A Simple Framework for Contrastive Learning of Visual Representations [1]

Петров Егор

МФТИ

23 марта 2024 г.

[1] основано на оригинальной статье

План

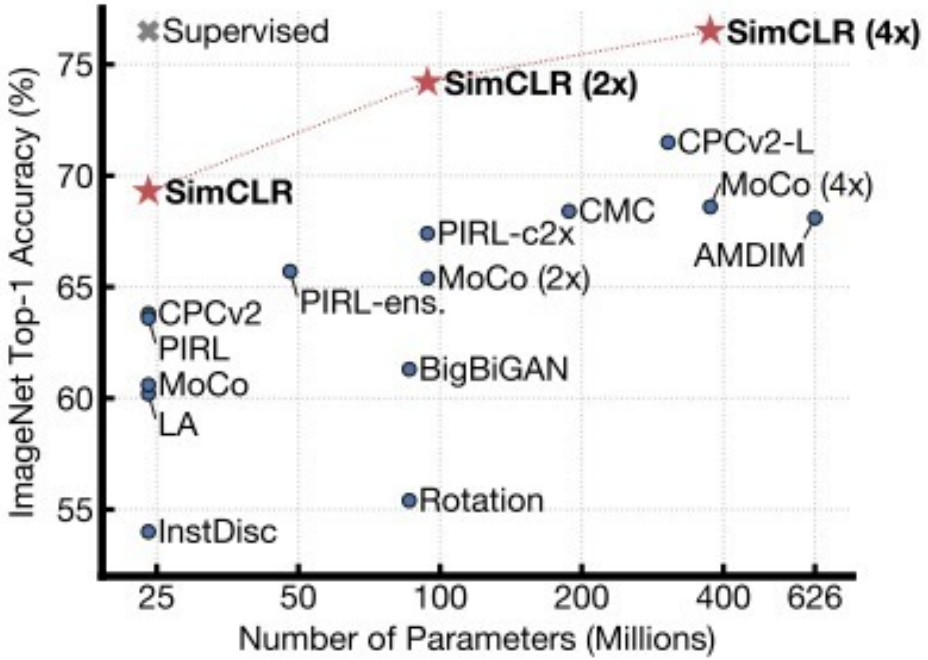
Вступление

Вступление

- В статье предоставляется $SimCLR$ - простая модель для создания эффективных визуальных представлений

Вступление

- В статье предоставляется $SimCLR$ - простая модель для создания эффективных визуальных представлений



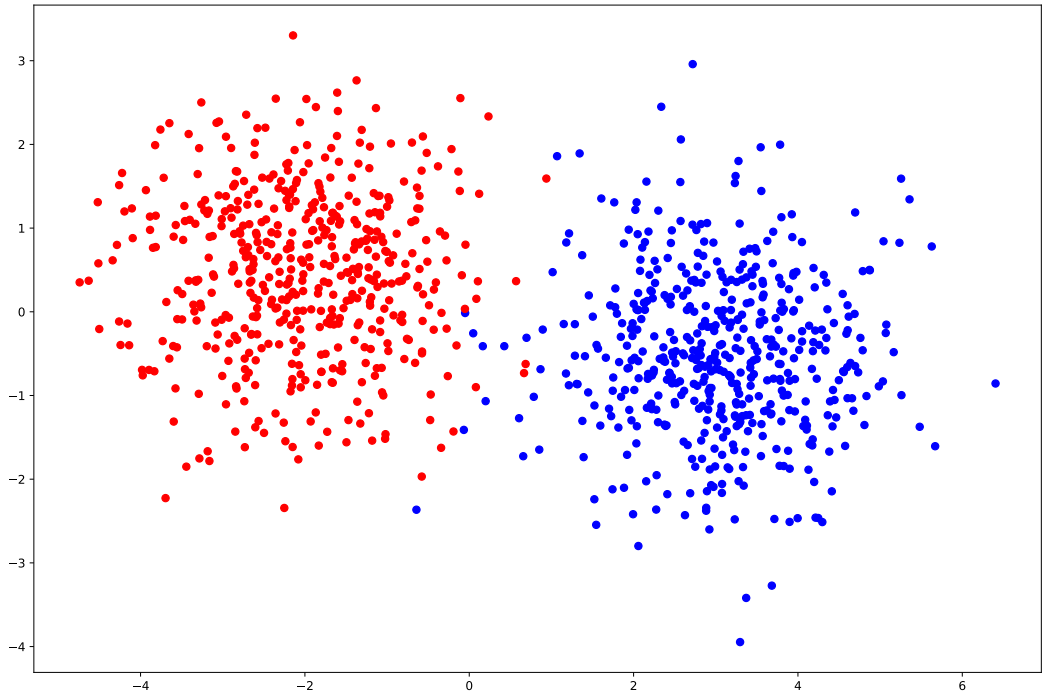
Основная задача

Основная задача

Основной задачей является получение векторного представления картинки по неразмеченным данным, то есть $image \to (x_1, x_2, ..., x_n)$

