Лекция 11

Библиотека DEAP

DEAP

- Библиотека для Python с открытым исходным кодом, предоставляющая базовый функционал для проведения оптимизации генетическими алгоритмами.
- Наличие открытого кода позволяет изучить “внутренности” вычислений, а также на их основе создавать модификации алгоритмов.
- Алгоритм позволяет работать с внешними функциями не заложенными изначально в библиотеку.
- Литература - “Генетические алгоритмы на Python” - Эйял Вирсански
- https://deap.readthedocs.io/en/master/#
- https://github.com/deap/deap

Creator

- Подмодуль библиотеки creator отвечает за мета-фабрику по созданию новых классов через наследование.
- метод create - создает новый класс с заданным именем путем наследования и добавлением атрибутов класса/объекта
- Сигнатура: deap.creator.create(name, base[, attribute[, ...]])
- Не изменяемые атрибуты становятся атрибутами класса, а изменяемые атрибуты становятся атрибутами объекта.

```
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
```

Toolbox

- deap.base.Toolbox - содержит необходимые операции для проведения вычисления. Изначально не все операции заданы, и нужно их зарегистрировать.
- метод register - регистрирует новую функцию для последующего использования.
- Сигнатура: register(alias, method[, argument[, ...]])
- Для генетического алгоритма должны быть зарегистрированы функции: "evaluate" - оценить индивидуума, "select" - отбор, "mate" - скрещивание, "mutate" - мутация.

```
toolbox = base.Toolbox()
toolbox.register("zeroOrOne", random.randint, 0, 1)
```

Вспомогательные классы

- deap.base.Fitness - класс приспособленности индивидуума. Обязательно должен быть атрибутом объекта индивидуума. Хранит: values - значения (может быть несколько), weights - веса, wvalues - взвешенные значения
- deap.tools.Statistics - класс для сбора статистики. В конструкторе принимает ключ. Через метод register можно задать то какую статистику собирать.

```
stats = tools.Statistics(lambda ind: ind.fitness.values)
stats.register("max", numpy.max)
stats.register("avg", numpy.mean)
```

- deap.tools.HallOfFame - “зал славы”, содержит N лучших решений

```
hof = tools.HallOfFame(HALL_OF_FAME_SIZE)
print("Best Ever Individual = ", hof.items[0])
```

Операторы ГА - Скрещивание

- Скрещивание должно принимать 2-х особей, и изменять их. Также и возвращает новых особей. Может принимать больше 2-х аргументов для настройки параметров.
  - cxOnePoint - одноточечное скрещивание
  - cxTwoPoint - двухточечное скрещивание
  - cxUniform - равномерное скрещивание
  - cxOrdered - упорядоченное скрещивание
  - cxBlend - скрещивание смешением
  - cxSimulatedBinary - имитация двоичного скрещивания

Операторы ГА - Мутация

- Мутация первым аргументом должно принимать особь для мутации, и изменять ее. Может принимать доп. параметры для настройки алгоритма.
  - mutGaussian - вещественная мутация по нормальному закону
  - mutShuffleIndexes - мутация перетасовкой
  - mutFlipBit - инвертирование бита
  - mutInversion - мутация обращением

Операторы ГА - Отбор

- Отбор первым аргументом должен принимать популяцию, а затем возвращать популяцию для дальнейшей обработки. Может принимать доп. параметры для настройки алгоритма.
  - selTournament - турнирный отбор
  - selRoulette - отбор по правилу рулетки
  - selRandom - отбор случайным образом
  - selBest - отбор лучших
  - selWorst - отбор худших

Простой ГА

- Простой ГА представлен функцией eaSimple. Возвращает конечную популяцию и объект LogBook, содержащий номер, кол-во вычислений и статистику каждой популяции
- Принимает в качестве параметров:
  - population - популяция
  - toolbox - набор зарегистрированных функций
  - cxpb - вероятность скрещивания
  - mutpb - вероятность мутации
  - ngen - количество поколений
  - stats - объект статистики [опциональный]
  - halloffame - “зал. славы” [опциональный]