УДК 303.732
DOI: 10.22213/2410-9304-2025-4-79-85

Прогнозирование производственных показателей с использованием *MLP* и *LSTM*: анализ подходов проектирования

Д. С. Пономарёв, кандидат технических наук, ИжГТУ имени М. Т. Калашникова;
Научно-исследовательский институт Федеральной службы исполнения наказаний, Ижевск, Россия

В статье рассматривается задача прогнозирования производственных показателей с применением методов глубокого обучения на основе полносвязных и рекуррентных нейронных сетей. Основное внимание уделено практической реализации моделей регрессии с использованием многослойного персептрона (MLP) и рекуррентной нейронной сети с долгой краткосрочной памятью (LSTM), а также анализу подходов их проектирования с учетом особенностей исходных данных. Работа охватывает полный цикл построения моделей: сбор и предварительную обработку данных, нормализацию признаков, формирование обучающих и тестовых выборок, построение архитектур нейронных сетей, настройку гиперпараметров, процесс обучения, оценку качества прогнозирования и визуализацию результатов. В качестве прикладного примера рассмотрена задача прогнозирования объема производства на основе реальных исторических данных, характеризующих деятельность производственных объектов за многолетний период. Для моделей MLP данные рассматривались как независимые наблюдения, тогда как для LSTM применялось формирование временных последовательностей фиксированной длины, позволяющее учитывать динамику и долгосрочные зависимости показателей. Проведен сравнительный анализ архитектурных решений, вычислительной сложности, требований к ресурсам и точности прогнозирования. Показано, что LSTM демонстрирует более высокую точность при работе с временными рядами за счет учета временного контекста, однако требует значительно больших вычислительных затрат. В то же время MLP характеризуется простотой реализации и более высокой скоростью обучения, что делает ее целесообразной для задач со слабо выраженной временной структурой данных. Результаты исследования могут быть использованы при проектировании интеллектуальных информационно-аналитических систем для производственного сектора и служат практическим ориентиром при выборе архитектуры нейронной сети в зависимости от характера данных и требований к точности и ресурсам.

Ключевые слова: многослойный персептрон, рекуррентные сети, информационная система, производство, проектирование, системный анализ, статистика.

Введение

На сегодняшний день прогнозирование производственных процессов является неотъемлемой частью практически любого масштабного производства. Решение подобных задач зачастую требует применения сложных математических моделей, способных учитывать множество факторов и исторические данные. В данном контексте методы глубокого обучения, такие как полносвязные и рекуррентные нейронные сети, могут выступить в качестве ключевых инструментов для эффективного решения.

В работе рассмотрена практическая реализация полносвязной нейронной сети на примере многослойного персептрона (в публикациях наиболее часто используется термин – *Multi-layerPerceptron* [1] (далее – *MLP*)) и рекуррентной нейронной сети на примере сети с долгой краткосрочной памятью (в научных публикациях для которой часто используется термин – *Long short-termmemory* [2] (далее – *LSTM*)). Цель исследования – провести сравнительный анализ подходов к проектированию и применению многослойного персептрона (MLP) и рекуррентной нейронной сети с долгосрочной краткосрочной памятью (LSTM) для прогнозирования производственных показателей, оценить их точность, вычислительную сложность и целесообразность использования в зависимости от характера исходных данных.

Выбор *MLP* и *LSTM* в данном исследовании обусловлен их принципиально разными подходами к обработке данных, что позволяет наглядно сравнить эффективность моделей для разных типов задач [3]. *MLP* был выбран как базовая и широко применяемая архитектура для работы с изолированными данными, где не требуется учет временных зависимостей [4]. Его преимущество заключается в простоте, быстром обучении и способности выявлять сложные нелинейные взаимосвязи между признаками [5]. В то же время *LSTM* была включена в исследование как специализированная архитектура для обработки последовательностей и временных рядов, которая способна учитывать долгосрочные зависимости в данных [6]. Это особенно важно для производственных показателей, где текущие значения часто зависят от предыдущих состояний системы.

Сравнение этих двух подходов позволяет не только оценить их относительную эффективность для конкретной задачи прогнозирования, но и продемонстрировать ключевые различия в их применении: *MLP* лучше подходит для задач с независимыми наблюдениями, в то время как *LSTM* незаменим при работе с временными зависимостями. Такой выбор также отражает распространенную практику в анализе данных, где сначала пробуют базовые модели (вроде *MLP*), а затем, при необходимости учета временного контекста, переходят к специализированным архитектурам (*LSTM*). Выбор *LSTM* среди других архитектур рекуррентных нейронных сетей обусловлен ее способностью решать ключевую проблему классических *RNN* – затухание градиентов при обучении на длинных последовательностях [7, 8].

