

Deep Anomaly Detection Using Geometric Transformations

Павел Латышев 151

Задача поиска аномалий

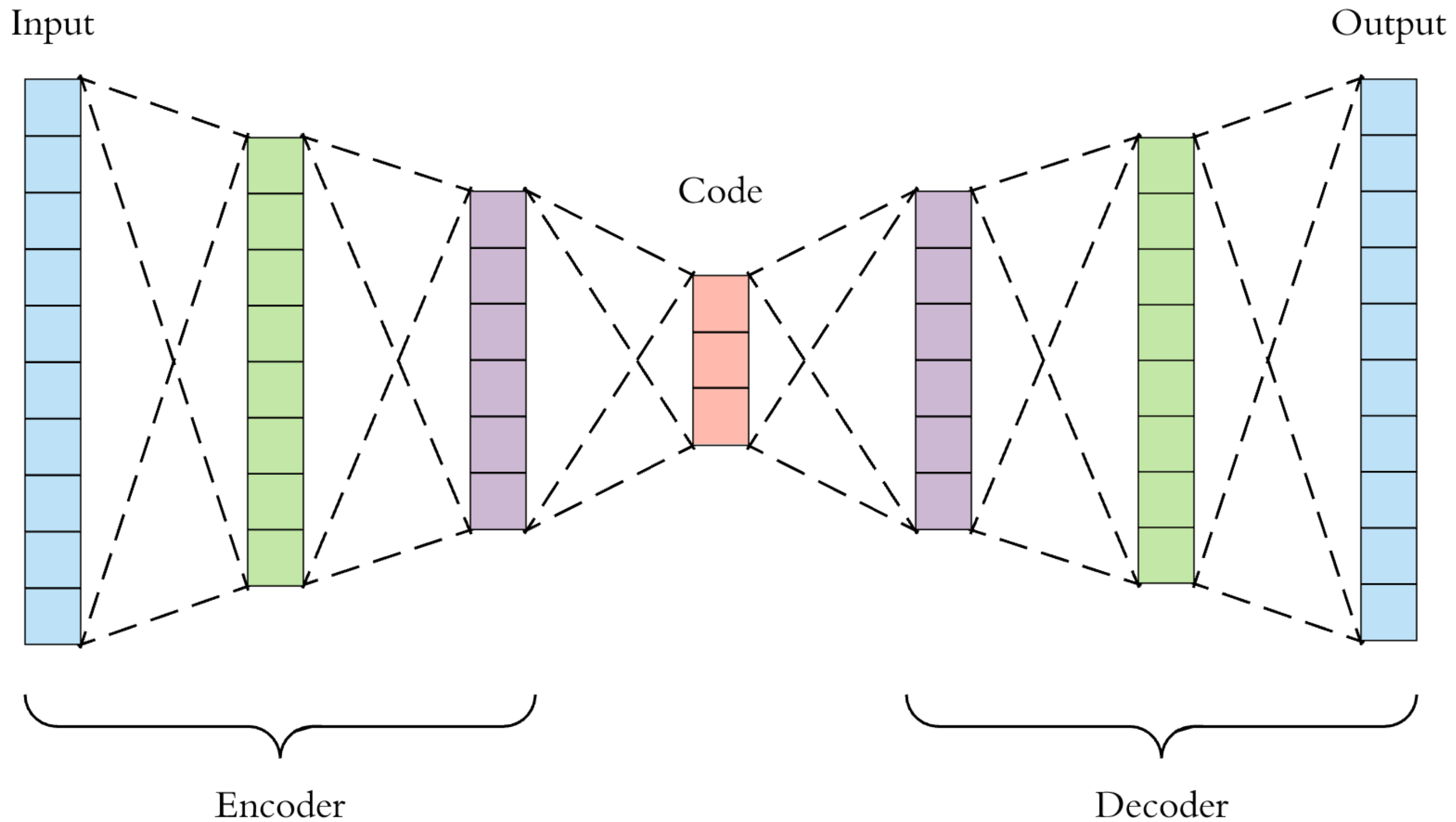
Дана выборка объектов из нормального класса, необходимо обучить классификатор, находящий аномалии, то есть объекты, не принадлежащие нормальному классу.



