# Методы векторного представления глубоких генеративных моделей

**Мария Никитина**
nikitina.mariia@phystech.edu

**Антон Бишук**
anton.bishuk@mail.ru

**Олег Бахтеев**
bakhteev(at)phystech.edu

15 мая 2025 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 евклидового расстояния на векторных представлениях, использовать иерархию;
3. Представить модель как граф и работать с пространством графов.

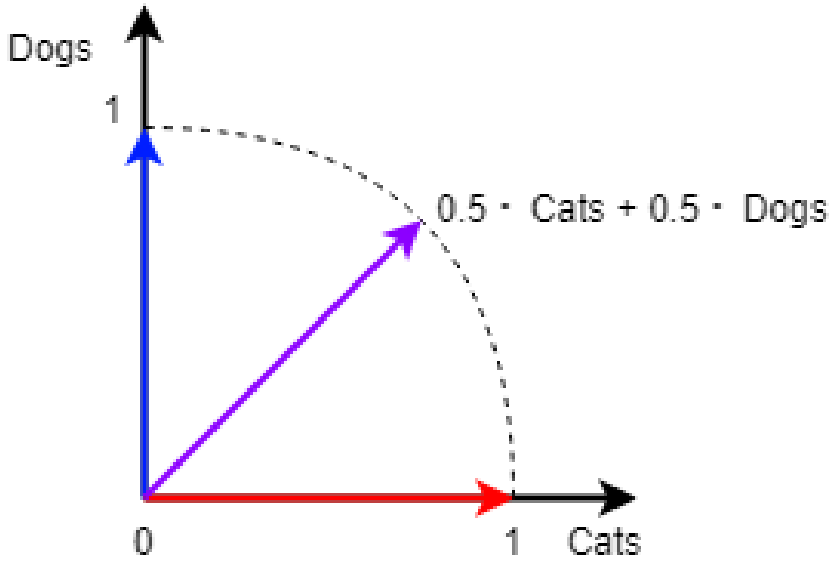


Рис. 1: Требуемое соотношение между моделями в векторном пространстве

## 2 Связанные работы

Существует большое количество работ, направленных на использование энкодера поверх других моделей. Но большинство из них используют дискриминативные модели. Например, энкодер и диффузия [5], кодирование модели NAS [1] или целый набор векторов из раздичных моделей [2].

Изучение представлений помогает понять закономерности в работе нейронных сетей. В [4] представлен метод SANE. Он обучает универсальные представления нейронных сетей, которые масштабируются для больших моделей различных архитектур и предназначенных для разных задач. Для решения SANE испрользует гиперпредставления для последовательной обработки подмножеств весов нейронной сети, что позволяет представить большие нейронные сети как набор токенов и перевести в пространство представления.

В данной работе большое внимание будет уделено кодированию модели с помощью сингулярных чисел её весов. Описание свойств сингулярных чисел в модели представлено в [3].

# 3 Вазимная информация модели и данных

Чтобы оценить возможность модели описывать данные, на которых она обучалась, необходимо оценить потери информации между исходными данными, весами модели при обучении подвыбоке и последующей векторизацией. Для оценки используется такая метрика, как совместная информация:

$$I(X;Y) = HY - H(Y \mid X).$$

**Теорема 1.** *$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$ – множество независимых векторов (пусть $\mathbf{x}_i \in \mathbb{R}^d$ и $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$).*

*$X_1 \in \mathbb{R}^{m \times d}$ – его подмножество, формируемое путём независимого включения каждого вектора $\mathbf{x}_i$ с вероятностью $p$.*

*AE – линейный автоэнкодер с весами $W \in \mathbb{R}^{k \times d}$. AE обучен на $X_1$, то есть $W = W(X_1)$.*

*Тогда имеем следующие попарные взаимные информации:*

*1.*
$$I(X; X_1) \approx np \cdot H(\mathbf{x}_i), \quad H(\mathbf{x}_i) = \frac{1}{2} \log\left((2\pi e)^d |\Sigma|\right);$$

*2.*
$$I(X_1; \mathbf{W}) = \frac{m}{2} \log \frac{|\boldsymbol{\Sigma}|}{|(\mathbf{I} - \mathbf{W}_2\mathbf{W}_1)\boldsymbol{\Sigma}(\mathbf{I} - \mathbf{W}_2\mathbf{W}_1)^T|}$$

*3.*
$$I(X_1; \mathbf{S}) \approx \frac{m}{4} \log\left((2\pi e)^k \prod_{i=1}^{k} \lambda_i^2\right).$$

*Доказательство.* 1. Обозначим индикаторы включения $Z_i \sim \text{Bernoulli}(p)$, тогда $X_1 = \{\mathbf{x}_i \mid Z_i = 1\}$.

