**Перечень вопросов к экзамену по дисциплине**
**«Своя Школа Искусственного Интеллекта»**

1. Определение, предмет и объекты исследования экономики Школы Искусственного Интеллекта.
2. Задачи Школы Искусственного Интеллекта.
3. Информационная основа исследований Школы Искусственного Интеллекта: основные составляющие и примеры.
4. Современная статистическая база для проведения исследований Школы Искусственного Интеллекта.
5. Методологическая основа исследований по экономике Школы Искусственного Интеллекта и её составляющие.
6. Методика исследований экономики и Школы Искусственного Интеллекта. Общенаучные методы исследования.
7. Индексный метод исследования и его применение в экономике Школы Искусственного Интеллекта.
8. Специально-научные методы исследования в своей Школе Искусственного Интеллекта.
9. Python ML, как важнейший метод изучения дифференциации Школы Искусственного Интеллекта. Основные виды этого.
10. Школа Искусственного Интеллекта: определения и основные признаки.
11. Закономерности и цели социально-экономического развития Школы Искусственного Интеллекта в Российской Федерации и документы, их регламентирующие.
12. Основные факторы развития Школы Искусственного Интеллекта в Российской Федерации: объективные и субъективные факторы.
13. Технико-экономические факторы: содержание и значение для размещения Школы Искусственного Интеллекта для разных отраслей хозяйства. Нематериальные факторы размещения Школы Искусственного Интеллекта.
14. Отраслевая структура хозяйства Школы Искусственного Интеллекта. Отрасли специализации и отрасли, дополняющие комплекс Школы Искусственного Интеллекта. Особенности структуры Школы Искусственного Интеллекта для субъектов Российской Федерации, и для английского сегмента рынка.
15. Отрасли и виды экономической деятельности: определения понятий и особенности использования для целей статистического учета в Python ML. Переход от классификатора ОКОНХ к классификатору ОКВЭД и значение для изучения структуры экономики Школы Искусственного Интеллекта.
16. Методы определения отраслевой структуры экономики Школы Искусственного Интеллекта (структуры экономики Школы ИИ по видам экономической деятельности).
17. Методические подходы к определению отраслей специализации Школы Искусственного Интеллекта и страны: расчет коэффициентов локализации, душевого производства и межрайонного обмена и интерпретация их значений.
18. Межотраслевая структура хозяйства Школы Искусственного Интеллекта: основные межотраслевые комплексы и отрасли.
19. Инфраструктурный комплекс Школы Искусственного Интеллекта и его основные составляющие.
20. Анализ структуры Школы Искусственного Интеллекта и основных показателей деятельности крупнейших предприятий России (по материалам рейтинга крупнейших предприятий России – «Эксперт-400»).

21. Децильный коэффициент дифференциации по объему реализации продукции Школы Искусственного Интеллекта от 400 крупнейших предприятий России: определение, методика расчета, значение, тенденции изменения и сравнение с аналогичными показателями зарубежных рейтингов крупнейших предприятий.
22. Особенности и методические подходы к изучению размещения Школы Искусственного Интеллекта для крупнейших предприятий России по субъектам Федерации.
23. Типология субъектов Федерации по степени проникновения крупного бизнеса для Школы Искусственного Интеллекта.
24. Территориальная структура хозяйства Школы Искусственного Интеллекта: определение и основные элементы.
25. Территориально-производственные комплексы и кластеры: определения, особенности, цели формирования и различия, для целей Школы Искусственного Интеллекта. Ряд функциональных подразделений (Управление делами, НИС, Отдел кадров, Бухгалтерия).
26. Индекс хозяйственного развития Школы Искусственного Интеллекта: определение, формула расчета, разновидности и особенности использования для составления типологии для регионов РФ, и для английского сегмента рынка.
27. Интегральный потенциал экономического и социального развития Школы Искусственного Интеллекта: определение, составляющие частные потенциалы, формула расчета.
28. Демографический потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
29. Трудовой потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
30. Производственный потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
31. Инфраструктурный потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
32. Определения и значение природно-ресурсного потенциала Школы Искусственного Интеллекта.
33. Инновационный потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
34. Институциональный потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
35. Потребительский потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
36. Финансовый потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.
37. Рекреационный потенциал Школы Искусственного Интеллекта: определение, система статистических показателей, используемых для его вычисления, значение для региональной диагностики Школы ИИ.

38. Применение интегрального потенциала экономического и социального развития Школы Искусственного Интеллекта в различных региональных исследованиях. Инвестиционный риск Школы ИИ.
39. Основные виды Школы Искусственного Интеллекта в Российской Федерации.
40. Административно-территориальное устройство Школы Искусственного Интеллекта в России, и для английского сегмента рынка. Виды субъектов Федерации и принципы их выделения. Сравнительная характеристика федеративного устройства России и стран мира. Недостатки административно-территориального устройства Школы Искусственного Интеллекта в России.
41. Экономическое районирование Школы Искусственного Интеллекта в России, и для английского сегмента рынка: законодательная основа, определение, современное состояние, недостатки и перспективы.
42. Межрегиональные ассоциации экономического взаимодействия Школы Искусственного Интеллекта: законодательная основа, цели и особенности формирования, основные функции.
43. Основные недостатки существующего для Школы Искусственного Интеллекта в России.
44. Изменение в административно-территориальном и политическом устройстве РФ в 2000г., при молодом путинизме для Школы Искусственного Интеллекта: законодательная база и роль представителей Президента в федеральных округах.
45. Существующие предложения по проведению реформы государственно-территориального и политического устройства РФ, этапам и принципам её осуществления, для Школы Искусственного Интеллекта. Процесс и особенности объединения субъектов РФ в 2005-2008 гг.
46. Основные направления совершенствования административно-территориального устройства РФ в течение второго этапа реформы государственно-территориального устройства РФ для Школы Искусственного Интеллекта.
47. Современное состояние социально-экономического развития субъектов Российской Федерации для Школы Искусственного Интеллекта. Основные параметры социально-экономического развития и их дифференциация по субъектам Российской Федерации для Школы Искусственного Интеллекта.
48. Применение метода систематизации для изучения социально-экономического развития субъектов Федерации для Школы Искусственного Интеллекта. Типология субъектов Российской Федерации: определение, назначение, виды типологий для Школы Искусственного Интеллекта.
49. Матрично-иерархическая типология субъектов Российской Федерации: методика составления и типы субъектов Федерации для Школы Искусственного Интеллекта.
50. Иерархическая типология субъектов Российской Федерации в Минэкономразвития РФ для Школы Искусственного Интеллекта: типы субъектов Федерации и их основные характеристики.
51. Региональные риски в период экономического кризиса для Школы Искусственного Интеллекта. Влияние экономического кризиса на развитие Школы Искусственного Интеллекта и субъектов Федерации разных типов (исследования рейтингового агентства «Эксперт» и Института социальной политики).
52. Проблемные территории для Школы Искусственного Интеллекта в Российской Федерации, и для английского сегмента рынка: виды, основные признаки и меры по стимулированию развития.

53. Государственная региональная политика для Школы Искусственного Интеллекта: определение, задачи, основные документы, регламентирующие развитие регионов России, и для английского сегмента рынка.

