

### Part C (10 points)

Your friend is building a device to unlock a door when it hears a secret knock pattern, and realizes that a neural network could do it, if only it had a sense of timing. You suggest feeding the output of one neuron into the input of one of its ancestors, and thereby get a dependence on timing.

1. You think about training the network by standard back-propagation, but decide that you can't. Why?

В чистом виде метод обратного распространения ошибок не может быть применен к такой нейронной сети, т.к. нельзя посчитать частные производные из-за циклических зависимостей.

The solution is clear: Genetic Algorithms! You'll set up a population of identical neural networks with random weights, you discretize your input every 100 milliseconds into a sequence  $k_1 \dots k_n$  of 0 if silence and 1 if a knock was heard, ensuring that  $k_i$  is always 1, and timing out eventually. You'll choose the fittest few neural networks at each step. Your friend jots down a few ideas for fitness functions:

- A. Whether the full knock pattern was correctly classified
- B. The length of the subsequence  $k_1 \dots k_i$  that is correctly classified
- C. The length of the longest subsequence  $k_i \dots k_j$  that is correctly classified
- D. The number of  $k_i$  correctly classified
- E. The number of knock subsequences  $k_i \dots k_j$  that are correctly classified

2. Select all of these fitness functions that one cannot evaluate using the neural net as a black box:

Если выходом нейронной сети является последовательность  $t_1 \dots t_n$  ( $t_i$  равно 1, если  $k_i$  принадлежит паролю, иначе 0), то все функции можно вычислить. Если же выход сети бинарен, то только А является вычисляемой.

3. Select all of the fitness functions that will immediately trap the genetic algorithm in a fitness plateau:

А и Е, так как они не отвечают настоящей оценке.  
В и С тоже не отвечают ей, но рано или поздно они могут привести к решению.

4. Select all of the fitness functions that do not correlate with the actual fitness:

А: если правильно угадано 0 элементов или  $n-1$  элемент, то они считаются равными, что неверно.  
В: если правильно угадана  $n$ -ть  $k_2 \dots k_n$  или только первый элемент, то последний считается лучше, что неверно.  
С: если правильно угадан каждый второй элемент или  $n$ -ть  $k_1 \dots k_{(n-1)/2}$ , то последняя считается лучше, что неверно.  
Е: если правильно угаданы  $n$ -ти  $k_1 \dots k_2, k_{(n-1)} \dots k_n$  или  $n$ -ть  $k_1 \dots k_{(n-1)}$ , то последняя считается хуже, что неверно.

5. Having selected a fitness function, you decide to mutate weights randomly, and choose about half of the weights from each parent for crossover. Your friend uses the GA to train an NN on the example knock sequence, and it consistently says true for that knock sequence. Excited, he installs it, goes outside, waits for it to lock, someone runs by, and the door opens. What was missing from his training data?

Он не учел то, что источник звука должен касаться двери.

6. Having added that, he retrain the system on all the training data, and it's classifying things perfectly, and he goes outside and waits for it to lock, knocks the secret pattern, trying again and again, but you eventually have to let him back in. What happened?

Он установил датчик касания не с той стороны.