Изучение динамики клеточных автоматов с помощью нейронных сетей

Мазяр В. А., СПбГУ, Санкт-Петербург st087826@student.spbu.ru,
Мокаев Т.Н., СПбГУ, Санкт-Петербург t.mokaev@spbu.ru

Аннотация

Исследование динамики клеточных автоматов с помощью нейронных сетей представляет собой актуальную область исследований. Данная работа посвящена изучению возможностей применения нейронных сетей для прогнозирования динамики клеточных автоматов. В работе, на примере известного клеточного автомата – игра «Жизнь», будут рассмотрены и сравнены между собой некоторые методы обучения с учителем, а также предложен и проанализирован способ их ускорения.

Введение

Клеточные автоматы [1] – дискретная модель, основанная на пространстве, состоящем из смежных клеток. Каждая клетка может находиться в одном из конечного числа состояний $\sigma \in \Sigma \equiv \{0, 1, 2, ...k - 1, k\}$. Модель может быть бесконечной, конечной закольцованной и конечной замкнутой. Каждая клетка имеет свою окрестность. Устанавливаются правила, которые определяют переходы между состояниями клеток. Один шаг автомата - обход всех клеток, определяющий новое состояние $\sigma(t + 1)$ на основе о текущем состоянии клетки $\sigma(t)$ и ее окрестности $\mathcal{N}(t)$ согласно правилам φ : $\sigma(t + 1) = \varphi(\sigma(t), \mathcal{N}(t))$.

В математическом моделировании неизвестных систем, так называемых «черных ящиков», применяются различные модели, и в том числе клеточные автоматы, для изучения и предсказания поведения таких систем. Клеточные автоматы находят свое применение в криптографии, биологии и моделировании физических, экономических, популяционных процессах. Например, с помощью клеточных автоматов можно моделировать процесс роста кристаллов [2].

Описание модели

Игра «Жизнь»[3] — клеточный автомат, придуманный английским математиком Джоном Конвеем. Она имеет следующие правила:

- Игрок только размещает или случайно генерирует нулевое поколение;

- Каждая клетка на этой поверхности имеет восемь соседей, окружающих ее, и может быть живой или мертвой;
- Каждое следующее поколение рассчитывается на основе предыдущего с использованием следующих правил:
  - если мертвая клетка имеет три живые соседние клетки, то там зарождается жизнь;
  - если живая клетка имеет две или три живые соседние клетки, то эта клетка продолжает жить. В противном случае, клетка умирает.
- Игра прекращается, если
  - конфигурация на очередном шаге точно повторяет одну из более ранних конфигураций;
  - на очередном шаге ни одна из клеток не меняет свое состояние.

Многослойный перцептрон

Рассмотрим перцептрон с двумя входными узлами (состояние клетки и количество соседей у нее) и одним выходным (следующее состояние). Потребуется один скрытый слой [4], так как аппроксимация игры «Жизнь» не настолько сложная задача, и два скрытых узла в нем по первому пункту правил Хитона [4]. Получается перцептрон (рис 1.) с шестью весами w_i и тремя порогами b_i, которые нужно настроить.

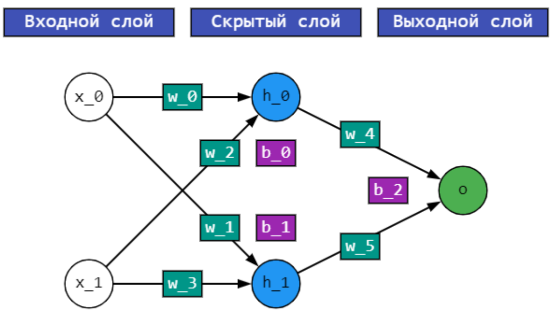

Рис. 1: Архитектура многослойного перцептрона.

К качестве функции активации возьмем сигмоиду $f(x) = 1/(1 + e^{-x})$. Она «упаковывает» интервал от $-\infty$ до $+\infty$ и выдает результаты в интервале (0, 1).

В данной работе постараемся обучить перцептрон до первого вида обучения, когда сеть развивается точно так же, как исходная система, начиная с любой начальной конфигурации, из статьи [5]. Начальные веса будут генерироваться от -10 до 10, а пороги - от -4 до 1.

Алгоритм обратного распространения ошибки

В статье [5] предлагается обучить нейронную сеть с помощью дельта-правила, однако рассмотрим его улучшение – **алгоритм обратного распространения ошибки**, который является алгоритмом машинного обучения на основе градиентного спуска по средней квадратичной ошибке выходного слоя.