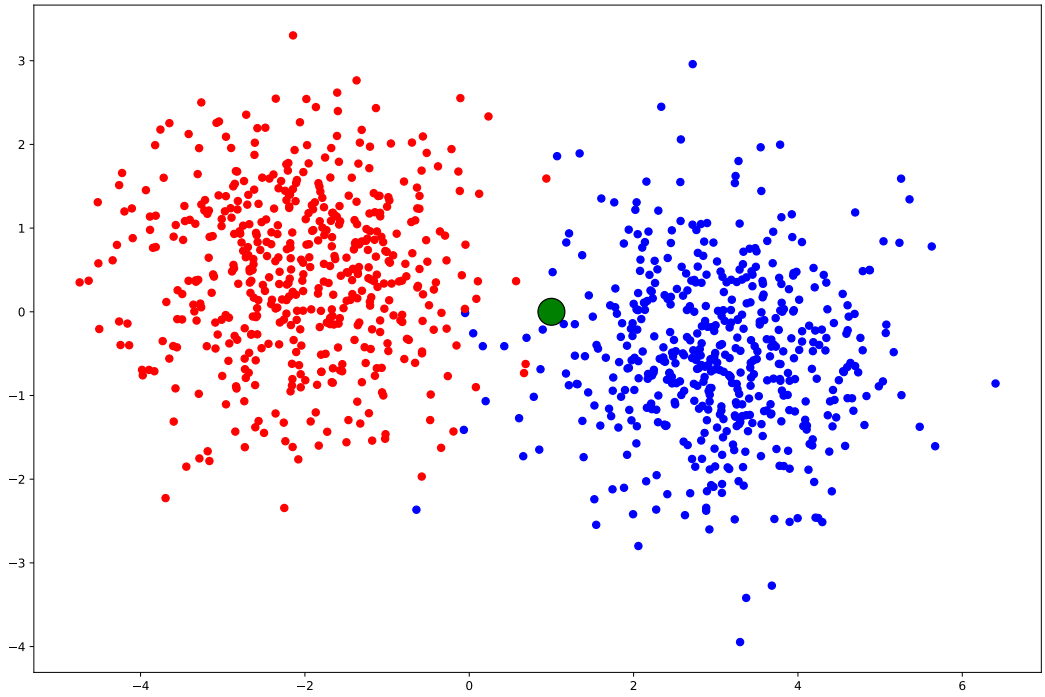
Основная задача

Основной задачей является получение векторного представления картинки по неразмеченным данным, то есть $image \to (x_1, x_2, ..., x_n)$



