**Генерация изображений с помощью вариационного автоэнкодера (VAE)**

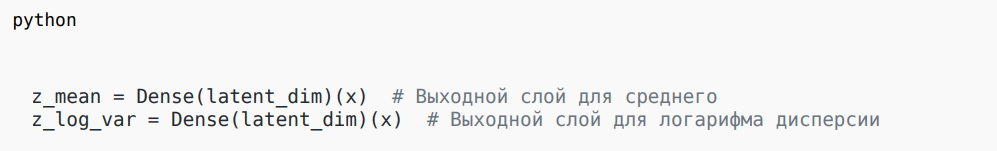
**Цель работы:** Изучить принципы работы вариационного автоэнкодера (VAE) и научиться генерировать новые изображения с его помощью, используя набор данных MNIST.

**Теоретическая часть: Вариационные автоэнкодеры (VAE)**

Вариационный автоэнкодер (VAE) — это особый вид автоэнкодера, который учится представлять входные данные как вероятностные распределения в латентном пространстве, а не как фиксированные точки. Это позволяет VAE генерировать новые данные, сэмплируя значения из этих распределений.

**Задание**

1. **Загрузка и предобработка данных**:
   * Загрузите и нормализуйте данные MNIST, чтобы значения пикселей были от 0 до 1.
2. **Создание вариационного автоэнкодера (VAE)**:
   * **Кодировщик**: создайте свёрточный кодировщик, как вы делали ранее, но добавьте два выходных слоя:
     + Один слой для среднего (μ).
     + Другой слой для логарифма дисперсии (log(σ²)).
   * **Подсказка**: Для создания этих выходных слоёв используйте Dense слои:



1. **Репараметризация**:

* Используйте формулу z = μ + σ \* ε, чтобы сэмплировать случайные латентные векторы. Для этого реализуйте лямбда-слой:
* **Подсказка**: Вот пример кода для репараметризации

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

1. **Декодировщик**: создайте декодер, используя транспонированные свёрточные слои для восстановления изображений из латентного пространства.
2. **Определение функции потерь**:

* Функция потерь состоит из двух частей:
  + **Реконструкционная ошибка**: вычисляется как бинарная кросс-энтропия между входным и восстановленным изображением.
  + **KL-дивергенция**: вычисляет расхождение между распределением кодов и стандартным нормальным распределением.
* **Подсказка**: Вот пример кода для функции потерь

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

1. **Обучение модели**:

* Скомпилируйте модель с оптимизатором Adam и обучите её на тренировочных данных.

1. **Генерация новых изображений**:

* Создайте случайные латентные векторы и используйте декодер для генерации изображений.
* Мы создаём равномерную сетку значений в латентном пространстве (от -4 до 4), чтобы охватить большую часть значений, которые VAE может генерировать. Это позволяет увидеть, какие изображения декодер создаёт для разных латентных векторов.

1. **Визуализация латентного пространства**:

* Отобразите латентные представления тестовых изображений и используйте цветовую палитру, чтобы отметить различные цифры.
* Используйте кодирующую часть сети для генерации пространства представлений на основе тестовой части датасета.
* Используйте метки цифр, чтобы раскрасить точки и увидеть, как разные классы распределяются в латентном пространстве.