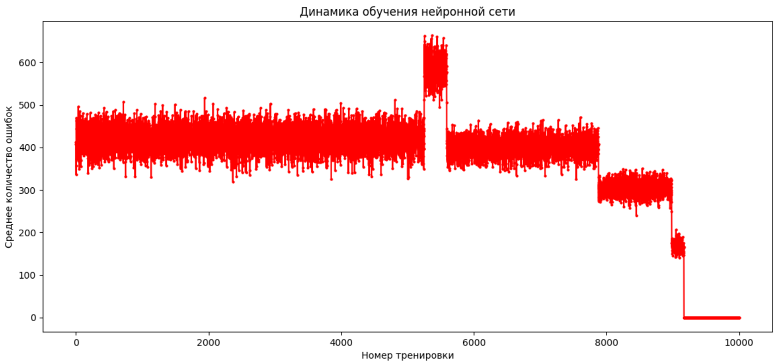

Рис. 2: график удачной попытки обучения на 10000 эпох.

Попытки обучения без лимитов поколений оказались неэффективны, потому что нейросеть долго обучалась правилу появлений «живых» клеток, так как в большинстве примеров преобладали «мертвые», которые тянули веса и пороги в сторону обучения правил о них, и не могла «понять», что «живые» клетки могут умирать от перенаселения, потому что примеров с этим явлением было крайне мало.

Для того чтобы решить эти проблемы и ускорить обучение, стоит поставить лимит 2, потому что в нулевом поколении преобладают «живые» клетки, а в первом - «мертвые», что поможет сети параллельно «выучить» правила для них обоих. Также ускорение позволило быстрее выявлять проблемы алгоритма (застревание в локальных минимумах функции ошибки и «паралича сети») и перезапускать обучение.

Коэфф.	Значение
w_0	0.11068505585632368
w_1	1.6140490074482765
w_2	1.1231821052847883
w_3	2.1197172145813132
w_4	-11.069308691851203
w_5	11.417776366120652
b_0	-3.9229089223740177
b_1	-4.201344761961087
b_2	-5.64691200600882

Таблица 1: Веса и пороги «обученной» нейронной сети с алгоритмом обратного распространения ошибки

Главная проблема – постоянная скорость обучения, зависящая от градиента ошибки. 1 успешное обучение на 150 попыток.

Удачная попытка обучения представлена на рисунке 2. А её веса и пороги приведены в таблице 1. Оптимальная скорость обучения варьируется от 0.2 до 0.5.

Модифицированный алгоритм устойчивого обратного распространения с уменьшением параметров

Справиться с недостатками прошлого алгоритма может **модифицированный алгоритм устойчивого обратного распространения с уменьшением параметров** [6], который использует только знаки частных производных для подстройки параметрических коэффициентов $\Delta_{i,j}$ и действует независимо для каждого параметра.

Если на текущем шаге частная производная по соответствующему весу или порогу поменяла свой знак, то это говорит о том, что последнее изменение было большим, и алгоритм проскочил локальный минимум. Однако выполнять «откат» будет контрпродуктивно, если общая ошибка уменьшилась. Данный алгоритм объединяет “индивидуальную” информацию о поверхности ошибки с ошибкой сети, чтобы решить для каждого параметра в отдельности, следует ли отменять шаг.

Алгоритм показал себя лучше предыдущего в скорости обучения. Проблемы застреваний в локальных минимумах функции ошибки также возник-

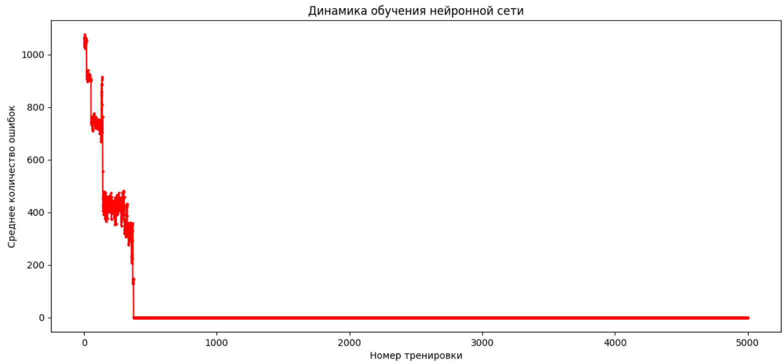

Рис. 3: график удачной попытки обучения на 5000 эпох.