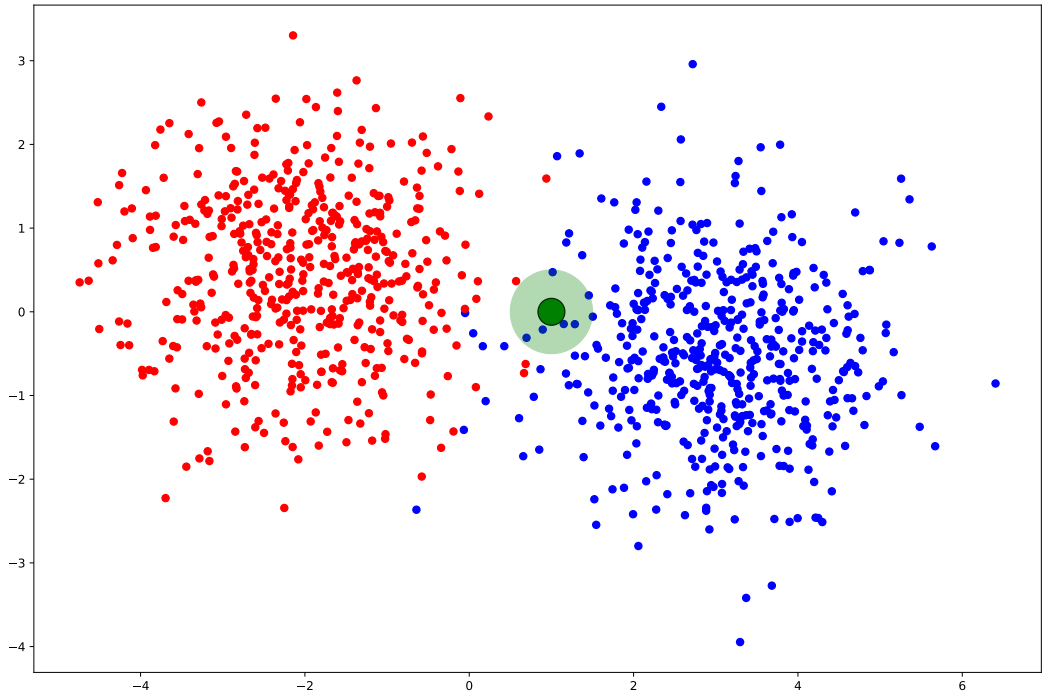
Основная задача

Основной задачей является получение векторного представления картинки по неразмеченным данным, то есть $image \to (x_1, x_2, ..., x_n)$



Основная задача

Основной задачей является получение векторного представления картинки по неразмеченным данным, то есть $image \to (x_1, x_2, ..., x_n)$



Способ решения

Способ решения

Unsupervised learning

Unsupervised learning (или *Обучение без учителя*) — это метод машинного обучения, при котором, в отличие от обучения с учителем, алгоритмы изучают закономерности исключительно на основе неразмеченных данных.

Способ решения

Unsupervised learning

Unsupervised learning (или *Обучение без учителя*) — это метод машинного обучения, при котором, в отличие от обучения с учителем, алгоритмы изучают закономерности исключительно на основе неразмеченных данных.

Contrastive learning

Contrastive learning (или *Контрастное обучение*) — это метод глубокого обучения для обучения представлению (representation) без учителя. Цель состоит в том, чтобы изучить представление данных таким образом, чтобы похожие экземпляры находились близко друг к другу в пространстве представления, а разнородные — далеко друг от друга.

Фреймворк для контрастного обучения

Фреймворк для контрасntого обучения

Представляемый в данной статье фреймворк состоит из следующих основных частей

Фреймворк для контраснтого обучения

Представляемый в данной статье фреймворк состоит из следующих основных частей

- Модуль стохастической аугментации данных

Фреймворк для контрастного обучения

Представляемый в данной статье фреймворк состоит из следующих основных частей

- Модуль стохастической аугментации данных
- Основанный на нейронной сети энкодер $f(\cdot)$

Фреймворк для контрастного обучения

Представляемый в данной статье фреймворк состоит из следующих основных частей

- Модуль стохастической аугментации данных
- Основанный на нейронной сети энкодер $f(\cdot)$
- Небольшая нейронная сеть, проекция - *projection head* $g(\cdot)$

Фреймворк для контрасntого обучения

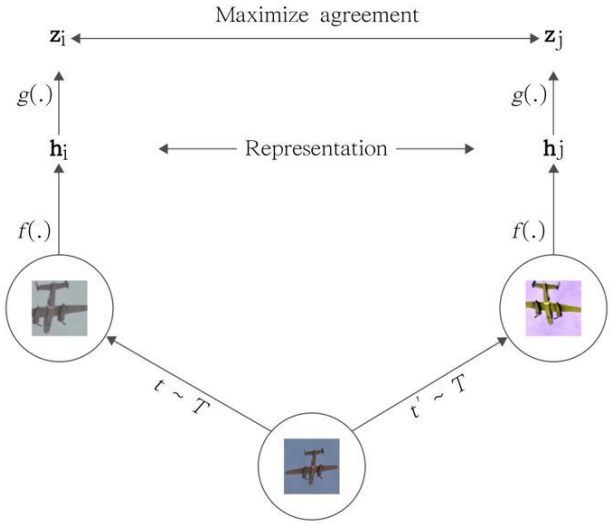
Представляемый в данной статье фреймворк состоит из следующих основных частей

- Модуль стохастической аугментации данных
- Основанный на нейронной сети энкодер $f(\cdot)$
- Небольшая нейронная сеть, проекция - *projection head* $g(\cdot)$
- Функция потерь для контрастного обучения

Фреймворк для контрастного обучения

Представляемый в данной статье фреймворк состоит из следующих основных частей

- Модуль стохастической аугментации данных
- Основанный на нейронной сети энкодер $f(\cdot)$
- Небольшая нейронная сеть, проекция - *projection head* $g(\cdot)$
- Функция потерь для контрастного обучения



Алгоритм

Algorithm 1 SimCLR's main learning algorithm.