Существующие подходы

- One Class SVM на картинках
- One Class SVM на латентном представлении автоэнкодера
- Оценка качества реконструкции картинки автоэнкодером или GAN

Autoencoder



ADGAN

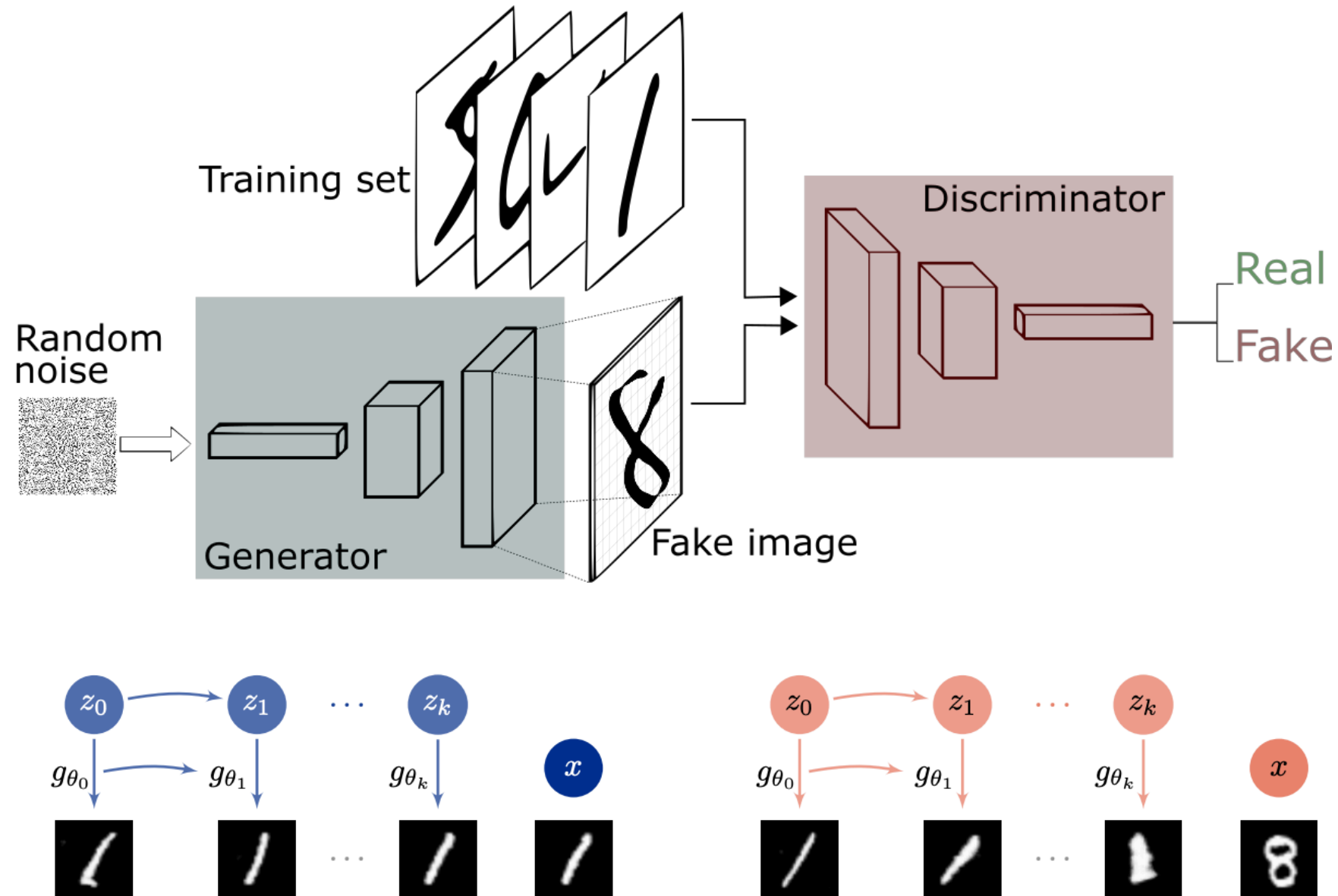


Figure 1: An illustration of ADGAN. In this example, ones from MNIST are considered normal. After an initial draw from p_z , the loss between the first generation $g_{\theta_0}(z_0)$ and the image x whose anomaly we are assessing is computed. This information is used to generate a consecutive image $g_{\theta_1}(z_1)$ more alike x . After k steps, samples are scored. If x is similar to the training data (blue example), then a similar object should be contained in the image of g_{θ_k} . For a dissimilar x (red example), no similar image is found, resulting in a large loss.

