Научная статья
УДК 004.032.26
DOI: https://doi.org/10.18127/j19998554-202405-09

Сравнительное исследование эффективности автоэнкодеров в задачах обнаружения аномалий

В.Е. Марлей[1], **А.Н. Терехов**[2], **Ю.А. Гатчин**[3], **В.И. Милушков**[4], **Н.Н. Лиманский**[5]

[1,4,5] Государственный университет морского
и речного флота имени адмирала С.О. Макарова (Санкт-Петербург, Россия)
[2] Санкт-Петербургский государственный университет (Санкт-Петербург, Россия)
[3] Национальный исследовательский университет ИТМО (Санкт-Петербург, Россия)
[1] vmarley@mail.ru, [2] ant@tercom.ru, [3] gatchin1952@mail.ru, [4] info@sohoware.ru, [5] info@sohoware.ru

Аннотация

Постановка проблемы. Автокодировщики представляют собой мощные инструменты для решения задач обнаружения аномалий благодаря их способностям обучаться сжатию и восстановлению нормальных данных. Основная идея использования автокодировщиков заключается в создании моделей, способных эффективно обрабатывать нормальные данные. Однако при этом возникает сложность при восстановлении аномальных показателей, что приводит к увеличению ошибок реконструкции. Основными типами автоэнкодеров являются: сверточный (CAE), вариационный (VAE) и состязательный (AAE).

Цель. Оценить производительность различных моделей автоэнкодеров в задаче обнаружения аномалий на наборе данных MNIST, а также выявить их преимущества и ограничения.

Результаты. Показано, что все три модели автоэнкодеров имеют высокие показатели при обнаружении аномалий, однако наблюдается снижение производительности и вычислительных затрат. Отмечено, что CAE показал лучшие результаты по скорости, но в некоторых случаях уступил VAE и AAE по точности. Выявлено, что, несмотря на более сложную архитектуру и увеличенное время обучения, VAE и AAE продемонстрировали лишь незначительное улучшение по сравнению с CAE.

Практическая значимость. Простота и скорость CAE могут оказаться более предпочтительными для ряда задач, в то время как VAE и AAE могут быть полезны в случаях, где критичны дополнительные возможности, такие как генерация новых данных или улучшенная устойчивость к шумам. Различия в результатах реконструкции моделей автоэнкодеров и их интерпретации открывают новые возможности для дальнейших исследований, в том числе для разработки гибридных подходов, которые могут сочетать сильные стороны этих моделей.

Ключевые слова

Обнаружение аномалий, автоэнкодеры, сверточный автоэнкодер, вариационный автоэнкодер, состязательный автоэнкодер, MNIST

Для цитирования

Марлей В.Е., Терехов А.Н., Гатчин Ю.А., Милушков В.И., Лиманский Н.Н. Сравнительное исследование эффективности автоэнкодеров в задачах обнаружения аномалий // Нейрокомпьютеры: разработка, применение. 2024. Т. 26. № 5. С. 96–106. DOI: https://doi.org/10.18127/j19998554-202405-09

A brief version in English is given at the end of the article

Введение

Автокодировщики представляют собой мощные инструменты для решения задач обнаружения аномалий благодаря их способностям обучаться сжатию и восстановлению нормальных данных [1]. Основная идея использования автокодировщиков заключается в создании моделей, способных эффективно обрабатывать нормальные данные. Однако при этом возникает сложность при восстановлении аномальных показателей, что приводит к увеличению ошибок реконструкции. Такой подход особенно применяется в условиях, когда аномалии представляют собой редкие или нетипичные события, которые значительно отличаются от основной массы данных [2].

Примеры использования автокодировщиков для обнаружения аномалий можно встретить в различных областях. В финансовой сфере их используют для отслеживания мошеннических транзакций, когда необычные шаблоны могут проявлять подозрительную активность [3]. В промышленности автокодировщики помогают обнаруживать дефекты на производственных линиях, анализируя изображения или сигналы с датчиков и выявляя отклонения от нормальной работы оборудования [4–6]. В медицинской диагностике автокодировщики используются для обнаружения патологий в изображениях, таких как

рентгеновские снимки или МРТ, где аномальные образования могут значительно отличаться от здоровых тканей.

Рассмотрим *три различных варианта построения автоэнкодеров*:

1) сверточный автоэнкодер (CAE) [7];

2) вариационный автоэнкодер (VAE) [8];

3) состязательный автоэнкодер (AAE) [3].

Эти модели различаются не только по архитектуре, но и по методам регуляризации скрытого пространства, что влияет на их способность обнаруживать аномалии и вычислительную эффективность.

Ц е л ь р а б о т ы – оценить производительность различных моделей автоэнкодеров в задаче обнаружения аномалий на наборе данных MNIST, а также выявить их преимущества и ограничения.

Автокодировщики для обнаружения аномалий

Автокодировщики являются эффективным средством для решения задач по обнаружению аномалий, поскольку они обучаются сжатию и восстановлению нормальных данных. Архитектура автокодировщика включает два важных компонента: *кодировщик* и *декодировщик* [2].

Кодировщик берет входной образец

$$x = \mathbb{R}^{C \times H \times W}$$

и преобразует его в скрытое представление

$$z = \mathbb{R}^{C_b \times H_b \times W_b},$$

где C – число каналов; H – высота; W – ширина изображения.

