Self-learning Monte-Carlo algorithm

Сорокин Никита*[a], Щур Лев Николаевич[b]

[a]Студент бакалавриата, Высшая Школа Экономики
[b]Научный руководитель, Заведующий кафедрой, Профессор:Московский институт электроники и математики им. А.Н. Тихонова / Базовая кафедра «Прикладные информационно-коммуникационные средства и системы» ВЦ РАН

АННОТАЦИЯ
Изучена возможность повышения точности оценки критической температуры при использование нейронных сетей, обученных на одной модели и примененной к другой модели.

ВВЕДЕНИЕ
Методы Монте Карло — это группа численных методов для изучения случайных процессов. Суть метода заключается в следующем: процесс описывается математической моделью с использованием генератора случайных величин, модель многократно обсчитывается, на основе сгенерированных данных вычисляются стат величины рассматриваемого процесса. Однако во многих задачах при увеличении сложности(размерности пространства и тд), количество операций, требуемых для генерации данных и подсчета статистик растет экспоненциально. Данная проблема похожа на проблему, которая часто встречается в машинном обучении называющаяся “проклятие размерности“ , однако с данной проблемой научились эффективно бороться в Data science сообществе.

Цель данной работы повторить и проверить результаты статьи[2] а также попробовать улучшить обобщаю способность модели и улучшить качество определение температуры Кюри с помощью экспериментов над архитектурой моделей.

В данной статье будет рассматриваться нахождение температуры Кюри для квадратных и треугольных решеток модели Изинга при помощи полносвязных и сверточных нейросетей, аналогично статье[2].

МОДЕЛЬ ИЗИНГА
Модель Изинга описывает спины $\sigma_i = +-1$ для $i = 1..N$ находящиеся на решетках разных типов, также она может быть охарактеризована энергией своего состояния, которая описывается формулой $H = -J\sum_{i,j}\sigma_i * \sigma_j$ [3], где $\sigma_{i,j}$ равна -1 или 1(спин вниз или вверх). Особый интерес представляет температура Кюри(температура фазового перехода), при которой вещество переходит в другое состояние(ферромагнетик в парамагнетик или наоборот, зависит от параметра J), что также характеризуется резким скачком значения теплоемкости $C_v = \beta^2\frac{\partial E}{\partial \beta}$. В данной статье мы рассмотрим решетки квадратные и треугольное решетки.

ГЕНЕРАЦИЯ РЕШЕТОК
Генерация квадратных и треугольных происходила при помощи алгоритма Вольфа [5][1]. Алгоритм реализован на языке C++ и Python. Ключевой особенностью алгоритма является шаг Вольфа, во время которого состояние частиц обновляется группами(кластерами) Рис.1. На каждом шаге алгоритма случайно равновероятно выбирается узел решетки, которая становится первым элементом кластера и помещается в очередь. Далее добавляются все “соседи”(разнонаправленные по спину) выбранной узел с вероятностью $p = 1 - e^{-2*\beta}$, аналогичная процедура повторяется для всех узлов, которые оказались в очереди или стеке, до того момента

пока очередь или стек не опустеют. Далее, меняем значения всех спинов, оказавшихся в кластере на противоположное.

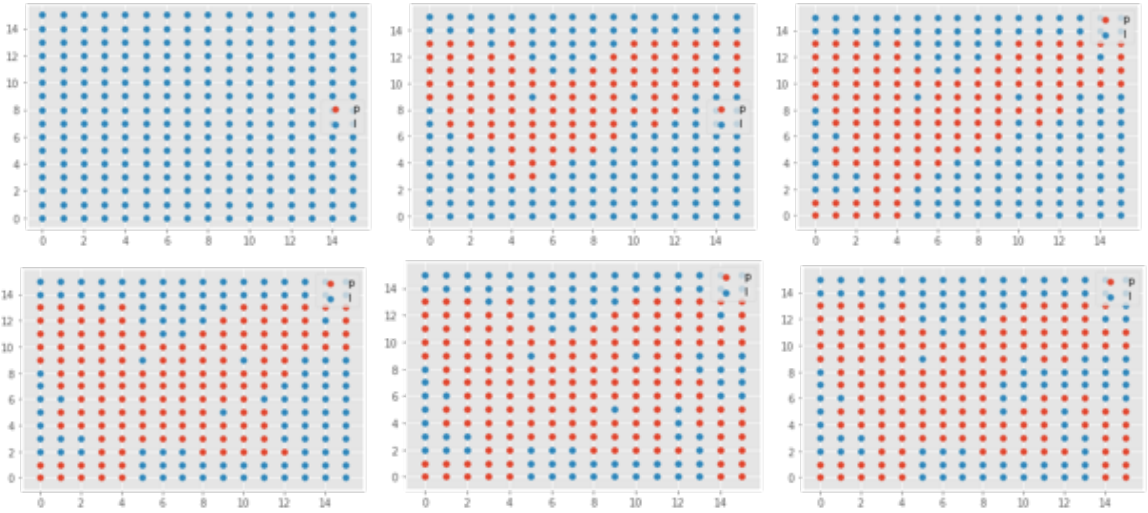

Рис. 1. Пример 5 “шагов” для решетки размера 16x16.

