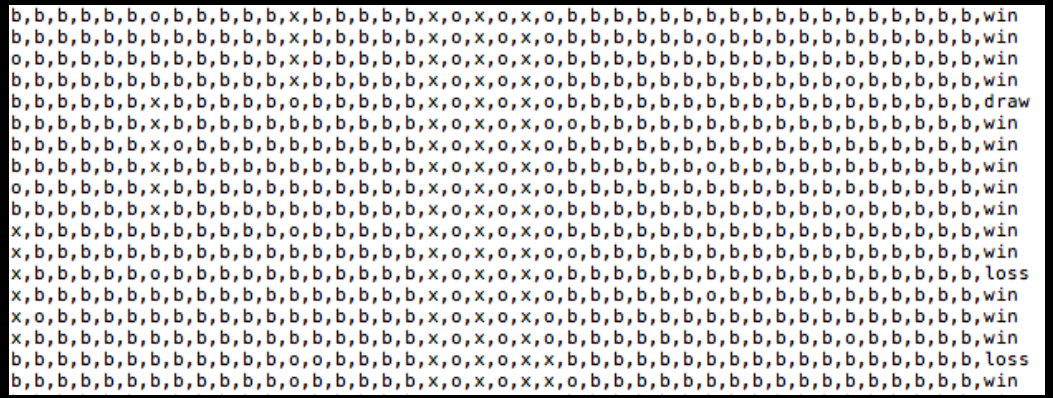


http://www.hamleys.com/images/_lib/connect-4-grid-10222-0-1417083604000.jpg

Connect Four

Input and Output Representation

- Board Positions Map to Input Nodes
- Red = +1, Yellow = -1, Empty = 0
- Three output nodes corresponding to Red win, Red loss, and Draw



Structure Matters

- Positions in Connect Four map to nodes in the neural network

- Different board positions have different influence on game result

http://commons.wikimedia.org/wiki/File:Connect_Four.gif

- Backprop adjusts weights based on the error a node is connected to, but never attempts to simplify the network

Differences from TSP ACO

- No “tour”

  Ants can go from one node to many others, or to none at all based on pheromone + heuristic + randomization

- No guarantee of visiting each node in network

  Probabilities determined on a bell curve, not a linear probability sum

- And again, a different heuristic measurement based on the problem

Ant Network and Heuristic design

- Use ants to assemble neural network with different structure

- Pheromone laid down on edges allows for convergence on good structure

- Heuristic searches the fully connected network structure

  Frequency of Edge Weight Changes in Training / (Epochs * Training Data Size)

Total Algorithm

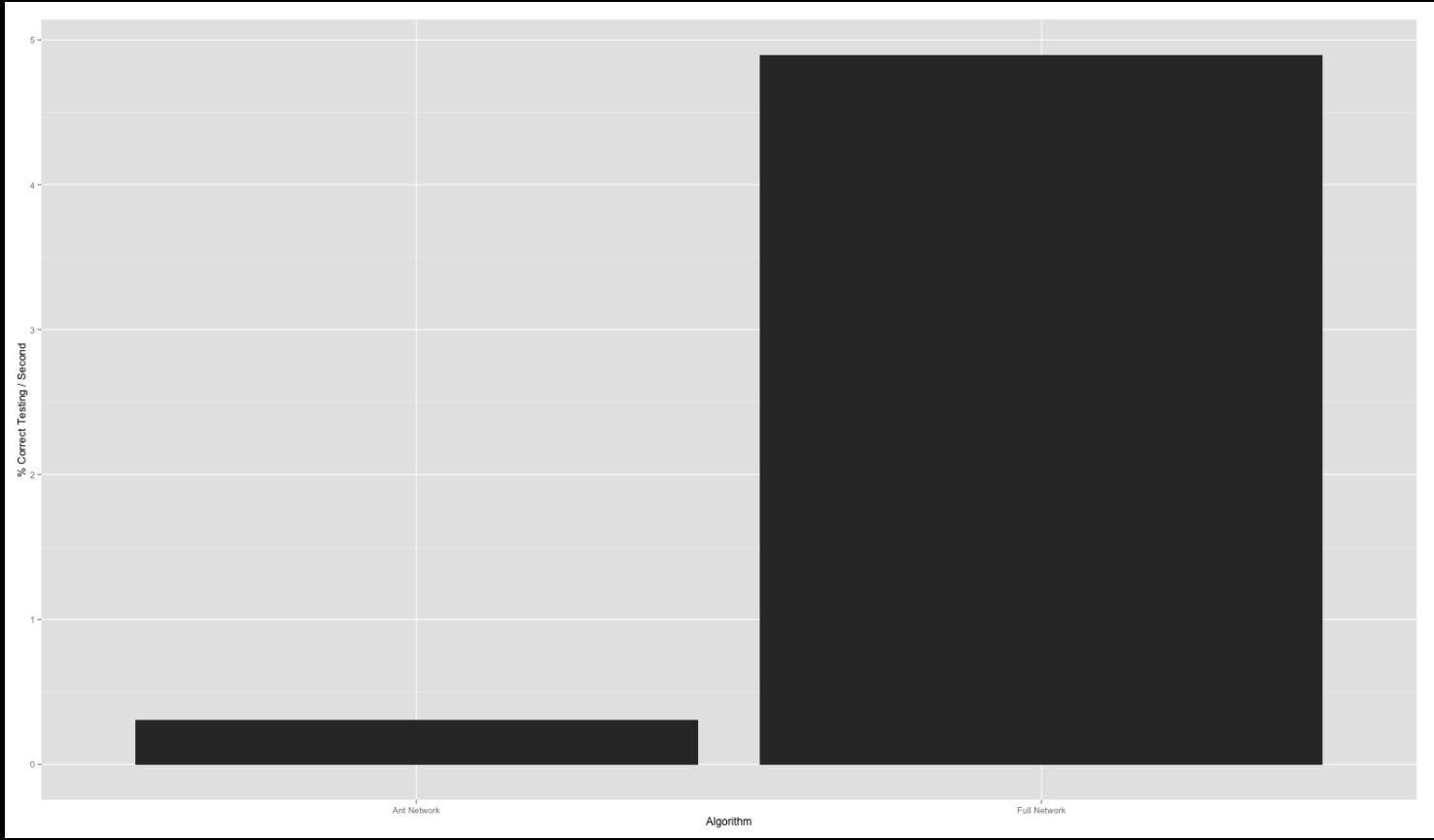
```
For every epoch
    Train Original Network
Test Original Network
For every iteration
    For every ant
        Assemble Structure
        For every epoch
            Train Network
        Test Network
     Update Pheromone
Get Best Ant Structure
For every epoch
    Train Best Network
Test Best Network
```