Данное скрытое представление имеет меньшие размеры по сравнению с исходными данными, что представляет собой так называемое «бутылочное горлышко». Задача такого сокращения – уменьшить объем входных данных до меньшего размера, сохраняя при этом наиболее важную информацию, необходимую для последующего восстановления информации декодировщиком. Процесс сжатия и создания «бутылочного горлышка» заставляет модель фокусироваться на существенных характеристиках входного изображения, устраняя менее значимые детали. В результате автокодировщик вынужден обучаться выявлению ключевых особенностей, которые необходимы для реконструкции.

После того как данные прошли через кодировщик и были преобразованы в компактное скрытое представление, декодировщик вступает в игру, выполняя обратный процесс. Основная задача декодировщика – восстановить исходные данные из сжатого представления. Это требует от модели способности точно восстановить те ключевые особенности, которые были выделены кодировщиком и сохранены в ограниченном пространстве скрытых переменных.

Декодировщик обычно представляет собой серию слоев, которые постепенно восстанавливают размер и структуру данных, приближая их к исходным характеристикам.

На рис. 1 представлен пример работы автоэнкодера.

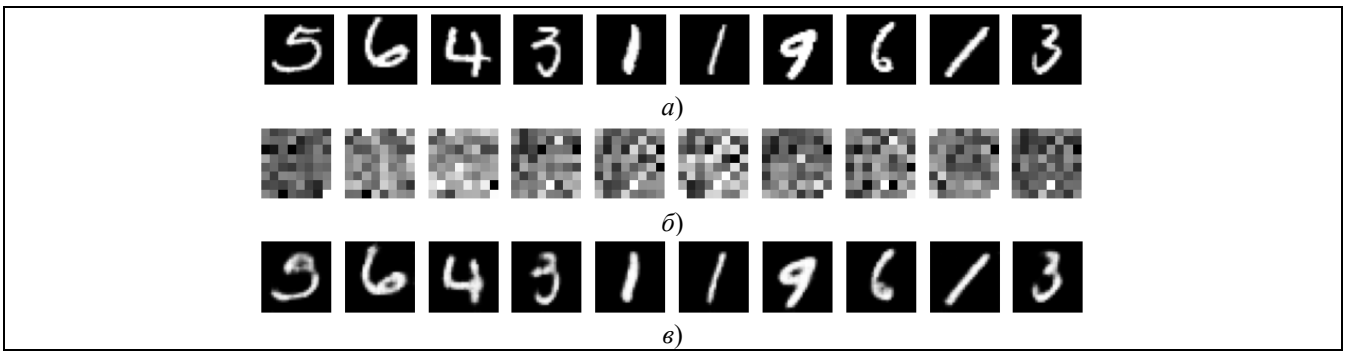

Рис. 1. Пример работы автоэнкодера: *а* – изображения, подаваемые на вход энкодера; *б* – соответствующие латентные представления; *в* – декодированные изображения, полученные из латентных представлений

Fig. 1. An example of the operation of an autoencoder: *a* – images supplied to the encoder input; *b* – corresponding latent representations; *c* – decoded images obtained from latent representations

Если автокодировщик обучен на изображениях, считающихся нормальными, то он будет способен эффективно сжимать и восстанавливать эти нормальные образцы. Например, если автокодировщик обучен только на изображениях цифры «5», то он научится сжимать и восстанавливать именно эти изображения (рис. 2).

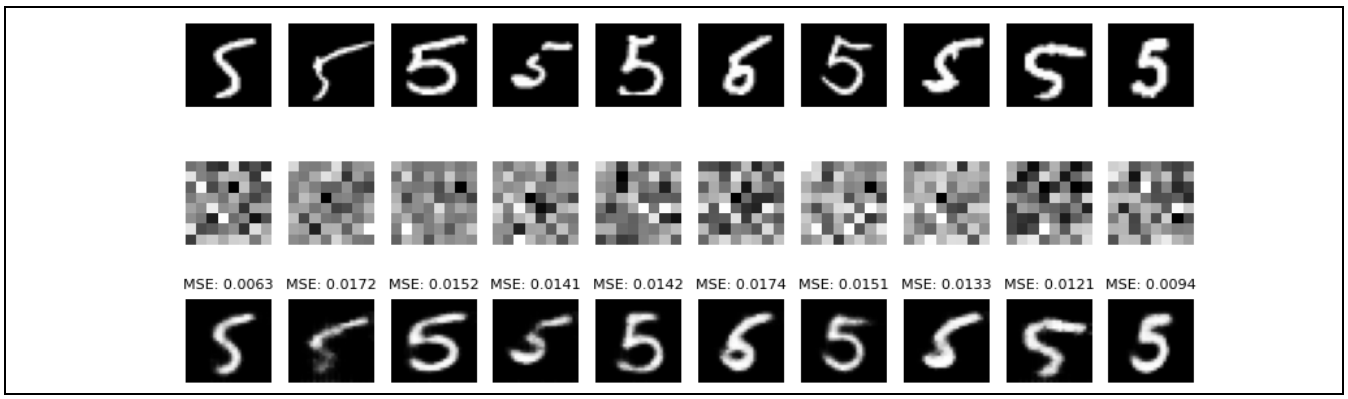

Рис. 2. Пример работы автоэнкодера, обученного на изображениях цифры «5»

Fig. 2. An example of the operation of an autoencoder trained on the images of the digit "5"

Если на вход подается аномальное изображение, т.е. любое изображение, не являющееся цифрой «5», то модель не сможет его точно восстановить, поскольку аномалии не были представлены в обучающих данных. Это ведет к большей ошибке восстановления для аномальных данных (рис. 3).