ОБУЧЕНИЕ НЕЙРОСЕТЕЙ

Ключевой идеей для нахождения температуры фазового перехода, с использованием моделей машинного обучения, это классификация состояния, в котором находится вещество. Тогда, в те решетки, для которых модель будет выдавать вероятность принадлежности к одному из состояний максимально близкую к 0.5, должны быть сгенерированы при температуре близкой к температуре Кюри.

Первым baseline решением была нейросеть состоящая из одного полносвязного слоя с 100 нейронами и функцией активации ReLu[4]. На вход модели подавались состояние спинов на решетке, массив из состоящий из значений 1 и -1 размера N*N, где N размер исходной решетки. Далее, к выходу из скрытого слоя нейросети представляющем из себя матрицу (кол-во объектов в обучающей выборок)x100, применялась функция активации ReLu и получившиеся матрица подавалась на выходной слой представляющий из себя 2 нейрона, в следствии чего мы получаем матрицу размера (кол-во объектов в обучающей выборок)*2. Чтобы получить вероятности принадлежности к каждому из состояний мы применяем к получившейся матрице функцию SoftMax[6](оптимизатор в для каждой модели Adam). После этого с помощь метода обратного распространения ошибки происходит обновление весов нейросети. Обучение происходило на квадратной решетки размера N, а предсказание на треугольной решетке размера N Рис.2. Качество классификации представлено в Табл.1., как можно заметить из графиков Рис.2, предсказываемая температура Кюри приближается к температуре кюри для бесконечной решетки(3.65). Также, была протестирована сверточная нейроная сеть с одним сверточным слоем, ядром размера 3X3, шагом 1, полносвязным слоем состоящем из 64 нейронов и dropout слоем[7] Табл.2.

Данное решение позволяет определить температуру фазового перехода, однако не обладает достаточной обобщающей способностью, хотелось бы использовать одну модель для решеток размерности меньше N. Для этого была пред обучена полносвязная нейросеть на решетках размера N = 256, а предсказание происходило на решетках меньшего размера с зануленными весами там, где состояние частицы не известно. Таким образом, данные подаваемые на вход были аналогичны данным в предыдущем решении, однако для того чтобы подогнать под нужную для входа нейросети размерность, при этом не потеряв обобщающую способность модели, неизвестные состояния инициализируются нулями, что схожа с идей padding, которая используется в компьютерном зрении Рис.3 Табл. 3. Аналогично были проведены эксперименты со сверточной сетью Табл.

4.

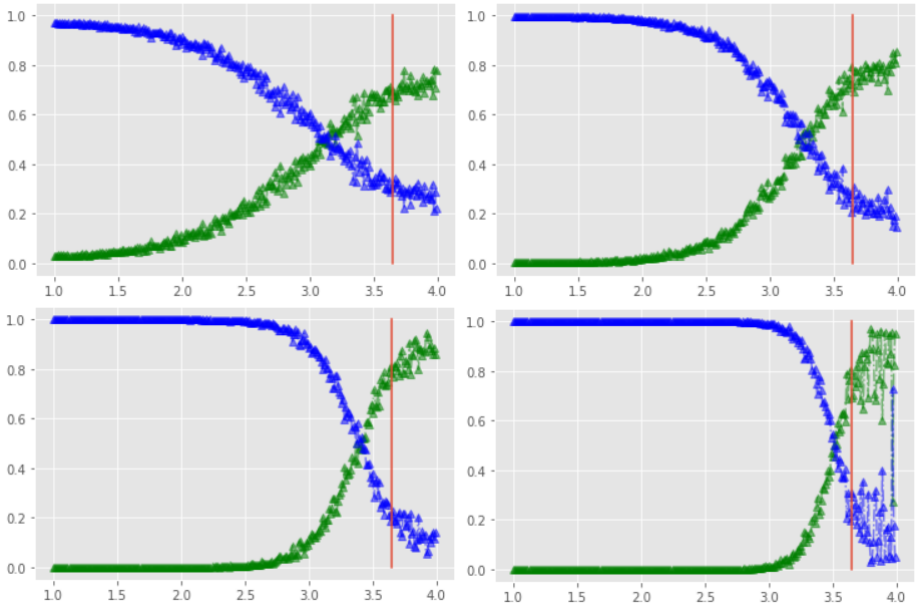

Рис. 2. Графики зависимости температуры от вероятности для решеток размера 16, 32, 64, 128

