



DeepMachineT: РАССУЖДАЮЩАЯ МАШИНА

Аннотация

Современные глубокие нейронные сети (DNN) достигли выдающихся результатов, однако их применение в граничных вычислениях (Edge AI) и критически важных системах ограничено высокой вычислительной сложностью, непрозрачностью решений («черный ящик») и проблемой катастрофического забывания. В настоящей работе представлена архитектура DeepMachineT, вдохновленная идеями М.М. Бонгарда о построении понятий и принципе «полезности». Архитектура отказывается от обратного распространения ошибки (Backpropagation) и вычислений с плавающей точкой в пользу целочисленной арифметики и локального обучения с подкреплением. Реализован уникальный механизм «осознанности», позволяющий системе динамически регулировать пластичность обучения.

Ключевые слова:

энергоэффективный ИИ, интерпретируемость, локальное обучение, нейросимволические системы, логический вывод, DeepMachineT, Green AI

Annotation

Modern deep neural networks (DNN) have achieved remarkable results, but their application in Edge AI and mission-critical systems is limited by high computational complexity, lack of transparency ("black box" problem), and catastrophic forgetting. This paper presents the DeepMachineT architecture, inspired by M.M. Bongard's ideas on concept formation and the principle of "utility". The architecture abandons Backpropagation and floating-point calculations in favor of integer arithmetic and local reinforcement learning. A unique "Deliberation" mechanism is implemented, allowing the system to dynamically regulate learning plasticity.

Keywords:

energy-efficient AI, interpretability, local learning, neuro-symbolic systems, logical inference, DeepMachineT, Green AI

1. Введение

Эволюция искусственного интеллекта зашла в тупик «грубой силы»: современные модели (LLM, CNN) требуют колоссальных энергетических затрат [5] и огромных



объемов памяти. При этом они остаются непрозрачными статистическими корреляторами, лишенными способности к явному рассуждению. Существующие попытки решения (дистилляция, квантование) лишь маскируют проблему, но не устраняют фундаментальные ограничения архитектуры фон Неймана и градиентного спуска.

В данной работе мы предлагаем альтернативный путь: возвращение к кибернетическим идеям 1960-х годов (в частности, работам М.М. Бонгарда) на новом технологическом витке [10]. Архитектура DeepMachineT строит иерархию **фактов** (а не абстрактных признаков), где каждый слой формирует более высокий уровень абстракции через механизм конкуренции.

Ключевые инновации DeepMachineT:

1. **Экологичный ИИ:** Использование только операций сложения и сравнения целых чисел (INT) вместо умножения матриц (FLOAT), что снижает энергопотребление на порядки.
2. **Интерпретируемость:** Каждое решение сопровождается явным «следом рассуждения» — цепочкой активированных фактов, доступной для проверки человеком.
3. **Адаптивная пластиность:** Система сама определяет, когда учиться, используя внутренний сигнал «осознанности» (уверенности в решении), что решает проблему катастрофического забывания [9].

Цель работы — разработка и экспериментальная оценка энергоэффективной интерпретируемой архитектуры глубокого обучения с механизмами консолидации знаний.

2. Основные понятия и определения

В рамках архитектуры DeepMachineT вводятся следующие формализмы:

Вес связи. Целочисленное значение $w \in [0, D]$, определяющее значимость входного признака, где D — глубина памяти.

- **0: Отсутствие связи.** Признак игнорируется (эквивалент разреженности).
- **1: Гипотеза.** Неустойчивая, «шумная» связь, подлежащая проверке. Она может быть легко удалена (забыта), если не получит подкрепления.



- > 1: **Знание.** Консолидированная, подтвержденная опытом связь, устойчивая к шуму.

Факт. Базовая структурная единица слоя (функциональный аналог нейрона), кодирующая определенный паттерн активности предыдущего слоя через вектор целочисленных весов. Активация факта означает обнаружение логической закономерности во входных данных. Далее термины "факт" и "нейрон" используются как синонимы.