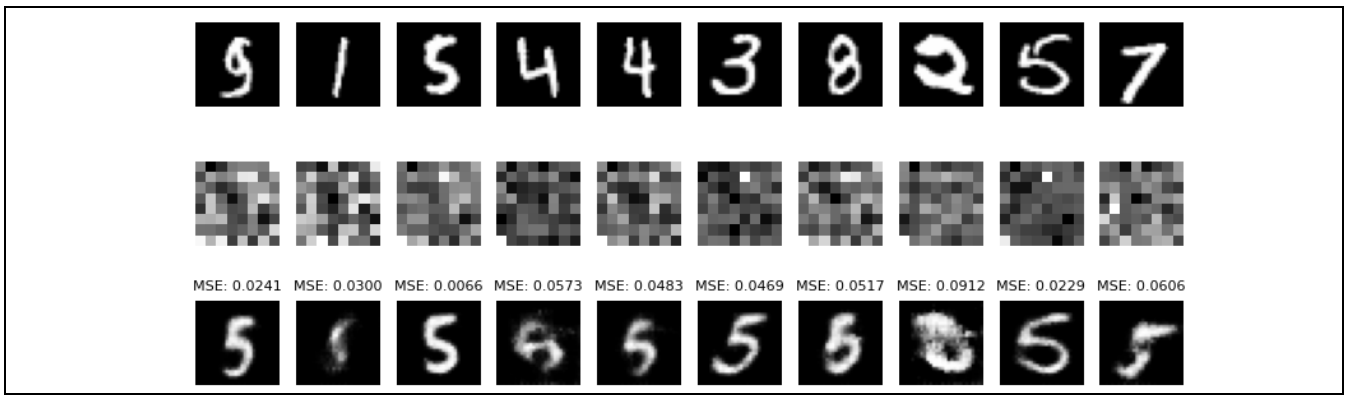

Рис. 3. Пример работы автоэнкодера, обученного на изображениях цифры «5», не способного восстанавливать изображения других цифр

Fig. 3. An example of the operation of an autoencoder trained on images of the digit "5", which is not able to restore images of other digits

Ошибка восстановления $R(x,\tilde{x}) = \mathbb{R}^{C\times H\times W}$ определяется как среднеквадратичная ошибка (MSE):

$$R(x,\tilde{x}) = \frac{1}{C\times H\times W}\sum_{i=1}^{C}\sum_{j=1}^{H}\sum_{k=1}^{W}\left(x_{ijk}-\tilde{x}_{ijk}\right)^2, \quad (1)$$

где x – исходное изображение; $\tilde{x}$ – восстановленное изображение.

Нормальное изображение приводит к меньшей ошибке восстановления, поскольку модель обучена на подобных данных, в то время как аномальное изображение, будучи непривычным для модели, вызывает большую ошибку восстановления. Следовательно, значение $R(x,\tilde{x})$ можно использовать для различения нормальных и аномальных образцов [9, 10].

«Стандартный» сверточный автокодировщик

«Стандартный» сверточный автокодировщик состоит из двух основных компонентов: *сверточного кодировщика* ECAE и *сверточного декодировщика* DCAE. Кодировщик выполняет сжатие входных дан-

ных x до латентного представления z, после чего декодировщик восстанавливает z обратно до размеров исходных данных

$$\tilde{x} = D_{\text{CAE}}\left(E_{\text{CAE}}\left(x\right)\right) = D_{\text{CAE}}\left(z\right). \tag{2}$$

Параметры CAE оптимизируются путем минимизации функции потерь, основанной на среднеквадратичной ошибке. Функция MSE определяется следующим образом:

$$L_{\text{MSE}}\left(x,\tilde{x}\right) = \frac{1}{H}\frac{1}{W}\sum_{m=1}^{H}\sum_{n=1}^{W}\left(x\left(m,n\right)-\tilde{x}\left(m,n\right)\right)^2, \tag{3}$$

где $x\left(m,n\right)$ и $\tilde{x}\left(m,n\right)$ – интенсивность пикселей на координатах $\left(m,n\right)$ для исходного и восстановленного изображений соответственно, при этом $m \in \left[1,...,H\right]$ и $n \in \left[1,...,W\right]$.

Вариационный автокодировщик

Вариационный автокодировщик имеет структуру, схожую с классическим сверточным автокодировщиком. Он включает в себя кодировщик EVAE и декодировщик DVAE. Кодировщик сжимает входные данные до латентного представления меньшей размерности, а декодировщик восстанавливает их обратно до исходных размеров.

Однако вместо фиксированного латентного представления, как в обычных автокодировщиках, VAE стремится моделировать входные данные как гауссовское распределение с параметрами μ (среднее значение) и σ (дисперсия). Кодировщик VAE берет входное изображение x и производит два выхода: вектор средних значений μ и вектор логарифмов дисперсий $\ln(\sigma^2)$. На основе этих параметров создается латентное представление z следующим образом:

$$z = \mu + \sigma\varepsilon, \tag{4}$$

где ε – случайная величина, взятая из стандартного нормального распределения $N(0,1)$.

Это позволяет VAE обучаться генерировать латентные представления, которые следуют априорному распределению $p\theta(z)$ [8, 11].

