

Генетическое программирование

Сергей Николенко

Машинное обучение — ИТМО, осень 2006

Outline

- 1 Генетическое программирование
 - Основная идея
 - Схема
- 2 Генетическое программирование в деталях
 - Начальные данные и подробная схема
 - Генетические операции
 - Пример
 - Обсуждение
- 3 Крестики-нолики
 - Постановка задачи и структура сети
 - Алгоритм
 - Что получилось и что дальше?
- 4 Netflix Prize

История

- Ещё Тьюринг писал о генерации программ посредством мутаций и естественного отбора.
- Затем об этом вспомнили в середине 1980-х, была разработана основная парадигма.

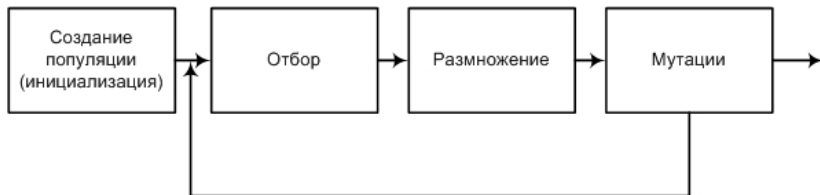
Успехи генетического программирования

- Был получен ряд результатов, которые вполне могут соперничать с человеческими.
- Например, в 2002 году были разработаны три автоматических контроллера, которые были лучше, чем все ранее известные. На эти контроллеры подана патентная заявка, которую, скорее всего, удовлетворят.
- Фактически, генетическое программирование уже может совершать настоящие открытия.
- См. <http://www.genetic-programming.com>

Генетические алгоритмы

Вспомним схему генетического алгоритма:

- Сгенерировать начальную популяцию.
- Пока не достигнуто значение, большее $Fitness_{max}$:
 - Выбрать часть существующей популяции (отдавая предпочтение более приспособленным особям).
 - Применить к этой части генетические операции, породив потомков.
 - Подсчитать $Fitness$ для особей новой популяции.



Генетическое программирование

- Генетическое программирование — применение генетических алгоритмов, популяцией которых являются программы (алгоритмы).
- Т.е. фактически мы уже знаем, что делать. Давайте конкретизируем.

Outline

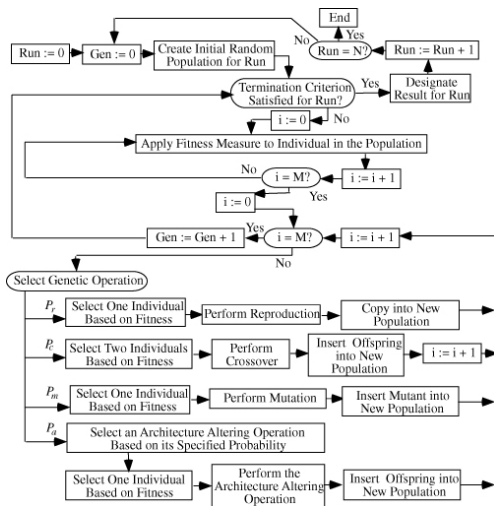
- 1 Генетическое программирование
 - Основная идея
 - Схема
- 2 Генетическое программирование в деталях
 - Начальные данные и подробная схема
 - Генетические операции
 - Пример
 - Обсуждение
- 3 Крестики-нолики
 - Постановка задачи и структура сети
 - Алгоритм
 - Что получилось и что дальше?
- 4 Netflix Prize

Начальные данные

Мы будем строить программу как дерево. Человек должен указать:

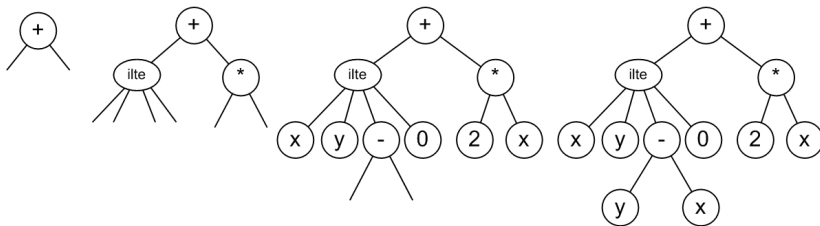
- Набор *терминалов* — независимых переменных, нульарных функций, констант — которые будут стоять в листьях дерева.
- Набор примитивных функций, которые будут в других узлах дерева.
- Мету приспособленности (fitness measure).
- Параметры запуска программы (увидим ниже).
- Критерий остановки и цель всего процесса.

