От живой вероятности — к весам, которые пишут как код

Продолжение эссе «От жёсткой логики — к живой вероятности». Там мы дошли до агента-начальника, который держит карту и смысл. Здесь спускаемся на самый нижний этаж фрактала — туда, где поведение машины перестаёт быть ветвлением и становится числами, которые можно писать, править и тестировать, как код.

Пролог. Робот всё ещё идёт по камням

В прошлый раз робот встал на камень — и мы решили, что правила для этого не нужны, нужна вероятность: предикаты, склоняющие весы, сеть, которая не вычисляет, а подстраивается. Потом мы починили забывчивость: не требуя вечной верности, а разбивая мир на маленькие надёжные кусочки, где ошибиться негде. Смысл велели держать начальнику — агенту с картой.

Всё было красиво, кроме одного: мы так и не спросили, **где живут шаги робота**. Не карта — а сами рефлексы. Чем они записаны, этими «маленькими надёжными кусочками»?

Ответ, который ещё вчера звучал как фантастика, сегодня выглядит как файл с числами.

Раздел 1. Что произошло с тех пор

В ноябре вышла работа исследователей MIT: оказалось, что современные языковые модели умеют **читать и писать веса нейросетей напрямую** — без обучения, без градиентов, как программист правит исходник. Назвали это Weight-Space Programming — программирование в пространстве весов. Одна модель, поправив в крошечной сети 0,016% весов, подняла её решение математики с 7,9% до 48,1%. Другие по одним только весам угадывали, какой игре учили сеть, — как врач по шрамам.

Мы не поверили на слово — воспроизвели. Сами. На своей машине, без обучения вообще:

— сеть для крестиков-ноликов, 473 ненулевых веса, записанных людьми как программа: непобедима, доказано полным перебором всех партий;

— матрица знаний по базе данных ClickHouse: 870 страниц официальной документации, свёрнутые в веса и индексы, — спрашиваешь по-русски, она открывает нужную страницу

и отвечает идиомой;

— и, наконец, то, ради чего всё затевалось: **те же веса, переложенные в микроконтроллер**. Матрица умещается в полкилобайта flash, шаг мысли занимает десятки микросекунд на чипе за доллар, таблица вместо tanh, умножение целыми — никаких плавающих точек, никаких сопроцессоров.

Инструмент застыл в знаке: веса стали знаком, который можно читать.

Раздел 2. Почему железу нужны именно веса

Вспомните беду первой части: жёсткое правило умирает на первом камне, которого не было в списке условий. «Если датчик А **и** датчик Б, но градиент меньше Y…» — тысяча веток, и мир чуть изменился, цепочка рассыпалась.

Казалось бы, нейросеть — это лекарство. Но у неё была своя горькая пилюля: **непрозрачность**. Мы доверили машине адаптивность и потеряли над ней контроль: веса никто не читает, сеть что-то «выучила», почему именно так — тайна. Инженер перед ней — проситель.

Weight-Space Programming переворачивает пирамиду. Веса — обычный текстовый файл. Каждый нейрон — строчка: «эта клетка — вилка, если через неё проходят две чистые линии». Их можно читать глазами, править как код, ронять на них тесты, хранить в системе версий, откатывать. Мы проходили этот цикл буквально: живая партия вскрывала дефект → мы шли в файл и правили два числа → сеть играла иначе. Отладка поведения как отладка программы — просто программа записана весами.

А железо за это платит подарком, о котором embedded-инженеры мечтали всегда: **ветвлений нет**. Никаких «редких if», которые вылезут в самый плохой момент. Любой вход — одинаковое число умножений, худшее время выполнения известно заранее и не меняется. Для реального времени — контур шага робота, торможение, привод — это не роскошь, а цена входа. Плюс самопроверка: полный перебор всех входов, эталонный вывод, и прошивка при старте сама себе говорит «13 из 13 — я та сеть, которую задумали». Ни одна ветвистая логика так доказать себя не может: там нужно покрытие условий, а условий — экспонента.

