Machine Learning

Lecture 11

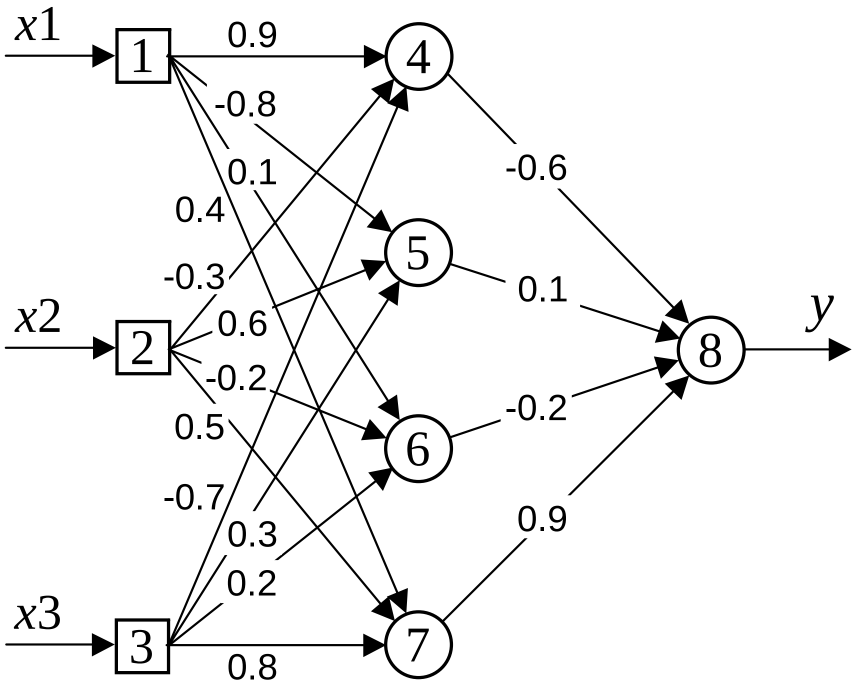
Using of Genetic Algorithms for learning and evolution of neural networks

Using GA for learning of MLP

- For evolution of weights
- For evolution of structure (topology)

Encoding a set of weights in a chromosome

From neuron:	1	2	3	4	5	6	7	8
To neuron: 1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0.9	-0.3	-0.7	0	0	0	0	0
5	-0.8	0.6	0.3	0	0	0	0	0
6	0.1	-0.2	0.2	0	0	0	0	0
7	0.4	0.5	0.8	0	0	0	0	0
8	0	0	0	-0.6	0.1	-0.2	0.9	0



