

---

# МЕТОДЫ ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ ГЛУБОКИХ ГЕНЕРАТИВНЫХ МОДЕЛЕЙ

---

**Мария Никитина**

`nikitina.mariia@phystech.edu`

**Антон Бишук**

`anton.bishuk@mail.ru`

**Олег Бахтеев**

`bakhteev(at)phystech.edu`

22 декабря 2024 г.

## АННОТАЦИЯ

Увеличение времени и ресурсов, затрачиваемых на обучение больших моделей привели к появлению большого количества работ, направленных на поиск уже существующей обученной модели, подходящей под новую задачу. Множество исследований направлено на поиск пространства моделей-датасетов, с помощью которого можно отыскать уже существующую модель, хорошо подходящую под новую задачу. Однако, исследования проводятся в основном в области дискриминативных моделей. Эта работа направлена на поиск векторного представления генеративных моделей, описывающего статистические свойства датасетов, на которых они обучены. Таким образом с помощью такого пространства можно подбирать подходящую генеративную модель, используя привычные операции с векторами. Эксперименты проводятся на VAE и Autoencoder.

# 1 Введение

Пусть задан некоторый набор генеративных моделей, описывающий разные выборки/генеральные совокупности данных. Требуется предложить метод векторного представления этих моделей, который будет сохранять статистические свойства данных. С помощью такого представления можно облегчить поиск подходящей обученной модели без больших затрат времени и ресурсов на обучения. Также поиск по пространству генеративных моделей может быть применим для анализа качества работы разных архитектур на задаче генерации требуемых данных.

Векторное пространство должно отвечать следующим требованиям:

1. Расстояние между векторными представлениями моделей для близких выборок должно быть невелико (при условии, что сами модели хорошо их описывают);
2. Модели, обученные на композиции/смеси выборок должны учитывать свойства всех выборок, входящих в смесь.

Для выполнения данных требования и решения задачи в данной статье исследуются и сравниваются три варианта решения:

1. Один из возможных вариантов: сумма векторных представлений моделей, полученных по датасетам  $D_1$ ,  $D_2$  должна приблизительно соответствовать векторному представлению датасета  $D_1 + D_2$ . Пример для итогового пространства в виде единичной сферы представлен на рис. 1;
2. Вместо использования евклидова расстояния на векторных представлениях, использовать иерархию;
3. Представить модель как граф и работать с пространством графов.

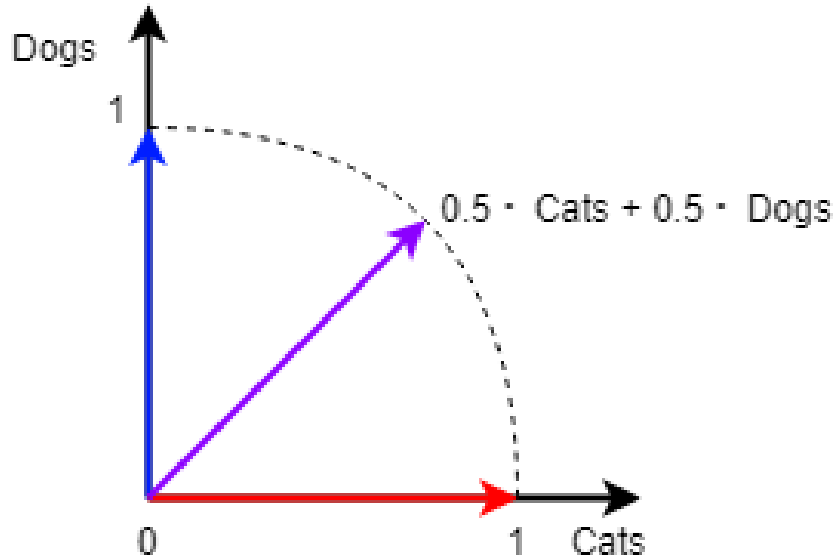


Рис. 1: Требуемое соотношение между моделями в векторном пространстве

## 2 Вычислительные эксперименты

### 2.1 Архитектура

Архитектура решения представлена на рис 2.

