

Причины выбора Python как язык программирования нейросетей

Савостьянов Артур Гарриевич,

бакалавр

Владивостокский государственный университет

Россия. Владивосток

E-mail: steelbeat13356@mail.ru; Тел. +79940227261

ул. Гоголя, 41, г. Владивосток, Приморский край, Россия, 690014

В статье исследуются причины популярности языка программирования Python в сфере разработки нейросетей, машинного обучения и искусственного интеллекта. Анализируются факторы, делающие Python предпочтительным для специалистов, обсуждаются библиотеки TensorFlow и PyTorch. Выделены удобство использования и широкое сообщество разработчиков. Python — неотъемлемая часть создания инновационных нейросетей.

Ключевые слова: *нейронные сети, TensorFlow, PyTorch, инструменты глубокого обучения, Python.*

Reasons for choosing Python as a neural network programming language

The article examines the reasons for the popularity of the Python programming language in the field of neural network development, machine learning and artificial intelligence. The factors that make Python preferable for specialists are analyzed, and the TensorFlow and PyTorch libraries are discussed. Usability and a broad community of developers are highlighted. Python is an integral part of creating innovative neural networks.

Keywords: *Neural networks, TensorFlow, PyTorch, deep learning tools, Python.*

Python, язык программирования с открытым исходным кодом, стал неоспоримым лидером в области разработки нейросетей, машинного обучения и искусственного интеллекта. Сегодня мы погружаемся в мир причин, сделавших Python предпочтительным инструментом для специалистов в этой области.

Рост интереса к Python в сфере нейросетей можно объяснить его простотой и эффективностью в разработке сложных моделей. Не только опытные программисты, но и начинающие исследователи могут легко изучить Python и начать создавать свои нейросети. Гибкость и мощь этого языка позволяют создавать разнообразные модели, от простых нейронных сетей до сложных глубоких архитектур.

Объектом исследования является язык программирования Python и его библиотеки, предназначенные для разработки нейросетей. Целью данной работы является выявление причин, по которым Python является предпочтительным выбором языка программирования при разработке систем с участием нейросетей. Задачами работы являются анализ инструментов языка программирования Python и их сравнение с другими языками программирования, используемыми для разработки нейросетей. Используемые методы теоретического исследования: метод анализа, системный подход.

Одним из ключевых преимуществ Python является его обширная экосистема библиотек, специально разработанных для работы с нейросетями и глубоким обучением. В частности, библиотеки TensorFlow и PyTorch стали стандартом в индустрии и обеспечивают широкие возможности для разработки и оптимизации нейронных сетей.

Разработанная командой Google Brain, TensorFlow стала одной из самых популярных библиотек для создания и обучения нейронных сетей. Ее широкие возможности включают в себя:

1. TensorFlow работает на основе графов вычислений, где узлы представляют операции, а ребра — данные. Это позволяет оптимизировать и распараллеливать вычисления;

2. TensorFlow предоставляет множество инструментов для создания различных видов нейросетей, включая свёрточные сети, рекуррентные сети, а также генеративно-сопоставительные сети (GANs).

Центральным понятием в TensorFlow являются тензоры — многомерные массивы, используемые для хранения данных. Библиотека также обеспечивает автоматическое дифференцирование, что упрощает создание и обучение моделей. Пример создания нейронной сети в TensorFlow представлен на рисунке 1.

```
import tensorflow as tf

# Определение модели
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)), # Преобразование двумерного изображения в одномерный массив
    tf.keras.layers.Dense(128, activation='relu'), # Полносвязный слой с 128 нейронами и функцией активации ReLU
    tf.keras.layers.Dense(10, activation='softmax') # Выходной слой с 10 нейронами (по одному для каждой цифры) и функцией активации softmax
])

# Компиляция модели
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Вывод информации о модели
model.summary()
```

Рис.1. Пример создания простой нейросети

Конечно, до полноценной нейронной сети не хватает множество параметров, но на данном этапе нейросеть является работоспособной. На рисунке 2 представлена схема нейронной сети распознавания цифр, состоящей из трех слоев: входного, скрытого и выходной слой.