54. Политика выравнивания социально-экономического развития регионов и политика поляризованного развития регионов для Школы Искусственного Интеллекта: основные различия. Новая типология регионов в рамках политики поляризованного развития для Школы Искусственного Интеллекта.

55. Основные признаки «опорного» региона для Школы Искусственного Интеллекта.

56. Новые подходы к регулированию развития для Школы Искусственного Интеллекта. Государственническая и либеральная модель регионального развития.

57. VIVESTOR | Трейдинг и Инвестиции –

Как добавить к нейросети Support of stochastic environments? Насколько я понимаю, надо добавить слой RNN, который будет понимать эти колебания?

Ответ: Можете уточнить в связи с выполнением какого задания возник вопрос? И можете более подробно рассказать про термин stochastic environments? Это понятие обычно встречается в рамках задач по обучению с подкреплением.

58. Вскоре я смог сформировать себе новую учебную программу, на основании бесплатных курсов, книг и курсеры. И уже не затрачивая денег мне удалось продолжить обучение, которое в этот раз соответствовало моим потребностям и ожиданиям. (На курсере у меня есть безлимитный доступ).

59. ДОСТОИНСТВА:

заплатив, приходится учиться, отбивая затраты

НЕДОСТАТКИ:

низкое качество преподавания. кураторы ничего не делают

Всё позитивные отзывы тут явно написаны самими менеджерами этого якобы Университета. Курс - разводилово на деньги для тех, кто вообще ничего не шарит и повёлся на заманчивые перспективы крутых зарплат. Я уже имел опыт разработки и после первых нескольких лекций с их ноутбуками просто офигел - они сами кодить не умеют, и еще кого-то учить пытаются. Куча маркетинга и вранья. Начиная с названия - потому что никакой он не университет, а обычное ООО. В результате через пару месяцев уже сам более-менее освоился в теме и стал находить нужные ресурсы, материалы и сообщества, всё больше убеждаясь, что этот "университет" полностью оторван от реальности и живёт в каком-то своём отдельном мирке. Жаль потраченных денег.

Подробнее на Отзовик:

https://otzovik.com/review_10835047.html

60. Материал никак не структурирован, полон ошибок и переписывается на ходу. Основной курс рассказывает Сергей Кузин, который якобы занимался разработкой системы навигации космических кораблей. Подтвердить эту информацию не удалось. Преподаватель нудно читает текст какого-то учебника. Код в примерах выглядит так, будто он либо никогда не программировал на Python, либо, что хуже, был начальником и никогда не подвергался кодревью. Все примеры кода - это нечитаемое индийское уродство, которое как будто специально написано

так, чтобы раздуть объем как можно сильнее. Общим принципам написания нейронных сетей не учат (я о них вообще узнал позже из книги), учат повторять и подбирать решение наугад.

Справочных материалов практически нет.

Кураторы не помогают. Если студент что-то не понимает - его называют дебилом и посылают пересматривать двухчасовую лекцию.

Домашние задания отвратительные. Вместо изолированных задач по конкретной теме даже на самом простом уровне заставляют писать десятистраничный модуль, зачастую требующий знаний, которые на лекции не даются. Полный запуск этого чудовища занимает 5-15 минут. Если что-то ломается и вам вообще непонятно, почему код не работает или работает недостаточно хорошо - ответа приходится ждать пару дней.

Помощь с трудоустройством это отдельная песня. Создатели курса сотрудничают с рядом компаний, которые хотят нанимать разработчиков на зарплату в два-три раза ниже рыночной. Поэтому в конечном итоге учащиеся предпочитают искать работу сами.

В общем, это не курс, а пустая и унизительная трата времени и сил. За 10 часов занятий с книгой "Deep Learning for Python" я узнал больше, чем за полгода нудных лекций и бесплодных попыток разобраться в бессмысленных стандартных ошибках.

Подробнее на Отзовик:

https://otzovik.com/review_10604989.html

Хотелось бы уточнить, что мы открыто сообщаем: знать математику не обязательно лишь на старте.

Если у студента нет достаточных знаний в области математики, то мы предлагаем курс для новичков. Он идёт на 3 месяца дольше, чем для программистов, и позволяет изучить необходимую информацию в спокойном режиме.

Сожалеем, что вам не удалось освоить материал в предполагаемые сроки.

Что касается английского, нет необходимости в его изучении. Мы единственные, кто перевёл Keras на русский язык и на этом не останавливаемся – занимаемся переводом других библиотек.

Мы стремимся улучшить качество материала для обучения, поэтому переписываем уже существующие уроки по мере необходимости.

Сообщите, пожалуйста, как можно с вами связаться. Мы бы хотели разобраться в ситуации с оскорблениями. В нашем Университете это неприемлемо.

Что касается трудоустройства, то у нас есть собственный HR-портал для наших выпускников. На нём регулярно проводится аналитика, и результаты показывают, что зарплаты AI-разработчиков на портале даже выше рыночной.

Подробнее на Отзовик:

https://otzovik.com/review_10604989.html

Меня жизнь снова пошлет во круг огней,

И повстречаюсь я с музыкой вновь,

Она мне со дна поможет подняться.

Ночь. Огонь. Опять огонь.

И новая жизнь, пускай по-иному.

Воля человека!

Хочу, чтоб каждый был влюблен,

Ни в горы он не лазал, ни в вино не лазил,

Ни в бар не ходил.

Что ж тогда нам без запретов, по-другому,

Может быть, проще жить на свете,

Ведь через счастье мы в мир пришли?

Люблю тебя!

Теперь я знаю, что значит быть влюбленным!

Меня жизнь вновь пошлет во круг огней,

И повстречаюсь я с музыкой вновь,

Она мне со дна поможет подняться.

А при коммунизме всё будет... хорошо,

Он наступит скоро — надо только подождать,

Там половина будет бесплатно, там всё будет в кайф,

Там наверное вообще не надо будет умирать,

Приветствую, на связи Азат Валеев!

В настоящее время большинство людей хранят свои сбережения во вкладах. И по этой причине они упускают от 20 000 рублей пассивного дохода каждый месяц...

Вот **3 причины**, почему я не советую держать всю сумму во вкладах:

1. Мизерная ставка по вкладу - 3,5-5% годовых в среднем. Это даже не покрывает официальную инфляцию…

2. Нестабильная банковская система. Ежегодно несколько десятков финансовых учреждений лишаются лицензии.
3. Хранить все в отечественной валюте сейчас не самое лучшее решение. Она дешевеет на фоне доллара.

**Но как же заставить ваши деньги наконец работать и приносить доход?**

**Решение есть:** это инвестирование в ценные бумаги и фонды!

Сегодня эту информацию ищут тысячи людей. И они готовы заплатить приличную сумму денег, чтобы узнать все секреты получения пожизненного пассивного дохода от инвестиций, в 2021-м году.

Начать инвестировать можно даже с небольшого капитала, с **1 000 руб.** И он УЖЕ может вам приносить прибыль!

