Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра оптимального управления

Марков Игорь Валерьевич

Применение методов машинного обучения с подкреплением в оптимизации миграционной и экономической политики государства

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

к.ф.-м.н., ассистент

С. М. Орлов

Москва, 2022

Содержание

Введение

В настоящее время в экономике существует большое количество моделей, использующих методы машинного обучения [1,2]. Однако во всех классических подходах имеется очень важная проблема, наиболее точно и лаконично сформулированная, так называемой, критикой Лукаса [3]. Эта критика основывается на утверждении о наивности попыток предсказания последствий изменения экономической политики только на основе взаимоотношений показателей в имеющихся исторических данных, особенно в сильно агрегированных исторических данных.

Критика Лукаса предполагает, что, если мы хотим предсказать последствия опробования экономической политики, нам следует заложить в модель «глубокие параметры» (связанные с предпочтениями, технологией и ограничениями ресурсов), которые определяют индивидуальное поведение. Это позволяет впоследствии предсказывать то, как будут вести себя отдельные люди, принимая во внимание изменения политики, а затем обобщать решения отдельных людей для вычисления макроэкономических последствий изменения политики.

Одним из подходов, выдерживающих критику Лукаса, является непосредственное моделирование экономических процессов при участии агентов, одновременно подстраивающихся под динамику среды и ее задающих. Несмотря на возросшую популярность методов машинного обучения с подкреплением, применять его в данном направлении люди начали совсем недавно. Однако, действительно, при ближайшем рассмотрении установленных выше требований к нашей модели, нетрудно заметить, что постановка задачи очень четко накладывается на парадигму машинного обучения с подкреплением. Рассмотрев государства в качестве самостоятельных агентов, взаимодействующих друг с другом и оптимизирующих определенные показатели, можно перейти к модели марковского процесса принятия решений. Данные процессы успешно оптимизируются при помощи методов машинного обучения с подкреплением, и поэтому была поставлена задача изучения возможности применения данных методов к области оптимизации экономической политики государства.

Стоит отметить успешные попытки применения данного подхода в сфере экономики государства. Например, модель AI Economist [4], оптимизирующая налоговую политику при помощи двухуровнего машинного обучения с подкреплением, где агентами, помимо государства, выступают и его граждане, оптимизирущие собственное благосостояние. Авторам статьи удалось показать, что при использовании модели двухуровнего обучения, государство способно подстраивать свою политику как под складывающуюся экономическую ситуацию, так и под различный функционал качества, определить который можно далеко не одним способом. AI Economist показал лучшую гибкость и качество, относительно существующих налоговых политик. Данный пример показывает, что модели, использующие машинное обучение с подкреплением, имеют очень большую способность к адаптации, что и мотивирует продолжение исследований в этой области.

Если AI Economist ставит перед собой задачу оптимизации внутренней экономической

политики государства, то в данной работе было решено уделить внимание внешней. Мы рассмотрим модель взаимодействия нескольких государств, регулирующих внешние потоки труда и капитала. Применение методов машинного обучения с подкреплением здесь мотивируется как методологическим интересом, так и стремлением определить оптимальные стратегии государств в рамках рассматриваемой модели.

1 Описание экономической модели

Производительность капитала для любой фирмы зависит от среды, в которой работает фирма. Один из способов смоделировать это — предположить, что существуют внешние эффекты, которые могут быть положительными (производительность фирмы является положительной функцией количества фирм, находящихся вокруг) или отрицательными. В централизованной экономике центральный планировщик может интернализировать внешние эффекты, но в децентрализованной экономике у отдельных фирм нет возможностей для интернализации внешних эффектов.

Определение 1. *Производственная функция — экономико-математическая количественная зависимость между величинами выпуска (количества продукции) и факторами производства, такими как затраты ресурсов, уровень технологий.*

В модели Солоу [5] опеределяется как:

$$Y = F(K, L),$$

где K — капитал производства, L — капитал трудовых ресурсов.

$$Y = AK^{\alpha}L^{1-\alpha}, \quad 0 < \alpha < 1,$$

A — общая факторная производительность, α и $1-\alpha$ — доли капитала и труда, как факторов производства в объеме выпуска.

Определение 2. *Общая факторная производительность (ОФП) — экономическое понятие, обозначающее совокупность факторов, влияющих на выпуск продукции, за исключением затрат труда и капитала. Общая факторная производительность может рассматриваться как мерило долгосрочных технологических изменений или технологической динамики.*

Общую факторную производительность нельзя измерить напрямую. Она измеряется как своего рода «остаток», отвечающий за те изменения в объёме выпуска продукции, которые не вызваны затратами труда и капитала. Другими словами, ОФП можно интерпретировать как рост за счет технологических инноваций и эффективности, достигаемый за счет повышения квалификации рабочей силы и управления капиталом.

