ЛЕКЦИЯ 8

НЕЙРОННЫЕ СЕТИ И ТЕХНОЛОГИИ ГЛУБОКОГО ОБУЧЕНИЯ

Основные понятия • определения • схемы • таблицы • графики • примеры

Преподаватель: проф.Ахмедиярова А.Т.

Структура и основные разделы

01 Искусственные нейронные сети

02 Искусственный нейрон

03 Биологическая аналогия и ограничения аналогии

04 Персептрон

05 Слои нейронной сети

06 Многослойный персептрон MLP

07 Функции активации

08 Прямое распространение

09 Функция потерь

10 Обучение нейронной сети

11 Алгоритм обратного распространения ошибки

12 Градиентный спуск

Искусственные нейронные сети

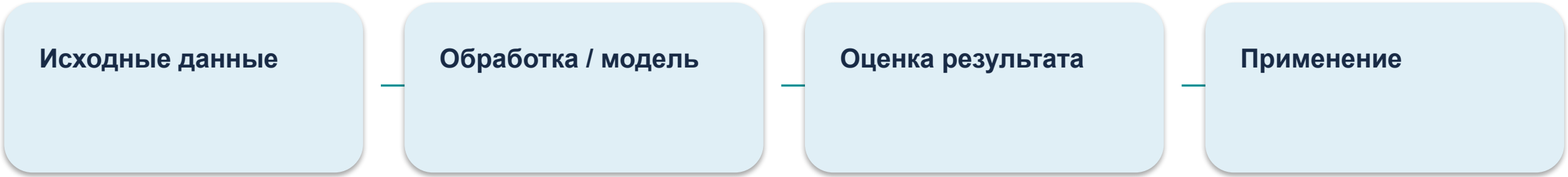
Искусственная нейронная сеть (ИНС) — вычислительная модель, состоящая из взаимосвязанных вычислительных элементов — искусственных нейронов. Нейроны получают входные значения, преобразуют их с учётом весовых коэффициентов и передают результат следующим элементам сети.

Ключевая идея: Искусственная нейронная сеть (ИНС) — вычислительная модель, состоящая из взаимосвязанных вычислительных элементов — искусстве

Искусственные нейронные сети — продолжение

Нейронные сети используются для классификации, аппроксимации функций, распознавания образов, обработки изображений, текста и последовательностей. В загруженных учебных материалах отдельно рассматриваются искусственный нейрон, синаптические связи, функции активации, однослойные и многослойные сети, обучение с учителем и алгоритм обратного распространения ошибки.

Схема: Искусственные нейронные сети



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Искусственный нейрон

Искусственный нейрон получает несколько входных сигналов x_1, x_2, …, x_n. Каждому входу соответствует вес w_i. Сначала вычисляется взвешенная сумма входов и добавляется смещение b: $z = w_1x_1 + w_2x_2 + … + w_nx_n + b$ После этого к z применяется функция активации f(z), и формируется выход нейрона: $y = f(z)$ Вес показывает, насколько сильно соответствующий вход влияет на результат.

Ключевая идея: Искусственный нейрон получает несколько входных сигналов x_1, x_2, …, x_n. Каждому входу соответствует вес w_i. Сначала вычисляет

Искусственный нейрон — продолжение

Смещение позволяет сдвигать границу решения. В процессе обучения веса и смещения изменяются так, чтобы уменьшить ошибку модели.

Сравнительная таблица: Искусственный нейрон

Искусственный нейрон	Искусственный нейрон получает несколько входных сигналов x_1, x_2, …, x_n. Каждому входу соответствует вес w_i. Сначала
Биологическая аналогия и ограничения аналогии	$z = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$
Персептрон	После этого к z применяется функция активации f(z), и формируется выход нейрона:
Слои нейронной сети	$y = f(z)$

Биологическая аналогия и ограничения аналогии

Термины «нейрон», «синапс» и «сеть» исторически связаны с биологической аналогией. В искусственной модели входные связи сопоставляются с синапсами, а веса — с силой влияния сигналов. Однако искусственный нейрон является математической вычислительной моделью и не воспроизводит полностью работу биологической нервной системы.

Персептрон

Персептрон — одна из простейших моделей искусственного нейрона, применяемая для решения задач классификации. Он вычисляет взвешенную сумму входов и на основании функции активации принимает решение о принадлежности объекта к классу. Пример. Пусть требуется определить, является ли сетевое событие подозрительным.