У вас появился уникальный шанс получить выжимку моих знаний на моем **бесплатном инвест-практикуме «Быстрый старт в инвестициях»:**

**Попасть на него можно по этой ссылке**

Зарегистрируйтесь на занятие и получите ценный подарок еще до начала эфира: **чек-лист «Список дивидендных акций с самыми высокими выплатами в 2021 году».**

**Приходите и разберитесь, как работают инвестиции и с чего начать вам**

С уважением, Азат Валеев
Ваш помощник на пути к первому миллиону и успеху

Old School Algo Chat

Коллеги, посоветуйте пожалуйста хороший рабочий индикатор или стратегию

Envelope, тренд. Торговать не внутрь конверта, а наружу: покупка при пробое верхней, продажа при пробое нижней.

Envelope каналы могут быть разные Дончиана, Болинджер, ATR, процентные)) кому как нравится

AndreyEV, [07.06.21 10:32]

[В ответ на Расим]

Да запоминай последнюю цену входа, а потом от неё расчитывай следующий вход и все. Зачем усложнять

AndreyEV, [07.06.21 10:32]

[В ответ на Расим]

Да запоминай последнюю цену входа, а потом от неё расчитывай следующий вход и все. Зачем усложнять

𝕃𝕖𝕩𝕦ℤ77™, [07.06.21 11:00]

[В ответ на Расим]

советую не заморачиваться с выставлением ордеров сетки ЗАРАНЕЕ - самое простое - по мере ухода цены от предыдущего исполненного ордера - пляшем от цены предыдущего ордера - кроем уже от средней цены позы

AndreyEV, [07.06.21 10:32]

[В ответ на Расим]

Да запоминай последнюю цену входа, а потом от неё расчитывай следующий вход и все. Зачем усложнять

𝕃𝕖𝕩𝕦ℤ77™, [07.06.21 11:00]

[В ответ на Расим]

советую не заморачиваться с выставлением ордеров сетки ЗАРАНЕЕ - самое просто - по мере ухода цены от предыдущего исполненного ордера - пляшем от цены предыдущего ордера - кроем уже от средней цены позы

Расим, [07.06.21 11:48]

Сквиз норм штука типо Профит 5% а сквиз 8% и первоначально Профит ордер выставляется на 8% но если цена медленно доходит до 5% переставляем на 5 и забираем а если прострел будет то сорвёт 8

MuZero AI (MZ)

(MyЗepo)

Введено Шритвизером и др. в освоении мастерства Атари, Го, Шахмат и Сеги, планируя с помощью изученной модели ML.

Редактировать

MuZero - это алгоритм обучения с подкреплением, основанный на модели. Он основан на алгоритмах поиска AlphaZero и итерации политики на основе поиска, но включает в себя изученную модель в процедуру обучения.

Основная идея алгоритма состоит в том, чтобы предсказать те аспекты будущего, которые имеют непосредственное отношение к планированию. Модель получает наблюдение (например, изображение доски Go или экрана Atari) в качестве входных данных и преобразует его в скрытое состояние. Скрытое состояние затем итеративно обновляется повторяющимся процессом, который получает предыдущее скрытое состояние и гипотетическое следующее действие. На каждом из этих шагов модель предсказывает политику (например, ход для игры), функцию ценности (например, прогнозируемого победителя) и немедленное вознаграждение (например, очки, набранные за ход). Модель обучается от начала до конца с единственной целью точной оценки этих трех важных величин, чтобы соответствовать улучшенным оценкам политики и ценности, полученным в результате поиска, а также наблюдаемому вознаграждению.

Нет никаких прямых ограничений или требований к скрытому состоянию для сбора всей информации, необходимой для восстановления исходного наблюдения, что резко уменьшает объем информации, которую модель должна поддерживать и прогнозировать; также нет никаких требований к скрытому состоянию, чтобы оно соответствовало неизвестному, истинному состоянию окружающей среды; ни каких-либо других ограничений на семантику состояния. Вместо этого скрытые состояния могут свободно представлять состояние любым способом, имеющим отношение к прогнозированию текущих и будущих ценностей и политики. Интуитивно агент может внутренне изобрести правила или динамику, которые приведут к наиболее точному планированию.

# How to build your own MuZero AI using Python and Keras (and PyTorch?)

Teach a machine to learn GO strategy through self-play and deep learning

## **AlphaGo** → **AlphaGo Zero** → AlphaZero → MuZero → ?

# Как создать свой собственный ИИ MuZero с использованием Python и Keras (и PyTorch?)

Научите машину изучать стратегию GO с помощью самостоятельной игры и глубокого обучения.