Более подробная схема



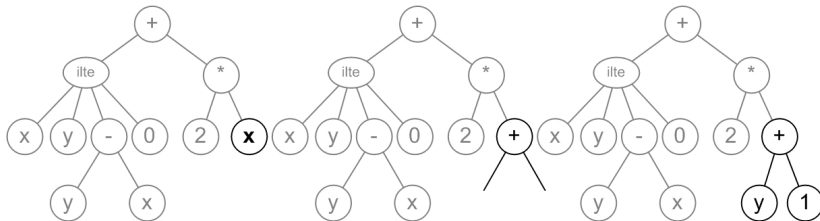
Случайное порождение

- Случайно выбираем узлы из списка, включая терминалы, а затем генерируем следующие уровни на основе аргументности выбранных функций.
- Можно задать максимальную глубину.



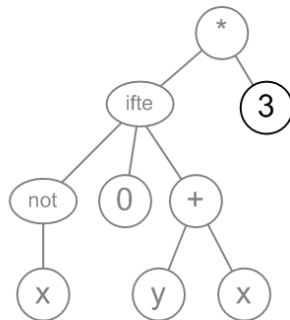
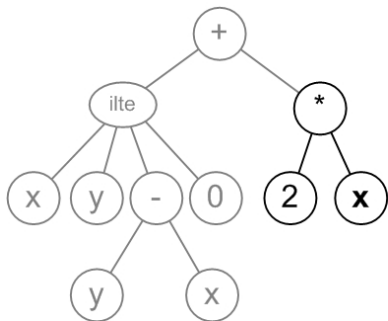
Мутация

- Выбрать случайный узел и вырастить из него новую случайную программу.
- Узел не обязательно должен быть листом



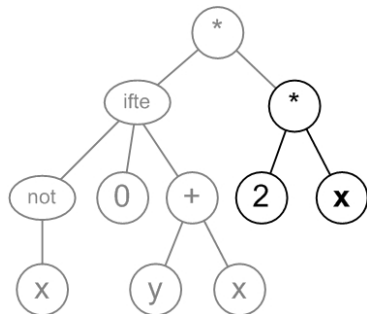
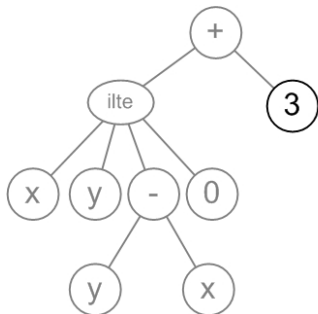
Кроссовер

- Выбрать две случайные программы.
- Выбрать в них два случайных узла и поменять местами поддеревья, растущие из этих узлов
- Вариант: одну из программ не трогать, только скопировать её поддерево во вторую.



Кроссовер

- Выбрать две случайные программы.
- Выбрать в них два случайных узла и поменять местами поддеревья, растущие из этих узлов
- Вариант: одну из программ не трогать, только скопировать её поддерево во вторую.



Постановка задачи

- Будем реализовывать программу, подсчитывающую функцию $x^2 + x + 1$.
- Множество терминалов — одна переменная x и константы (например, от -5 до 5).
- Множество элементарных функций — $\{+, -, *, \%\}$.

