**НЕЙРОСЕТЬ**

**Нейронная сеть –** это разновидность машинного обучения, при котором компьютерная программа работает по принципу человеческого мозга, используя различные нейронный связи.

**По какому принципу работает нейронная сеть**

* В нейросеть загружается некоторое количество конкретных, необходимых для эксперимента или исследования, данных.
* Информация передается с помощью искусственных синапсов от искусственного нейрона к нейрону, от слоя к слою, каждый нейрон может иметь несколько входящих синапсов с данными.
* Данные, полученные каждым нейроном, представляют собой сумму всех данных, умноженных на коэффициент веса каждого искусственного синапса.
* Полученные значения формируют выходные сигналы, которые передаются до тех пор, пока информация не достигнет конечного выхода.

## Виды нейронных сетей

Все нейронные сети можно разделить на несколько видов: однослойные, многослойные, прямого распространения, рекуррентные.

* Однослойные сети сразу же выдают результат после загрузки в них некоторого массива данных.
* Многослойные сети прогоняют вводную информацию через несколько промежуточных слоев и принципом своей работы больше напоминают биологическую нейронную сеть. Выходная информация получается после прохождения всех слоев, на которых происходит обработка и анализ.
* Сети прямого распространения чаще всего используются для распознавания образов, классификации и кластеризации данных — они направлены в одну сторону и не умеют перенаправлять информацию обратно. Ввели данные — получили ответ.
* Рекуррентные сети перенаправляют информацию туда и обратно, пока не получат конечный результат. Они используют эффект кратковременной памяти, на основании которого информация дополняется и восстанавливается. Такие сети чаще используются для прогнозирования.

## Область применения

Перечисленными сферами использование нейросетей не ограничивается, есть и другие существующие и перспективные способы задействовать их для решения различных задач:

* Машинное обучение является одной из разновидностей искусственного интеллекта. Google, Яндекс, Бинг, Байду активно применяют machine learning для повышения релевантность результатов запросам пользователей. Алгоритмы самообучаются, опираясь на миллионы однотипных фраз, вводимых в поисковую строку.
* Для нормального функционирования роботов необходимо разрабатывать множество алгоритмов, и здесь не обойтись без нейросетей.
* Возможности ИНС используются архитекторами компьютерных сетей, чтобы справиться с проблемой параллельных вычислений.
* В математике нейронные сети позволяют быстрее решать сложные задачи.

## Достоинства:

* способность игнорировать постороннюю информацию. Нейросети после обучения ведут себя так: убирают лишнюю информацию, не имеющую отношения к поставленной задаче;
* возможность сохранять работоспособность в случае утраты отдельных элементов, т.е. повреждение некоторых компонентов не мешает выдавать верный результат;
* высокая скорость работы. Благодаря тому, что ИНС состоит из тысяч микропроцессоров, взаимодействующих между собой, задачи решаются намного быстрее, чем стандартными способами;

**Недостатки:**

* предлагаемый ИНС ответ не будет абсолютно точным, только примерным. Следовательно, нельзя полностью полагаться на результаты работы нейросети, поскольку всегда есть вероятность недостоверного решения;
* каждый искусственный нейрон действует независимо от соседних, он не соотносит свое поведение с другими микропроцессорами. Специфика ИНС заключается в том, что нет гарантии абсолютной правдивости результата.

## МЕТОДЫ ПРОГРАММИРОВАНИЯ

## Процедурное программирование

Процедурный код - это тот, который непосредственно инструктирует устройство о том, как выполнить задачу в логических шагах. Эта парадигма использует линейный нисходящий подход и рассматривает данные и процедуры как две разные сущности. Основываясь на концепции вызова процедуры, процедурное программирование делит программу на процедуры, которые также известны как подпрограммы или функции, просто содержащие ряд шагов, которые необходимо выполнить.

#### **Преимущества:**

* процедурное программирование отлично подходит для программирования общего назначения;
* простота кодирования наряду с простотой реализации компиляторов и интерпретаторов;
* большое разнообразие книг и материалов онлайн-курсов, доступных по проверенным алгоритмам, что упрощает процесс обучения;
* исходный код переносим, поэтому его можно использовать и для другого процессора;
* код можно повторно использовать в разных частях программы без необходимости его копирования;
* благодаря технике процедурного программирования потребность в памяти также сокращается;
* поток программы можно легко отслеживать.

#### **Недостатки:**

* программный код сложнее писать, когда используется процедурное программирование;
* процедурный код часто не может быть повторно использован, что может привести к необходимости воссоздания кода, если он необходим для использования в другом приложении;
* сложно связать с объектами реального мира;
* важность придается операции, а не данным, которые могут создавать проблемы в некоторых случаях, чувствительных к данным;
* данные доступны для всей программы, что делает ее не такой безопасной.

## Объектно-ориентированное программирование

ООП - это инкапсуляция, идея которой заключается в том, что каждый объект, содержащий программу, является самодостаточным, что означает, что все компоненты, составляющие объект, находятся внутри самого объекта. Теперь, поскольку каждый модуль в рамках этой парадигмы является самодостаточным, объекты могут быть взяты из одной программы и использованы для решения другой проблемы практически без изменений.

#### **Преимущества:**

* благодаря модульности и инкапсуляции ООП обеспечивает простоту управления;
* ООП имитирует реальный мир, упрощая его понимание;
* поскольку объекты являются целыми внутри себя, их можно повторно использовать в других программах.