Постановка задачи

\mathcal{X} - пространство изображений

$X \subseteq \mathcal{X}$ - множество нормальных изображений

$S \subseteq X$ - обучающая выборка

$$h_S(x) : \mathcal{X} \rightarrow \{0,1\}, \quad h_S(x) = 1 \Leftrightarrow x \in X$$

$$h_S^\lambda(x) = \begin{cases} 1 & n_S(x) \geq \lambda \\ 0 & n_S(x) < \lambda \end{cases}$$

Предложенный метод

- Выберем множество преобразований T (K комбинаций из горизонтального отражения, переноса и поворота)
- Применим преобразования к датасету
- Обучим классификатор определять тип преобразования
- Сумма вероятностей каждого класса или Dirichlet normality score - ответ искомого классификатора n

Построение датасета

$\mathcal{T} = \{T_0, T_1, \dots, T_{k-1}\}$ - набор геометрических преобразований

$$S_{\mathcal{T}} = \{(T_j(x), j) : x \in S, T_j \in \mathcal{T}\}$$

На этом датасете обучаем k-классовый классификатор

$$f_{\theta} : \mathcal{R} \rightarrow \mathbb{R}^k$$

Упрощенный normality score

Пусть $y(x) = \text{softmax}(f_\theta(x))$

$$\hat{n}_S = \frac{1}{k} \sum_{j=0}^{k-1} [y(T_j(x))]_j$$

- Легко реализуется
- Быстро вычисляется
- Работает чуть хуже, чем Dirichlet normality score

Dirichlet normality score

$$n_s(x) = \sum_{i=0}^{k-1} \log p(y(T_i(x)) | T_i)$$

Приблизим

$$y(T_i(x)) | T_i \sim \text{Dir}(a_i), \quad a_i \in \mathbb{R}_+^k, x \sim p_X(x), i \sim \mathbf{Uni}(0, k-1)$$

$$n_S(x) = \sum_{i=0}^{k-1} \left[\log \Gamma\left(\sum_{j=0}^{k-1} [\hat{a}_i]_j\right) - \sum_{j=0}^{k-1} \log \Gamma([\hat{a}_i]_j) + \sum_{j=0}^{k-1} ([\hat{a}_i]_j - 1) \log y(T_i(x))_j \right]$$

$$n_S(x) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} ([\hat{a}_i]_j - 1) \log y(T_i(x))_j$$

Алгоритм

Algorithm 1 Deep Anomaly Detection Using Geometric Transformations

Input: S : a set of “normal” images. $\mathcal{T} = \{T_0, T_1, \dots, T_{k-1}\}$: a set of geometric transformations.
 f_θ : a softmax classifier parametrized by θ .

Output: A normality scoring function $n_S(x)$.

```
1: procedure GETNORMALITYSCORE( $S, \mathcal{T}, f_\theta$ )
2:    $S_{\mathcal{T}} \leftarrow \{(T_j(x), j) : x \in S, T_j \in \mathcal{T}\}$ 
3:   while not converged do
4:     Train  $f_\theta$  on the labeled set  $S_{\mathcal{T}}$ 
5:   end while
6:    $n \leftarrow |S|$ 
7:   for  $i \in \{0, \dots, k-1\}$  do
8:      $S_i \leftarrow \{\mathbf{y}(T_i(x)) | x \in S\}$ 
9:      $\bar{\mathbf{s}} \leftarrow \frac{1}{n} \sum_{\mathbf{s} \in S_i} \mathbf{s}$ 
10:     $\bar{l} \leftarrow \frac{1}{n} \sum_{\mathbf{s} \in S_i} \log \mathbf{s}$ 
11:     $\tilde{\alpha}_i \leftarrow \bar{\mathbf{s}} \frac{(k-1)(-\Psi(1))}{\bar{\mathbf{s}} \cdot \log \bar{\mathbf{s}} - \bar{\mathbf{s}} \cdot \bar{l}}$  ▷ Initialization from [31]
12:    while not converged do
13:       $\tilde{\alpha}_i \leftarrow \Psi^{-1} \left( \Psi \left( \sum_j [\alpha_i]_j \right) + \bar{l} \right)$  ▷ Fixed point method from [21]
14:    end while
15:  end for
16:  return  $n_S(x) \triangleq \sum_{i=0}^{k-1} (\tilde{\alpha}_i - 1) \cdot \log \mathbf{y}(T_i(x))$ 
17: end procedure
```

Выбранные трансформации

- Горизонтальное отражение
- Перенос на -0.25, 0, 0.25 по вертикали и горизонтали
- Поворот на 0, 90, 180, 270 градусов по часовой стрелке

Всего $2 * 3 * 3 * 4 = 72$ комбинации

Пробовали другие типы трансформаций: гауссово размытие, гамма-коррекцию, качество ухудшалось