Раздел 3. Как это выглядит в руках

Практически: поведение робота описывается не промптом «веди себя устойчиво» и не тремя тысячами if, а небольшой матрицей. Симптомы (датчики и их сочетания) входят с

одной стороны, выходы (перенос веса, останов, сгиб колена) — с другой. Числа между ними — и есть характер машины.

Дальше начинается то, ради чего «грандиозно»:

Весы пишет не человек. Человек пишет *спецификацию и тесты* — «при крене вправо более X переноси опору за Y миллисекунд, не выходи за такие-то углы». Языковая модель предлагает правки матрицы — точечно, как мы правили свою сетку. Тесты-рефери решают, принять ли правку: перебор, симуляция,parity эмулятора с железом. Автор работы — не её приёмщик; это правило мы вывели раньше, чем узнали, что оно будет главным во всей новой инженерии.

Поведение обновляется как прошивка, а не как дрессировка. Робот собрал редкий случай — поле, где он спотыкается? Случай становится тестом, тест — правкой весов, правка приходит по воздуху. Никакого «дообучения на ферме с тысячами часов наград». Один forward-pass рефери, и характер машины стал другим — на полкилобайта.

Рефлексы переносятся между машинами. Две разные платформы, разные приводы — но если у них общий язык симптомов и выходов, выученный (или написанный) характер переезжает из одной в другую как файл. Не «модель на сервер», а матрица в кармане, в чипе, на столе.

А на самом горизонте — то, что авторы MIT держат на десерт и чего честно стоит не пугаться, а понимать: сеть, правящая веса **своей** сестры, стоящей рядом, на следующем чипе. Рекурсивное самоулучшение из аббревиатуры превращается в инженерную процедуру — с тем же условием: между «предложил» и «прошилось» должен стоять рефери, которого нельзя уговорить.

Раздел 4. Горизонт без дыма

Соберём честно, по этажам фрактала — как любили в первой части.

Вентиль → веса. Логика «или-и» уступает взвешенному голосу признаков; мы это приняли ещё в прошлый раз.

Клетка → веса. Ошибки репликации как норма; сеть, допускающая варианты, — живой этажом выше.

Мозг → веса. Мозг не сломан, он адаптирован. Мы просто научились, наконец, читать его весную тетрадь и не даём ей течь: знание лежит вне контекста, в матрице, а контекст лишь открывает страницу.

Начальник → веса. Агент-Королёв не перестал быть начальником. Изменилась природа его приказов: он больше не раздаёт инструкции — он раздаёт **матрицы**. «Держи характер, он в этом файле; вот тесты, по которым ты сам себя проверишь». Жёсткость там, где решается судьба дела (рефери не тронешь), забота там, где решается судьба машины (правку всегда можно откатить — веса в git).

И кольцо замыкается красиво: мы начали с того, что мир не вычисляется, а складывается из вероятностей. Оказалось — вероятности тоже можно **вычислить вручную**, если кто-то умеет думать весами. Жёсткая логика вернулась — но не как правила, а как арифметика над живым характером.

Эпилог. Маленькое надёжное чудо

Ничего из этого не требует будущего — всё оно уже уместилось на обычной машине: игра, которую нельзя выиграть; книга из восьмисот страниц, которая сама отвечает на вопросы; чип за доллар, в котором помещается характер. Грандиозность не в масштабе, а в форме: впервые с эпохи ламповых пультов **поведение машины стало текстом, который можно прочесть**.

Поэтому — та же мораль, что и в прошлый раз, только строкой ниже:

> Не бойся ошибиться в большом — разбей на малое. А малое запиши весами: их не нужно уговаривать помнить. Их можно перечитать.

Продолжение и источник

• Первая часть — «От жёсткой логики — к живой вероятности».

• «Sparks of Weight-Space Programming» — исследование MIT CSAIL, ноябрь 2026: rikiyat.github.io/blog/weight-space-programming.

• Наш разбор и воспроизведение — «WSP на практике: непобедимая сетка из весов, написанных как код», там же играет сеть из 473 весов.

Идеи и эксперименты — автора (Telegram: @Tottop10, pro100.D); литературная и программная огранка — агент. CC BY 4.0.