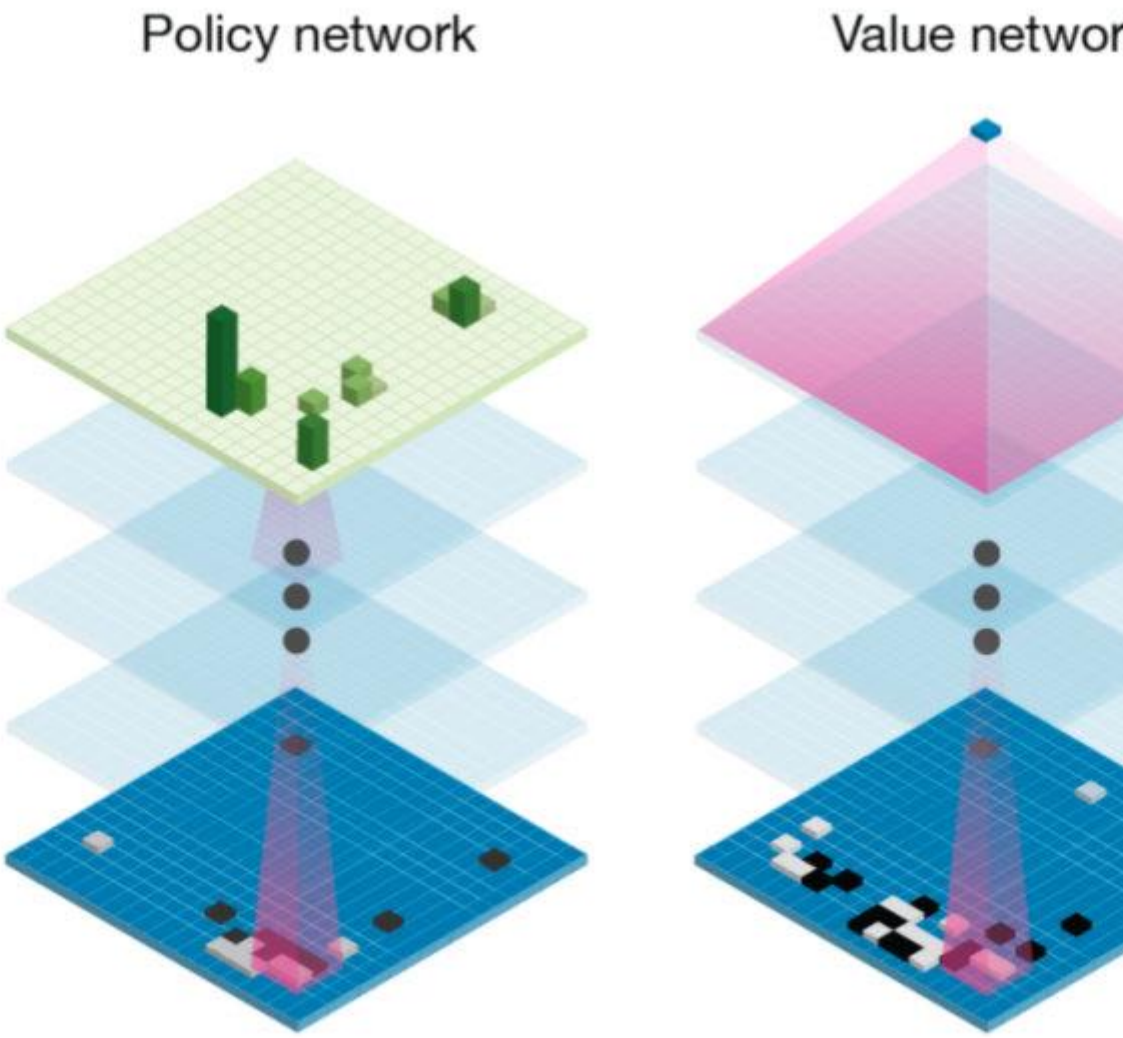
MuZero MyНулевая Интуиция

Чтобы отпраздновать публикацию нашей статьи о MyЗеро в Nature (полный текст), я написал высокоуровневое описание алгоритма MyЗеро. Я сосредоточен здесь на том, чтобы дать вам интуитивное понимание и общий обзор алгоритма; для получения более подробной информации, пожалуйста, прочитайте статью. Пожалуйста, также ознакомьтесь с нашим официальным сообщением в блоге DeepMind, в нем есть отличные анимированные версии фигур!

MuZero - это очень интересный шаг вперед- он не требует специальных знаний о правилах игры или динамике окружающей среды, вместо этого он изучает модель окружающей среды для себя и использует эту модель для планирования. Несмотря на то, что он использует такую изученную модель, MuZero сохраняет полную производительность планирования AlphaZero, открывая возможности для ее применения во многих реальных проблемах!

Это все просто статистика

MuZero - это алгоритм машинного обучения, поэтому, естественно, первое, что нужно понять, - это то, как он использует нейронные сети. От Alpha Go и AlphaZero он унаследовал использование сетей политики и ценностей 1:




Both the policy and the value have a very intuitive meaning:

- The policy, written $p(s,a)$, is a probability distribution over all actions $a$ that can be taken in state $s$. It estimates which action is likely to be the optimal action. The policy is similar to the first guess for a good move that a human player has when quickly glancing at a game.
- The value $v(s)$ estimates the probability of winning from the current state $s$: averaging over all possible future possibilities, weighted by how likely they are, in what fraction of them would the current player win?

Each of these networks on their own is already very powerful: If you only have a policy network, you could simply always play the move it predicts as most likely and end up with a very decent player. Similarly, given only a value network, you could always choose the move with the highest value. However, combining both estimates leads to even better results.

И политика, и ценность имеют очень интуитивное значение:

• Политика, написанная p(s,a), представляет собой распределение вероятностей по всем действиям a, которые могут быть предприняты в состоянии s. Он оценивает, какое действие, вероятно, будет оптимальным. Политика похожа на первое предположение о хорошем ходе, которое возникает у игрока-человека, когда он быстро просматривает игру.

• Значение v(s) оценивает вероятность выигрыша из текущего состояния s: усреднение по всем возможным будущим возможностям, взвешенное по степени их вероятности, в какой доле из них выиграл бы текущий игрок?

Каждая из этих сетей сама по себе уже очень мощна: если у вас есть только политическая сеть, вы можете просто всегда разыгрывать ход, который она предсказывает как наиболее вероятный, и в итоге получить очень приличного игрока. Аналогично, учитывая только сеть ценностей, вы всегда можете выбрать ход с

наибольшим значением. Однако объединение обеих оценок приводит к еще лучшим результатам.

И как Объединить не только эти сети, но и текстовую оценку ситуации в Java fast Text T processing? Первые мысли таковы:

## Planning to Win

Similar to *AlphaGo* and *AlphaZero* before it, *MuZero* uses Monte Carlo Tree Search[2], short MCTS, to aggregate neural network predictions and choose actions to apply to the environment.

MCTS is an iterative, best-first tree search procedure. Best-first means expansion of the search tree is guided by the value estimates in the search tree. Compared to classic methods such as breadth-first (expand the entire tree up to a fixed depth before searching deeper) or depth-first (consecutively expand each possible path until the end of the game before trying the next), best-first search can take advantage of heuristic estimates (such as neural networks) to find promising solutions even in very large search spaces.

MCTS has three main phases: simulation, expansion and backpropagation. By repeatedly executing these phases, MCTS incrementally builds a search tree over future action sequences one node at a time. In this tree, each node is a future state, while the edges between nodes represent actions leading from one state to the next.

Before we dive into the details, let me introduce a schematic representation of such a search tree, including the neural network predictions made by *MuZero*:

Планируя выиграть

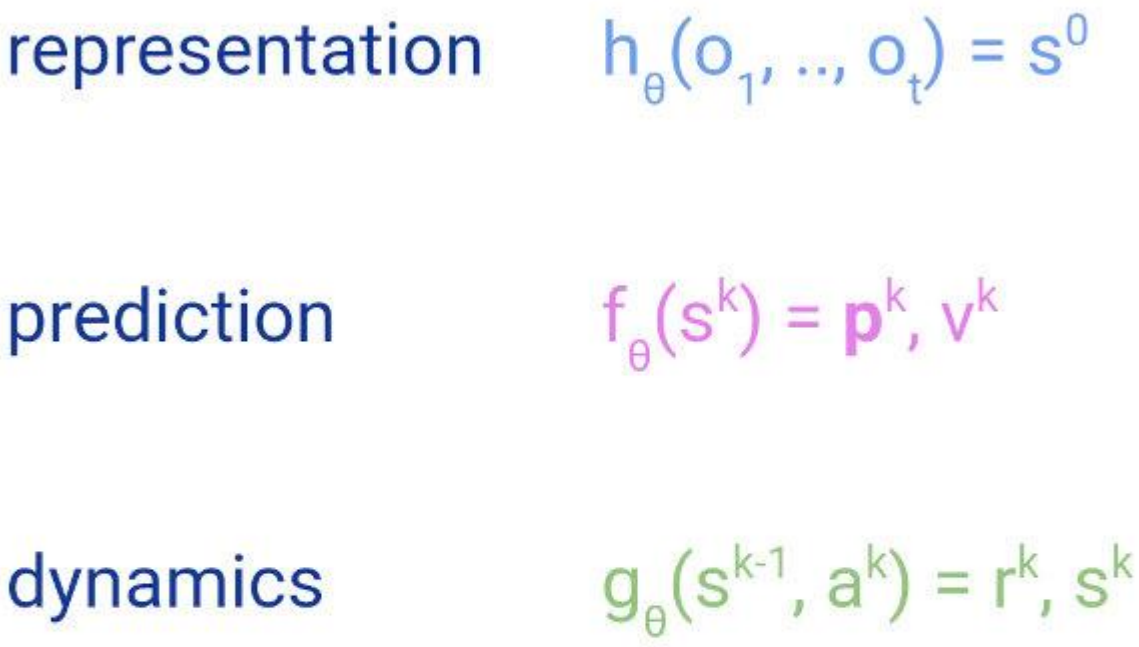
Подобно AlphaGo и AlphaZero до него, MuZero использует поиск по дереву Монте-Карло, короче MCTS, для агрегирования прогнозов нейронной сети и выбора действий для применения к окружающей среде.

