



Белявцев И.П., Старков С.О.

ИАТЭ НИЯУ МИФИ

djbelyak@gmail.com

Введение

Конечные автоматы активно применяются в различных областях науки и техники, особенно в области информатики. Одной из областей применения автоматов является моделирование[1]. Описание процессов в виде конечных автоматов удобно для человека. В случае сложных процессов допустимо описание процесса в виде композиции автоматов[2]. Композиция автоматов все также понятна для человека и может быть заменена одним большим автоматом.

1. Постановка задачи

Одной из задач, которую можно решать над автоматными моделями – это поиск кратчайших последовательностей входных символов (событий) из начального состояния в одно из конечных состояний. Так как одним из возможных представлений функции переходов является диаграмма состояний или граф переходов, то задача поиска оптимальной последовательности событий сводится к поиску кратчайшего расстояния от вершины начального состояния до вершины одного из конечных состояний.

Есть множество эффективных алгоритмов поиска пути в графе, однако, сложность алгоритмов зависит от количества вершин и ребер графа. Также задача усложняется необходимостью проверки каждой конечной вершины. В связи с этим ставится задача построения функции

$$f : Q \rightarrow X,$$

выбирающей следующий входной символ таким образом, что-

бы за минимальное количество символов автомат достиг одного из конечных состояний.

Для построения данной функции создадим искусственную нейронную сеть прямого распространения, которая принимает состояние автомата (представленное в One-Hot кодировке), а на выходе выдает вектор размерности $|X|$, представляющий меру близости до конечного состояния.

2. Алгоритм обучения

Инициализация. Конечный автомат A переводится в состояние q_0 . Искусственная нейронная сеть N инициализируется случайными малыми весами. Список D пуст. Стек S пуст.

Шаг 1. Искусственная нейронная сеть N по текущему состоянию q_i вычисляет меру близости. По мере близости выбирается следующий входной символ x_i . При $(q_i, x_i) \in D$, x_i заменяется на случайно выбранный другой элемент множества X . Кортеж (q_i, x_i) добавляется в стек S . Автомат A получает на вход символ x_i и переходит в следующее состояние q_{i+1} .

Шаг 2. Если $q_{i+1} \in F$ переходим к шагу 3, в противном случае – переходим к шагу 1.

Шаг 3. Из стека S по очереди извлекаются кортежи (q_i, x_i) , дополняются порядковым номером в стеке записываются в список D . Если (q_i, x_i, n_i) уже был записан в списке D , то номер обновляется минимальным значением.

Шаг 4. Нейронная сеть N обучается методом обратного распространения ошибки. Для каждого q_i формируется вектор размерно-

сти $|X|$. Если для (q_i, x_j, n_j) , есть значения в D , то j -ый компонент вектора будет равен $1 - \frac{n_j}{\sum_{(q_i, x_k) \in D} n_k}$. В противном случае – j -ый компонент вектора равен 0.

Шаг 5. При необходимости продолжить обучение. Конечный автомат A переводится в состояние q_0 . Переходим к шагу 1.

3. Результаты

Данный алгоритм был реализован с помощью фреймворка машинного обучения TensorFlow. Результаты тестирования на различных случайно сгенерированных графах (количество вершин от 10 до 10000) подтверждают работоспособность данного подхода.

Список литературы

- [1] Поликарпова Н. И., Шалыто А. А. Автоматное программирование — СПб., Питер, 2009. — 176 с.
- [2] Гуренко В. В. Введение в теорию автоматов: Электронное учебное издание/ В. В. Гуренко. — Москва: МГТУ им. Н.Э. Баумана, 2013. — 62 с.

Дополнительные материалы

Текст данного доклада и дополнительные материалы доступны по ссылке

<https://djbelyak.ru/talk/mephi-2018-talk/>

