

Кафедра прикладной математики, информационных технологий и информационной безопасности

Тема доклада:
"Поиск экстремальных конфигураций в геометрических задачах при помощи библиотеки PyTorch"

Выполнил:
Студент 2 курса факультета математики и компьютерных наук
группа 2 ПМ, Никитенко Владислав

Майкоп
2021

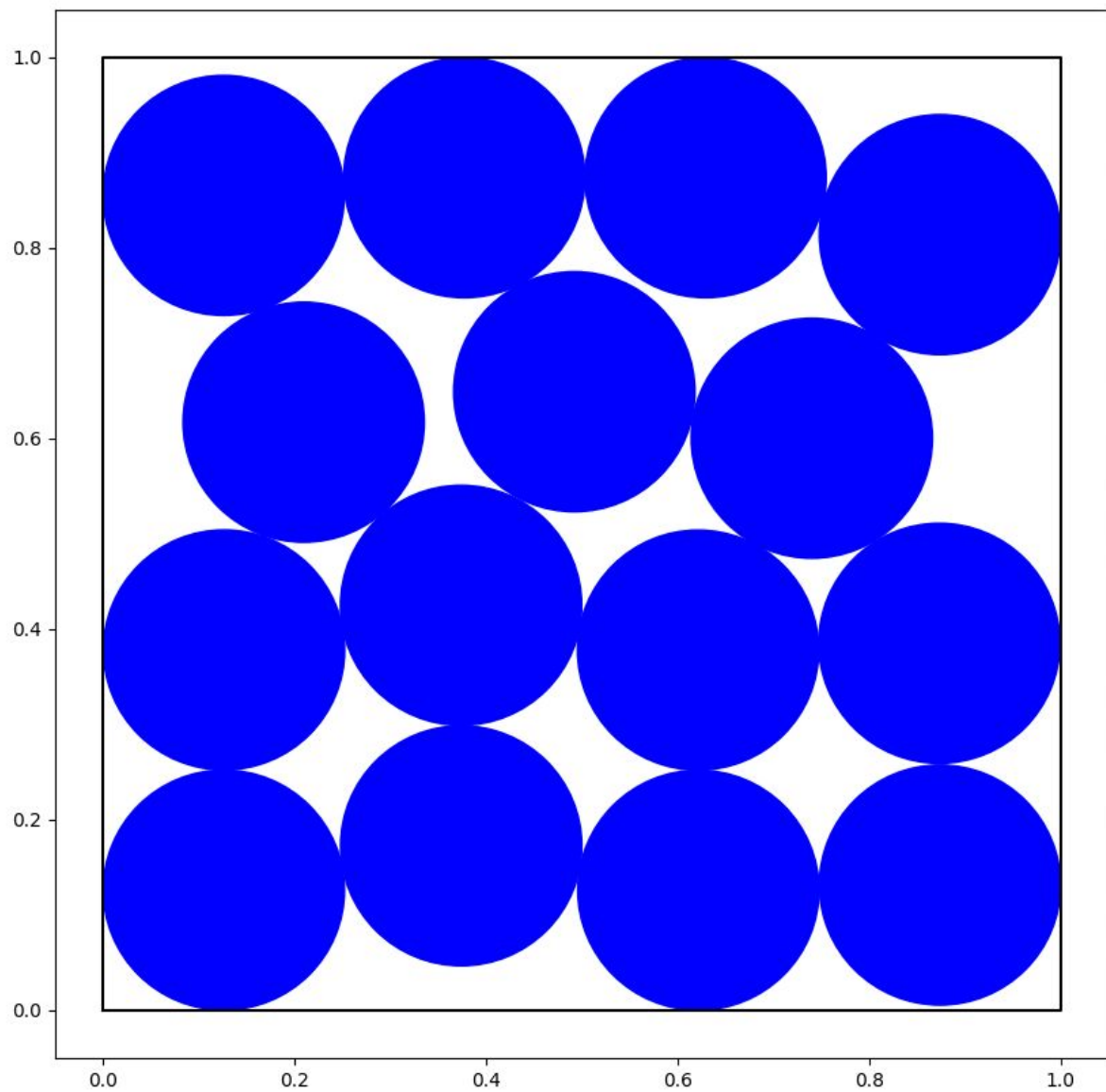
Постановка задачи

Задачи упаковки — это класс задач оптимизации в математике, в которых пытаются упаковать объекты в контейнеры. Цель упаковки — либо упаковать отдельный контейнер как можно плотнее, либо упаковать все объекты, используя как можно меньше контейнеров, либо нахождение конфигурации, которая упаковывает один контейнер с максимальной плотностью. При этом объекты не должны пересекаться и объекты не должны пересекать стены контейнера.

Работа программы

Создается квадратная область, в нее помещаются случайным образом сгенерированные n точек. Далее программа увеличивает их радиусы, при этом ищет лучшее расположение. Все вычисления происходят посредством нейронной сети, которая обучается градиентным спуском.

*Градиентный спуск — это эвристический алгоритм, который выбирает случайную точку, рассчитывает направление скорейшего убывания/возрастания функции (пользуясь градиентом функции в данной точке), а затем пошагово рассчитывает новые значения функции, двигаясь в выбранную сторону. Если убывание/возрастание значения функции становится слишком медленным, алгоритм останавливается и говорит, что нашел минимум.



ПРИМЕР
ВЫПОЛНЕНИЯ



Скорость обучения

Скорость обучения стоит воспринимать как ширину шагов. В некоторых случаях бывает так, что слишком широкие шаги вообще не позволяют достичь минимума, и машина бесконечно перешагивает через него, затем градиент «разворачивает» ее обратно, и алгоритм снова перескакивает через минимум. Маленькая скорость обучения хоть и придает точности, зато, конечно, увеличивает время на обучение нейросети.

