Introduction to Machine Learning - Questions Evolutionary Algorithms

Prepared by Antoine Cully

Autumn 2020/2021

1 Sample answers

Here are an example of answers for the questions on evolutionary algorithms. Other solutions might be possible.

1. First, we need to count the number of weights (+bias) we have to evolve. (3+1)*4+(4+1)*1=21 (Note: the rest of the network's structure (e.g., activation) is fixed). So, we need to find the (real) values of 21 parameters. As the problem is continuous, we have to use an ES or evolutionary algorithm (not a GA). I will use (mu+lambda)-ES but any other evolutionary algorithm would be fine as well. First, let's define the genotype: A vector of 21 floats (or double). (We can also add a 22nd float to govern the mutation rate of the mutation operator). The phenotype is a fully functional neural network using the values from the genotype as weight value. The development of a genotype to a phenotype is as follows: The 12 first values are remapped to form a 3x4 matrix, which will represent the weight matrix of the first layer, the 4 next values, will be the biases. Finally, the 4 last values will be mapped to a 4x1 matrix as the weight matrix of the output layer, and the last value will be the bias of the output.

2. Now that the genotype, phenotype and the function to develop a genotype into a phenotype is defined, we can focus on the genetic operators.

   - Mutation operator: as we use an ES, the mutation operator is the addition of a Gaussian noise (either with a fixed sigma, or a sigma obtained from the 22nd gene of the genotype).
   - The cross-over operator is not mandatory in ES
   - For the selection operator, we use the standard procedure of (mu+lambda)-ES: out of our population of mu+lambda individual, we select the mu best individual. Then, we generate lambda offsprings by randomly selecting (uniform selection) one of the mu individuals as parents.
   - The population for the next generation is defined as the mu selected individuals + the lambda generated offsprings

3. The fitness function used for this task can be a performance metrics, for instance classification rate, of the produced neural network on the training dataset. The fitness value is bounded between 0% and 100%, where 100% is the best possible value.

4. The pseudo-code of the ES algorithm is defined in slide 42 of the course. Hyper-parameter of the algorithm (mu and lambda are fixed by using the heuristic that lambda should be 5 times larger than mu):

   - mu: 5
   - Lambda: 25
   - Sigma (to tau0). For instance, we can fix tau0 to 1/sqrt(21).
   - Number of generations: I would start with 1000 generations, but if we detect convergence before we can stop.