Московский государственный университет имени М.В. Ломоносова
Механико-математический факультет
Кафедра Математической теории интеллектуальных систем (МаТИС)

Курсовая работа

Сравнительный анализ автоэнкодера и вариационного автоэнкодера

Студент 331 группы
Ряховский Алексей Сергеевич

Научный руководитель:
Миронов Андрей Михайлович

Москва, 2025

Содержание

Введение **2**
Актуальность . . . 2
Практическая значимость . . . 2

Теоретические основы **3**
Автоэнкодер (Autoencoder - AE) . . . 3
Определение и архитектура . . . 3
Функции потерь . . . 3
Сверточные автоэнкодеры (Convolutional Autoencoders - CAE) . . . 3
Вариационный автоэнкодер (VAE) . . . 4
Вероятностный подход и генеративная природа . . . 4
Архитектура . . . 4
Функция потерь . . . 5
Трюк с репараметризацией (Reparameterization Trick) . . . 6

Постановка эксперимента **7**
Датасет . . . 7
Целевые метрики . . . 8
Формирование выборки и обучение . . . 8
Сценарии оценки моделей . . . 8
Визуализация . . . 9
Архитектуры моделей . . . 9

Результаты эксперимента **10**
Реконструкция VAE/AE . . . 10
Устранение шумов VAE/AE . . . 11
Классификация по латентному пространству VAE/AE . . . 12

Выводы **13**

Введение

Автоэнкодеры (AE) и вариационные автоэнкодеры (VAE) являются фундаментальными архитектурами нейронных сетей в области обучения без учителя, предназначенными для изучения эффективных представлений данных. AE нацелены на сжатие данных в низкоразмерное латентное пространство и их последующую реконструкцию с минимальными потерями. VAE, являясь генеративной моделью, расширяют эту концепцию, моделируя распределение вероятностей в латентном пространстве, что позволяет не только реконструировать, но и генерировать новые данные.

Несмотря на общую структуру (энкодер-декодер), AE и VAE различаются математическими основаниями, функциями потерь и, как следствие, свойствами латентных пространств и областями применения. Понимание этих различий критично для выбора адекватной модели. Данная работа посвящена сравнительному анализу AE и VAE на задачах реконструкции изображений, классификации на основе латентных признаков и шумоподавления с использованием датасета MNIST.

Актуальность

Данное исследование представляется актуальным, поскольку прямое и систематическое сравнение этих двух типов моделей на стандартизированных задачах и общепринятом датасете, таком как MNIST, позволяет выявить их относительные преимущества и недостатки в контролируемых условиях. Такое исследование предоставляет исследователям и инженерам обоснованные критерии для выбора наиболее подходящей архитектуры для решения их задач. Понимание фундаментальных различий и сравнительных характеристик AE и VAE напрямую влияет на эффективность решения практических задач и способствует разработке новых, более совершенных моделей глубокого обучения. Это особенно важно в контексте постоянно растущего объема данных и возрастающей потребности в их эффективной обработке, анализе и интерпретации.

Практическая значимость

Результаты данной работы обладают непосредственной практической ценностью. Во-первых, они предоставляют разработчикам количественные и качественные ориентиры для выбора между AE и VAE в зависимости от приоритетов конкретной задачи: будь то максимальная точность восстановления данных, качество генерируемых признаков для последующих задач классификации или эффективность в удалении шумов. Во-вторых, понимание сильных и слабых сторон каждой модели позволяет более эффективно распределять вычислительные ресурсы и время на разработку. В-третьих, детальное математическое описание и сравнительный анализ служат ценным образовательным ресурсом для студентов и специалистов, осваивающих методы глубокого обучения, особенно для тех, кто обучается на программах с углубленным изучением математики.

Теоретические основы

Автоэнкодер (Autoencoder - AE)

Определение и архитектура

Автоэнкодер (AE) представляет собой тип нейронной сети, используемый для обучения эффективных кодированных представлений данных без учителя (обучение представлений). AE состоит из двух основных частей:

- **Энкодер** (e): Преобразует входные данные $x \in \mathbb{R}^D$ в латентное (скрытое) представление $h \in \mathbb{R}^L$, обычно с $L < D$. Это можно записать как $h = e_\phi(x)$, где ϕ — параметры энкодера.
- **Декодер** (d): Восстанавливает (реконструирует) входные данные $\hat{x} \in \mathbb{R}^D$ из латентного представления h. Это можно записать как $\hat{x} = d_\theta(h)$, где θ — параметры декодера. Параметры ϕ и θ оптимизируются совместно с целью минимизации ошибки реконструкции.