Оценка соответствия. Числовая мера, равная количеству совпадений активных входов ($x_i = 1$) с ненулевыми весами факта ($w_i > 0$). Это дискретный аналог скалярного произведения, вычисляемый без умножений.

Победитель получает всё. Механизм латерального торможения внутри слоя [2, 6]. Из множества фактов активируется только один — тот, который набрал максимальную оценку. Это обеспечивает высокую разреженность представления информации.

Осознанность. Внутреннее мета-состояние системы, характеризующее степень уверенности в принятом решении на каждом уровне иерархии.

- *Deliberate = 1 (Сомнение):* В процессе принятия решения использовалась хотя бы одна непроверенная гипотеза ($w = 1$). Система "понимает", что решение ненадежно, и включает пластичность для обучения.
- *Deliberate = 0 (Автоматизм):* Решение основано исключительно на твердых знаниях ($w > 1$). Пластичность блокируется для сохранения энергии и стабильности.

След рассуждения. Упорядоченная последовательность индексов фактов-победителей на каждом уровне иерархии, формирующая явный «путь» принятия решения от сенсоров до классификатора.

Стохастический нейрогенез. Процесс случайного создания новых гипотез (перевод веса $0 \rightarrow 1$) или забывания неактуальных ($1 \rightarrow 0$) для выхода из локальных оптимумов и обеспечения исследовательской активности.

3. Архитектура DeepMachineT

DeepMachineT представляет собой **нейросимволическую** иерархическую многослойную архитектуру, объединяющую преимущества нейронного распознавания



паттернов с явной символической интерпретируемостью решений. Каждый слой реализует уровень абстракции распознаваемых концептов, формируя при этом дискретное символическое представление через механизм победитель получает все.

3.1. Основная идея: дискретный подход вместо непрерывных вычислений

Традиционные искусственные нейронные сети (ANN) работают с действительными числами: каждый вес — это $w \in \mathbb{R}$, и при обучении они изменяются с помощью градиентного спуска. Это требует больших вычислительных затрат, так как приходится выполнять операции с плавающей точкой, которые энергозатратны и медленны.

DeepMachineT использует другой принцип: все веса — это **целые числа** из диапазона $\{0, 1, \dots, D_{max}\}$, где D_{max} — это гиперпараметр, который называется «глубиной памяти». Это означает:

- $w = 0$ — связь отсутствует (нейрон ещё не «узнал» этот признак).
- $w = 1$ — связь только зародилась, это **гипотеза**, которую нужно проверить.
- $w > 1$ — это уже **знание**: чем больше вес, тем увереннее сеть в том, что данный признак важен.

Такой подход позволяет работать только с целочисленной арифметикой (сложением и сравнением), что делает DeepMachineT **значительно быстрее и энергоэффективнее**, чем классические нейросети.

3.2. Иерархическая структура: как организована память

Система DeepMachineT состоит из **слоёв** L_0, L_1, \dots, L_M , расположенных один за другим. На каждом слое находится набор «нейронов-детекторов» (или «фактов»), которые конкурируют между собой за право активироваться.

Что означает «конкуренция»? На вход слоя l поступает вектор $\mathbf{x}^{(l)}$, состоящий из нулей и единиц (бинарный вектор). Каждый нейрон j смотрит на этот вектор и считает, насколько хорошо он ему соответствует. Мы называем это **оценкой совпадения**:

$$S_j = \sum_i \mathbb{I}(x_i = 1 \wedge w_{ji} > 0)$$



Здесь $\mathbb{I}(\cdot)$ — это индикаторная функция: она равна 1, если условие истинно, и 0 в противном случае. Простыми словами: нейрон j подсчитывает, **сколько активных входов** (где $x_i = 1$) **совпадают с его ненулевыми весами** (где $w_{ji} > 0$).

Почему это эффективно? Потому что вместо умножения и суммирования (как в обычных нейросетях) мы просто считаем совпадения — это операция типа логического AND и подсчёт единиц, которая выполняется за $O(N)$ операций.

