Глубокие самообучающиеся агенты в мультиагентной системе маршрутизации

Мухутдинов Дмитрий, группа M3438
Научный руководитель: Фильченков А. А., к.ф-м.н., доцент кафедры КТ
Рецензент: Тарасов В. Б., к.т.н., МГТУ им. Баумана

Кафедра Компьютерных Технологий
Факультет Информационных Технологий и Программирования
Университет ИТМО, Санкт-Петербург

8 июня 2017 г.

Задача маршрутизации

- Сетевой роутинг
- Транспортная логистика
- Управление конвейерными системами
- Автоматическое управление городским трафиком
- ...

Существующие решения

- Link-state
  - Open Shortest Path First (OSPF)
  - IS-IS
- Distance-vector
  - RIP
  - IGRP
- Прочие
  - AntNet
  - ...

Цель работы

- Примерно все алгоритмы маршрутизации заточены под компьютерные сети
- В других задачах существуют свои, более сложные условия
  - Скорую нужно пропустить сквозь пробку, а обычный автомобиль — нет
  - Хочется минимизировать энергопотребление конвейеров
  - ...
- Задача: построить алгоритм, способный адаптироваться под гетерогенные условия

Идея

- Обучение с подкреплением
- Нейросети в качестве обучающихся агентов
- Q-routing (Boyan & Littman, 1994)
  - Не получил широкого распространения в компьютерных сетях (использует слишком много служебных пакетов)
  - В других задачах (трафик, конвейеры) это не является проблемой.

Постановка задачи в терминах RL

- Рассмотрим *пакет* в сети как обучающегося агента, взаимодействующего с сетью как со средой
- Полное состояние среды неизвестно, состояние текущего роутера — *наблюдение* пакета
- Действие — переход к одному из соседей
- Q-learning:

$$Q(o_t, a_t) \leftarrow r_t + \gamma \cdot \max_{a \in \mathcal{A}_{o_{t+1}}} Q(o_{t+1}, a)$$

- Принцип аналогичный Q-routing:

$$Q_x(d, y) \leftarrow (t_{finish} - t_{start}) + \max_{z \in \{V | (y,z) \in E\}} Q_y(d, z)$$

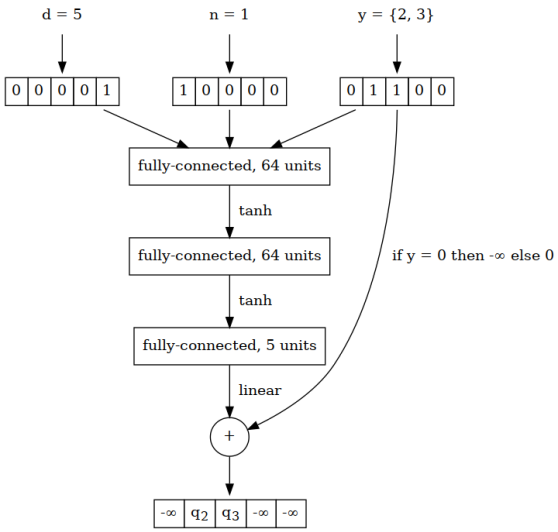
Базовая архитектура нейросети

Вход нейросети $o = (d, n, y_1 ... y_n, o')$, где:

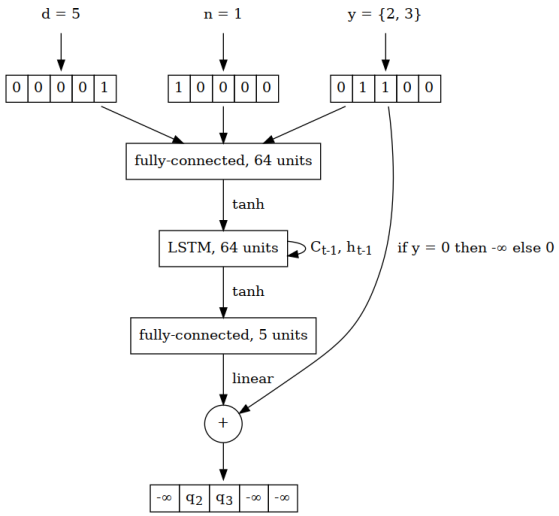
- d — узел назначения, n — текущий узел, $y_1 ... y_n$ — номера соседей, o' — любая дополнительная информация
- Выходы нейросети $a_1 ... a_n$ — оценки $Q(o, a)$ для всех узлов сети ($-\infty$ для узлов, не являющихся соседями)
- Кодируем номера унитарным кодом, чтобы избежать корреляции результатов
- Используем RMSProp для оптимизации

Варианты архитектуры нейросети

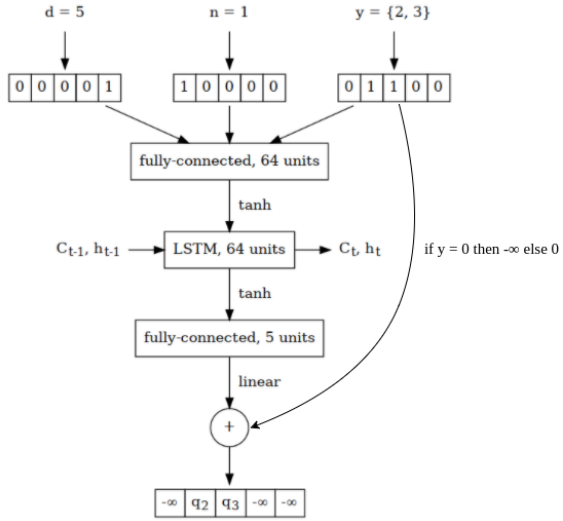
Feed-forward



LSTM с “памятью роутера”



LSTM с “памятью пакета”



Проблема нестационарности

- Q-обучение, вообще говоря, не сходится к оптимальному решению в случае бесконечного числа состояний
- Другие агенты меняют свое поведение — среда нестационарна, вследствие чего experience replay (Mnih et al., 2015) не работает
- При обучении с нуля на равномерной низкой нагрузке нейросети не могут найти оптимальные пути

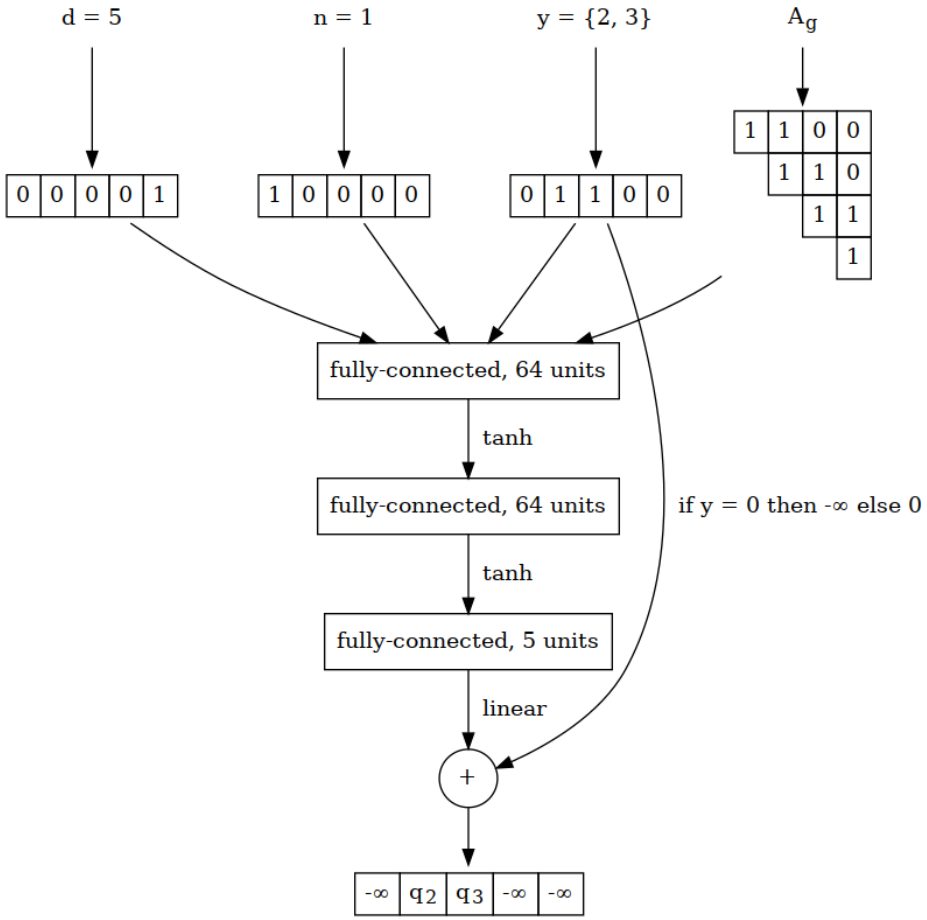
Решение проблемы нестационарности

- Предобучение сети (bootstrapping)
  - Собираем данные работы алгоритма кратчайших путей
  - Обучаем одну нейросеть на данных от всех роутеров
  - Предобученная сеть повторяет работу алгоритма кратчайших путей
- Отказ от experience replay во время работы
  - Условия работы в сети меняются
  - Старый опыт перестает быть актуальным
  - Если обращать на него внимание, адаптивность к изменяющимся условиям страдает

Расширение наблюдаемого состояния

Добавим на вход дополнительную информацию

- Топологию сети полезно учитывать
- Используем link-state протокол для обновления информации о топологии
- Подаем матрицу смежности графа на вход



Компромисс между исследованием и использованием

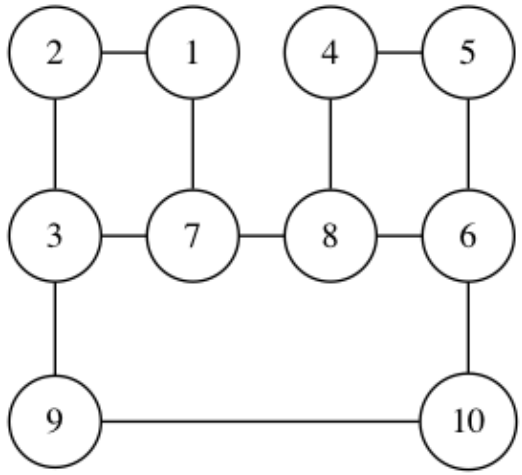
- Алгоритм может застревать в локальном максимуме
- Сложности к возвращению к оптимальной стратегии после изменений условий среды
- Решение: используем softmax-стратегию, чтобы иметь шанс выбрать не максимально “выгодное” действие:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for} \quad j = 1 \,..\, K$$

