



Банк России



## Прогнозирование в агентно-ориентированных моделях на основе амортизированных нейронных сетей

Серия докладов об экономических исследованиях

№ 115 / июль 2023

Д. Кошелев  
А. Пономаренко  
С. Селезнев

**Денис Кошелев**

Банк России, Департамент исследований и прогнозирования

E-mail: [koshelevdm@cbr.ru](mailto:koshelevdm@cbr.ru)**Алексей Пономаренко**

Банк России, Департамент исследований и прогнозирования

E-mail: [ponomarenkoaa@cbr.ru](mailto:ponomarenkoaa@cbr.ru)**Сергей Селезнев**

Банк России, Департамент исследований и прогнозирования

E-mail: [seleznevsm@cbr.ru](mailto:seleznevsm@cbr.ru)

Препринты Банка России проходят процедуру анонимного рецензирования со стороны членов Консультативного исследовательского совета Банка России и внешних рецензентов.

Авторы выражают благодарность Сергею Иващенко за его комментарии и советы.

Содержание настоящего доклада по экономическим исследованиям отражает личную позицию авторов. Результаты исследования являются предварительными и публикуются с целью стимулировать обсуждение и получить комментарии для возможной дальнейшей доработки материала. Содержание и результаты исследования не следует рассматривать, в том числе цитировать в каких-либо изданиях, как официальную позицию Банка России или указание на официальную политику или решения регулятора. Любые ошибки в данном материале являются исключительно авторскими.

Все права защищены. Воспроизведение представленных материалов допускается только с разрешения авторов.

Фото на обложке: Shutterstock/FOTODOM

107016, Москва, ул. Неглинная, 12

Тел.: +7 499 300-30-00, +7 495 621-64-65 (факс)

Официальный сайт Банка России: [www.cbr.ru](http://www.cbr.ru)

© **Центральный банк Российской Федерации, 2023**

## Резюме

В этой статье мы предлагаем новую процедуру для безусловного и условного прогнозирования в агентно-ориентированных моделях. Наш алгоритм основан на применении амортизированных нейронных сетей и состоит из двух шагов. На первом шаге симулируются искусственные наборы данных из модели. На втором шаге на искусственных данных обучается нейронная сеть, которая на основе истории наблюдений прогнозирует следующие значения переменных. Основным достоинством предложенного алгоритма является его скорость, ведь после процедуры обучения он может быть использован для получения прогнозов на практически любых данных без дополнительных симуляций или переобучения нейронной сети.

**JEL коды:** C11, C15, C32, C45, C53, C63.

**Ключевые слова:** агентно-ориентированные модели, амортизированный симуляционный алгоритм оценки, байесовские модели, нейронные сети, прогнозирование.

**Оглавление**

1.	Введение.....	5
2.	Алгоритм.....	7
2.1.	Безусловное прогнозирование .....	7
2.2.	Условное прогнозирование .....	9
3.	Метрики качества.....	10
4.	Результаты экспериментов .....	13
4.1.	Проверка концепции .....	13
4.2.	Безусловный прогноз.....	15
4.3.	Условный прогноз.....	16
5.	Дискуссия.....	17
6.	Заключение.....	19
	Список литературы .....	20
	Приложение А. Таблицы и рисунки.....	23
	Приложение Б. Модели .....	32
Б.1.	Байесовская AR(1)-модель.....	32
Б.2.	ABM .....	32
	Приложение В. Условный прогноз для универсальной архитектуры .....	36

## 1. Введение

Агентно-ориентированные модели (Agent-Based Models, далее – ABM) становятся все более популярными как среди экономистов в академическом сообществе, так и в экономических институтах. Являясь симуляционными, они не требуют теоретического решения и позволяют легко расширить различного рода предпосылки, такие как гомогенность/репрезентативность агентов, их рациональность и наличие полной информации, которые часто закладываются в классические экономические модели (см. Fagiolo and Roventini (2017), Haldane and Turrell (2018) и Axtell and Farmer (2022)). Однако их гибкость, которая позволяет моделировать динамику множества агентов, служит и их недостатком. Реалистичные ABM зачастую содержат от сотен (см. Delli Gatti et al. (2011)) до миллионов агентов (см. Poledna et al. (2023)) и представляются в виде нелинейных моделей пространства состояний, которые содержат пропорциональное числу агентов количество скрытых переменных. Оценка параметров и прогноз в моделях такого размера являются вычислительно сложными задачами и до конца не изучены в литературе.

Оценке параметров ABM посвящено большое количество статей (см. обзор в Duer et al. (2022a)), и, несмотря на то что работа еще ведется, использование симуляционных алгоритмов оценки (далее – SBI, см. Cranmer et al. (2021)) позволило достичь определенного прогресса в этой области. Прогнозирование же на основе ABM является во многом «неизведанной территорией». Axtell and Farmer (2022) в разделе, посвященном будущим возможностям и вызовам в ABM, характеризуют состояние этого направления следующим образом: *«Традиционные макромоделли строятся в терминах агрегированных переменных, которые совпадают с доступными данными об экономике. Это удобно, поскольку эти данные можно напрямую использовать для инициализации моделей. Однако ABM – это динамические системы, которые моделируют мир на уровне отдельных агентов, таких как домохозяйства и фирмы. Чтобы запустить модель, необходимо инициализировать состояния отдельных агентов таким образом, чтобы они соотносились с агрегированными данными. Для того чтобы сделать это корректным образом, требуются все данные об агентах на микроуровне, которые обычно недоступны.*

*При отсутствии таких данных возникает необходимость в разработке процедуры инициализации, которая формирует правдоподобные состояния для каждого агента, соотносимые после агрегации с наблюдаемыми данными. Однако отдельные состояния также должны быть совместимы друг с другом и внутренней динамикой модели. Если состояния несовместимы в этом смысле, то модель будет генерировать поведение,*

*результатом которого станут плохие прогнозы. Учитывая то, что прогнозы модели никогда не бывают идеальными, – это рекуррентная проблема; с течением времени прогнозы неизбежно разойдутся с наблюдаемыми данными, и процедуру инициализации будет необходимо повторять снова и снова. Поиск хороших методов для этого – открытая задача, которую необходимо решить, если мы хотим использовать АВМ для прогнозирования временных рядов»<sup>1</sup>.*

В настоящий момент на практике прогнозирование в АВМ осуществляется либо, как пишут Axtell and Farmer (2022), на основе инициализации всех состояний с использованием микроданных или их распределений с последующей симуляцией данных (см. Hommes et al. (2022), либо с использованием не требующих инициализации состояний всех агентов суррогатных процедур на основе приближения совместных распределений последних наблюдаемых данных и прогнозных значений переменных (см. Delli Gatti and Grazzini (2020)<sup>2</sup>. Первый метод, как отмечается у Axtell and Farmer (2022), требует знания большого количества данных, которые не всегда доступны исследователям, а второй является вычислительно сложным для задач с более чем двумя переменными и обладает лишь приближенными свойствами с точки зрения аппроксимации прогнозов.

В данной работе мы предлагаем новый метод для безусловного и условного прогнозирования в АВМ, который позволяет строить прогнозы на основе наблюдаемых данных, не проводя при этом инициализации всех скрытых состояний. Этот метод основан на идее метаобучения (см. Finn and Levine (2019) на симулированных данных<sup>3</sup> и состоит из двух этапов. На первом этапе, как и в SBI, симулируется множество искусственных наборов данных. На втором этапе, как и в алгоритмах метаобучения, с помощью набора похожих задач обучается алгоритм прогнозирования, который на основе истории наблюдений для конкретного датасета, прогнозирует его будущие значения. В отличие от алгоритмов, предложенных ранее, наш алгоритм обладает свойством амортизации, то есть – обученный однажды, он может быть использован практически для любых данных. Более того, в отличие от прогнозирования на основе суррогатных алгоритмов, как, например, в Delli Gatti and Grazzini (2020), описанный ниже алгоритм обладает хорошими теоретическими свойствами и

---

<sup>1</sup> Перевод авторов.

<sup>2</sup> Несмотря на наличие ряда работ по оценке ненаблюдаемых состояний моделей на основе фильтров частиц (см. Loh (2018), нам не удалось найти работы, в которых использовались бы оценки на основе фильтра частиц для прогнозирования в АВМ-моделях, сравнимых по размеру с используемыми на практике.

<sup>3</sup> См. примеры обучения прогнозных моделей на симулированных данных в контексте метаобучения, представленные в Harrison et al. (2020) и Garnelo et al. (2018), которые послужили мотивацией для описанного в этой работе алгоритма.

при достаточно гибкой архитектуре нейронной сети должен сходиться в точности к апостериорному распределению прогноза.

Несмотря на то что основной интерес для нас представляет применение алгоритма именно в контексте АВМ, используемая процедура является достаточно общей и может быть применена для широкого круга задач, поэтому в разделе 2 мы описываем процедуры условного и безусловного прогнозирования в общем виде, практически не касаясь специфики АВМ. В разделе 3 обсуждаются метрики, которые будут в дальнейшем использованы для оценки качества предложенного алгоритма. Раздел 4 посвящен описанию результатов экспериментов. В разделе 5 мы обсуждаем несколько важных вопросов, касающихся оценки качества работы алгоритма и его потенциальные расширения. В разделе 6 содержатся заключительные выводы.

## 2. Алгоритм

Мы предполагаем, что модель задается в виде классической байесовской модели пространства состояний с априорным распределением на параметры модели  $p(\theta)$ , уравнениями состояний  $s_t \sim p(s|s_{t-1}, \theta)$  и уравнениями наблюдений  $y_t \sim p(y|s_t, \theta)$ <sup>4</sup>, где  $\theta$  – вектор параметров модели,  $s_t$  – вектор состояний модели в момент времени  $t$ , а  $y_t$  – вектор наблюдаемых переменных в момент времени  $t$ . При этом дополнительно предполагается, что есть возможность симулировать наборы данных, состоящие из наблюдаемых переменных.

В подразделах 2.1 и 2.2 мы опишем базовую процедуру безусловного прогнозирования, а также ее модификацию на случай условного прогнозирования.

### 2.1. Безусловное прогнозирование

Основная идея состоит в том, что прогноз из байесовской модели пространства состояний в момент времени  $t$  может быть, по сути, представлен как оценка ненаблюдаемых в этот момент состояний  $y_{t+1}, \dots, y_{t+h}$ . Для решения этой задачи мы модифицируем алгоритм оценки некоторых характеристик предельных распределений состояний, описанный в Khabibullin and Seleznev (2022). Алгоритм может быть представлен как двухшаговая процедура. На первом этапе генерируется множество искусственных наборов данных разной длины  $t$ , состоящих из истории наблюдаемых переменных  $x = \{y_1, \dots, y_t\}$  и будущих значений

---

<sup>4</sup> В АВМ это уравнение может иметь как стохастический вид, например, в случае наличия ошибок измерения, которые вводятся в модель чтобы нивелировать небольшие сутевые расхождения между модельными и измеряемыми в жизни показателями, так и детерминистический вид в случае, если модельные переменные однозначно связаны с наблюдаемыми.

этих переменных  $y = \{y_{t+1}, \dots, y_{t+h}\}$ . На втором этапе обучается нейронная сеть, которая по  $x$  предсказывает  $y$ . Формальное описание алгоритма представлено ниже:

**Алгоритм 1. Алгоритм безусловного прогнозирования с использованием амортизированной нейронной сети** ( $N$  – количество симуляций,  $T_{min}$  – минимальная длина рядов для прогнозирования,  $T_{max}$  – максимальная длина рядов для прогнозирования,  $h$  – максимальный горизонт прогнозирования,  $L$  – функция потерь)

1. Генерация искусственных данных:

$$X = \{\}, Y = \{\}$$

Для  $n = 1, \dots, N$ :

1.a. Семплировать параметры из априорного распределения:

$$\theta^n \sim p(\theta)$$

1.b. Семплировать состояния модели при условии параметров:

$$s_0^n \sim p(s_0 | \theta^n)$$

$$s_t^n \sim p(s | s_{t-1}^n, \theta^n), \quad t = 1, \dots, T_{max}$$

1.c. Семплировать наблюдаемые переменные при условии параметров и состояний:

$$y_t^n \sim p(y | s_t^n, \theta^n), \quad t = 1, \dots, T_{max}$$

1.d. Для  $\tau = T_{min}, \dots, T_{max} - 1$  создать наборы данных:

$$x^{n,\tau} = \{y_1^n, \dots, y_\tau^n\}$$

$$y^{n,\tau} = \{y_{\tau+1}^n, \dots, y_{\tau+\min(h, T_{max}-\tau)}^n\}^5$$

и добавить их в выборку:

$$X = X + \{x^{n,\tau}\}$$

$$Y = Y + \{y^{n,\tau}\}$$

2. Обучение нейронной сети:

$$\varphi^* = \underset{\varphi}{\operatorname{argmin}} \sum_{x^i \in X, y^i \in Y} L(y^i, f_\varphi(x^i))$$

В данной работе в качестве функции потерь мы берем кросс-энтропию с диагональным нормальным распределением, которая позволяет корректно оценивать среднее и стандартное отклонение прогнозов, однако, как отмечается в Khabibullin and Seleznev (2022), может быть использован любой M-оценщик (см. главу 5 в Van der Vaart (2000)). Архитектура нейронной сети похожа на архитектуру для DSGE-модели из Khabibullin and Seleznev (2022) и состоит из

<sup>5</sup> Вместо всего набора наблюдаемых переменных может быть выбрано некоторое подмножество интересующих переменных. Это может быть особенно актуально, когда размерность вектора наблюдаемых переменных велика.