Если $L < D$, автоэнкодер называется неполным (undercomplete) и вынужден изучать наиболее важные признаки данных для их сжатия.

Функции потерь

Целью обучения автоэнкодера является минимизация функции потерь $L(x, \hat{x})$, которая измеряет ошибку реконструкции между исходными данными x и реконструированными данными $\hat{x}$. Для вещественных данных часто используется среднеквадратическая ошибка (MSE):

$$L_{MSE}(x, \hat{x}) = \frac{1}{D} \sum_{j=1}^{D} (x_j - \hat{x}_j)^2$$

Для бинарных данных или данных в диапазоне, интерпретируемых как вероятности (например, если последний слой декодера использует сигмоидную функцию активации), альтернативой является бинарная перекрестная энтропия (BCE):

$$L_{BCE}(x, \hat{x}) = -\frac{1}{D} \sum_{j=1}^{D} (x_j \log \hat{x_j} + (1 - x_j) \log(1 - \hat{x_j}))$$

В данной работе для AE в качестве функции потерять был выбран MSE. Также он используется для оценки качества работы моделей в задачах реконструкции и шумоподавления.

Сверточные автоэнкодеры (Convolutional Autoencoders - CAE)

Существуют различные типы автоэнкодеров, расширяющие базовую архитектуру, но в данном сравнении мы рассмотрим только сверточные автоэнкодеры. Они используют сверточные слои в энкодере и транспонированные сверточные (или деконволюционные) слои в декодере, что особенно эффективно для обработки пространственных данных, таких как изображения. Применение CAE для датасета MNIST является естественным выбором, поскольку сверточные слои эффективно улавливают локальные паттерны (штрихи, изгибы, углы), формирующие цифры, и обладают свойством частичной инвариантности к сдвигам, что важно для распознавания рукописных символов.

Вариационный автоэнкодер (VAE)

Вероятностный подход и генеративная природа

Вариационный автоэнкодер (VAE) —- это более сложная архитектура, которая относится к классу генеративных моделей и основана на вероятностном подходе к процессам кодирования и декодирования данных. В отличие от детерминированного кодирования, применяемого в классических автоэнкодерах (где каждый вход x отображается в единственную точку h латентного пространства), VAE отображает входные данные x в параметры вероятностного распределения в латентном пространстве. Обычно это многомерное Гауссово распределение, характеризуемое вектором средних значений и вектором дисперсий.

Такой вероятностный подход наделяет VAE важным свойством: он позволяет не только реконструировать исходные данные, но и генерировать новые, правдоподобные образцы данных. Генерация происходит путем сэмплирования точки из латентного распределения (обычно из априорного, например, стандартного нормального) и последующей ее передачи через декодер. Именно вероятностная природа энкодера (кодирование в распределение, а не в детерминированную точку) и специфическая функция потерь (включающая член KL-дивергенции, о котором будет сказано ниже) обеспечивают VAE генеративными способностями.

Архитектура

Архитектура VAE, как и у AE, включает энкодер и декодер, но их функции и интерпретация несколько иные.

- **Энкодер** (также называемаый моделью распознавания или inference model) $q_\phi(z|x)$: Для каждого входного объекта x энкодер VAE предсказывает параметры (например, вектор средних $\mu_\phi(x)$ и вектор логарифмов диагональных элементов матрицы ковариации, часто представляемый как $\log(\sigma^2_\phi(x))$) апостериорного распределения $q_\phi(z|x)$ в латентном пространстве $z \in \mathbb{R}^L$. Это распределение описывает правдоподобие различных латентных векторов z для данного входа x. Чаще всего в качестве $q_\phi(z|x)$ используется многомерное нормальное (Гауссово) распределение с диагональной матрицей ковариации:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x)))$$

- **Латентное пространство** z: Во время обучения латентный вектор z сэмплируется из апостериорного распределения $q_\phi(z|x)$, предсказанного энкодером. Во время генерации новых данных (после обучения модели) z сэмплируется из априорного распределения $p(z)$. В качестве априорного распределения обычно выбирают стандартное многомерное нормальное распределение $\mathcal{N}(0, I)$, где I – единичная матрица. Это делается для того, чтобы латентное пространство было центрировано около нуля и имело определенную структуру.

