
Использование генетических алгоритмов для повышения скорости обучения нейронных сетей

В.А. Цыганков, О.А. Шабалина, А.В. Катаев

Волгоградский государственный технический университет, Волгоград

Аннотация: Целью данной работы является исследование способов ускорения обучения нейронных сетей с помощью генетических алгоритмов и изучение зависимости скорости работы генетических алгоритмов от коэффициента мутации. В данном исследовании была реализована программа на графической платформе «Unity» с использованием генетических алгоритмов и мутациями для определения оптимального их коэффициента. Проведенный эксперимент показал, что скорость обучения действительно зависит от коэффициента мутации, а наибольшая скорость обучения получилась при 5-7,5%.

Ключевые слова: машинное обучение, глубокое обучение, генетический алгоритм, оптимизация, нейронная сеть, искусственный нейрон, мутация, искусственный интеллект, неигровой персонаж, оптимизация.

Введение

В настоящее время искусственные нейронные сети (далее НС) все чаще используется в научном сообществе при имитации теоретических объектов и явлений. Все проблемы с качеством полученной от них информации напрямую связаны с их способностью копировать реальный объект и выдавать прогнозы относительно его поведения.

Для повышения точности симуляции объекта НС нужно обучать и адаптировать к исполнительской среде. Основным фактором в этом процессе выступает время, которое увеличивается от количества имитируемых параметров. Для очень сложных моделей этот процесс может растянуться на месяцы и годы (Например, имитация галактики «Млечный Путь»). Для ускорения этого этапа, компании арендуют или покупают суперкомпьютеры. В случае аренды они платят за время его использования, а в случае покупки – время суперкомпьютера делится на все отделы компании.

Скорость обучения нейронных сетей зависит от метода обучения, объемов данных и количества имитируемых параметров. Работа посвящена исследованию способов ускорения оптимизации НС.

Анализ способов обучения нейронных сетей

В настоящее время написано большое количество работ с разбором проблем, которые появляются в процессе обучения НС [1-3]. Так, в работе [4] описаны факторы, влияющие на скорость обучения нейронных сетей: плохое обусловливание гессенских матриц, появление седловых точек, градиенты взрыва, малое количество обучающей информации. Гессенские матрицы описывают кривизну функции переменных. Их плохое обусловливание приводит к очень медленным шагам обработки, из-за чего существенно увеличивается время обучения. Седловые точки могут быть перепутаны с локальными минимумами. Это приводит к выводу некорректного ответа и ошибочным прогнозам. Градиенты взрыва заставляют норму градиента расти экспоненциально и делают обучение нестабильным. Аппроксимация градиентов с использованием слишком маленького количества информации обучающих примеров может привести к неточным оценкам и заставить градиент двигаться в направлении, отличном от наиболее оптимального.

Поэтому в процессе обучения требуется оптимизация входных данных, уровня контроля искусственного интеллекта над проектом и его поведенческой линии. Она позволяет нивелировать эти недостатки путем адаптации системы к выстроенным условиям для более эффективного выполнения представленной задачи. Для этого, в настоящее время, используют три основных метода обучения: метод обратного распространения, метод упругого распространения и генетические алгоритмы обучения.

Метод обратного распространения [5-6] основан на распределении штрафа за ошибку между весами, внесшими вклад в выход. Каждый вес

вносит вклад в несколько выходов, таким образом, алгоритм обучения НС усложняется от количества уровней весов и их выходов. Для алгоритма на его основе используют дифференцируемые функции активации [7]. Это связано с тем, что распространение в обратном направлении определяется разностью между ответами, а также произведением между ним и производной функцией от входного значения. Далее рассчитывают градиент для всех исходящих связей. После этого, с учетом полученных данных, проводят обновление весов и вычисляют величину изменения с помощью функции МОР. Вместе с тем учитывают и скорость обучения в моменте из-за чего скорость работы данного метода велика.

В работе [8] рассмотрен метод упругого распространения. Он был предложен как альтернатива предыдущему способу обучения, который требует слишком много времени и становится неудобным, если результаты нужно получить в короткие сроки. Для подгонки весовых коэффициентов он использует только знаки производных частного случая. При этом выдерживается правило, позволяющее определить значение коррекции коэффициента веса. Данный метод быстро обрабатывает однопараметрические функции, но имеет проблемы в обработке многопараметрических.