Input: batch size N, constant τ, structure of f, g, T.

for sampled minibatch $\{x_k\}_{k=1}^{N}$ **do**

for all $k \in \{1, ..., N\}$ **do**

draw two augmentation functions $t \sim T$, $t' \sim T$

\# the first augmentation

$\tilde{x}_{2k-1} \leftarrow t(x_k)$

$h_{2k-1} \leftarrow f(\tilde{x}_{2k-1})$ ▷ representation

$z_{2k-1} \leftarrow g(h_{2k1})$ ▷ projection

\# the second augmentation

$\tilde{x}_{2k} \leftarrow t'(x_k)$

$h_{2k} \leftarrow f(\tilde{x}_{2k})$ ▷ representation

$z_{2k} \leftarrow g(h_{2k})$ ▷ projection

end for

for all $i \in \{1, ..., 2N\}$ and $j \in \{1, ..., 2N\}$ **do**

$s_{i,j} = z_i^T \cdot z_j / \|z_i\| \|z_j\|$ ▷ pairwise similarity

end for

define $l(i, j)$ as $l(i, j) = -\log(\frac{exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} exp(s_{i,j}/\tau)})$

$L = \frac{1}{2N} \sum_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)]$

update networks f and g to minimize L

end for

return encoder network $f(\cdot)$, and throw away $g(\cdot)$

Аугментация данных

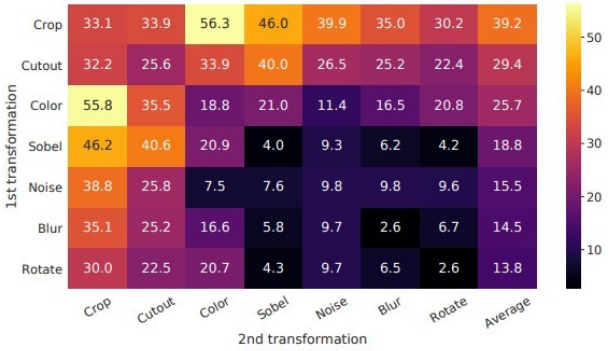
Аугментация данных

(a) Композиции аугментаций

Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

(b) Сравнение с supervised методами

Аугментация данных



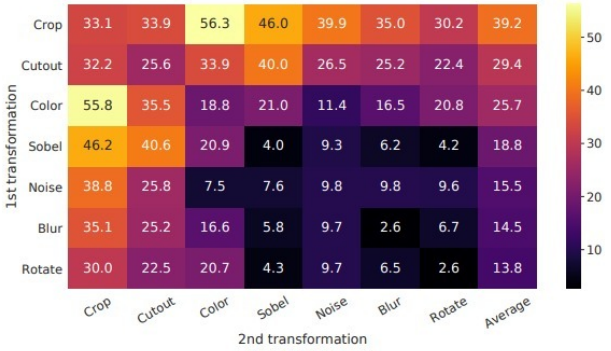
(a) Композиции аугментаций

Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

(b) Сравнение с supervised методами

- Композиция аугментации данных имеет решающее значение для изучения хороших представлений

Аугментация данных



(a) Композиции аугментаций

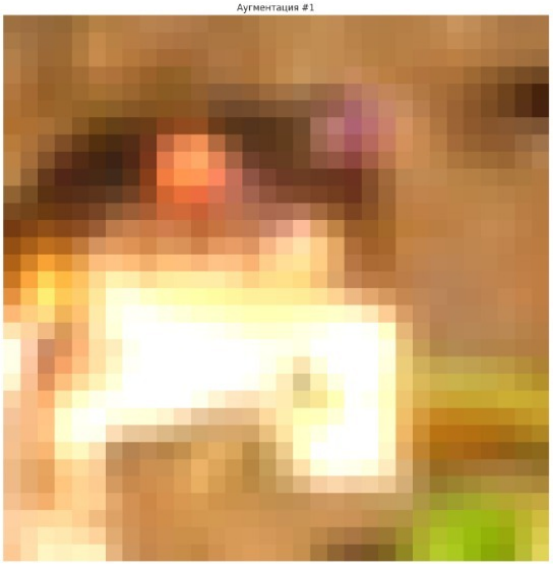
Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