- **Декодер** (также называемый генеративной моделью или generative model) $p_\theta(x|z)$: Декодер VAE принимает на вход латентный вектор z (сэмплированный либо из $q_\phi(z|x)$ во время обучения, либо из $p(z)$ во время генерации) и генерирует (реконструирует) данные x' в исходном пространстве. Модель $p_\theta(x|z)$ также является вероятностной. Она определяет распределение правдоподобия для x при заданном z. Например:

– Если исходные данные x являются вещественными (например, нормализованные значения пикселей изображения), то $p_\theta(x|z)$ может быть Гауссовым распределением $\mathcal{N}(x|d_\theta(z), \sigma^2_{dec}I)$, где $d_\theta(z)$– это выход нейронной сети декодера (предсказанное среднее для x), а σ^2_{dec}– фиксированная дисперсия (или также предсказываемая декодером). В этом случае ошибка реконструкции будет связана с MSE.

– Если исходные данные x являются бинарными (например, черно-белые изображения MNIST, где пиксели могут принимать значения 0 или 1), то $p_\theta(x|z)$ часто моделируется как произведение независимых распределений Бернулли для каждого пикселя. В этом случае ошибка реконструкции соответствует бинарной перекрестной энтропии (Binary Cross-Entropy, BCE). Здесь θ – параметры нейронной сети декодера.

Функция потерь

Обучение VAE происходит путем максимизации так называемой доказательной нижней оценки (Evidence Lower Bound, ELBO) логарифма правдоподобия данных $\log p(x)$. Максимизация ELBO эквивалентна минимизации отрицательной ELBO, которая и используется в качестве функции потерь L_{VAE}:

$$L_{VAE} = \underbrace{-\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{Ошибка реконструкции}} + \underbrace{D_{KL}(q_\phi(z|x)||p(z))}_{\text{KL-дивергенция}}$$

Эта функция потерь состоит из двух основных компонент:

- **Ошибка реконструкции (Reconstruction Loss):** $-\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ измеряет, насколько хорошо декодер восстанавливает входной объект x из латентного представления z, сэмплированного из распределения $q_\phi(z|x)$, предсказанного энкодером. Фактически, он стремится максимизировать правдоподобие исходных данных при условии их латентного представления. Как упоминалось ранее, если $p_\theta(x|z)$ – Гауссово распределение с выходом декодера $d_\theta(z)$ в качестве среднего, то этот член (с точностью до константы и множителя) будет эквивалентен среднеквадратичной ошибке между x и $d_\theta(z)$. Для бинарных данных (например, изображения MNIST, где значения пикселей нормализованы в диапазон и могут интерпретироваться как вероятности) и Бернуллиевского $p_\theta(x|z)$, этот член соответствует бинарной перекрестной энтропии.

- **KL-дивергенция (Kullback-Leibler Divergence):** $D_{KL}(q_\phi(z|x)||p(z))$ измеряет расхождение между апостериорным распределением $q_\phi(z|x)$, предсказанным энкодером для данного x, и априорным распределением $p(z)$. Этот член действует как регуляризатор, заставляя распределения $q_\phi(z|x)$, генерируемые энкодером для различных входных объектов, быть близкими к априорному распределению $p(z)$. Это способствует формированию гладкого, непрерывного и структурированного латентного пространства, что критически важно для генеративных возможностей VAE: если латентное пространство хорошо "организовано то сэмплирование из него и интерполяция между точками будут приводить к осмысленным результатам. Если в качестве $q_\phi(z|x)$ используется Гауссово распределение $\mathcal{N}(z|\mu, \text{diag}(\sigma^2))$ и в качестве $p(z)$ стандартное Гауссово распределение $\mathcal{N}(z|0, I)$, то KL-дивергенция между ними имеет аналитическую форму:

$$D_{KL}(\mathcal{N}(\mu, \text{diag}(\sigma^2))||\mathcal{N}(0, I)) = \frac{1}{2}\sum_{j=1}^{L}(\sigma_j^2 + \mu_j^2 - 1 - \log\sigma_j^2)$$

где L — размерность латентного пространства, а суммирование ведется по всем компонентам векторов μ и σ.