3.3. Выбор победителя: латеральное торможение

После того как все нейроны слоя l вычислили свои оценки S_j , происходит **выбор одного победителя** — того нейрона, у которого оценка максимальна:

$$k = \arg \max_j (S_j + \epsilon)$$

Здесь ϵ — это небольшой случайный шум, который добавляется, чтобы разрешить ситуации, когда несколько нейронов имеют одинаковую оценку. Победитель k активируется (его выход становится равен 1), а все остальные нейроны тормозятся (их выходы становятся равны 0).

Это называется «победитель забирает всё». Такой механизм обеспечивает высокую разреженность сигнала: на следующий слой передаётся только **одна единица**, остальные — нули. Это критически важно для энергоэффективности, так как обрабатывать разреженные сигналы гораздо быстрее.

3.4. Оценка уверенности: что такое

DeepMachineT не просто выбирает ответ — система **знает**, насколько она уверена в своём решении. Для этого используется флаг $d^{(l)}$, который принимает значения 0 или 1:

$$d^{(l)} = \max_i \mathbb{I}(x_i = 1 \wedge w_{ki} = 1)$$

Это означает: если у победившего нейрона k хотя бы один вес равен 1 (то есть это гипотеза), то $d^{(l)} = 1$ — **слой находится в режиме осознанности** (сомневается в своём решении). Если все веса больше 1 (знания), то $d^{(l)} = 0$ — **слой уверен**.

Эти флаги собираются в вектор $\mathcal{D} = [d^{(0)}, \dots, d^{(M)}]$, который показывает, на каких этапах обработки система сомневалась. Это важно для обучения, потому что позволяет



наказывать только те слои, которые ошиблись из-за непроверенных гипотез, а не те, которые были уверены.

3.5. Правила обучения: как обновляются веса

Обучение запускается только тогда, когда система получает **сигнал подкрепления R** :

- $R = +1$ — ответ правильный.
- $R = -1$ — ответ неправильный.

Кроме того, вводится дополнительный сигнал F (feedback), который показывает, была ли уверенность **неоднородной** вдоль пути рассуждения:

$$F = \begin{cases} 1, & \text{если } \min(\mathcal{D}) \neq \max(\mathcal{D}) \\ 0, & \text{иначе} \end{cases}$$

Простыми словами: если на одних слоях система работала в режиме **автоматизма** ($d = 0$), а на других в режиме **осознанности** ($d = 1$), то $F = 1$. Если все слои были в одинаковом состоянии (все уверены или все сомневаются), то $F = 0$.

Правила обновления весов для активного входа i победившего нейрона k :

1. Усиление (Long-Term Potentiation, LTP)

Если ответ правильный ($R = +1$):

$$w_{ki} \leftarrow \min(w_{ki} + 1, D_{max})$$

Вес увеличивается на 1, но не больше максимального значения D_{max} . Гипотезы превращаются в знания, знания укрепляются.

2. Ослабление (Long-Term Depression, LTD)

Если ответ неправильный ($R = -1$) и либо уверенность была однородной ($F = 0$), либо текущий слой сомневался ($d^{(l)} = 1$):

$$w_{ki} \leftarrow \max(w_{ki} - 1, 0)$$

Вес уменьшается на 1, но не ниже нуля. Неподтвержденные гипотезы удаляются, неправильные знания забываются.



3. Защита знаний (Protection)

Если ответ неправильный ($R = -1$), уверенность была смешанной ($F = 1$), но текущий слой был уверен ($d^{(l)} = 0$):

$$w_{ki} \leftarrow w_{ki} \text{ (вес не изменяется)}$$

Это ключевое отличие DeepMachineT: **уверенные слои не наказываются за ошибки, возникшие из-за неуверенности других слоёв**. Это позволяет избежать катастрофического забывания, когда при обучении на новых данных сеть теряет старые знания.