В задачах моделирования и оптимизации применяются генетические алгоритмы обучения (Далее ГА) [9]. Они осуществляются при помощи случайного подбора, последующего комбинирования и вариации требуемых параметров с помощью использования механизмов, при которых, наименее эффективные варианты отбрасываются.

Объединение ГА и НС известно под аббревиатурой COGANN или «Эволюционный алгоритм» [10]. Процесс обучения проходит в несколько этапов: формирование начальной популяции и её обучение, оценка пригодности, ранжирование, селекция и скрещивание, мутации. Вначале

необходимо сформировать начальную популяцию последовательностей – хромосом, т.е. множество потенциальных решений. Начальная популяция формируется случайно, но можно в неё включать также решения, которые получены с использованием другого оптимизационного метода. Размер начальной популяции определяется экспериментально. Затем осуществляется процесс синтеза новых решений (поколений) с использованием трех основных операторов: репродукция, скрещивание, мутация [11]. Для каждого индивидуума определяется функция пригодности, она производит числовую оценку, для определения пригодности конкретного индивидуума и отбирает наиболее подходящие для выполнения задачи. Поскольку приспособленность любого индивидуума не зависит от всех остальных, эти вычисления можно производить параллельно. Скрещивание предполагает создание потомка из генетического кода одного или нескольких родителей без каких-либо изменений. Мутация используется для внесения разнообразия по ходу обучения, так как разнообразие новой популяции будет ограничиваться разнообразием старой и будет с каждым новым поколением только уменьшаться. После этого создается новая популяция и цикл повторяется по новой, пока не будет достигнут требуемый результат.

Наиболее распространено использование ГА при оптимизации многопараметрических функций. Многие реальные задачи могут быть сформулированы, как поиск оптимального значения, где значение – некая сложная функция, зависящая от некоторых входных параметров. Одним из наиболее важных преимуществ ГА для обучения НС является отсутствие необходимости информации о поведении функции и незначительное влияние возможных разрывов на процессы оптимизации.

Таким образом, среди рассмотренных методов ГА позволяет наиболее быстро оптимизировать многопараметрические функции путем их совокупной параллельной обработки.

Зависимость скорости обучения от коэффициента мутации

Одним из параметров, влияющих на скорость обучения, является коэффициент мутации. Для изучения зависимости скорости работы генетических алгоритмов для обучения НС от коэффициента мутации создается имитационная модель жизни объектов с изменяемым поведением в виртуальной среде. В течение своей жизни, определяемой заданным промежутком времени, каждый объект стремится достичь максимально возможное количество целей, предопределенных средой. Для моделирования поведения объектов использован искусственный интеллект (далее ИИ), для его обучения используются ГА. ИИ может управлять движениями поколения неигровых персонажей (далее НИП), основываясь на результатах поколения-предка.

Время существования каждого поколения составляет 100 секунд. На поле для каждого объекта может находиться только одна цель, в случае её достижения должна появиться другая. На объекты действует гравитация и инерция аналогичные земной.

Для проведения эксперимента была реализована программа на графической платформе «Unity» с использованием генетических алгоритмов и мутациями. Сцена состоит из поля, на котором находятся два типа объектов: объекты НИП и объекты «Цель». НИП состоит из головы (Воксель, способный поворачиваться на 180° в вертикальной и на 30° в горизонтальной плоскости) и тела (пяти последовательно соединенных вокселей, способных поворачиваться на 180° в вертикальной плоскости относительно друг друга). Цель – одиночный воксель.

Сцена состоит из поля, на котором находятся 100 «комнат», в которых располагаются по одному НИП и Цели. Цель появляется в пределах своей комнаты, но НИП может её покинуть. Каждый НИП видит только свою цель и не может взять чужую. Как только он достигает Цели, она появляется в

другом месте комнаты, а счетчик его увеличивается на единицу и записывается время достижения данной Цели. Они представлены на рис. 1.