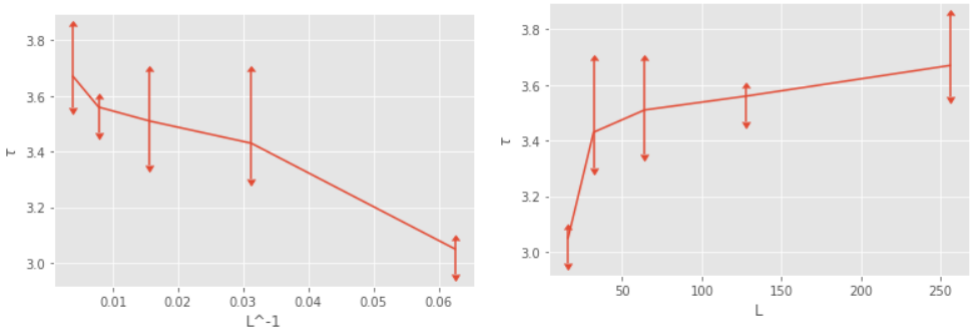

Рис. 3. Грифики зависимости температуры фазового перехода от размера решетки для полносвязной нейросети с Padding

	Размер решетки	Температура фазового перехода	Точность
1	16	3.09	0.784884
2	32	3.23	0.848804
3	64	3.40	0.900033
4	128	3.51	0.932027
5	256	3.59	0.942990

Таблица 1. Качество полносвязной нейросети

	Размер решетки	Температура фазового перехода	Точность
1	16	3.01	0.81681
2	32	3.10	0.85601
3	64	3.45	0.85601
4	128	3.54	0.85627
5	256	3.67	0.955990

Таблица 2. Качество сверточной нейросети при обучении 1 к 1

	Размер решетки	Температура фазового перехода	Точность
1	16	1.13	0.126
2	32	3.15	0.797
3	64	3.50	0.897
4	128	3.55	0.919
5	256	3.67	0.9460

Таблица 3. Качество полносвязной нейросети с Padding

	Размер решетки	Температура фазового перехода	Точность
1	16	2.95	0.75
2	32	3.14	0.77
3	64	3.5	0.79
4	128	3.54	0.84
5	256	3.6	0.94

Таблица 4. Качество сверточной нейросети с Padding

РЕЗУЛЬТАТЫ

Как можно заметить качество в случае сверточных нейросетей оказалось чуть лучше чем для полносвязных, однако оно не столь значимо, поэтому оно могло быть вызвано не дообучением полносвязных нейросетей табл.1-4. Итоговая модель в статье[2] показывала лучшее качество порядка 99% против наших 94.60% табл.3, однако это может этого они достичь за счет переобучения, тк график[2] хуже чем рис.2, содержит резкие скачки. А в нашем же случае температура фазового перехода приближается практически линейно к температуре для фазового перехода для решетки бесконечного размера равной 3.65, с учетом погрешности.

ЗАКЛЮЧЕНИЕ

Мы проверили результаты полученные в статье[2] на решетках большего размера и изменили конфигурацию исходной сети, что привело к лучшему показателю линейного приближения температуры фазового перехода с увеличением размера решетки к температуре решетки бесконечного размера. Также было придумано частичное решение проблемы оценки температуры Кюри для решеток разного размера.

ИСХОДНЫЕ КОДЫ

https://github.com/Shroedingercat/Self-learning-Monte-carlo-algorithms

ССЫЛКИ

Список литературы

1. Statistical Mechanics Algorithms and Computations Werner Krauth Laboratoire de Physique Statistique, Ecole Normale Sup′erieure, Paris
2. Letter Published: 13 February 2017 Machine learning phases of matter Juan Carrasquilla and Roger G. Melko ttps://www.nature.com/articles/nphys4035
3. Brout R. Phase transition,
4. Rectified Linear Units (ReLU) in Deep Learning https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning
5. U. Wolff, Phys. Rev. Lett. 62Ъ (1988) 361
6. Softmax function https://en.wikipedia.org/wiki/Softmax$_{f}unction$
7. Dropout — метод решения проблемы переобучения в нейронных сетях https://habr.com/ru/company/wunderfund/blog/330814/