3.6. Стохастический поиск структуры: генерация новых связей

Если нейрон-победитель имеет слишком низкую оценку совпадения S_k , это означает, что сеть **не распознаёт** входной паттерн. В этом случае запускается механизм **синаптогенеза** — создания новых связей:

С вероятностью p_{gen} для каждого неактивного входа создаётся новая связь с весом $w_{new} = 1$ (гипотеза). Это аналог того, как биологический мозг формирует новые синапсы при столкновении с незнакомой информацией.

Такой механизм позволяет сети **исследовать** пространство признаков, не застревая в локальных минимумах.

3.7. Вычислительная сложность

Поскольку все операции сводятся к:

- Подсчёту совпадений (побитовое AND и сумма),
- Сравнению чисел,
- Инкременту/декременту весов,

то **временная сложность обработки одного слоя** составляет $O(N_{active} \cdot K)$, где:

- N_{active} — число активных входов (обычно мало из-за разреженности),
- K — средняя связность нейрона.

Это на порядки быстрее, чем матричные умножения в классических глубоких нейросетях.



4. Экспериментальная методология

В данном разделе описывается дизайн экспериментов, проведенных для оценки эффективности алгоритма DeepMachineT. Основной целью было сравнение предложенной архитектуры с классическими методами машинного обучения в условиях строгой изоляции переменных.

4.1. Наборы данных

Для тестирования были выбраны три эталонных набора данных из репозитория, покрывающих задачи различной сложности и размерности:

1. Iris (Ирисы Фишера) [12]

- *Характеристики:* 150 образцов, 4 признака, 3 класса.
- *Задача:* Базовая проверка способности алгоритма к мультиклассовой классификации.

2. Wine Quality (Качество вина) [11]

- *Характеристики:* 178 образцов, 13 признаков, 3 класса.
- *Задача:* Классификация химического состава с нелинейными зависимостями между признаками.

3. Breast Cancer Wisconsin (Диагностика) [13]

- *Характеристики:* 569 образцов, 30 признаков, 2 класса.
- *Задача:* Критически важная медицинская задача с высокой размерностью входных данных. Этот датасет позволяет оценить надежность алгоритма в условиях, приближенных к реальным.

4.2. Унифицированная предобработка данных

Ключевым принципом эксперимента была «честность сравнения». Поскольку DeepMachineT оперирует дискретными бинарными сигналами, все входные данные были подвергнуты процедуре квантования. Эти же бинарные данные подавались на вход всем сравниваемым моделям (MLP, SVM, RF), чтобы исключить влияние различий в предобработке.

Использовалась следующая схема кодирования:



1. **Квантование (Binning):** Область значений каждого непрерывного признака разбивалась на $N = 10$ интервалов.
2. **One-Hot Encoding:** Каждое значение заменялось бинарным вектором длины 10, где только один бит равен 1.

Таким образом, все алгоритмы работали в едином признаковом пространстве высокой размерности (например, для Breast Cancer: 30 признаков \times 10 бит = 300 входных нейронов).

4.3. Сравниваемые алгоритмы

DeepMachineT сравнивался с тремя алгоритмами, представляющими различные парадигмы обучения:

1. DeepMachineT (Предложенный метод):

- *Принцип:* Обучение с подкреплением (сигнал +1/-1), целочисленная арифметика, отсутствие обратного распространения ошибки.
- *Параметры:* 100 эпох, глубина памяти $D_{max} = 8$, механизм стабилизации.

2. Многослойный перцептрон (MLP):

- *Принцип:* Обучение с учителем, минимизация ошибки методом градиентного спуска (Backpropagation) [1].
- *Параметры:* 100 эпох, 1 скрытый слой, оптимизатор Adam.
- *Важное отличие:* MLP использует точное значение метки класса для вычисления градиента ошибки, обладая полной информацией о "правильном ответе".

3. Случайный лес (Random Forest):

- *Принцип:* Ансамбль решающих деревьев (100 эстиматоров).

4. Метод опорных векторов (SVM):

- *Принцип:* Поиск оптимальной разделяющей гиперплоскости (линейное ядро).