ли, однако алгоритм «выбирался» из малых минимумов. И «параличей сети» не возникло: алгоритм возвращался из области насыщения с помощью параметрических коэффициентов. Однако процент успешных обучений вырос лишь немного: 3 успешных на 150 попыток.

Одна из удачных попыток обучения представлена на рисунке 3. А её веса и пороги приведены в таблице 2. В ходе работы были выявлены оптимальные параметры $\theta^+ = 1.2$, $\theta^- = 0.5$, $\Delta_{min} = 0.00001$, $\Delta_{max} = 0.01$, а начальный параметрический коэффициент для каждого веса и порога 0.0005.

Заключение

В процессе работы над статьей был реализован многослойный перцептрон с двумя алгоритмами его обучения. Был предложен способ сократить время обучения, выставив лимит поколений на каждом этапе до 2. По результатам может быть сделан вывод, что несмотря на то, что игра «Жизнь» – самый примитивный клеточный автомат, но тем не менее обладает непростой производной среднеквадратичной ошибки, что приводит к трудностям в процессе обучения нейронных сетей. Алгоритмы полагаются на удачные случайные начальные веса и пороги, которые быстро сходятся к решению [7]. Из этого следует, что среднеквадратичная ошибка имеет большое количество минимумов разной «глубины».

С учетом этой сложности, предложим два варианта решения проблемы. Первый - бесконечное запускание модифицированного алгоритма устойчивого обратного распространения с уменьшением параметров со случайными весами и порогами и лимитом поколений, пока не будет найден глобальный

Коэфф.	Значение
w_0	1.7747016838681917
w_1	0.18769040773208562
w_2	2.391405692710275
w_3	1.2853763296893144
w_4	42.575969840202355
w_5	-57.147982604809194
b_0	-5.276672938117282
b_1	-4.997180607315179
b_2	-17.444888764084585

Таблица 2: Веса и пороги «обученной» нейронной сети с модифицированным алгоритмом устойчивого обратного распространения с уменьшением параметров

минимум. Второй - создание идеального в теории алгоритма для игры, который сможет сканировать все локальные минимумы с возможностью «выпрыгнуть» на следующий при не нахождении там глобального минимума.

Кроме того, возможно улучшение архитектуры нейронной сети до сетей Колмогорова-Арнольда [8], которая может обеспечить более эффективное обучение. Помимо этого допустим переход на Q-обучение, основанный на выборе следующего поколения с помощью так называемой Q-таблицы, что является аналогом обучения с учителем для выбранной игры.

Список литературы

[1] Wolfram S. «Statistical Mechanics of Cellular Automata». Reviews of Modern Physics, 2013. https://content.wolfram.com/sw-publications/2020/08/statistical-mechanics-cellular-automata.pdf

[2] Сериков Д., Очоа Бикэ А.. «Теория клеточных автоматов как метод описания процесса кристаллизации урана». Electronic archive of Tomsk Polytechnic University. https://earchive.tpu.ru/bitstream/11683/17630/1/conference_tpu-2015-C49-097.pdf

[3] Gardner, M. «The fantastic combinations of John Conway's new solitaire game 'life'». Scientific American 223, 1970. https://web.stanford.edu/class/sts145/Library/life.pdf

[4] Heaton J.. «Introduction to Neural Networks With Java». Heaton Research, 2008. https://typeset.io/pdf/introduction-to-neural-networks-with-java-3yzltarx3p.pdf

[5] Wulff N. H., Hertz J. A.. «Learning cellular automaton dynamics with neural networks». NIPS'92: Proceedings of the 5th International Conference on Neural Information Processing Systems, 1992. https://proceedings.neurips.cc/paper/1992/file/d6c651ddcd97183b2e40 bc464231c962-Paper.pdf

[6] Igel C., Husken M..«Empirical evaluation of the improved Rprop learning algorithms»Institut für Neuroinformatik, Ruhr-Universität Bochum, 2001. https://sci2s.ugr.es/keel/pdf/algorithm/articulo/2003-Neuro-Igel-IRprop+.pdf

[7] Springer J., Kenyon. G. «It's Hard for Neural Networks To Learn the Game of Life». International Joint Conference on Neural Networks (IJCNN), 2021. https://openreview.net/pdf?id=uKZsVyFKbaj

[8] Liu Z., Wang Y., Vaidya S., Ruehle F., Halverson J., Soljačić M., Hou T., Tegmark M.. «KAN: Kolmogorov-Arnold Networks». arXiv:2404.19756, 2024. https://arxiv.org/pdf/2404.19756