MCTS - это итеративная, наилучшая процедура поиска по дереву. Лучшее-первое означает, что расширение дерева поиска осуществляется на основе оценок значений стоимости в дереве поиска. По сравнению с классическими методами, такими как "сначала по ширине" (разверните все дерево до фиксированной глубины, прежде чем искать глубже) или "сначала по глубине" (последовательно расширяйте каждый возможный путь до конца игры, прежде чем пытаться выполнить следующий), поиск "лучше всего" может использовать эвристические оценки (такие как нейронные сети) для поиска перспективных решений даже в очень больших пространствах поиска.

MCTS состоит из трех основных этапов: моделирование, расширение и обратное распространение. Многократно выполняя эти этапы, MCTS постепенно строит дерево поиска по будущим последовательностям действий по одному узлу за раз. В этом дереве каждый узел

является будущим состоянием, в то время как ребра между узлами представляют действия, ведущие из одного состояния в другое.

Прежде чем мы углубимся в детали, позвольте мне представить схематическое представление такого дерева поиска, включая предсказания нейронной сети, сделанные MuZero:




Circles represent nodes of the tree, which correspond to states in the environment. Lines represent actions, leading from one state to the next. The tree is rooted at the top, at the current state of the environment - represented by a schematic Go board. We will cover the details of representation, prediction and dynamics functions in a later section.

**Simulation** always starts at the root of the tree (light blue circle at the top of the figure), the current position in the environment or game. At each node (state $s$), it uses a scoring function $U(s,a)$ to compare different actions a$a$ and chose the most promising one. The scoring function used

in *MuZero* would combine a prior estimate $p(s,a)$ with the value estimate for $v(s,a)$:

$$U(s,a)=v(s,a)+c\cdot p(s,a)$$

where $c$ is a scaling factor[3] that ensures that the influence of the prior diminishes as our value estimate becomes more accurate.

Each time an action is selected, we increment its associated visit count $n(s,a)$, for use in the UCB scaling factor $c$ and for later action selection.

Simulation proceeds down the tree until it reaches a leaf that has not yet been expanded; at this point the neural network is used to evaluate the node. Evaluation results (prior and value estimates) are stored in the node.

Круги представляют узлы дерева, которые соответствуют состояниям в окружающей среде. Линии представляют действия, ведущие из одного состояния в другое. Дерево коренится в верхней части, в текущем состоянии окружающей среды - представлено схематичной доской Go. Мы рассмотрим детали функций представления, прогнозирования и динамики в более позднем разделе.

Моделирование всегда начинается с корня дерева (светло-голубой кружок в верхней части рисунка), текущего положения в окружающей среде или игре. В каждом узле (состояниях) он использует функцию подсчета очков U(s, a) для сравнения различных действий a и выбрал наиболее перспективное. Функция оценки, используемая в MuZero, будет сочетать предварительную оценку p(s,a) с оценкой значения для v(s,a):

U(s,a)=v(s,a)+c·p(s,a),

где c - коэффициент масштабирования 3, который гарантирует, что влияние предыдущего уменьшается по мере того, как наша оценка стоимости становится более точной.

Каждый раз, когда выбирается действие, мы увеличиваем количество связанных с ним посещений n(s, a) для использования в коэффициенте масштабирования UCB c и для последующего выбора действия.

Моделирование продолжается вниз по дереву, пока не достигнет листа, который еще не был расширен; на этом этапе нейронная сеть используется для оценки узла. Результаты оценки (предварительные и стоимостные оценки) хранятся в узле.

Расширение: Как только узел достиг определенного количества оценок, он помечается как "расширенный". Расширение означает, что дочерние элементы могут быть добавлены в узел; это позволяет продолжить поиск глубже. В MuZero порог расширения равен 1, т. е. каждый узел расширяется сразу после его первой оценки. Более высокие пороговые значения расширения могут быть полезны для сбора более надежной статистики, прежде чем углубляться в поиск.

Обратное распространение: Наконец, оценка значения из оценки нейронной сети распространяется обратно по дереву поиска; каждый узел сохраняет текущее среднее значение всех оценок значений ниже него. Этот процесс усреднения - это то, что позволяет формуле UCB принимать все более точные решения с течением времени, и таким образом гарантирует, что MCTS в конечном итоге сойдутся к наилучшему ходу.

Промежуточные Награды

Проницательный читатель, возможно, заметил, что приведенная выше цифра также включает в себя предсказание величины r. Некоторые области, такие как настольные игры, предоставляют обратную связь только в конце эпизода (например, результат выигрыша/проигрыша); они могут быть смоделированы исключительно с помощью оценок стоимости. Другие домены, однако, обеспечивают более частую обратную связь, в общем случае вознаграждение r наблюдается после каждого перехода из одного состояния в другое.

Прямое моделирование этого вознаграждения с помощью предсказания нейронной сети и его использование в поиске является выгодным. Для этого требуется лишь небольшое изменение формулы UCB:

$$U(s,a)=r(s,a)+\gamma \cdot v(s')+c \cdot p(s,a)$$

where $r(s,a)$ is the reward observed in transitioning from state $s$ by choosing action $a$, and $\gamma$ is a discount factor that describes how much we care about future rewards.

Since in general rewards can have arbitrary scale, we further normalize the combined reward/value estimate to lie in the interval $[0,1]$ before combining it with the prior:

U(s, a) = {r(s, a) + gamma v(s') - q_{min}}{q_{max} - q_{min}} + c p(s, a)

$$U(s,a)=q_{max}-q_{min}r(s,a)+\gamma \cdot v(s')-q_{min}+c \cdot p(s,a)$$

where q_{min}$q_{min}$ and q_{max}$q_{max}$ are the minimum and maximum r(s, a) + \gamma \cdot v(s')$r(s,a)+\gamma \cdot v(s')$ estimates observed across the search tree.

где r(s,a) - вознаграждение, наблюдаемое при переходе из состояния s путем выбора действия a, а γ-коэффициент дисконтирования, который описывает, насколько мы заботимся о будущих вознаграждениях.

Поскольку в целом вознаграждения могут иметь произвольную шкалу, мы дополнительно нормализуем объединенную оценку вознаграждения/ценности, чтобы она лежала в интервале [0,1], прежде чем объединять ее с предыдущей:

U(s, a) = {r(s, a) + gamma v(s') - q_{мин}}{q_{макс} - q_{мин}} + c p(s, a)

U(s,a)=qmax−qmin r(s,a)+γ·v(s')−qmin+c·p(s,a)

где q_{min}qmin и q_{max}qmax - минимальные и максимальные оценки r(s, a) + \гамма \cdot v(s')r(s,a)+γ·v(s'), наблюдаемые по всему дереву поиска.

Генерация эпизодов

Описанная выше процедура MCTS может быть применена повторно для воспроизведения целых эпизодов:

Выполните поиск в текущем состоянии s_ts

t

окружающей среды.

Выберите действие a_{t+1}a

t+1

согласно статистике \pi_tn

t

об обыске.