$$P(a \mid o) = \sigma(Q(o, a))$$

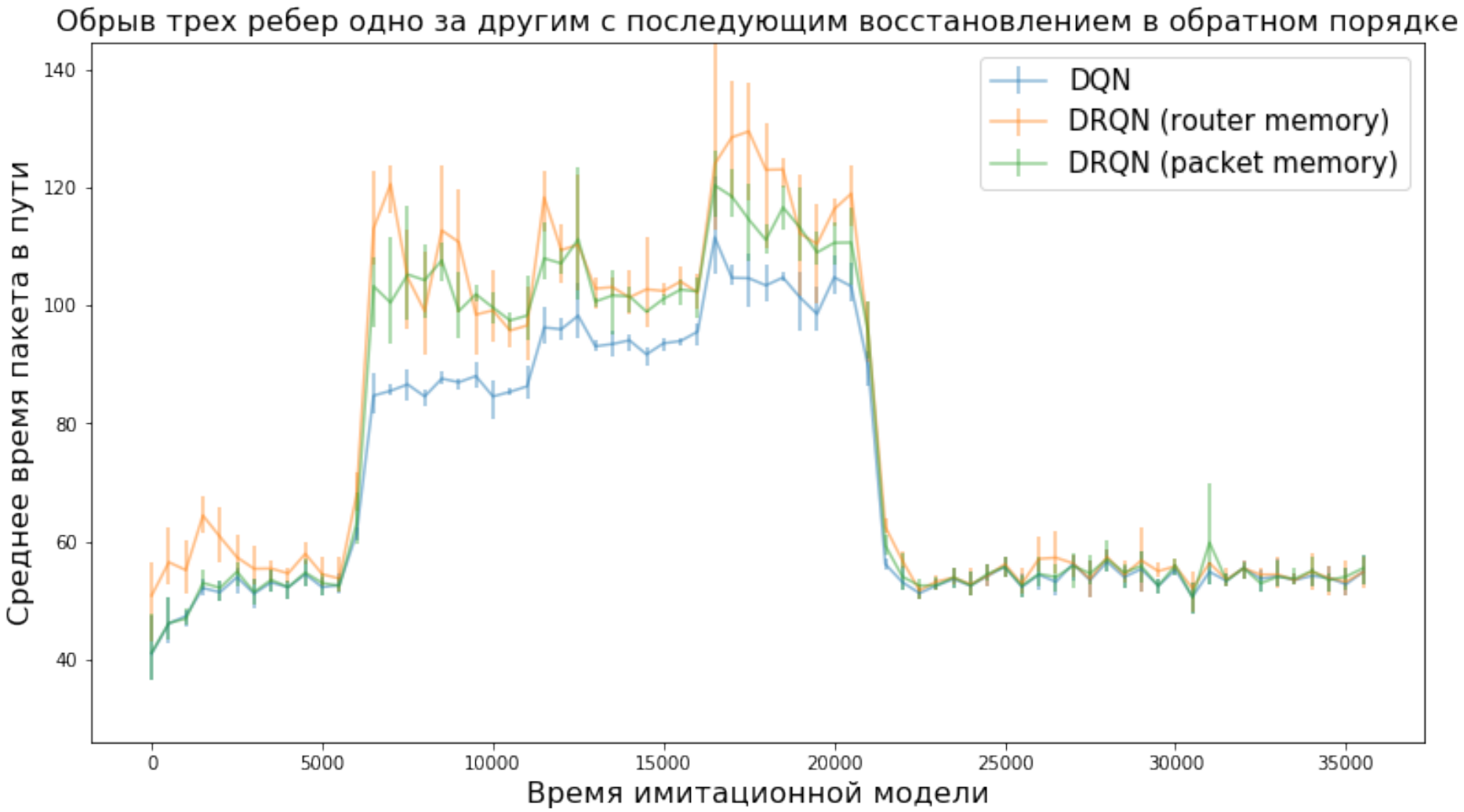
Эксперименты в модели компьютерной сети

- Запуск экспериментов в симуляторе
- Сравнение архитектур между собой и сравнение с алгоритмами link-state и Q-routing
- Служебные сообщения доставляются бесплатно

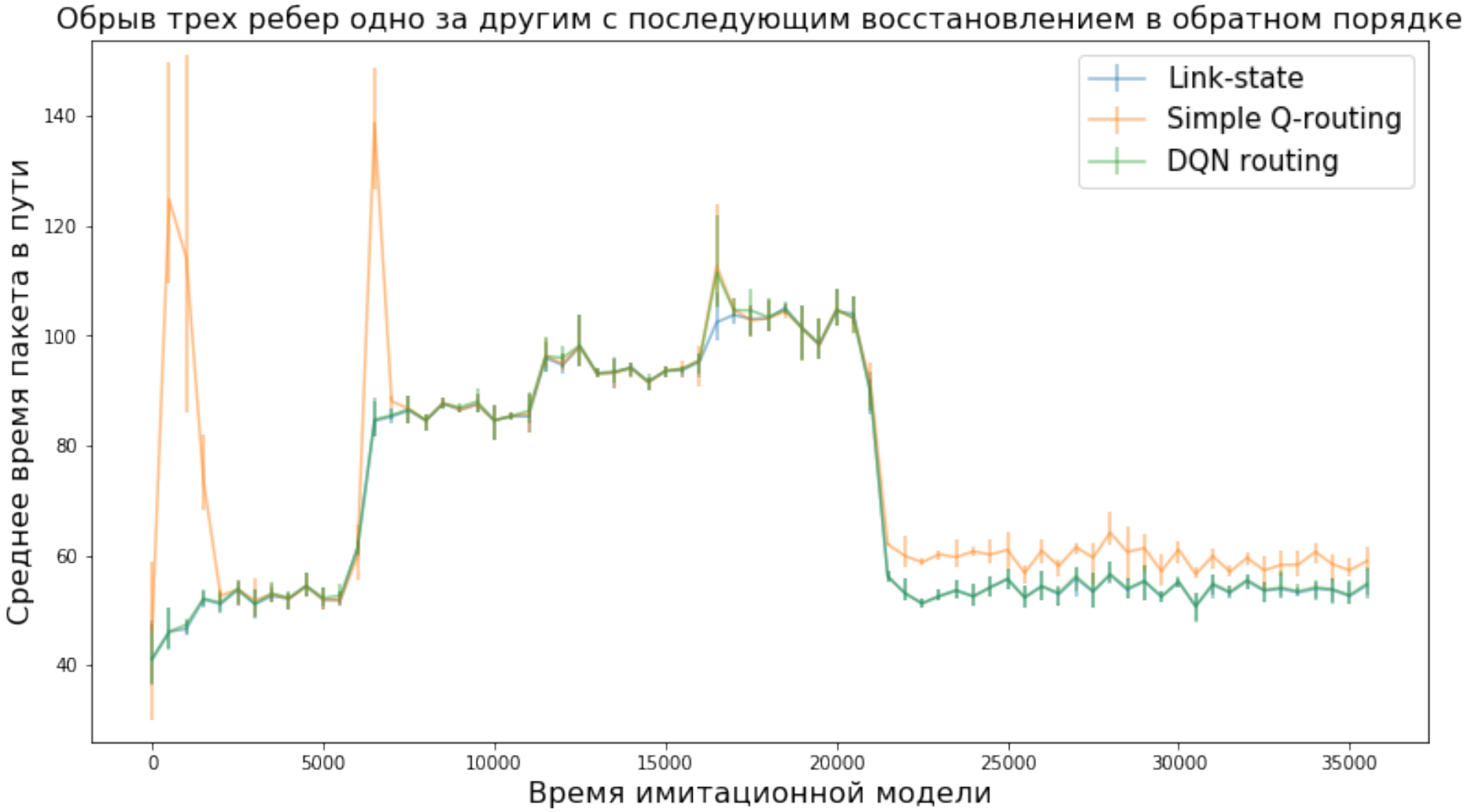


Топология сети для тестов

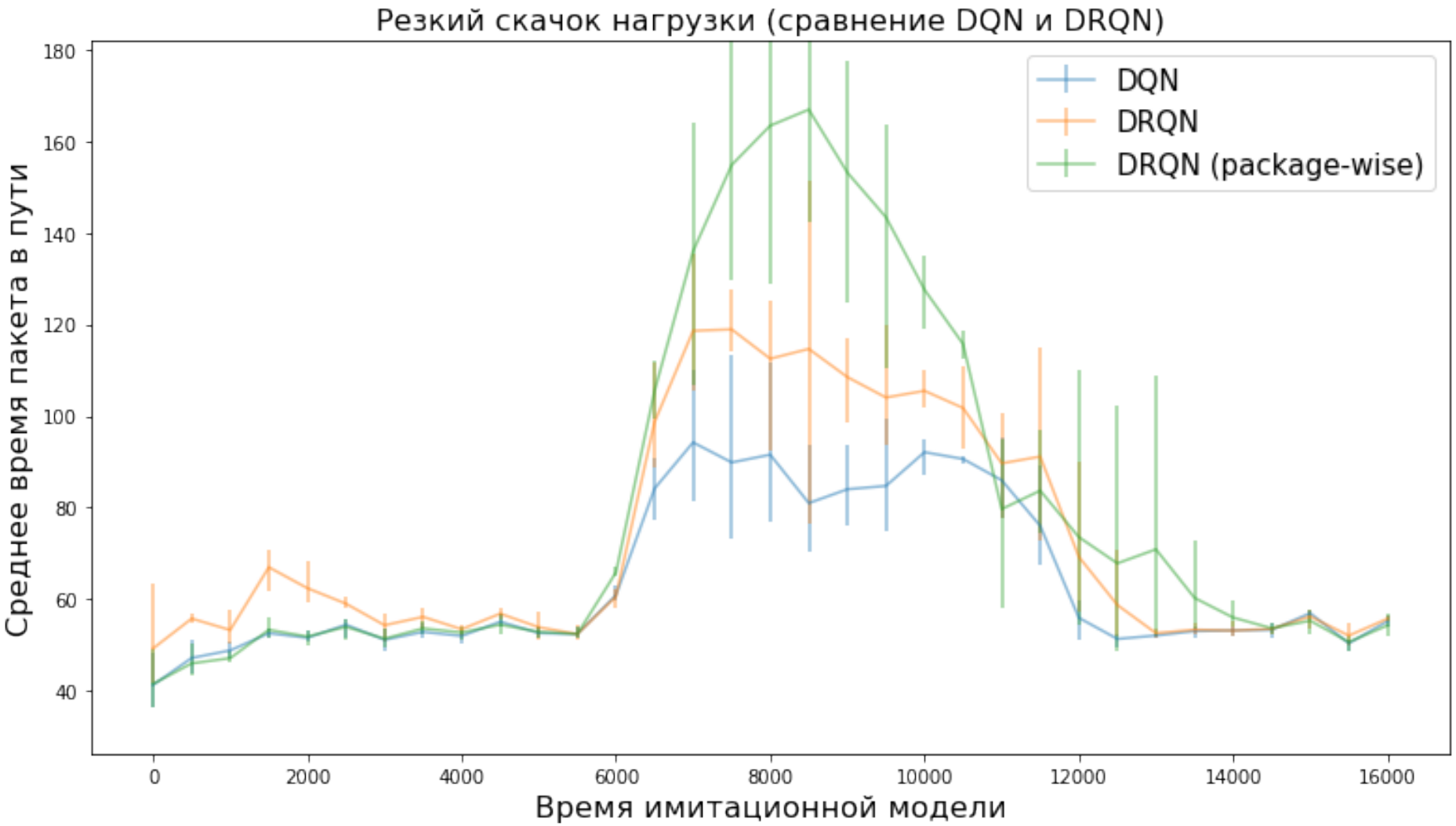
Изменение топологии сети: сравнение архитектур нейросетей



