

1. Введение

Данный текст написан по [материалам семинара Нейросети 2.0](#), так как, из-за ограниченности времени и формата семинара, а также сложности предмета, невозможно донести всю полноту своей позиции. Под нейросетями в данной материале подразумеваются современные математические модели с глубоким обучением.

В ходе выступления на семинаре я постоянно называл нейросети в их текущей ипостаси калькулятором, хранилищем и даже гадостью – это была жесткая реакция, как противовес тому, что некоторые участники семинара постоянно пытаются одушевить нейросети и приписать ИТ системам, построенным на базе нейросетей некую субъектность - “ведем диалог с нейросетью”, “нейросеть подошла к решению задачи творчески” и т.д.

Современные решения на базе нейросети в своей основе являются **огромными статическими наборами параметров**, которые формируются и изменяются только на этапе обучения нейросети, а **на этапе использования остаются неизменными**, а значит, запрос непосредственно к нейросети с некоторыми заданными параметрами всегда будет выдавать один и тот же результат, вне зависимости от истории общения с ней (от контекста). Например при общении с чат-ботом, построенным на базе NLP (natural language processing, обработка естественного языка), вся вариабельность диалога и учет контекста общения (запоминания фактов и переменных диалога) выполняется не нейросетью, а машиной состояний, которая интегрирована с ней или же парой слоев дополнительной нейросети построенной поверх мощной векторной модели. Но эти пару слоев не запоминают факты и переменные, а лишь **обеспечивают динамическое улучшение распознавания предложений**.

И это нормальный подход для бизнес задач, так как транзакции в ходе любой деятельности всегда запоминаются в базах данных, поверх которых построена машина состояний организации, а значит любые нейросети будут встраиваться бизнесом в такую машину состояний, в которой и хранится текущий контекст.

С другой стороны, задача рисования картинки по текстовому описанию, которую в качестве примера привел Максим, это задача другого класса, так сказать, задача на креативность и творчество, поэтому я решил посмотреть как устроена архитектура такой системы и какие нейросети там используются. **Генеративные модели — это сети без учителя (нет необходимости вручную готовить размеченные наборы данных), для обучения используется вторая (как правило сверточная) сеть-дискриминатор. Две нейросети соревнуются, одна пытается сформировать изображение максимально близкое к изображениям в заданном обучающем датасете, а вторая пытается определить является ли входное изображение “истинным” или сгенерированным. Начальный сигнал генерации задается рандомным образцом пикселей.**

Речь идет о мультимодальной генеративной нейросети, которая создает рисунок по текстовому описанию.

Источник: [Мультимодальные нейронные сети, как искусство](#)

Источник: [Как нейронные сети рисуют картины](#)

Казалось бы вот она встреча двух нейросетей, о которой столько говорилось на семинаре! Но ведь речь шла о совсем другой архитектуре. Лучше всех мысль об ансамбле нейросетей выразил Андрей Шел. - цитирую: **“Нужно иметь ввиду взаимодействия разных, понятие другого здесь очень важно, разных нейросетей, причём не на уровне только внешних интерфейсов, но и на уровне их внутреннего устройства, потому что некоторые узлы, некоторое непустое множество узлов, оно присутствует и в одной и в другой нейросети, ну и в 3-й, 4-й и так далее. “**

Однако, организация взаимодействия разных видов нейросетей потребует другие алгоритмы, другую методологию, другие языки программирования, другие библиотеки. В разговоре с опытным разработчиком в этой сфере (который десятилетия в теме), я спросил “Почему используете Python (этот язык плохо приспособлен для организации параллельных вычислений и асинхронного программирования) для проектов в области AGI (Общий Искусственный Интеллект) ?”. Ответ был: “Что-нибудь сначала создадим, а потом будем разбираться”. Для справки - все основные разработки в области AI и ML ведутся в Python.

Глава 3, возможно, будет сложной для неспециалистов, я постарался максимально просто изложить состояние дел в современной ИТ архитектуре.

С точки зрения ИТ, рекурсия и фрактал — это понятия относящиеся к архитектуре фон Неймана, когда имеется программный автомат, который шаг за шагом последовательно запускает функции и процедуры.

Но современная ИТ архитектура - это асинхронная парадигма ([Asynchronous Paradigm](#)), асинхронная логика и функция обратного вызова ([Callback](#)), ниже, в разделе 3, будет показан генезис этой архитектуры.

В этой архитектуре то, что верхнеуровнево воспринимается как рекурсия, на самом деле, на уровне программного кода является наиболее часто исполняемые замкнутые последовательности асинхронных функций, связанные через функции обратного вызова. Со стороны наблюдателя может показаться, что это рекурсия, но это не так. С фрактальностью ситуация сложнее, об этом позже.

Версия #4

GRN создал 18 July 2022 11:25:45

GRN обновил 6 August 2022 10:34:23