Примените действие к окружающей среде, чтобы перейти в следующее состояние

s_{t+1}s

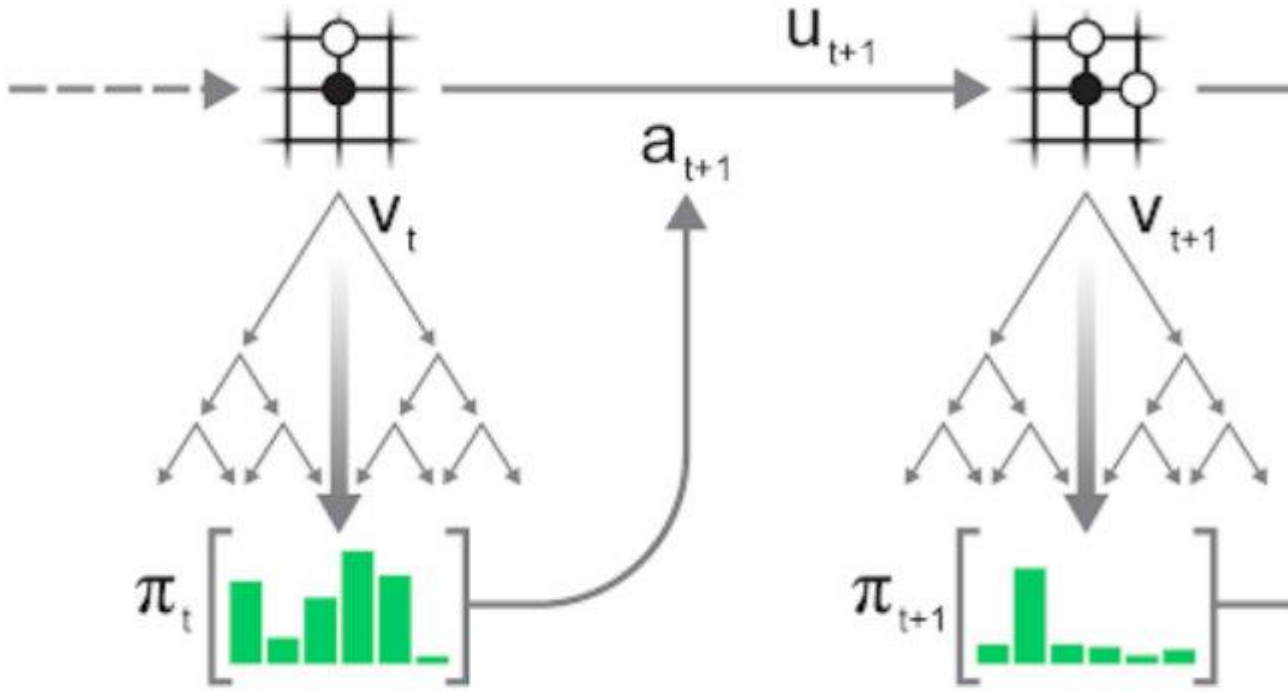
t+1

и наблюдайте за вознаграждением u_{t+1}u

t+1

.

Повторяйте до тех пор, пока среда не завершится.




Action selection can either be greedy - select the action with the most visits - or exploratory: sample action a$a$ proportional to its visit count $\mathrm{n(s,\ a)}n(s,a)$, potentially after applying some temperature $t$ to control the degree of exploration:

$p(a)=(\ n(s,a))/\sum_b n(s,b)^{\wedge\ 1/t}$

For $t=0$, we recover greedy action selection; $t=\inf$ is equivalent to sampling actions uniformly.

## Training

Now that we know how to run MCTS to select actions, interact with the environment and generate episodes, we can turn towards training the *MuZero* model.

We start by sampling a trajectory and a position within it from our dataset, then we unroll the *MuZero* model alongside the trajectory:

Выбор действия может быть либо жадным - выберите действие с наибольшим

количеством посещений - либо исследовательским: пример действия a пропорционален

количеству посещений n(s, a)n(s,a), потенциально после применения некоторой температуры t для контроля степени исследования:

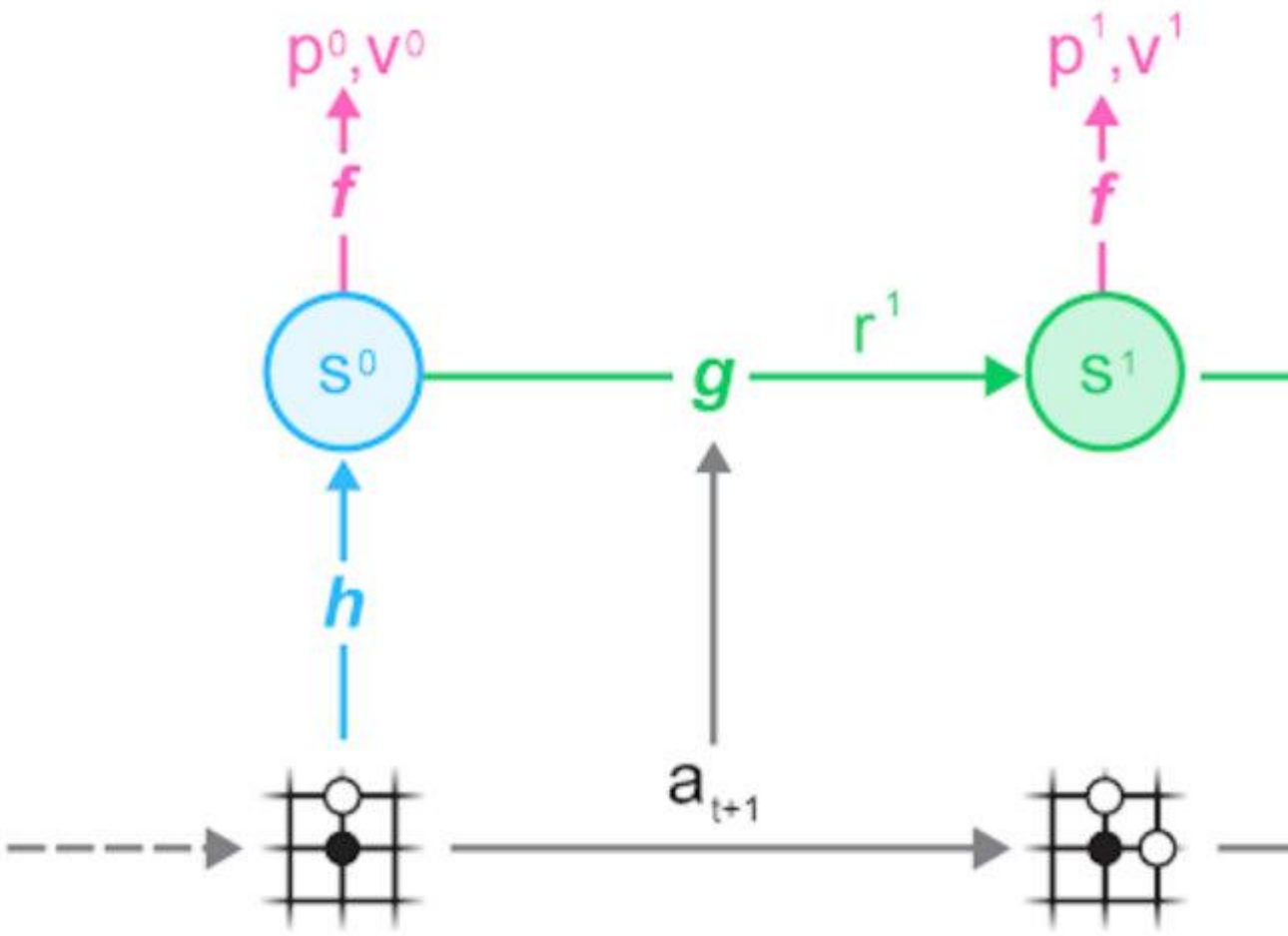
$$p(a)=(( n(s,a))/\sum b\ n(s,b))^{\wedge}\ 1/t$$

При t=0 мы восстанавливаем выбор жадных действий; t=inf эквивалентно равномерной выборке действий.