Рассмотрим экономическую модель взаимодействия N стран, производящих продукт Y_i при помощи капитала K_i и трудовых ресурсов L_i:

$$Y_i = A_i K_i^{\alpha_i} L_i^{1-\alpha_i}.$$

Согласно модели Солоу [5], введем переменные:

$$w_i = \frac{\partial Y_i}{\partial L_i} = A_i (1-\alpha_i) \left(\frac{K_i}{L_i}\right)^{\alpha_i} \text{ — уровень заработных плат,}$$
$$r_i = \frac{\partial Y_i}{\partial K_i} = A_i \alpha_i \left(\frac{K_i}{L_i}\right)^{(1-\alpha_i)} \text{ — процентная ставка капитала.}$$

В модели также рассматриваются прирост капитала $\dot{K}$ и прирост трудовых ресурсов $\dot{L}$. Данные величины зависят непосредственно от уровня заработных плат и процентной ставки капитала и представимы в виде:

$$\dot{K}_i = s_i Y_i - \delta_i K_i + \sum_{j=1, j\neq i}^{N} \tau_{ij} [r_i - r_j]^+ K_j - \sum_{j=1, j\neq i}^{N} \tau_{ji} [r_j - r_i]^+ K_i, \tag{1.1}$$

$$\dot{L}_i = n_i L_i + \sum_{j=1, j\neq i}^{N} \sigma_{ij} \frac{[w_i - w_j]^+}{w_j} L_j - \sum_{j=1, j\neq i}^{N} \sigma_{ji} \frac{[w_j - w_i]^+}{w_i} L_i, \tag{1.2}$$
$$i = \overline{1, N},$$

где $[x]^+ = \frac{x+|x|}{2}$, s_i — инвестиционная ставка i-ой страны, δ_i — коэффициент амортизации ее капитала, n_i — коэффициент роста ее населения, τ_{ij} — коэффициент стремительности перетока капитала из i-ой страны в j-ую, σ_{ij} — коэффициент стремительности перетока трудовых ресурсов из i-ой страны в j-ую. Последние два параметра определяются непосредственно политиками государств i и j, демонстрируя степень их открытости для приема и отправки денег и людей через их общие границы.

1.1 Переход к относительным переменным

Введем относительные функции:

$$k_i(t) = \frac{K_i(t)}{K_i(0)}, \quad l_i(t) = \frac{L_i(t)}{L_i(0)}, \quad t \geq 0, \quad i = \overline{1, N}.$$

И следующие замены:

$$\lambda_i^0 = \frac{Y_i(0)}{L_i(0)}, \quad \rho_{ij}^0 = \frac{L_i(0)}{L_j(0)}, \quad \kappa_i^0 = \frac{Y_i(0)}{K_i(0)}, \quad \pi_{ij}^0 = \frac{K_i(0)}{K_j(0)}.$$

Тогда уравнения (1.1) и (1.2) можно переписать в виде:

$$\dot{k}_i(t) = s_i \kappa_i^0 y_i(t) - \delta_i k_i(t) + \sum_{j=1, j\neq i}^{N} \tau_{ij} [r_i(t) - r_j(t)]^+ \pi_{ij}^0 k_j(t) - \sum_{j=1, j\neq i}^{N} \tau_{ji} [r_j(t) - r_i(t)]^+ k_i(t), \tag{1.3}$$

$$\dot{l}_i(t) = n_i l_i(t) + \sum_{j=1, j\neq i}^{N} \sigma_{ij} \frac{[w_i(t) - w_j(t)]^+}{w_j(t)} \rho_{ij}^0 l_j(t) - \sum_{j=1, j\neq i}^{N} \sigma_{ji} \frac{[w_j(t) - w_i(t)]^+}{w_i(t)} l_i(t), \tag{1.4}$$

где относительная производственная функция, уровень заработных плат и процентная ставка капитала заданы следующими равенствами:

$$y_i(t) = \frac{Y_i(t)}{Y_i(0)} = k_i(t)^{\alpha_i} l_i(t)^{1-\alpha_i}, \quad w_i(t) = \lambda_i^0(1-\alpha_i)\left(\frac{k_i(t)}{l_i(t)}\right)^{\alpha_i}, \quad r_i(t) = \kappa_i^0\alpha_i\left(\frac{l_i(t)}{k_i(t)}\right)^{1-\alpha_i},$$

$$t \geq 0, \quad i = \overline{1,N}. \tag{1.5}$$

Зададим начальные условия для описанных выше уравнений:

$$k_i(0) = 1, \quad l_i(0) = 1, \quad i = \overline{1,N}. \tag{1.6}$$

Таким образом, уравнения (1.3) – (1.5) вместе с начальными условиями (1.6) описывают динамику исследуемой системы.

1.2 Функция полезности

Одной из важнейших особенностей экономических моделей является специфика введения функции полезности потребителя. Эта специфика и определяет в значительной степени дальнейшие построения и выводы моделей. Среди бесконечного количества благ, имеющихся на рынке, потребитель должен выбрать те, что, с одной стороны, удовлетворяют его потребности, а, с другой, соотносятся с его возможностями. Каждый потребитель заинтересован максимизировать общее количество полезности, которое он получает.

Определение 3. *Функция полезности является мерой соотношения между объемами потребляемых благ и уровнем полезности и может быть представлена в виде:*

$$U = \int_0^\infty e^{-dt} L(t) \ln\left(\frac{(1-s)Y(t)}{L(t)}\right) dt,$$

где Y — производственная функция, $L(t)$ - объем труда за время t, s - ставка инвестирования, d - коэффициент дисконтирования.