сверточных, рекуррентных и полносвязных слоев<sup>6</sup>. Три ключевых отличия: использование односторонних рекуррентных блоков, чтобы избежать «заглядывания в будущее», размерность финальных выходов, которая в нашем случае равна размерности  $y$ , а также добавление сквозной связи между входным и выходным слоями (см. He et al. (2015)).

Предложенный алгоритм обладает свойством амортизации, то есть после обучения нейронной сети он позволяет использовать ее для любых наборов данных, не проводя при этом вычислительно сложных процедур, таких как фильтр частиц (см. Gordon et al (1993)). Вместо этого данные подаются на вход нейронной сети, и за сотые/десятые доли секунды на выходе получают необходимые характеристики распределения прогнозов (среднее и стандартное отклонение в нашем случае).

## 2.2. Условное прогнозирование

Главным отличием условного прогнозирования от безусловного является наличие сценария. Чтобы учесть наличие сценария в Алгоритме 1, мы модифицируем матрицу признаков  $x$ , добавляя в нее переменные сценария  $z$ , предполагая, что они могут быть выражены через наблюдаемые переменные. В этом случае алгоритм прогнозирования запишется следующим образом:

**Алгоритм 2. Алгоритм условного прогнозирования с использованием амортизированной нейронной сети** ( $N$  – количество симуляций,  $T_{min}$  – минимальная длина рядов для прогнозирования,  $T_{max}$  – максимальная длина рядов для прогнозирования,  $h$  – максимальный горизонт прогнозирования,  $L$  – функция потерь)

1. Генерация искусственных данных:

$$X = \{\}, Y = \{\}$$

Для  $n = 1, \dots, N$ :

1.a. Семплировать параметры из априорного распределения:

$$\theta^n \sim p(\theta)$$

1.b. Семплировать состояния модели при условии параметров:

$$s_0^n \sim p(s_0 | \theta^n)$$

$$s_t^n \sim p(s_t | s_{t-1}^n, \theta^n), \quad t = 1, \dots, T_{max}$$

1.c. Семплировать наблюдаемые переменные при условии параметров и состояний:

<sup>6</sup> Четыре типа одномерных сверточных слоев с 16 фильтрами каждый и длинами 3, 5, 7 и 9 соответственно, двухслойный управляемый рекуррентный блок с размерностью 64 и три полносвязных слоя с промежуточной размерностью 100 и усеченным линейным преобразованием (ReLU).

$$y_t^n \sim p(y|s_t^n, \theta^n), \quad t = 1, \dots, T_{max}$$

1.d. Для  $\tau = T_{min}, \dots, T_{max} - 1$  создать наборы данных:

$$x^{n,\tau} = \{[y_1^n, z_1^n(y_1^n, y_2^n, \dots, y_{1+\min(h, T_{max}-1)}^n)], \dots, [y_\tau^n, z_\tau^n(y_1^n, \dots, y_\tau^n, y_{\tau+1}^n, \dots, y_{\tau+\min(h, T_{max}-\tau)}^n)]\}$$

$$y^{n,\tau} = \{y_{\tau+1}^n, \dots, y_{\tau+\min(h, T_{max}-\tau)}^n\}$$

и добавить их в выборку:

$$X = X + \{x^{n,\tau}\}$$

$$Y = Y + \{y^{n,\tau}\}$$

2. Обучение нейронной сети:

$$\varphi^* = \operatorname{argmin}_{\varphi} \sum_{x^i \in X, y^i \in X} L(y^i, f_{\varphi}(x^i))$$

Архитектура нейронной сети для Алгоритма 2 похожа на архитектуру нейронной сети для Алгоритма 1, за исключением того, что на уровне рекуррентного слоя в качестве входов дополнительно добавляются переменные сценария. Отметим, что наличие сценариев именно в терминах наблюдаемых переменных позволяет передавать данные на уровне рекуррентных или сверточных слоев, так как для корректного обучения нейронной сети с такой архитектурой необходимо знание сценариев в прошлых периодах. Чтобы инкорпорировать сценарии, которые потенциально могут включать ненаблюдаемые переменные (например, для расчета импульсных откликов, зависящих от истории наблюдений), в архитектуру нейронной сети, они должны подаваться на вход нейронной сети на слоях, после которых не требуется знания сценариев в другие периоды времени. Для архитектуры, предложенной в подразделе 2.1, это может быть сделано на этапе полносвязных слоев.

Несмотря на то что такая архитектура нейронной сети является более общей с точки зрения количества решаемых задач, в рамках дополнительных экспериментов (см. Приложение В) мы обнаружили, что такие сети зачастую куда более сложно обучаемы и требуют более гибкой архитектуры на уровне полносвязных слоев. Поэтому для моделей, где прогноз строится при условии только наблюдаемых переменных, мы рекомендуем использовать именно Алгоритм 2.

### 3. Метрики качества

Алгоритмы, предложенные в предыдущем разделе, при стремящемся к бесконечности количестве симуляций и достаточно гибкой нейронной сети должны сходиться в точности к среднему и стандартному отклонению апостериорного распределения прогнозов. Однако на

практике количество наблюдений и размер нейронной сети хоть и велики, но тем не менее конечны, а функция потерь содержит множество локальных оптимумов. Все это может приводить к тому, что обученные нейронные сети могут показывать результаты, отличные от апостериорных прогнозов<sup>7</sup>, поэтому проверка качества получаемых аппроксимаций является важным шагом. В данном разделе описываются две метрики, которые мы предлагаем использовать для проверки качества обучения нейронных сетей.

Безусловно, наилучшим с точки зрения ответа на вопрос о качестве аппроксимации характеристик апостериорного распределения прогнозов является сравнение с характеристиками самого апостериорного распределения прогнозов. К сожалению, для большинства симуляционных моделей и, в частности, для АВМ, которые являются фокусом данной работы, построение точного апостериорного распределения прогнозов невозможно, а его аппроксимации на основе МСМС-алгоритмов (см. Andrieu et al. (2010) или последовательных алгоритмов Монте-Карло (SMC, см. Chopin et al. (2012), требуют применения вычислительно сложных процедур с использованием фильтра частиц (см. Gordon et al. (1993) или его вариаций.

**Стандартизация прогнозов.** В исследованиях по вероятностному прогнозированию временных рядов прогнозы часто проверяются путем калибровки интервалов или преобразования кумулятивных распределений (Probability Integral Transforms)<sup>8</sup>. Эти методы не могут быть применены напрямую для ситуации, когда оцениваются только средние и стандартные отклонения прогноза, так как вероятностное распределение прогнозов не задано полностью. Но можно заметить, что аналогично преобразованным кумулятивным распределениям, которые должны совпадать с нормальным распределением с нулевым средним и единичной дисперсией, а также иметь нулевую автокорреляцию, в нашем случае стандартизированные прогнозы (с удаленным средним и поделенные на стандартное отклонение) для хорошо обученной модели должны обладать теми же свойствами, за исключением формы распределения<sup>9</sup>. Таким образом, для тестирования качества мы смотрим на среднее и стандартное отклонение стандартизированных ошибок прогноза, а также на среднее и стандартное отклонение произведения отстоящих на  $k$  периодов стандартизированных ошибок. Последнее, по сути, эквивалентно проверке наличия

---

<sup>7</sup> См. представленную в Lueckmann et al. (2021) в контексте нахождения апостериорных распределений дискуссию относительно качества работы различного рода симуляционных алгоритмов, в том числе амортизированных нейронных сетей.

<sup>8</sup> См., например, Clark (2011).

<sup>9</sup> Распределение не обязательно должно быть нормальным.

автокорреляции, но не требует корректировки асимптотического распределения ввиду конечности длины временных рядов, поэтому симулируя множество временных рядов, можно не опасаться, что распределение оценок корреляций будет иметь среднее и стандартное отклонение отличные от нуля и единицы.

Стоит отметить два момента относительно этой метрики качества. Во-первых, результаты теста, описанного выше, стоит рассматривать скорее как один из этапов верификации качества модели, и непрохождение его должно служить сигналом о проблемах с качеством нейронной сети. Обратная же ситуация, когда тест проходит, не является гарантией, что модель хорошо работает. К примеру, если модель сценарного прогнозирования плохо обучена и никак не принимает во внимание наличие сценария, а выдает лишь безусловный прогноз, тест будет пройден. Во-вторых, мы не можем применять стандартную формальную проверку гипотез относительно среднего и стандартного отклонения, так как дрейф коэффициентов нейронной сети при ненулевых скоростях обучения (см. Mandt et al. (2017) вносит сравнимый вклад в распределение оценок среднего и стандартного отклонения<sup>10</sup>. Поэтому ниже мы смотрим на эти величины без формального тестирования гипотез.

**Сравнение с моделью-бенчмарком.** Как было сказано выше, прохождение теста на стандартизированных ошибках прогноза является лишь косвенным подтверждением качества обучения модели, так как в числе прочего, он может быть пройден моделями, которые не учитывают часть релевантной информации. Чтобы посмотреть на то, насколько нейронная сеть учитывает историческую информацию, мы оцениваем снизу качество прогнозирования на тестовом датасете.

Такая оценка может быть сделана с помощью модели-бенчмарка. Заметим, что по построению функции потерь, нейронная сеть после оптимизации должна давать наименьшую среднеквадратичную ошибку прогноза (далее – MSFE) для каждой переменной и для каждого горизонта. Более того, средние предельные логарифмы правдоподобия прогнозов (далее – LPS) должны быть наибольшими в классе нормальных распределений. Эти два факта позволяют сделать вывод, что, какую бы модель прогнозирования мы ни построили на тестовых данных, она не должна превосходить нейронную сеть по MSFE и LPS.

Модель-бенчмарк должна выбираться исходя из баланса гибкости и времени обучения. С одной стороны, чем более гибкая модель, тем более строгой будет нижняя граница

---

<sup>10</sup> Вероятно, оценки могут быть улучшены путем применения ансамблей моделей на основе нескольких запусков процедуры обучения либо с применением усреднения на различных итерациях в рамках одной процедуры обучения. Однако этот вопрос лежит вне рамок этой работы.

точности. С другой стороны, оценка нижней границы должна занимать адекватное время. Репрезентативный тестовый набор зачастую содержит тысячи или десятки тысяч наборов временных рядов, поэтому необходимо обучать модель сотни тысяч раз (количество тестовых наборов, умноженное на количество периодов, в которые делается прогноз) для проверки одного прогнозного горизонта для одной переменной.

В данной работе в качестве модели-бенчмарка была выбрана векторная авторегрессия (VAR) с гребневой регуляризацией. Она достаточно легко вычислительно оцениваема и позволяет за адекватное время рассчитывать как условные, так и безусловные прогнозы. При этом VAR-модель в экспериментах ставится в максимально комфортные условия, делая прохождение теста более сложным. Мы начинаем прогнозировать только с 101-го периода, чтобы дать VAR-модели как можно больше информации для обучения коэффициентов (переоцениваются рекуррентно на расширяющемся окне), а также оптимизируем по сетке количество лагов и параметр регуляризации на прогнозных данных.

#### 4. Результаты экспериментов

В данном разделе мы покажем, как предложенный алгоритм работает при построении прогнозов в ABM. Однако перед тем, как переходить к ABM, в которых сравнение с MCMC является вычислительно трудоемким, в подразделе 4.1 мы покажем, как наша процедура и метрики качества ведут себя на примере байесовской AR(1)-регрессии. Затем в подразделах 4.2 и 4.3 будут показаны результаты для условного и безусловного прогнозирования в ABM.