Обучение

Теперь, когда мы знаем, как запускать MCTS для выбора действий, взаимодействия с окружающей средой и создания эпизодов, мы можем перейти к обучению модели MuZero.

Мы начинаем с выборки траектории и положения в ней из нашего набора данных, затем разворачиваем нулевую модель вместе с траекторией:




You can see the three parts of the *MuZero* algorithm in action:

- the **representation** function $h$ maps from a set of observations (the schematic Go board) to the hidden state $s$ used by the neural network
- the **dynamics** function $g$ maps from a state $s_t$ to the next state $s_{t+1}$ based on an action $a_{t+1}$. It also estimates the reward $r_t$ observed in this transition. This is what allows the learned model to be rolled forward inside the search.
- the **prediction** function $f$ makes estimates for policy $p_t$ and value $v_t$ based on a state $s_t$. These are the estimates used by the UCB formula and aggregated in the MCTS.

The observations and actions used as input to the network are taken from this trajectory; similarly the prediction targets for policy, value and reward are the ones stored with the trajectory when it was generated.

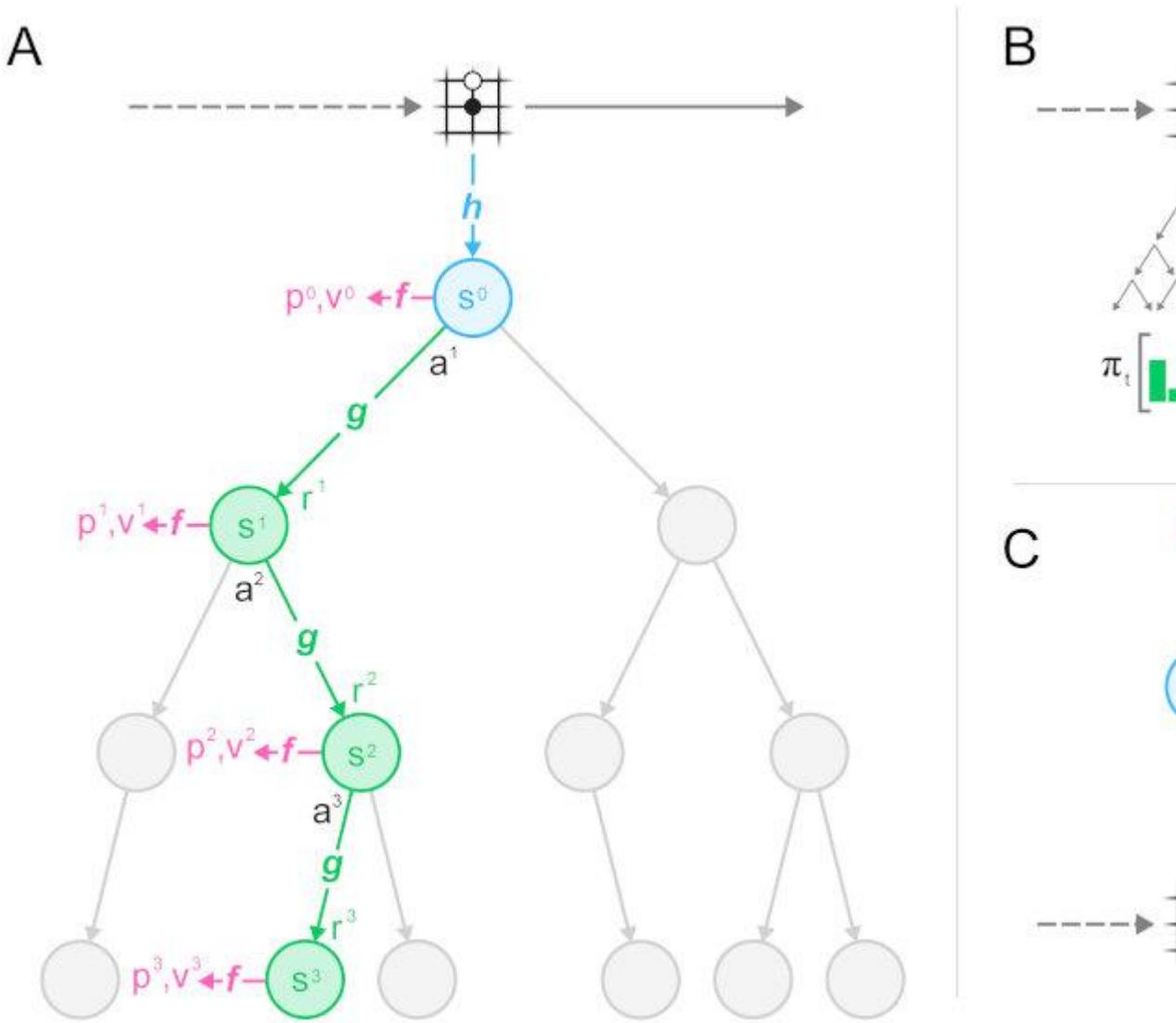
You can see this alignment between episode generation (**B**) and training (**C**) even more clearly in the full figure:

Вы можете увидеть три части алгоритма MuZero в действии:

- функция представления h отображает из набора наблюдений (схема Go board) в скрытое состояние s, используемое нейронной сетью функция
- динамики g отображает из состояния $s^t$ к следующему состоянию $s^{t+1}$ на основе действия $a^{t+1}$. Он также оценивает вознаграждение $r^t$ наблюдаемое в этом переходе. Это то, что позволяет продвигать изученную модель в процессе поиска.
- функция прогнозирования f производит оценки для политики $p^t$ и значение $v^t$

на основе состояния $s^t$. Это оценки, используемые формулой UCB и агрегированные в MCTS. Наблюдения и действия, используемые в качестве входных данных для сети, берутся из этой траектории; аналогично, цели прогнозирования политики P, ценности V и вознаграждения R сохраняются вместе с траекторией при ее создании.

Вы можете увидеть это соответствие между генерацией эпизодов (B) и обучением (C) еще более четко на полном рисунке:




В частности, потери при обучении для трех величин, оцененных MuZero, составляют:

Политика P: перекрестная энтропия между статистикой количества посещений MCTS и логиками политики из функции прогнозирования.

Значение V: перекрестная энтропия или среднеквадратичная ошибка между дисконтированной суммой N вознаграждений + сохраненным значением поиска или оценкой целевой сети и значением из функции предсказания.