Функция потерь VAE, таким образом, представляет собой компромисс между двумя целями. Первый член (ошибка реконструкции) стремится к тому, чтобы модель как можно точнее восстанавливала входные данные. Второй член (KL-дивергенция) стремится к тому, чтобы латентное пространство было хорошо структурировано и близко к априорному распределению. Этот компромисс часто приводит к тому, что реконструкции, получаемые с помощью VAE, могут быть несколько более "размытыми" или менее точными по сравнению с реконструкциями классических AE, которые оптимизируют исключительно ошибку реконструкции. Однако эта "цена" платится за получение генеративных свойств и более полезной структуры латентного пространства.

Трюк с репараметризацией (Reparameterization Trick)

Для обучения VAE методом обратного распространения ошибки необходимо дифференцировать функцию потерь ELBO по параметрам энкодера ϕ и декодера θ. Однако член реконструкции содержит операцию сэмплирования $z \sim q_\theta(zx)$, которая является стохастической и не позволяет градиенту проходить напрямую. Эту проблему решает трюк с репараметризацией. Идея заключается в том, чтобы выразить случайную переменную z как детерминированную функцию от параметров энкодера $(\mu_\phi(x), \sigma_\phi(x))$ и вспомогательной независимой случайной переменной ϵ, имеющей фиксированное распределение (например, $\varepsilon \sim \mathcal{N}(0, I)$). Для Гауссова случая z вычисляется как :

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \varepsilon$$

где $\odot$ обозначает поэлементное умножение. Таким образом, стохастичность выносится за пределы параметров модели, и градиенты могут беспрепятственно распространяться через $\mu_\phi(x)$ и $\sigma_\phi(x)$.

Постановка эксперимента

Датасет

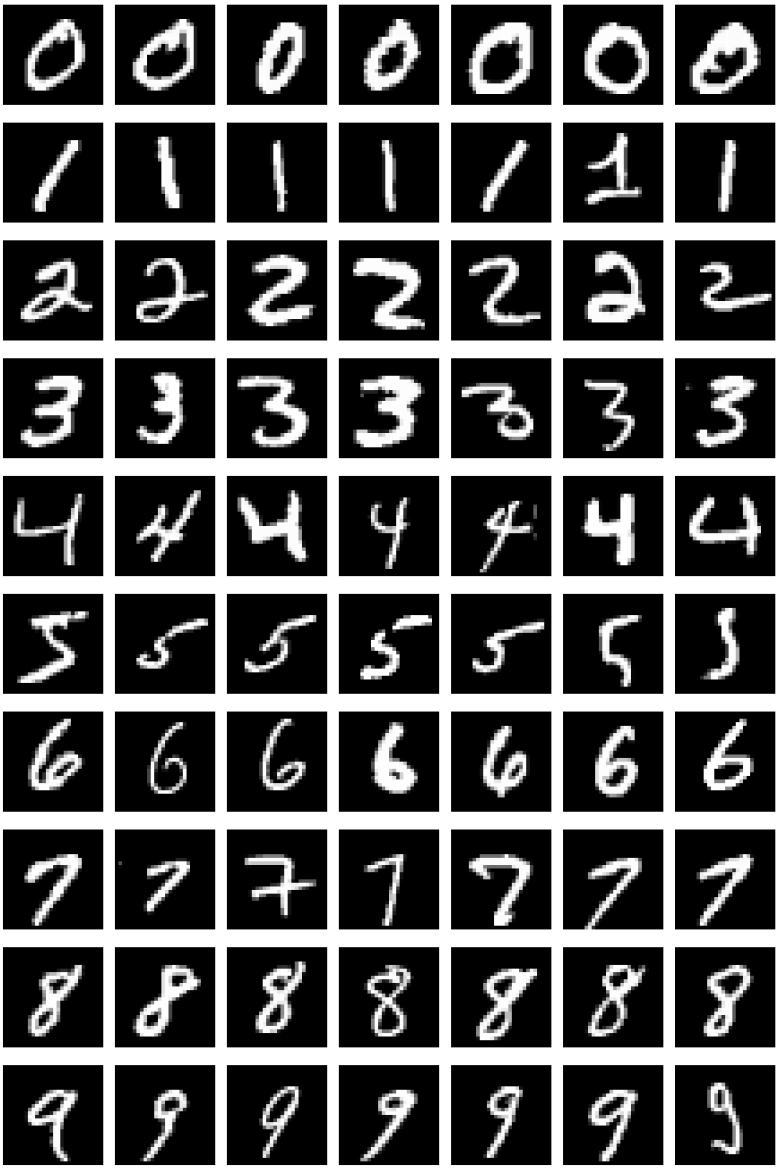

Рис. 1: Примеры изображений `MNIST`

Используется датасет `MNIST`, содержащий изображения рукописных цифр размером 28×28 пикселей. В наборе представлено 10 классов (цифры от 0 до 9), при этом на каждый класс приходится по $6,000$ изображений в обучающей выборке и по $1,000$ — в тестовой.