##### 4.1. Проверка концепции

В качестве проверки предложенного алгоритма мы используем байесовскую AR(1)-модель (см. Приложение Б.1). Для оценки модели были сгенерированы 1 010 000 временных рядов длиной 200, из которых 10 000 были взяты в качестве тестовой выборки. Мы выбрали  $T_{min} = 50$ , так как заметили небольшую деградацию качества по сравнению с MCMC для моделей с  $T_{min}$  и  $T_{max}$ , отличающимися на порядок. При этом длины рядов от 50 до 200 соответствуют квартальным данным от 12,5 до 50 лет, что соотносится с длиной рядов, используемых макроэкономистами на практике. Итоговый тренировочный датасет для каждого из горизонтов прогнозирования составил примерно 150 000 000 примеров ( $\sim N(T_{max} - T_{min})$ ), а для теста – 1 500 000. Модель обучается алгоритмом ADAM (см. Kingma and Ba (2014)) 500 000 итераций с размером батча 100 и скоростью обучения  $\epsilon_n$ :

$$\varepsilon_n = \begin{cases} 10^{-4}, & \text{если } n < 3 \times 10^5 \\ 10^{-5}, & \text{если } n \geq 3 \times 10^5 \end{cases}$$

Процесс обучения занимает около пяти часов на видеокарте NVIDIA GeForce RTX 2070 для нейронной сети, написанной на PyTorch (Paszke et al. (2019)).

На рисунке 1 (см. Приложение А) показаны примеры работы нейронной сети, а также результаты прогнозов с помощью МСМС-алгоритма на случайно сгенерированных данных (не использовавшихся при обучении) с длинами выборки 50, 100, 150 и 200 для горизонтов прогнозирования от 1 до 12. На рисунке показано, что прогнозы хорошо аппроксимируются нейронной сетью – как в случаях, когда автокорреляция близка к нулю (верхний график), средней автокорреляции (два средних графика), так и в случаях высокой персистентности (нижний график).

Рисунки 2 и 3 демонстрируют соответственно, как распределены ошибки и произведения ошибок прогноза для различных горизонтов прогнозирования на тестовых данных. Также вверху графиков подписаны средние и стандартные отклонения. Нейронная сеть, аппроксимирующая апостериорное распределение прогнозов байесовской AR(1)-модели проходит тест на стандартизированных ошибках, так как для всех распределений средние и стандартные отклонения близки к 0 и 1. Заметим, что распределения ошибок близки к нормальному, однако, как упоминалось выше, в отличие от преобразования кумулятивных распределений это не обязательное условие нашей проверки<sup>11</sup>.

В таблице 1 представлено сравнение MSFE- и LPS-ошибок нейронной сети относительно AR(1)-модели, которая является процессом порождения данных для каждого ряда. Можно заметить, что на всех горизонтах прогнозирования нейронная сеть значительно превосходит AR(1)-модель. Это может показаться контринтуитивным, однако в этом нет ничего странного, так как байесовская модель знает чуть больше информации. В частности, в дополнение к процессу порождения данных она знает информацию о том, как распределены параметры тестовых датасетов.

Подводя итог, можно сказать, что нейронная сеть хорошо справляется с нашим «игрушечным» примером. Характеристики апостериорных распределений прогнозов практически совпадают с характеристиками прогнозов на базе МСМС-алгоритма, что является главной мерой качества обучения. Более того, как и ожидалось, такая нейронная сеть

---

<sup>11</sup> Как мы обнаружили, для безусловного прогноза в АВМ распределения не являются нормальными (графики не представлены ради экономии места).

легко проходит косвенные тесты оценки качества, которые станут для нас основными при работе с АВМ.

#### 4.2. Безусловный прогноз

Для демонстрации свойств предложенного алгоритма для АВМ, мы построили упрощенную версию модели из Delli Gati et al. (2011), которая содержит 50 С-фирм, 500 потребителей, государственный сектор и банк (см. Приложение Б.2). Во всех примерах для безусловного и условного прогнозов мы генерируем 510 000 наборов данных (10 000 из которых – тестовые данные) длиной 300, из которых 100 удаляются как начальный период. Как мы увидели из экспериментов, такой набор является минимальным для предотвращения переобучения в модели с четырьмя наблюдаемыми переменными (см. ниже) без регуляризации. Конечно же, процедуры регуляризации, такие как ранняя остановка (early stopping), L1/L2-регуляризация или прореживание (dropout<sup>12</sup>), могут быть использованы для сокращения количества симуляций, однако вопрос подбора корректной регуляризации и ее параметров лежит вне рамок этой работы.

В отличие от AR(1), в случае АВМ симуляции данных занимают большую часть времени по сравнению с обучением нейронной сети. Чтобы ускорить процедуру симуляции, мы векторизовали модель, что позволило генерировать 10 000 симуляций одновременно в одном ноутбуке в Python. Мы запускали параллельно пять ноутбуков, что позволило генерировать 50 000 симуляций примерно за три часа на CPU (Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 16GB RAM). В итоге финальный датасет был сгенерирован за 30 часов. Обучение сети для каждого из трех наборов переменных, описанных ниже, занимает около пяти часов и проводилось с аналогичными байесовской AR(1)-модели гиперпараметрами.

Мы оценили модель для двух, трех и четырех наблюдаемых переменных:

$$X_2 = \{10 \log(1 + P), 10U\}$$

$$X_3 = \{10 \log(1 + P), 10U, 10 \log(1 + C)\}$$

$$X_4 = \{10 \log(1 + P), 10U, 10 \log(1 + C), \log(1 + L)\},$$

где  $X_2$ ,  $X_3$ ,  $X_4$  – обозначения датасетов из двух, трех и четырех переменных соответственно;  $P$  – средний уровень цен С-фирм;  $U$  – уровень безработицы;  $C$  – уровень потребления;  $L$  – объем кредитов.

Для переменных цен, потребления и кредитов мы сделали близкое к логарифмическому преобразование, которое часто используется при прогнозировании

---

<sup>12</sup> См. Srivastava et al. (2014).

макроэкономических переменных, однако позволяет нам обучать модель без дополнительных фильтров вырожденных случаев, когда значения в данных равны нулю. Также данные были шкалированы, чтобы стандартные отклонения переменных в отдельных наборах данных в среднем имели порядок 1.

В таблице 2 представлены средние и стандартные отклонения распределений ошибок прогноза для моделей, обученных на датасетах из двух, трех и четырех переменных. Можно увидеть, что во всех случаях средние близки к 0, а стандартные отклонения – к 1, что означает, что обученные нейронные сети проходят первую часть теста, предложенного в разделе 3. При этом, как упоминалось при описании теста, отклонения в несколько сотых вызваны небольшой осцилляцией коэффициентов при не стремящихся к нулю скоростях обучения. При расчете стандартных отклонений для произведения стандартизированных ошибок прогноза мы усекали выборку, удаляя из расчетов точки, которые по модулю больше 30 (порядка 100 из 1 500 000 тестовых точек). Это делалось ввиду того, что стандартные отклонения произведения ошибок – это четвертые моменты распределения ошибок, и наличие выбросов может значительно влиять на оценки. Такие выбросы связаны с очень экстремальными единичными симуляциями для безработицы и кредитов. Примеры этих симуляций изображены на рисунке 4. За исключением таких случаев, как показывает таблица 3, характеристики близки к идеальным и отличаются не более чем на несколько сотых.

Нейронная сеть, аппроксимирующая АВМ, демонстрирует хорошие прогнозные свойства по сравнению с моделью-бенчмарком, которой в данной работе является VAR-модель. В таблице 4 показано сравнение в терминах MSFE и LPS, из которого можно заметить, что для всех моделей и горизонтов прогнозирования нейронная сеть не уступает (статистически значимо) VAR-модели. Более того она практически всегда (в 212 из 216 случаях) превосходит бенчмарк на 1%-ном уровне значимости, что вместе с результатами прошлой проверки позволяет надеяться на достаточно хорошую аппроксимацию апостериорных прогнозов АВМ. Примеры самих прогнозов на случайно сгенерированных данных для АВМ с четырьмя переменными представлены на рисунке 5.

### **4.3. Условный прогноз**

Чтобы проиллюстрировать свойства нейронных сетей при амортизации задач условного прогнозирования в АВМ, используется та же модель, что и в подразделе 4.2. Мы обучаем нейронную сеть для прогнозирования потребления на 12 периодов при условии



безработицы<sup>13</sup>, известной на всем прогнозном горизонте. Эти переменные сильно связаны между собой (средняя корреляция по модулю больше 0,5), поэтому безработица будет достаточно информативной относительно потребления и прогноз должен сильно отличаться от безусловного. Это позволяет максимально наглядно показать, как работает Алгоритм 2.

Пример работы обученной нейронной сети на тех же случайных данных для потребления и безработицы, что и для рисунка 5, показан на рисунке 6. Можно увидеть, что результаты условного прогноза намного более точные и с более узкими интервалами, чем безусловные прогнозы из модели на основе четырех переменных, что служит косвенным подтверждением того, что нейронная сеть хорошо обучилась. Также об этом сигнализируют и метрики качества, предложенные в разделе 3. Как показывают рисунки 7 и 8 и значения средних и стандартных отклонений изображенных на них распределений (близкие к 0 и 1) для ошибок прогнозов и их произведений, нейронная сеть проходит проверку на основе стандартизированных ошибок прогноза.

В качестве модели бенчмарка для второго теста из раздела 3, как и ранее, была оценена VAR-модель, условные прогнозы из которой строились при помощи сглаживания Калмана (см. Durbin and Koopman (2002)). Для всех горизонтов прогнозирования нейронная сеть обходит VAR-модель (см. таблицу 5), что, как и результаты, описанные выше, является косвенным доказательством обучения нейронной сети.

## 5. Дискуссия

В этом разделе мы обсудим вопросы, которые считаем важными в контексте будущего применения и развития описанных алгоритмов. В частности, это вопросы, которые связаны с 1) оценкой качества алгоритмов на основе косвенных метрик, 2) решением задач сценарного прогнозирования с учетом сценариев, построенных на ненаблюдаемых переменных, 3) использованием алгоритма для АВМ на микроданных, 4) построением модели при фиксированном вычислительном бюджете на симуляции.

**Использование косвенных метрик качества.** Результаты прошлого раздела позволяют сделать вывод, что предложенная амортизированная процедура прогнозирования в АВМ на основе нейронных сетей показывает адекватные результаты. Тем не менее оценка качества как для условного, так и для безусловного прогноза основана на косвенных метриках, что само по себе хоть и отсекает неадекватные результаты, не может служить стопроцентной гарантией качественной аппроксимации апостериорного распределения

---

<sup>13</sup> Переменные преобразованы так же, как и в подразделе 4.2.

прогнозов. Модель-бенчмарк служит лишь некой нижней границей качества прогноза и может быть недостаточно жесткой, а калибровка на основе стандартизированных прогнозов имеет проблемы, схожие с валидацией параметров байесовских моделей (см. Cook et al. (2006), Talts et al. (2018)), в том смысле, что эта проверка может быть пройдена даже моделями, характеристики распределений которых далеки от характеристик апостериорного распределения<sup>14</sup>. Более того, в случае ненулевых скоростей обучения, формально последняя процедура не является строгой из-за того, что она не основана на статистических тестах, так как ошибки, вносимые осцилляцией сравнимы со статистической погрешностью или даже превосходят ее. Использование косвенных метрик создает риск того, что прогнозы из недообученных моделей (которые отличаются от апостериорных) будут приняты как истинные прогнозы из АВМ, что может негативно сказаться на принимаемых на их основе решениях<sup>15</sup>. Поэтому мы считаем усовершенствование процедур тестирования качества аппроксимации прогнозов одним из важнейших направлений для дальнейшего развития прогнозирования в АВМ и среди прочего видим большой потенциал для тестирования амортизированных алгоритмов в формализации тестов на основе стандартизированных ошибок прогноза и построении тестов на зависимость ошибок прогноза от прошлых данных.

**Сценарное прогнозирование на основе ненаблюдаемых переменных.** Мы также верим, что, несмотря на то, что нам не удалось добиться сходимости к оптимуму алгоритма для сценарного анализа, зависящего не только от наблюдаемых переменных, но и от ненаблюдаемых (см. Приложение В), будущие исследования помогут подобрать архитектуру нейронных сетей и алгоритмы обучения, которые будут более успешны и помогут решать более широкий круг задач, чем просто безусловное и условное прогнозирование.

