

진화 AI 기말 프로젝트

20224025

AI 대학원

한명훈

1. 구현 방법

1) Tree Structure

트리의 노드는 세 개의 유형으로 구분하였으며 각각 연산자, 상수, 변수이다. 연산자의 종류는 사칙연산, 삼각함수, 제곱, 루트, ReLU(), ABS(), Log가 있다.

2) Tree Generation

트리는 최대 깊이를 설정하였고 노드 생성 시 확률에 따라 상수, 변수, 연산자 여부를 결정하였다. 유형이 결정되면 변수의 경우 균등한 확률로 연산자를 랜덤하게 선택하였고, 변수도 동일한 방식으로 입력 값을 선택하였다. 상수의 경우 값의 범위를 지정하여 랜덤하게 지정하였다. 자식 노드가 필요한 경우(연산자인 경우) 동일한 방식으로 노드를 생성하여 연결하였다.

3) Evolution Operator

(1) Selection

- * Roulette selection 사용

(2) Crossover

어떤 노드를 바꾸기 위해서는 해당 노드의 부모 노드가 필요하기 때문에 트리에서 자식이 있는 노드를 모두 탐색하였다. 두 개의 크로스 오버 연산 대상이 되는 트리에 대해 이를 수행하고 각각에 대해 바뀔 노드를 랜덤하게 선정한 후 서로 교체하였다.

(3) Mutation

트리를 바꾸는 경우 수식이 크게 바뀔 가능성이 크기 때문에 기존에 잘 찾은 수식에 대해 상수값을 조금씩 바꾸며 최적화하기 어렵다. 따라서 상수 값만을 다시 설정하는 mutation을 따로 만들어 적용해보았다.

type1: tree mutation

- * 전체 수식 트리에서 변환 가능한 노드 중 하나를 선택하여 새로운 노드로 변경한다. 여기서 새로운 노드는 하나의 완전한 수식 트리를 의미한다. 이때 트리가 한 번에 크게 바뀌는 것을 막기 위해 depth가 클수록 더 높은 확률로 선택되도록 하였다.

type2: value mutation

- * 전체 수식 트리에서 상수에 해당하는 노드 중 하나를 선택하여 값을 임의로 변경한다.

2. 코드 구조

- 1) **GA.py**: 기본적인 진화 알고리즘에 필요한 틀(거의 안쓰임)
- 2) **GP.py**: Generic Programming class
- 3) **GPFinal.py**: GP를 Final 프로젝트에 맞게 확장
- 4) **GPMModule.py**: GP의 수식 트리 구현을 위한 클래스 모음
- 5) **circumstance.py**: final 과제 문제 (x, y, z)에 대한 fitness 테스트
-> GPFinal에 새로 작성하여 사용.
- 6) **main.py**: 실행 코드

3. 결과

1) 수식

```
(log(abs((max(0, (sin(log(abs(0.186297))) + sin(max(0, max(0,
(((sin(abs(sin(abs(-2.000000)))) / ((x / x) - (-0.210962 + sin((max(0,
0.000000)**(1/2)))))) / abs(x)) / ((cos(0.869229)**2) / ((abs(-1.324065) + 1.448155) *
log(abs(y))))))))))**2))**2)
```

2) RMSE

106.52

3) 노드 개수

28개