Ключевая идея: Персептрон — одна из простейших моделей искусственного нейрона, применяемая для решения задач классификации. Он вычисляет взв

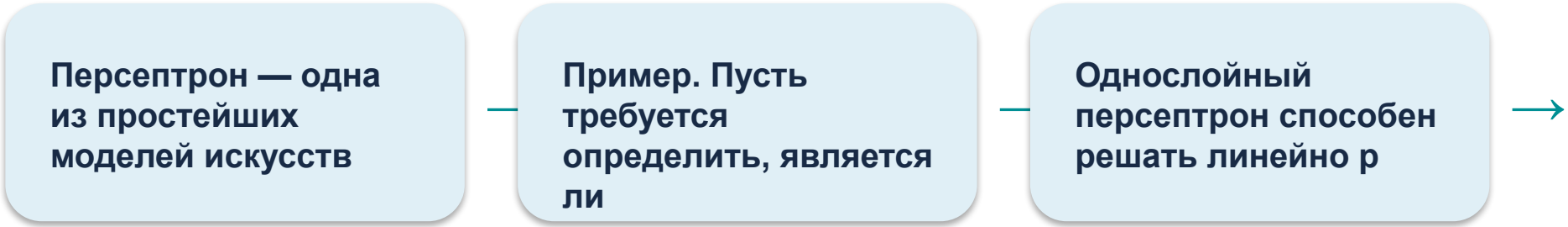
Персептрон — продолжение

Входами могут быть количество неуспешных входов, число соединений за короткий интервал и объём переданных данных. Персептрон формирует взвешенную комбинацию признаков и выдаёт решение. Однослойный персептрон способен решать линейно разделимые задачи. Классический пример ограничения — функция XOR: её классы нельзя разделить одной прямой в двумерном пространстве.

Персептрон — продолжение

Для моделирования нелинейных зависимостей используются многослойные сети.

Схема: Персептрон



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Слои нейронной сети

Нейроны объединяются в слои. Входной слой получает признаки объекта. Один или несколько скрытых слоёв выполняют промежуточные преобразования. Выходной слой формирует прогноз. Если сеть содержит несколько скрытых слоёв, говорят о глубокой нейронной сети. Термин «глубина» связан прежде всего с количеством последовательных уровней преобразования данных.

Ключевая идея: Нейроны объединяются в слои. Входной слой получает признаки объекта. Один или несколько скрытых слоёв выполняют промежуточные

Многослойный персептрон MLP

Multilayer Perceptron (MLP) — многослойная сеть прямого распространения. Каждый нейрон слоя получает значения от предыдущего слоя, вычисляет взвешенную сумму и применяет функцию активации. MLP может моделировать нелинейные зависимости благодаря скрытым слоям и нелинейным функциям активации.

Ключевая идея: Multilayer Perceptron (MLP) — многослойная сеть прямого распространения. Каждый нейрон слоя получает значения от предыдущего

Многослойный персептрон MLP — продолжение

Он применяется к задачам классификации и регрессии, особенно когда данные представлены фиксированным набором числовых признаков.

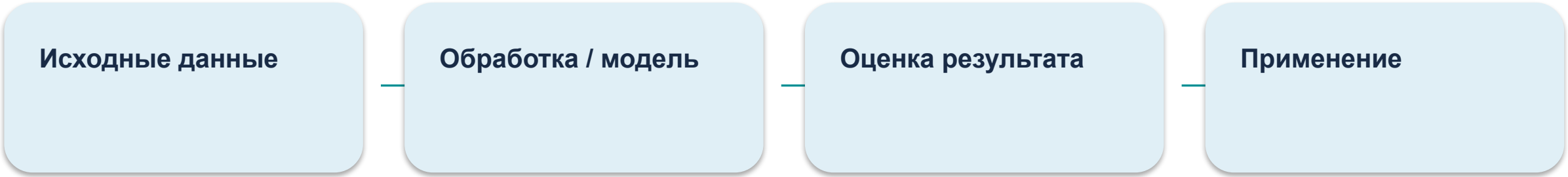
Сравнительная таблица: Многослойный персептрон MLP

Многослойный персептрон MLP	Multilayer Perceptron (MLP) — многослойная сеть прямого распространения. Каждый нейрон слоя получает значения от пр
Функции активации	MLP может моделировать нелинейные зависимости благодаря скрытым слоям и нелинейным функциям активации. Он применяет

Функции активации

Если каждый слой выполнял бы только линейное преобразование, последовательность нескольких таких слоёв всё равно могла бы быть сведена к одному линейному преобразованию. Нелинейные функции активации позволяют сети описывать более сложные зависимости.