Скорость обучения слишком велика:



Цель

ПОДБОР
СКОРОСТИ
ОБУЧЕНИЯ



Скорость обучения слишком мала:

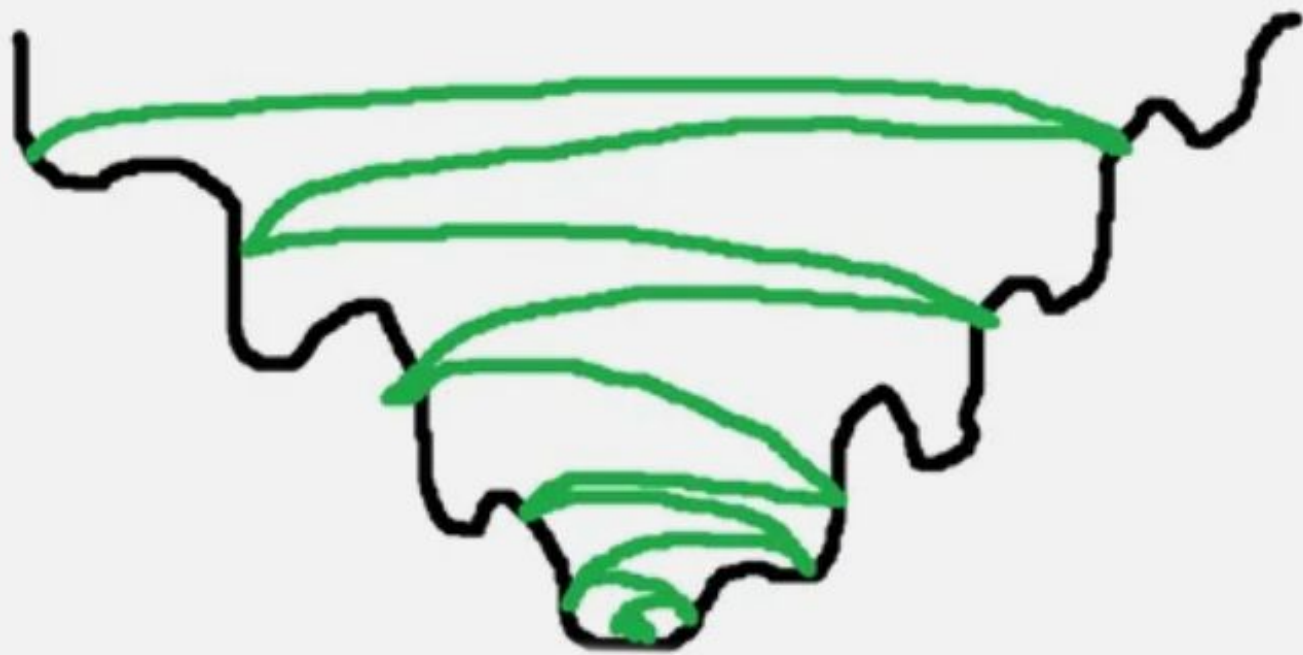


Цель

ПОДБОР
СКОРОСТИ
ОБУЧЕНИЯ



Затухающая скорость обучения:



Цель

ПОДБОР
СКОРОСТИ
ОБУЧЕНИЯ



20_10: (100, 14.419323831796646, 0.14419323831796646)
30_10: (100, 14.557886108756065, 0.14557886108756066)
40_10: (100, 14.594097211956978, 0.14594097211956977)
50_10: (100, 14.652108877897263, 0.14652108877897263)
60_10: (100, 14.65576159954071, 0.1465576159954071)
70_10: (100, 14.666099473834038, 0.14666099473834038)
80_10: (100, 14.678517773747444, 0.14678517773747443)
90_10: (100, 14.695189774036407, 0.14695189774036407)
100_10: (100, 14.747388735413551, 0.14747388735413552)
110_10: (100, 14.727175563573837, 0.14727175563573838)
120_10: (100, 14.719042524695396, 0.14719042524695397)
130_10: (100, 14.752179980278015, 0.14752179980278016)
140_10: (100, 14.742900416254997, 0.14742900416254998)
150_10: (100, 14.759741485118866, 0.14759741485118866)
160_10: (100, 14.738642171025276, 0.14738642171025276)
170_10: (100, 14.739636912941933, 0.14739636912941934)
180_10: (100, 14.766204684972763, 0.14766204684972764)
190_10: (100, 14.741886541247368, 0.14741886541247368)
200_10: (100, 14.74767418205738, 0.1474767418205738)
210_10: (100, 14.759666308760643, 0.14759666308760644)
220_10: (100, 14.758980199694633, 0.14758980199694632)
230_10: (100, 14.75654511153698, 0.14756545111536978)
240_10: (100, 14.767590835690498, 0.14767590835690497)
250_10: (100, 14.766855612397194, 0.14766855612397195)
260_10: (100, 14.757829666137695, 0.14757829666137695)
270_10: (100, 14.743579804897308, 0.14743579804897308)
280_10: (100, 14.758969902992249, 0.14758969902992247)
290_10: (100, 14.758339539170265, 0.14758339539170265)
300_10: (100, 14.746790930628777, 0.14746790930628775)

ПОДБОР ЗНАЧЕНИЯ PATIENCE



Первые выводы

За 5000 запусков программы получилось 304 рекордных значения, что составляет 6.08%

*Данные вычислялись при количестве шаров равном 15, значение считалось рекордным при разнице с рекордом меньше либо равной 10^{-5}

Спасибо за внимание!