**Применение алгоритма при работе с микроданными.** Помимо оценки на агрегированных данных, на практике при работе с АВМ часто возникает необходимость работы с моделями, которые соотносятся с историческими микроданными или их распределениями (см. Nommes et al. (2022)). Описанные в разделе 3 Алгоритмы 1 и 2 могут быть использованы и для таких задач путем добавления соответствующих переменных в набор наблюдаемых. Для случая микроданных – это сами микроданные<sup>16</sup>, а в случае

---

<sup>14</sup> К примеру, этот тест может быть пройден априорным распределением.

<sup>15</sup> Отметим, что такие риски связаны не только с нашими алгоритмами, а применимы и к другим методам построения приближенных прогнозов в структурных моделях.

<sup>16</sup> Изменения структуры нейронной сети, которые проводят на первых слоях операции агрегации или учитывают более конкретно структуру данных, например граф взаимосвязей (см. Dyer et al. (2022b)), могут понадобиться для уменьшения времени и повышения качества обучения.

распределений могут быть взяты достаточные статистики (если доступны) или же небольшой набор характеристик, описывающий свойства распределения<sup>17</sup>.

**Фиксированный бюджет на симуляции.** Время симуляций в АВМ может быть велико и занимать до нескольких минут для одной симуляции. Это может приводить к тому, что время, необходимое для генерации нужного количества искусственных данных, может быть непомерно велико. В литературе по SBI есть множество многоаундовых способов, которые работают в таких ограничениях (см. Papamakarios and Murray (2016), Lueckmann et al. (2017), Greenberg, Nonnenmacher and Macke (2019)). Мы советуем использовать идею о прямоугольном обрезании априорных распределений параметров из Miller et al. (2021), которая в случае расширения нашего алгоритма может быть использована, чтобы сконцентрироваться на симуляциях в регионах, где параметры более или менее соответствуют наблюдаемым данным. Несмотря на то что, как и другие алгоритмы, этот подход не обладает свойством амортизации, он тем не менее амортизируем локально в определенных границах. Четкие границы амортизации позволяют оценивать его локальные свойства с использованием описанных ранее метрик качества. Более того, как и для других алгоритмов, локальная амортизация в пространстве параметров позволяет надеяться, что при поступлении новых данных (дополнительных точек во временном разрезе) старые симуляции могут быть переиспользованы, так как апостериорное распределение параметров обычно не сильно изменяется при поступлении нескольких новых точек.

## 6. Заключение

В данной работе мы описали алгоритм прогнозирования для АВМ моделей на основе амортизированных нейронных сетей, который после предварительного обучения позволяет практически мгновенно оценивать характеристики апостериорного распределения прогнозов. Результаты экспериментов для AR(1) и верификация на основе косвенных метрик качества для АВМ показывают, что алгоритм демонстрирует хорошие свойства. Мы надеемся, что наша работа послужит толчком для развития процедур прогнозирования в АВМ на основе нейронных сетей, а также поможет в будущем сделать прогнозирование в АВМ рутинной процедурой.

---

<sup>17</sup> См., например, Chang et al. (2021) для представления динамики распределений через набор характеристик в контексте функциональных VAR-моделей.

**Список литературы**

- Andrieu, C., A. Doucet and R. Holenstein (2010). Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society Series B*, 72(3): 269–342.
- Axtell, R.L. and J.D. Farmer (2022). Agent Based Modeling in Economics and Finance: Past, Present and Future. *INET Oxford Working Paper*, 2022–10.
- Chang, M., Chen, X. and F. Schorfheide (2021). Heterogeneity and Aggregate Fluctuations. *NBER Working Paper*, 28853.
- Chopin, N., P.E. Jacob and O. Papaspiliopoulos (2012). SMC<sup>2</sup>: An Efficient Algorithm for Sequential Analysis of State Space Models. *Journal of the Royal Statistical Society Series B*, 75(3): 397–426.
- Clark, T. (2011). Real-time Density Forecasts from Bayesian Vector Autoregressions with Stochastic Volatility. *Journal of Business and Economic Statistics*, 29: 327–341.
- Cook, S.R., Gelman, A. and D.B. Rubin (2006). Validation of Software for Bayesian Models Using Posterior Quantiles. *Journal of Computational and Graphical Statistics*, 15(3): 675–692.
- Cranmer, K., J. Brehmer and G. Louppe (2020). The Frontier of Simulation-Based Inference. *Proceedings of the National Academy of Sciences*, 117(48): 30055–30062.
- Delli Gatti, D., Desiderio, S., Gaffeo, E., Cirillo, P. and M. Gallegati (2011). Macroeconomics from the Bottom-up. *Springer Science and Business Media*.
- Delli Gatti, D. and J. Grazzini (2020). Rising to the Challenge: Bayesian Estimation and Forecasting Techniques for Macroeconomic Agent Based Models. *Journal of Economic Behavior and Organization*, 178: 875–902.
- Durbin, J. and S.J. Koopman (2002). A Simple and Efficient Simulation Smoother for State Space Time Series Analysis. *Biometrika*, 89(3): 603–615.
- Dyer, J., Cannon, P., Farmer, J.D. and S.M. Schmon (2022a). Black-box Bayesian Inference for Economic Agent-Based Models. *arXiv: 2202.00625*.
- Dyer, J., Cannon, P., Farmer, J.D. and S.M. Schmon (2022b). Calibrating Agent-based Models to Microdata with Graph Neural Networks. *International Conference on Machine Learning*.
- Fagiolo, G. and A. Roventini (2017). Macroeconomic Policy in DSGE and Agent-Based Models Redux: New Developments and Challenges Ahead. *Journal of Artificial Societies and Social Simulations*, 20(1).
- Finn, C. and S. Levine (2019). Meta-Learning: from Few-Shot Learning to Rapid Reinforcement Learning. *The International Conference on Machine Learning*, Tutorial.

Garnelo, M., Rosenbaum, D., Maddison, C.J., Romalho, T., Saxton, D., Shanahan, M., Teh, Y.W., Rezende, D.J. and S.M. Ali Eslami (2018). Conditional Neural Processes. *In Proceedings of the 35<sup>th</sup> International Conference on Machine Learning*.

Gordon, N.J., Salmond, D.J. and A.F.M. Smith (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2): 107–113.

Greenberg, D., M. Nonnenmacher, and J. Macke (2019). Automatic Posterior Transformation for Likelihood-Free Inference. *In Proceedings of the 36<sup>th</sup> International Conference on Machine Learning*.

Haldane, A.G. and A.E. Turrell (2018). Drawing on Different Disciplines: Macroeconomic Agent-Based Models. *Journal of Evolutionary Economics*, 29: 39–66.

Harrison, J., A. Sharma and M. Pavone (2020). Meta-Learning Priors for Efficient Online Bayesian Regression. *Algorithmic Foundations of Robotics XIII*, 14: 318–337.

He, K., Zhang, X., Ren, S. and J. Sun (2015). Deep Residual Learning for Image Recognition. *arXiv: 1512.03385*.

Hommes, C., He, M., Poledna, S., Siqueira, M. and Y.Zhang (2022). CANVAS: A Canadian Behavioral Agent-Based Model. *Bank of Canada Staff Working Paper*, 2022–51.

Khabibullin, R. and S. Seleznev (2022). Fast Estimation of Bayesian State Space Models Using Amortized Simulation-Based Inference. *Bank of Russia Working Paper Series*, № 104.

Kingma, D.P. and J. Ba (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Lueckmann, J.-M., J. Boelts, D.S. Greenberg, P.J. Gonçalves and J.H. Macke (2021). Benchmarking Simulation-Based Inference. *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Lueckmann, J.-M., P.J. Goncalves, G. Bassetto, K. Öcal, M. Nonnenmacher and J.H. Macke (2017). Flexible Statistical Inference for Mechanistic Models of Neural Dynamics. *In Advances in Neural Information Processing Systems*, 30: 1289–1299.

Lux, T. (2018). Estimation of Agent-Based Models Using Sequential Monte Carlo Methods. *Journal of Economic Dynamics and Control*, 91: 391–408.

Mandt, S., M.D. Hoffman and D.M. Blei (2017). Stochastic Gradient Descent as Approximate Bayesian Inference. *Journal of Machine Learning Research*, 18: 1–35.

Miller, B.K., Cole, A., Forre, P., Louppe, G. and C. Weniger (2021). Truncated Marginal Neural Ratio Estimation. *Neural Information Processing Systems*.

Papamakarios, G. and I. Murray (2016). Fast  $\epsilon$ -free Inference of Simulation Models with Bayesian Conditional Density Estimation. *In Advances in Neural Information Processing Systems*, 29: 1028–1036.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *In Advances in Neural Information Processing Systems*, 32: 8024–8035.

Poledna, S., Miess, M.G., Hommes, C. and K. Rabitsch (2023). Economic Forecasting with an Agent-Based Model. *European Economic Review*, 151.

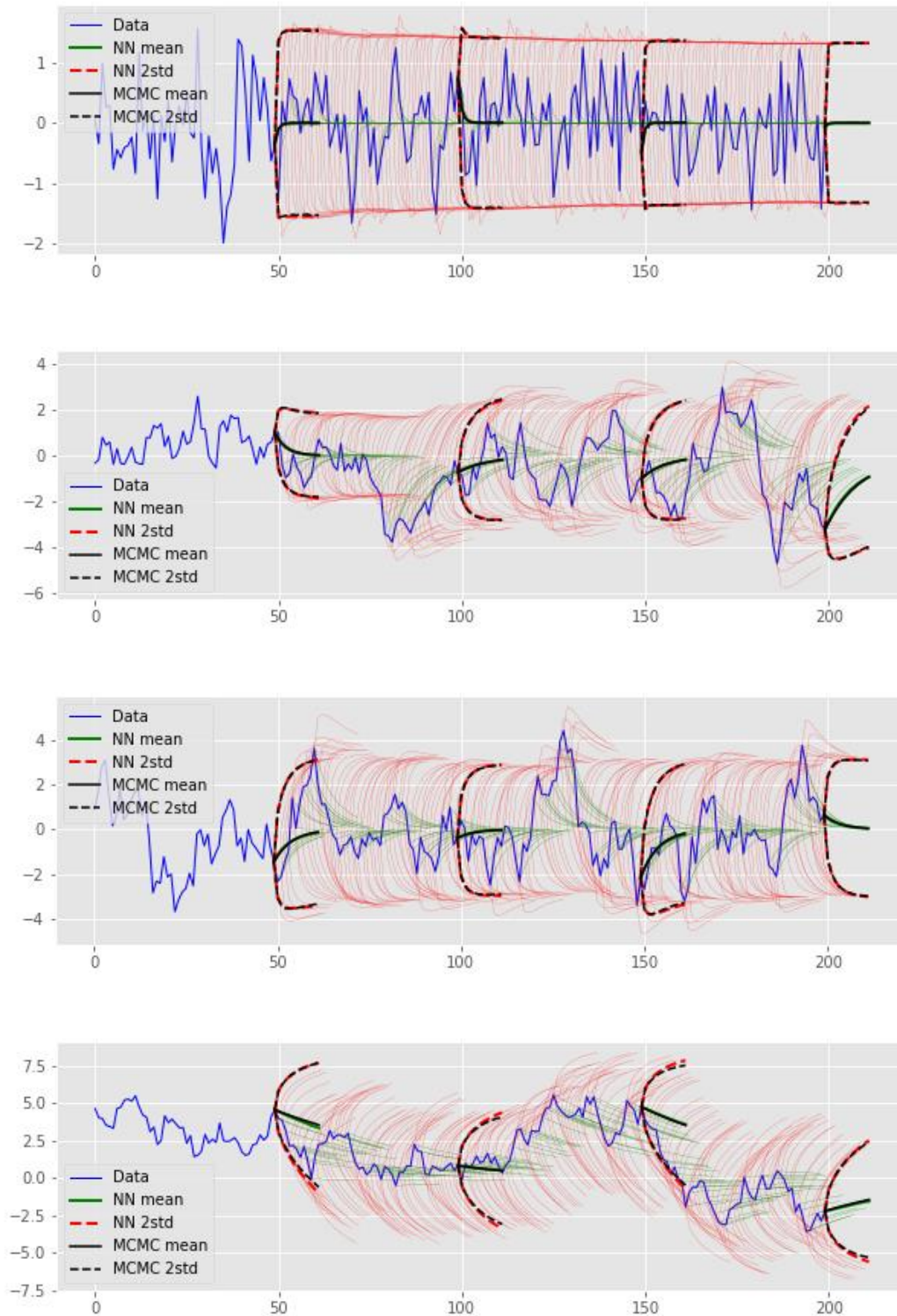
Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A. and A. Gelman (2018). Validating Bayesian Inference Algorithms with Simulation Based Calibration. *arXiv: 1804.06788v2*.