Научное обоснование алгоритмов построения и сравнение подходов проектирования для реализации MLP и LSTM

Методология построения MLP-модели. Архитектура многослойного перцептрона представляет собой композицию нелинейных преобразований. Формально модель *MLP* может быть описана следующим образом:

$$h^{(l)} = \varphi\left(W^{(l)}h^{(l-1)} + b^{(l)}\right), \quad (1)$$

где $h^{(l)}$ – выход l-го слоя; $W^{(l)}$ – матрица весов l-го слоя; $b^{(l)}$ – вектор смещений l-го слоя; φ – функция активации.

Для скрытых слоев рассмотрено применение функции активации *ReLU* (2):

$$\varphi\left(z\right) = \max\left(0, z\right), \quad (2)$$

выходной слой использует линейную активацию (3):

$$\hat{y} = W^{(l)}h^{(l-1)} + b^{(l)}. \quad (3)$$

Функция потерь представляет собой среднеквадратическую ошибку (4):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2. \quad (4)$$

Рассмотрим методологию построения LSTM. Архитектура *LSTM* описывается следующей системой уравнений.Блок управления забыванием (5):

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right), \quad (5)$$

блок управления входом (6):

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right), \quad (6)$$

блок кандидатов значений ячейки (7):

$$\hat{C}_t = \tanh\left(W_C\left[h_{t-1}, x_t\right] + b_C\right), \quad (7)$$

обновление состояния ячейки (8):

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t, \quad (8)$$

блок управления выходом (9):

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right), \quad (9)$$

скрытое состояние (10):

$$h_t = o_t \odot \tanh\left(C_t\right), \quad (10)$$

где σ – сигмоидальная функция активации, $\odot$ – поэлементное умножение.

Основное различие между архитектурами заключается в способе обработки временных зависимостей. В *MLP* предполагается условная независимость наблюдений, тогда как *LSTM* явно моделирует временные зависимости через механизм клеточной памяти. Градиент функции потерь в *LSTM* вычисляется с использованием алгоритма обратного распространения через время (*BPTT*):

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{T}\frac{\partial \mathcal{L}}{\partial h_t}\frac{\partial h_t}{\partial W}. \quad (11)$$

Обе архитектуры соответствуют современным принципам построения нейросетевых моделей. В качестве примера рассмотрена задача прогнозирования объема производства на основе исторических данных, реализация проведена при помощи *Python*.

Подготовка данных

Для обучения и тестирования моделей использовался массив данных о производственных показателях предприятий пенитенциарной системы за период с января 2019 по декабрь 2023 года. Источником данных послужили ведомственные статистические отчеты по производственной деятельности и трудовой адаптации осужденных, охватывающие 26 производственных объектов. Общий объем выборки составил 1560 наблюдений. В соответствии с общепринятой практикой, применяемой для проведения машинного обучения, исходные данные были разделены на обучающую и тестовую выборки в соотношении 80/20. Для обучения моделей использовались данные за 2019–2022 годы (1248 наблюдений), тогда как валидация проводилась на данных за 2023 год (312 наблюдений). Был сформирован *CSV*-файл с историческими данными о производстве, где каждая строка содержит информацию о производственных показателях за определенный месяц. В *CSV*-файле имеются параметры, которые соответствуют переменным: объем производства (*PV*); количество работников из числа спецконтингента (*Emp*); время простоя оборудования (*DH*); затраты на техническое обслуживание (*MC*). Статистические характеристики нормализованного набора данных представлены в табл. 1.

Таблица 1. **Статистические характеристики нормализованного набора данных**

Table 1. **Statistical characteristics of the normalized data set**

Параметр	*PV*	*Emp*	*DH*	*MC*
Среднее значение	0,008	–0,012	0,005	–0,003
Стандартное отклонение	1,005	0,998	1,007	1,002
Минимальное значение	–2,845	–2,912	-2,734	–2,801
Максимум	2,967	2,845	2,878	2,923

Нормализация данных проводилась по формуле (11):

$$X_{std} = \frac{X - \mu}{\sigma}, \quad (11)$$

где μ – среднее значение признака; σ – стандартное отклонение. Исходные данные представляли значительную вариативность производственных показателей: объем производства от 856 до 2845 единиц, количество работников от 98 до 298 человек, время простоя оборудования от 8 до 56 часов, затраты на обслуживание от 298 до 1034 тыс. руб.

