HYBRID METHODS

Using a GA to optimize NN weights:

- The GA replaces backpropagation (there are no backward passes)
- Each GA individual is a complete neural network (the GA encoding is a concatenation of the NN weights)
- Fitness of an individual is sum-squared error on the entire training set
  1. initialize a NN
  2. separate the chromosome into its substrings
  3. copy each substring into a NN weight (and thus, the entire individual sets the values of all NN weights)
  4. for each training case, run a forward pass only and compute the error (target minus actual output)
  5. the fitness of that individual the sum of the squares of the errors for all of the training cases
  6. repeat this process for all of the individuals in the GA population.
- a similar approach can be used for other population-based methods, such as PSO

Using a GA to optimize NN topology (simple approach)

- The GA evolves topology while backpropagation trains the weights
- The number of neurons is fixed and decided in advance
- Each GA individual is a connectivity matrix for the neurons
- Fitness of an individual is residual error after backpropagation, or total error in testing data (generalization)
  1. separate the chromosome into its substrings
  2. initialize the NN topology by connecting neurons according to the substring values
  3. train the neural network using backpropagation
  4. best NN topology evolves over time

<table>
<thead>
<tr>
<th>Connectivity Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Corresponding NN Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2 - 3 - 4 - 5 - 6</td>
</tr>
</tbody>
</table>

| string #1 | 000011000010101 |
| string #2 | 11101010111001 |
| string #3 | 0110011010101 |
| string #4 | 11100010110000 |
| string #5 | 00110100011100 |
| string #6 | 00011100011011 |

<table>
<thead>
<tr>
<th>sum² of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>(error #1)²</td>
</tr>
<tr>
<td>(error #2)²</td>
</tr>
<tr>
<td>(error #3)²</td>
</tr>
<tr>
<td>(error #4)²</td>
</tr>
<tr>
<td>(error #5)²</td>
</tr>
<tr>
<td>(error #6)²</td>
</tr>
<tr>
<td>(error #7)²</td>
</tr>
<tr>
<td>(error #8)²</td>
</tr>
<tr>
<td>(error #9)²</td>
</tr>
<tr>
<td>(error #10)²</td>
</tr>
</tbody>
</table>