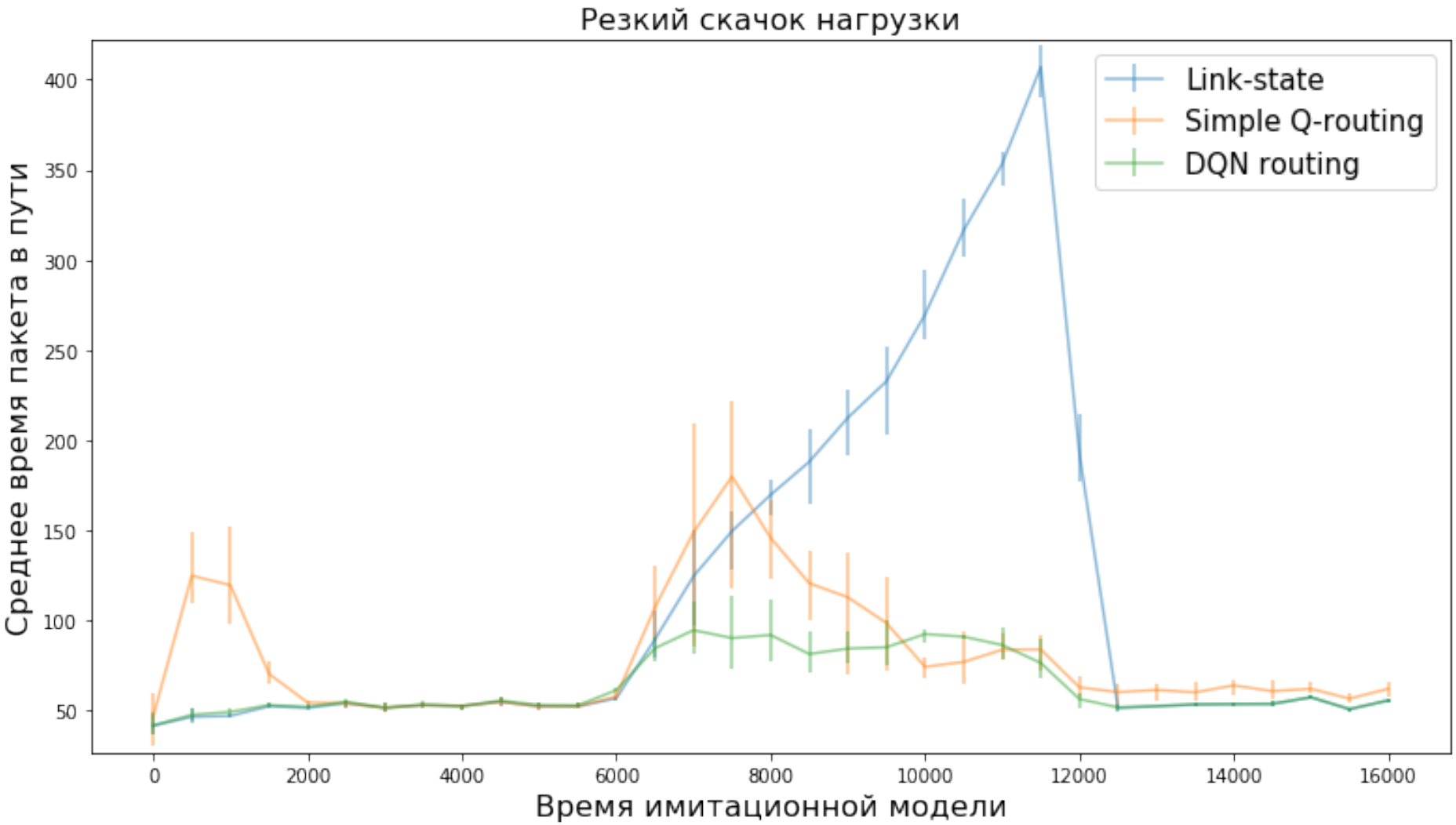
Изменение топологии сети: сравнение с link-state и Q-routing



Изменение нагрузки: сравнение с архитектур нейросетей

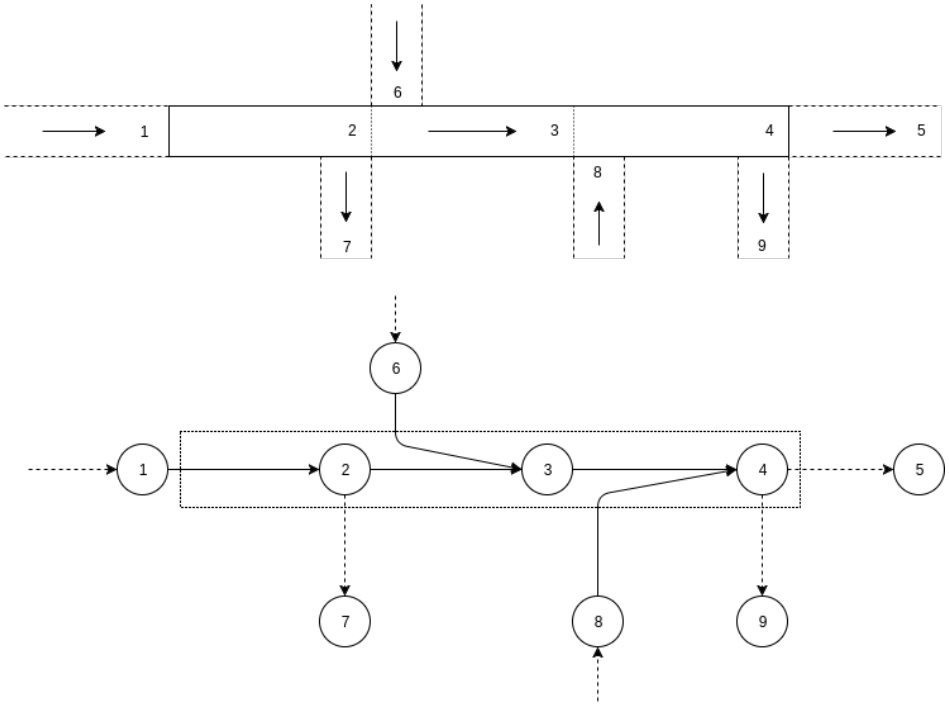


Изменение нагрузки: сравнение с link-state и Q-routing



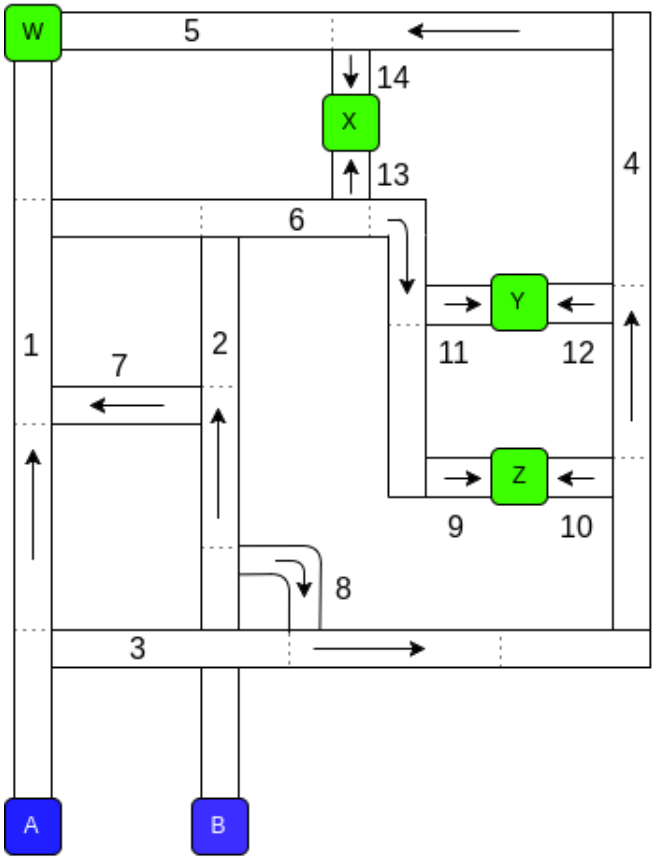
Эксперименты в модели системы багажных конвейеров

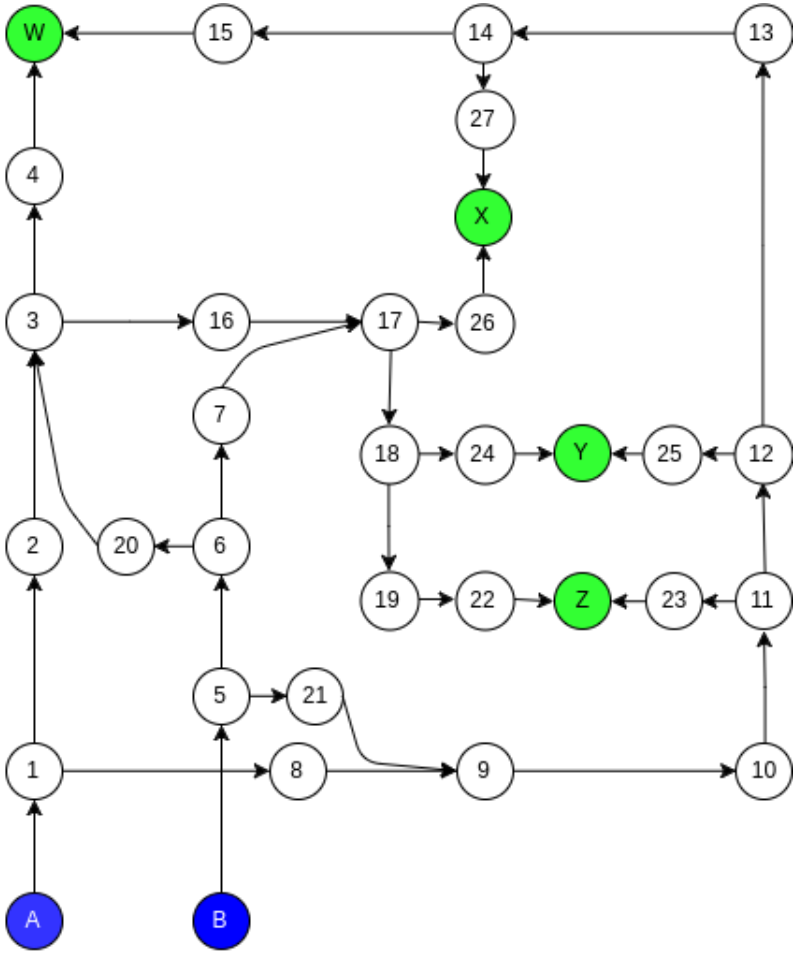
- Конвейерная сеть моделируется как ориентированный граф
- Каждая секция конвейера – отдельная вершина
- Отдельно выделены входные и выходные вершины
- В оптизируемую функцию включено *энергопотребление*
- Важность экономии энергии регулируется параметром α



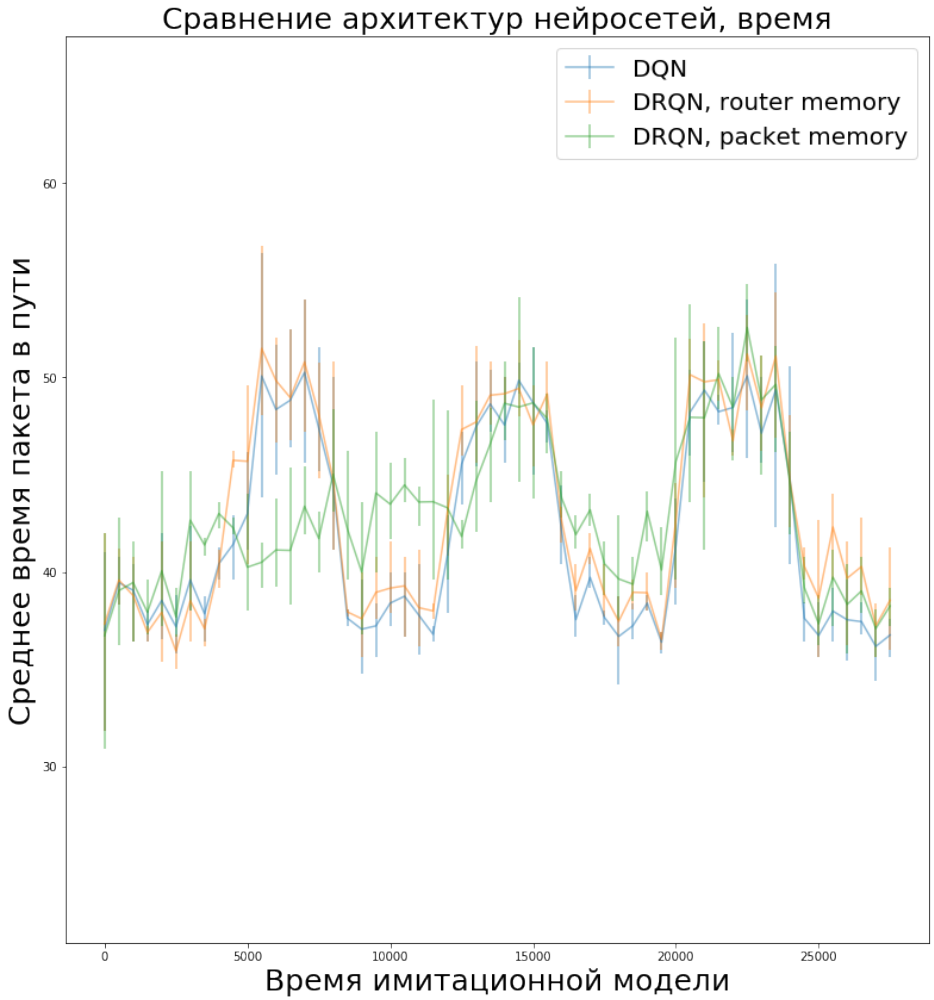
Участок конвейерной сети и соответствующий участок графа