Схема: Функции активации



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Прямое распространение

При прямом распространении (forward propagation) данные последовательно проходят от входного слоя через скрытые слои к выходу. Каждый слой вычисляет свои значения на основе результатов предыдущего слоя. Например, сеть получает четыре признака сетевого соединения.

Ключевая идея: При прямом распространении (forward propagation) данные последовательно проходят от входного слоя через скрытые слои к выходу

Прямое распространение — продолжение

Первый скрытый слой формирует промежуточные комбинации этих признаков, второй слой преобразует их далее, а выходной слой выдаёт вероятность принадлежности соединения к классу атаки.

Функция потерь

Для обучения необходимо количественно измерить расхождение между прогнозом модели и правильным ответом. Эту роль выполняет функция потерь (loss function).

Обучение нейронной сети

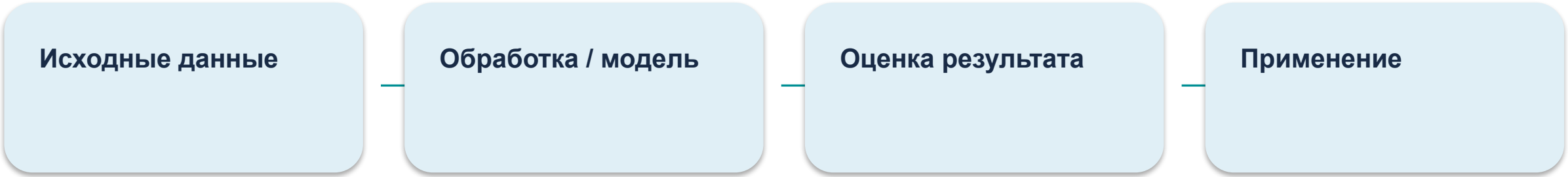
Обучение сети состоит в настройке весов и смещений на обучающей выборке. Для каждого набора входов выполняется прямое распространение, вычисляется ошибка, после чего определяется, как следует изменить параметры, чтобы уменьшить функцию потерь. Этот процесс повторяется многократно. Один полный проход по обучающей выборке называется эпохой.

Ключевая идея: Обучение сети состоит в настройке весов и смещений на обучающей выборке. Для каждого набора входов выполняется прямое распрос

Обучение нейронной сети — продолжение

Данные обычно подаются небольшими группами — mini-batches.

Схема: Обучение нейронной сети



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Сравнительная таблица: Обучение нейронной сети

Обучение нейронной сети	Обучение сети состоит в настройке весов и смещений на обучающей выборке. Для каждого набора входов выполняется прям
Алгоритм обратного распространения ошибки	Этот процесс повторяется многократно. Один полный проход по обучающей выборке называется эпохой. Данные обычно пода

Алгоритм обратного распространения ошибки

Backpropagation — алгоритм вычисления градиентов функции потерь по параметрам многослойной нейронной сети. Ошибка распространяется от выходного слоя к предыдущим слоям, что позволяет определить вклад отдельных весов в итоговую ошибку. Концептуальная последовательность выглядит так: 1. Выполнить прямое распространение и получить прогноз. 2. Вычислить функцию потерь. 3.

Ключевая идея: Backpropagation — алгоритм вычисления градиентов функции потерь по параметрам многослойной нейронной сети. Ошибка распростран

Алгоритм обратного распространения ошибки — продолжение

Определить градиенты функции потерь относительно параметров выходного слоя. 4. По правилу цепочки вычислить градиенты для предыдущих слоёв. 5. Передать рассчитанные градиенты оптимизатору для обновления параметров. 6. Повторять процесс на следующих mini-batches и эпохах. Backpropagation не является самостоятельным оптимизатором.

Алгоритм обратного распространения ошибки — продолжение

Он вычисляет градиенты; изменение параметров выполняет выбранный алгоритм оптимизации.