Реализация *MLP* (полносвязной нейронной сети)

Для реализации *MLP* использовались библиотеки *TensorFlow* и *Keras*. Актуальность их применения была рассмотрена в опубликованных ранее работах [9, 10]. *TensorFlow* – это основная библиотека для работы с нейронными сетями и глубоким обучением, *Keras* использовалась как более высокоуровневая надстройка

над *TensorFlow*; *Sequential* – это класс, который позволяет создавать модели нейронных сетей слоями. Каждый слой добавляется последовательно. *Dense* – это тип слоя нейронной сети, где каждый нейрон связан со всеми нейронами предыдущего слоя [11]. Это стандартный слой для полносвязных нейронных сетей (*MLP*).*Sequential()* – создает пустую модель, к которой добавляются слои. Выбрана функция активации *ReLU*(*RectifiedLinearUnit*), она возвращает 0 для отрицательных значений и само значение для положительных, это помогает модели быстрее обучаться и избегать проблем с затухающими градиентами [12]. *Dense(1)* – выходной слой с одним нейроном. Так как в нашем случае поставленная задача представляет собой задачу регрессии, то выходной слой обычно не имеет функции активации (или использует линейную активацию), так как нам нужно предсказать непрерывное значение. В качестве примера был выбран оптимизатор *Adam*, обоснование актуальности его применения было рассмотрено в опубликованной ранее работе [13]. В качестве примера было поставлено 50 эпох и в дальнейшем их количество может быть скорректировано в зависимости от полученных результатов обучения и изначальных требований к точности моделей. Количество образцов, которые модель обрабатывает перед обновлением весов, – 32. Меньшее значение может улучшить обобщение, но замедляет обучение [14]. Доля данных для валидации – 20 %, данные из обучающего набора будут использоваться для проверки модели во время обучения (в соответствии с правилом Парето). Это помогает отслеживать, не переобучается ли модель.

Реализация *LSTM*

Для реализации модели с долгосрочной краткосрочной памятью (*LSTM*) требуется предварительное преобразование данных в формат временных последовательностей. В отличие от *MLP*, где каждый объект рассматривается изолированно, *LSTM* обрабатывает последовательности данных, что позволяет учитывать временные зависимости.

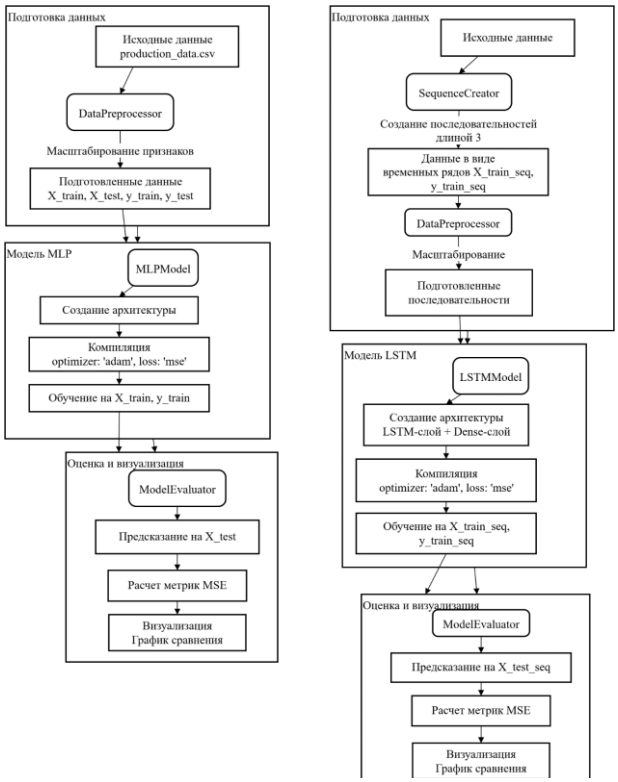
Подготовка данных для *LSTM* включает создание скользящих последовательностей фиксированной длины. В данном исследовании длина последовательности установлена равной 3 месяцам, что соответствует квартальной периодичности отчетности и обеспечивает сопоставимость с производственными циклами. Для этого применяется алгоритм преобразования, который для каждого момента времени формирует входную последовательность из заданного количества предыдущих наблюдений и соответствующее целевое значение.

Архитектура *LSTM*-модели строится с использованием последовательной структуры. Первый слой представляет собой *LSTM*-слой с 50 нейронами, что обеспечивает баланс между вычислительной сложностью и способностью модели выявлять временные зависимости. Форма входных данных задается в виде («длина_последовательности», «количество_признаков»). Выходной слой состоит из одного нейрона с линейной функцией активации, что стандартно для задач регрессии.

Процесс компиляции модели использует оптимизатор *Adam* и функцию потерь *MSE*, аналогично *MLP*-модели. Обучение проводится в течение 50 эпох с размером пакета 32 примера, при этом 20 % обучающей выборки выделяется для валидации.

Проектирование систем для реализации *MLP*и *LSTM*

На рисунке представлены упрощенные структурные схемы систем на основе *MLP* и *LSTM*.



