

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 15

Анализ свойств и характеристик прототипов IoT-устройств с искусственным интеллектом на базе нейрокомпьютеров. Оценка ценности внедрения и защита интеллектуальной собственности

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент · Кафедра «Искусственный интеллект и big data»
«Технологии в интеллектуальных системах управления в IoT» · 4 курс, бакалавриат

1 Цель лекции

Сформировать системное представление о методах исследования, верификации и сравнительной оценки интеллектуальных IoT-прототипов, обосновании их технологической и экономической ценности и выборе правовых механизмов охраны результатов интеллектуальной деятельности.

2 План лекции

1. Понятие интеллектуального IoT-прототипа и этапы жизненного цикла
2. Архитектура нейрокомпьютерной системы управления и требования к прототипу
3. Технические характеристики и измеримые показатели эффективности
4. Верификация, валидация и организация экспериментального исследования
5. Надёжность, безопасность, киберзащита и эксплуатационные ограничения
6. Технико-экономическая оценка и готовность к внедрению
7. Интеллектуальная собственность: объекты охраны, патенты и лицензии
8. Пример комплексной оценки IoT-системы с моделью YOLO

1 Интеллектуальный IoT-прототип как объект исследования

Интеллектуальный IoT-прототип — опытная реализация технической системы, которая измеряет параметры среды, передаёт данные по сети, анализирует их с использованием алгоритмов искусственного интеллекта и формирует информационные или управляющие воздействия. В отличие от демонстрационного макета исследовательский прототип должен иметь формализованные требования, повторяемую методику испытаний и измеримые критерии успешности.

Типовой жизненный цикл включает постановку задачи, разработку требований, выбор аппаратной платформы, проектирование сенсорного тракта, сбор и разметку данных, обучение модели, развёртывание на периферийном вычислителе, интеграцию с исполнительными механизмами, испытания и документирование результатов. Переход между этапами должен сопровождаться контрольными артефактами: спецификацией, схемой, кодом, протоколом измерений и отчётом.

Термины «нейрокомпьютер», «нейронный мини-ПК» и «нейронный микроконтроллер» в учебной практике могут обозначать разные устройства. В технической документации их необходимо различать: одноплатный компьютер (например, Raspberry Pi), микроконтроллер с поддержкой TinyML, специализированный NPU/TPU-ускоритель и модуль GPU для Edge AI имеют различные ограничения по памяти, питанию и программной среде.

2 Архитектура интеллектуальной IoT-системы

Функциональная цепочка: объект → датчики/камера → предварительная обработка → модель ИИ → логика принятия решений → исполнительное устройство → объект. Параллельно реализуется телеметрический контур: периферийное устройство → локальная сеть/шлюз → MQTT или HTTPS → сервер мониторинга/архив. При обрыве сети локальные функции безопасности должны сохраняться.

В составе аппаратной платформы выделяют: вычислительный модуль, камеру или датчики, интерфейсы GPIO/I²C/SPI/UART, источник питания, сетевой модуль и средства аппаратной защиты. Программная архитектура включает драйверы, сбор данных, инференс, планировщик задач, журналирование, конфигурацию и обновление ПО.

Для систем с физическим управлением необходимо отделять рекомендательное решение нейросети от исполнительного контура. Нейросеть может обнаруживать дефект, но команда остановки конвейера должна проходить через проверяемую логику блокировок, которая учитывает ложные срабатывания и требования безопасности.

2 Архитектура интеллектуальной IoT-системы (продолжение)

2.1. Формальная модель обработки

Пусть x_k — измеренный вектор параметров, $z_k = f\theta(x_k)$ — выход модели ИИ, $u_k = \pi(z_k, s_k)$ — управляющее воздействие с учётом состояния s_k. Тогда объект можно описать дискретной моделью: $s_{k+1} = F(s_k, u_k, w_k)$, где w_k — внешние возмущения. Качество всей системы зависит не только от точности $f\theta$, но и от устойчивости F, задержки обработки и правил формирования u_k.