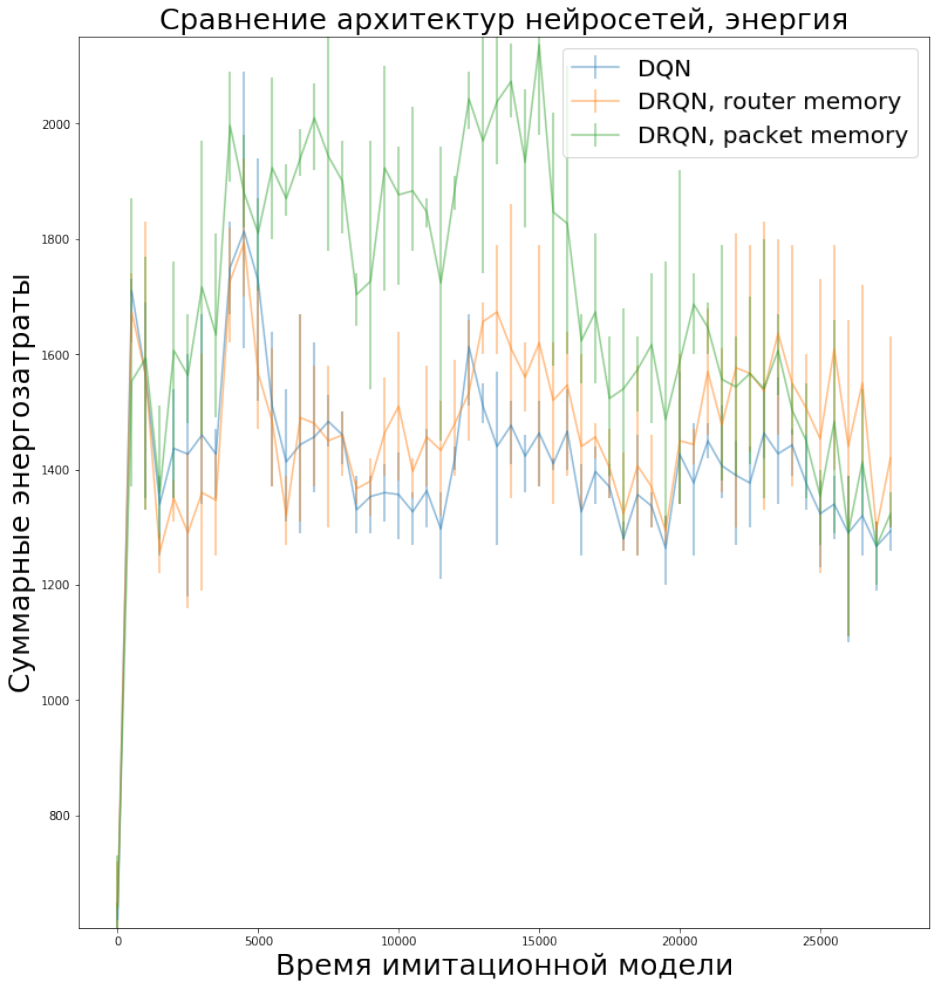
Модель конвейерной системы для проведения экспериментов



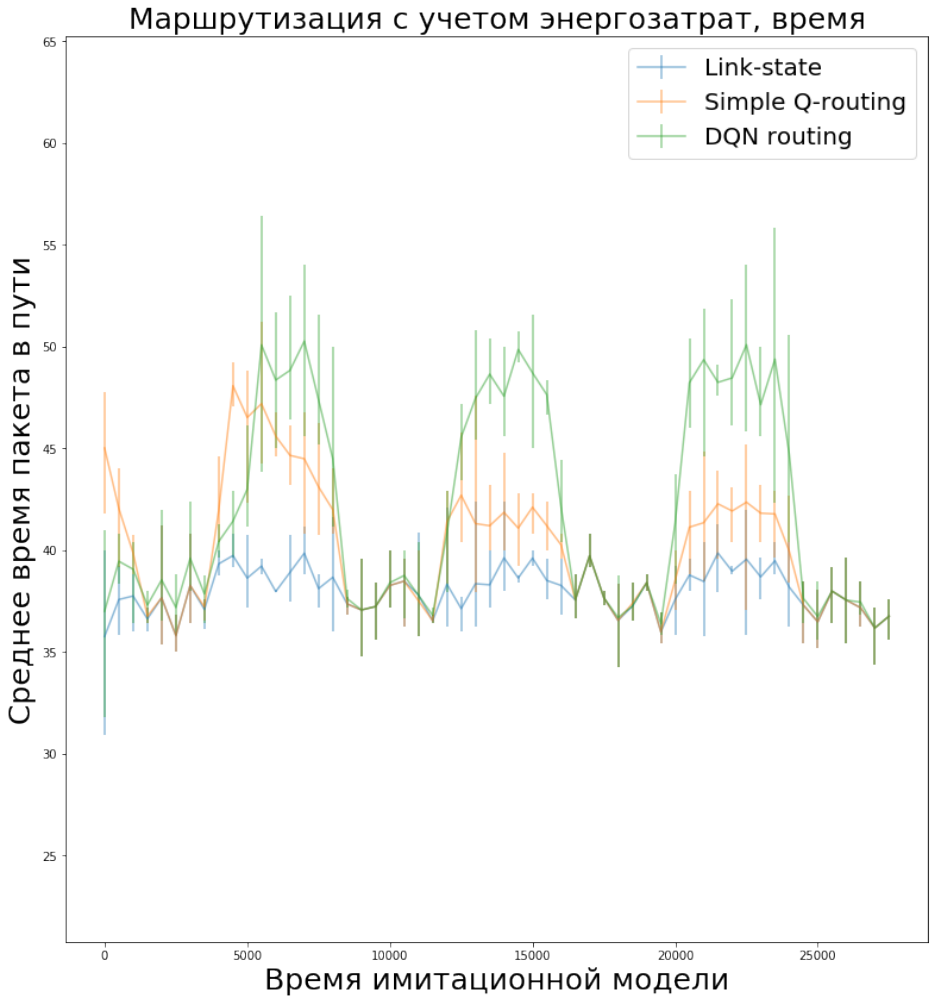


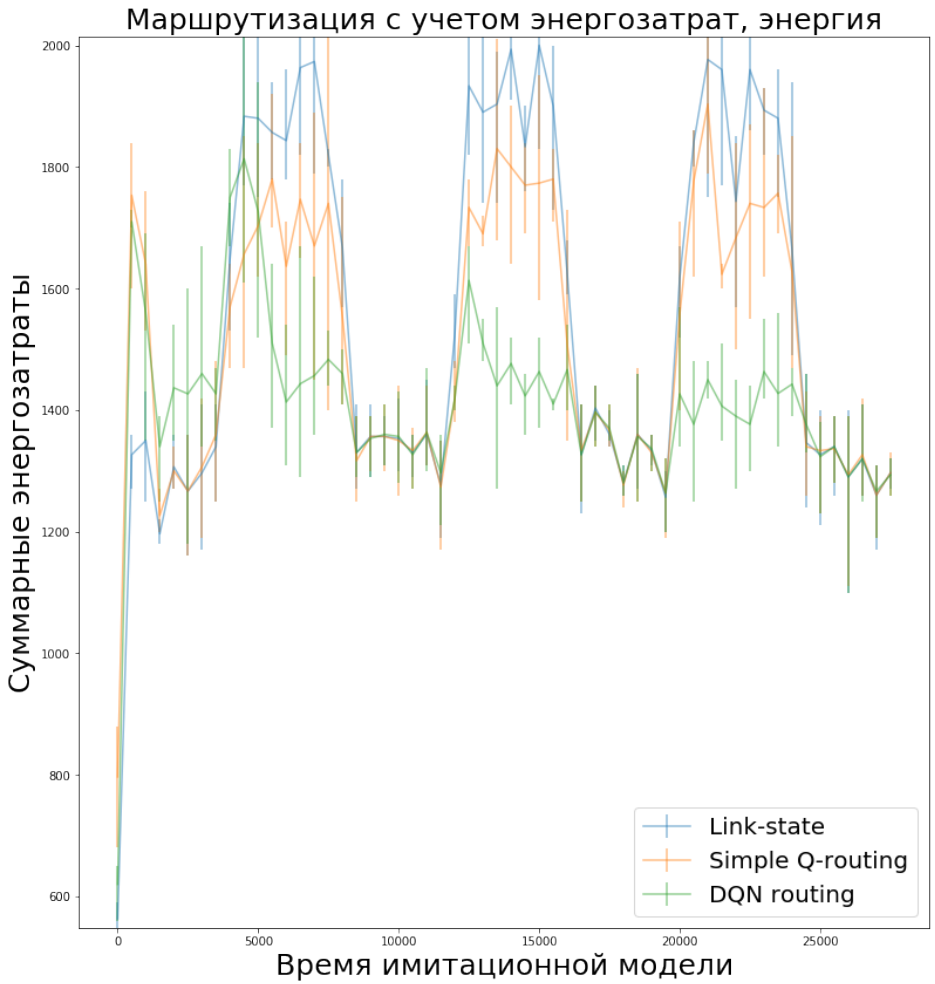
Неравномерный поток до выходных вершин: сравнение архитектур нейросетей