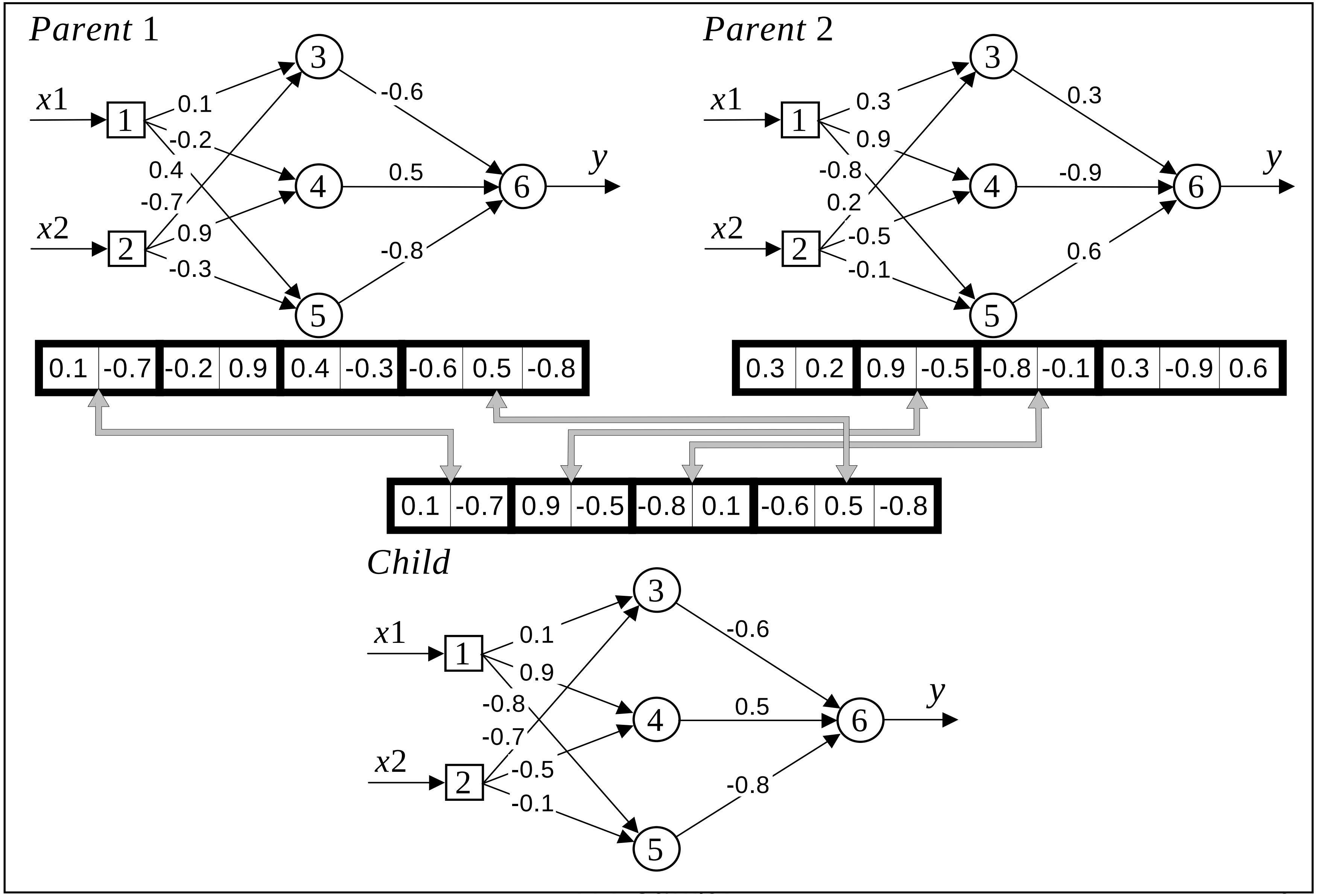
Chromosome:

0.9	-0.3	-0.7	-0.8	0.6	0.3	0.1	-0.2	0.2	0.4	0.5	0.8	-0.6	0.1	-0.2	0.9

- The second step is to define a fitness function for evaluating the chromosome’s performance. This function must estimate the performance of a given neural network. We can apply here a simple function defined by the sum of squared errors.
- The training set of examples is presented to the network, and the sum of squared errors is calculated. The smaller the sum, the fitter the chromosome. **The genetic algorithm attempts to find a set of weights that minimises the sum of squared errors.**

- The third step is to choose the genetic operators – crossover and mutation. A crossover operator takes two parent chromosomes and creates a single child with genetic material from both parents. Each gene in the child's chromosome is represented by the corresponding gene of the randomly selected parent.
- A mutation operator selects a gene in a chromosome and adds a small random value between –1 and 1 to each weight in this gene.