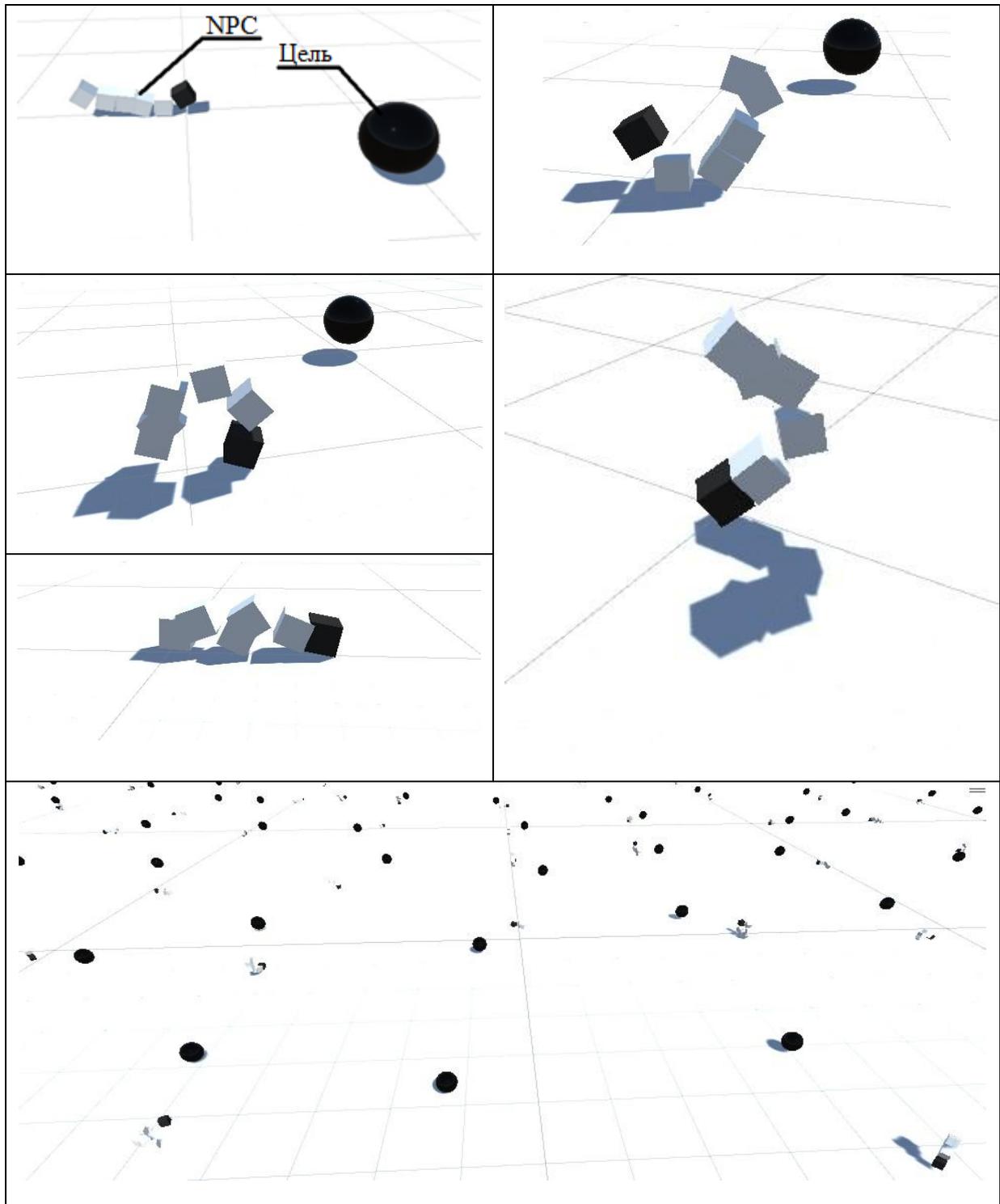


Рис. 1. – НИП и Цель

До начала теста выставлялся определенный статичный коэффициент мутации (Коэффициент мутации оценивается в процентах - это означает, что

каждый символ в паттерне объекта имеет шанс быть изменённым другим случайным символом) и анализировалась эффективность обучения ИИ для достижения поставленной цели. Каждому НИП присваивается свой счётчик, в который записывается количество и время каждой собранной Цели.

