Problems on Basic Data Structures

Koscom Algorithm Lecture

신승우

Wednesday 5th September, 2018

Outline

Introduction to Genetic Algorithm

유전 알고리즘은 어떤 문제를 해결하기 위한 가설들을 자연에서의 유전 과정과 흡사한 방법을 이용하여 진화시키고, fitness function을 이용하여 이를 검증, 발전시켜나가는 방법입니다. 대단히 우연성에 강하게 의존할 것 같은 알고리즘이지만, 의외로 많은 곳에서 활발하게 쓰이고 있습니다.

Process of Genetic Algorithm

크게

- 초기화 : 가설들을 유전자로 표현하고, 랜덤한 유전자를 생성합니다.
- 선택 : 초기화된 유전자들을 **fitness** 함수를 이용하여 평가하고, 이를 선택한다.
- 교차 : 선택된 유전자들을 확률적으로 섞여 변이를 일으켜, 다음 세대의 유전자를 만든다.
- 변이 : 만들어진 다음 세대 유전자에 인위적으로 돌연변이를 일으킨다. 대개의 경우 매우 낮은 확률로 변이를 준다.
- 대치 : 후대 유전자로 현 유전자 풀을 교체한다.
- 반복 : 이 과정을 반복한다.

의 과정을 거친다. 이를 위해서, fitness함수를 적절히 정의하는 것이 필수적이다.

유전 알고리즘의 예시

0000-1111 중 가장 큰 이진수를 찾아보는 문제를 풀어보자. 이 때, 유전자는 정수 4개의 리스트로 생각할 수 있다. 여기서 fitness 함수는 1의 갯수로 두고 진화를 시켜본다고 생각하자.

유전 알고리즘의 예시 : 초기화

10개의 유전자를 아래와 같이 랜덤하게 생성하였다.
[0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 1, 0], [1, 1, 0, 1], [0, 0, 0, 0], [0, 0, 0, 0], [1, 0, 0, 0], [1, 0, 1, 1], [0, 0, 1, 1], [0, 0, 0, 1]

유전 알고리즘의 예시 : 선택

위에서 정한 fitness 함수가 높은 유전자 순으로 배열하면 아래와 같다.
[1, 1, 0, 1], [1, 0, 1, 1] : 3
[1, 0, 1, 0], [0, 0, 1, 1] : 2
[0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0], [0, 0, 0, 1] : 1
[0, 0, 0, 0], [0, 0, 0, 0] : 0
그 후, 이 중 상위 몇 개만을 골라[1] 다음 세대를 만드는 데 사용할 것이다.

[1] 고르는 방법은 토너먼트 선택, 랭킹 기반 선택 등 다양한 방법이 있다.

유전 알고리즘의 예시 : 교차

앞에서 골라진 유전자가 다음의 4개라고 하자.
[1, 1, 0, 1], [1, 0, 1, 1], [1, 0, 1, 0], [0, 0, 1, 1]
이 때, 이를 랜덤하게 교배한다. 여기서는 간단하게 두 유전자가 같은 비트는 그대로 사용하고, 다른 비트는 둘 중 랜덤하게 고르기로 하자. 예를 들어서 아래와 같이 교배할 수 있다.
[1, 1, 0, 1] + [1, 0, 1, 1] = [1*, 1-, 1+, 1*]
이렇게 교배하여 만든 유전자 중 상위 10개를 고르자.

유전 알고리즘의 예시 : 변이/대치

골라진 10개 중 랜덤하게 돌연변이를 줄 수 있다. 돌연변이를 준 후, 다음 세대의 유전자로 기존 유전자를 교체한 후, 이를 반복한다.

Application of Genetic Algorithm

- AutoML : Applying Evolution Algorithm in Deep Learning
- Fault-Localization Problem

Spectra-based Fault-Localization Problem

어떤 프로그램에 대해서 m개의 테스트를 시행하였다. 이 때,

- Spectra : 프로그램의 각 라인에 대해서, (e_p, e_f, n_p, n_f)의 튜플을 말한다. 이 때, e_p, e_f는 각각 그 라인이 통과한/실패한 테스트에서 실행된 횟수를 말하고, n_p, n_f는 각각 그 라인이 통과한/실패한 테스트에서 실행되지 않은 횟수를 말한다.
- Fault-Localization : 테스트를 통과하지 못한 것이 어떤 라인에 의한 것인지 찾는 것을 의미한다.

이 때, Spectra를 이용해서 Fault Localization 문제를 푸는 것을 Spectra-based Fault Localization이라고 한다.

Risk Evaluation Metric

SBFL은 일반적으로 spectra에 기반한 metric들을 이용하여 각 라인이 얼마나 에러에 기여한지 평가한다. 예컨대, e_f가 실패한 테스트의 수와 거의 일치한다면, 그 라인이 에러에 기여했을 확률은 지극히 높을 것이다. 이런 metric을 Risk Evaluation Metric이라고 하며, 좋은 metric을 찾는 것이 SBFL에서 매우 중요하다.

Applying Genetic Algorithm for Metric Generation

이제, metric을 tree로 보고, 랜덤하게 유전자를 준비하여 가지치기와 붙이기를 통해서 유전자 교배를 시행, 이를 통해서 metric을 진화시킬 수 있다. 논문 Evolving Human Competitive Spectra-Based Fault Localisation Techniques 에서는 이와 같은 방법을 통해서 인간이 만든 Risk Evaluation Metric과 비슷하거나 더 좋은 성능을 가지는 metric을 만드는 것에 성공하였다.