Оптимизация изображения под нормальный класс

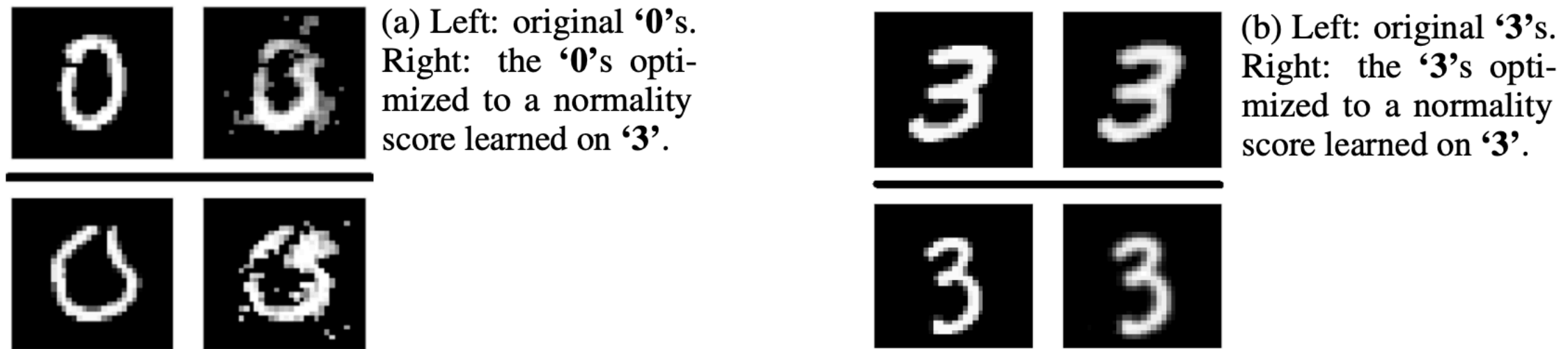


Figure 1: Optimizing digit images to maximize the normality score

Результаты

Dataset	c_i	RAW	OC-SVM		DAGMM	DSEBM	AD-GAN	OURS
			CAE	E2E				
CIFAR-10 (32x32x3)	0	70.6	74.9	61.7±1.3	41.4±2.3	56.0±6.9	64.9	74.7±0.4
	1	51.3	51.7	65.9±0.7	57.1±2.0	48.3±1.8	39.0	95.7±0.0
	2	69.1	68.9	50.8±0.3	53.8±4.0	61.9±0.1	65.2	78.1±0.4
	3	52.4	52.8	59.1±0.4	51.2±0.8	50.1±0.4	48.1	72.4±0.5
	4	77.3	76.7	60.9±0.3	52.2±7.3	73.3±0.2	73.5	87.8±0.2
	5	51.2	52.9	65.7±0.8	49.3±3.6	60.5±0.3	47.6	87.8±0.1
	6	74.1	70.9	67.7±0.8	64.9±1.7	68.4±0.3	62.3	83.4±0.5
	7	52.6	53.1	67.3±0.3	55.3±0.8	53.3±0.7	48.7	95.5±0.1
	8	70.9	71.0	75.9±0.4	51.9±2.4	73.9±0.3	66.0	93.3±0.0
	9	50.6	50.6	73.1±0.4	54.2±5.8	63.6±3.1	37.8	91.3±0.1
<i>avg</i>		62.0	62.4	64.8	53.1	60.9	55.3	86.0
Fashion-MNIST (32x32x1)	0	98.2	97.7	-	42.1±9.1	91.6±1.2	89.9	99.4±0.0
	1	90.3	89.9	-	55.1±3.5	71.8±0.5	81.9	97.6±0.1
	2	90.7	91.4	-	50.4±7.3	88.3±0.2	87.6	91.1±0.2
	3	94.2	90.7	-	57.0±6.7	87.3±3.6	91.2	89.9±0.4
	4	89.4	89.1	-	26.9±5.4	85.2±0.9	86.5	92.1±0.0
	5	91.8	88.5	-	70.5±9.7	87.1±0.0	89.6	93.4±0.9
	6	83.4	81.7	-	48.3±5.0	73.4±4.1	74.3	83.3±0.1
	7	98.8	98.7	-	83.5±11.4	98.1±0.0	97.2	98.9±0.1
	8	91.9	90.6	-	49.9±7.2	86.0±3.2	89.0	90.8±0.1
	9	99.0	98.6	-	34.0±3.0	97.1±0.3	97.1	99.2±0.0
<i>avg</i>		92.8	91.7	-	51.8	86.6	88.4	93.5
CatsVsDogs (64x64x3)	0	50.4	55.2	-	43.4±0.5	47.1±1.7	50.7	88.3±0.3
	1	53.0	49.9	-	52.0±1.9	56.1±1.2	48.1	89.2±0.3
	<i>avg</i>	51.7	52.5	-	47.7	51.6	49.4	88.8

Литература

- Deep Anomaly Detection Using Geometric Transformations <https://arxiv.org/abs/1805.10917>
- Anomaly Detection with Generative Adversarial Networks <https://openreview.net/forum?id=S1EfyIZ0Z>
- <https://skymind.ai/wiki/generative-adversarial-network-gan>
- <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>