Van der Vaart, A.W. (2000). *Asymptotic Statistics*. Cambridge University Press.

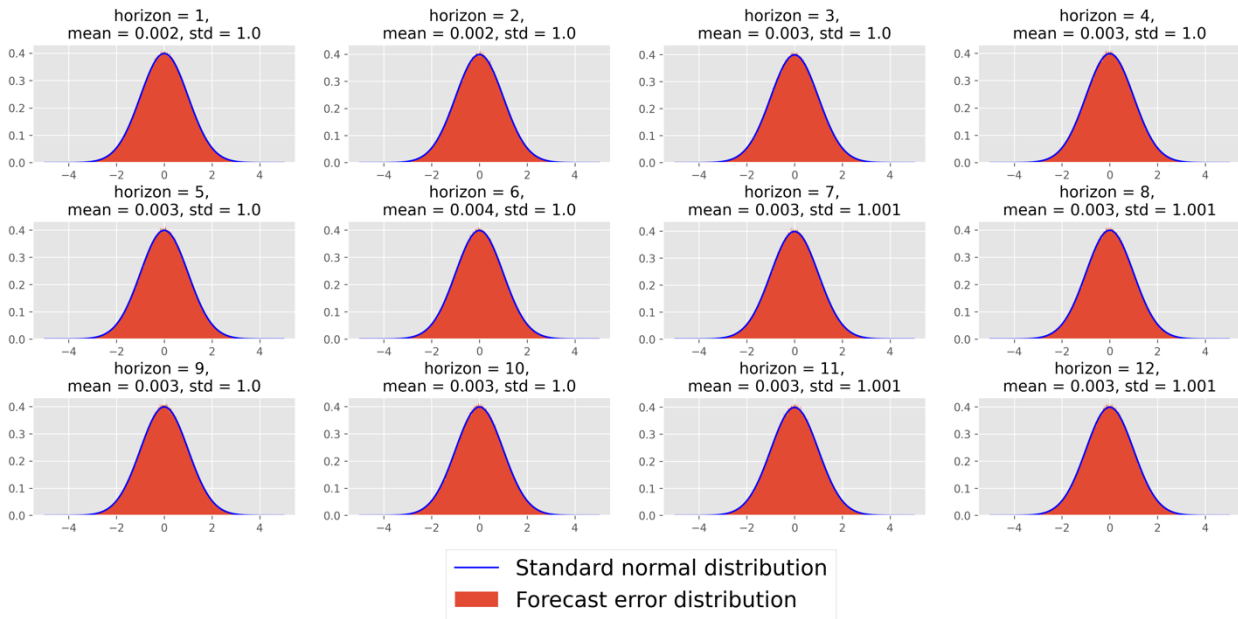
## Приложение А. Таблицы и рисунки

Рисунок 1. Сравнение прогнозов с использованием нейронной сети с прогнозами MCMC-алгоритма для байесовской AR(1)-модели

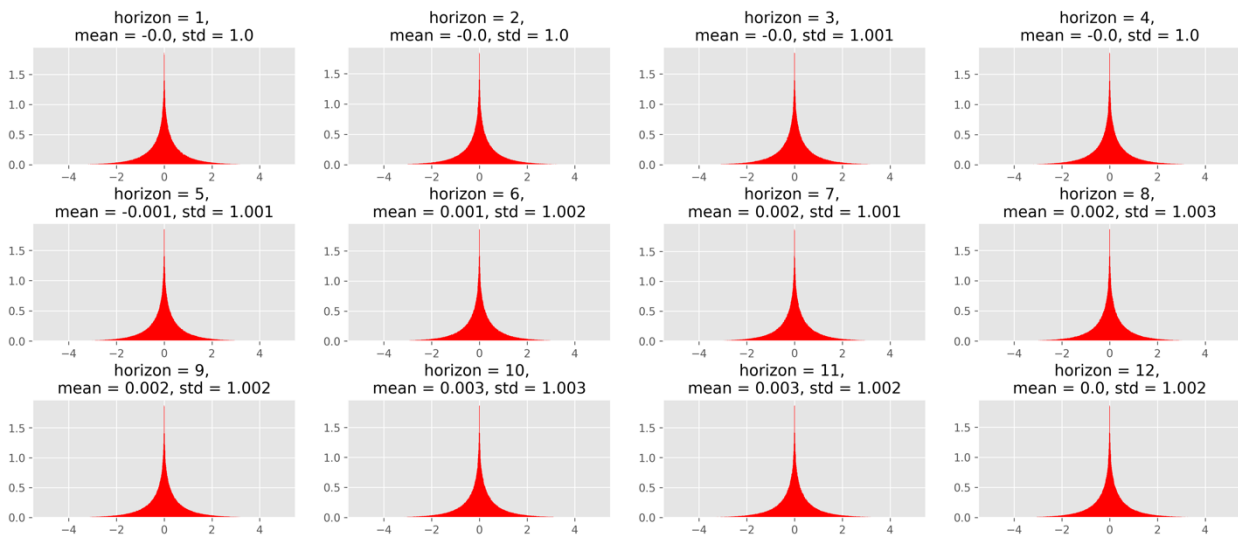




**Рисунок 2. Распределения стандартизированных ошибок прогнозов для  
байесовской AR(1)-модели**



**Рисунок 3. Распределения произведений стандартизированных ошибок прогнозов для  
байесовской AR(1)-модели**



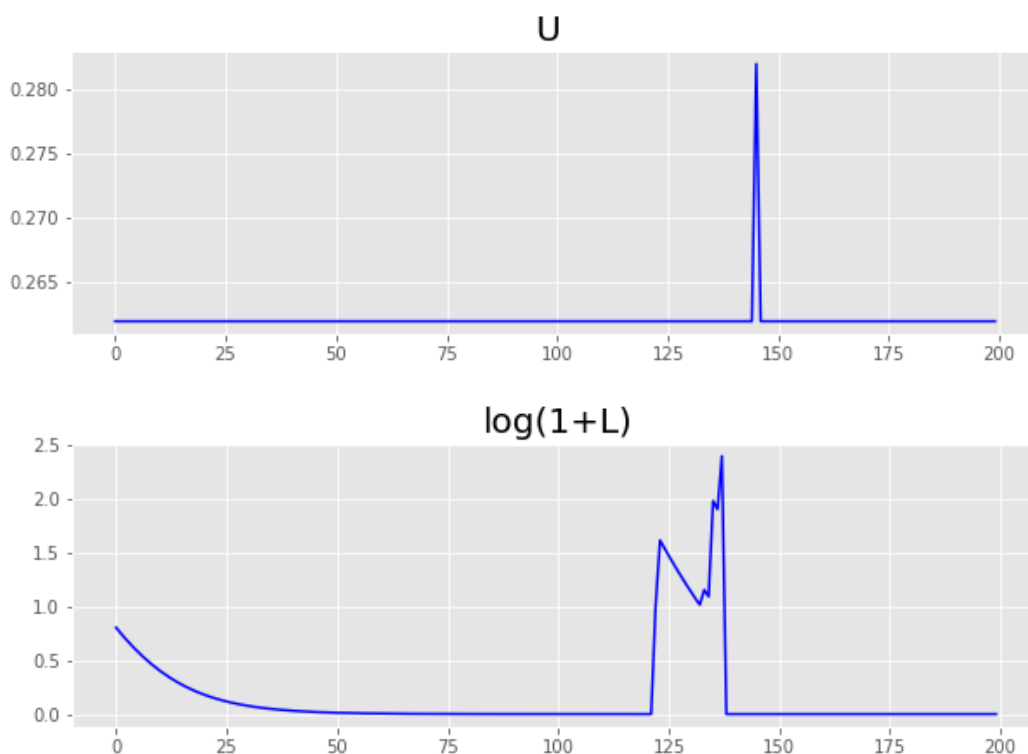


**Таблица 1. Сравнение прогнозных свойств AR-модели и нейронной сети, аппроксимирующей байесовскую AR-модель (MSFE – отношение среднеквадратичных ошибок AR и нейронной сети, LPS – разность средних логарифмов правдоподобия прогнозов AR и нейронной сети)**

Горизонт прогнозирования	MSFE	LPS
1	1,009***	-0,005***
2	1,015***	-0,008***
3	1,02***	-0,009***
4	1,025***	-0,011***
5	1,029***	-0,012***
6	1,033***	-0,013***
7	1,038***	-0,014***
8	1,041***	-0,015***
9	1,045***	-0,016***
10	1,049***	-0,016***
11	1,052***	-0,017***
12	1,055***	-0,018***

Уровень значимости: \*(10%), \*\*(5%), \*\*\*(1%)

**Рисунок 4. Примеры экстремальных симуляций для безработицы и кредитов в АВМ (на графиках изображены разные симуляции)**



**Таблица 2. Верификация распределений стандартизированных ошибок прогнозов для  
безусловного прогноза в АВМ**

Горизонт прогнозирования	2 переменные				3 переменные					
	ИПЦ		Безработица		ИПЦ		Безработица		Потребление	
	mean	std	mean	std	mean	std	mean	std	mean	std
1	-0,009	1	-0,004	0,997	-0,006	0,991	0,004	1,006	0,017	1,002
2	-0,014	1,002	-0,008	0,998	-0,013	0,992	0,005	1,004	0,013	1,001
3	-0,017	1,003	-0,008	0,999	-0,015	0,992	0,007	1,003	0,014	1,001
4	-0,018	1,003	-0,009	0,999	-0,017	0,992	0,008	1,004	0,014	1
5	-0,02	1,003	-0,01	0,999	-0,018	0,992	0,008	1,003	0,016	1
6	-0,021	1,003	-0,009	0,999	-0,019	0,993	0,007	1,005	0,015	1
7	-0,022	1,002	-0,009	0,998	-0,02	0,992	0,007	1,006	0,016	1
8	-0,022	1,002	-0,009	0,998	-0,02	0,993	0,006	1,004	0,016	1
9	-0,023	1,002	-0,009	0,997	-0,021	0,993	0,006	1	0,017	1
10	-0,024	1,002	-0,009	0,997	-0,021	0,993	0,006	0,999	0,017	1
11	-0,024	1,001	-0,008	0,996	-0,022	0,993	0,006	0,998	0,018	1,001
12	-0,024	1,001	-0,009	0,995	-0,023	0,993	0,005	0,996	0,017	1,001

Горизонт прогнозирования	4 переменные							
	ИПЦ		Безработица		Потребление		Кредиты	
	mean	std	mean	std	mean	std	mean	std
1	0,005	0,99	-0,001	0,989	-0,006	0,997	0	0,993
2	0	0,99	0,002	0,995	-0,006	0,998	0	0,999
3	0	0,99	0,002	0,992	-0,007	0,998	0,002	1,001
4	0,002	0,991	0,002	1,005	-0,007	0,997	0,001	0,996
5	0,004	0,991	0,003	1,005	-0,008	0,997	-0,001	0,995
6	0,004	0,991	0,004	1,002	-0,008	0,997	-0,001	0,995
7	0,005	0,991	0,005	1,002	-0,009	0,997	-0,001	0,993
8	0,004	0,991	0,005	1,001	-0,008	0,998	-0,002	0,995
9	0,004	0,992	0,006	1,002	-0,009	0,998	-0,002	0,993
10	0,004	0,992	0,006	0,999	-0,008	0,998	-0,003	0,993
11	0,004	0,992	0,007	0,997	-0,009	0,998	-0,002	0,991
12	0,004	0,992	0,006	0,996	-0,009	0,999	-0,003	0,991

**Таблица 3. Верификация распределений произведений стандартизированных ошибок прогнозов для безусловного прогноза в АВМ**