Поскольку $X_1$ однозначно определяется $X$ и $Z = (Z_1, \ldots, Z_n)$, то:

$$H(X_1 \mid X) = H(Z \mid X) = H(Z),$$

так как $Z$ не зависит от $X$ (выбор подмножества случайный и независимый от значений векторов).

Таким образом:

$$I(X; X_1) = H(X_1) - H(Z).$$

Так как $Z_i$ независимы и $Z_i \sim \text{Bernoulli}(p)$, то:

$$H(Z) = \sum_{i=1}^{n} H(Z_i) = nH_{\text{bin}}(p),$$

где $H_{\text{bin}}(p) = -p \log p - (1-p) \log(1-p)$ – энтропия Бернулли.

Подмножество $X_1$ состоит из случайного числа $m = \sum_{i=1}^{n} Z_i$ векторов (где $m \sim \text{Binomial}(n, p)$), и для каждого фиксированного $m$ векторы в $X_1$ — это $m$ независимых векторов.

Таким образом, условная энтропия при фиксированном $m$:

$$H(X_1 \mid m) = m \cdot H(\mathbf{x}_i) = m \cdot \frac{1}{2} \log\left((2\pi e)^d |\Sigma|\right).$$

Тогда полная энтропия:

$$H(X_1) = \sum_{m=0}^{n} P(m) \cdot H(X_1 \mid m) + H(m),$$

где $H(m)$ — это энтропия биномиального распределения:

$$H(m) = H_{\text{binomial}}(n, p).$$

Тогда:

$$H(X_1) = H(m) + \mathbb{E}[H(X_1 \mid m)] = H_{\text{binomial}}(n, p) + \mathbb{E}[m] \cdot H(\mathbf{x}_i).$$

Поскольку $\mathbb{E}[m] = np$, то:

$$H(X_1) = H_{\text{binomial}}(n, p) + np \cdot \frac{1}{2} \log\left((2\pi e)^d |\Sigma|\right).$$

Для взаимной информации получаем:

$$I(X; X_1) = H_{\text{binomial}}(n, p) + np \cdot \frac{1}{2} \log\left((2\pi e)^d |\Sigma|\right) - nH_{\text{bin}}(p).$$

При $n \to \infty$: $H_{\text{binomial}}(n, p) \approx nH_{\text{bin}}(p)$. Тогда:

$$I(X; X_1) \approx np \cdot \frac{1}{2} \log\left((2\pi e)^d |\Sigma|\right).$$

2. Автоэнкодер (линейный, без активаций):

   Кодировщик: $\mathbf{z} = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1$

   Декодировщик: $\hat{\mathbf{x}} = \mathbf{W}_2\mathbf{z} + \mathbf{b}_2$

   Выход автоэнкодера:

$$\hat{\mathbf{x}} = \mathbf{W}_2\mathbf{W}_1\mathbf{x} + \mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2$$

   Обозначим $\mathbf{A} = \mathbf{W}_2\mathbf{W}_1$, $\mathbf{c} = \mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2$. Тогда:

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{c}$$

   Поскольку $\mathbf{x}$ гауссовский, а $\hat{\mathbf{x}}$ — его линейное преобразование, то:

$$\hat{\mathbf{x}} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}_x + \mathbf{c}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$$

   Ошибка восстановления:

$$\boldsymbol{\varepsilon} = \mathbf{x} - \hat{\mathbf{x}} = (\mathbf{I} - \mathbf{A})\mathbf{x} - \mathbf{c}$$

$$\boldsymbol{\varepsilon} \sim \mathcal{N}\left((\mathbf{I}-\mathbf{A})\boldsymbol{\mu}_x - \mathbf{c}, (\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}(\mathbf{I}-\mathbf{A})^T\right)$$