Постановка задачи

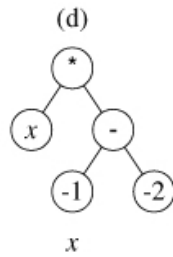
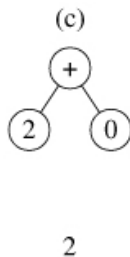
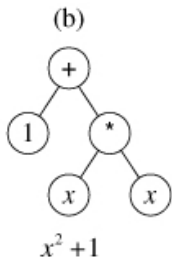
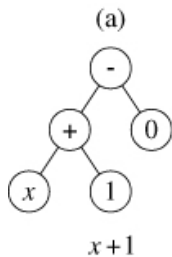
- Какую выбрать функцию ошибки?

Постановка задачи

- Какую выбрать функцию ошибки?
- Функция ошибки — интеграл от разности между функцией, которую реализует программа из популяции, и целевой функцией $x^2 + x + 1$.

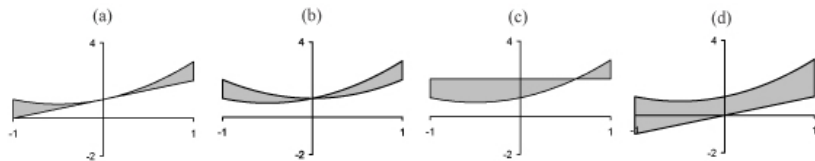
Первый шаг

Сгенерировали случайную популяцию:

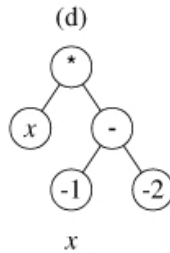
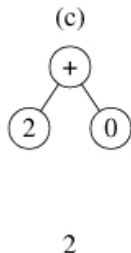
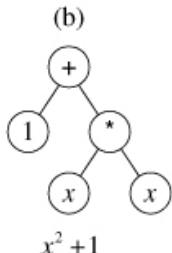
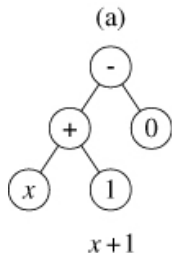


Первый шаг

Её приспособленность далека от идеала:



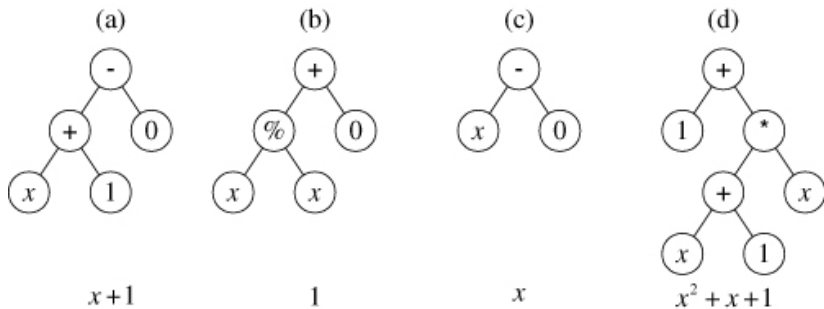
Второй шаг



- Выбрали (a) для воспроизводства.
- Выбрали (c) для мутации, поменяли 2 на дерево, подсчитывающее $(x\%x)$.
- Выбрали (a) и (b) для кроссовера, поменяли местами дерево с корнем в + и x .

После второго шага

Получилось:



Осталось подсчитать функцию приспособленности новой популяции и убедиться, что в ней есть идеальная особь.

Выработка задачи для решения

Какие задачи можно решать генетическим программированием? Давайте выработаем «техническое задание» одной из таких задач.

Упражнение

Реализовать выбранную задачу генетического программирования.

Outline

- 1 Генетическое программирование
 - Основная идея
 - Схема
- 2 Генетическое программирование в деталях
 - Начальные данные и подробная схема
 - Генетические операции
 - Пример
 - Обсуждение
- 3 Крестики-нолики
 - Постановка задачи и структура сети
 - Алгоритм
 - Что получилось и что дальше?
- 4 Netflix Prize

Задача

- Задача: реализовать программу, которая бы играла в «крестики-нолики», самообучаясь по дороге.
- Как вы думаете, какой конкретный аппарат лучше применить?