Целевые метрики

1. **Среднеквадратическая ошибка (Mean Squared Error, MSE):** Применяется для оценки качества реконструкции и шумоподавления. Меньшее значение MSE соответствует лучшему качеству реконструкции/шумоподавления. Рассчитывается по формуле:

$$\text{MSE} = \frac{1}{M \cdot H \cdot W} \sum_{i=1}^{M} \sum_{j=1}^{H} \sum_{k=1}^{W} (x_{ijk} - \hat{x}_{ijk})^2$$

   где M – количество изображений, H, W – высота и ширина, x_{ijk} - значение пикселя исходного изображения, $\hat{x}_{ijk}$ - значение пикселя реконструированного изображения.

2. **Точность классификации (Accuracy):** Будет использоваться для оценки информативности латентных представлений, полученных с помощью AE и VAE, с точки зрения их пригодности для решения задачи классификации. Более высокая точность классификации на основе латентных признаков указывает на то, что модель смогла извлечь более релевантную для разделения классов информацию. Формула для Accuracy:

$$\text{Accuracy} = \frac{\text{Число правильно классифицированных примеров}}{\text{Общее число примеров}}$$

Формирование выборки и обучение

- Все изображения нормализуются в диапазон $[0, 1]$ и преобразуются в тензоры.
- Обучение моделей проводится с использованием стохастического градиентного спуска (Adam) на тренировочной выборке.
- Для задачи шумоподавления к изображениям добавляется Гауссов шум с нулевым средним и заданной дисперсией.

Сценарии оценки моделей

- **Реконструкция**:
  - Оценивается способность моделей точно восстанавливать входные данные (в нашем случае изображения).
  - Обучаем модели на тестовых (оригинальных) данных, считаем оценку MSE на тестовых данных, сравниваем модели.
- **Классификация по латентному пространству**:
  - Из латентных векторов, полученных из моделей AE/VAE, извлекаются признаки.
  - Обучается классификатор (дерево решений) для предсказания класса изображения.
  - Оценивается точность классификации (Accuracy) на тестовой выборке.

- **Шумоподавление**:
  - Добавляем к оригинальным изображениям Гауссов шум с нулевым средним и заданной дисперсией, подаем модели на вход зашумлённое изображение, выход сравнивается с чистым оригиналом.
  - Среднеквадратическая ошибка (MSE) реконструкции используется как метрика качества восстановления.

Визуализация

- Реконструкции и денойзинг визуализируются в виде таблиц изображений (оригинал, результат VAE, результат AE).
- Также производится t-SNE-визуализация латентных пространств для обоих автоэнкодеров.