Crossover in weight optimisation

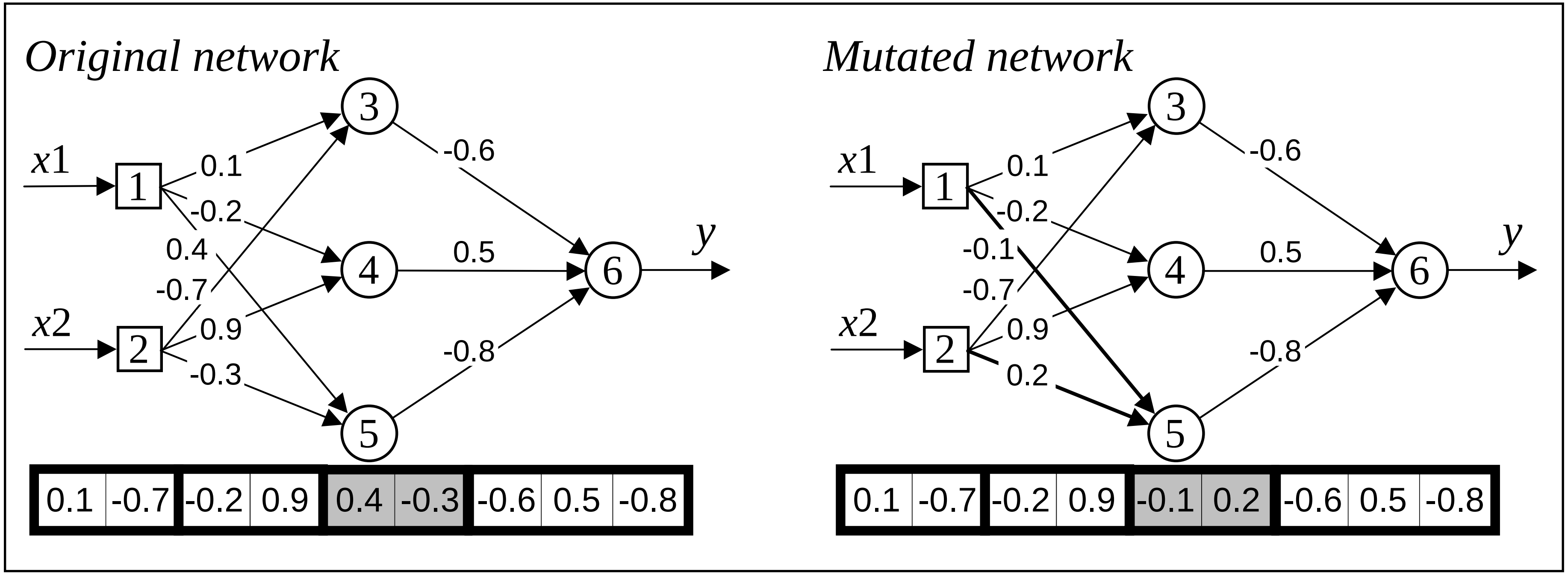


Parent 1: | 0.1 | -0.7 | -0.2 | 0.9 | 0.4 | -0.3 | -0.6 | 0.5 | -0.8 |

Parent 2: | 0.3 | 0.2 | 0.9 | -0.5 | -0.8 | -0.1 | 0.3 | -0.9 | 0.6 |

Child: | 0.1 | -0.7 | 0.9 | -0.5 | -0.8 | 0.1 | -0.6 | 0.5 | -0.8 |

Mutation in weight optimisation



0.1	-0.7	-0.2	0.9	0.4	-0.3	-0.6	0.5	-0.8

0.1	-0.7	-0.2	0.9	-0.1	0.2	-0.6	0.5	-0.8

Can genetic algorithms help us in selecting the network architecture?

The architecture of the network (i.e. the number of neurons and their interconnections) often determines the success or failure of the application. Usually the network architecture is decided by trial and error; there is a great need for a method of automatically designing the architecture for a particular application. Genetic algorithms may well be suited for this task.

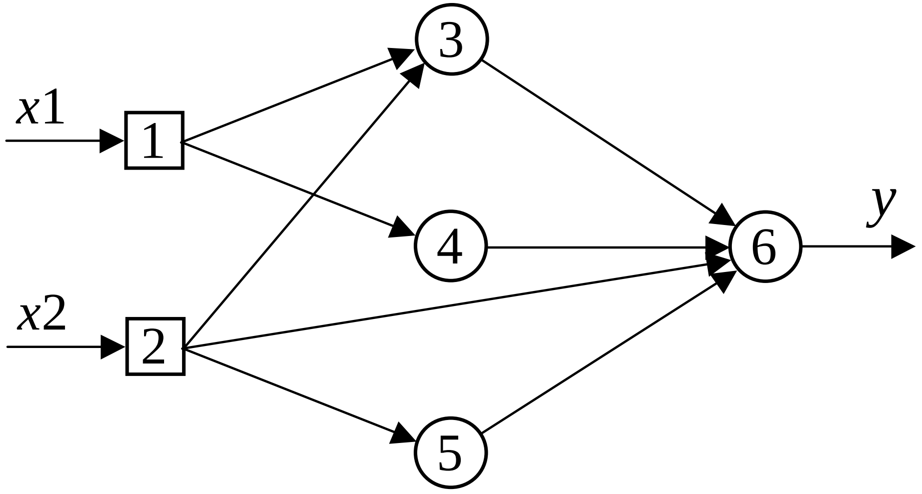
- The basic idea behind evolving a suitable network architecture is to conduct a genetic search in a population of possible architectures.
- We must first choose a method of encoding a network's architecture into a chromosome.

Encoding the network architecture

- The connection topology of a neural network can be represented by a square connectivity matrix.
- Each entry in the matrix defines the type of connection from one neuron (column) to another (row), where 0 means no connection and 1 denotes connection for which the weight can be changed through learning.
- To transform the connectivity matrix into a chromosome, we need only to string the rows of the matrix together.