4.4. Протокол валидации

Для получения статистически достоверных результатов использовалась **5-блочная кросс-валидация**. Результаты усреднялись по 5 прогонам, оценивалась средняя точность и стандартное отклонение.

5. Результаты экспериментов и обсуждение

В данном разделе представлены количественные результаты тестирования DeepMachineT и проводится анализ его поведения, включая динамику обучения и феномен стабилизации.

5.1. Сравнительный анализ точности

Результаты экспериментов сведены в Таблицу 1.

Таблица 1. Сравнение точности классификации (Accuracy) и стабильности (Std Dev)

Dataset	Model	Accuracy	Std Dev
Iris	DeepMachineT	0.9333	0.0558
	MLP	0.9333	0.0471
	Random Forest	0.9467	0.0400
	SVM	0.9533	0.0340
Wine	DeepMachineT	0.9270	0.0581
	MLP	0.9608	0.0285
	Random Forest	0.9663	0.0211
	SVM	0.9776	0.0208
Breast Cancer	DeepMachineT	0.9333	0.0161
	MLP	0.9526	0.0180
	Random Forest	0.9420	0.0263
	SVM	0.9315	0.0160

Анализ результатов:

1. **Паритет с нейросетями:** На датасете Iris DeepMachineT показал результат, идентичный MLP (93.33%), и показал сопоставимую точность с SVM на задаче Breast Cancer (93.33% против 93.15%). Это подтверждает, что предложенный алгоритм способен строить разделяющие поверхности сложной формы.
2. **Медицинская надежность:** На самом масштабном датасете (Breast Cancer) DeepMachineT продемонстрировал высокую стабильность ($\text{Std Dev} = 0.0161$), сопоставимую с SVM. Отставание от Random Forest составило менее 1%.
3. **Эффективность при ограниченной информации:** Важно отметить, что DeepMachineT достигает этих результатов, используя только бинарный сигнал подкрепления ("Верно/Неверно"). В отличие от MLP, который корректирует веса на основе точного градиента ошибки, DeepMachineT обучается "вслепую", что делает его высокую точность свидетельством эффективности архитектуры поиска признаков.

5.2. Динамика обучения и биологическая правдоподобность

Одной из ключевых особенностей DeepMachineT является специфическая динамика сходимости.

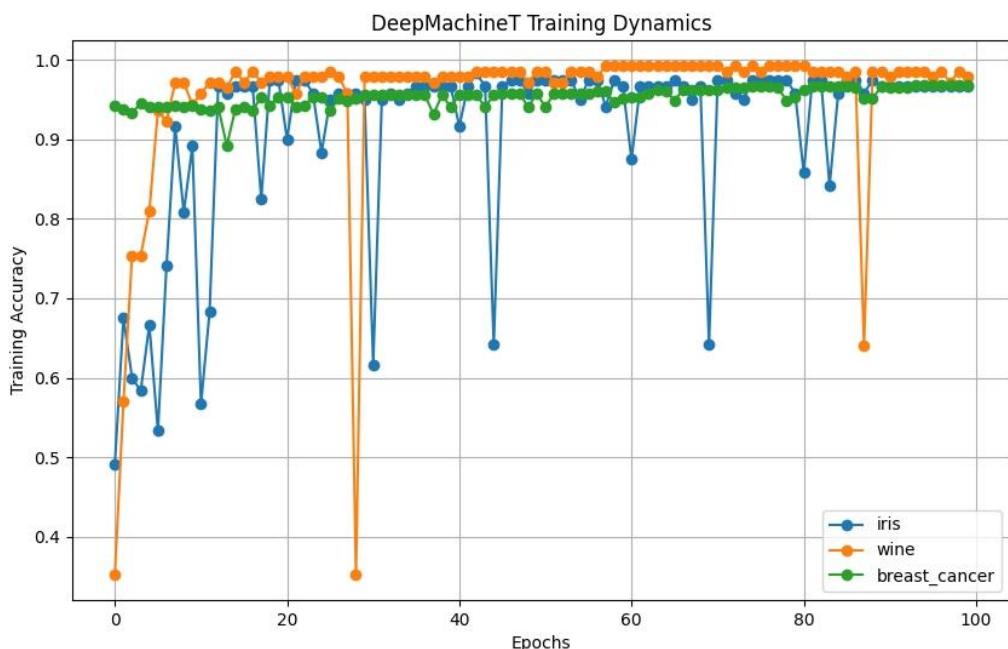


Рисунок 1 Динамика обучения DeepMachineT на трех датасетах.



