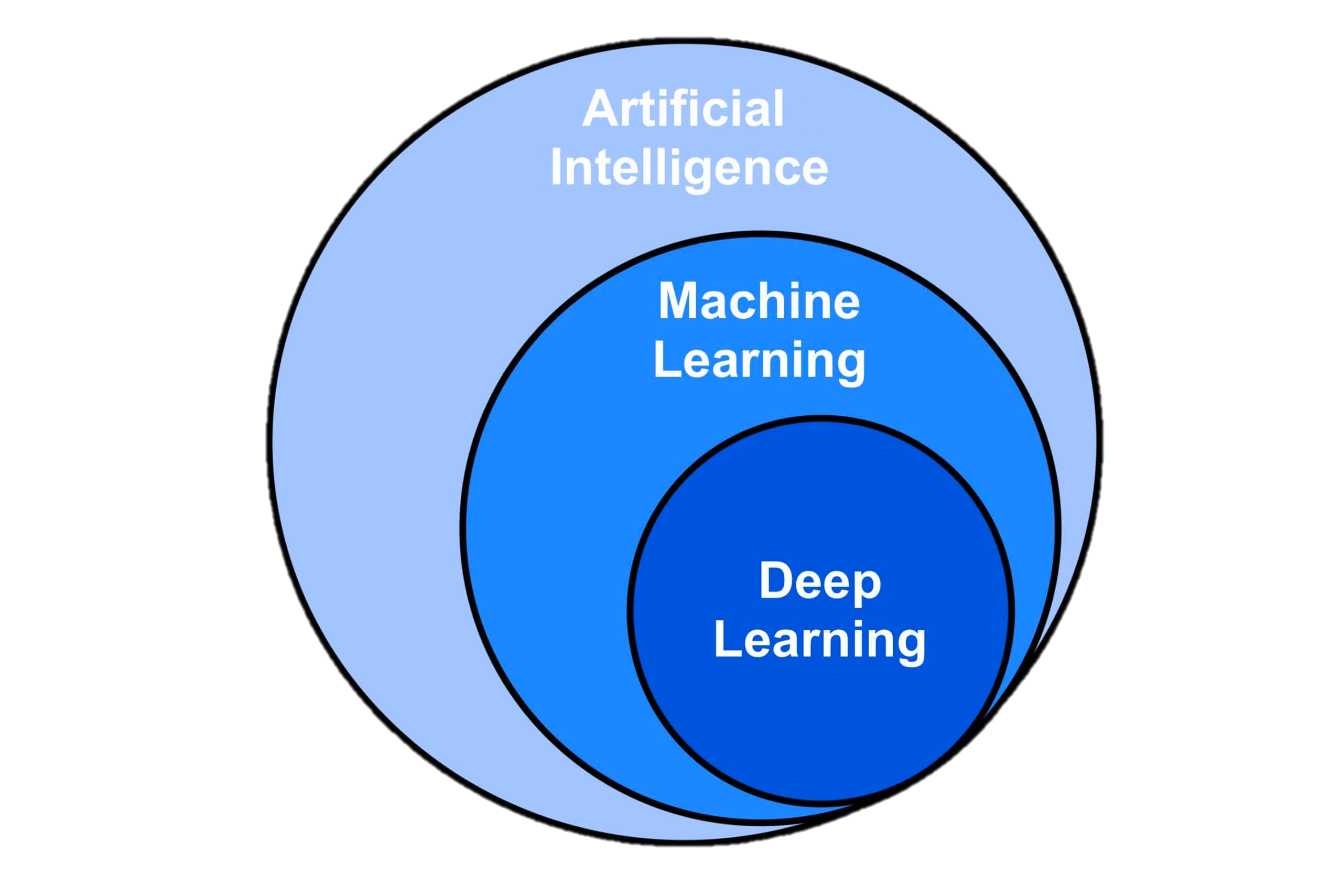
11. Инструментальные средства и языки программирования, применяемые для разработки систем искусственного интеллекта.



*Что такое ИИ?*

**Иску́сственный интелле́кт** — свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека; наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ.

**Искусственный общий интеллект** (**ИИ**) - это способность интеллектуального агента понимать или изучать любую интеллектуальную задачу, которую может выполнить человек.

Определение, дано Джоном Маккарти в 1956 году, Поясняя своё определение, Джон Маккарти указывает: «Проблема состоит в том, что пока мы не можем в целом определить, какие вычислительные процедуры мы хотим называть интеллектуальными. Мы понимаем некоторые механизмы интеллекта и не понимаем остальные. Поэтому под интеллектом в пределах этой науки понимается только вычислительная составляющая способности достигать целей в мире».

В основном ИИ – это Экспертные системы, помогающие принять решение.

Поэтому предлагаю от этого всеобъемлющего понятия перейти к более конкретным, а именно машинное обучение и входящие в него понятия.

**Машинное обучение**

Класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.