Рассмотрим функцию полезности потребления i-ой страны для нашей задачи в терминах введенных относительных показателей:

$$U_i = \int_0^\infty e^{-d_i t} L_i(t) \ln\left(\frac{(1-s_i)Y_i(t)}{L_i(t)}\right) dt = L_i(0)\int_0^\infty e^{-d_i t} l_i(t) \ln\left(\frac{(1-s_i)\lambda_i^0 y_i(t)}{l_i(t)}\right) dt.$$

Таким образом, i-ая страна стремится максимизировать функционал U_i, управляя параметрами $\tau_{ij}, \tau_{ji}, \sigma_{ij}, \sigma_{ji}$.

2 Постановка задачи в терминах машинного обучения с подкреплением

Введем следующий функционал:

$$\hat{U}_{i,t_0} = \int_{t_0}^{t_0+1} l_i(t) \ln \left(\frac{(1-s_i)\lambda_i^0 y_i(t)}{l_i(t)} \right) dt$$

Определим среду.

Каждый агент представляет собой страну.

Каждая сессия длится ровно N шагов, состояние после N-го шага считается терминальным. Важно, что на каждом шаге все агенты получают одно и то же наблюдение, совершают действия, и лишь затем происходит расчет награды и генерация следующего состояния. На каждом шаге t_0 возрастает на единицу, а также l_i^0 и k_i^0 приравниваются к $l_i(1)$ и $k_i(1)$ из предыдущего шага соответственно.

i-ая страна контролирует значения параметров $\tau_{ij}^i, \tau_{ji}^i, \sigma_{ij}^i, \sigma_{ji}^i$. Итоговые же значения параметров высчитываются согласно формулам:

$$\tau_{ij} = \frac{\tau_{ij}^i + \tau_{ij}^j}{2}, \quad \sigma_{ij} = \frac{\sigma_{ij}^i + \sigma_{ij}^j}{2},$$
$$i = \overline{1,N}, \quad j = \overline{1,N}, \quad i \neq j.$$

Состояние можно описать набором всех переменных среды, однако такое описание было бы излишним, поскольку обучаемые агенты предназначены для оптимизации политики своей страны в конкретной конфигурации некоторых стационарных параметров. Следовательно, состоянием будем считать набор всех меняющихся параметров среды:

$$\{t_0, k_i^0, l_i^0, \tau_{ij}^i, \sigma_{ij}^j\},$$

$$i = \overline{1,N}, \quad j = \overline{1,N}, \quad i \neq j.$$

В качестве наблюдений агенты будут получать весь этот набор.

Действием i-ого агента будем считать изменение параметров $\tau_{ij}^i, \tau_{ji}^i, \sigma_{ij}^i, \sigma_{ji}^i$ не более, чем на один шаг по дискретной сетке значений (-1, 0, либо 1 для каждого параметра). Размер сетки и границы значений параметров являются парметрами среды.

Награду i-го агента в момент времени t_0 определим следующим образом:

$$\frac{\hat{U}_{i,t_0} - \hat{U}_{i,t_0-1}}{\hat{U}_{i,t_0}}.$$

Взятие отношения функций полезности потребления вместо их абсолютных значений мотивировано ее экспоненциальным ростом по времени. Таким образом, абсолютное значение

функции потребления на первых шагах симуляции было бы незначительно мало относительно последних, и агент не имел бы достаточной мотивации действовать оптимально в начале сессии. Предложенный вид награды решает данную проблему, заставляя агента максимизировать процентный рост функции полезности потребления, что неизбежно влечет за собой и максимизацию ее абсолютного значения.

3 Метод решения

Ниже в разделах 3.1-3.4 кратко описываются используемые методы и приёмы машинного обучения, а в разделе 3.5 описывается схема, использованная для решения поставленной задачи.

3.1 Deep Q-learning

Определение 4. *Функция ценности действия в состоянии (Q-функция) в состоянии S для действия A под политикой π — это матожидание дисконтированной награды агента при условии, что в текущем состоянии S агент совершит действие A, а далее будет действовать согласно политике π:*

$$Q_\pi(s,a) := \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[R_t + \sum_{k=1}^{\infty} \gamma^k R_{t+k} \,\middle|\, S_t = s, A_t = a\right].$$

Определение 5. *ϵ-жадная политика — это политика, выбирающая в состоянии S случайное действие с вероятностью ϵ и действие, максимизирующее Q-функцию, с вероятностью $1-\epsilon$:*

$$\pi(S) = \begin{cases} random\ action\ with\ probability\ \epsilon \\ argmax_a Q(S,a)\ with\ probability\ 1-\epsilon \end{cases}$$

Классический метод Q-learning основан на оценке агентом функции ценности действия в состоянии, при помощи которой агент может оценивать выгоду совершения того или иного действия в состоянии и принимать решения. На основе получаемого от среды вознаграждения агент постепенно уточняет свою оценку, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Главное преимущество Q-learning состоит в том, что алгоритм способен сравнивать ожидаемую полезность доступных действий, не формируя явной модели среды.

Algorithm 1 Псевдокод классической версии Q-learning

```
Initialize γ ∈ [0, 1)                                   ▷ коэффициент дисконтирования
Initialize π_ε, ε ∈ [0, 1]                              ▷ ε-жадная политика
Initialize α ∈ (0, 1)                                   ▷ Параметр скорости обучения
Q(s, a) ← Random                                        ▷ Случайно инициализируем Q-функцию
for each episode do
    Initialize S                                        ▷ Начальное состояние
    while S is not terminal do
        A ← π_ε(s)                                      ▷ Выбираем действие, согласно ε-жадной политике
        Take action A, observe R, S'
        Q(S, A) ← Q(S, A) + α[R + γ max_a Q(S', a) − Q(S, A)]   ▷ Обновляем Q-функцию
        S ← S'
    end while
end for
```

Deep Q-learning отличается от классической версии тем, что использует нейронную сеть для оценки Q-функции. Это становится особенно важно в случае, когда множество состояний очень велико, либо бесконечно.