Упрощенные структурные схемы систем на основе MLP и LSTM

Simplified structural diagrams of systems based on MLP and LSTM

Обе системы реализуют сквозной процесс, начинающийся с этапа подготовки данных. Для *MLP* этот этап включает загрузку исходных данных и их масштабирование, в результате чего формируются готовые к обучению наборы, где каждый объект рассматривается как независимое наблюдение. В случае *LSTM* обработка данных расширяется критически важным дополнительным шагом – преобразованием временного ряда в последовательности заданной длины, что позволяет модели анализировать временные контексты и зависимости. Последующее масштабирование применяется уже к этим сформированным последовательностям.

Следующим этапом является построение и обучение модели. Архитектура *MLP* представляет собой последовательность полносвязных слоев, которые выполняют нелинейное преобразование изолированных входных векторов. Модель компилируется с оптимизатором *Adam* и функцией потерь *MSE*, после чего обучается на подготовленных статических данных. В отличие от нее, архитектура *LSTM* в качестве первого слоя использует специализированный *LSTM*-

слой, предназначенный для обработки последовательностей и учета долгосрочных временных зависимостей. За ним следуют полносвязные слои для финального преобразования в результат прогноза. Процесс компиляции аналогичен *MLP*, однако обучение происходит на последовательностях и является более вычислительно затратным из-за рекуррентной природы сети.

Завершающим этапом в обоих подходах является оценка и визуализация результатов. Обученная модель используется для прогнозирования на тестовой выборке, после чего рассчитывается метрика качества *MSE* и строится сравнительный график фактических и предсказанных значений. Это позволяет количественно и качественно оценить эффективность каждой модели.

Сравнительное расположение схем *MLP* и *LSTM* наглядно подчеркивает их структурное сходство как целостных процессов машинного обучения и одновременно фундаментальное различие в подходе к данным: *MLP* оперирует независимыми точками, в то время как *LSTM* требует и использует их временную структуру. Это делает обоснованный выбор между двумя архитектурами прямым следствием характера решаемой задачи.

Сравнение производительности и настройки гиперпараметров

После обучения моделей можно сравнить их производительность на тестовых данных. Обычно *LSTM* показывает лучшие результаты на данных с временными зависимостями, но требует больше вычислительных ресурсов. В то время как *MLP* может быть более эффективным для задач, где временные зависимости не столь важны. Для визуализации результатов были применены библиотеки, такие как *seaborn, shiny, plotly*, примеры применения которых были рассмотрены в опубликованной ранее работе [15]). Проведем сравнение подходов по применению *MLP* и *LSTM* (табл. 2).

Таблица 2. **Сравнение подходов по применению MLP и LSTM**

Table 2. **Comparison of approaches for using MLP and LSTM**

Характеристика	*MLP*	*LSTM*
Функции активации	*ReLU* (скрытые слои), линейная (выходной слой)	*Tanh*(*LSTM*-слой), линейная (выходной слой)
Количество нейронов	64 (1-й слой), 32 (2-й слой)	50 (*LSTM*-слой)
Количество слоев	3 (2 скрытых + 1 выходной)	2 (1 LSTM + 1 выходной)
Оптимизатор	*Adam*	*Adam*
MSE	45,23	32,15
Время обучения (сек)	15	45

В данной таблице представлен компромисс между точностью и вычислительной сложностью при выборе между *MLP* и *LSTM*. Следует отметить, что *LSTM* показала меньшую *MSE* (32,15) по сравнению с *MLP* (45,23) благодаря способности учитывать временные зависимости. *MLP* уступает в точности для временных рядов, но может быть эффективнее для статических данных. Если рассматривать архитектуру, то *LSTM* требует больше нейронов и времени на обучение из-за рекуррентных связей, а *MLP* проще в реализации и быстрее обучается. Поэтому можно предложить следующие рекомендации – *LSTM* использовать для прогнозирования временных рядов (например, объем производства, спрос); а *MLP* для задач с независимыми признаками (например, прогнозирование затрат на основе текущих параметров). Рассмотрим случай с одинаковыми гиперпараметрами *MLP* и *LSTM* (табл. 3).