Истинное апостериорное распределение $p\theta(z|x)$ для непрерывного латентного пространства невозможно вычислить точно, поэтому используется аппроксимация $q\Phi(z|x)$. Вариационный вывод заключается в определении параметризованного семейства распределений и выборе распределения, которое минимизирует ошибку аппроксимации. Это достигается путем оптимизации функции потерь, включающей два компонента:

1) *ошибку восстановления*, которая показывает, насколько хорошо восстановленное изображение $\tilde{x}$ соответствует исходному изображению x;

2) *дивергенцию Кульбака-Лейблера* (KL-дивергенция), которая измеряет разницу между аппроксимированным апостериорным распределением $q\Phi(z|x)$ и априорным распределением $p\theta(z)$ [12].

Функция потерь VAE может быть записана следующим образом:

$$L_{\text{VAE}}\left(x,\tilde{x}\right) = L_{\text{MSE}}\left(x,\tilde{x}\right) + D_{\text{KL}}\left(q\Phi\left(z \mid x\right) \| \, p\theta\left(z\right)\right). \tag{5}$$

Использование VAE позволяет моделировать более сложные распределения данных и генерировать новые образцы, сходные с обучающими данными.

Состязательный автокодировщик

Состязательный автокодировщик функционирует в вероятностной манере и обладает структурой, схожей с вариационным автокодировщиком. Оба метода накладывают априорное распределение на латентное представление z, однако они используют разные подходы для согласования этого представления с априорным распределением. В VAE для этого применяется KL-дивергенция, которая минимизирует

расхождение между апостериорным и априорным распределениями. В отличие от этого, AAE использует концепцию генеративных состязательных сетей (GAN).

В AAE кодировщик и декодировщик обучаются аналогично стандартному CAE, минимизируя среднеквадратичную ошибку для обеспечения высокого качества восстановления данных. Однако, в отличие от VAE, AAE включает дополнительный компонент GAN, состоящий из дискриминативной сети. Эта сеть помогает регулировать латентный вектор, обучая кодировщик генерировать латентные представления, которые дискриминативная сеть не может отличить от настоящих априорных распределений.

Обучение AAE происходит в двух фазах. В *первой фазе* (восстановления) кодировщик EAAE и декодировщик DAAE минимизируют функцию потерь MSE, как в стандартном CAE. Во *второй фазе* (регуляризации) кодировщик работает как генератор в состязательной сети, стремясь «обмануть» дискриминатор AAE, заставляя его думать, что сгенерированные латентные векторы произошли из истинного априорного распределения. Дискриминативная сеть, в свою очередь, обучается определять, являются ли латентные векторы «реальными» или «поддельными» с использованием расстояния Васерштейна. Ограничения Липшица поддерживаются путем обрезки весов до фиксированного диапазона при каждом обновлении весов. Таким образом, AAE сочетает преимущества автокодировщиков и GAN [3, 13].

Материалы и методы исследования

Архитектура кодировщика и декодировщика в моделях CAE, VAE и AAE реализована одинаково. В случае VAE бутылочное горлышко состоит из двух внутренних слоев: первый отвечает за латентный вектор средних значений, а второй – за латентный вектор стандартных отклонений. Все типы автоэнкодеров с аналогичными архитектурами обучались при одинаковых условиях, включая количество эпох, скорость обучения и размер минипакета для каждого эксперимента. Для оптимизации параметров автоэнкодера применялся алгоритм Adam (Adaptive Moment Estimation), который реализуется с помощью torch.optim.Adam. Кодировщики и декодировщики автоэнкодеров реализованы одинаково, но вычисления латентного вектора в бутылочном горлышке различаются.

В данном исследовании используется датасет MNIST [9], состоящий из изображений рукописных цифр, для оценки качества обнаружения аномалий с помощью трех типов автоэнкодеров: CAE, VAE и AAE. В качестве нормальных данных выбраны изображения цифры «5», тогда как все остальные цифры из набора данных рассматриваются как аномальные. Такая выборка позволяет провести четкую дифференциацию между нормальными и аномальными данными. Обучение всех трех моделей проводится на изображениях цифры «5», что позволяет автоэнкодерам научиться реконструировать нормальные данные, а затем использовать их способности для выявления отклонений при обработке аномальных изображений.

Архитектуры сетей

Рассмотрим архитектуры CAE, VAE и AAE, предназначенные для обработки изображений размером 28×28 с одним каналом.

Архитектура CAE включает в себя кодер, состоящий из следующих слоев: первый сверточный слой имеет 32 фильтра размером 3×3 с активацией ReLU и padding='same', за которым следует MaxPooling слой размером 2×2. Второй сверточный слой содержит 64 фильтра размером 3×3 с теми же параметрами активации и пулинга. Далее следуют два сверточных слоя по 128 фильтров и один слой с 256 фильтрами, все с размером 3×3 и активацией ReLU. Затем выходы этих слоев проходят через слой выравнивания (Flatten) и полносвязный слой с 100 нейронами, преобразуясь в латентное пространство размерностью 20.

Декодер зеркально повторяет структуру кодера, начиная с полносвязного слоя, и последовательно восстанавливает изображение с помощью слоев, аналогичных используемым в кодере, но с применением операций транспонированной свертки и апсемплинга, завершаясь слоем с одним каналом, использующим активацию сигмоиды.

Архитектура VAE построена на аналогичной основе, но с добавлением вероятностного компонента в латентное пространство. В кодере используются слои с теми же параметрами: начальный сверточный слой с 32 фильтрами размером 3×3, за которым следуют MaxPooling слой размером 2×2, затем сверточные слои с 64 фильтрами и двумя слоями по 128 фильтров и завершающий слой с 256 фильтрами. После

этапа выравнивания (Flatten) выходы подаются на два полносвязных слоя: один для вычисления среднего (μ) и другой для логарифма дисперсии ($\ln\sigma$), каждый из которых содержит 20 нейронов. Латентное пространство формируется путем репараметризации, где используется специальная функция sampling, чтобы учесть стохастичность.