Results - Does it work? Sort of.

Fully Connected Mean: **71.8%**

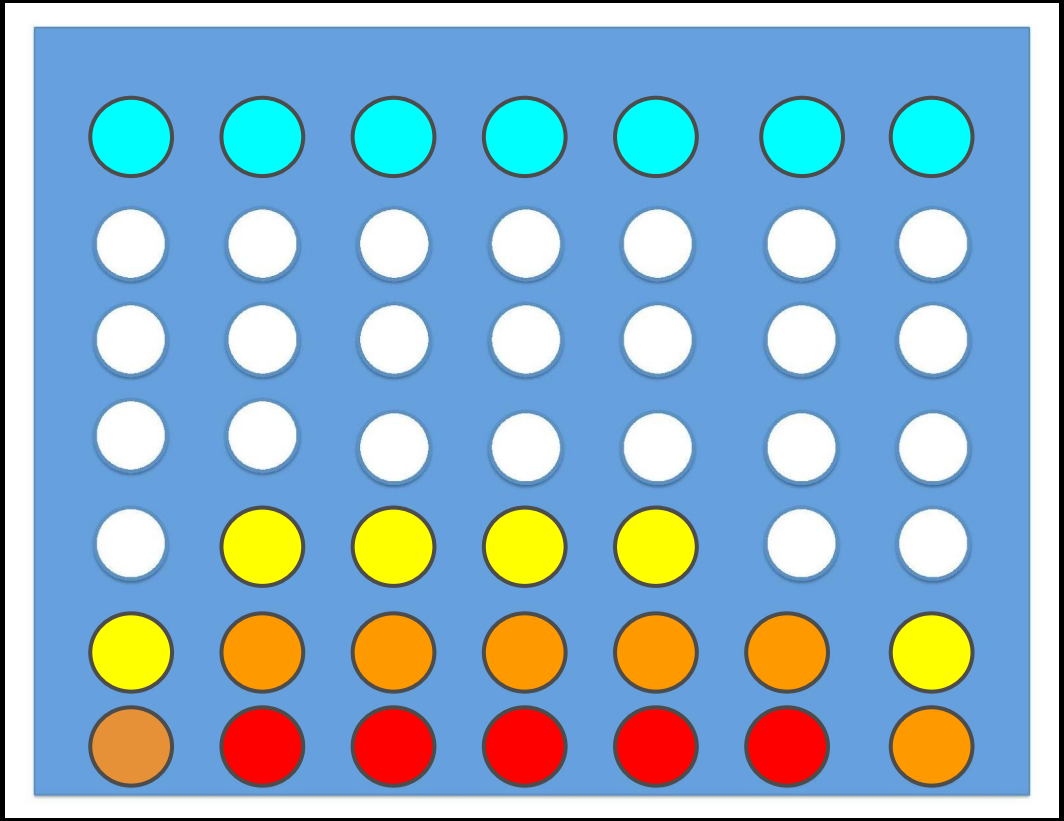
Ant Mean: **70.1%**

Results - But is it faster? *Definitely not.*



Results

Results



Further Work

- Rigorous mathematical investigation into heuristics, probabilities, etc.

- Different kinds of problems may benefit from misleading or useless sections of data being removed through ACO treatment

Further Further Work

- Different kinds of problems can be dealt with using ACO Nets with different heuristics (one that eliminates edges with very strong error, etc.)

- Computational investigation into efficiency gains from a smaller network

Conclusion

- ACO/ NN Hybrid is slower during training, but may be more efficient if you do much more testing than training

- Algorithm gives insight into the importance of structure in NN’s in general