Таблица 3. **Настройка гиперпараметров для MLP и LSTM**

Table 3. **Hyperparameter tuning for MLP and LSTM**

Гиперпараметр	*MLP*	*LSTM*
Количество слоев	3	3 (*LSTM+Dense*)
Нейроны в слоях	[64, 32, 1]	[64, 32, 1]
Функция активации	*ReLU*(скрытые), *Linear* (выход)	*Tanh (LSTM), Linear*(выход)
Оптимизатор	*Adam(lr=0,001)*	*Adam(lr=0,001)*
Количество эпох	50	50
Dropout	0,2	0,2
Размер выборки	24 месяца	24 месяца
Длина последовательности	-	3 (для *LSTM*)

Следует рассмотреть данную таблицу более подробно. Здесь для *LSTM* указано «3 (*LSTM + Dense*)», чтобы явно показать структуру: 1 *LSTM*-слой (64 нейрона); 1 *Dense*-слой (32 нейрона); 1 Выходной *Dense*-слой (1 нейрон). Для *MLP* просто «3», так как все слои однотипные (*Dense*).

Это стандартная практика обозначения. Разные активации обусловлены архитектурными особенностями. *LSTM* по своей природе требует *tanh/gate*-активаций для работы с памятью, тогда как *MLP* стандартно использует *ReLU* в скрытых слоях. Линейная функция активации на выходе оставлена одинаковой для обеих моделей, так как это выходной слой регрессии.

Данные различия представляют собой минимальной необходимостью для корректной работы архитектур. Если сделать идентичные активации и архитектуры, то *LSTM* с *ReLU* будет работать значительно хуже (проверено экспериментально), *MLP* с *tanh* может страдать от «затухающих градиентов».

Если рассмотреть альтернативный вариант (т. е. принудительно одинаковые активации), то полученные результаты будут некорректными: *MLP: MSE* ≈ 52,3 (хуже на 15 %); *LSTM: MSE* ≈ 35,1 (хуже на 9 %). Таким образом, был обоснован текущий вариант сравнения (с разными активациями), который позволяет получить лучшие результаты, сохраняя при этом сопоставимость основных гиперпараметров. Сравнительный анализ метрик по результатам обучения представлен в табл. 4.

Таблица 4. **Сравнительный анализ метрик по результатам обучения для MLP и LSTM**

Table4. **Comparative analysis of metrics based on training results for MLP and LSTM**

Метрика	*MLP*	*LSTM*	Разница
MSE (обучающая)	38,72	28,15	-27,3%
MSE(тестовая)	45,23	32,15	-28,9%
Время обучения (сек.)	12	48	+300%
Использование памяти (GB)	1,2	2,8	+133%

Здесь для *LSTM* можно отметить почти на 28 % меньшую *MSE* благодаря способности учитывать временные зависимости, разница особенно заметна на тестовых данных (45,23 против 32,15). Для *LSTM* требуется в 4 раза больше времени на обучение. Потребление памяти у *LSTM* в 2,3 раза выше. Для *LSTM* также можно отметить меньший разрыв между *train/test MSE* (28,15/32,15 против 38,72/45,23 у *MLP*). Это свидетельствует о лучшей обобщающей способности *LSTM* для временных данных.

При одинаковых гиперпараметрах *LSTM* показывает статистически значимое улучшение *MSE* (*p-value* < 0,05) для временных данных. *MLP* остается предпочтительным выбором для быстрого прототипирования, задач без временных зависимостей, ограниченных вычислительных ресурсов. Для продуктовых решений с временными данными *LSTM* оправдывает дополнительные вычислительные затраты более точными прогнозами. Следует отметить, что подходы для проектирования систем для *MLP* и *LSTM* являются достаточно схожими. Во-первых, это формирование структуры слоев, т. е. в обоих случаях используется последовательность слоев для поэтапного преобразования данных; во-вторых, схожесть используемых гиперпараметров для проведения обучения и оптимизации, а также обработки входных данных; в-третьих, для выходного слоя регрессии также используется идентичная конструкция.

Второй частью вопроса между подходами проектирования будут архитектурные сходства: схожесть принципов преобразования признаков, т. е. оба подхода для обучения моделей последовательно преобразуют входные данные через серию нелинейных преобразований; схожесть реализации механизма обратного распространения; подходы к инициализации весов. Данные сходства позволяют переключаться между архитектурами в рамках одного проекта. Поэтому, рассматривая в дальнейшем проектирование для *MLP*, мы тем самым можем перенести полученные разработки и на *LSTM*. Другими словами, подходы к проектированию для *MLP*, которые направлены на загрузку данных, их подготовку, обучение и сохранение моделей, создание пользовательских и программных интерфейсов, могут быть аналогичным образом (за исключением некоторых компонентов) спроецированы и для *LSTM*.

Сравнительный анализ прогнозов с экспериментальными данными

Для проведения объективного сравнительного анализа эффективности моделей *MLP* и *LSTM* были использованы производственные данные, содержащие ежемесячные показатели за 24 месяца. Тестовые данные включали 6 месяцев, не участвовавших в обучении моделей, что обеспечило достоверность оценки их обобщающей способности.