Algorithm 2 Псевдокод Deep Q-learning

Initialize $\gamma \in [0, 1)$ ▷ коэффициент дисконтирования
Initialize $\pi_\epsilon, \epsilon \in [0, 1]$ ▷ ϵ-жадная политика
Initialize network $Q(s, a)$ with random weights ▷ Случайно инициализируем веса Q
for each episode **do**
  Initialize S ▷ Начальное состояние
  while S is not terminal **do**
    $A \leftarrow \pi_\epsilon(s)$ ▷ Выбираем действие, согласно ϵ-жадной политике
    Take action A, observe R, S'
    $y \leftarrow \begin{cases} R, & if\ S'\ is\ terminial \\ R + \gamma \max_a Q(S', a), & otherwise \end{cases}$ ▷ Считаем новое значение Q-функции
    Perform gradient descent step on $[y - Q(S, a)]^2$ ▷ Обновляем веса нейронной сети
    $S \leftarrow S'$
  end while
end for

3.2 Experience Replay

Experience Replay — это прием, используемый в алгоритме Deep Q-learning, позволяющий извлекать больше информации из действий, совершенных на предыдущих шагах, а также добиться лучшей сходимости Q-функции.

Данный прием заключается в хранении буфера размера N, состоящего из кортежей типа $\{s, a, r, s', f\}$, где на каждом шаге обучения в буффер кладутся текущее состояние s, совершенное действие a, награда за совершенное действие r, следующее состояние s' и флаг f, отвечающий за терминальность состояния s'. Затем при обновлении весов нейросети используется градиентный спуск по случайному батчу из M кортежей, извлеченных из буфера. При переполнении буфера из него удаляется кортеж, добавленный туда ранее всех.

3.3 Target Network

Target Network — это прием, используемый в алгоритме Deep Q-learning, позволяющий сделать обучение нейросети стабильнее, а также улучшить сходимость.

Заключается данный прием в модификации таргета при обучении нейросети. Теперь в алгоритм добавляется вторая нейронная сеть Q_{target} с той же архитектурой, что и Q. Она используется для оценки Q-функции следующего состояния, после того, как было предпринято действие и получена награда в текущем. Q_{target} не обучается, а лишь копирует веса Q через каждые C шагов.

Algorithm 3 Псевдокод Deep Q-learning с использованием Experience Replay и Target Network

Initialize $\gamma \in [0, 1)$ ▷ коэффициент дисконтирования
Initialize $\pi_\epsilon, \epsilon \in [0, 1]$ ▷ ϵ-жадная политика
Initialize experience replay buffer D with size N
Initialize M ▷ Размер батча для шага обучения
Initialize $Q(s, a)$ with random weights ▷ Случайно инициализируем веса Q
$Q_{target}(s, a) \leftarrow Q(s, a)$
Initialize C ▷ Частота обновления весов Q_{target}

Initialize S ▷ Начальное состояние
while size of D $<$ N **do** ▷ Заполняем буфер повтора
    Take random action A, observe R, S', *done*
    Store transition $(s, a, r, s', done)$ in experience replay buffer D
end while

for each step **do** ▷ Процесс обучения
    $A \leftarrow \pi_\epsilon(s)$ ▷ Выбираем действие, согласно ϵ-жадной политике
    Take action A, observe R, S', *done*
    Store transition $(s, a, r, s', done)$ in experience replay buffer D
    Sample random batch B of M transitions from D ▷ Сэмплируем батч для шага обучения
    for every transition $(s_i, a_i, r_i, s'_i, done_i)$ in B **do**
        $$y_i \leftarrow \begin{cases} r_i \quad if\ done_i \\ r_i + \gamma \max_{a'} Q_{target}(s'_i, a') \end{cases}$$
    end for
    $L \leftarrow \frac{1}{M} \sum_{i=0}^{M-1} (Q(s_i, a_i) - y_i)^2$ ▷ Считаем значние лосс-функции
    Update Q using the SGD algorithm by minimizing loss L ▷ Шаг обучения нейронной сети
    Every C steps, copy weights from Q to Q_{target} ▷ Обновляем веса Q_{target}
    $S \leftarrow S'$
end for

3.4 Многослойный перцептрон

Определение 6. *Многослойный перцептрон (MLP) — это класс нейронных сетей, состоящих из входного слоя, скрытых слоев и выходного слоя. За исключением входных, все нейроны используют нелинейную функцию активации.*

В качестве активационных функций наиболее часто используются усеченное линейное преобразование (ReLU), сигмоида и гиперболический тангенс.

MLP позволяют находить приближенные решения для чрезвычайно сложных задач. В частности, они являются универсальным аппроксиматором функций, поэтому с успехом используются в построении регрессионных моделей.

3.5 Метод решения

Для решении задачи был испльзован алгоритм Deep Q-Learning с использованием Experience Replay и Target Network. Также был применен прием с линейным уменьшением ϵ по времени.

