Lecture 10 - Genetic programming

- The characteristic property of genetic programming is a tree structure that represents solutions; it allows one to represent expressions and programs.
- The solution space consists of all combinations of expressions composed of tree leaves and tree nodes.
- Some solutions may generate invalid code, or code that generates errors. One way of solving this problem is to never use functions that that introduce such disturbances (for example define division by zero to be equal to 1).

Methods of generating random solutions

- Full - if the depth of the existing tree is below some threshold randomly pick nodes from the functions' set and if the depth exceeds some limit pick the tree leave from the set of terminals (terminal nodes). All the trees will have the same depth.
- Grow - if the depth is below some threshold pick either a node or terminal, if the depth exceeds the limit pick only from the set of terminals.
- Ramped half-and-half - generate half of the population from the full method, half from the grow method.

Mutation

- Cut your tree in a random position and generate a new subtree in place of the old one.

Crossover

- Taking two parents and exchanging some subtrees from each parent.

- To protect against trees growing indefinitely penalties for the depth of the tree may be introduced, or a limit of depth.
- Evaluation can be conducted by having test cases, and seeing how many of them this code has passed.
- Genetic programming is very hard, the fitness function is not continuous and ragged, it barely works.

Symbolic regression

- Symbolic regression is an application of GP where we are looking for a function that fits as precisely as possible the given points.
- Traditional regression only looks for coefficients, but in GP it is possible to also look for different classes of functions, hence this regression method is called symbolic.