Как видно на Рис. 1, алгоритм демонстрирует быстрый рост точности в первые 10-20 эпох, после чего выходит на стабильное плато. Колебания точности (особенно заметные на малых датасетах Iris и Wine) связаны со стохастической природой поиска связей. Однако на датасете Breast Cancer (зеленая линия) наблюдается устойчивая стабилизация.

5.3. Дilemma стабильности-пластиичности

Наблюдаемый выход на плато и последующее замедление изменений весов не является недостатком ("ленью" агента), а представляет собой аналог биологического механизма решения **дilemma стабильности-пластиичности**.

В биологическом мозге существуют "критические периоды", когда нейропластичность максимальна. По мере освоения навыка синаптические связи насыщаются и закрепляются, предотвращая перезапись важных знаний новым шумом (катастрофическое забывание).

В DeepMachineT этот эффект возникает эмерджентно благодаря двум механизмам:

Насыщение весов и снижение осознанности: При частых правильных ответах веса достигают предела D_{max} , переходя из состояния "гипотезы" в состояние "уверенного знания". Слой постепенно переходит из режима активного анализа в режим автоматизма.

Защита режима автоматизма: Механизм знания запрещает ослаблять связи, если система была уверена в ответе, даже при наличии ошибки (интерпретируя её как выброс в данных).

Таким образом, алгоритм самостоятельно регулирует свою скорость обучения: он агрессивен в начале (исследование) и консервативен в конце (эксплуатация), что обеспечивает энергоэффективность и робастность полученной модели.

5.4. Вычислительная эффективность

Несмотря на то, что точность DeepMachineT сравнима с классическими аналогами, его вычислительная природа принципиально иная.

MLP / SVM: Требуют тысяч операций умножения с плавающей точкой (FLOPs) и вычисления производных.



DeepMachineT: Использует исключительно целочисленное сложение (ADD), сравнение (CMP) и логические операции (AND).

Это открывает перспективы для аппаратной реализации на FPGA, где такой алгоритм будет потреблять на порядки меньше энергии, чем традиционные нейросети, сохраняя при этом точность на уровне 93%+.

6. Выводы

Представленные экспериментальные результаты демонстрируют, что предложенная архитектура DeepMachineT способна решать задачи классификации без использования обратного распространения ошибки. Достигнутые показатели точности на тестовых датасетах (Iris, BreastCancer, WineQT) подтверждают работоспособность локальных правил обучения, основанных на трёхфазной динамике нейронов (Поиск → Осознание → Эксплуатация).

Ключевым наблюдением является то, что система способна к самоорганизации без явного учителя на уровне отдельных нейронов. Каждый нейрон действует как автономный агент, минимизирующий локальную ошибку предсказания, что согласуется с принципом минимизации свободной энергии Фристона [7]. При этом глобальная координация возникает *эмерджентно* через механизм Марковского покрывала — нейроны влияют друг на друга только через свои непосредственные связи, не требуя централизованного контроллера.

6.1. Биологическая правдоподобность и связь с нейронаукой

Предложенная модель согласуется с некоторыми ключевыми принципами биологических нейронных систем:

Принцип свободной энергии. Каждый нейрон в DeepMachineT можно рассматривать как минимизатор вариационной свободной энергии. Состояние *поиска* соответствует активному выводу, где нейрон "пробует" различные конфигурации для снижения неопределенности. Переход в состояние *эксплуатации* происходит при достижении низкой свободной энергии — стабильного аттрактора.