3 Технические показатели и критерии сравнения

Оценка должна проводиться при одинаковых входных данных, разрешении кадров, версии модели, условиях освещения, температуре и продолжительности испытания. Номинальные показатели TOPS не равны фактической производительности приложения: существенны совместимость операторов, загрузка памяти, передача данных и постобработка.

3 Технические показатели и критерии сравнения (продолжение)

Группа	Показатель	Метод определения
Вычисления	Latency, мс	Время от поступления кадра до готового решения
Производительность	FPS, кадр/с	Число обработанных кадров / время
Качество ИИ	Precision, Recall, F1, mAP	На независимом размеченном наборе
Энергетика	P, Вт; E, Дж/кадр	Измерение мощности и энергии
Связь	RTT, потери пакетов	Измерения в заданных сетевых условиях
Эксплуатация	Температура, троттлинг	Длительный прогон под нагрузкой
Надёжность	Доля успешных циклов	Протокол повторных испытаний

Технические показатели и критерии сравнения (продолжение)

Средняя задержка: $\bar{L} = (1/N)\,\Sigma_i L_i$. Для систем реального времени важны не только среднее значение, но и 95-й и 99-й процентили: L_{95} и L_{99}. Сквозная задержка может быть представлена как Ltotal = Lcapture + Lpre + Linfer + Lpost + Lnet + Lact. Участки следует измерять отдельно.

Частота обработки FPS = Nframes / T. Если система обрабатывает кадры последовательно, оценка FPS ≈ 1000/Ltotal применима только при отсутствии параллелизма и очередей. Энергия на один обработанный кадр Eframe = Pavg / FPS, если Pavg измерена в ваттах, а FPS — в кадрах/с.

Для классификации Precision = TP/(TP+FP), Recall = TP/(TP+FN), F1 = 2PR/(P+R). Для обнаружения объектов дополнительно используются IoU и mAP с указанием порога IoU и протокола усреднения. Нельзя подменять mAP точностью классификации.

4 Верификация и валидация прототипов

Верификация отвечает на вопрос «соответствует ли реализация спецификации?», валидация — «решает ли система практическую задачу в реальных условиях?». Например, успешный запуск модели на Raspberry Pi подтверждает работоспособность программного контура, но не доказывает пригодность к промышленной эксплуатации при запылённости, вибрации и изменении освещения.

План эксперимента должен включать гипотезу, критерии приёмки, контрольные факторы, число повторов, способы регистрации данных и правила исключения аномалий. Наборы обучения, валидации и тестирования следует разделять так, чтобы изображения одного и того же объекта или близкие кадры не создавали утечку данных между выборками.

Для устойчивой оценки применяют сценарии: нормальная работа; плохое освещение; частичное перекрытие объекта; ухудшение сети; кратковременная потеря питания; перегрев; отказ датчика. Для каждого сценария фиксируют качество ИИ, задержку, состояние управляющего контура и восстановление после отказа.

Верификация и валидация прототипов (продолжение)

4.1. Статистическая обработка

Для повторных испытаний вычисляют выборочное среднее $\bar{x} = (1/n)\Sigma x_i$ и стандартное отклонение $s = \sqrt{[\Sigma(x_i - \bar{x})^2/(n-1)]}$. При сравнении двух платформ следует учитывать доверительные интервалы и характер распределения данных. Для задержек с асимметричным распределением полезно сообщать медиану и межквартильный размах наряду с процентилями.

Необходимо различать статистическую значимость и практическую ценность: снижение задержки на несколько миллисекунд может быть статистически подтверждено, но не менять безопасность или производительность процесса. Критерий приёмки должен быть связан с требованиями объекта управления.

6 V-модель разработки и проверки



Рис. 27. Каждому уровню проектирования соответствует уровень проверки.

5 Надёжность, безопасность и киберзащита

IoT-система расширяет поверхность атаки: устройства доступны через сеть, обмениваются телеметрией, хранят конфигурации и иногда выполняют удалённые команды. Поэтому необходимо применять аутентификацию, разграничение доступа, шифрование каналов, управление ключами, безопасные обновления и журналирование действий.