Encoding of the network topology

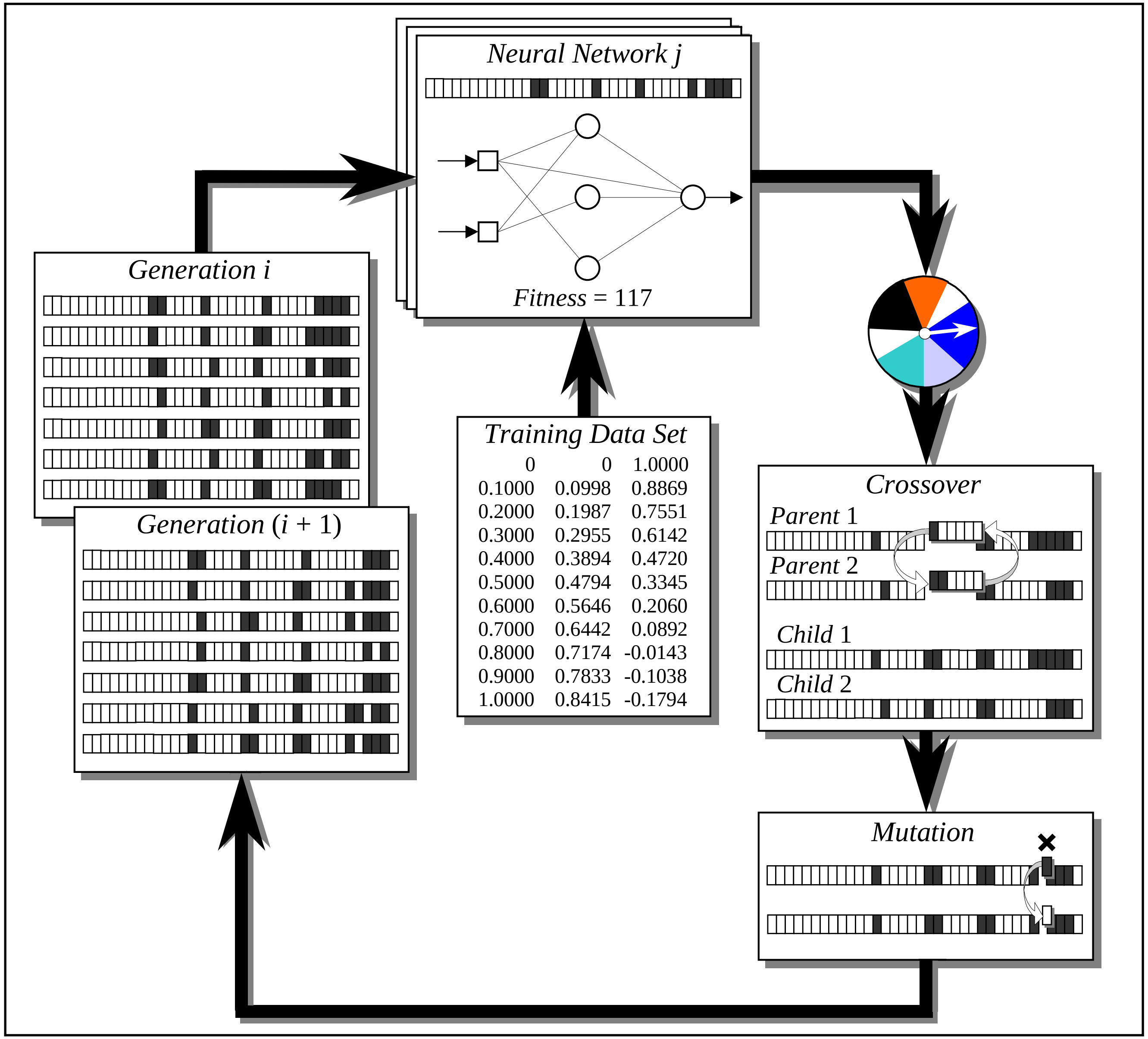
To neuron: \ *From neuron*:	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	1	1	0	0	0	0
4	1	0	0	0	0	0
5	0	1	0	0	0	0
6	0	1	1	1	1	0



Chromosome:

0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0

The cycle of evolving a neural network topology



0	0	1.0000
0.1000	0.0998	0.8869
0.2000	0.1987	0.7551
0.3000	0.2955	0.6142
0.4000	0.3894	0.4720
0.5000	0.4794	0.3345
0.6000	0.5646	0.2060
0.7000	0.6442	0.0892
0.8000	0.7174	-0.0143
0.9000	0.7833	-0.1038
1.0000	0.8415	-0.1794