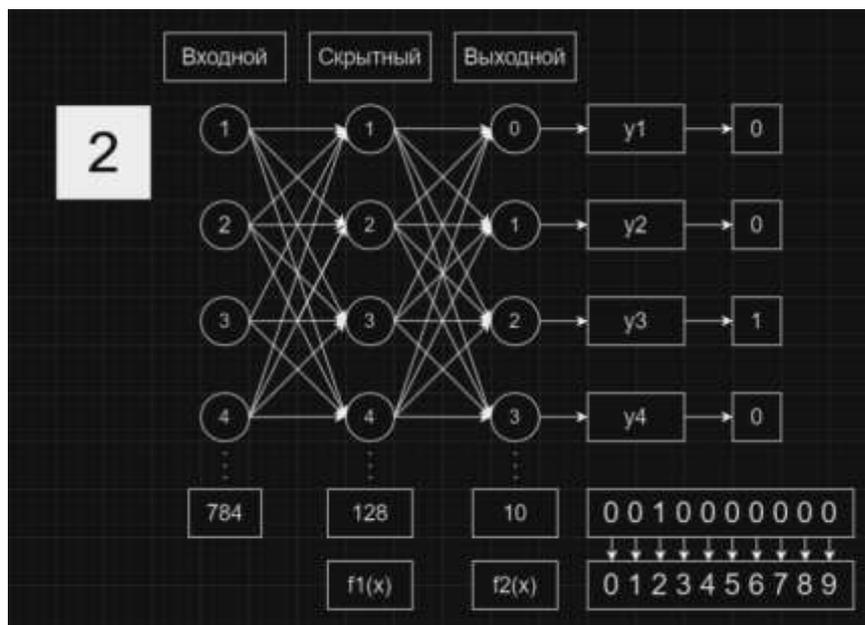


Рис.2. Схема нейросети

Входной слой «*tf.keras.layers.Flatten(input_shape=(28, 28))*» преобразует входные данные из двумерного массива 28x28 в одномерный массив 784. Это необходимо для подачи данных на вход полносвязному слою нейронной сети.

Скрытый слой «*tf.keras.layers.Dense(128, activation='relu')*» с 128 нейронами с функцией подачи данных на вход полносвязному слою нейросети.

Выходной слой «*tf.keras.layers.Dense(10, activation='softmax')*» с 10 нейронами (по одному для каждого класса, в данном случае для цифр от 0 до 10) и функцией активации softmax. Softmax преобразует выходы нейронов в вероятности принадлежности к каждому классу.

Pytorch, разработанный Facebook, стал очень популярным инструментом для исследователей и разработчиков. Он выделяется своей простотой использования и гибкостью. Основные возможности PyTorch включают в себя:

1. Динамические вычислительные графы: в отличие от TensorFlow, PyTorch использует динамические вычислительные графы. Это означает, граф строится во время выполнения программы, что делает его более гибким и удобным для экспериментов;
2. Легкость в использовании: Pytorch предлагает простой и интуитивно понятный API, который позволяет быстро создавать и обучать модели нейросетей;
3. Автоматическое дифференцирование: как и в TensorFlow, Pytorch обеспечивает автоматическое дифференцирование, что упрощает создание и оптимизацию моделей.

Обе библиотеки имеют свои преимущества и подходы к созданию нейронных сетей. TensorFlow обычно рекомендуют для производственного использования и масштабируемых проектов, в то время как PyTorch обладает преимуществами в исследованиях и прототипировании моделей благодаря своей простоте и гибкости.

На рисунке 2 также показан пример выхода нейронной сети для классификации цифры «2». Нейроны выходного слоя активируются в соответствии с вероятностями принадлежности к каждому классу (цифрам от 0 до 9). Например, вероятность принадлежности к классу «2» выше всего, поэтому соответствующий нейрон имеет наибольшее значение.

Рассмотрим языки программирования Java и R, которые также имеют свои преимущества и недостатки при работе с нейросетями.

Java, хотя и не настолько популярен в области нейросетей, имеет свои преимущества: Java работает на различных платформах, что делает его удобным выбором для разработки кросс-платформенных приложений нейросетей, в некоторых случаях может показывать более высокую производительность благодаря своей виртуальной машине и оптимизации исполнения, строгая типизация может помочь выявлять ошибки в кое на более ранних этапах разработки.