Горизонт прогнозирования	2 переменные				3 переменные					
	ИПЦ		Безработица		ИПЦ		Безработица		Потребление	
	mean	std	mean	std	mean	std	mean	std	mean	std
1	0,005	1,028	0	0,975	0,015	1,013	-0,011	0,998	-0,006	1,017
2	0,008	1,018	0,003	0,971	0,007	1,002	-0,005	0,983	0	1,011
3	0,007	1,016	0,002	0,971	0,004	0,997	-0,004	0,979	-0,001	1,011
4	0,004	1,018	0,003	0,974	0,003	0,997	-0,003	0,979	-0,001	1,005
5	0,004	1,017	0,001	0,975	0,001	0,997	-0,004	0,976	-0,004	1,006
6	0,003	1,018	0,001	0,973	-0,001	0,998	-0,005	0,974	-0,005	1,004
7	0,002	1,018	0	0,976	-0,003	1	-0,005	0,974	-0,005	1,003
8	0,002	1,019	0	0,979	-0,004	1,002	-0,005	0,976	-0,006	1,004
9	0,002	1,019	-0,001	0,982	-0,006	1,003	-0,006	0,979	-0,007	1,007
10	0,002	1,02	-0,004	0,982	-0,008	1,004	-0,009	0,978	-0,006	1,009
11	0,002	1,017	-0,005	0,983	-0,009	1,005	-0,01	0,979	-0,009	1,01
12	0,002	1,016	-0,005	0,988	-0,009	1,004	-0,011	0,981	-0,008	1,009

Горизонт прогнозирования	4 переменные							
	ИПЦ		Безработица		Потребление		Кредиты	
	mean	std	mean	std	mean	std	mean	std
1	0,02	1,033	0	1,015	-0,014	1,039	0,014	1,014
2	0,016	1,013	0,004	1,012	-0,005	1,025	0,001	1,013
3	0,009	1,007	0,001	1,009	-0,005	1,022	0	1,007
4	0,008	1,005	0	1	-0,003	1,012	0	1
5	0,007	1,005	-0,002	1,001	-0,005	1,012	0	1,002
6	0,005	1,006	-0,004	0,995	-0,007	1,008	0,001	1,008
7	0,002	1,009	-0,007	0,996	-0,008	1,008	-0,001	1,011
8	0	1,011	-0,007	0,994	-0,008	1,007	-0,001	1,014
9	-0,001	1,014	-0,01	1	-0,01	1,011	-0,003	1,014
10	-0,004	1,015	-0,013	0,996	-0,009	1,013	-0,003	1,017
11	-0,005	1,014	-0,015	0,996	-0,013	1,013	-0,006	1,02
12	-0,006	1,014	-0,018	0,999	-0,013	1,014	-0,007	1,026

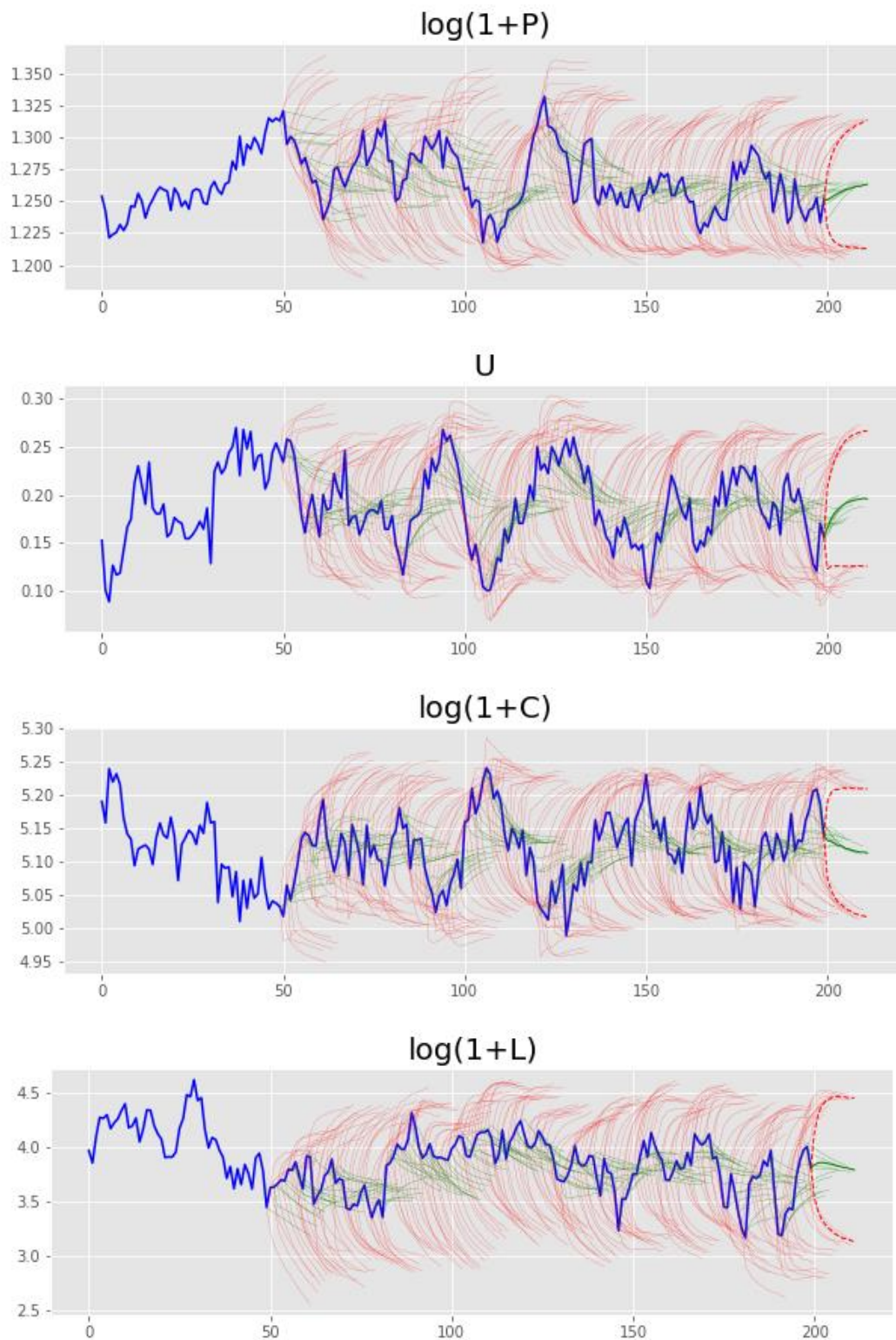
**Таблица 4. Сравнение прогнозных свойств VAR-модели и нейронной сети, аппроксимирующей ABM (MSFE – отношение среднеквадратичных ошибок VAR и нейронной сети, LPS – разность средних логарифмов правдоподобия прогнозов VAR и нейронной сети)**

Горизонт прогнозирования	2 переменные				3 переменные					
	ИПЦ		Безработица		ИПЦ		Безработица		Потребление	
	MSFE	LPS	MSFE	LPS	MSFE	LPS	MSFE	LPS	MSFE	LPS
1	1,05***	-0,17***	1,09***	-0,58***	1,06***	-0,11***	1,11***	-0,18***	1,02**	-0,11**
2	1,08***	-0,15***	1,11***	-0,55***	1,08***	-0,16***	1,13***	-0,17***	1,03***	-0,08***
3	1,09***	-0,18***	1,12***	-0,55***	1,1***	-0,2***	1,14***	-0,18***	1,05***	-0,04***
4	1,1***	-0,21***	1,14***	-0,57***	1,11***	-0,23***	1,16***	-0,19***	1,06***	-0,05***
5	1,11***	-0,25***	1,16***	-0,59***	1,11***	-0,25***	1,17***	-0,2***	1,08***	-0,06***
6	1,11***	-0,27***	1,17***	-0,6***	1,11***	-0,27***	1,18***	-0,2***	1,09***	-0,06***
7	1,11***	-0,29***	1,18***	-0,62***	1,12***	-0,28***	1,19***	-0,21***	1,1***	-0,07***
8	1,11***	-0,31***	1,19***	-0,64***	1,12***	-0,28***	1,2***	-0,22***	1,1***	-0,07***
9	1,12***	-0,33***	1,2***	-0,66***	1,12***	-0,28***	1,2***	-0,22***	1,11***	-0,07***
10	1,12***	-0,34***	1,2***	-0,68***	1,13***	-0,28***	1,2***	-0,23***	1,11***	-0,07***
11	1,12***	-0,35***	1,2***	-0,7***	1,13***	-0,28***	1,21***	-0,24***	1,11***	-0,08***
12	1,13***	-0,37***	1,21***	-0,7***	1,13***	-0,28***	1,21***	-0,24***	1,12***	-0,08***

Горизонт прогнозирования	4 переменные							
	ИПЦ		Безработица		Потребление		Кредиты	
	MSFE	LPS	MSFE	LPS	MSFE	LPS	MSFE	LPS
1	1,07***	-0,11***	1,09***	-0,33***	0,980	-0,09**	1,35***	-0,96***
2	1,09***	-0,16***	1,1***	-0,3***	1,06***	-0,07***	1,24***	-0,93***
3	1,12***	-0,21***	1,12***	-0,29***	1,08***	-0,06***	1,25***	-1,02***
4	1,13***	-0,25***	1,14***	-0,27***	1,11***	-0,07***	1,28***	-1,12***
5	1,14***	-0,27***	1,15***	-0,26***	1,12***	-0,07***	1,32***	-1,2***
6	1,15***	-0,28***	1,16***	-0,26***	1,14***	-0,08***	1,35***	-1,28***
7	1,16***	-0,28***	1,17***	-0,25***	1,16***	-0,08***	1,39***	-1,35***
8	1,17***	-0,29***	1,18***	-0,25***	1,19***	-0,08***	1,42***	-1,41***
9	1,18***	-0,29***	1,18***	-0,25***	1,22***	-0,09***	1,45***	-1,47***
10	1,19***	-0,29***	1,19***	-0,26***	1,27***	-0,09***	1,48***	-1,52***
11	1,2***	-0,29***	1,19***	-0,26***	1,28***	-0,1***	1,5***	-1,57***
12	1,21***	-0,29***	1,19***	-0,27***	1,29***	-0,1***	1,53***	-1,62***

Уровень значимости: \*(10%), \*\*(5%), \*\*\*(1%)

Рисунок 5. Прогнозы с использованием нейронной сети для АВМ с 4 переменными

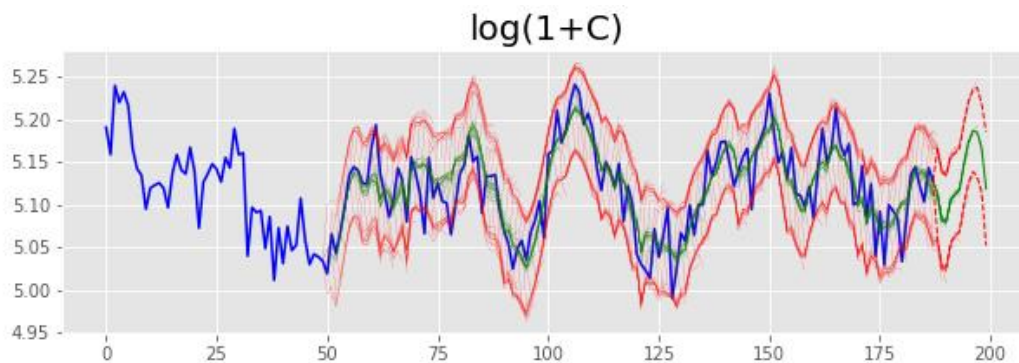


**Таблица 5. Сравнение прогнозных свойств VAR-модели и нейронной сети, аппроксимирующей условный прогноз из ABM (MSFE – отношение среднеквадратичных ошибок VAR и нейронной сети, LPS – разность средних логарифмов правдоподобия прогнозов VAR и нейронной сети)**