Градиентный спуск

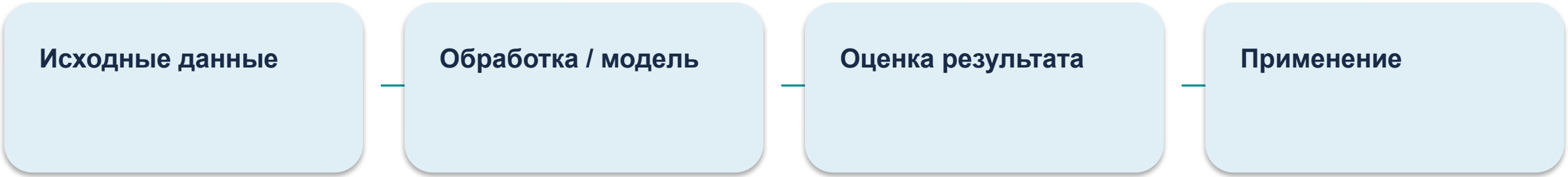
Градиент показывает направление наиболее быстрого увеличения функции. Чтобы уменьшать ошибку, параметры изменяются в противоположном направлении: w_new = w_old - η · ∂L/∂w Здесь η — learning rate, или скорость обучения. Слишком большое значение может сделать обучение нестабильным, а слишком малое — существенно замедлить его.

Ключевая идея: Градиент показывает направление наиболее быстрого увеличения функции. Чтобы уменьшать ошибку, параметры изменяются в противоп

Оптимизаторы

Выбор оптимизатора и learning rate относится к настройке модели. Один и тот же оптимизатор не гарантирует лучший результат для всех задач.

Схема: Оптимизаторы



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Эпоха, batch и итерация

Например, если обучающая выборка содержит 10 000 объектов, а batch size равен 100, то одна эпоха содержит примерно 100 итераций.

Сравнительная таблица: Эпоха, batch и итерация

Эпоха, batch и итерация	Например, если обучающая выборка содержит 10 000 объектов, а batch size равен 100, то одна эпоха содержит примерно

Недообучение и переобучение

Недообучение (underfitting) возникает, когда модель недостаточно хорошо описывает даже обучающие данные. Причинами могут быть слишком простая архитектура, недостаточное время обучения или неподходящие признаки. Переобучение (overfitting) возникает, когда модель слишком хорошо подстраивается под обучающую выборку, но хуже работает на новых данных.

Ключевая идея: Недообучение (underfitting) возникает, когда модель недостаточно хорошо описывает даже обучающие данные. Причинами могут быть

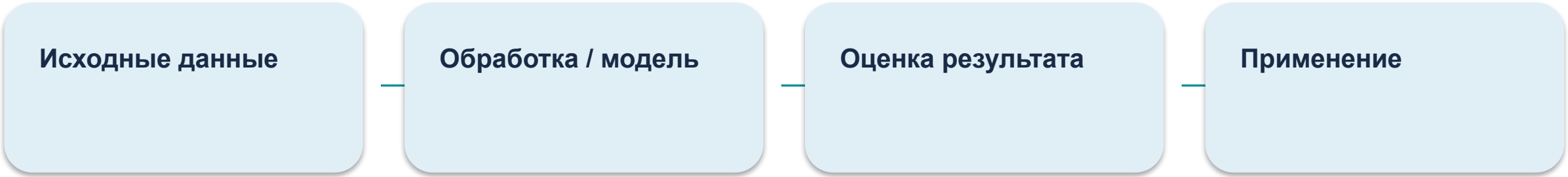
Недообучение и переобучение — продолжение

В загруженных материалах проблемы недообучения, переобучения и валидации выделены как важная часть обучения нейронных сетей.

Методы уменьшения переобучения

Для повышения способности модели к обобщению применяются различные подходы: увеличение и улучшение обучающих данных, регуляризация, Dropout, ранняя остановка (Early Stopping), уменьшение избыточной сложности сети и контроль качества на валидационной выборке.

Схема: Методы уменьшения переобучения



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Глубокое обучение

Глубокое обучение (Deep Learning) — направление машинного обучения, основанное на многослойных нейронных сетях. Глубокие модели способны последовательно формировать всё более сложные представления исходных данных. Для изображения ранние уровни могут реагировать на простые локальные структуры, а последующие — формировать более сложные признаки.

Ключевая идея: Глубокое обучение (Deep Learning) — направление машинного обучения, основанное на многослойных нейронных сетях. Глубокие моде

Глубокое обучение — продолжение

Для текста и последовательностей глубокие архитектуры обучаются представлениям, отражающим контекст и зависимости между элементами.