#### **Недостатки:**

* объектно-ориентированные программы, как правило, работают медленнее и занимают большой объем памяти;
* чрезмерное обобщение;
* создание программ, построенных с использованием этой парадигмы, может занять больше времени.

## Функциональное программирование

Функциональное [программирование - это парадигма](https://translated.turbopages.org/proxy_u/en-ru.ru.3a061c31-63ef666f-fbf24d79-74722d776562/https/hackr.io/blog/programming-paradigms) программирования, в которой составление функций становится основной движущей силой разработки. Это декларативный тип стиля программирования, который фокусируется на том, что решать, а не на том, как решать.

#### **Преимущества:**

* функциональное программирование предлагает защищенную среду;
* многие другие языки требуют значительного объема информации для правильного выполнения операций, функциональное программирование устраняет необходимость в большом объеме кода, необходимого для определения состояний;
* поскольку эта парадигма зависит только от входных аргументов, побочных эффектов нет.

#### **Недостатки:**

* использование функционального программирования исключительно в коммерческой разработке программного обеспечения не рекомендуется;
* требует большого объема памяти и времени;
* может оказаться менее эффективным, чем другие парадигмы.

## ЯЗЫКИ ПРОГРАММИРОВАНИЯ

## Java

## Java – один из самых востребованных языков программирования в мире и один из двух официальных языков программирования, используемых в разработке Android. Язык Java, весьма востребовательный и способен создавать широкий спектр различных приложений, игр и инструментов.

**Преимущество:**

* мультифункциональность;
* достаточно простой синтаксис;
* независимость (код сможет работать на любой платформе, поддерживающей Java);
* надежность (благодаря строгой статической типизации);
* возможность параллельной разработки;
* хорошую организацию кода и его легкую поддержку;
* универсальность использования различных классов.

**Недостатки:**

* требуемое большое количество памяти;
* низкая скорость по сравнению с С и С++;
* отсутствие поддержки низкоуровневого программирования.

**JavaScript**

JavaScript – язык, который в своей основе содержит скрипт. Работать со скриптами сможет каждый браузер, что делает язык кроссплатформенным. Так же это мультипарадигменный язык. Он поддерживает несколько стилей одновременно.

**Преимущества:**

* наличие полной интеграции с версткой страниц и серверной частью;
* рациональность применения и простота. Решить элементарную задачу при помощи JS удается за несколько минут. Сам ЯП обладает понятным синтаксисом, который не требует углубленного изучения;
* скорость работы. Может сделать итоговое приложение более быстрым и удобным;
* производительность;
* комфортность использования пользовательских интерфейсов;
* наличие собственной мощной экосистемы (инфраструктуры).

**Недостатки:**

* отсутствие возможности чтения и загрузки документов. Основная причина наличия этого минуса – соображения безопасности;
* отсутствие удаленного доступа. Полноценно для сетевого ПО соответствующий язык не используется;
* нестрогая типизация;
* вольную трактовку типов. Нестыковки в коде будут игнорироваться;
* доступность для конкурентов. Связано это с высокой читаемостью исходного кода JavaScript.

## C++

## C++ — [компилируемый](https://programmers.fandom.com/ru/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) статически типизированный язык программирования общего назначения. Поддерживая разные парадигмы программирования, сочетает свойства как высокоуровневых, так и низкоуровневых языков.

## Достоинства:

* высокая совместимость с языком С;
* поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщенное программирование;
* имеется возможность работы на низком уровне с памятью, адресами, портами;
* возможность создания обобщённых контейнеров и алгоритмов для разных типов данных, их специализация и вычисления на этапе компиляции, используя шаблоны;
* эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы.

**Недостатки:**

* синтаксис, провоцирующий ошибки;
* препроцессор, унаследованный от С, очень примитивен;
* Плохая поддержка модульности. Подключение интерфейса внешнего модуля через препроцессорную вставку заголовочного файла (#include) серьёзно замедляет компиляцию при подключении большого количества модулей

## C#

## C# — это язык программирования от компании Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и т.д.

## **Преимущества:**

* строгую типизацию;
* сохранение концепций объектно-ориентированного программирования;
* функциональность;
* достаточно мощный инструментарий;
* стабильную работу через Visual Studio;
* компактный и легко читаемый код;
* понятный даже новичкам синтаксис.

## Недостатки:

* синтаксис, безграмотное применение которого провоцирует разного рода ошибки утилит;
* библиотеки, которые не подойдут для выполнения специфических задач;
* читаемость кода – конкуренты смогут с легкостью «разобрать» программное обеспечение и изучить его состав.

## Python

## Python — высокоуровневый язык программирования общего назначения. Он имеет минималистичный синтаксис и направлен на повышение читаемости кода (и в целом на повышение производительности разработчика).

**Преимущества:**

* низкий порог вхождения;
* язык широкого применения;
* минималистичный синтаксис;
* кроссплатформенность;
* открытый исходный код интерпретатора CPython;
* поддержка многих IDE;
* огромное количество библиотек;

**Недостатки:**

* низкая производительность;
* отсутствие статической типизации.

В плане метода программирования лучше использовать объектно-ориентированное программирования, так какможно ускорить процесс написания кода и сделать его более читаемым для дальнейшего обслуживания.

Если говорить о языках, которые однозначно необходимы в данной области, то стоит обратить внимание на Python. Потому что этот язык имеет хороший набор библиотек, предназначенных для оперативного создания программы с нейросетью.