После начала теста, спустя некоторый промежуток времени, нынешнее поколение «погибает», но НИП с наибольшим счетом (и с минимальным временем, если таких индивидуумов несколько) передает свой паттерн поведения следующему поколению с небольшими изменениями, зависящими от коэффициента мутации.

Начальная популяция НИП имела случайные значения в паттерне поведения и не могла достигнуть заданной цели. Но с каждой итерацией их поведение оптимизировалось по селективному методу при помощи ГА и количество собранных Целей увеличивалось. Отбор проводился элитарным методом (только один предок с наивысшим счетом и наименьшим временем «давал» потомство). Таким образом, скорость обучения НС происходила по типу функции кубического корня (Рис. 2), пока их поведение не было максимально оптимизировано. После этого количество собранных за одно поколение целей не перестало изменяться и остановилось на 185 – 190 штук за цикл.

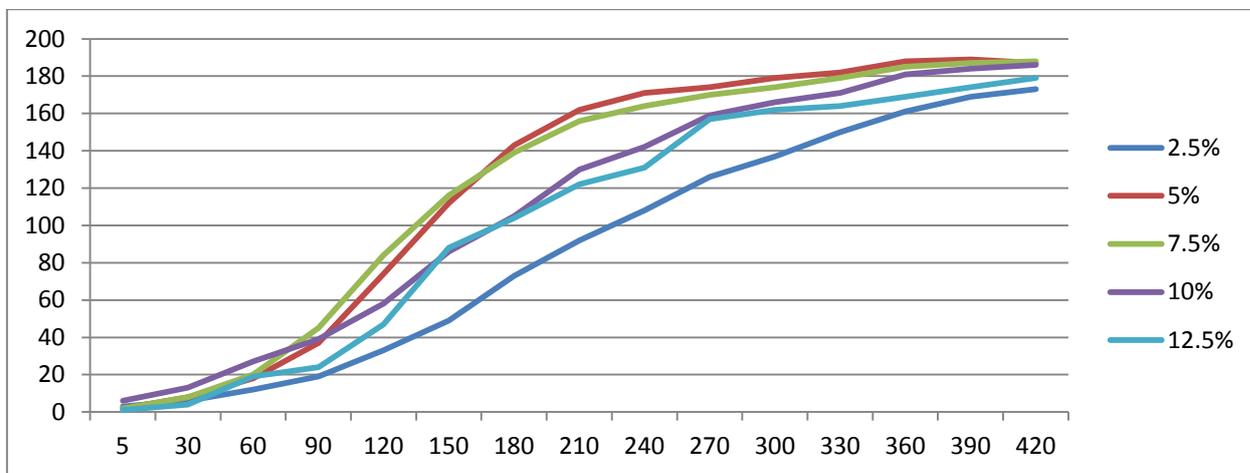


Рис. 2. – График зависимости количества собранных целей (Y) от поколения (X)

После проведенного эксперимента выяснилось, что наибольшая скорость обучения получилась при коэффициенте мутаций в 5-7,5%. При более низком значении процесс идет более стабильно, но медленней. При процентах более 7,5 график приобретает скачкообразный вид, при этом скорость обучения снижается.

Литература

1. Алешин С.П., Бородина Е.А. Нейросетевое распознавание классов в режиме реального времени // Инженерный вестник Дона, 2013, №1. URL: ivdon.ru/magazine/archive/n1y2013/1494
2. Кураева Е.С. Проблемы обучения нейронных сетей // Молодой ученый. Май 2020. №14 (304). С. 90. URL: moluch.ru/archive/304/68605/
3. Самонина О. А. Методы и проблемы обучения многослойной нейронной сети, используемой для оценки характеристик проектируемых информационных систем // Известия Петербургского университета путей сообщения. 2008. №4. С. 148-159. URL: moluch.ru/archive/304/68605/ URL: cyberleninka.ru/article/n/metody-i-problemy-obucheniya-mnogosloynnoy-neyronnoy-seti-ispolzuemoy-dlya-otsenki-harakteristik-proektiruemyh-informatsionnyh-sistem/viewer
4. Ian G., Yoshua B., Aaron C. Deep Learning. MIT Press book 2018. pp. 653.
5. Паттерсон Д., Гибсон А. Глубокое обучение с точки зрения практика. Москва: ДМК Пресс, 2018. 418 с.
6. Горелова А.В., Любимова Т.В. Алгоритм обратного распространения ошибки // Наука и современность. 2015. №38. С. 151-156. URL: cyberleninka.ru/article/n/algorithm-obratnogo-rasprostraneniya-oshibki