Декодер в VAE аналогичен декодеру CAE, начиная с полносвязного слоя со 100 нейронами и постепенно восстанавливая изображение через транспонированные сверточные слои с теми же параметрами (128 и 64 фильтра) и операции апсемплинга. Дополнительно в VAE применяется кастомный слой KL-дивергенции, который минимизирует расхождение между латентным распределением и стандартным нормальным распределением, обеспечивая более надежное обучение модели.

Архитектура AAE использует принципиально иную структуру, в которой латентное пространство кодируется аналогично VAE, но затем проходит через дискриминатор, как в генеративно-состязательных сетях. Кодер AAE имеет ту же структуру, что и в CAE и VAE: первый сверточный слой с 32 фильтрами (3×3), MaxPooling слой (2×2), затем сверточные слои с 64 фильтрами, два слоя по 128 фильтров и один слой с 256 фильтрами. После слоя выравнивания (Flatten) и полносвязного слоя со 100 нейронами данные кодируются в латентное пространство размерностью 20.

Декодер AAE идентичен декодеру CAE и VAE, начиная с полносвязного слоя со 100 нейронами и восстанавливая изображение через транспонированные сверточные слои с 128 и 64 фильтрами и операции апсемплинга. Однако ключевым отличием AAE является наличие дискриминатора, который получает латентные представления и обучается отличать их от шума, сгенерированного из заранее заданного нормального распределения. Дискриминатор включает в себя несколько полносвязных слоев: первый слой с 128 нейронами, затем слой с 64 нейронами и завершается слоем с одним нейроном с активацией сигмоиды, который выдает вероятность того, что входное латентное представление принадлежит исходному распределению. Дискриминатор и кодер обучаются в соревновательной манере: кодер пытается «обмануть» дискриминатор, создавая латентные представления, которые невозможно отличить от выборок из заданного распределения.

Метрики оценки

Для оценки качества решений по задаче классификации данных как «нормальные» или «аномальные» применялись *две основные метрики*:

1) *кривые характеристик приемника-оператора* (ROC, Receiver Operating Characteristic);

2) *кривые точности-полноты* (PR, Precision-Recall).

Эти метрики позволяют наглядно оценить способность модели различать нормальные и аномальные данные в контексте задачи обнаружения аномалий.

Кривая ROC представляет зависимость между чувствительностью (True Positive Rate, TPR) и специфичностью (False Positive Rate, FPR) модели при различных пороговых значениях, используемых для определения того, является ли пример аномальным или нормальным. Каждая точка на кривой ROC соответствует определенному пороговому значению ошибки восстановления. Модель, способная хорошо разделять нормальные и аномальные данные, будет иметь кривую ROC, близкую к левому верхнему углу графика. Для количественной оценки качества модели используется площадь под кривой ROC (AUC-ROC). Значение AUC-ROC варьируется от 0 до 1, где 1 соответствует идеальной модели, а 0,5 указывает на модель, случайным образом классифицирующую данные [14].

Кривая PR отражает зависимость между точностью (Precision) и полнотой (Recall) при различных пороговых значениях. Эта метрика особенно полезна в случаях, когда классы несбалансированы, например, когда аномалии составляют лишь небольшую часть от общего объема данных. Высокая точность означает, что модель минимизирует количество ложных срабатываний (False Positives), а высокая полнота показывает, что модель способна обнаружить большинство аномалий (True Positives). Как и в случае с ROC, для оценки модели используется площадь под кривой PR (AUC-PR). Значение AUC-PR также варьируется от 0 до 1, где 1 соответствует модели с идеальной точностью и полнотой [15].

Обе кривые строятся путем изменения порогового значения, применяемого к ошибке восстановления, и позволяют сравнивать различные модели или подходы к обнаружению аномалий. В частности, для задачи классификации данных, как «нормальные» или «аномальные», высокое значение AUC-ROC и AUC-PR указывает на то, что модель эффективно разделяет нормальные и аномальные данные. Это яв-

ляется ключевым критерием успешности в задачах обнаружения аномалий. Данные метрики обеспечивают комплексную оценку, где AUC-ROC дает представление о балансе между чувствительностью и специфичностью, а AUC-PR акцентирует внимание на реальных значениях точности и полноты в условиях несбалансированных данных.

Результаты исследования

Все эксперименты и результаты были проведены и получены с использованием набора данных MNIST, который содержит изображения рукописных цифр, распределенных по десяти классам. Производительность моделей оценивалась на задаче обнаружения аномалий, где в качестве аномальных данных использовались цифры (0, 1, 2, 3, 4, 6, 7, 8, 9), которых не было в тренировочном наборе для конкретной модели, а в качестве нормальных данных принята цифра «5». Результаты представлены в виде численных метрик, таких как площадь под кривыми ROC и PR (AUC-ROC и AUC-PR).