Взаимная информация между $\mathbf{x}$ и весами $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2)$:

$$I(\mathbf{x}; \mathbf{W}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{W})$$

Для гауссовского вектора:

$$H(\mathbf{x}) = \frac{1}{2}\log\left((2\pi e)^d|\boldsymbol{\Sigma}|\right)$$

Условное распределение $\mathbf{x}$ при фиксированных $\mathbf{W}$ определяется ошибкой $\boldsymbol{\varepsilon}$:

$$H(\mathbf{x}|\mathbf{W}) = H(\boldsymbol{\varepsilon}) = \frac{1}{2}\log\left((2\pi e)^d|(\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}(\mathbf{I}-\mathbf{A})^T|\right)$$

Подставляем $H(\mathbf{x})$ и $H(\mathbf{x}|\mathbf{W})$:

$$I(\mathbf{x}; \mathbf{W}) = \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}|}{|(\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}(\mathbf{I}-\mathbf{A})^T|}$$

где $\mathbf{A} = \mathbf{W}_2\mathbf{W}_1$.

Если автоэнкодер обучен как PCA (т.е. $\mathbf{W}_1 = \mathbf{U}^T$, $\mathbf{W}_2 = \mathbf{U}$, где $\mathbf{U}$ — матрица главных компонент), то:

$$\mathbf{A} = \mathbf{U}\mathbf{U}^T$$

Ковариация ошибки:

$$(\mathbf{I}-\mathbf{U}\mathbf{U}^T)\boldsymbol{\Sigma}(\mathbf{I}-\mathbf{U}\mathbf{U}^T)^T = \boldsymbol{\Sigma} - \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$$

где $\boldsymbol{\Lambda}$ — диагональная матрица собственных значений.

Тогда:

$$I(\mathbf{x};\mathbf{W}) = \frac{1}{2}\log\frac{|\mathbf{\Sigma}|}{|\mathbf{\Sigma} - \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T|}$$

Если $\mathbf{A} \approx \mathbf{I}$ (идеальное восстановление), то $|\mathbf{I}-\mathbf{A}| \approx 0$ и $I(\mathbf{x};\mathbf{W}) \to \infty$.

Если $\mathbf{A} \approx 0$ (автоэнкодер ничего не учит), то $I(\mathbf{x};\mathbf{W}) \approx 0$.

Для линейного автоэнкодера и гауссовских данных взаимная информация (насколько веса модели $\mathbf{W}$ уменьшают неопределённость в данных $\mathbf{x}$) вычисляется как:

$$I(\mathbf{x};\mathbf{W}) = \frac{1}{2}\log\frac{|\mathbf{\Sigma}|}{|(\mathbf{I} - \mathbf{W}_2\mathbf{W}_1)\mathbf{\Sigma}(\mathbf{I} - \mathbf{W}_2\mathbf{W}_1)^T|}$$

3. Рассмотрим линейный автоэнкодер с гауссовскими входными данными $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. Пусть веса кодировщика $\mathbf{W}_1 \in \mathbb{R}^{k\times d}$ и декодировщика $\mathbf{W}_2 \in \mathbb{R}^{d\times k}$ имеют сингулярные разложения:

$$\mathbf{W}_1 = \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^T, \quad \mathbf{W}_2 = \mathbf{U}_2\mathbf{S}_2\mathbf{V}_2^T,$$

где $\mathbf{S}_1$ и $\mathbf{S}_2$ — диагональные матрицы сингулярных чисел.

Сингулярные числа $\mathbf{s} = \{s_i\}$ весов $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2)$ зависят от ковариации входных данных $\mathbf{\Sigma}$, так как обучение автоэнкодера минимизирует:

$$\mathbb{E}\|\mathbf{x} - \mathbf{W}_2\mathbf{W}_1\mathbf{x}\|^2.$$

В оптимальном случае (аналог PCA), $\mathbf{W}_2\mathbf{W}_1$ соответствует проекции на главные компоненты $\mathbf{x}$, а сингулярные числа связаны с собственными значениями $\mathbf{\Sigma}_x$.