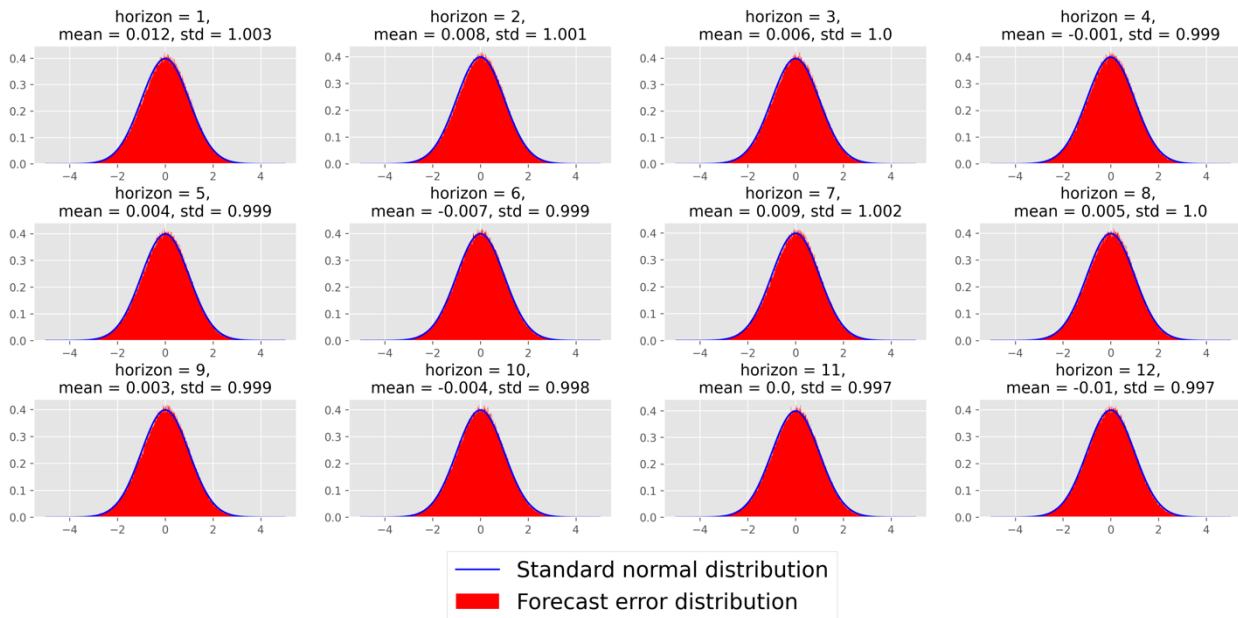
Горизонт прогнозирования	MSFE	LPS
1	1,231***	-0,242***
2	1,276***	-0,235***
3	1,318***	-0,199***
4	1,349***	-0,216***
5	1,38***	-0,236***
6	1,379***	-0,234***
7	1,389***	-0,24***
8	1,394***	-0,247***
9	1,392***	-0,254***
10	1,405***	-0,26***
11	1,374***	-0,276***
12	1,284***	-0,215***

Уровень значимости: \*(10%), \*\*(5%), \*\*\*(1%)

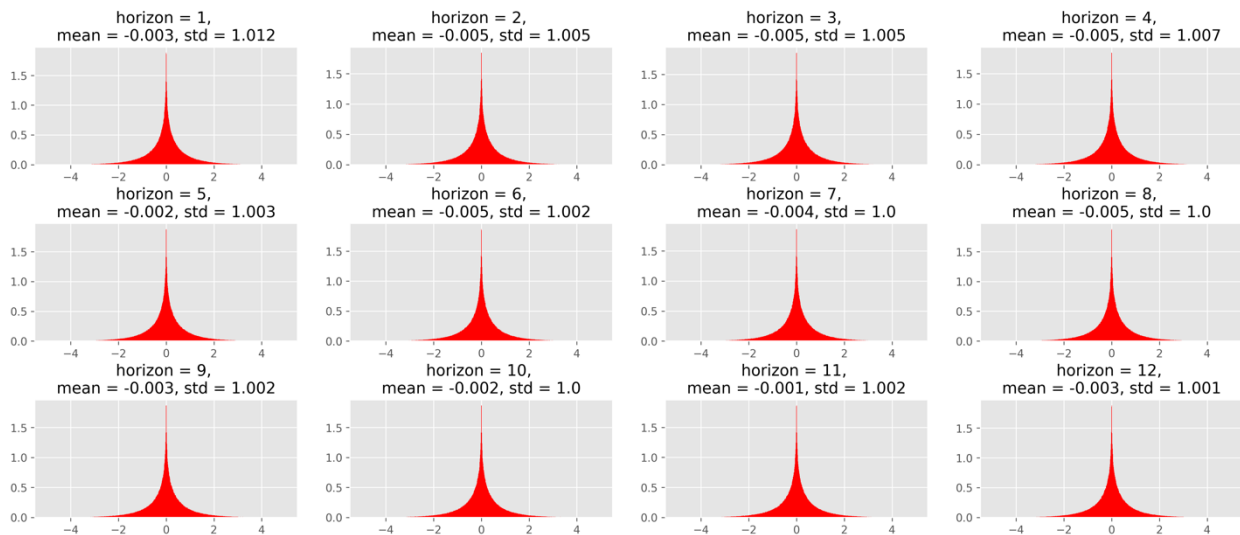
**Рисунок 6. Условный прогноз с использованием нейронной сети для ABM**



**Рисунок 7. Распределения стандартизированных ошибок прогнозов для условного прогноза из АВМ**



**Рисунок 8. Распределения произведений стандартизированных ошибок прогнозов для условного прогноза из АВМ**



## Приложение Б. Модели

### Б.1. Байесовская AR(1)-модель

Априорное распределение:

$$\sigma^2 \sim IG(3; 1), \quad \rho \sim N(0; \sigma), \quad \text{при условии } 0 \leq \rho < 1$$

где  $\sigma$  – стандартное отклонение шоков;  $\rho$  – коэффициент авторегрессии;  $IG(a, b)$  – обратное гамма-распределение с параметрами  $a$  и  $b$ ;  $N(m, s)$  – нормальное распределение со средним  $m$  и стандартным отклонением  $s$ .

Процесс порождения данных:

$$y_0 \sim N\left(0; \frac{\sigma}{\sqrt{1 - \rho^2}}\right)$$

$$y_t = \rho y_{t-1} + e_t, \quad t = 1, \dots, T$$

$$e_t \sim N(0; \sigma), \quad t = 1, \dots, T$$

где  $y_t$  – значение временного ряда в момент времени  $t$ ;  $e_t$  – случайный шок в момент времени  $t$ .

### Б.2. АВМ

Модель состоит из С-фирм, потребителей (работников), банка и государственного сектора. С-фирмы производят товары с использованием линейной технологии:

$$y_{i,t} = \alpha L_{i,t}$$

где  $\alpha$  – производительность труда;  $y_{i,t}$  – количество произведенных товаров фирмой  $i$  в периоде  $t$ ; а  $L_{i,t}$  – количество использованного для производства труда. Перед тем как производить товары, фирма делает несколько шагов.

Сначала она устанавливает цены и выбирает желаемый уровень выпуска, исходя из правил, похожих на те, что были предложены в Delli Gatti et al. (2011):

$$P_{i,t}^{lower} = \frac{w}{\alpha} + \frac{r_{i-1,t} B_{i,t-1}}{\max(y_{i,t-1}, \alpha)}$$

$$y_{i,t}^* = \max\left(y_{i,t-1} + \rho_{i,t} \left( I(P_{i,t-1} > P_{t-1}) I(\Delta_{i,t-1} > 0) - I(P_{i,t-1} < P_{t-1}) I(\Delta_{i,t-1} \leq 0) \right) y_{i,t-1}, \alpha \right)$$

$$P_{i,t} = \max\left( P_{i,t-1} + \eta_{i,t} \left( I(P_{i,t-1} < P_{t-1}) I(\Delta_{i,t-1} > 0) - I(P_{i,t-1} > P_{t-1}) I(\Delta_{i,t-1} \leq 0) \right) P_{i,t-1}, P_{i,t}^{lower} \right)$$

где  $P_{i,t}^{lower}$  – минимальная цена, устанавливаемая фирмой  $i$  в периоде  $t$ ;  $w$  – заработная плата одного работника;  $r_{i-1,t}$  – средняя ставка процента по кредитам  $i$ -ой фирмы в периоде  $t - 1$ ;  $B_{i,t-1}$  – объем кредитов  $i$ -ой фирмы в периоде  $t - 1$ ;  $y_{i,t}^*$  – желаемый выпуск  $i$ -ой фирмы в периоде  $t$ ;  $P_{i,t}$  – цена товаров  $i$ -ой фирмы в периоде  $t$ ;  $\rho_{i,t}$  и  $\eta_{i,t}$  – случайные величины из равномерного распределения с нулевой нижней границей и верхними границами  $\rho$  и  $\eta$



соответственно;  $P_{t-1}$  – средняя арифметическая цена товаров в периоде  $t - 1$ ;  $\Delta_{i,t-1}$  – избыточный спрос на продукцию  $i$ -ой фирмы в периоде  $t - 1$ ;  $I$  – индикатор, равный единице, если условие выполняется, и равный нулю, если нет.

После формирования цен и желаемого уровня выпуска, фирмы определяют количество денег, необходимое для финансирования производства. Эта сумма определяется зарплатами, которые необходимо выплатить сотрудникам, участвующим в производстве. Количество сотрудников, необходимое фирме для производства желаемого уровня выпуска рассчитывается по формуле:

$$L_{i,t}^* = \frac{y_{i,t}^*}{\alpha}$$

Если фирма понимает, что собственной ликвидности,  $M_{i,t-1}$ , не хватает, то она обращается за недостающим финансированием  $wL_{i,t}^* - M_{i,t-1}$  в банк.

Банк принимает решение о максимальном объеме выдаваемых  $i$ -й фирме кредитов на основе выполнения нормативов по капиталу и уже выданных кредитов. Итоговый размер предоставляемого финансирования фирме  $i$ :

$$B_{i,t}^{new} = \min \left( \max(wL_{i,t}^* - M_{i,t-1}, 0), \frac{E_{t-1}}{\zeta N_c} - B_{i,t-1} \right)$$

где  $E_{t-1}$  – собственные средства банка в периоде  $t - 1$ ,  $\zeta$  – константа, отвечающая за норматив достаточности капитала<sup>18</sup>;  $N_c$  – количество С-фирм. Ставка, по которой предоставляется новый кредит, устанавливается исходя из желаемого фирмой отношения заемных средств к активам ( $\lambda_{i,t}$ ):

$$r_{i,t}^{new} = r(1 + \mu_{i,t}\lambda_{i,t}^{0.5})$$

$$\lambda_{i,t} = \frac{\max(wL_{i,t}^* - M_{i,t-1}, 0) + B_{i,t-1}}{\max(wL_{i,t}^* - M_{i,t-1}, 0) + B_{i,t-1} + E_{i,t-1}^c + 10^{-8}}$$

где  $r_{i,t}^{new}$  – ставка по взятому  $i$ -ой фирмой в периоде  $t$  кредиту;  $r$  – безрисковая ставка процента;  $\mu_{i,t}$  – случайная величина из равномерного распределения с нулевой нижней границей и верхней границей  $\mu$ . Новый объем кредитов ( $B_{i,t}^*$ ) и средняя ставка ( $r_{i,t}$ ) для фирмы  $i$  запишутся как:

$$B_{i,t}^* = B_{i,t}^{new} + B_{i,t-1}$$

$$r_{i,t} = \frac{B_{i,t}^{new}}{B_{i,t}^*} r_{i,t}^{new} + \frac{B_{i,t-1}}{B_{i,t}^*} r_{i-1,t}$$

<sup>18</sup> Не обязательно равная ему.

Финальным шагом перед производством является найм труда. Зная уровень доступных средств и желаемый выпуск, фирмы размещают вакансии для поиска работников. Каждая фирма публикует новые вакансии, если

$$V_{i,t} = \text{round} \left( \min \left( L_{i,t}^* - L_{i,t-1}, \frac{M_{i,t-1} + B_{i,t}^{\text{new}}}{w} \right) \right) > 0$$

и увольняет сотрудников, если  $V_{i,t} < 0$ . Работа рынка труда происходит в  $Z_e$  раундов. Работники сортируются в случайном порядке. В каждом из раундов работающие приходят к своему работодателю, и если у фирмы излишек рабочей силы, то она увольняет тех, кто пришел раньше. Безработные приходят к случайно выбранной фирме и устраиваются туда, если есть незаполненные вакансии.