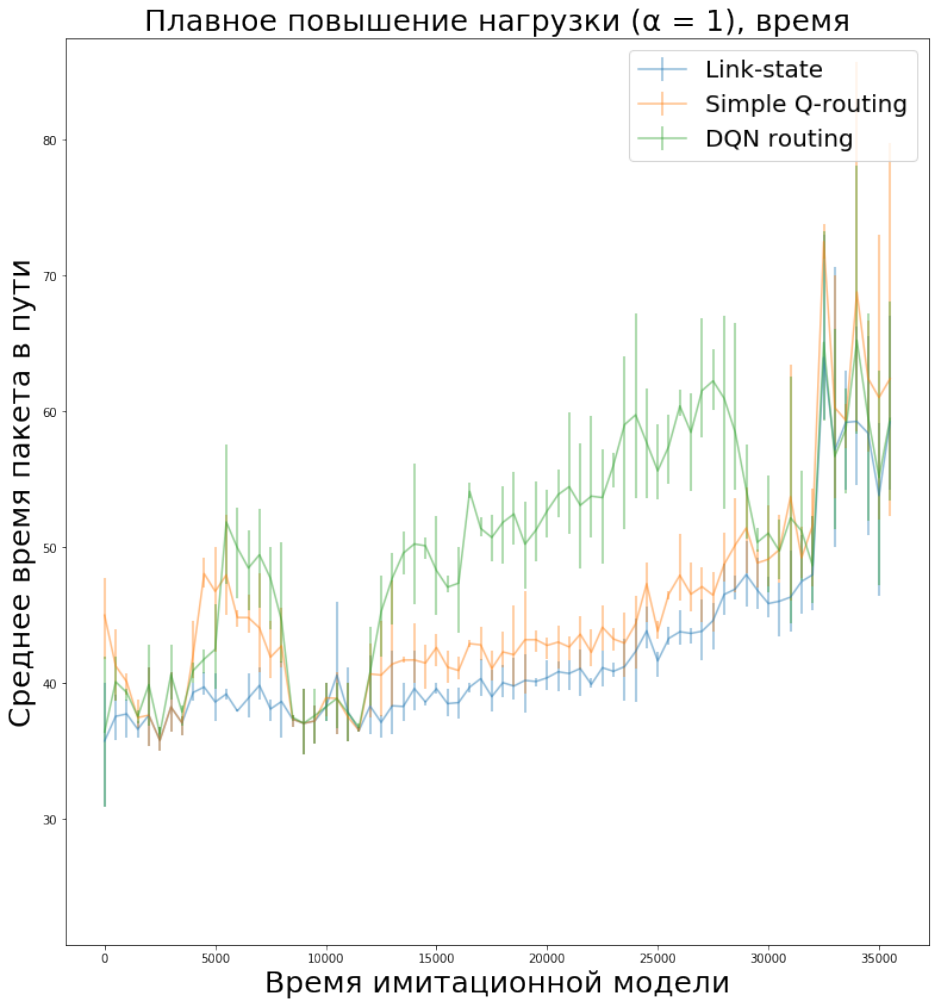


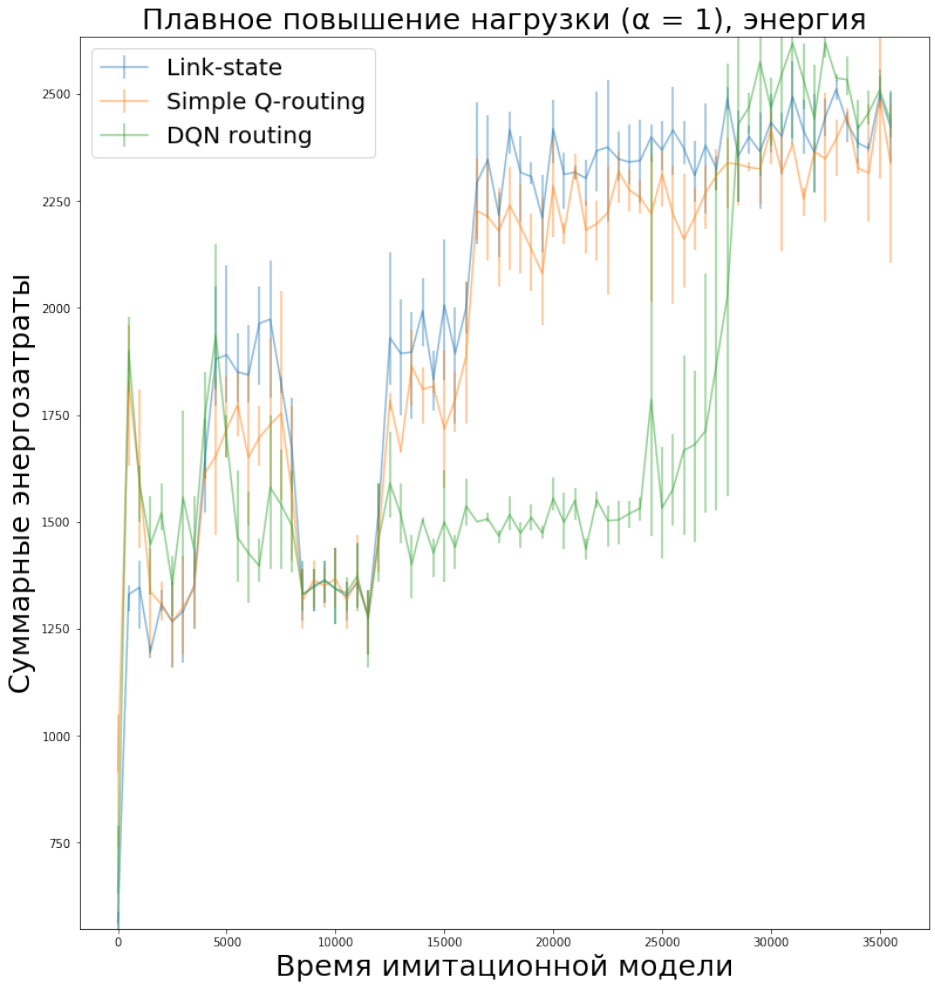
Неравномерный поток до выходных вершин: сравнение с link-state и Q-routing



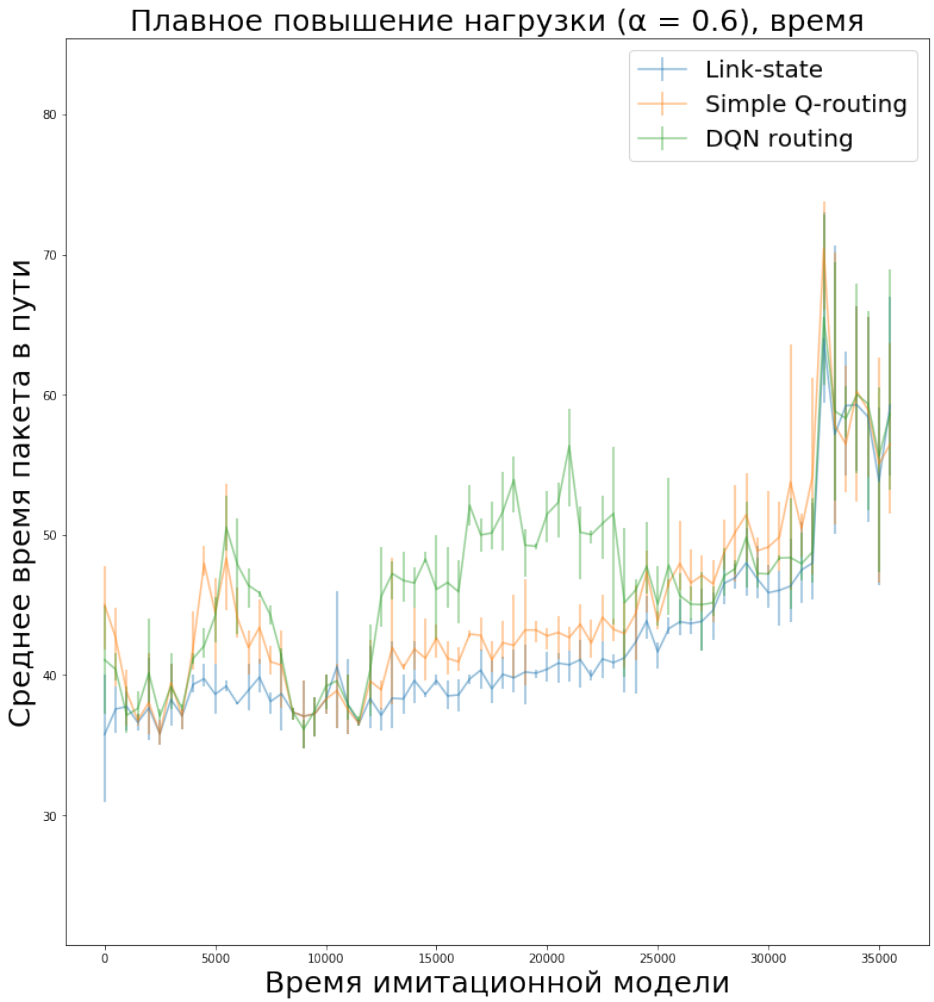


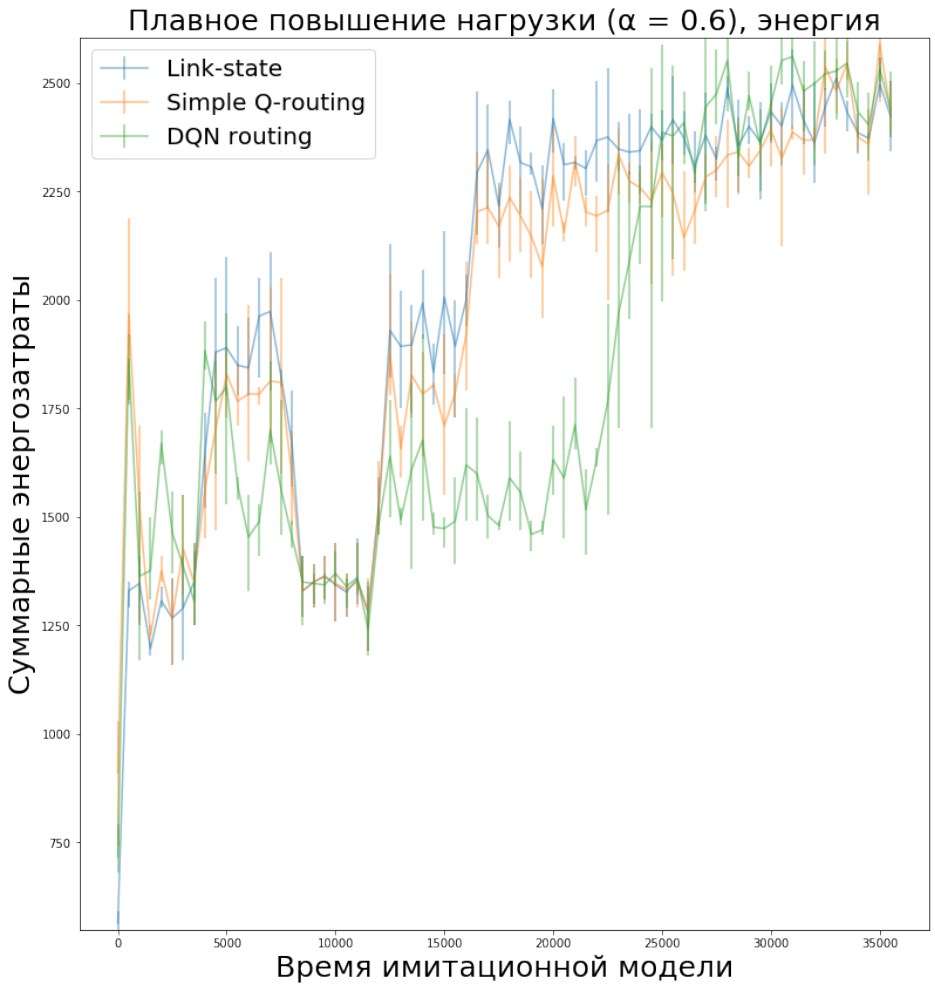
Плавное повышение нагрузки: $\alpha = 1$