Simulation and Object Oriented Language

초창기의 객체지향 언어는 시뮬레이션을 위해서 만들어진 경우가 많았습니다. 이는 객체지향이 시뮬레이션을 작성하기에 매우 적합하기 때문입니다. 객체지향으로 시뮬레이션을 작성하기 적합한 이유는, 객체들 간 상호작용 법칙과 초기조건만 지정해 주면 어렵지 않게 객체 간의 상호작용을 일으킬 수 있기 때문입니다.
가장 대표적인 예시로는 역학 시뮬레이터가 있습니다. 여기서는 2d 공간 내에서 입자간 상호작용을 설정해 주었을 때, 어떤 식으로 시뮬레이션이 작성되는지 살펴보겠습니다.

Basic Mechanics

물리적인 시뮬레이션을 만든다면, 객체간의 상호작용 법칙은 뉴턴역학을 따를 것입니다. 따라서 간단하게 뉴턴역학을 살펴보고자 합니다.
역학에서 풀고자 하는 문제는, 주어진 역장 $\vec{F}$에 대해서 $\vec{r}(t)$를 구하는 것을 목표로 합니다. 이 때, 이를 구하는 방법은 뉴턴의 제 2법칙인 $F = ma$를 이용하여 구할 수 있습니다.

Ideal Gas Simulation

$PV = nRT$의 시뮬레이션을 통한 검증을 해 보겠습니다.

Motivating Example

핸드폰 기지국에서 고객에게 신호를 보내는 상황을 생각해 봅시다. 이 때,

- 모든 고객이 받는 신호는 같습니다.
- 그런데, 모든 고객은 거기서 각자 다른 신호를 받아야만 합니다.

어떻게 하면 여러 고객에게 각자에 맞는 신호를 전달할 수 있을까요? 이를 해결하기 위해서 여러 방법이 제시되었는데, 그 중 하나가 Fourier Transform을 이용한 것입니다.

Motivating Example : Fourier Transform 들여다보기

먼저, 삼각함수의 다음 성질들을 살펴보자.

- $sin(x+y) = sin(x)cos(y) + sin(y)cos(x)$
- $cos(x+y) = cos(x)cos(y) - sin(x)sin(y)$
- $tan(x+y) = \frac{tan(x)tan(y)}{1-tan(x)tan(y)}$

위 식을 이용해서, 1) $sin(nx)sin(mx) = \frac{1}{2}(cos(m-n)x - cos(m+n)x)$ 임을 얻을 수 있다.

Proof of 1)

위 삼각함수 합차공식 중 $cos(x+y) = cos(x)cos(y) - sin(x)sin(y)$ 에서, x 대신 mx, y 대신 nx를 대입하면 다음과 같다.
$cos((m+n)x) = cos(mx)cos(nx) - sin(nx)sin(mx)$
비슷하게, mx, -nx를 각각 대입하면 다음과 같다.
$cos((m-n)x) = cos(mx)cos(-nx) - sin(mx)sin(-nx)$
여기서 $cos(-x) = cos(x), sin(-x) = -sin(x)$임을 이용하고, 위 두 식을 빼면 위 결과를 얻을 수 있다.

확장된 Motivation의 예시 : Fourier Transform 들여다보기

여기서, 위 1)을 이용하여 2) $\int_{-\pi}^{\pi} sin(nx)sin(mx)dx = C\delta_{nm}$ 임을 보일 수 있다. 여기서 δ_{nm}은 n=m이면 1이고, 아니면 0인 함수이다.

Proof of 2)

$\int cos(nx)dx = \frac{1}{n}sin(nx) + D$ 이다. 따라서
$\int \frac{1}{2}(cos(m-n)x - cos(m+n)x)dx =$
$\frac{1}{2(m-n)}sin((m-n)x) - \frac{1}{2(m+n)}sin((m+n)x)$
이다. 여기서 각각 $\pi, -\pi$를 넣어 계산하면 모든 항이 0이 되므로 0이다.
만약 m=n이라면,
$\int \frac{1}{2}(cos(0) - cos(2nx)dx = \frac{x}{2}$ 이므로 π가 된다.

확장된 Motivation의 예시 : 유사벡터

이제, 다음과 같은 집합 2개와 급수를 생각해 보자.

- $[-\pi, \pi]$를 정의역으로 가지는 모든 연속함수의 집합 F
- $S = \{sin(nx) | n = 1, 2, ...\} \cup 1$
- $\sum_{n=0}^{\inf}[\int_{-\pi}^{\pi} f(x)sin(nx)dx]sin(nx)$

여기서, 위 두 슬라이드에서의 결과($\int_{-\pi}^{\pi} sin(nx)sin(mx)dx = C\delta_{nm}$)를 이용하면 S의 원소들을 이용해서 F에서 일종의 좌표계를 얻을 수 있음을 알 수 있다. 이 좌표계는 $f(x) \rightarrow (\int_{-\pi}^{\pi} f(x)sin(nx)dx), n = 0, 1, ...$ 으로 주어진다.

확장된 Motivation의 예시 : 기지국에서 신호 보내기

따라서, 다음과 같이 기지국의 신호를 생각할 수 있다.

- 기지국에서 신호를 n명에게 보낸다고 할 때,
  - n명 각각은 0,1로 된 stream을 받을 수 있어야 한다.
  - 따라서, 2n개의 sin 기저를 준비해서 각각 유저에게 배포하고,
  - 유저는 본인이 가진 벡터를 이용하여 신호를 '내적' 하여, 남은 결과값만을 분석한다.

Implementation

Coding!