Взаимная информация между $\mathbf{x}$ и сингулярными числами $\mathbf{s}$:

$$I(\mathbf{x};\mathbf{s}) = H(\mathbf{s}) - H(\mathbf{s}|\mathbf{x}).$$

Для PCA-like автоэнкодера, $\mathbf{s}$ — это корни собственных значений $\mathbf{\Sigma}$. Если $\mathbf{\Sigma}_x$ имеет спектр $\{\lambda_i\}$, то $s_i \approx \sqrt{\lambda_i}$. Тогда энтропия:

$$H(\mathbf{s}) = \frac{1}{2} \log \left( (2\pi e)^k \prod_{i=1}^{k} \mathrm{Var}(s_i) \right).$$

При фиксированных $\mathbf{x}$, сингулярные числа $\mathbf{s}$ детерминированы (так как $\mathbf{W}$ обучаются на данных). Поэтому $H(\mathbf{s}|\mathbf{x}) = 0$.

Таким образом:

$$I(\mathbf{x}; \mathbf{s}) = H(\mathbf{s}) = \frac{1}{2} \log \left( (2\pi e)^k \prod_{i=1}^{k} \mathrm{Var}(s_i) \right).$$

Если автоэнкодер близок к PCA, то $s_i \approx \sqrt{\lambda_i}$, где $\lambda_i$ – собственные значения $\mathbf{\Sigma}$. Тогда:

$$I(\mathbf{x}; \mathbf{s}) \approx \frac{1}{4} \log \left( (2\pi e)^k \prod_{i=1}^{k} \lambda_i^2 \right).$$

Чем больше дисперсия сингулярных чисел $\mathbf{s}$, тем выше $I(\mathbf{x}; \mathbf{s})$.

Если $\mathbf{s}$ не зависят от $\mathbf{x}$ (например, случайные веса), то $I(\mathbf{x}; \mathbf{s}) = 0$.

□

# 4 Вычислительные эксперименты

## 4.1 Бинарная классификация

Если нужно создать вектор модели с требуемым свойством, то сначала следует проверить, насколько хорошо вектор модели описывает данные, используемые для обучения. Самый простой вариант: обучить $N$ автоэнкодеров на семплах из двух классов датасета, перевести их в векторы и

затем построить энкодер, определяющий, на данных из какого класса была обучена модель (архитектура представлена на рис. 2).

Чтобы не допустить переобучения классификатора и не завязываться на размерности, энкодеры векторизуются:

1. Сингулярные числа весов

2. Гистограмма значений весов (рис. 3)