Плавное повышение нагрузки: $\alpha = 0.6$





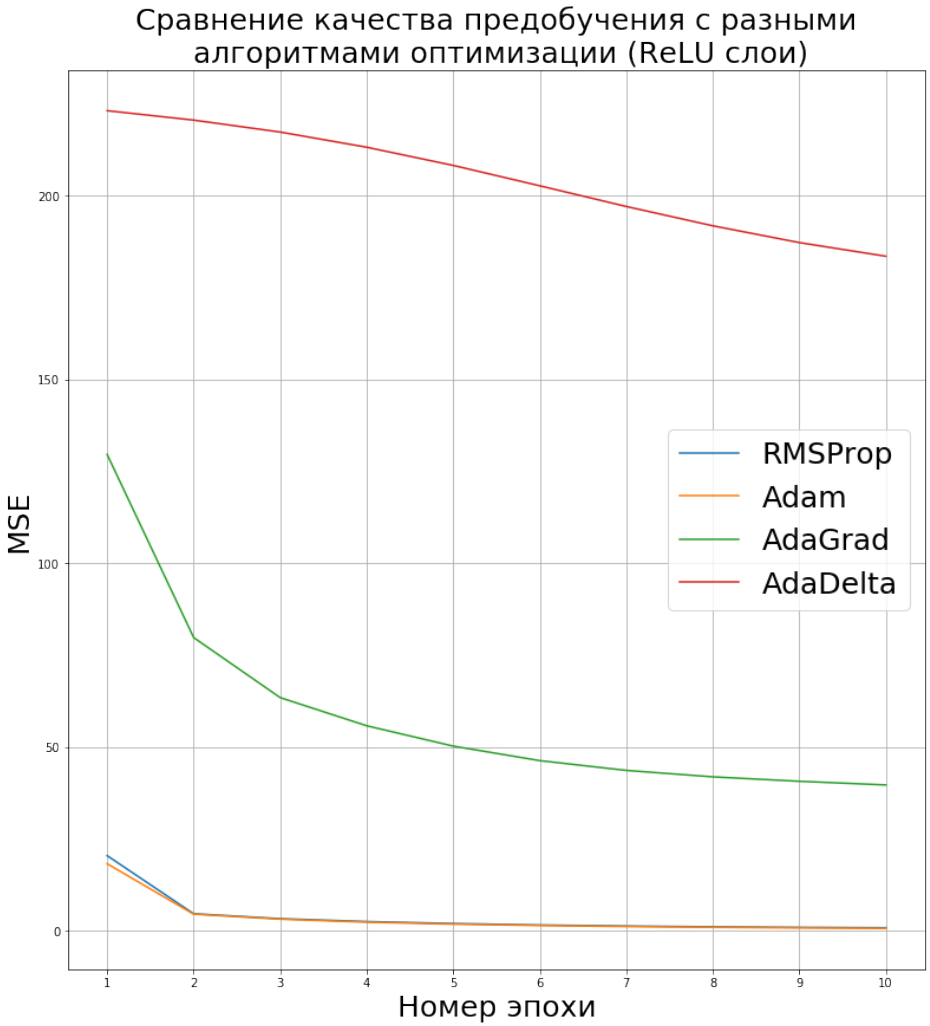
Итоги

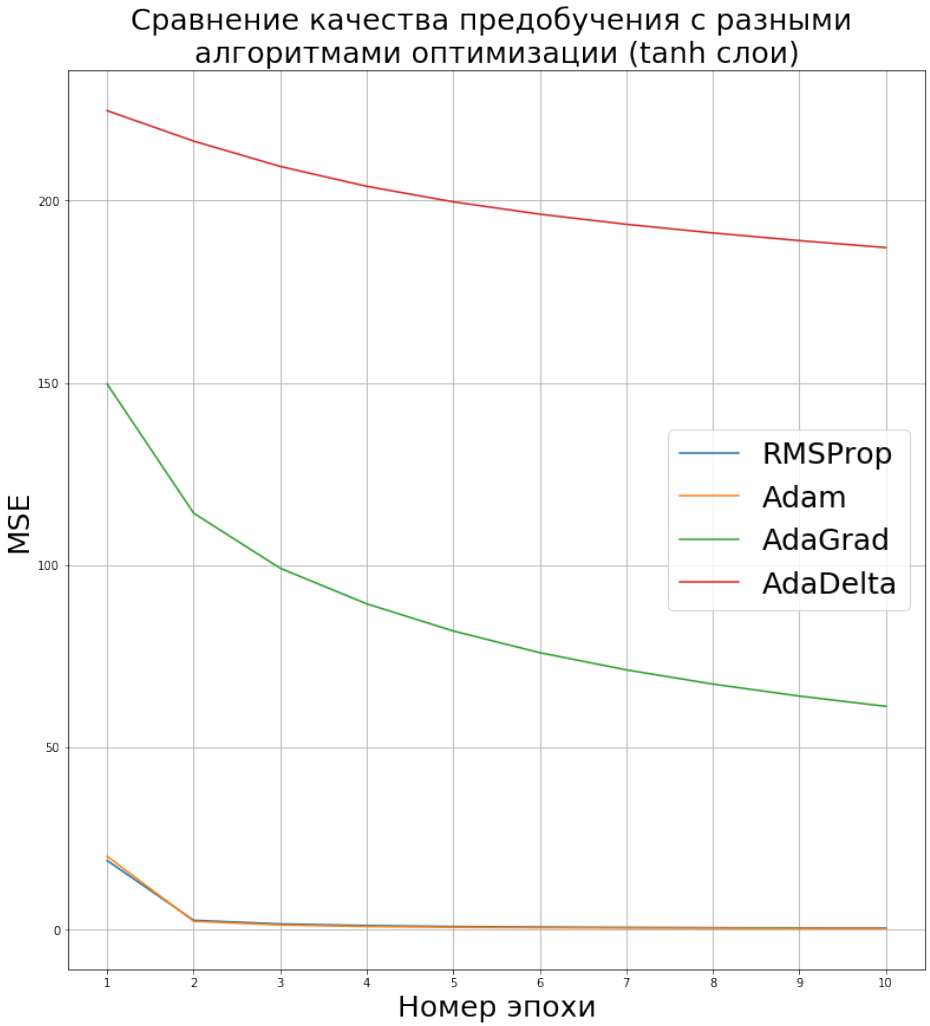
- Разработан алгоритм маршрутизации на основе мультиагентного обучения нейросетей с подкреплением – *DQN-routing*
- Сравнены три архитектуры нейросетей
- Алгоритм показал лучшее качество маршрутизации во время экспериментов по сравнению с link-state алгоритмом (доминирующее семейство алгоритмов в сетевом роутинге) и алгоритмом Q-routing (табличное обучение с подкреплением)
- Алгоритм продемонстрировал способность эффективной работы в условиях модели конвейерной системы с учетом энергопотребления.

Спасибо за внимание!

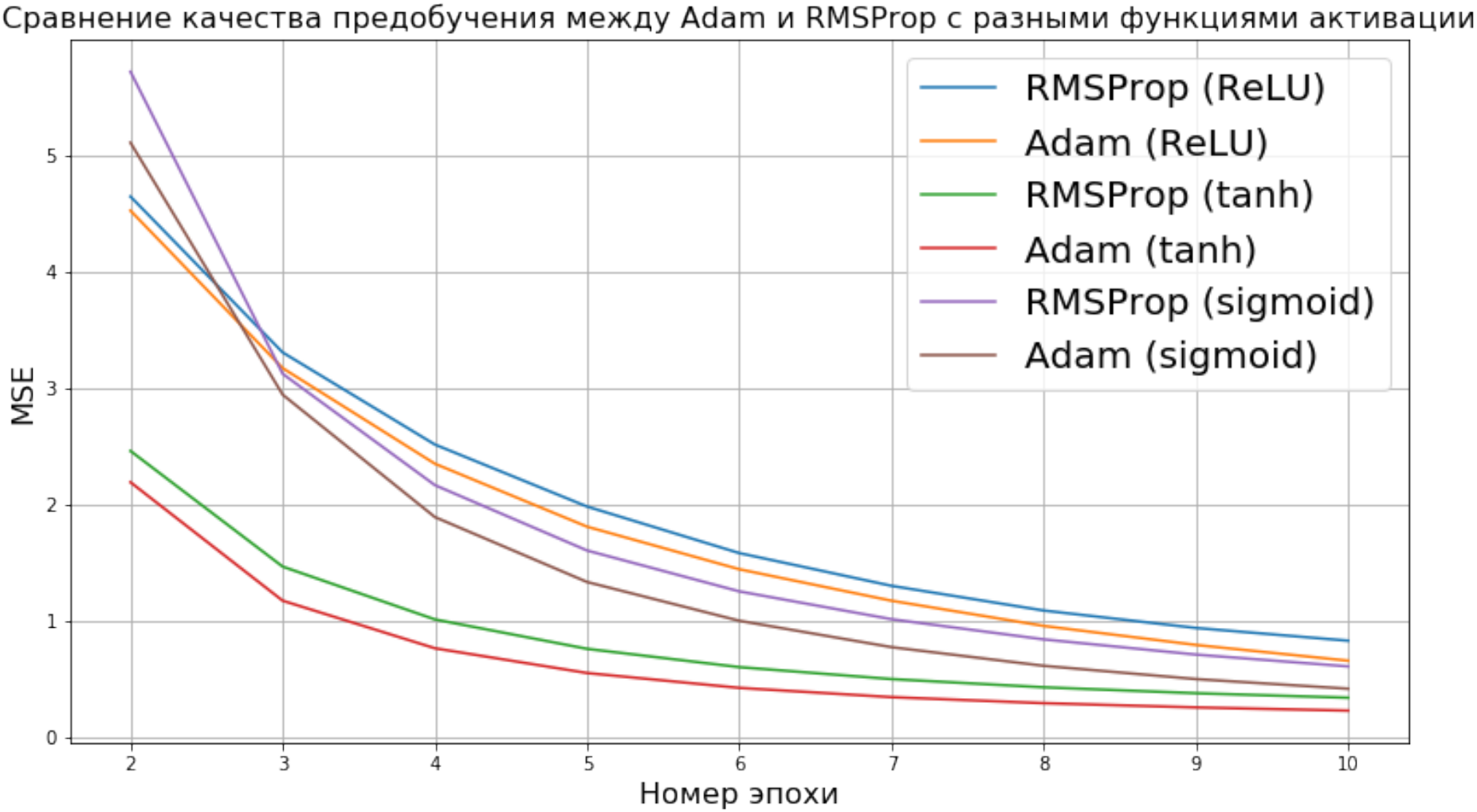
Иллюстрация несходимости алгоритма к оптимуму без предобучения