R, язык программирования и среда для статических вычислений и визуализации, также используется в области машинного обучения и нейросетей: обладает обширными библиотеками для статического анализа данных, что делает его хорошим выбором для предобработки данных в нейросетях, предлагает мощные инструменты для визуализации данных, что может быть полезно при анализе результатов обучения нейросетей, с появлением пакетов, таких как «keras» и «tensorflow» для R, он стал более привлекательным для разработки нейросетей. На рисунке 3 приведено сравнение Python с другими языками программирования, которые можно использовать для разработки нейросетей.

Факторы	Python	Java	R
Простота Использования	+	-	+/-
Библиотеки Для Глубокого Обучения	+	-	+/-
Производительность	+/-	+	-
Визуализация	+/-	-	+
Сообщество Разработчиков	+	+/-	+/-

Рис.3. Сравнение разных языков программирования нейросетей

Python остаётся лидирующим языком программирования в области нейросетей и машинного обучения благодаря своей простоте, обширным библиотекам и активному сообществу. Java может быть хорошим выбором для кроссплатформенных проектов, а R – для предобработки данных и визуализации.

Python предоставляет разнообразные инструменты для разработки нейросетей. Один из наиболее популярных инструментов – Jupyter Notebook, который обеспечивает создание и выполнение кода в интерактивном окружении, что особенно ценно для экспериментов и прототипирования моделей. Кроме того, популярными IDE для работы с Python являются PyCharm, Spyder и Visual Studio Code, которые предоставляют широкий спектр функциональных возможностей для разработки и отладки.

Python часто применяется для создания веб-приложений, включая те, которые используют нейронные сети. Фреймворки веб-разработки, такие как Flask и Django, позволяют разработчикам создавать и развертывать веб-сервисы для обучения и инференса нейронных сетей. Flask обладает минимальным набором инструментов для быстрой разработки веб-приложений, в то время как Django обеспечивает полноценный стек технологий для создания сложных веб-приложений.

Множество облачных провайдеров, таких как Amazon Web Services (AWS), Microsoft Azure и Google Cloud Platform (GCP), предоставляют облачные сервисы для обучения и развертывания нейронных сетей. Python часто используется для взаимодействия с этими сервисами через соответствующие SDK и API. Например, TensorFlow и PyTorch предоставляют интеграцию с облачными сервисами для обучения моделей на удаленных вычислительных ресурсах и развертывания моделей в облачной среде.

Python-экосистема также включает инструменты для контейнеризации приложений, такие как Docker, и управление версиями кода, например, Git. Эти инструменты могут быть полезны при развертывании приложений, особенно при работе с нейронными сетями.

Python имеет огромное и активное сообщество разработчиков, которые постоянно вносят вклад в его развитие. Это означает, что всегда можно найти поддержку, решение проблем и ответы на вопросы на различных форумах, в блогах, социальных сетях и других ресурсах.

Python предлагает обширный набор библиотек и инструментов для работы с нейросетями и машинным обучением. Например, TensorFlow, PyTorch, Keras, scikit-learn, NumPy, Pandas и многие другие. Эти библиотеки обеспечивают широкий спектр функциональности, начиная от создания моделей и обучения до визуализации и оценки результатов. Существует поддержка Python в различных облачных платформах, базах данных, веб-серверах и других сервисах.

Многие ключевые инструменты и библиотеки Python для работы с нейросетями распространяются под лицензией открытого исходного кода. Это способствует активному развитию и совершенствованию инструментов благодаря участию сообщества в их разработке.

Таким образом, Python имеет весь необходимый инструментарий для разработки нейросетей, легко интегрируется с другими технологиями и платформами, его библиотеки пользуются популярностью в сообществе разработчиков, которые оказывают им активную поддержку и обеспечивают стремительное развитие, что делает Python отличным решением для разработки комплексных систем на базе нейросетей.

1. Jethro Jeff – Ignite Your Creative Spark: [Электронный ресурс]. – Режим доступа: <https://jethrojeff.com/>;

2. Advantages of Python as an AI and ML Development Language: [Электронный ресурс]. – Режим доступа: <https://dzone.com/articles/advantages-of-python-as-an-ai-and-ml-development-l>;

3. Why use Python for Machine learning and AI: [Электронный ресурс]. – Режим доступа: <https://medium.com/@asif.choudhary/why-use-python-for-machine-learning-and-ai-b215bc7e2a11>.