Архитектуры моделей

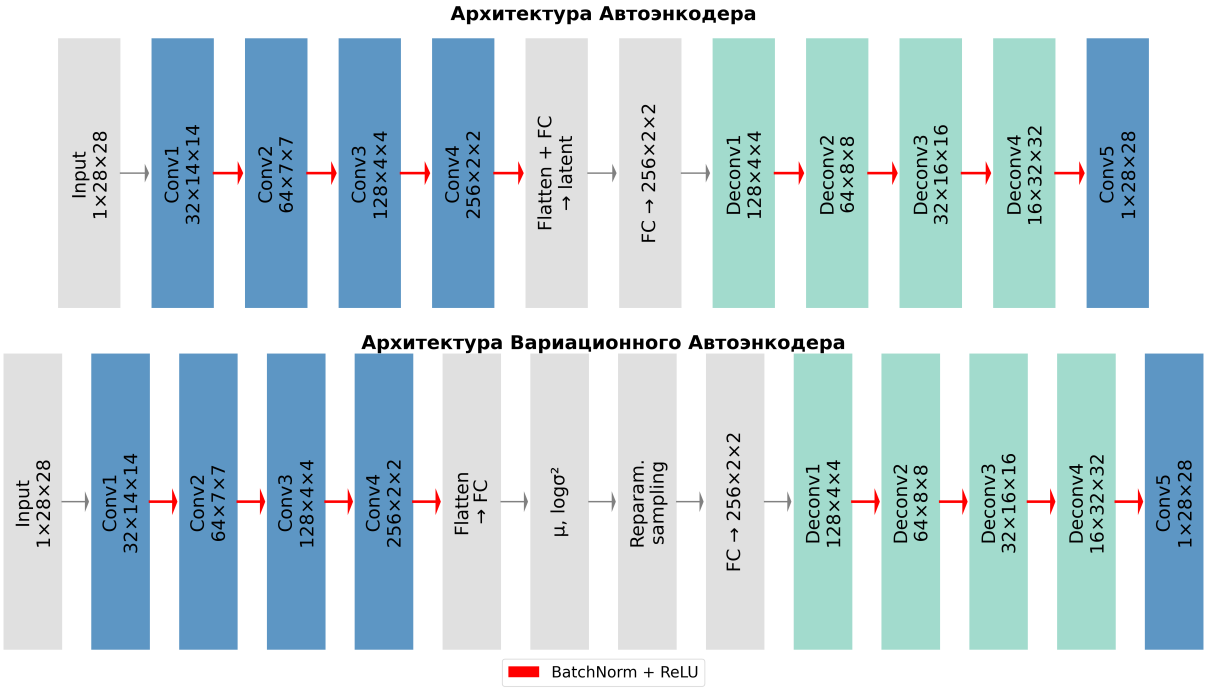

Рис. 2: Архитектуры AE и VAE

Результаты эксперимента

Реконструкция VAE/AE

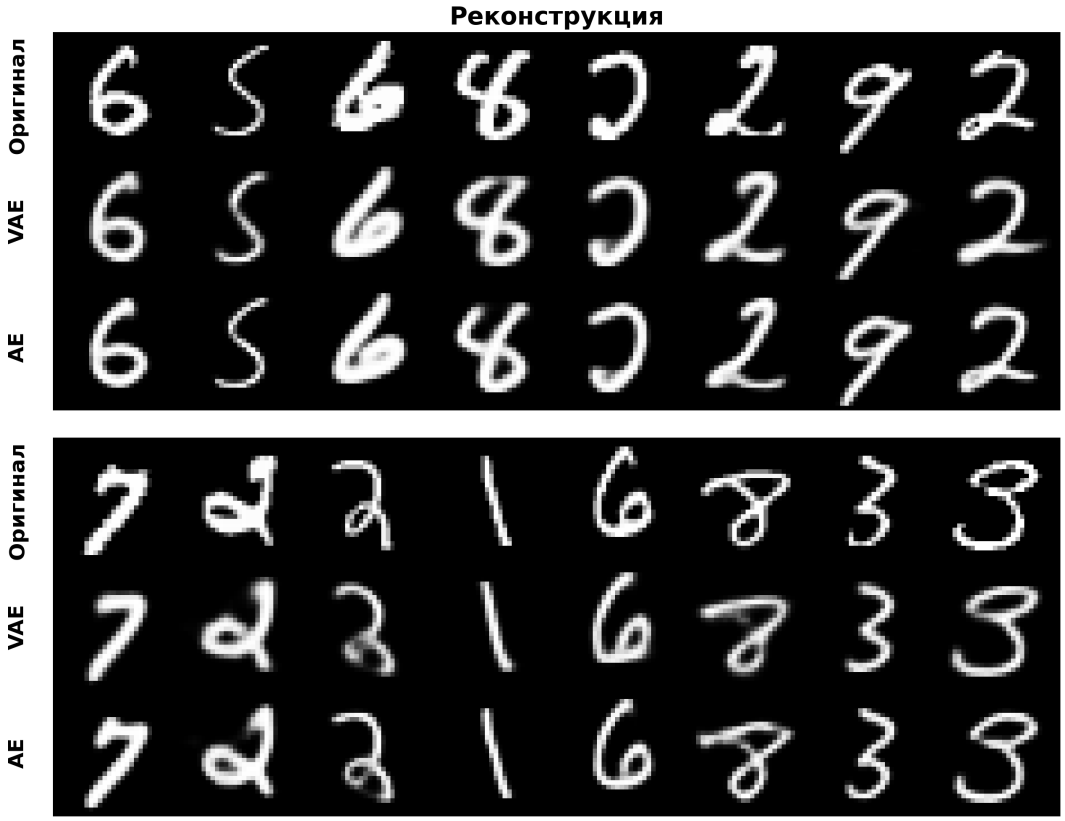

Рис. 3: Примеры результатов реконструкций AE и VAE

	AE	VAE
MSE	0.0047	0.0099

Таблица 1: Средние численные результаты реконструкции изображений AE/VAE

Несмотря на то, что визуально обе модели хорошо справляются с реконструкцией цифр, победил все равно автоэнкодер. Он и более точно передает границы цифр, практически полностью повторяя оригинал. MSE-loss у AE также ниже приблизительно в 2 раза, чем у VAE. Результаты VAE несколько более заметно отличаются от оригиналов, иногда пропадают или более размыто передаются детали цифр.