Сравнение алгоритмов оптимизации: скорость предобучения

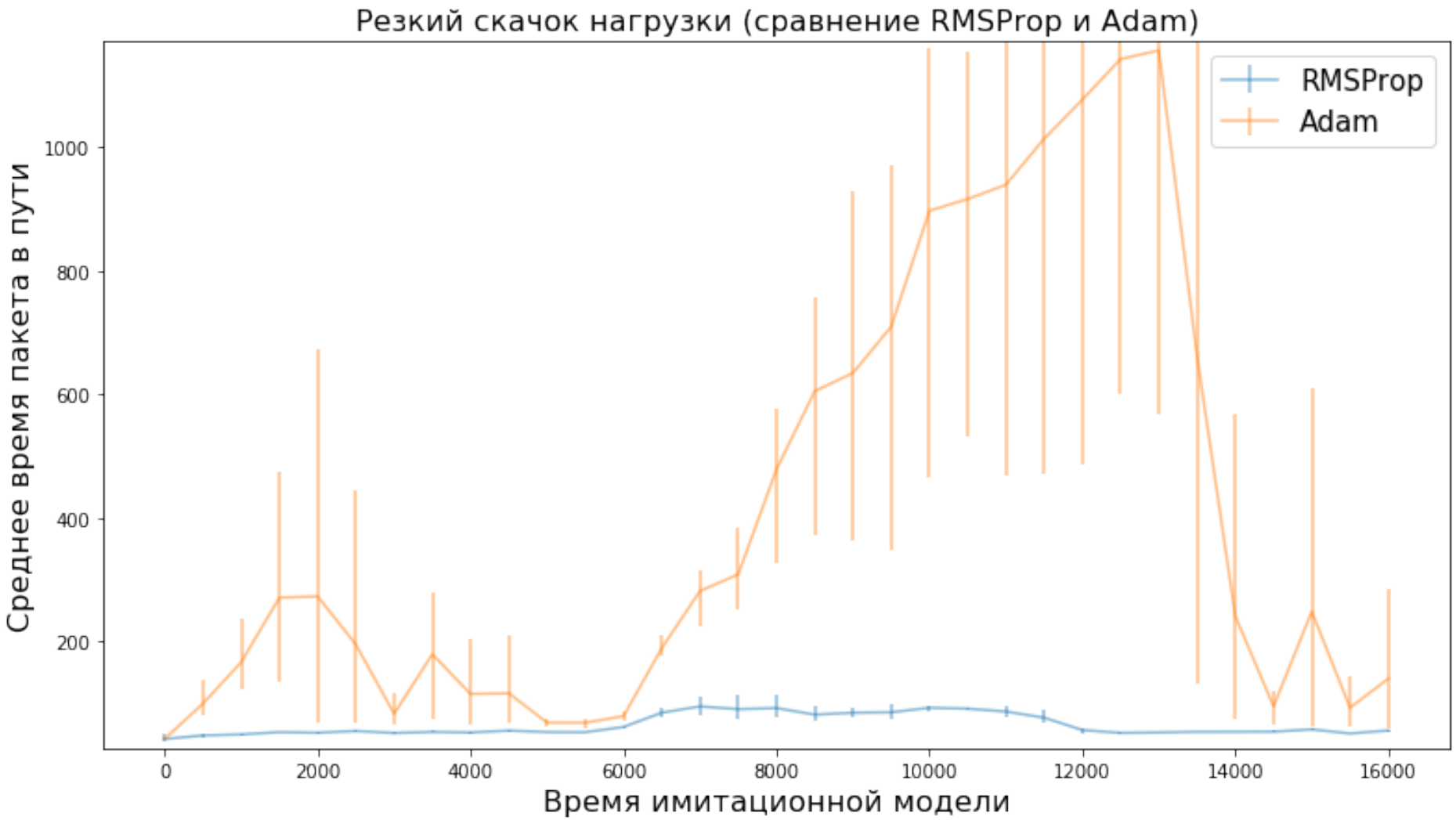




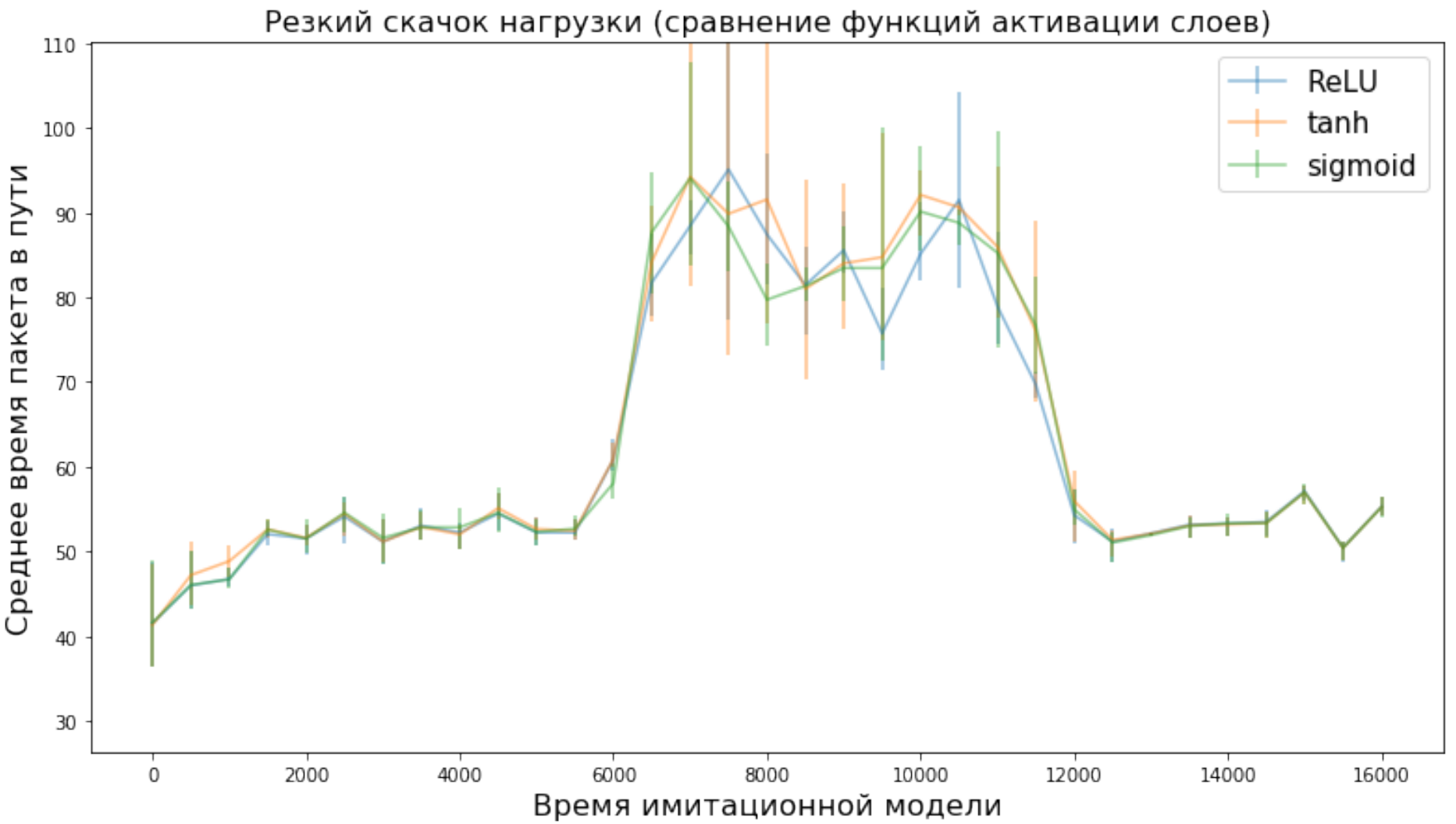
Сравнение алгоритмов оптимизации и функций активации: скорость предобучения



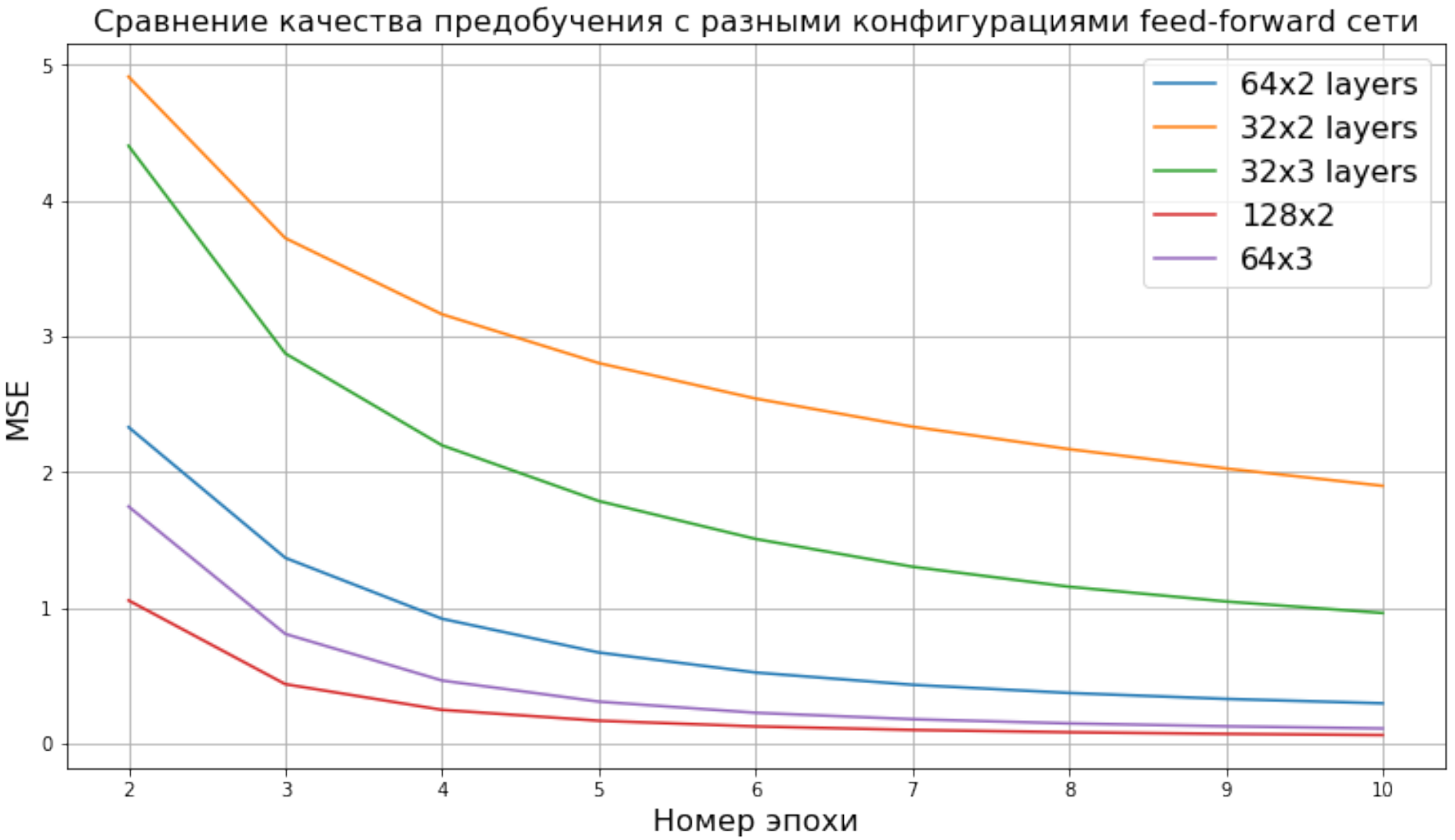
Сравнение RMSProp и Adam в модели компьютерной сети



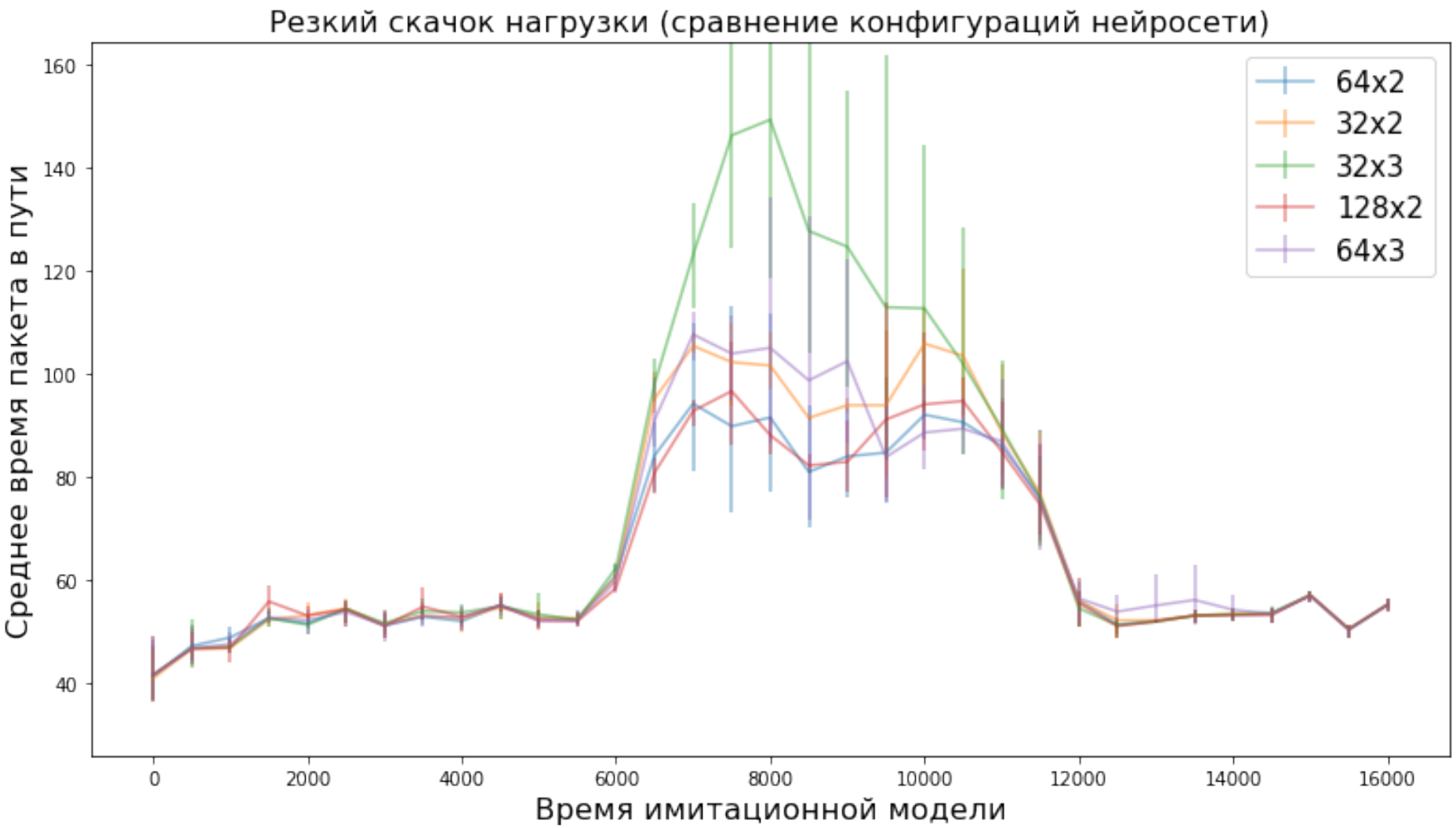
Сравнение фукнций активации с RMSProp в модели компьютерной сети