Для оценки Q-функции каждый агент использовал MLP со следующей архитектурой:

$$Observation \to Linear(StateDim, 64) \to ReLU \to$$

$$\to Linear(64, 128) \to ReLU \to$$

$$\to Linear(128, NumberOfActions) \to QValues$$

В качестве бейзлайна были также реализованы агенты, придерживающиеся наивных стратегий: безусловного открытия границ (далее, OBA — Opened Borders Agent), безусловного закрытия границ (далее, CBA - Closed Borders Agent) и случайной стратегии (далее, RA - Random Agent).

Для вычисления значений k и l использовался численный метод Эйлера для решения системы дифференциальных уравнений.

Для взятия интеграла в $\hat{U}_{i,t_0}$ использовался метод трапеций.

4 Результаты

Под поставленную задачу была написана программа на языке программирования Python. Код включает в себя реализацию среды, агентов, процесса обучения и процесса валидации. Приведенные далее графики были построены при помощи модуля matplotlib.

4.1 Параметры модели

Были проведены эксперименты для среды с двумя странами: страной с развивающейся экономикой и страной с развитой. Ниже приведены параметры среды и агентов, при помощи которых были получены рассматриваемые результаты. Значения экономических параметров среды были взяты из World Bank Open Data [10].

	Развитая страна	Развивающаяся страна
Количество шагов в сессии	100	
Количество дискретных значений τ	10	
Количество дискретных значений σ	10	
Количество сегментов разбиения в методе Эйлера	10	
Диапазон значений τ	[0, 0.7]	
Диапазон значений σ	[0, 0.003]	
α_i	$\frac{1}{3}$	$\frac{1}{3}$
n_i	0.0143	0.0042
σ_i	0.05	0.05
s_i	0.2185	0.228
λ_i^0	12369	91363
κ_{0i}	0.324	0.2193
ρ^0	5.4	$\frac{1}{5.4}$

Таблица 1: Параметры среды

Размер буфера experience replay	10000
Размер батча	64
Периодичность копирования весов Q в Q_{target}	5000
Метод оптимизации	Adam (adaptive moment estimation)
Скорость обучения (learning rate)	10^{-5}

Таблица 2: Параметры агентов

4.2 Обучение

Ниже приведены графики обучения для трех различных сценариев: обучение обоих агентов, обучение только развивающейся страны и обучение только развитой страны. Действия

необучаемых агентов можно считать условно случайными, поскольку Q-функция, согласно которой они совершают действия, представляет собой нейронную сеть, инициализированную случайными весами.

Каждый из рис. 1-3 состоит из четырех графиков, позволяющих отследить динамику процесса обучения каждого из агентов:

1. Левый верхний угол. График динамики средней награды за сессию для каждого агента. По данному графику можно судить о раскладе сил агентов во времени, а также, косвенно, о сходимости процесса обучения.

2. Правый верхний угол. Сглаженный график динамики temporal-difference-лосса. График отображает способность агента к предсказанию получаемой награды.

3. Левый нижний угол. График показывает оценку агентом ценности начального состояния. Иначе говоря, предсказываемую агентом дисконтированную награду за всю сессию.

4. Правый нижний угол. Сглаженный график значения нормы градиента при шаге градиентного спуска нейронной сети. Отображает степень изменения весов нейронной сети, предсказывающей Q-функцию.

4.2.1 Обучение обоих агентов

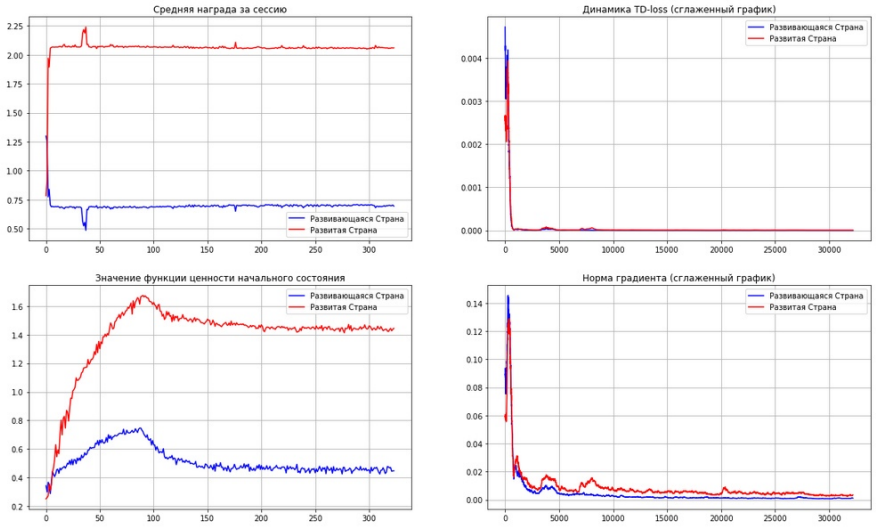

Рис. 1: Оба агента обучаются

Агенты сходятся к своим оптимальным политикам довольно быстро, лишь изредка отклоняясь от них. Агенты также достаточно быстро научаются предсказывать получаемую награду, однако сходимость значений функции значимости начального состояния занимает значительно больше времени.