В ходе экспериментов, направленных на сравнение трех типов автоэнкодеров, *сверточный автоэнкодер* показал высокие результаты в задаче обнаружения аномалий. В частности, CAE достиг значений ROC-AUC 0,9999 и PR-AUC 0,9999 для класса цифры «0», что свидетельствует о практически идеальной точности в идентификации аномалий для данного класса. Для класса цифры «2» CAE продемонстрировал результаты ROC-AUC 0,9943 и PR-AUC 0,9957, что также является весьма высоким показателем. Однако для класса цифры «6» модель показала значительное снижение производительности, достигнув ROC-AUC 0,9671 и PR-AUC 0,9774. Для других классов, таких как цифры «1», «3», «4», «6», «7», «8» и «9», CAE показал стабильные и высокие результаты с ROC-AUC и PR-AUC, колеблющимися в пределах от 0,990 до 0,999. Время обучения для CAE составило в среднем 344,67 с, что делает его относительно быстрым инструментом, хотя иногда менее точным в сложных случаях.

Вариационный автоэнкодер продемонстрировал несколько лучшие результаты по сравнению с CAE, особенно в более сложных задачах. Например, для класса цифры «0» VAE показал аналогичные результаты с ROC-AUC 0,9999 и PR-AUC 0,9991, а для класса цифры «2» он продемонстрировал еще более высокие показатели – ROC-AUC 0,9999 и PR-AUC 0,9998. VAE особенно выделился при работе с классом цифры «6», где он достиг ROC-AUC 0,9969 и PR-AUC 0,9951, что значительно превосходит результаты CAE для этого класса. Для остальных классов «1», «3», «4», «6», «7», «8» и «9» VAE также показал высокие и стабильные результаты, подтвержденные значениями ROC-AUC и PR-AUC в пределах от 0,990 до 0,999. Однако время обучения для VAE оказалось несколько больше – в среднем 386,24 с, что указывает на более высокие вычислительные затраты по сравнению с CAE.

Состязательный автоэнкодер также продемонстрировал высокие результаты в задаче обнаружения аномалий, особенно в сложных случаях. Для класса цифры «0» AAE показал идентичные значения ROC-AUC 0,9999 и PR-AUC 0,9999, как и CAE. Для класса цифры «2» результаты были также высокими – ROC-AUC 0,9977 и PR-AUC 0,9976. При работе с классом цифры «6» AAE достиг ROC-AUC 0,9951 и PR-AUC 0,9939, что, хотя и уступает VAE, всё же является высоким показателем. Для других классов «1», «3», «4», «6», «7», «8» и «9» AAE продемонстрировал стабильные результаты, схожие с результатами CAE и VAE, с ROC-AUC и PR-AUC, находящимися в диапазоне от 0,990 до 0,999. Однако AAE показал самое долгое время обучения среди всех моделей – 624,12 с, что делает его самым ресурсоёмким инструментом из представленных, несмотря на его высокую точность.

Заключение

Приведено сравнение трех типов сверточных автоэнкодеров: стандартного, вариационного и состязательного на известном наборе данных MNIST. Основное внимание было уделено способности этих моделей к обнаружению аномальных классов, где средняя ошибка реконструкции выступала в роли «оценки аномалии». Чем выше эта ошибка для определенного изображения, тем больше вероятность, что оно принадлежит к аномальному классу. Таким образом, посредством пороговой оценки средней ошибки реконструкции удалось классифицировать изображения как нормальные или аномальные.

Результаты, полученные с использованием всех трех типов автоэнкодеров, продемонстрировали высокую точность в задаче обнаружения аномальных классов для набора данных MNIST. Однако между моделями наблюдаются значительные различия как в плане сложности реализации, так и по времен-

ным затратам на обучение. CAE требует самой простой реализации и наименьшего времени обучения, что делает его привлекательным для задач, требующих быстрого прототипирования и вычислительной эффективности. Обучение VAE требует большего времени из-за дополнительных вычислительных затрат на шаг репараметризации и использования двух параметрических функций (μ и σ) для моделирования латентного пространства. AAE, будучи самой сложной моделью, требует еще большего времени на обучение, поскольку дополнительно включает обучение дискриминатора для состязательной регуляризации.

Несмотря на усложненную архитектуру и более длительное время обучения, модели AAE и VAE продемонстрировали лишь незначительное улучшение результатов по сравнению с CAE, и это улучшение было заметно лишь для некоторых классов MNIST. В некоторых случаях прирост производительности в задаче обнаружения аномалий был незначительным или отсутствовал вовсе. Тем не менее, результаты демонстрируют, что, хотя все три типа автоэнкодеров основаны на одном и том же принципе кодирования и декодирования информации, их базовые концепции и архитектурные подходы существенно различаются, что отражается в различиях в результатах реконструкции и их интерпретации.

Эти различия открывают новые возможности для дальнейших исследований, в том числе для разработки гибридных подходов, которые могут сочетать сильные стороны различных моделей. Однако при выборе модели для конкретного приложения важно учитывать, оправдано ли использование более сложных архитектур, если они не дают значительного прироста производительности. Простота и скорость CAE могут оказаться более предпочтительными для ряда задач, в то время как VAE и AAE могут быть полезны в случаях, где критичны дополнительные возможности, такие как генерация новых данных или улучшенная устойчивость к шумам. Таким образом, выбор подходящей модели должен основываться на конкретных требованиях задачи и доступных ресурсах.