Основные архитектуры глубокого обучения

Загруженное пособие по ИИ включает MLP и backpropagation в главу о нейронных сетях, а другой учебный материал дополнительно рассматривает CNN, RNN, LSTM и GRU при обучении нейронных сетей.

Сравнительная таблица: Основные архитектуры глубокого обучения

Основные архитектуры глубокого обучения	Загруженное пособие по ИИ включает MLP и backpropagation в главу о нейронных сетях, а другой учебный материал допол

Свёрточные нейронные сети CNN

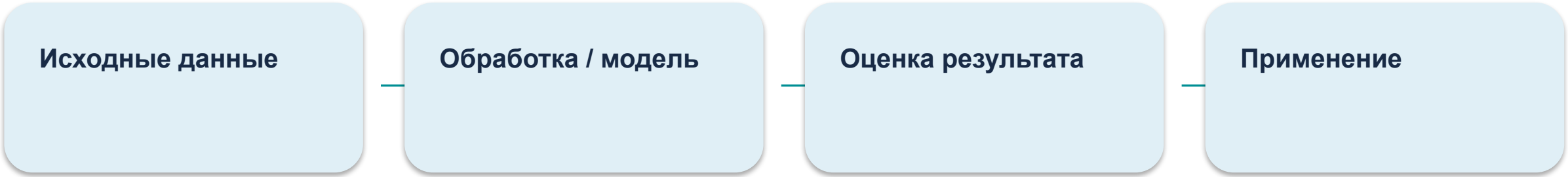
CNN применяют свёрточные операции, позволяющие обнаруживать локальные закономерности. Особенно широко они используются при обработке изображений. В задачах информационной безопасности свёрточные модели также могут применяться к специально сформированным матрицам признаков, последовательностям сигналов и другим структурированным представлениям.

Ключевая идея: CNN применяют свёрточные операции, позволяющие обнаруживать локальные закономерности. Особенно широко они используются при об

Свёрточные нейронные сети CNN — продолжение

Основные компоненты CNN — свёрточные слои, функции активации, операции уменьшения пространственного размера и итоговые слои классификации.

Схема: Свёрточные нейронные сети CNN



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Рекуррентные сети RNN, LSTM и GRU

RNN предназначены для последовательных данных и используют информацию о предыдущих шагах. Это делает их применимыми к временным рядам, тексту, речи и последовательностям событий. Обычные RNN испытывают трудности при моделировании длинных зависимостей.

Ключевая идея: RNN предназначены для последовательных данных и используют информацию о предыдущих шагах. Это делает их применимыми к временн

Рекуррентные сети RNN, LSTM и GRU — продолжение

Архитектуры LSTM и GRU вводят специальные механизмы управления состоянием, позволяющие лучше сохранять существенную информацию на более длинных интервалах.

Нейронные сети в информационной безопасности

Пример. Для классификации сетевого трафика MLP может получать duration, packets, bytes, failed_logins и другие числовые признаки. Выходной слой формирует вероятность классов normal и attack. Качество оценивается на данных, не использованных при настройке параметров.

TensorFlow и Keras

TensorFlow — программная платформа для построения, обучения и применения моделей машинного и глубокого обучения. Keras предоставляет высокоуровневый интерфейс, позволяющий компактно описывать архитектуру нейронной сети.

Ключевая идея: TensorFlow — программная платформа для построения, обучения и применения моделей машинного и глубокого обучения. Keras предос

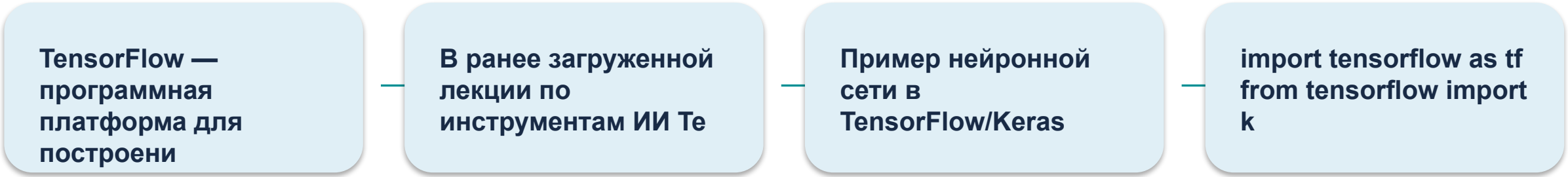
TensorFlow и Keras — продолжение

В ранее загруженной лекции по инструментам ИИ TensorFlow рассматривается как платформа для операций с тензорами и обучения моделей, а Keras — как высокоуровневый интерфейс создания нейронных сетей. Это соответствует структуре учебного пособия, где TensorFlow и Keras включены в главу о нейронных сетях и глубоком обучении.