(b) Сравнение с supervised методами

- Композиция аугментации данных имеет решающее значение для изучения хороших представлений
- Контрастное обучение требует более сильной аугментации данных, чем обучение с учителем

Аугментация данных

Пример применяемых аугментаций



Архитектура encoder и projection head

Архитектура encoder и projection head

- Контрастное обучение без учителя получает больше пользы от больших моделей, чем их supervised аналоги

Архитектура encoder и projection head

- Контрастное обучение без учителя получает больше пользы от больших моделей, чем их supervised аналоги
- Нелинейная projection head улучшает качество представления предыдущего слоя

Архитектура encoder и projection head

- Контрастное обучение без учителя получает больше пользы от больших моделей, чем их supervised аналоги
- Нелинейная projection head улучшает качество представления предыдущего слоя

Projection head

Нелинейная *projection head* - простая нейронная сеть с одним скрытым слоем $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$, где σ - функция, дающая нелинейность, в данном случае ReLU

Архитектура encoder и projection head

- Контрастное обучение без учителя получает больше пользы от больших моделей, чем их supervised аналоги
- Нелинейная projection head улучшает качество представления предыдущего слоя

Projection head

Нелинейная *projection head* - простая нейронная сеть с одним скрытым слоем $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$, где σ - функция, дающая нелинейность, в данном случае ReLU

- Нелинейная проекция лучше линейной (+3%), и это намного лучше, чем отсутствие таковой (>10%).

Архитектура encoder и projection head

- Контрастное обучение без учителя получает больше пользы от больших моделей, чем их supervised аналоги
- Нелинейная projection head улучшает качество представления предыдущего слоя

Projection head

Нелинейная *projection head* - простая нейронная сеть с одним скрытым слоем $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$, где σ - функция, дающая нелинейность, в данном случае ReLU

- Нелинейная проекция лучше линейной (+3%), и это намного лучше, чем отсутствие таковой (>10%).
- Само представление данных необхожимо смотреть по слою до проекции слой z_i содержит на (>10%) меньше информации, чем предыдущий слой h_i

Размер батча и функция потерь

Размер батча и функция потерь

- Контрастное обучение получает больше пользы от большего размера батча и количества эпох, по сравнению с supervised моделями

Размер батча и функция потерь

- Контрастное обучение получает больше пользы от большего размера батча и количества эпох, по сравнению с supervised моделями
- В работе используется NT-Xent (Normalized Temperature-scaled Cross Entropy), сравниевается с лоссами, описанными в предыдущих работах (Margin, NT-Logistic)

Размер батча и функция потерь

- Контрастное обучение получает больше пользы от большего размера батча и количества эпох, по сравнению с supervised моделями
- В работе используется NT-Xent (Normalized Temperature-scaled Cross Entropy), сравниевается с лоссами, описанными в предыдущих работах (Margin, NT-Logistic)
- NT-Xent $= u^T v^+/\tau - \log(\sum_{v \in \{v^+, v^-\}} exp(u^T v/\tau))$

Размер батча и функция потерь

- Контрастное обучение получает больше пользы от большего размера батча и количества эпох, по сравнению с supervised моделями
- В работе используется NT-Xent (Normalized Temperature-scaled Cross Entropy), сравниевается с лоссами, описанными в предыдущих работах (Margin, NT-Logistic)
- NT-Xent $= u^T v^+/\tau - \log(\sum_{v \in \{v^+, v^-\}} exp(u^T v/\tau))$

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

Размер батча и функция потерь

- Контрастное обучение получает больше пользы от большего размера батча и количества эпох, по сравнению с supervised моделями
- В работе используется NT-Xent (Normalized Temperature-scaled Cross Entropy), сравниевается с лоссами, описанными в предыдущих работах (Margin, NT-Logistic)
- NT-Xent $= u^T v^+/\tau - \log(\sum_{v \in \{v^+, v^-\}} exp(u^T v/\tau))$

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

- NT-Xent loss с изменяемой температурой работает лучше аналогов (Margin просто максимизирует разницу между позитивным и негативным примерами, в NT-Logostic суммируются логарифмы соответсвующих значений сигмоид)