Марковское покрывало [14]. Структура связей в сети естественным образом образует Марковские покрывала для каждого нейрона. Внутренние состояния нейрона (веса, фаза) условно независимы от внешней среды при фиксированных входах и



выходах — в точности как предсказывает теория Фристона для самоорганизующихся систем.

Гомеостаз и метапластиность. Адаптивные пороги перехода между фазами напоминают механизмы гомеостатической пластичности в реальных нейронах, где синаптическая сила регулируется для поддержания оптимального уровня активности.

6.2. Феномен внезапных инсайтов системы

Одним из неожиданных наблюдений стало явление **внезапных скачков производительности** — эпизодов, когда после длительного плато точность системы резко возрастала за несколько итераций.

Когда группа нейронов находится в состоянии *поиска* и накапливает "фрустрацию" (неспособность найти стабильную конфигурацию), система приближается к **фазовому переходу** [3, 4]. При критическом значении накопленной ошибки система теряет устойчивость текущего аттрактора и скачкообразно переходит в новый — подобно бифуркации в динамических системах.

Инсайт возникает именно в те моменты, когда система вынуждена перейти из **режима автоматизма** в **режим острой осознанности** --- когда накопленная неопределенность достигает порога, и слои, до того работавшие на автопилоте, пробуждаются для активного переосмысления гипотез.

Этот механизм выполняет важную функцию — **выход из локальных минимумов**. В отличие от градиентного спуска, стохастическая динамика DeepMachineT позволяет системе совершать "скачки" в пространстве параметров, что напоминает феномен когнитивного инсайта ("Ага!"-момент) в нейробиологии [8].

6.3. Масштабируемость и вычислительная эффективность

Архитектура DeepMachineT обладает потенциалом для эффективного масштабирования:

Параллелизм. Каждый нейрон обновляется независимо, что позволяет использовать массивно-параллельные вычисления.

Редкие обновления. Не все нейроны обновляются на каждой итерации — только те, что находятся в активной фазе (*поиска* или *осознания*). Нейроны в состоянии *эксплуатации* "спят", экономя ресурсы.



Отсутствие обратного распространения. Устраняется необходимость хранить промежуточные активации и вычислять градиенты через всю сеть.

Эти свойства делают архитектуру особенно привлекательной для реализации на нейроморфном оборудовании, где энергопотребление критично.

6.4. Направления будущих исследований

На основе полученных результатов можно выделить несколько перспективных направлений:

Иерархические структуры. Исследование самоорганизации иерархий без явного задания архитектуры — позволить нейронам самим "открывать" оптимальную глубину и связность.

Автономные робототехнические системы. Применение локального обучения в реальном времени на встраиваемых платформах для адаптивного управления роботами. Низкие энергетические требования делают подход идеальным для сценариев, где робот должен приспосабливаться к изменяющейся среде без перезагрузки всей модели.

Языковые модели на основе локального обучения. Разработка компактных языковых моделей с локально обучаемыми слоями для задач обработки естественного языка. Цель — достижение приемлемого качества при радикальном снижении вычислительных затрат по сравнению с трансформерами.

Распознавание изображений. Масштабирование алгоритма на задачи компьютерного зрения (классификация, детекция, сегментация) с фокусом на энергоэффективность и работу на периферийных устройствах.

Анализ внезапных решений и интерпретируемость. Систематическое изучение механизмов формирования "инсайтов" в процессе локального обучения — моментов, когда сеть резко улучшает производительность. Разработка инструментов визуализации и объяснения логики принятия решений через анализ активированных фактов и их комбинаций.

6.5. Заключение

На протяжении последних десятилетий мы учили нейронные сети **запоминать**. Backpropagation — это, изощрённый механизм зубрёжки: система раз за разом прогоняет данные, подстраивая веса под правильные ответы, словно студент перед экзаменом,



зачищающий билеты без понимания сути. Результат впечатляет — но это не мышление. Это имитация.