Машинное обучение (ML) — это использование математических моделей данных, которые помогают компьютеру обучаться без непосредственных инструкций. Оно считается одной из форм искусственного интеллекта (ИИ). При машинном обучении с помощью алгоритмов выявляются закономерности в данных.

При таком определении опять-же может показаться что подходят любые языки программирования. Так и есть, рассмотрим наиболее используемые.

**Языки:**

Сегодня можно предположить, что наиболее востребованным языком программирования в профессиональной области машинного обучения является Python. Затем идет Java, затем R и C ++.

Проведем небольшой сравнительный анализ этих языков в концепции машинного обучения

**R**

R – язык программирования с открытым исходным кодом, используемый в основном для статистических вычислений. За последние годы он стал популярным и пользуется большим успехом у ученых. R обычно не используется в промышленных производственных средах, но его популярность в промышленных приложениях растет из-за повышенного интереса к науке о данных.

Популярные пакеты для машинного обучения в R:

* caret (сокращение от Classification And REgression Training), библиотека для создания прогностических моделей.
* randomForest, пакет для классификации и регрессии.
* e1071, пакет, предоставляющий функции статистики и теории вероятностей.

**C++**

Язык C++ обычно используется для машинного обучения и искусственного интеллекта в игровых или роботизированных приложениях (в том числе для передвижения роботов). Разработчики встроенных вычислительных аппаратов и инженеры-электроники предпочитают использовать C++ или C в приложениях для машинного обучения благодаря их гибкости и предлагаемому уровню контроля.

Популярными библиотеками машинного обучения C++ являются масштабируемый пакет mlpack, Dlib с широким набором алгоритмов машинного обучения и модульная библиотека с открытым исходным кодом Shark.

**Python**

Популярность пайтона может быть объяснена как простым синтаксисом так и большим числом систем глубокого обучения и различных полезных для этого бибилиотек (TensorFlow, PyTorch и Keras.). Он подходит как для начинающих так и для разработки реальных проектов.

Он также мультипарадигмален.

**Java**

Java обычно используется корпорациями и сторонними разработчиками настольных приложений, которые работают над корпоративным машинным обучением. Что касается приложений машинного обучения в промышленности, обычно Java чаще, чем Python, используется в сетевой безопасности, в том числе для предотвращения кибератак и обнаружения мошенничества.

Среди библиотек машинного обучения для Java:

* Deeplearning4j, распределенная библиотека с открытым исходным кодом для глубокого обучения, написанная для Java и Scala.
* MALLET  (MAchine Learning for LanguagE Toolkit), набор инструментов для машинного обучения по текстам, который включает поддержку обработки естественного языка, моделирования тем, классификации документов и кластеризации.
* Weka, набор алгоритмов машинного обучения для обработки интеллектуального анализа данных.

**Основные Фреймворки на данный момент:**

## TensorFlow

TensorFlow — это комплексная платформа для машинного обучения с открытым исходным кодом. Она была разработана командой Google Brain как продолжение закрытой системы машинного обучения DistBelief, однако в ноябре 2015 года компания передумала и открыла фреймворк для свободного доступа.

в первую очередь Python, а также Javascript, C++ и Java. R, C#, Haskell, Go и Swift.

Платформа предоставляет интуитивно понятные высокоуровневые API-интерфейсы, например Keras, с быстрым выполнением, что обеспечивает немедленную итерацию модели и простую отладку. За счёт мультиплатформенности решение позволяет обучать и разворачивать модели в облаке и локально, независимо от используемого пользователем языка.

**Плюсы:**

* Отличный фреймворк для создания нейронных сетей, которые будут работать в продакшене.
* Берёт на себя оптимизацию ресурсов для вычислений.
* Огромное комьюнити.
* За счёт популярности выше вероятность, что проблему, подобную вашей, уже решили.

**Минусы:**

* Сложен в использовании и освоении.
* Недружелюбный.
* Необходимо постоянно контролировать используемую видеопамять.
* Имеет свои стандарты.
* Плохая документация.
* У вас всегда есть пять способов решить задачу, но три из них deprecated, один не работает, а тот, который работает, — не задокументирован.

## Keras

Keras — открытая среда глубокого обучения, написанная на Python. Она была разработана инженером из Google Франсуа Шолле и представлена в марте 2015 года.

Фреймворк нацелен на оперативную работу с нейросетями и является компактным, модульным и расширяемым. Подходит для небольших проектов, так как создать что-то масштабное на нём сложно и он явно будет проигрывать в производительности нейросетей тому же TensorFlow.

Keras работает поверх TensorFlow, CNTK и Theano и предоставляет интуитивно понятный API, который, по мнению наших инженеров, пока что является лучшим в своём роде.

**Плюсы:**

* Удобен в использовании.
* Лёгок в освоении.
* Быстроразвивающийся фреймворк.
* Хорошая документация.
* Встроен в TF.

**Минусы:**

* Не подходит для больших проектов.