Список источников

1. *Ваняшкин Ю.Ю., Макаров Д.А., Попова И.А., Соболева Е.Д.* Применение автокодировщиков для устранения шумов с изображений // StudNet. 2020. Т. 3. № 10. С. 27.
2. *Лазарев А.С., Туровский Ф.А., Пивоваров С.А.* Автокодировщик // Сборник статей Междунар. науч.-практич. конф. «Инновационные технологии в науке нового времени». Уфа: Аэтерна. 2017. Т. 3. С. 82–84.
3. *Калинин М.О.* Предотвращение угроз кибербезопасности в самоорганизующихся m2m-средах в условиях недостатка обучающих наборов данных // Цифровая экономика и Индустрия 5.0: развитие в новой реальности. 2022. С. 107–128. DOI 10.18720/IEP/2022.3/4.
4. *Милушков В.И., Лиманский Н.Н., Марлей В.Е.* Интеграция гибридного полуконтролируемого и контрастного обучения для автоматической классификации дефектов в производственных данных: повышение точности контроля качества продукции // Перспективы науки. 2024. № 6(177). С. 81–86.
5. *Лиманский Н.Н., Милушков В.И., Марлей В.Е.* Обнаружение аномалий в роботизированных системах: сравнительный анализ ConvLSTM с механизмом внимания и традиционных подходов // Перспективы науки. 2024. № 6(177). С. 77–80.
6. *Кураедов В.И.* Применение методов машинного обучения для оптимизации процесса генерации тестовых последовательностей при проектировании интегральных схем // Нейрокомпьютеры: разработка, применение. 2024. Т. 26. № 1. С. 14-22. DOI 10.18127/j19998554-202401-02.
7. *Волков А.К., Миронова Л.В., Потапова С.Е.* Применение предварительно обученных нейронных сетей для решения задачи обратного поиска рентгеновских изображений запрещенных предметов и веществ // Научный вестник Московского государственного технического университета гражданской авиации. 2024. Т. 27. № 2. С. 8–24. DOI 10.26467/2079-0619-2024-27-2-8-24.
8. *Беляков А.Н., Жуков В.П., Широков М.О.* Повышение качества распознавания образов с помощью модифицированного вариационного автоэнкодера // Материалы Междунар. науч.-технич. конф. «Состояние и перспективы развития электро- и теплотехнологии». Иваново: Ивановский государственный энергетический университет им. В.И. Ленина. 2023. Т. 2. С. 362–365.
9. *LeCun Y., Cortes C., Burges C.J.C.* MNIST handwritten digit database. [Электронный ресурс] – Режим доступа: https://yann.lecun.com/ exdb/mnist/, дата обращения 15.07.2024.
10. *Bergmann P. Löwe S., Fauser M., Sattlegger D., Steger C.* Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. [Электронный ресурс] – Режим доступа: https://arxiv.org/pdf/1807.02011, дата обращения 15.07.2024.
11. *Фигурнов М.В., Струминский К.А., Ветров Д.П.* Устойчивый к шуму метод обучения вариационного автокодировщика // Интеллектуальные системы. Теория и приложения. 2017. Т. 21. № 2. С. 90–109.
12. *Осипов К.Н., Заморёнов М.В.* Минимизация информационного расхождения Кульбака-Лейблера в задачах автоматизированной обработки измерительной информации // Известия Тульского государственного университета. Технические науки. 2019. № 3. С. 195–200.

13. *Сухань А.А.* Генеративно-состязательные нейронные сети в задачах определения трендов // Московский экономический журнал. 2019. № 6. С. 32. DOI 10.24411/2413-046X-2019-16031.
14. *Богданов Л.Ю.* Оценка эффективности бинарных классификаторов на основе логистической регрессии методом ROC-анализа // Вестник Саратовского государственного технического университета. 2010. Т. 4. № 2(50). С. 92–97.
15. *Костин Д.В., Шелухин О.И.* Сравнительный анализ алгоритмов машинного обучения для проведения классификации сетевого зашифрованного трафика // T-Comm: Телекоммуникации и транспорт. 2016. Т. 10. № 9. С. 43–52.

Информация об авторах

Владимир Евгеньевич Марлей – д.т.н., профессор
SPIN-код: 7564-1020
Андрей Николаевич Терехов – д.ф.-м.н., профессор
SPIN-код: 3834-3035
Юрий Арменакович Гатчин – д.т.н., профессор
SPIN-код: 4621-0078
Виталий Игоревич Милушков – аспирант
SPIN-код: не представлен
Николай Николаевич Лиманский – аспирант
SPIN-код: не представлен

Статья поступила в редакцию 26.08.2024
Одобрена после рецензирования 09.09.2024
Принята к публикации 26.09.2024

Original article

Comparative study of the effectiveness of autoencoders in anomaly detection tasks

V.E. Marley[1], **A.N. Terekhov**[2], **Yu.A. Gatchin**[3], **V.I. Milushkov**[4], **N.N. Limansky**[5]

[1,4,5] Admiral S.O. Makarov State University of Marine and River Fleet (Saint Petersburg, Russia)
[2] Saint Petersburg State University (Saint Petersburg, Russia)
[3] ITMO National Research University (Saint Petersburg, Russia)
[1] vmarley@mail.ru, [2] ant@tercom.ru, [3] gatchin1952@mail.ru, [4] info@sohoware.ru, [5] info@sohoware.ru

Abstract

Auto encoders are powerful tools for solving anomaly detection tasks due to their ability to learn how to compress and restore normal data. The main idea of using autoencoders is to create models capable of efficiently processing normal data. However, this makes it difficult to restore abnormal indicators, which leads to an increase in reconstruction errors. In this article, three types of autoencoders were investigated – convolutional autoencoder (CAE), variational autoencoder (VAE) and adversarial autoencoder (AAE).