1. Случайное сэмплирование долей классов в датасете для  $n$  моделей;
2. Обучение  $n$  моделей на соответствующих датасетах;
3. Получение векторов из обученных моделей;
4. Обучение энкодера на полученных моделях. Предсказание:
  - (a) Вектора на части единичной сферы
  - (b) Расстояния между двумя моделями

Метрики для реализации обучения:

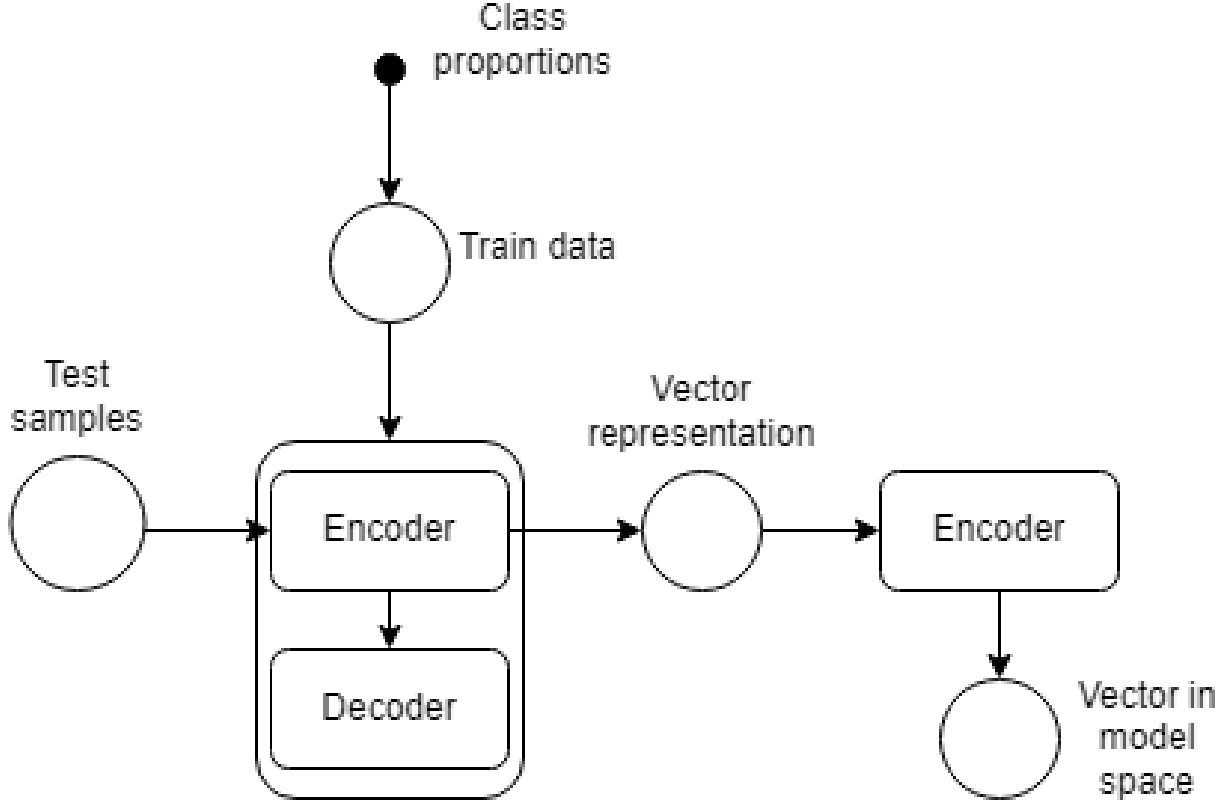


Рис. 2: Архитектура эксперимента

1. Contrastive N-pair loss, где  $\text{Encoder}(m_i) = \mathbf{x}$ ,  $\mathbf{d}_i = \mathbf{x}_i^+$ , остальные элементы батча:  $\mathbf{x}_i^-$ :

$$\mathcal{L}_{N\text{-pair}}(f) = -\log \frac{\exp(\mathbf{x}^T \mathbf{x}_i^+)}{\exp(\mathbf{x}^T \mathbf{x}_i^+) + \sum_{i=1}^{N-1} \exp(\mathbf{x}^T \mathbf{x}_i^-)};$$

2. Угол между вектором модели  $\text{Encoder}(m_i)$  и датасета  $\mathbf{d}_i$ :

$$\mathcal{L}_{\cos}(\text{Encoder}(m_i), \mathbf{d}_i) = \cos(\text{Encoder}(m_i) \cdot \mathbf{d}_i);$$

3. Triplet loss:

$$\mathcal{L}_{\text{Triplet}} = \sum_{x \in \chi} \max(0, \|\mathbf{x} - \mathbf{x}^+\|_2^2 - \|\mathbf{x} - \mathbf{x}^-\|_2^2 + \varepsilon);$$

4. MSE между  $\text{Encoder}(m_i)$  и  $\mathbf{d}_i$ :

$$\mathcal{L}_{MSE}(\text{Encoder}(m_i), \mathbf{d}_i) = \|\text{Encoder}(m_i) - \mathbf{d}_i\|_2^2.$$

## 2.2 Предсказание вектора в пространстве моделей-датасетов

Для базового эксперимента берётся 3 наиболее удалённых класса из датасета CIFAR. Для поиска таких классов использовалось евклидово расстояние на эмбедингах, полученных из выходного слоя ResNet.