4.2.2 Обучение только развивающейся страны

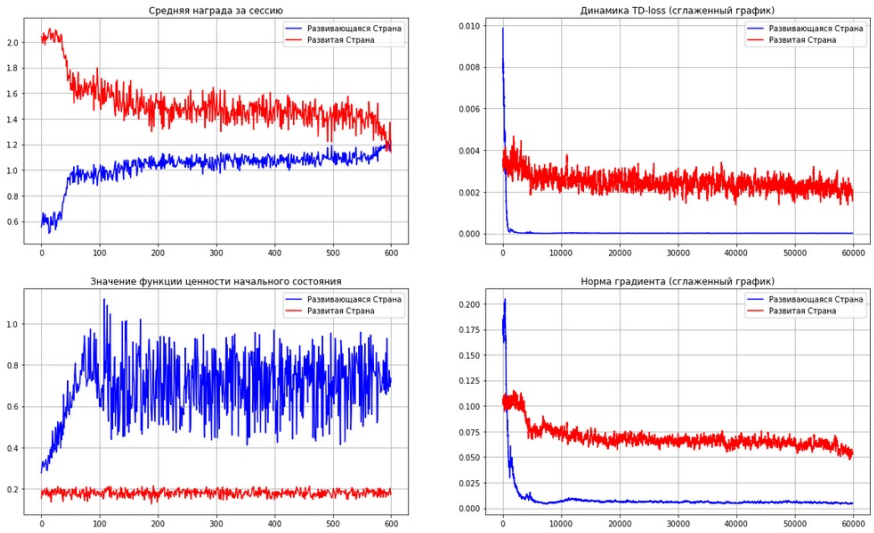

Рис. 2: Обучается только развивающаяся страна

В данном сценарии развивающаяся страна достаточно быстро находит субоптимальную политику, однако не перестает ее улучшать вплоть до окончания обучения. Такой эффект можно объяснить большой вариативностью поведения развитой страны (из-за условной случайности ее политики), на приспособление к которой требуется время. Можем также заметить, что получамые награды разительно отличаются от предыдущего сценария, из чего следует, что обучение проходит успешно, а случайная политика оказывается не готова выдерживать сопротивление обученного оппонента.

4.2.3 Обучение только развитой страны

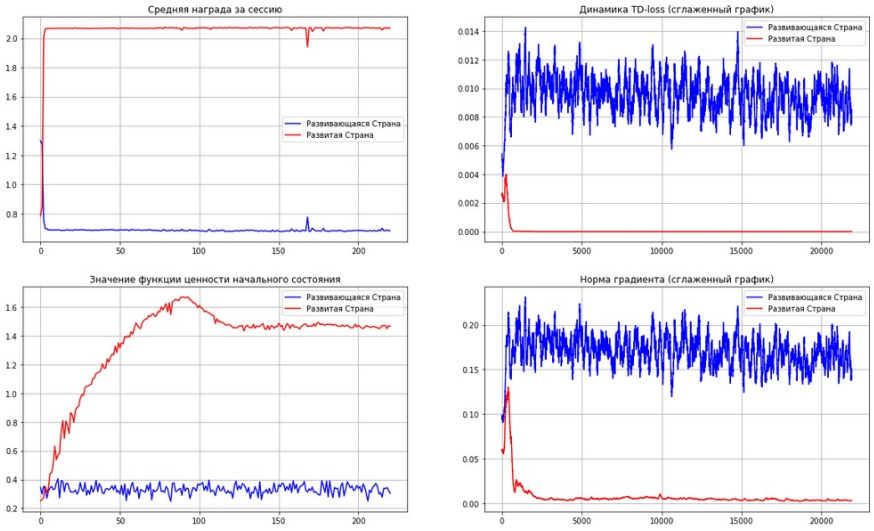

Рис. 3: Обучается только развитая страна

Довольно примитивный сценарий, в котором развитая страна без труда находит способ доминировать над развивающейся. Видим, что все графики сходятся быстро, а также что результаты вновь заметно отличаются от полученных в первом сценарии: развитой стране удается получать еще более высокую награду, что свидетельствует об успешности процесса обучения.

4.3 Валидация

Ниже приведены результаты валидации в виде таблицы.

Строка — агент-развивающаяся страна, столбец — агент-развитая страна.

Значения в ячейках — средняя награда первого и второго агента соответственно, полученная в результации ста симуляций среды, нормированная на количество шагов в каждой симуляции, умноженная на 100.

	RA	OBA	CBA	DQNA
RA	0.660 2.047	0.363 2.228	1.041 1.629	0.363 2.229
OBA	0.375 2.223	0.16 2.333	0.687 2.067	0.012 2.335
CBA	1.027 1.669	0.683 2.065	1.391 0.503	0.688 2.069
DQNA	1.033 1.651	0.688 2.073	1.392 0.502	0.700 2.074

Видим, что для каждой страны максимальной награды удается достичь обученному агенту. Однако политика закрытия границ развивающейся страной и политика открытия границ развитой дают очень близкие к максимальным результаты. Случайные политики оказываются абсолютно несостоятельны, а политика открытия границ развивающейся страной и политика закрытия границ развитой и вовсе показывают минимальные результаты.

4.4 Динамика экономических показателей

Ниже представлены графики различных экономических показателей, полученных в результате симуляции одной сессии для двух обученных агентов.