Устранение шумов VAE/AE

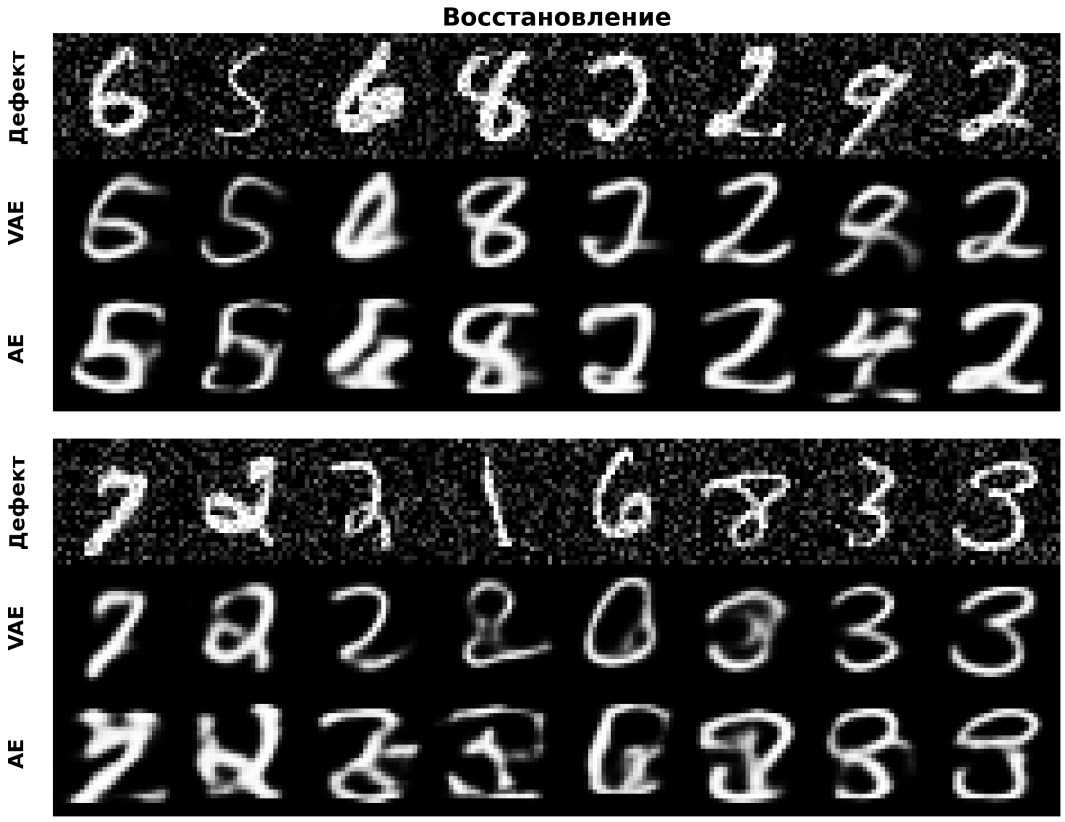

Рис. 4: Примеры результатов шумоподавления AE и VAE

	AE	VAE
MSE	0.077	0.044

Таблица 2: Средние численные результаты шумоподавления изображений AE/VAE

Визуально в этом сравнении однозначно победил VAE. Он чаще, чем AE выдавал более “адекватный“ результат. Стоит заметить, что VAE чаще выдавал корректный класс цифр, теряя немного в самой структуре цифры, в то время как AE часто выдавал неосмысленный результат. По метрикам также победил VAE, показав в среднем на 75% лучший MSE-loss.