После этапа найма фирмы производят товары, выплачивают зарплату и переходят к продаже товаров. Как и на рынке труда, на рынке товаров работает механизм поиска. Потребители формируют свой желаемый бюджет на потребление ( $CB_{j,t}^*$ ), который складывается из оценки постоянного дохода ( $PI_{j,t}$ ) и текущего благосостояния ( $W_{j,t}$ ):

$$PI_{j,t} = \Sigma PI_{j,t-1} + (1 - \Sigma) I_{j,t}$$

$$CB_{j,t}^* = \psi PI_{j,t} + \chi W_{j,t}$$

где  $I_{j,t}$  — доход  $j$ -го потребителя в период времени  $t$ , равный сумме дивидендов за вычетом докапитализации фирм банкротов (см. ниже) и зарплаты после налогов, если он работает, и пособия по безработице  $zw$ , если он безработный;  $\Sigma$  — параметр сглаживания при оценке постоянного дохода;  $\psi$  и  $\chi$  — доли постоянного дохода и благосостояния, которые идут на формирование желаемого бюджета на потребление. Если желаемый бюджет на потребление меньше текущего благосостояния, то он и составляет бюджет на потребление, иначе бюджет на потребление составляет половину благосостояния. Имея в распоряжении бюджет на потребление, потребители случайным образом выбирают  $Z_c$  фирм и затем посещают их в  $Z_c$  раундах. В первом раунде потребитель идет к фирме с наименьшей ценой из выбранных и покупает у нее товары. Если у фирмы недостаточно товаров для удовлетворения спроса потребителя, то он покупает все, что есть, и переносит оставшийся бюджет на следующий раунд.

Каждый период фирмы возвращают банкам фиксированную долю задолженности  $\theta$ . Фирмы с положительной прибылью платят дивиденды домохозяйствам (всем одинаково) как минимум из доли  $\tau$  от прибыли и текущей наличности, имеющейся на руках у фирмы. Те фирмы, которые по итогам всех предыдущих шагов имеют отрицательный собственный капитал, объявляются банкротами (банк теряет разницу между суммой кредита и ликвидными

средствами), а все их сотрудники становятся безработными. Их заменяют новые фирмы, которые входят на рынок с нулевой задолженностью, а также капиталом и ликвидностью, равными средним собственным средствам не обанкротившихся фирм. Создание фирм финансируется домохозяйствами в равных пропорциях. Если суммарное благосостояние домохозяйств меньше, чем нужное для создания новых фирм количество денег, то размеры новых фирм пропорционально уменьшаются. Кроме того, если при докапитализации какому-то из домохозяйств не хватает благосостояния для финансирования всех фирм, он получает безвозмездный трансферт от государства в размере недостающих средств.

При входе фирмы должны установить значения переменных, таких как цена, прошлый выпуск и спрос на продукцию. Мы рассматриваем два варианта того, как это делается, в некотором смысле закладывая равномерное априорное распределение на две разные модели. В первой из них фирмы устанавливают показатели исходя из показателей фирмы предшественника, во второй – на основе средних по рынку.

На этом период заканчивается, все агрегированные переменные рассчитываются и добавляются в набор наблюдаемых данных как переменные периода  $t$ .

**Таблица Б1. Значения параметров и априорные распределения**

Параметр	Описание	Значение	Априорное распределение
$N_w$	Количество домохозяйств	500	-
$N_c$	Количество С-фирм	50	-
$Z_c$	Количество раундов на рынке товаров	2	-
$Z_e$	Количество раундов на рынке труда	4	-
$\alpha$	Производительность труда	0,5	-
$\tau$	Доля дивидендов от прибыли	0,2	-
$r$	Безрисковая ставка процента	0,01	-
$w$	Зарплата	1	-
$z$	Отношение пособия по безработице к зарплате	0,5	-
$\Sigma$	Параметр сглаживания при оценке постоянного дохода	-	<i>Uniform(0; 1)</i>
$\psi$	Доля постоянного дохода, которая идет на потребление	-	<i>Uniform(0,5; 1)</i>
$\chi$	Доля благосостояния, которая идет на потребление	-	<i>Uniform(0; 1)</i>
$\rho$	Верхняя граница изменения выпуска	-	<i>Uniform(0; 0,5)</i>
$\eta$	Верхняя граница изменения цен	-	<i>Uniform(0; 0,5)</i>
$\mu$	Верхняя граница риск-премии	-	<i>Uniform(0; 2)</i>
$\theta$	Доля возвращаемых кредитов	-	<i>Uniform(0,01; 0,1)</i>
$t_w$	Налог на зарплату	-	<i>Uniform(0; 0,3)</i>
<i>Newvalues</i>	Механизм установления цен после банкротства	-	<i>Binomial(2; 0,5)</i>

### **Приложение В. Условный прогноз для универсальной архитектуры**

В этом приложении мы проводим дополнительные эксперименты для нейронной сети с универсальной архитектурой, которая может быть использована не только в случае, когда сценарии строятся на основе наблюдаемых переменных. Как указывалось в подразделе 3.2, одна из реализаций такой архитектуры основана на добавлении сценариев на уровне полносвязных слоев. В рамках экспериментов мы перебрали различные гиперпараметры, в том числе на уровне архитектуры полносвязных слоев, и показываем здесь результаты для модели с пятью слоями и 100 нейронами на каждом из них. Эта модель обучалась в течение 10 часов (1 000 000 итераций с размером батча 100) и показала наилучшие результаты. В качестве задачи для обучения была использована та же самая задача, что и для условного прогноза. Несмотря на то что она не содержит ненаблюдаемых переменных, эта задача подходит для тестирования более универсальной архитектуры и ее свойств достаточно хорошо, так как обученная в подразделе 4.3 модель может быть использована как ориентир при оценке качества.

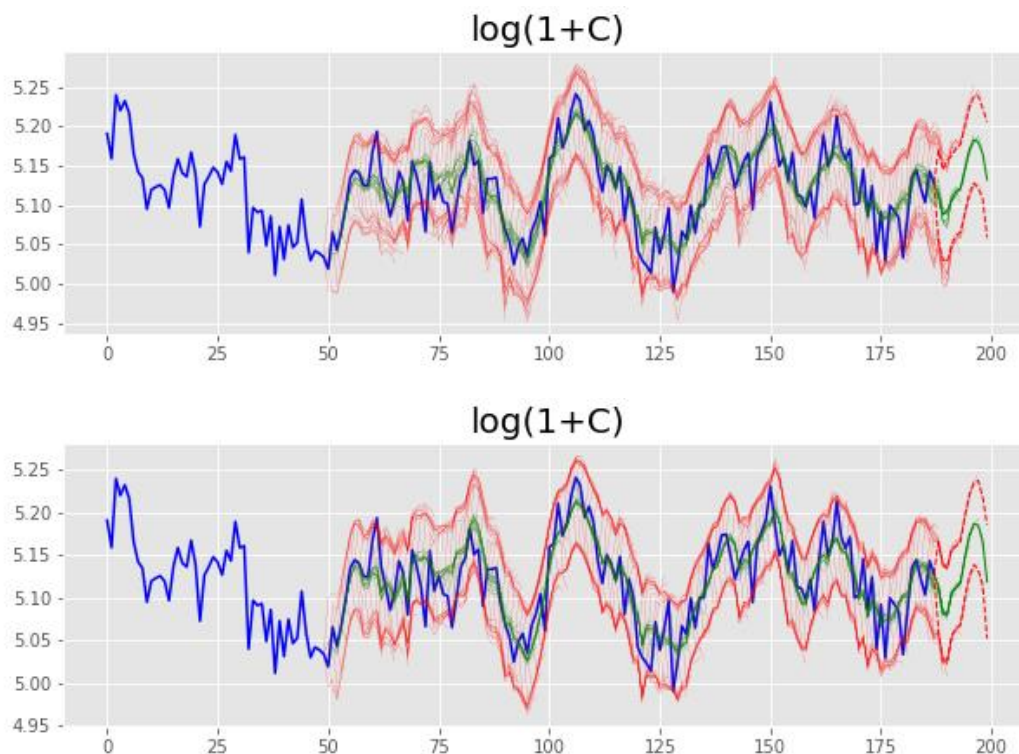
На рисунке В1 показаны прогнозы на тех же данных, что и в основной части статьи. Хотя результаты и похожи, они тем не менее отличаются. При этом, как видно из сравнения результатов таблицы В1 и таблицы 5, именно модель с универсальной архитектурой показывает худшие результаты. Тест на стандартизированных ошибках (рисунки В2 и В3) также показывает наличие большего смещения средних, чем в случае архитектуры, описанной в подразделе 3.2. Более того, несмотря на то что смещения средних в некоторых случаях по величине сравнимы с теми, которые были получены для модели безусловного прогноза (таблицы 3 и 4), они имеют систематический характер и, как мы увидели из процесса обучения, не связаны с дрейфом коэффициентов. Тем не менее, чтобы такого рода выводы были более системными, необходимо развитие формальных статистических процедур или моделей усреднения, как отмечалось ранее в разделе 5.

Все вышеизложенное сигнализирует о том, что рассматриваемая здесь альтернативная архитектура не позволяет получить прогноз, который аппроксимировал бы характеристики апостериорного распределения. Тем не менее, как видно из рисунка В1, полученные результаты близки к результатам подраздела 4.3, и в некоторых задачах такого качества аппроксимации может быть достаточно. Однако решение относительно достаточности качества аппроксимации должно приниматься для каждой конкретной задачи по отдельности, исходя из целей построения модели и рисков, которые несут в себе ошибки аппроксимации.

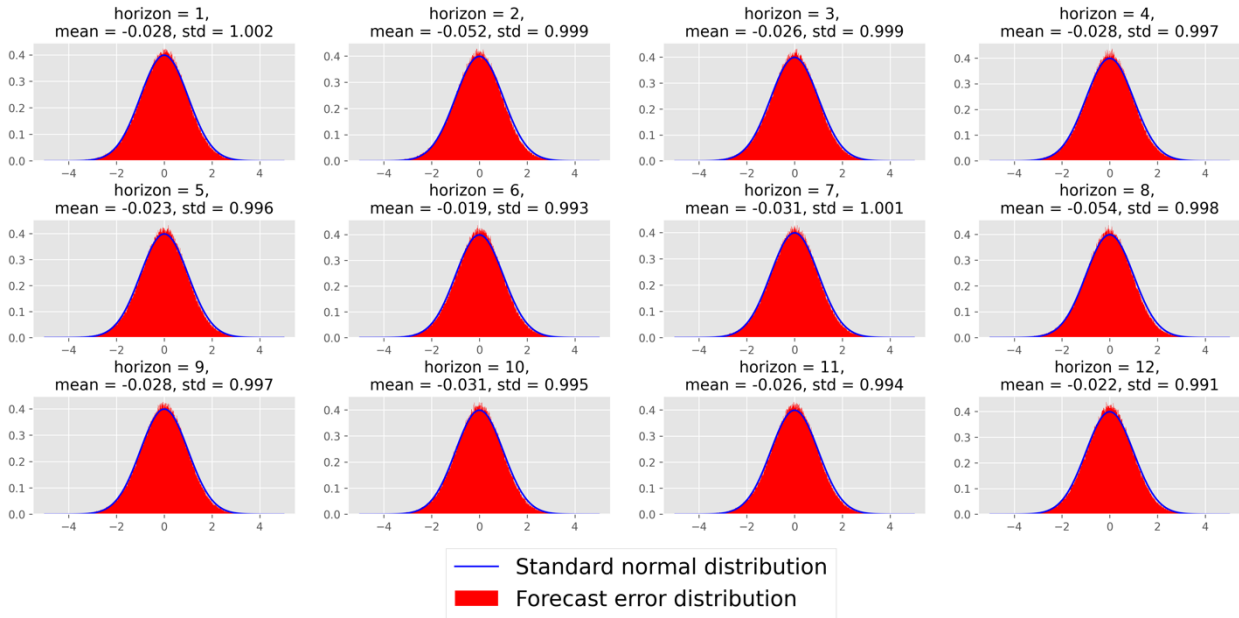
**Таблица В1. Сравнение прогнозных свойств VAR-модели и нейронной сети, аппроксимирующей условный прогноз из АВМ, универсальная архитектура (MSFE – отношение среднеквадратичных ошибок VAR и нейронной сети, LPS – разность средних логарифмов правдоподобия прогнозов VAR и нейронной сети)**

Горизонт прогнозирования	MSFE	LPS
1	1,076***	-0,175**
2	1,101***	-0,161***
3	1,127***	-0,119***
4	1,16***	-0,134***
5	1,17***	-0,148***
6	1,167***	-0,143***
7	1,172***	-0,146***
8	1,174***	-0,152***
9	1,18***	-0,158***
10	1,192***	-0,165***
11	1,181***	-0,183***
12	1,112***	-0,121***

**Рисунок В1. Условный прогноз с использованием нейронной сети для АВМ (верхний график – универсальная архитектура, нижний график – архитектура из подраздела 3.2)**



**Рисунок В2. Распределения стандартизированных ошибок прогнозов для условного прогноза из АВМ, универсальная архитектура**



**Рисунок В3. Распределения произведений стандартизированных ошибок прогнозов для условного прогноза из АВМ, универсальная архитектура**