TensorFlow и Keras — продолжение

Пример нейронной сети в TensorFlow/Keras import tensorflow as tf from tensorflow import keras from tensorflow.keras import layers model = keras.Sequential([layers.Input(shape=(10,)), layers.Dense(32, activation='relu'), layers.Dense(16, activation='relu'), layers.Dense(1, activation='sigmoid')]) model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']) history = model.fit(X_train, y_train, validation_split=0.2, epochs=30, batch_size=32) test_loss, test_accuracy = model.evaluate(X_test, y_test)

Схема: TensorFlow и Keras



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Сравнительная таблица: TensorFlow и Keras

TensorFlow и Keras	TensorFlow — программная платформа для построения, обучения и применения моделей машинного и глубокого обучения. Ke
Разбор примера	В ранее загруженной лекции по инструментам ИИ TensorFlow рассматривается как платформа для операций с тензорами и о
Как анализировать процесс обучения	Пример нейронной сети в TensorFlow/Keras
Преимущества и ограничения нейронных сетей	import tensorflow as tf from tensorflow import keras from tensorflow.keras import layers model = keras.Sequential([

Разбор примера

Пример многоклассовой классификации model = keras.Sequential([layers.Input(shape=(20,)), layers.Dense(64, activation='relu'), layers.Dense(32, activation='relu'), layers.Dense(4, activation='softmax')]) model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy']) Если необходимо различать четыре категории событий, выходной слой содержит четыре нейрона.

Ключевая идея: Пример многоклассовой классификации

Разбор примера — продолжение

Softmax формирует распределение оценок по классам, а класс с наибольшей оценкой может быть выбран как итоговый прогноз.

Как анализировать процесс обучения

Во время обучения необходимо отслеживать не только итоговую метрику на test, но и динамику loss и метрик на train и validation. Если train loss продолжает снижаться, а validation loss начинает расти, это типичный признак возможного переобучения. При сравнении моделей важно использовать одинаковое разбиение данных и не подбирать архитектуру по тестовой выборке.

Ключевая идея: Во время обучения необходимо отслеживать не только итоговую метрику на test, но и динамику loss и метрик на train и validatio

Как анализировать процесс обучения — продолжение

Test должен оставаться независимым источником финальной оценки.

Типичные ошибки при разработке нейронной сети

Считать увеличение числа слоёв гарантией повышения качества. Оценивать модель только на обучающей выборке. Использовать test для постоянного подбора архитектуры и гиперпараметров. Не масштабировать числовые признаки, когда это необходимо для устойчивого обучения. Выбирать функцию активации и функцию потерь без учёта типа задачи. Игнорировать дисбаланс классов.

Ключевая идея: Считать увеличение числа слоёв гарантией повышения качества.

Типичные ошибки при разработке нейронной сети — продолжение

Считать высокую accuracy достаточной для задачи обнаружения редких атак.
Продолжать обучение после явного ухудшения качества на validation без анализа переобучения.

Сравнительная таблица: Типичные ошибки при разработке нейронной сети

Типичные ошибки при разработке нейронной сети	Считать увеличение числа слоёв гарантией повышения качества.
Сравнение классического машинного и глубокого обучения	Оценивать модель только на обучающей выборке.
Список источников	Использовать test для постоянного подбора архитектуры и гиперпараметров.

Сравнение классического машинного и глубокого обучения

Основные выводы: Искусственный нейрон вычисляет взвешенную сумму входов и применяет функцию активации. Персептрон является простейшей нейросетевой моделью и ограничен линейной разделимостью. Многослойные сети используют скрытые слои и нелинейные функции активации. Обучение включает прямое распространение, вычисление функции потерь, backpropagation и обновление параметров.