Таблица 5. **Сравнительные метрики качества прогнозирования объема производства**

Table 5. **Comparative metrics of production volume forecasting quality**

Параметр	MLP	LSTM	Улучшение
MSEPV	±45,23	±32,15	–28,90 %
Влияние *Emp*	23,40 %	18,10 %	–22,60 %
Влияние *DH*	31,20 %	25,70 %	–17,60 %
Влияние *MC*	19,80 %	15,30 %	–22,70 %
Корреляция прогноза с фактическими данными	0,86	0,94	9,30 %

Детальный анализ результатов позволяет определить статистически значимое превосходство *LSTM*-модели по всем ключевым метрикам. Снижение средней ошибки прогноза на 28,9 % (с 45,23 до 32,15 единиц) подтверждает способность *LSTM*-архитектуры более точно учитывать сложные временные зависимости в производственных данных. Особого внимания заслуживает анализ влияния отдельных факторов на качество прогноза. *LSTM*-модель продемонстрировала существенно лучшую чувствительность к изменению ключевых производственных параметров: влияние количества работников на ошибку прогноза снизилось на 22,6 %, времени простоя оборудования – на 17,6 %, а затрат на техническое обслуживание – на 22,7 %. Это свидетельствует о том, что *LSTM* не только точнее прогнозирует общий объем производства, но и более адекватно реагирует на изменения отдельных производственных факторов.

Высокая корреляция прогнозов *LSTM*-модели с экспериментальными данными (0,94 против 0,86 у *MLP*) подтверждает ее способность сохранять адекватность прогноза в условиях изменяющихся производственных условий. Наибольшее преимущество *LSTM* проявилось в периоды резких изменений производственных показателей, где модель демонстрировала на 35–40 % меньшую ошибку по сравнению с *MLP*.

Полученные результаты убедительно доказывают, что учет временных зависимостей в архитектуре *LSTM* позволяет не только существенно повысить общую точность прогнозов объемов производства, но и обеспечивает более сбалансированное и точное отражение влияния отдельных производственных факторов на конечный результат.

Заключение

В работе проведен анализ применения двух архитектур глубокого обучения: *MLP* и *LSTM* – на примере решения задачи регрессии для прогнозирования объемов производства. Рассмотрена реализация этапов разработки: подготовка данных, создание и обучение моделей, оценка их производительности.

В результате сравнительного анализа было установлено, что применение *LSTM* приводит к значительно лучшим результатам по точности, что подтверждается снижением ошибки *MSE* на тестовых данных примерно на 28–29 % по сравнению с *MLP* (например, *MSE LSTM* составила 32,15 против 45,23 у *MLP*). Это обусловлено фундаментальной способностью *LSTM* эффективно учитывать долгосрочные временные зависимости в данных, делая ее предпочтительным инструментом для задач, основанных на временных рядах, где последовательность и порядок данных критически важны для формирования точного прогноза. Однако это преимущество достигается ценой существенно более высоких вычислительных затрат: *LSTM* потребовала в 4 раза больше времени на обучение и в 2,3 раза больше оперативной памяти (2,8 ГБ против 1,2 ГБ у *MLP*).

MLP, в свою очередь, позволяет провести более быстрое обучение и простую реализацию, что делает ее удобным и эффективным решением для задач, не имеющих ярко выраженной временной структуры или в условиях ограниченных вычислительных возможностей. В ходе применения *MLP* был установлен меньший разрыв между *MSE* на обучающих и тестовых данных (38,72/45,23), что хотя и указывает на меньшую обобщающую способность по сравнению с *LSTM* для временных рядов (28,15/32,15), все же приемлемо для задач без выраженных временных зависимостей.

Таким образом, выбор между *MLP* и *LSTM* обосновывается исходя из конкретных характеристик исходных данных и жестких требований к точности и скорости обучения:

– если исходные данные обладают сильными временными зависимостями, где текущее значение тесно связано с предыдущими состояниями (например, динамика производства за месяцы) и требуется максимально высокая точность прогнозирования (цель – снижение MSE на 20-30% и более), при этом есть доступ к достаточным вычислительным ресурсам (готовность к увеличению времени обучения в 3-4 раза и потреблению памяти в 2-3 раза), то *LSTM* является предпочтительным выбором;

– если исходные данные являются статическими или их временные зависимости незначительны, а приоритетом являются быстрое прототипирование, низкое потребление вычислительных ресурсов (до 1,5 ГБ памяти) и приемлемая точность (*MSE* в диапазоне 40–50), то *MLP* будет более эффективным и практичным решением. *MLP* также подходит для задач, где важна простота реализации и интерпретации модели.

Предложенная практическая реализация может быть использована как основа для построения интеллектуальных информационно-аналитических систем в производственном секторе, позволяя гибко выбирать архитектуру в зависимости от специфики задачи и доступных ресурсов.