Функция потерь

Функция потерь

Рассмотрим реализации предоставленной функции потерь

Функция потерь

Рассмотрим реализации предоставленной функции потерь

```python
# Контрастная функция потерь
def compute_loss(model, images, tmp=0.5):
    # Создаем аугментации
    images_aug1 = augment_fn(images)
    images_aug2 = augment_fn(images)

    # Передаем картинки в модель
    z1 = model(images_aug1)
    z2 = model(images_aug2)

    # Считаем похожесть пар
    sim_pairs = F.cosine_similarity(z1[None,:,:], z2[:,None,:], dim=-1)

    # Ставим контрастные метки
    target = torch.arange(sim_pairs.shape[0]).to(device)

    # Считаем loss
    loss = F.cross_entropy(sim_pairs / tmp, target, reduction="mean")

    return loss
```

Оценка качества моделей

Оценка качества моделей

Существует два основных способа оценить качество моделей

Оценка качества моделей

Существует два основных способа оценить качество моделей

- Linear evaluation

Linear evaluation

Linear evaluation — это метод оценки качества модели, при котором сама модель замораживается, а обучется только классифицирующая часть

Оценка качества моделей

Существует два основных способа оценить качество моделей

- Linear evaluation

Linear evaluation

Linear evaluation — это метод оценки качества модели, при котором сама модель замораживается, а обучется только классифицирующая часть

- Fine tuning

Fine tuning

Fine tuning — это метод оценки качества модели, при котором сама модель *не* замораживается и обучается на новых данных одновременно с классификатором