7. Пучков Е.В. Сравнительный анализ алгоритмов обучения искусственной нейронной сети // Инженерный вестник Дона, 2013, №1. URL: ivdon.ru/ru/magazine/archive/n4y2013/2135
8. Перков А. С., Жангиров Т. Р., Лисс А. А. Сравнение методов обучения нейронных сетей в задаче классификации // Известия СПбГЭТУ «ЛЭТИ». Июнь 2019. № 6. С. 1-9. URL: researchgate.net/publication/334849265_Comparison_of_neural_network_training_methods_in_classification_problem_in_Russian
9. Stastny J., Skorpil V. Genetic algorithm and neural network. Brno University of Technology. August 2007. pp. 6.
10. Schaffer J. D., Whitley L. D., Eshelman L. Combinations of genetic algorithms and neural networks: a survey of the state of the art. COGANN-92, 1992. pp. 116
11. Шумков Е.А., Чистик И.К. Использование генетических алгоритмов для обучения нейронных сетей // Политематический сетевой электронный научный журнал Кубанский государственный аграрный университет. 2013. №91. С. 455-464. URL: cyberleninka.ru/article/n/ispolzovanie-geneticheskikh-algoritmov-dlya-obucheniya-neyronnyh-setey

References

1. Aleshin S.P., Borodina E.A. Inzhenernyj vestnik Dona. January 2013, №1. URL: ivdon.ru/magazine/archive/n1y2013/1494
 2. Kuraeva E.S. Young Scientist. May 2020, №14 (304). URL: moluch.ru/archive/304/68605/
 3. Samonina O. A. Izvestiya Peterburgskogo universiteta putej soobshcheniya. 2008, №4 URL: cyberleninka.ru/article/n/metody-i-problemy-obucheniya-mnogosloynoy-neyronnoy-seti-ispolzuemoy-dlya-otsenki-harakteristik-proektiruemyh-informatsionnyh-sistem/viewer
 4. Ian G., Yoshua B., Aaron C. Deep Learning. MIT Press book, 2018. 653 p.
-



5. Patterson D., Gibson A. Glubokoe obuchenie s toчки zrenija praktika. [Deep Learning from a Practitioner's Point of View]. Moskva: DMK Press, 2018. 418 p.
6. Gorelova A.V., Ljubimova T.V. Nauka i sovremennost. 2015, №38. URL: cyberleninka.ru/article/n/algorithm-obratnogo-rasprostraneniya-oshibki
7. Puchkov E.V. Inzhenernyj vestnik Dona. December 2013, №1. URL: ivdon.ru/ru/magazine/archive/n4y2013/2135
8. Perkov A.S., Zhangirov T.R., Liss A.A. Izvestiya SPbGETU «LETI». June 2019, № 6. URL: researchgate.net/publication/334849265_Comparison_of_neural_network_training_methods_in_classification_problem_in_Russian
9. Stastny J., Skorpil V. Genetic algorithm and neural network. Brno University of Technology. August 2007. pp. 6.
10. Schaffer J. D., Whitley L. D., Eshelman L. Combinations of genetic algorithms and neural networks: a survey of the state of the art. COGANN-92, 1992. С. 116
11. Shumkov E.A., Chistik I.K. Politematicheskij setevoy elektronnyj nauchnyj zhurnal. 2013, №91. URL: cyberleninka.ru/article/n/ispolzovanie-geneticheskikh-algoritmov-dlya-obucheniya-neyronnyh-setey