При анализе отказов выделяют: ложное распознавание, отсутствие распознавания, потерю связи, отказ питания, зависание процесса, переполнение очереди и некорректную команду исполнительному механизму. Для каждого отказа определяют последствия, способы обнаружения и безопасное состояние. Важен принцип fail-safe: при неопределённости система не должна создавать опасное управляющее воздействие.

При использовании камер требуется учитывать защиту персональных данных и конфиденциальность изображений. Данные для обучения должны собираться на законных основаниях; хранение изображений следует ограничивать целью и сроком исследования.

6 Технико-экономическая оценка внедрения

Ценность прототипа определяется не наличием нейронного ускорителя, а измеримым улучшением производственного процесса: сокращением брака, простоев, энергопотребления, затрат на инспекцию либо повышением безопасности. Экономическая оценка должна учитывать стоимость оборудования, интеграции, программного обеспечения, сопровождения, сетевой инфраструктуры и обучения персонала.

Совокупная стоимость владения за период H: $TCO = CAPEX + \Sigma_t(OPEX_t + Maintenance_t + Energy_t)$. Упрощённый срок окупаемости $PB = I_0 / Bannual$ применим при приблизительно постоянном годовом чистом эффекте $Bannual > 0$. Для проектов с разновременными потоками предпочтительна чистая приведённая стоимость: $NPV = -I_0 + \Sigma_t CF_t/(1+r)^t$.

Перед масштабированием пилота рекомендуется проверить технологическую готовность, воспроизводимость результатов, возможность сервисного обслуживания и совместимость с существующими PLC/SCADA/MES. Пилотное решение может иметь высокую точность распознавания, но низкую экономическую ценность при дорогой интеграции.

7 Интеллектуальная собственность и коммерциализация

В результате разработки IoT-прототипа могут возникнуть различные объекты интеллектуальной собственности: исходный код, техническая документация, архитектурные решения, схемы, базы данных, обученные модели, дизайн корпуса и фирменные обозначения. Правовой режим каждого объекта различается и зависит от применимого законодательства и договоров с участниками проекта.

Патентная охрана технического решения требует отдельной оценки патентоспособности; публикация до подачи заявки в ряде юрисдикций может повлиять на новизну. Авторское право обычно охраняет форму выражения программного кода и документации, но не любую идею алгоритма как таковую. Для конфиденциальных технологических параметров может использоваться режим коммерческой тайны при соблюдении организационных мер.

Использование YOLO, OpenCV, TensorFlow Lite, PyTorch и иных компонентов требует проверки лицензий конкретных версий, весов моделей и наборов данных. Открытый исходный код не означает отсутствия ограничений на распространение или коммерческое использование. Особенно важно отдельно проверять лицензию программного пакета, предобученных весов и датасета.

Интеллектуальная собственность и коммерциализация (продолжение)

7.1. Документирование результатов

Для защиты и последующей передачи технологии необходимо сохранять историю разработки, версии исходного кода, схемы, спецификации оборудования, протоколы испытаний, происхождение датасетов, описание обучающих процедур и договорённости о правах участников. Перед публикацией статьи или демонстрацией на выставке целесообразно провести внутреннюю проверку возможности патентования.

9 Объекты охраны интеллектуальной собственности

Изобретение	**Полезная модель**
новое техническое решение	конструкция устройства
Программа для ЭВМ	**Коммерческая тайна**
авторское право на код	режим конфиденциальности

Рис. 28. Формы правовой охраны выбираются по характеру результата; детали уточняются у патентного специалиста.

8 Расчётный пример: система визуального контроля конвейера

Рассмотрим учебный прототип: камера фиксирует изделия, модель обнаружения объектов определяет наличие дефекта, Raspberry Pi с аппаратным ускорителем формирует сигнал для контроллера сортировочного механизма. Следующие численные данные являются условными и служат только для иллюстрации методики.