Сравнение результатов

Сравнение результатов

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Сравнение результатов

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

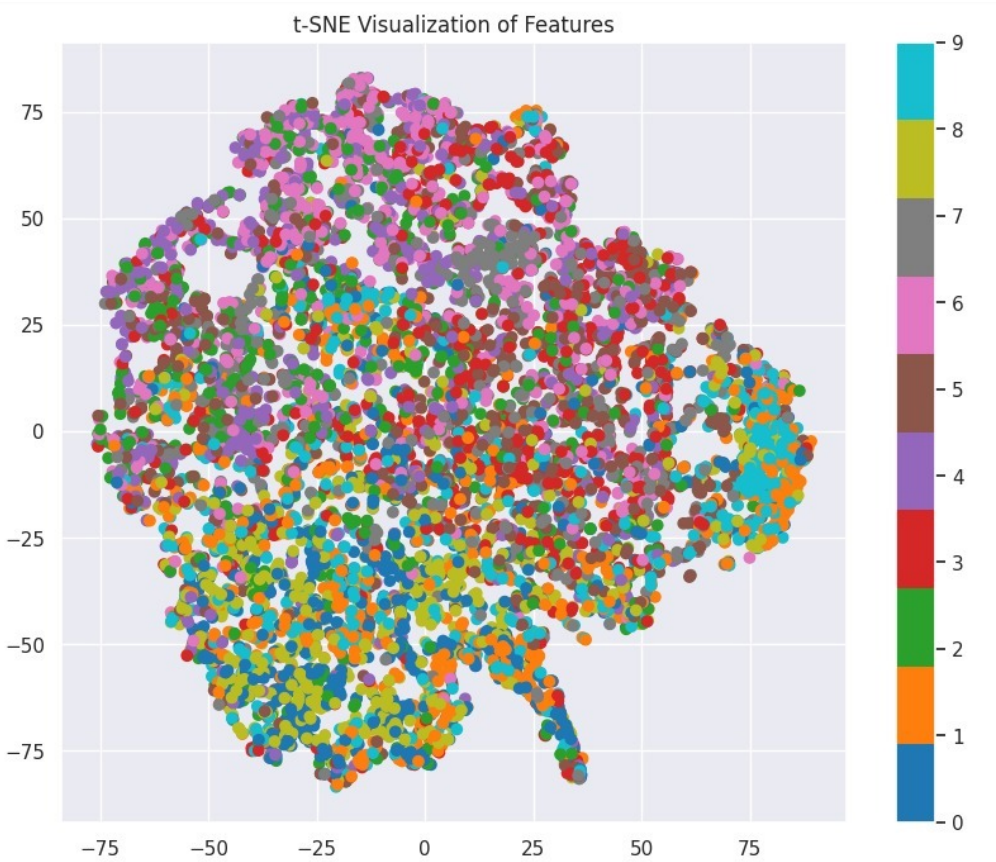
Пример работы

Пример работы

Посмотрим на результат работы для небольшого числа эпох

Пример работы

Посмотрим на результат работы для небольшого числа эпох



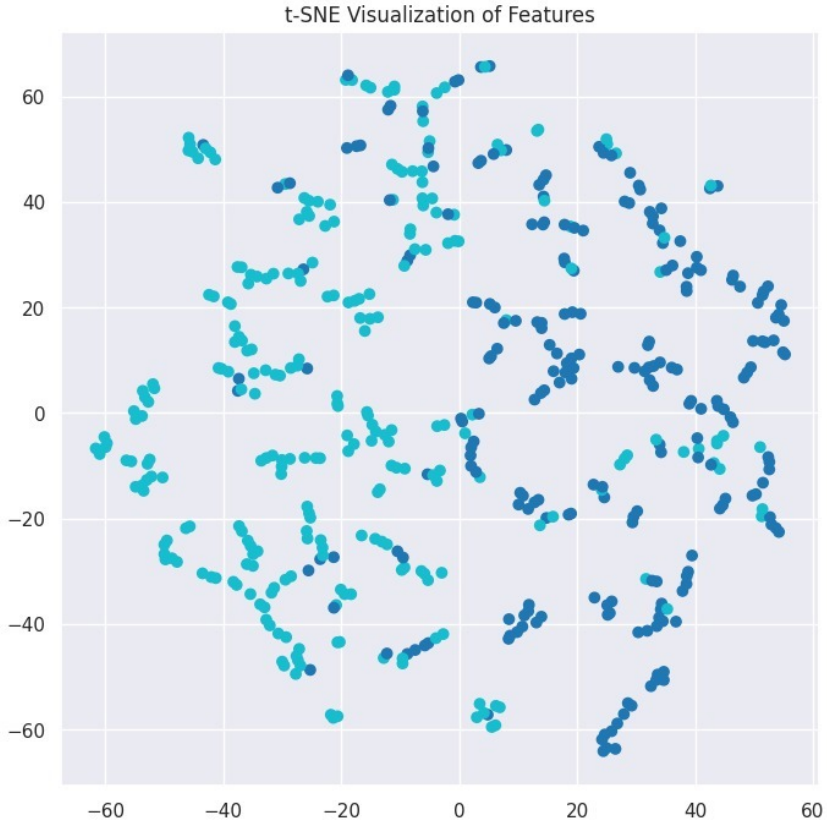
Пример работы

Пример работы

Заметим, что классы уже неплохо разбиваются попарно

Пример работы

Заметим, что классы уже неплохо разбиваются попарно



Итог

Итог

Уникальные аспекты SimCLR

Итог

Уникальные аспекты SimCLR

- Аугментация данных: SimCLR использует сильную аугментацию, включающую случайные срезки, повороты, изменения цветов и Гауссов блюр. Последовательно применяя эти дополнения к различным представлениям одного и того же изображения, SimCLR побуждает модель изучать надежные и значимые представления.

Итог

Уникальные аспекты SimCLR

- Аугментация данных: SimCLR использует сильную аугментацию, включающую случайные срезки, повороты, изменения цветов и Гауссов блюр. Последовательно применяя эти дополнения к различным представлениям одного и того же изображения, SimCLR побуждает модель изучать надежные и значимые представления.
- Больший размер батча: SimCLR использует больший размер батча по сравнению с предыдущими методами, что помогает увеличить количество негативных примеров, рассматриваемых во время обучения. Это позволяет модели изучить больше отличительных признаков.

Итог

Уникальные аспекты SimCLR

- Аугментация данных: SimCLR использует сильную аугментацию, включающую случайные срезки, повороты, изменения цветов и Гауссов блюр. Последовательно применяя эти дополнения к различным представлениям одного и того же изображения, SimCLR побуждает модель изучать надежные и значимые представления.
- Больший размер батча: SimCLR использует больший размер батча по сравнению с предыдущими методами, что помогает увеличить количество негативных примеров, рассматриваемых во время обучения. Это позволяет модели изучить больше отличительных признаков.
- Projection Head and Contrastive Loss: SimCLR представляет проекцию, которая отображает расширенные данные в пространство объектов, где применяются контрастные потери, что позволяет изучать более сложные представления по сравнению с линейными проекциями.