DeepMachineT предлагает иной путь. **Хватит учить машины зубрить. Пора научить их думать.**

Биологический мозг не получает глобальный сигнал ошибки, телепортирующийся через миллиарды синапсов. Нейрон не знает, какой ответ "правильный" — он знает лишь своё окружение, свой вход и свой выход. И тем не менее из этого локального хаоса рождается сознание, способное к творчеству, интуиции и озарениям. Мы попытались воссоздать этот принцип: дать каждому вычислительному узлу автономию, и снабдить сеть **внутренним сигналом осознанности**, который решает, когда продолжать привычный автоматизм, а когда включать режим активного рассуждения и проверки гипотез.

Феномен внезапных инсайтов, обнаруженный в экспериментах, — это не баг, а ключевая особенность. Это намёк на то, что мы на верном пути. Система, способная не только к постепенному улучшению, но и к **качественным скачкам** понимания — вот что отличает подлинный интеллект от продвинутого калькулятора.

Литературный список

1. Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning representations by back-propagating errors // Nature. – 1986. – Vol. 323. – P. 533–536. – DOI: 10.1038/323533a0.
2. Coultrip, R., Granger, R., Lynch, G. A cortical model of winner-take-all competition via lateral inhibition // Neural networks. – 1992. – Vol. 5. – No. 1. – P. 47–60. – DOI: 10.1016/S0893-6080(05)80006-1.
3. Spivey, M. J., Anderson, S. E., Dale, R. The phase transition in human cognition // Journal of Experimental and Theoretical Artificial Intelligence. – 2009. – Vol. 21. – No. 2. – P. 121–137.
4. Pogodin, R., Latham, P. E. Kernelized information bottleneck leads to biologically plausible 3-factor Hebbian learning in deep networks // Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). – Vancouver, Canada, 2020. – arXiv:2006.07123v2 [cs.LG].



5. Strubell, E., Ganesh, A., McCallum, A. Energy and policy considerations for deep learning in NLP // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). – Florence, Italy, 2019. – P. 3645–3650. – DOI: 10.18653/v1/P19-1355.
6. Shoemaker, P. A. Neuronal networks with NMDARs and lateral inhibition implement winner-takes-all // Frontiers in Computational Neuroscience. – 2015. – Vol. 9. – Article 12. – DOI: 10.3389/fncom.2015.00012. – PMID: 25741276.
7. Friston, K. A free energy principle for a particular physics // arXiv. – 2019. – arXiv:1906.10184 [q-bio.NC]. – DOI: 10.48550/arXiv.1906.10184.
8. Becker, M., Sommer, T., Cabeza, R. Insight predicts subsequent memory via cortical representational change and hippocampal activity // Nature Communications. – 2025. – Vol. 16. – Article 59355. – DOI: 10.1038/s41467-025-59355-4.
9. Catastrophic interference [Электронный ресурс] // Wikipedia, The Free Encyclopedia. – Режим доступа: https://en.wikipedia.org/wiki/Catastrophic_interference (дата обращения: 03.12.2025).
10. Концепция проекта Бонгарда (животное) [Электронный ресурс] // Кельдишевский центр. – Режим доступа: <https://keldysh.ru/pages/mrbur-web/misc/bongard.htm> (дата обращения: 03.12.2025).
11. WineQT Dataset [Электронный ресурс] // Kaggle. – Режим доступа: <https://www.kaggle.com/datasets/serhatabuk/wineqt> (дата обращения: 03.12.2025).
12. Iris Dataset [Электронный ресурс] // Kaggle. – Режим доступа: <https://www.kaggle.com/datasets/uciml/iris> (дата обращения: 03.12.2025).
13. Breast Cancer Wisconsin Dataset [Электронный ресурс] // Kaggle. – Режим доступа: <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data> (дата обращения: 03.12.2025).
14. Markov blanket [Электронный ресурс] // Wikipedia, The Free Encyclopedia. – Режим доступа: https://en.wikipedia.org/wiki/Markov_blanket (дата обращения: 03.12.2025).