Ключевая идея: Основные выводы:

Сравнение классического машинного и глубокого обучения — продолжение

Learning rate, batch size, число эпох и архитектура являются важными параметрами процесса обучения. Недообучение и переобучение контролируются по обучающим и валидационным данным. Глубокое обучение использует многослойные архитектуры, включая MLP, CNN, RNN/LSTM/GRU и Transformer. TensorFlow и Keras позволяют реализовывать и обучать нейронные сети средствами Python. Контрольные вопросы: 1.

Сравнение классического машинного и глубокого обучения — продолжение

Что такое искусственная нейронная сеть? 2. Из каких элементов состоит искусственный нейрон? 3. Какую роль выполняют веса? 4. Для чего используется смещение b? 5. Что такое функция активации? 6. Что представляет собой персептрон? 7. Что означает линейная разделимость? 8. Почему однослойный персептрон не решает XOR? 9. Какие слои содержит многослойная сеть? 10. Что такое MLP? 11.

Сравнение классического машинного и глубокого обучения — продолжение

Зачем нейронной сети нелинейные функции активации? 12. Чем sigmoid отличается от tanh? 13. Как работает ReLU? 14. Для чего используется Softmax? 15. Что такое forward propagation? 16. Что такое функция потерь? 17. Какие функции потерь применяются для классификации и регрессии? 18. Что такое эпоха? 19. Что такое batch и batch size? 20. Что такое backpropagation? 21.

Сравнение классического машинного и глубокого обучения — продолжение

Какую роль играет правило цепочки? 22. Чем backpropagation отличается от оптимизатора? 23. Что такое градиентный спуск? 24. Как learning rate влияет на обучение? 25. Что такое Adam? 26. Чем недообучение отличается от переобучения? 27. Как обнаружить переобучение по train и validation?

Список источников

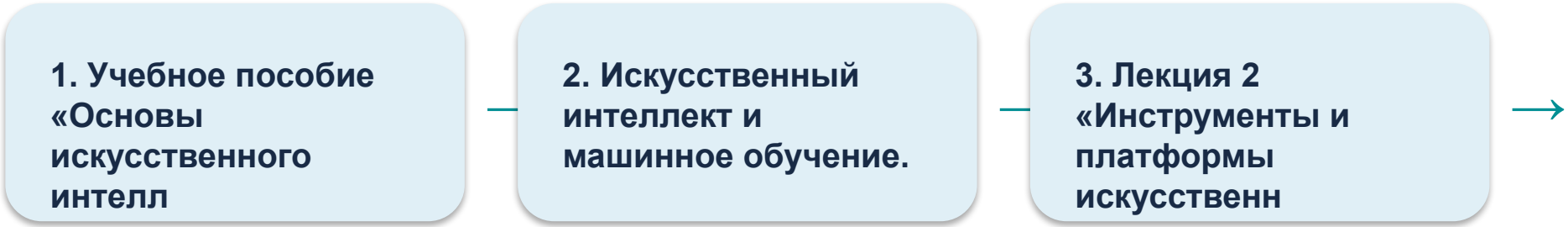
1. Учебное пособие «Основы искусственного интеллекта». Глава 8 «Нейронные сети и глубокое обучение»: персептрон, MLP, Backpropagation, TensorFlow и Keras. 2. Искусственный интеллект и машинное обучение. Учебно-методические материалы.

Ключевая идея: 1. Учебное пособие «Основы искусственного интеллекта». Глава 8 «Нейронные сети и глубокое обучение»: персептрон, MLP, Backpro

Список источников — продолжение

Разделы «Искусственные нейронные сети», «Персептроны», «Обучение искусственных нейронных сетей»: искусственный нейрон, функции активации, однослойные и многослойные сети, обратное распространение ошибки, CNN, RNN, LSTM, GRU, недообучение и переобучение. 3. Лекция 2 «Инструменты и платформы искусственного интеллекта». Разделы TensorFlow, Keras и PyTorch.

Схема: Список источников



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

График: иллюстрация зависимости



Иллюстративные значения (не результаты эксперимента).

Итоги и вопросы для обсуждения

1. Объясните понятие «Искусственные нейронные сети» и приведите пример.
2. Объясните понятие «Искусственный нейрон» и приведите пример.
3. Объясните понятие «Биологическая аналогия и ограничения аналогии» и приведите пример.
4. Объясните понятие «Персептрон» и приведите пример.
5. Объясните понятие «Слои нейронной сети» и приведите пример.
6. Объясните понятие «Многослойный персептрон MLP» и приведите пример.
7. Объясните понятие «Функции активации» и приведите пример.