

**УДК 004.8**

*Бербасов В.Д.*

*Студент магистратуры*

*Научный руководитель: Назаренко Т.Т., к.т.н., доцент*

*Поволжский государственный университет телекоммуникаций и информатики*

## **СРАВНИТЕЛЬНЫЙ ОБЗОР БИБЛИОТЕК НЕЙРОННЫХ СЕТЕЙ KERAS И PYTORCH**

*Аннотация. В этой обзорной статье рассматриваются отличительные черты библиотек нейронных сетей Keras и PyTorch, разъясняются их сильные стороны и варианты использования в интересах разработчиков и исследователей. В среде, где библиотеки нейронных сетей имеют решающее значение для современных моделей глубокого обучения, Keras и PyTorch стали выдающимися платформами. В статье рассматривается удобный дизайн и совместимость Keras с различными серверными движками, а также множество предварительно созданных слоев и настраиваемость, ориентированная как на новичков, так и на опытных пользователей. В статье представлен сравнительный анализ двух библиотек с акцентом на простоту Keras и исследовательскую направленность PyTorch.*

*Ключевые слова. Библиотеки нейронных сетей, Keras, PyTorch, модели глубокого обучения, совместимость, предварительно созданные слои, сравнительный анализ, интеграция, Python, Numpy.*

*Berbasov V.D.*

*Master's student*

*Scientific adviser: Nazarenko T.T., Cand. Sc. (Technology), associate professor*

*Volga State University of Telecommunications and Informatics*

## COMPARATIVE REVIEW OF KERAS AND PYTORCH NEURAL NETWORK LIBRARIES

*Annotation. This overview article explores the distinguishing features of the Keras and PyTorch neural network libraries, explaining their strengths and their use cases for the benefit of developers and researchers. In an environment where neural network libraries are critical to modern deep learning models, Keras and PyTorch have become prominent platforms. The article discusses the user-friendly design and compatibility of Keras with various back-end engines, as well as many pre-built layers and customization aimed at both beginners and advanced users. The article presents a comparative analysis of the two libraries with an emphasis on the simplicity of Keras and the research focus of PyTorch. Keywords. Neural Network Libraries, Keras, PyTorch, Deep Learning Models, Compatibility, Prebuilt Layers, Benchmarking, Integration, Python, Numpy.*

Библиотеки нейронных сетей являются важными компонентами современных моделей глубокого обучения, предоставляя необходимые инструменты для построения и обучения сложных нейронных сетей. Среди множества доступных вариантов Keras и PyTorch стали двумя ведущими платформами, каждая из которых привлекла популярность в сообществе машинного обучения. Эта обзорная статья направлена на то, чтобы углубиться в отличительные особенности этих библиотек, выделить их сильные стороны и идеальные варианты использования, чтобы разработчики и исследователи могли принимать обоснованные решения при выборе библиотеки, наиболее подходящей для их проектов.

Keras, основанная Франсуа Шоле, быстро приобрела известность благодаря удобному дизайну и интуитивно понятному интерфейсу. Являясь высокоуровневым API, Keras упрощает процесс построения и экспериментирования с нейронными сетями. Одним из заметных

преимуществ Keras является его совместимость с несколькими серверными движками, такими как TensorFlow и Theano[1].

Библиотека предлагает два основных варианта API для построения нейронных сетей: последовательный API и функциональный API. Последовательный API без проблем работает для создания простых линейных стеков слоев, что делает его отличным выбором для начинающих.

Keras выделяется своим удобным синтаксисом. Такая простота использования значительно сокращает время и усилия, необходимые для быстрого прототипирования и экспериментов с моделями. Кроме того, Keras может похвастаться богатой коллекцией предварительно созданных слоев, оптимизаторов и функций потерь, что упрощает реализацию современных архитектур без необходимости обширного кодирования.

Для разработчиков, нуждающихся в дополнительной настройке, Keras позволяет создавать собственные слои и обратные вызовы, предоставляя пользователям гибкость в адаптации библиотеки в соответствии со стоящей перед исследователем задачи.

PyTorch, разработанный исследовательской лабораторией искусственного интеллекта Facebook (FAIR), завоевал значительную популярность в сообществе машинного обучения, в первую очередь благодаря динамическому графу вычислений. Динамический характер PyTorch обеспечивает более гибкое и эффективное построение моделей, позволяя пользователям легко изменять свои модели во время выполнения. Динамический граф вычислений также упрощает процесс отладки и способствует более интуитивно понятному процессу разработки модели.

Фреймворк создан на базе Torch, библиотеки для языка Lua, которая предназначена для математических расчетов и машинного обучения. Но его особенность — следование стилю Python и философии программирования на этом языке[2].

Центральное место в мощности PyTorch занимает его система тензорных вычислений и автоматического дифференцирования, известная как autograd. Как исследователи, так и разработчики высоко оценивают императивный стиль программирования PyTorch, обеспечивающий интуитивно понятное выполнение кода и больший контроль над компонентами модели. Эта функция особенно привлекательна для исследователей и разработчиков, которым требуется детальный контроль над внутренней работой своих моделей.

Бесшовная интеграция с Python и NumPy облегчает манипулирование данными и их анализ, что делает PyTorch фаворитом в исследовательских средах, где предварительная обработка и анализ данных имеют решающее значение.

И у Keras, и у PyTorch есть свои сильные стороны, отвечающие различным предпочтениям пользователей и требованиям проекта. Keras отличается простотой и легкостью использования, что делает его отличной отправной точкой для начинающих и отличным выбором для сценариев быстрого прототипирования. Его обширные предварительно созданные модули и согласованный дизайн API обеспечивают оптимизированный процесс разработки без ущерба для производительности.

И наоборот, динамический граф вычислений PyTorch и парадигма императивного программирования делают его очень привлекательным для исследовательского сообщества. Исследователи могут итеративно изменять свои модели, что ускоряет эксперименты и упрощает отладку. Детальный контроль над компонентами модели делает PyTorch оптимальным выбором для исследовательских задач, особенно в академических условиях.

В то время как Keras имеет преимущество благодаря удобному синтаксису, плавная интеграция PyTorch с Python и NumPy еще больше укрепляет его позиции в академическом сообществе.

В заключение, Keras и PyTorch представляют собой две отличные библиотеки нейронных сетей, каждая со своим набором преимуществ и идеальными вариантами использования. Простота использования и универсальность Keras делают его отличным выбором для начинающих и быстрого прототипирования, а динамические вычисления PyTorch и возможности, ориентированные на исследования, делают его предпочтительным вариантом для разработчиков и академических сред.

У обеих библиотек имеется обширное международное сообщество с открытыми форумами, где начинающие и разработчики могут что-либо узнать или дать какой-либо совет [3].

В конечном счете, разработчики и исследователи должны учитывать свои требования к проекту и желаемый уровень контроля при выборе между Keras и PyTorch. Обе библиотеки продолжают развиваться разработчиками, принося новые изменения для улучшения качества выполнения стоящих перед пользователями задач.

#### **Использованные источники:**

1. Keras: библиотека глубокого обучения на Python. Вы только что открыли для себя Keras. [Электронный ресурс] URL: <https://ru-keras.com/home/> (дата обращения: 10.08.2023)
2. Стивенс, Э. PyTorch. Освещающая глубокое обучение [Текст]: профессиональная литература / Э. Стивенс, Т. Виман, Л. Антига.: Питер, 2022. - 576 с. - (Профессиональная литература).
3. Welcome to the TensorFlow Forum [Электронный ресурс] Режим доступа: <https://discuss.tensorflow.org/c/keras/16/none> (дата обращения: 10.08.2023)