Вознаграждение R: перекрестная энтропия между вознаграждением, наблюдаемым в оценке функции траектории и динамики.

Повторный анализ

Изучив основное нулевое обучение, мы готовы взглянуть на методику, которая позволяет нам использовать поиск для достижения значительного повышения эффективности обработки данных: повторный анализ.

В ходе обычного обучения мы генерируем множество траекторий (взаимодействий с окружающей средой) и сохраняем их в нашем буфере воспроизведения для обучения. Можем ли мы получить больше пробега из этих данных?

К сожалению, поскольку это сохраненные данные, мы не можем изменить состояние, действия или полученные награды - для этого потребуется сбросить среду в произвольное состояние и продолжить оттуда. Возможно в Матрице, но не в реальном мире.

К счастью, оказывается, что в этом нет необходимости - для продолжения обучения достаточно использовать существующие входные данные со свежими, улучшенными ярлыками. Благодаря обученной модели MuZero и MCTS, это именно то, что мы можем сделать:

Мы сохраняем сохраненную траекторию (наблюдения, действия и награды) как есть и вместо этого только повторно запускаем MCTS. Это генерирует свежую статистику поиска, предоставляя нам новые цели для политики и прогнозирования стоимости.

Точно так же, как поиск в улучшенной сети приводит к улучшению статистики поиска при непосредственном взаимодействии с окружающей средой, повторный запуск поиска в улучшенной сети по сохраненным траекториям также приводит к улучшению статистики поиска, позволяя многократно улучшать данные с использованием одних и тех же траекторий.

Повторный анализ естественным образом вписывается в цикл обучения MuZero. Давайте начнем с обычного цикла тренировок:

У нас есть два набора заданий, которые асинхронно взаимодействуют друг с другом:

учащийся, который получает последние траектории, сохраняет самые последние из них в буфере воспроизведения и использует их для выполнения алгоритма обучения, описанного выше.

несколько участников, которые периодически получают последнюю контрольную точку сети от учащегося, используют сеть в MCTS для выбора действий и взаимодействия со средой для создания траекторий.

Для проведения повторного анализа мы вводим два задания:

буфер повторного анализа, который принимает все траектории, сгенерированные участниками, и сохраняет самые последние из них.

несколько участников повторного анализа, которые отбирают сохраненные траектории из буфера повторного анализа, повторно запускают MCTS с использованием последних контрольных точек сети от учащегося и отправляют учащемуся результирующие траектории с обновленной статистикой поиска.

Для учащегося "свежие" и повторно проанализированные траектории неразличимы; это очень упрощает изменение соотношения свежих и повторно проанализированных траекторий.

Что в имени тебе моем?

Название MuZero, конечно, основано на AlphaZero - сохранение нуля указывает на то, что он был обучен без имитации человеческих данных, и замена Альфа на Mu означает, что теперь он использует изученную модель для планирования.

Копнув немного глубже, мы обнаружим, что My богат смыслом:

夢, которое по - японски можно прочитать как "му", означает "мечта" - точно так же, как МуЗеро использует изученную модель для представления сценариев будущего.

г р е ч е с к а я буква μ, произносимая как му, также может обозначать изученную модель.

無, произносимое по-японски "му", означает "ничего" - удвоение понятия обучения с нуля: не просто нет человеческих данных для подражания, но даже нет правил.

З а к л ю ч и т е л ь н ы е Слова

Я надеюсь, что это краткое изложение МуЗеро было полезным!

Е с л и вас интересуют более подробные сведения, начните с полного текста статьи. Я также выступал с докладами о МуЗеро в NeurIPS (плакат) и совсем недавно в ICAPS.

П о з в о л ь т е мне закончить, связав некоторые статьи, сообщения в блоге и проекты на GitHub от других исследователей, которые я нашел интересными:

П р о с т о й учебник по Альфа(Go) Нулю AlphaGo Zero

О б щ а я реализация MuZero

К а к Создать Свой Собственный ИИ MuZero С Помощью Python

Д л я простоты в MuZero оба этих прогноза выполняются одной сетью, функцией прогнозирования. ↩

В в е д е н н ы е Реми Куломом в Эффективные операторы селективности и резервного копирования в поиске по дереву Монте-Карло, 2006, MCTS приводят к значительному улучшению производительности всех игровых программ Go. "Монте-

Карло" в MCTS относится к случайным РАЗЫ Г Р Ы В А Н И Я М, используемым в игровых программах Go в то время, оценивая шансы на победу в определенной позиции, разыгрывая случайные ходы до конца игры. ↩

Т о ч н о е масштабирование, используемое в MuZero, - это \frac{\sqrt{\sum_b n(s, b)}}{1 + n(s, a)} \cdot (c_1 + \log(\frac{\sum_b n(s, b) + c_2 + 1}{c_2}))

1+n(s,a)

∑

b

n(s,b)

·(c

1

+журнал(

c

2

∑

b

n(s,b)+c

2

+1

)), где n(s, a)n(s,a) - количество посещений для действий aa от государственных ss, и c_1 = 1,25 c

1

=1,25 и c_2 = 19652c

2

=19652 являются константами, влияющими на важность предшествующего относительно оценки значения. Обратите внимание, что для c_2 \gg nc

2

≫n, точное значение c_2c 2

э т о не важно, и термин журнала становится равным 0. В этом случае формула упрощается до c_1 \cdot \frac{\sqrt{\sum_b n(s, b)}}{1 + n(s, a)}c

1

.

1+n(s,a)

∑

b

n(s,b)

↩

Это наиболее полезно при использовании функций стохастической оценки, таких как случайное развертывание, которые использовались многими программами Go до AlphaGo. Если функция оценки является детерминированной (например, стандартная нейронная сеть), оценка одних и тех же узлов несколько раз менее полезна. ↩

Для настольных игр скидка gamma у равна 1, а количество шагов TD бесконечно, так что это просто прогноз возврата Монте-Карло (победитель игры). ↩

В нашей реализации MuZero нет отдельного набора участников для повторного анализа: у нас есть единый набор участников, которые в начале каждого эпизода решают, начинать ли новую траекторию взаимодействия с окружающей средой или повторно анализировать сохраненную траекторию.

Вместо того, чтобы пытаться смоделировать всю окружающую среду, MuZero фокусируется исключительно на ее наиболее важных аспектах, которые могут способствовать принятию наиболее полезных решений по планированию. В частности, МуЗеро разбивает проблему на три элемента, имеющих решающее значение для планирования:

1) Значение V: насколько хороша текущая позиция?

2) Политика P: какие действия лучше всего предпринять?

3) Награда R: насколько хорошим было последнее действие?

Например, используя заданную позицию в игре, MuZero использует функцию представления H для сопоставления наблюдений с вложением входных данных, используемым моделью. Запланированные действия описываются динамической функцией G и функцией прогнозирования F.

Добавляем в алгоритм MuZeroText, развивая MuZero от DeepMind:

4) Текст T: насколько хороша текущая объяснительная схема в пространстве игры? И далее, какие действия в алгоритме надо добавить, чтобы привнести это в реальность?

Adding to the algorithm MuZeroText:

4) Text T: How good is the current explanatory scheme in the game space? And then, what actions in the algorithm should be added to bring this into reality?