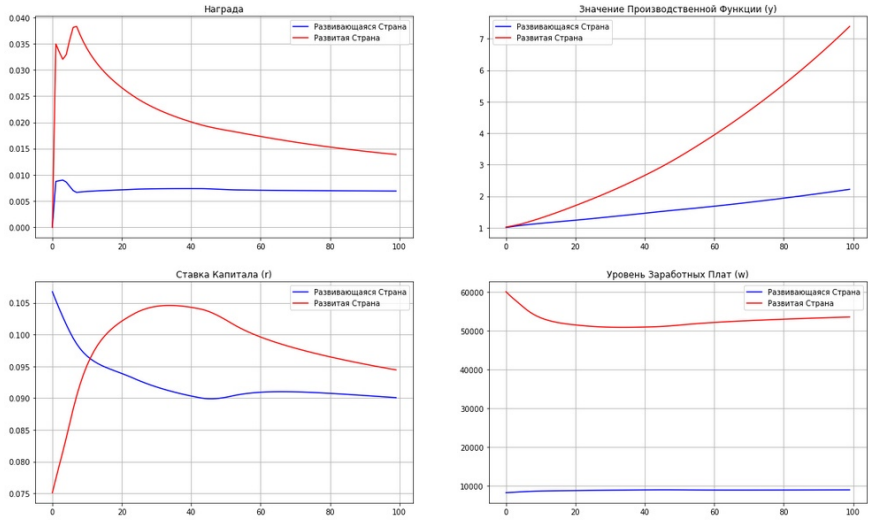

Рис. 4: Значение награды и других экономических показателей от времени

Левый верхний график отображает динимику значений моментальной награды агентов. Развивающаяся страна монотонно увеличивает свой функционал ползености потребления, тогда как развитая стремительно увеличивает его в начале сессии, постепенно сбавляя скорость в концу.

Правый верхний график показывает, что обеим странам действительно удается увеличивать значения своих производящих функций, что свидетельствует об удачном дизайне награды среды. Также стоит отметить, что рост функций производства имеет близкий к линейному характер.

Левый и правый нижние графики отображают ставку капитала r_i и уровень заработных плат w_i соответственно. Развитой стране удается довольно быстро увеличить свою ставку капитала, разворачивая тем самым поток капитала в свою сторону. Уровень заработных плат обеих стран остается практически неизменным в течение сессии.

