



Рис. 1 - Структура кластера Hadoop

Список использованных источников:

1. KevinT. Smith, BorisLublinsky. Professional Hadoop Solutions. 2013.
2. Sammer E. Hadoop Operations. 2012 - 41с.
3. Tom White. Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale. 2015 - 283с.

МЕТОДЫ УЛУЧШЕНИЯ АЛГОРИТМА Q-ОБУЧЕНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Волчек А.Ю., Соболев А.В.

Волорова Н.А. – к.т.н., доцент

В настоящее время применение машинного обучения помогает людям решать задачи в разных сферах жизни: медицина, физика, химия, системы безопасности, игровая индустрия и др. Уже сейчас есть прототипы самодвижущихся автомобилей, которые сами контролируют движение транспорта. Одним из главных методов для обучения такого рода искусственного интеллекта является метод обучения с подкреплением – самостоятельное и уже вполне сформировавшееся направление кибернетических исследований, одна из разновидностей которого – Q-обучение.

Q-обучение — метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучение с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из преимуществ Q-обучения — то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды. Применяется для ситуаций, которые можно представить в виде марковского процесса принятия решений.

Q-обучение является важной вехой, однако известны некоторые ограничения этого алгоритма, для которых предложены несколько улучшений. В работе предложено несколько улучшений, в каждом из которых поясняется причина ограничений. Мы постарались выделить разноплановые улучшения, относящиеся к различным этапам алгоритма.

Двойное Q-обучение. Параметры нейросети в глубоком Q-обучении оптимизируются стохастическим градиентным спуском, минимизируя функцию потерь:

$$(R_{t+1} + \gamma_{t+1} \max_{a'} q_{\theta}(S_{t+1}, a') - q_{\theta}(S_t, A_t))^2 \quad (1)$$

В работе (vanHasselt 2010) показано, что из-за применения операции максимизации в уравнении (1) оценка Q-функции почти всегда является смещённой. Предложенный в данной работе подход позволяет частично избавиться от этой проблемы. Предлагается поддерживать рядом ещё одну нейросеть (targetnetwork), параметры в которую копируются из основной раз в несколько итераций. Вместо максимизации в уравнении (1) мы сначала выбираем действие с максимальным значением Q-функции, основываясь на данных из основной сети, но дальше используем значения Q-функции для этого действия из дополнительной сети.

Приоритизированный буфер опыта. Буфер опыта (experiencereplay) значительно ускоряет обучение и улучшает его стабильность, позволяя обучаться на ранее виденных ситуациях. В обычном его варианте мы выбираем обучающее множество равновероятно среди всех его элементов. Очевидно, что на практике существует много “простых” ситуаций, для которых сеть выучила хорошую аппроксимацию Q-функции и некоторое количество “сложных”. В приоритизированном буфере опыта мы выбираем каждый элемент в

обучающее множество с вероятностью, пропорциональной значению функции потерь на этом элементе во время предыдущей итерации обучения.

Многошаговое Q-обучение. Обычное Q-обучение обновляет значение Q-функции на основе награды на данном шаге и Q-значения на следующем шаге для жадно выбранного наилучшего действия. Альтернативой этому подходу является использование аккумулированной награды за несколько шагов.

Предложенные методы улучшения алгоритма Q-обучения позволяют ускорить обучение и увеличить стабильность получаемых результатов. Мы рассказали про некоторые улучшения алгоритма Q-обучения, которые могут быть внедрены в стандартный алгоритм для получения state-of-the-art производительности. Кроме них существует множество других возможных кандидатов, которые мы не рассмотрели в данной работе.

Список использованных источников:

1. vanHasselt, H. 2010, DoubleQ-learning. "Advances in Neural Information Processing Systems".
2. Sutton, R. S., and Barto, A. G. 1998. "Reinforcement Learning: An Introduction". The MIT press, Cambridge MA.
3. Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2015. "Prioritized experience replay". In proc of ICLR.
4. van Hasselt, H; Guez, A.; and Silver, D. 2016. "Deep reinforcement learning with double Q-learning".
5. Sutton, R. S. 1988. "Learning to predict by the method of temporal differences".

АСПЕКТЫ СОЗДАНИЯ УНИВЕРСАЛЬНОЙ СИСТЕМЫ ОРГАНИЗАЦИИ УЧЁТА И ОТЧЁТНОСТИ ПРЕДПРИЯТИЯ В СФЕРЕ УСЛУГ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Воробей Ю.Г.

Чернявский Ю. А. – к. т. наук, доцент

На сегодняшний день лучшим решением при выборе типа приложения является универсальное приложение. Универсальное приложение может эксплуатироваться пользователями разных направлений одной сферы, где используются одни и те же функции. Например, в сфере услуг организацией работы с клиентами может заниматься как медицинские центры, так и салоны красоты. Данные для заполнения будут разные, однако функции, которые позволяют организовать запись клиентов, исследовать их поток и другие, одинаковы в обоих случаях.

Такие системы должны удовлетворять следующим критериям. Для начала они имеют глобальный характер, поэтому должны быть доступны для разных регионов мира. Для этого требуется вводить не только поддержку разных языков, но и поддержку разных способов представления информации, которая может отличаться в разных странах, таких как представление дробных чисел или валюты.

Одной из наиболее явных характеристик универсальных систем является их большая нагрузка из-за количества пользователей разных направлений. Поэтому важным критерием является устойчивость системы к сбоям и её быстродействие. Правильная архитектура расположение кэшей и создание пула соединений с базой данных поможет организовать быстродействующую систему, а различные инструменты по проверке качества кода и поиску возможных багов, а также разнообразное тестирование, поможет создать систему, достаточно устойчивую к возникновению ошибок.

Также для лучшего отклика системы необходимо установить так называемые "жадные таймауты". Это позволит пользователю не ждать долго ответа системы. При жадном таймауте пользователь просто получит сообщение, и запрос повторится. Главное поставить таймаут правильно, чтобы система не выкидывала его слишком часто, но и чтобы достаточно быстро реагировала.

Также важным критерием является быстрый поиск по запрашиваемым данным. Для этого как раз подходит организация системы кэшей и пула соединений с базой данных. Однако необходимо учитывать и правильность организации схемы базы данных, а также различные инструменты базы для организации данных и быстрого поиска.

Как ранее говорилось, один из наиболее важных критериев по созданию универсальной системы – её устойчивость к возникновению ошибок. Для этого систему тщательно тестируют и проводят через дополнительные инструменты по определению качества кода. В процессе выявляются важные аспекты, влияющие на её работу. Например, при функциональном тестировании можно определить, какие моменты не учтены при разработке системы, а при нагрузочном – сколько одновременных клиентов может выдержать система.

Таким образом, необходимо учесть множество критериев, которые влияют на создание универсальных систем. Поэтому создание таких систем является довольно трудоёмким процессом.