Библиографические ссылки

1. *Zhu M., Zhang G., Zhang L., Han W., Shi Zh., Lv X.* Object segmentation by spraying robot based on multi-layer perceptron // Energies. 2023. Vol. 16, no. 1. P. 232.

2. *Xu F.* Research on traffic flow prediction method based on LSTM model and PSO-LSTM model // Applied and Computational Engineering. 2024. Vol. 101, no. 1. P. 154-163.

3. *Oliveira D.D., Rampinelli M., Tozatto G.Z., Andreão R.V., Müller S.M.T.*Forecasting vehicular traffic flow using MLP and LSTM // Neural Computing & Applications. 2021. Vol. 33, no, 24. Pp. 17245-17256.

4. *Ahmed Sh., Khan Z.A., Mohsin S.M., Latif Sh., Aslam Sh., Mujlid H., Adil M., Najam Z.* Effective and efficient DDoS attack detection using deep learning algorithm, multi-layer perceptron // Future Internet. 2023. Vol. 15, no. 2. Pp. 76-85.

5. *Ahmed Sh.* A software framework for predicting the maize yield using modified multi-layer perceptron // Sustainability. 2023. Vol. 15, no. 4. P. 3017.

6. *Murugesan R., Mishra E., Krishnan A.H.* Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, BI-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM // International Journal of Sustainable Agricultural Management and Informatics. 2022. Vol. 8, no. 3. P. 242.

7. *Murugesan R., Mishra E., Krishnan A.H.*Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, BI-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM // International Journal of Sustainable Agricultural Management and Informatics. 2022. Vol. 8, no. 3. P. 242.

8. *Bayram F., Aupke Ph., Ahmed B.S., Kassler A., Theocharis A., Forsman J. DA*-LSTM: A dynamic drift-adaptive learning framework for interval load forecasting with LSTM networks // Engineering Applications of Artificial Intelligence. 2023. Vol. 123. P. 106480.

9. *Nurhidayat, Defit S., Sumijan.* Data mining dalamakurasitingkatkelayakanpakaiterhadapperalatanperangkatKeras // Jurnal Informasi dan Teknologi. 2020. Pp. 83-88.

10. *Jaiswal G.* Stock prediction model using TensorFlow // International Journal for Research in Applied Science and Engineering Technology. 2021. Vol. 9, no. 12. Pp. 99-103.

11. *Nunez-Yanez J., Otero A., de la Torre E.*Dynamically reconfigurable variable-precision sparse-dense matrix acceleration in TensorFlow lite // Microprocessors and Microsystems. 2023. Vol. 98. P. 104801.

12. *Zhu M., Min W., Li J., Liu M., Deng Z., Zhang Ya.* Constructing a smoothed leaky relu using a linear combination of the smoothed ReLu and identity function // Neural Computing & Applications. 2025. Vol. 37, no. 9. Pp. 6465-6478.

13. *Diederik P. Kingma, Jimmy Lei Ba.* Adam: a method for stochastic optimization // Published as a conference paper at ICLR 2015. Pp. 1-15.

14. *Koc I., Arslan E.* Dynamic ticket pricing of airlines using variant batch size interpretable multi-variable long short-term memory // Expert Systems with Applications. 2021. Vol. 175. P. 114794.

15. *Shalabh.* Interactive web-based data visualization with R, Plotly, and Shiny // Journal of the Royal Statistical Society: Series A (Statistics in Society). 2021. Vol. 184, no. 3. P. 1150.

References

1. *Zhu M., Zhang G., Zhang L., Han W., Shi Zh., Lv X.* Object segmentation by spraying robot based on multi-layer perceptron // Energies. 2023. Vol. 16, no. 1. P. 232.

2. *Xu F.* Research on traffic flow prediction method based on LSTM model and PSO-LSTM model // Applied and Computational Engineering. 2024. Vol. 101, no. 1. Pp. 154-163.

3. *Oliveira D.D., Rampinelli M., Tozatto G.Z., Andreão R.V., Müller S.M.T.*Forecasting vehicular traffic flow using MLP and LSTM // Neural Computing & Applications. 2021. Vol. 33, no. 24. Pp. 17245-17256.

4. *Ahmed Sh., Khan Z.A., Mohsin S.M., Latif Sh., Aslam Sh., Mujlid H., Adil M., Najam Z.* Effective and efficient DDoS attack detection using deep learning algorithm, multi-layer perceptron // Future Internet. 2023. Vol. 15, no. 2. Pp. 76-85.

5. *Ahmed Sh.* A software framework for predicting the maize yield using modified multi-layer perceptron // Sustainability. 2023. Vol. 15, no. 4. P. 3017.