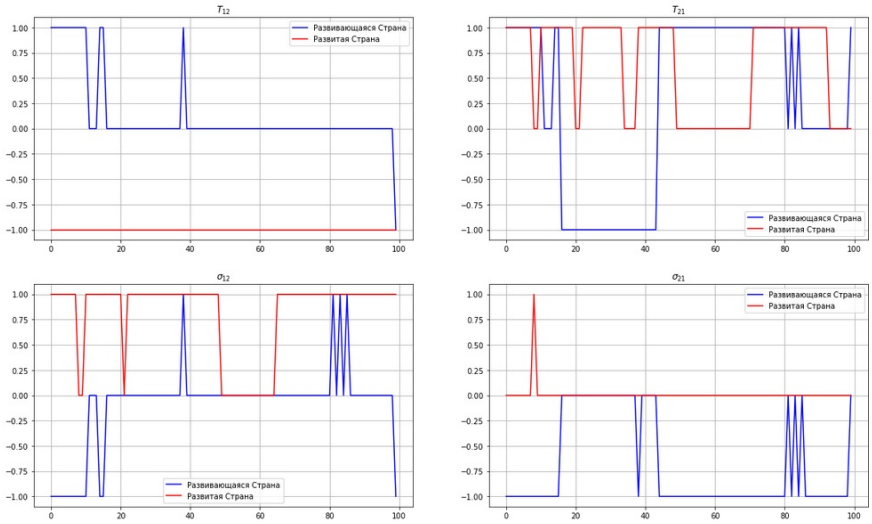

Рис. 5: Действия обученных агентов по каждому параметру от времени

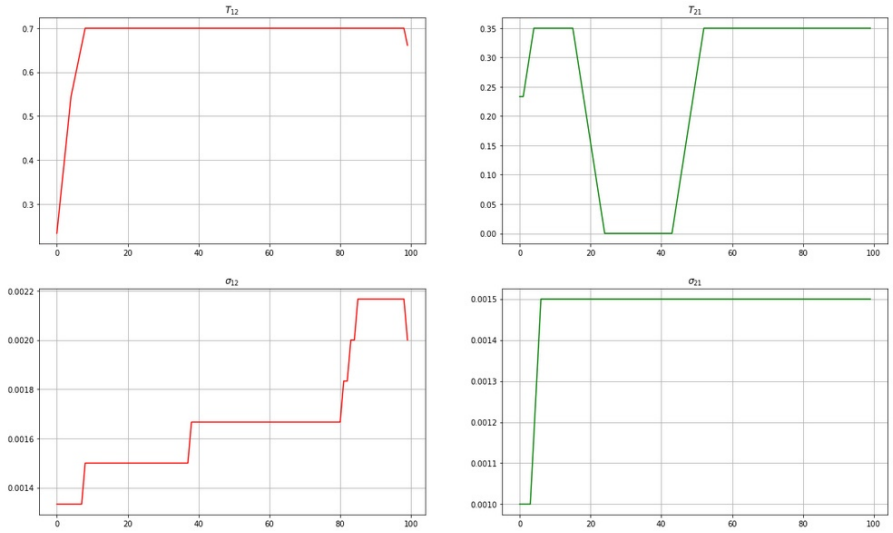

Рис. 6: Значения управляемых параметров от времени

Рис. 5 и 6 демонстрируют политику государств и динамику управляемых параметров во времени соответственно. Обе страны придерживаются довольно примитивной стратегии при контроле потока трудовых ресурсов: развивающаяся страна стремится максимально закрыть поток, а развитая — открыть. Политики же стран касательно потоков трудовых ресурсов меняются во времени: в начале обе страны стремятся максимально расширить поток, а затем, после разворота этого потока в сторону развитой страны, развивающаяся принимает политику его сужения, тогда как развитая начинает контролировать его в обе стороны в зависимости от ситуации.

5 Выводы

На основе полученных результатов можно сделать следующие выводы:

1. Рост награды одного агента имеет отрицательную корелляцию с ростом награды второго. Это означает, во-первых, что, максимизируя свои функционалы полезности потребления, агенты конкурируют друг с другом, а, во-вторых, что в рассматриваемой модели соблюдается в некоторым смысле баланс экономических показателей.

2. Политики агентов сходятся. Об этом свидетельствуют приведенные выше графики, и, в частности, график средней награды агентов за сессию.

3. Обученные агенты показывают лучший результат, чем наивные стратегии (безусловное открытие/закрытие границ, случайное изменение параметров). Несмотря на то, что агенты выучивают довольно близкие к наивным политики, более тонкий контроль параметров в определенных ситуациях позволяет им достичь более высокого результата.

4. Развивающаяся страна предпочитает закрывать свои границы, а развитая — открывать. Это обусловлено тем, что большую часть времени потоки направлены именно в сторону развитой страны.

5. Полезность потребления и производимый продукт обеих стран растут со временем. Данный факт свидетельствует о росте экономики внутри модели в целом, а также о результативности политик обоих государств.

6 Дальнейшие направления исследований

1. Симуляция среды с $N > 2$ странами. Возможны различные конфигурации: например, одна развитая страна против двух развивающихся или две развитых против одной развивающейся. Данный эксперимент требует достаточно высоких технических ресурсов, поскольку симуляция среды даже для двух агентов — довольно медленный процесс.

2. Варирование стационарных параметров среды, таких как $\alpha_i, \sigma_{i,i}$ и другие. В различной конфигурации этих параметров агенты могут выучивать различные стратегии, которые было бы интересно изучить.

3. Введение в награду штрафа за изменение параметров. Данный штраф мотивируется естественными издержками государства за изменение экономической политики.

4. Реализация «универсального» агента, т.е. агента, действующего в условиях изменчивости всех переменных среды. В этом случае в наблюдение агента будут входить все параметры, и, следовательно, процесс обучения будет проходить гораздо сложнее, поскольку политика агента будет обусловлена не только политикой стран-оппонентов, но и экономическими параметрами государств, участвующих в симуляции среды.

5. Введение в среду социального планировщика — агента, оптимизирующего общее благосостояние системы. Функционалом качества социального планировщика может служить, например, взвешенная сумма функционалов полезности потребления всех государств или индекс Джини, отвечающий за равномерность распределения благ между странами.

7 Заключение

По результатам проделанной работы можно заключить, что методы машинного обучения с подкреплением могут быть успешно применены к области оптимизации экономической и миграционной политик государства. Кроме того, данные методы хорошо адаптируются под малейшие изменения задачи, и, таким образом, имеют высокий потенциал послужить полезным инструментом для первичной оптимизации политики государства в условиях сильной изменчивости внешнеполитичисеких факторов и экономических целей государства.

Были проанализированы выучиваемые агентами политики, а также обозначены возможные пути дальнейших исследований в данной области.

Список литературы

[1] Мартынова Ю.А. Методы машинного обучения при оценке конкурентоспособности предприятия // Вопросы инновационной экономики. – 2020. – Том 10. – № 1. – С. 549-562. – doi: 10.18334/vinec.10.1.100669.

[2] Кричевский М.Л., Дмитриева С.В. Оценка эффективности труда методами машинного обучения // Экономика труда. – 2021. – Том 8. – № 9. – С. 945-960. – doi: 10.18334/et.8.9.113430.

[3] Lucas, Robert (1976), Econometric Policy Evaluation: A Critique, in Brunner, K. Meltzer, A., The Phillips Curve and Labor Markets, vol. 1, Carnegie-Rochester Conference Series on Public Policy, New York: American Elsevier, с. 19–46, ISBN 0444110070

[4] Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, Richard Socher. The AI Economist: Optimal Economic Policy Design via Two-level Deep Reinforcement Learning // arXiv:2108.02755 [cs.LG]

[5] Robert M. Solow.A Contribution to the Theory of Economic Growth. The Quarterly Journal of Economics, The MIT Press (1956), pp. 65-94 (29 pages)

[6] Sutton, R. S., Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[7] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, Will Dabney. Revisiting Fundamentals of Experience Replay // arXiv:2007.06700 [cs.LG]

[8] https://towardsdatascience.com/deep-q-network-dqn-ii-b6bf911b6b2c

[9] Hastie, Trevor. Tibshirani, Robert. Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.

[10] https://data.worldbank.org/