Классификация по латентному пространству VAE/AE

	AE	VAE
Accuracy	0.8779	0.8371

Таблица 3: Средний Accuracy классификации по латентному пространству у AE/VAE

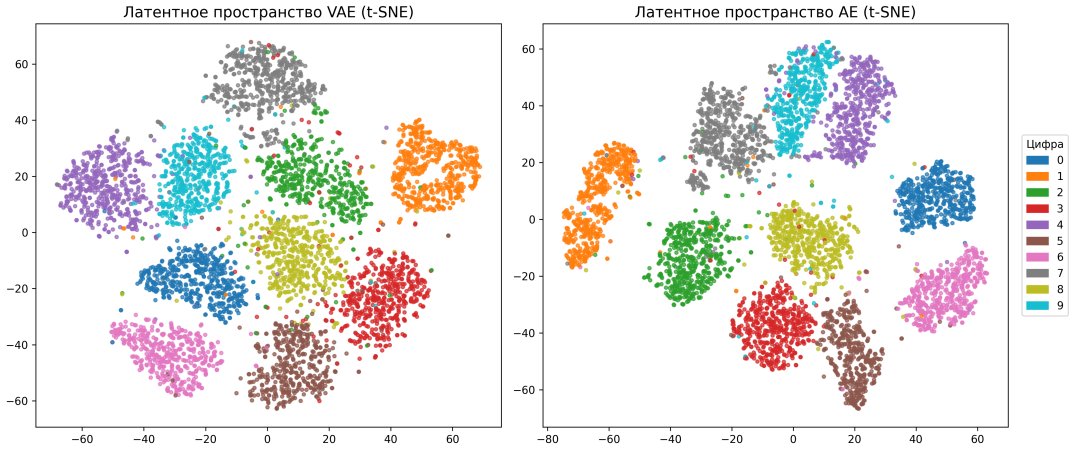

Рис. 5: Латентные пространства AE/VAE

По итогам классификации с использованием признаков, извлечённых из латентного пространства моделей, было установлено, что AE обеспечивает более высокую точность по сравнению с VAE, хотя отличия и не очень большие, примерно 5%. Также на Рис. 5 можно увидеть, что облака классов у AE более компактные и разграниченные, что и помогает лучше классифицировать вектора z. Что касается VAE, то у него латентное пространство более гладкое и непрерывное, тут кластеры находятся ближе друг к другу, также видно явно выраженную дисперсию точек.

Выводы

В рамках настоящей курсовой работы был проведен сравнительный анализ классического автоэнкодера (AE) и вариационного автоэнкодера (VAE) на задачах реконструкции изображений, устранения шумов и классификации на основе латентных представлений с использованием датасета MNIST. Анализ позволил выявить сильные и слабые стороны каждой архитектуры в контексте указанных задач:

1. **Реконструкция изображений:** AE показал более высокую точность (MSE 0.0047) по сравнению с VAE (MSE 0.0099), обеспечивая более четкие реконструкции. Это объясняется прямой минимизацией ошибки реконструкции в AE, в то время как VAE также оптимизирует KL-дивергенцию для регуляризации латентного пространства, что может приводить к некоторой потере деталей.

2. **Устранение шумов:** VAE продемонстрировал значительно лучшие результаты (MSE 0.044) по сравнению с AE (MSE 0.077). VAE чаще восстанавливал корректный класс цифры и генерировал более чистые изображения из зашумленных данных. Вероятностная природа VAE и регуляризация латентного пространства позволяют ему эффективнее отфильтровывать шум.

3. **Классификация по латентному пространству:** Латентные представления AE обеспечили более высокую точность классификации (Accuracy 0.8779) с помощью дерева решений, чем VAE (Accuracy 0.8371). Визуализация t-SNE показала более компактные и четко разделенные кластеры для AE. VAE, нацеленный на генерацию и гладкость латентного пространства, формирует более перекрывающиеся кластеры.

4. **Общий итог:** Выбор между AE и VAE зависит от задачи. AE предпочтительнее для точной реконструкции и получения дискриминативных признаков для классификации. VAE лучше подходит для генерации данных, интерполяции и шумоподавления, благодаря структурированному латентному пространству, что может достигаться ценой некоторого снижения точности реконструкции.

Список литературы

[1] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[2] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN 2011), Part I*, volume 6791 of *LNCS*, pages 52–59. Springer, 2011.

[3] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014)*, 2014. arXiv:1312.6114.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015. arXiv:1412.6980.

[6] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[9] Samreen Naeem, Aqib Ali, Sania Anam, and Muhammad Munawar Ahmed. An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, 13(1):911–921, 2023.