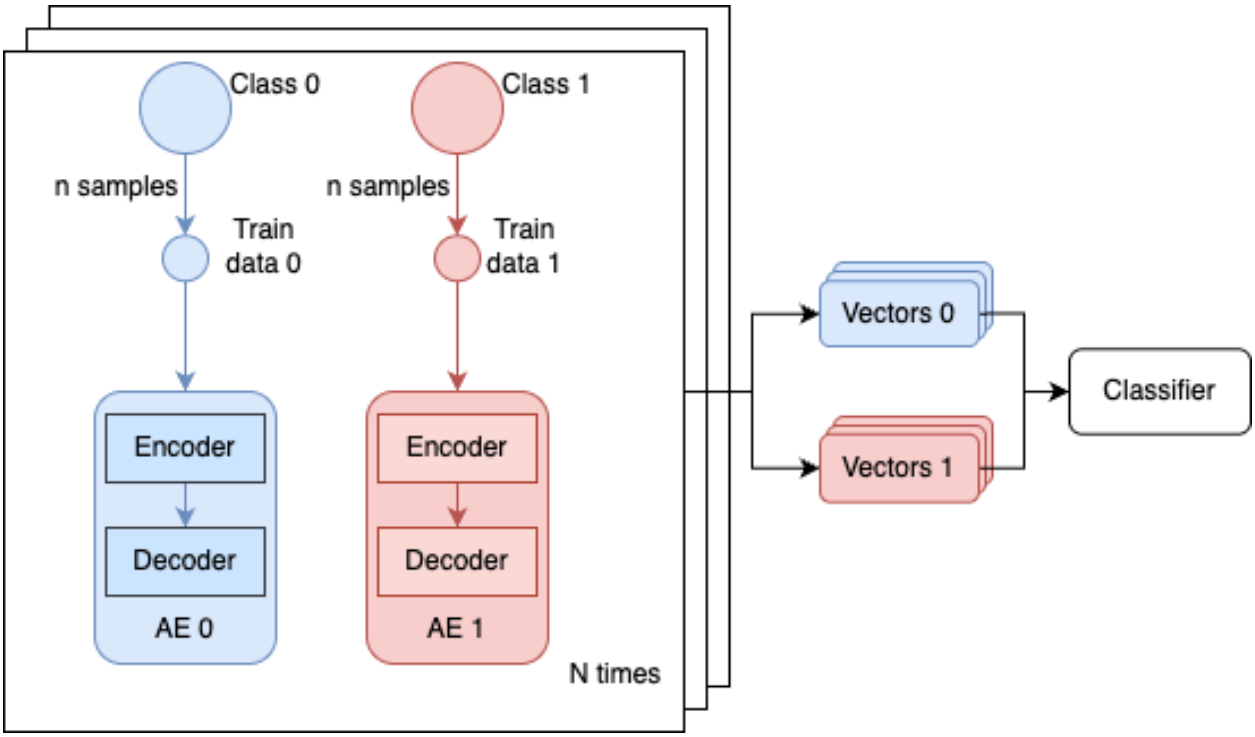


Рис. 2: Архитектура эксперимента с бинарной классификацией

Результаты эксперимента представлены в таблице 1. Можно заметить, что автоэнкодеры с линейным слоем, обученные на разных датасетах, отличить друг от друга легче, чем автоэнкодеры со свёрточными слоями. То есть, чем сложнее модель, тем больше информации теряется о ней и датасете при обучении и векторизации.

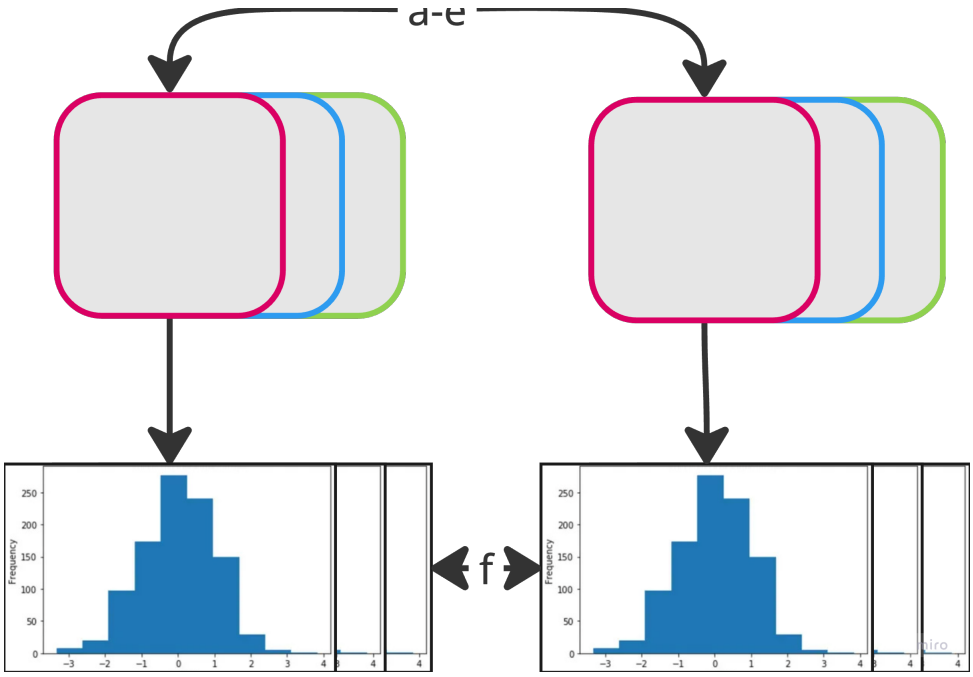


Рис. 3: Векторизация весов модели с помощью гистограмм

Таблица 1: Метрики качества предсказания класса, на котором обучалась модель

| | Precision | Recall |
|---|---|---|
| 1 линейный слой | 0.79 | 1.00 |
| 1 свёрточный слой | 0.55 | 0.78 |

## 4.2 Предсказане вектора в пространстве датасетов-моделей

Для базового эксперимента берётся 3 наиболее удалённых класса из датасета CIFAR. Для поиска таких классов использовалось евклидово расстояние на ембеддингах, полученных из выходного слоя ResNet. Архитектура решения представлена на рис 4.