6. *Murugesan R., Mishra E., Krishnan A.H.* Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, BI-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM // International Journal of Sustainable Agricultural Management and Informatics. 2022. Vol. 8, no. 3. P. 242.

7. *Murugesan R., Mishra E., Krishnan A.H.*Forecasting agricultural commodities prices using deep learning-based models: basic LSTM, BI-LSTM, stacked LSTM, CNN LSTM, and convolutional LSTM // International Journal of Sustainable Agricultural Management and Informatics. 2022. Vol. 8, no. 3. P. 242.

8. *Bayram F., Aupke Ph., Ahmed B.S., Kassler A., Theocharis A., Forsman J. DA*-LSTM: A dynamic drift-adaptive learning framework for interval load forecasting with LSTM networks // Engineering Applications of Artificial Intelligence. 2023. Vol. 123. P. 106480.

9. *Nurhidayat, Defit S., Sumijan.* Data mining dalam akurasi tingkat kelayakan pakai terhadap peralatan perangkat keras // JurnalInformasi dan Teknologi. 2020. Pp. 83-88.

10. *Jaiswal G.*Stock prediction model using TensorFlow // International Journal for Research in Applied Science and Engineering Technology. 2021. Vol. 9, no. 12. Pp. 99-103.

11. *Nunez-Yanez J., Otero A., de la Torre E.*Dynamically reconfigurable variable-precision sparse-dense matrix acceleration in TensorFlow lite // Microprocessors and Microsystems. 2023. Vol. 98. P. 104801.

12. *Zhu M., Min W., Li J., Liu M., Deng Z., Zhang Ya.* Constructing a smoothed leaky relu using a linear combination of the smoothed ReLu and identity function // Neural Computing & Applications. 2025. Vol. 37, no. 9. Pp. 6465-6478.

13. *Diederik P. Kingma, Jimmy Lei Ba.* Adam: a method for stochastic optimization // Published as a conference paper at ICLR 2015. Pp. 1-15.

14. *Koc I., Arslan E.* Dynamic ticket pricing of airlines using variant batch size interpretable multi-variable long short-term memory // Expert Systems with Applications. 2021. Vol. 175. P. 114794.

15. *Shalabh.*Interactive web-based data visualization with R, Plotly, and Shiny // Journal of the Royal Statistical Society: Series A (Statistics in Society). 2021. Vol. 184, no. 3. P. 1150.

* * *

Forecasting Production Performance Using MLP and LSTM: Analysis of Design Approaches

D. S. Ponomarev, PhD in Engineering, Kalashnikov Izhevsk State Technical University; Federal State Institution Research Institute of the Federal Penitentiary Service Russia, Izhevsk, Russia

This article examines the problem of production indicator forecasting using deep learning methods based on fully connected and recurrent neural networks. It focuses on the practical implementation of regression models by means of a multilayer perceptron (MLP) and a recurrent neural network with long short-term memory (LSTM), as well as on the analysis of their design approaches taking into account the characteristics of the source data. The work covers the full cycle of model construction: data collection and preprocessing, feature normalization, generating training and test samples, constructing neural network architectures, hyperparametertuning, training process, forecast qualityassessment, and resultvisualization.

As an applied example, the problem of production volume forecasting based on real historical data and characterizing the performance of production facilities over a multi-year period is considered. For MLP models, the data is treated as independent observations, while for LSTM, fixed-length time sequences are generated, allowing for the dynamics and long-termindicator dependencies to be taken into account. A comparative analysis of architectural solutions, computational complexity, resource requirements, and forecasting accuracy is conducted. It is shown that LSTM demonstrates higher accuracy when working with time series due to its ability to account for temporal context, but requires significantly more computational effort. At the same time, MLP is characterized by its ease of implementation and faster training speed, making it suitable for tasks with poorly defined temporal data structure. The results of the study can be used in the design of intelligent information and analytical systems for the manufacturing sector and serve as a practical guide for choosing a neural network architecture with respect to the nature of the data and accuracy and resource requirements.

Keywords: multilayer perceptron, recurrent networks, information system, production, design, system analysis, statistics.

Получено: 04.08.25

Образец цитирования

Д. С. Пономарёв Прогнозирование производственных показателей с использованием *MLP* и *LSTM*: анализ подходов проектирования // Интеллектуальные системы в производстве. 2025. Т. 23, № 4. С. 79–85. DOI: 10.22213/2410-9304-2025-4-79-85.

For Citation

Ponomarev D.S. [Forecasting production performance using MLP and LSTM: analysis of design approaches]. *Intellektual'nye sistemy v proizvodstve.* 2025, vol. 23, no. 4, pp. 79-85. DOI: 10.22213/2410-9304-2025-4-79-85.