Задача

- Задача: реализовать программу, которая бы играла в «крестики-нолики», самообучаясь по дороге.
- Как вы думаете, какой конкретный аппарат лучше применить?
- Это, скорее всего, не лучший способ, но я реализовал «крестики-нолики» нейронной сетью.

Структура сети

- Предположим, что играем на поле $n \times n$.
- Тогда у сети n^2 входов, реализующих позицию на доске (1 — стоит крестик, -1 — стоит нолик, 0 — пусто).
- Сеть будет глубины 2 — есть скрытый уровень и один перцептрон на втором уровне, результат которого подаётся на выход.

Структура сети

- Перцептроны — нелинейные (сигмоид).
- Результатом работы нейронной сети должна быть оценка позиции.
- Какой должен быть алгоритм?

Алгоритм

- Дана позиция.
- Для всех возможных своих ходов перебираем все возможные ходы противника.
- Выбираем ход, дающий максимум по своим ходам из минимумов по ходам противника:

$$Result = \arg \max_{OwnMoves} \min_{AdvMoves} Net(Board).$$

Алгоритм обучения

- Стало понятно, как применять полученную нейронную сеть; то есть как играть, если сеть уже обучена.
- Как обучать сеть? Мы не знаем целевой функции!

Алгоритм обучения

- Оказывается, что хорошо работает следующая эвристика: пусть дана запись партии, закончившейся с известным результатом (т.е. целевая функция у последней позиции есть).
- Тогда целевая функция позиции — это либо 1, 0 или 0.5, если уже стоит выигрыш, поражение или ничья, либо значение текущей гипотезы на *следующей* позиции в партии.
- Кажется странным: используем гипотезу для её же тренировки; но работает.

Что получилось

- Для «классических» крестиков-ноликов на доске 3×3 программа довольно быстро обучилась идеальной стратегии.
- Дальше начались проблемы со скоростью работы, требуются оптимизации, но на доске 10×10 в принципе ещё работает; правда, я пока не успел хорошо её обучить.

Возможные улучшения

Полно всего, что недоделано или сделано плохо:

- Алгоритм
- Идентификация одинаковых позиций
- Бесконечная доска
- Игра за любого игрока
- Оптимизации
- Интерфейс
- ...

Outline

- 1 Генетическое программирование
 - Основная идея
 - Схема
- 2 Генетическое программирование в деталях
 - Начальные данные и подробная схема
 - Генетические операции
 - Пример
 - Обсуждение
- 3 Крестики-нолики
 - Постановка задачи и структура сети
 - Алгоритм
 - Что получилось и что дальше?
- 4 Netflix Prize

Что такое Netflix

- Компания, которая продаёт фильмы по схеме:
 - платите за подписку;
 - заказываете до трёх DVD;
 - посмотрели — присылаете обратно, вам высылают новые DVD.

Суть задачи

- Netflix хочет научиться предсказывать вкусы своих пользователей, чтобы рекомендовать фильмы, которые им понравятся.
- Есть огромная база данных (100 миллионов заказов).
- Нужно написать программу, которая обучится по этой базе и сможет лучше других предсказать тестовый набор данных.

Призы

- Главный приз — \$1000000 — если удастся улучшить их алгоритм на 10%.
- Progress Prize — \$50000 ежегодно — если удастся улучшить ненамного и оказаться лучше других.
- Мера улучшения — среднеквадратичная ошибка. У них сейчас — около 0.95 звёздочек (по пятибалльной шкале).

Спасибо за внимание!

- Lecture notes, слайды и коды программ появятся на моей homepage:
`http://logic.pdmi.ras.ru/~sergey/index.php?page=teaching`
- Присылайте любые замечания, коды программ на других языках, решения упражнений, новые численные примеры и прочее по адресам:
`sergey@logic.pdmi.ras.ru, smartnik@inbox.ru`