1. Случайное сэмплирование долей классов в датасете для $N$ моделей;
2. Обучение $N$ моделей на соответствующих датасетах;
3. Получение векторов из обученных моделей;

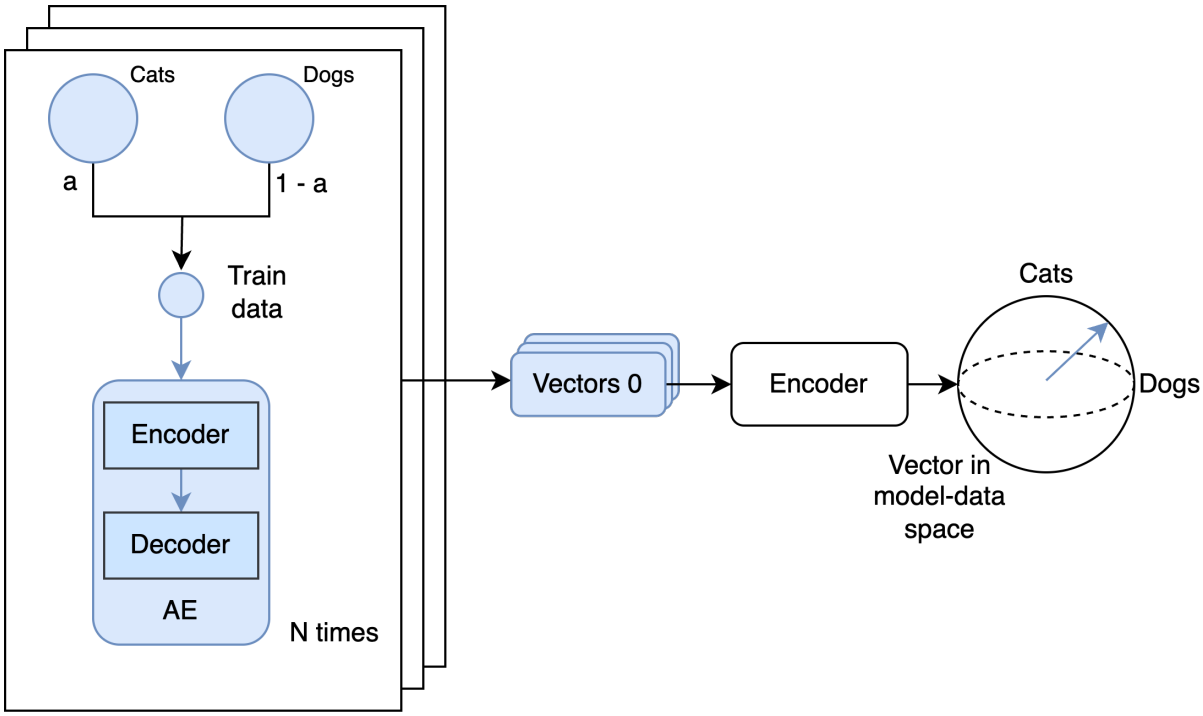


Рис. 4: Архитектура эксперимента с предсказанием вектора в пространстве датасетов-моделей

4. Обучение энкодера на полученных моделях. Предсказание:

   (a) Вектора на части единичной сферы

   (b) Расстояния между двумя моделями

Функции потерь для реализации обучения:

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

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

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:

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

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

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

# Список литературы

[1] Yash Akhauri and Mohamed S. Abdelfattah. Encodings for prediction-based neural architecture search, 2024.

[2] Wonyong Jeong, Hayeon Lee, Gun Park, Eunyoung Hyung, Jinheon Baek, and Sung Ju Hwang. Task-adaptive neural network search with meta-contrastive learning, 2021.

[3] Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning, 2018.

[4] Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning, 2024.

[5] Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion, 2024.