To evaluate the performance of various autoencoder models in the task of detecting anomalies on the MNIST dataset, as well as to identify their advantages and limitations.

The results of the study show that all three models of autoencoders have high performance in detecting anomalies, but there is a decrease in performance and computational costs. CAE showed the best results in terms of speed, but in some cases it was inferior to VAE and AAE in terms of accuracy. Despite the more complex architecture and increased training time, VAE and AAE showed only a slight improvement over CAE.

The simplicity and speed of CAE may be preferable for a number of tasks, while VAE and AAE may be useful in cases where additional capabilities such as generating new data or improved noise tolerance are critical. Differences in the results of reconstruction of autoencoder models and their interpretation open up new opportunities for further research, including the development of hybrid approaches that can combine the strengths of these models.

Keywords

Anomaly detection, autoencoders, convolutional autoencoder, Variational autoencoder, adversarial autoencoder, MNIST

For citation

Marley V.E., Terekhov A.N., Gatchin Yu.A., Milushkov V.I., Limansky N.N. Comparative study of the effectiveness of autoencoders in anomaly detection tasks. Neurocomputers. 2024. V. 26. № 5. P. 96–106. DOI: https://doi.org/10.18127/j19998554-202405-09 (In Russian)

References

1. *Vanyashkin Yu.Yu., Makarov D.A., Popova I.A., Soboleva E.D.* Application of autocoders for removing noises from images. StudNet. 2020. V. 3. №. 10. P. 27. (in Russian)
2. *Lazarev A.S., Turovsky F.A., Pivovarov S.A.* Autocoder. Collection of articles of the International Scientific and Practical Conference "Innovative technologies in modern science": Ufa: Aeterna. 2017. V. 3. P. 82–84. (in Russian)
3. 3 Kalinin M.O. Prevention of cyberthreats in self-organizing m2m environments in case of a lack of training datasets. Digital economy and Industry 5.0: development in a new reality. 2022. P. 107–128. DOI 10.18720/IEP/2022.3/4. (in Russian)
4. *Milushkov V.I., Limansky N.N., Marley V.E.* Integrating hybrid semi-supervised and contrastive learning for automatic defect classification in manufacturing data: Improving product quality control accuracy. Prospects of science. 2024. № 6(177). P. 81–86. (in Russian)
5. *Limansky N.N., Milushkov V.I., Marley V.E.* Anomaly detection in robotic systems: comparative analysis of ConvLSTM with attention mechanism and traditional approaches. Prospects of Science. 2024. № 6(177). P. 77–80. (in Russian)
6. *Kuraedov V.I.* Application of machine learning methods for automatic test pattern generation process optimization during circuit design. Neurocomputers. 2024. V. 26. № 1. P. 14-22. DOI 10.18127/j19998554-202401-02. (In Russian)
7. *Volkov A.K., Mironova L.V., Potapova S.E.* The use of pretrained neural networks for solving the problem of reverse searching of X-ray images of prohibited items and substances. Civil Aviation High Technologies. 2024. V. 27. № 2. P. 8–24. DOI 10.26467/2079-0619-2024-27-2-8-24. (in Russian)
8. *Belyakov A.N., Zhukov V.P., Shirokov M.O.* Improving the quality of pattern recognition using a modified variational autoencoder. Materials of the International Scientific and Technical Conference. "The state and prospects of development of electrical and thermal technology". Ivanovo: Ivanovo State Power Engineering University named after V.I. Lenin. 2023. V. 2. P. 362–365. (in Russian)
9. *LeCun Y., Cortes C., Burges C.J.C.* MNIST handwritten digit database. [Electronic resource] – Access mode: https://yann.lecun.com/exdb/mnist/, date of reference 15.07.2024.
10. *Bergmann P. Löwe S., Fauser M., Sattlegger D., Steger C.* Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. [Electronic resource] – Access mode: https://arxiv.org/pdf/1807.02011, date of reference 15.07.2024.
11. *Figurnov M.V., Struminsky K.A., Vetrov D.P.* Noise-robust method for training of variational autoencoder. Intelligent systems. Theory and applications. 2017. V. 21. № 2. P. 90–109. (in Russian)
12. *Osipov K.N., Zamorenov M.V.* Minimization of the Kulback-Leibler information discrepancy in the tasks of automated processing of measuring information. Izvestiya Tula State University. Technical sciences. 2019. № 3. P. 195–200. (in Russian)
13. *Sukhan A.A.* Generative-adversarial neural networks in the tasks of determining trends. Moscow Economic Journal. 2019. № 6. P. 32. DOI 10.24411/2413-046X-2019-16031. (in Russian)

14. *Bogdanov L.Y.* The evaluation of performance of binary classifiers based on logistic regression using ROC analysis. Bulletin of the Saratov State Technical University. 2010. V. 4. № 2(50). P. 92–97. (in Russian)
15. *Kostin D.V., Sheluhin O.I.* Comparison of machine learning algorithms for encrypted traffic classification. T-Comm. 2016. V. 10. № 9. P. 43–52. (in Russian)

Information about the authors

Vladimir E. Marley – Dr.Sc. (Eng.), Professor
Andrey N. Terekhov – Dr.Sc. (Phys.-Math.), Professor
Yuri A. Gatchin – Dr.Sc. (Eng.), Professor
Vitaly I. Milushkov – Post-graduate Student
Nikolai N. Limansky – Post-graduate Student

The article was submitted 26.08.2024
Approved after reviewing 09.09.2024
Accepted for publication 26.09.2024