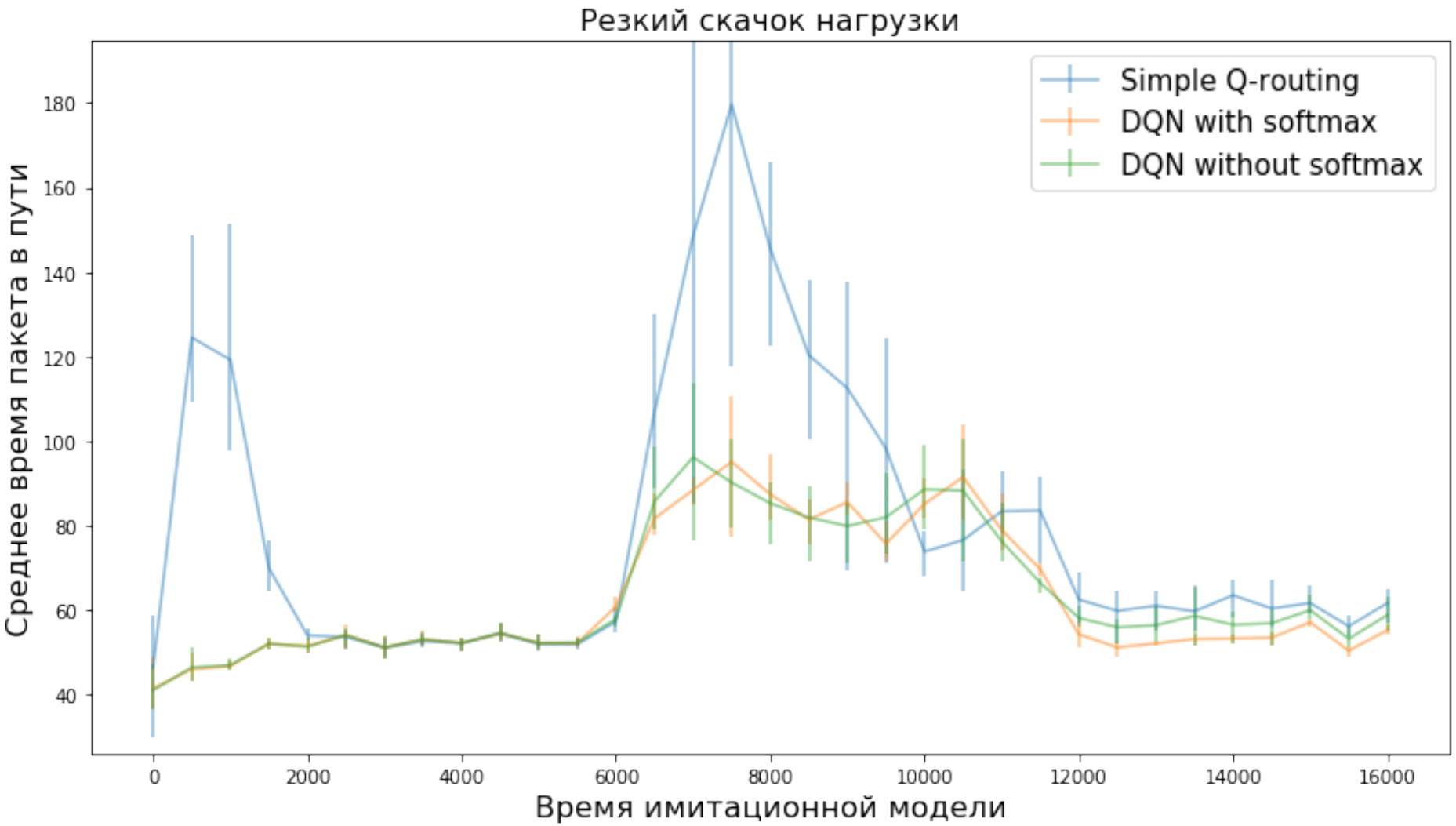
Сравнение различных конфигураций feed-forward нейросети по качеству предобучения



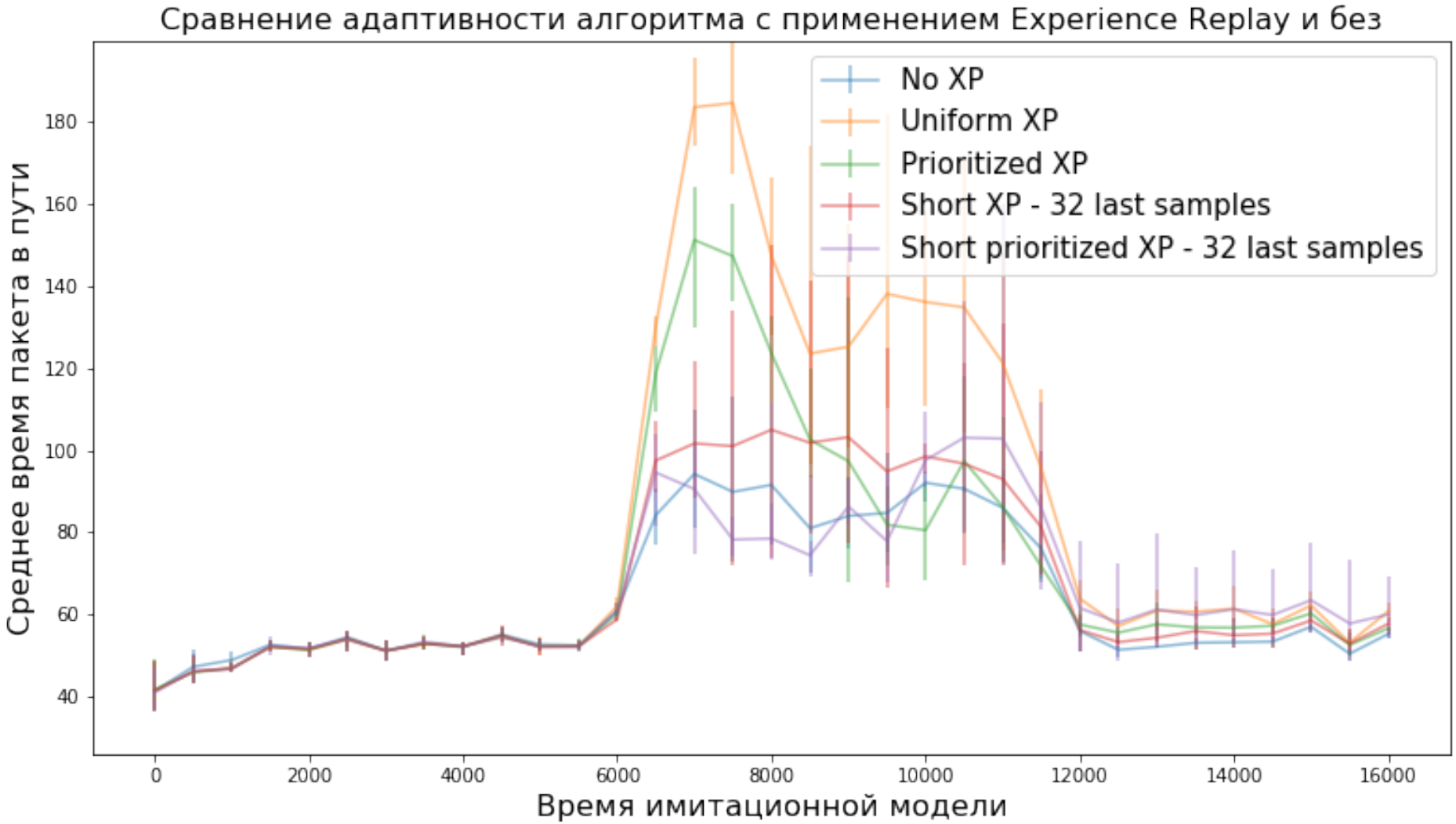
Сравнение различных конфигураций feed-forward нейросети при работе в имитационной модели



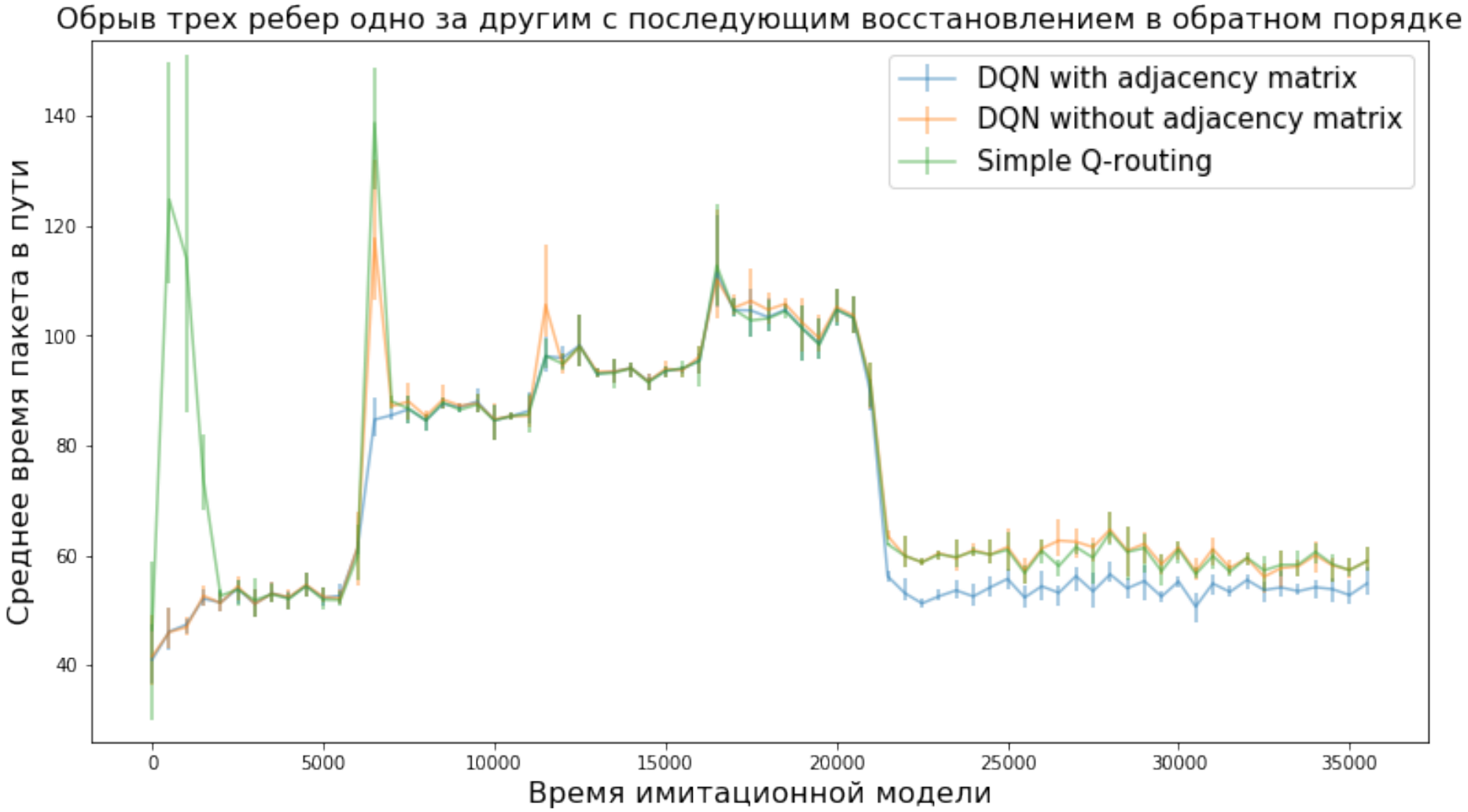
Влияние softmax-стратегии на работу алгоритма



Влияние различных видов experience replay на работу алгоритма



Влияние включения матрицы смежности в наблюдаемое состояние



Влияние включения информации о состоянии соседних конвейеров в наблюдаемое состояние