8

Расчётный пример: система визуального контроля конвейера (продолжение)

Показатель	Платформа A	Платформа B
Средняя задержка, мс	95	58
95-й процентиль, мс	140	83
Производительность, кадр/с	10,5	17,2
Средняя мощность, Вт	8	12
F1 на тестовом наборе	0,91	0,92

8 Расчётный пример: система визуального контроля конвейера (продолжение)

Для платформы A энергия на кадр составляет 8/10,5 ≈ 0,76 Дж/кадр, для платформы B — 12/17,2 ≈ 0,70 Дж/кадр. Следовательно, более высокая мгновенная мощность платформы B не обязательно означает худшую энергоэффективность на одну обработанную единицу. При этом необходимо учитывать энергопотребление в ожидании и фактический режим работы.

Предположим, допустимая задержка от камеры до исполнительного решения равна 100 мс. Платформа B удовлетворяет условию по 95-му процентилю (83 мс), а платформа A — нет (140 мс). Однако окончательное решение требует измерить задержку исполнительного механизма, надёжность и качество обнаружения на реальном конвейере.

Если инвестиции в систему составляют условно 2 400 000 тенге, а годовая экономия за вычетом эксплуатационных расходов — 800 000 тенге, простой срок окупаемости PB = 2 400 000 / 800 000 = 3 года. Этот показатель не учитывает дисконтирование и риски; для инвестиционного решения необходима оценка NPV по сценариям.

9 Методика самостоятельного исследования докторанта

Предлагается разработать паспорт IoT-прототипа на базе нейрокомпьютера и выполнить сопоставление двух вариантов аппаратного обеспечения. Исследование должно содержать схему системы, обоснование модели ИИ, описание тестового набора, измерение задержки и энергопотребления, анализ ошибок и расчёт экономического эффекта. Дополнительно следует составить перечень используемых лицензий и потенциальных результатов интеллектуальной деятельности.

Рекомендуемая структура отчёта: постановка задачи; обзор решений; техническая архитектура; материалы и методы; результаты экспериментов; обсуждение ограничений; экономическая оценка; анализ интеллектуальной собственности; выводы. Все количественные утверждения необходимо подтверждать собственными измерениями или проверяемыми источниками.

12 Заключение

Оценка интеллектуального IoT-прототипа должна быть комплексной. Точность нейронной модели, производительность вычислительного устройства, задержка связи, безопасность исполнительного контура, стоимость владения и правовой статус разработки образуют единую систему критериев. Только согласованная проверка этих характеристик позволяет аргументированно обосновать внедрение технологии в промышленную практику.

13 Контрольные вопросы

1. Что отличает исследовательский IoT-прототип от демонстрационного макета?
2. Какие функциональные блоки входят в архитектуру интеллектуальной IoT-системы?
3. Чем нейрокомпьютер отличается от микроконтроллера с TinyML?
4. Почему номинальный TOPS не определяет фактический FPS приложения?
5. Из каких компонентов складывается сквозная задержка?
6. Как вычислить энергию на один обработанный кадр?
7. Чем верификация отличается от валидации?
8. Какие метрики следует использовать для оценки обнаружения объектов?
9. Почему при измерении задержек важен 95-й процентиль?
10. Какие сценарии отказов необходимо предусмотреть для промышленного прототипа?
11. Что означает принцип fail-safe?

13 Контрольные вопросы (продолжение)

12. Какие затраты включаются в совокупную стоимость владения?
13. Как интерпретировать NPV и простой срок окупаемости?
14. Какие результаты разработки могут охраняться как интеллектуальная собственность?
15. Почему требуется проверять лицензии кода, весов модели и датасетов по отдельности?

14 Рекомендуемая литература и ресурсы

1. Перри Ли. Архитектура интернета вещей. М.: ДМК Пресс, 2018.

2. Матюшин А. О. Программирование микроконтроллеров: стратегия и тактика. М.: ДМК, 2016.

3. Документация OpenCV: https://docs.opencv.org/

4. Документация Raspberry Pi: https://www.raspberrypi.com/documentation/

5. Документация TensorFlow Lite: https://www.tensorflow.org/lite

6. Документация Ultralytics: https://docs.ultralytics.com/ (условия лицензирования проверяются для используемой версии).

7. WIPO — ресурсы по интеллектуальной собственности: https://www.wipo.int/