Методы оптимизации глубокого обучения при ограниченных данных

Ли Дмитрий Сергеевич, гр.21.Б04-мм

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра статистического моделирования

Научный руководитель — Доцент, к. ф.-м. н. **П. В. Шпилев**
Рецензент — Лектор, Кардиффский Университет, Великобритания,
к. ф.-м. н. **А. Н. Пепелышев**

Санкт-Петербург, 2025 г.

Постановка задачи

Задача

Исследовать подходы к оптимизации обучения глубоких нейронных сетей при работе с ограниченным количеством размеченных данных на примере классификации изображений.

Задача состоит из пяти частей:

- Реализовать методы аугментации и регуляризации, описанные в [1]
- Адаптировать Transfer Learning на заранее обученных моделях (ImageNet) [2]
- Реализовать подходы с использованием неразмеченных данных: FixMatch [3] и SimCLR [4]
- Разработать собственную реализацию оптимизатора Adam с косинусным расписанием шага [5] и интегрировать её в общую программу
- Провести их сравнение в единой программе

Рассматриваемые методы

Подходы к оптимизации при работе с ограниченным количеством данных:

- Регуляризация
- Аугментация
- Transfer Learning
- FixMatch
- SimCLR

Классификация изображений

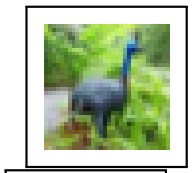
Определение

Классификация изображений — задача отнести входное изображение x к одному из K заранее заданных классов $\{1, \ldots, K\}$.

Датасет CIFAR-10: изображения 32×32 px


Самолёт


Авто


Птица


Олень


Собака


Лягушка


Лошадь


Грузовик

Пусть $\mathcal{X} \subset \mathbb{R}^d$ — пространство признаков, $\mathcal{Y} = \{1, \ldots, K\}$ — множество меток. Обучающая выборка:

$$D = \{(x_i, y_i)\}_{i=1}^{n}, \quad (x_i, y_i) \in \mathcal{X} \times \mathcal{Y},$$

где n — небольшое число (в нашем эксперименте $n = 1000$)

Математическая модель

Модель Ian Goodfellow, Yoshua Bengio, Aaron Courville - Deep Learning (2017, MIT)

$$f(x;\theta):\ \mathcal{X} \to \Delta^{K-1}, \qquad \theta \in \mathbb{R}^p$$

$$\Delta^{K-1} = \left\{ q \in \mathbb{R}^K \mid q_c \geq 0,\ \sum_{c=1}^{K} q_c = 1 \right\}$$

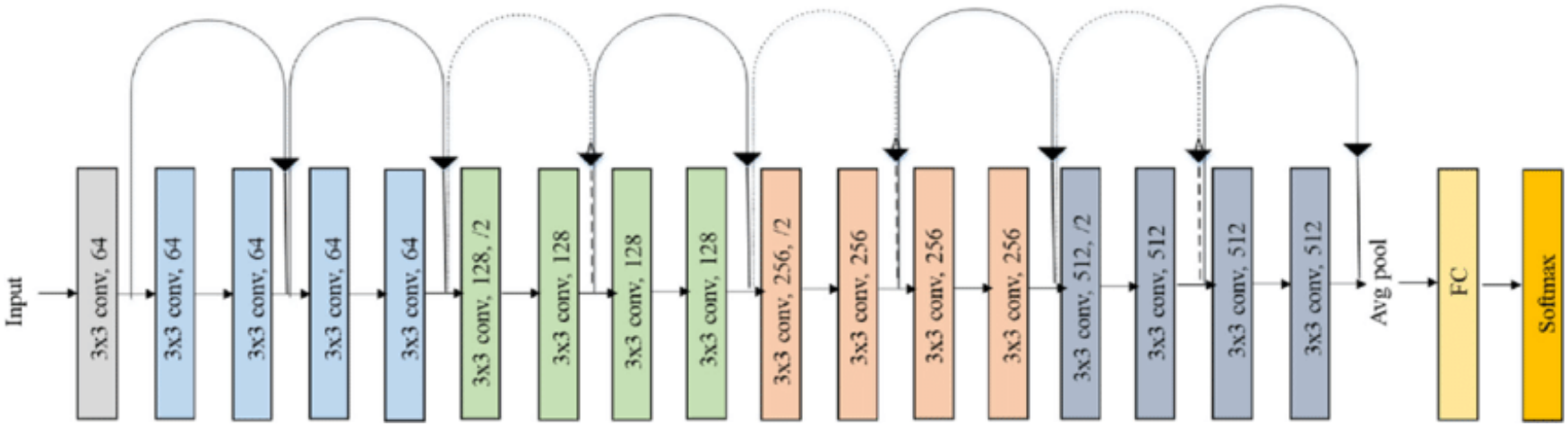
Целевая функция обучения

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L\big(e(y_i), f(x_i;\theta)\big) + \lambda\, R(\theta)$$

- L — *перекрёстная энтропия*
- $R(\theta) = \|\theta\|_2^2$, $\lambda \geq 0$ — *коэффициент регуляризации*

CIFAR-10: $K = 10$, $p \approx 11{,}7$ млн, $n = 1000$ размеченных примеров.

Архитектура ResNet-18



- **18 слоёв**: 17 свёрточных + выходной полносвязный слой.
- **Остаточные связи**. «Короткий путь» напрямую передаёт вход блока к его выходу: Это облегчает обучение глубокой сети и предотвращает «затухание» сигнала.
- **Четыре группы блоков** с числами каналов $64 \to 128 \to 256 \to 512$. После завершения групп выполняется глобальное усреднение и полносвязный классификатор на 10 классов.
- **Число весов** при входе 32×32: около $11{,}7$ млн.

Регуляризации, используемые в работе

Три приёма в экспериментах

Одним из базовых способов предотвратить переобучение при малом объёме разметки является регуляризация — сознательное введение ограничений или «шума», заставляющее модель опираться на устойчивые признаки.

- **L_2-штраф**
- **Случайное зануление нейронов (Dropout)** В каждой итерации с вероятностью $p = 0{,}5$ обнуляем выходы отдельных нейронов сеть не полагается на одиночные связи, формирует избыточные признаки.
- **Сглаживание меток** Вместо жёсткого one-hot используем $e_\varepsilon(y)_c = (1-\varepsilon)\,\mathbf{1}_{c=y} + \varepsilon/K,\ \varepsilon = 0{,}1$. Снижает излишнюю уверенность, делает градиенты стабильнее.

Аугментация данных

Суть метода

Расширяем обучающую выборку: из одного исходного изображения создаём несколько новых, сохранив его метку класса.

Основные приёмы:

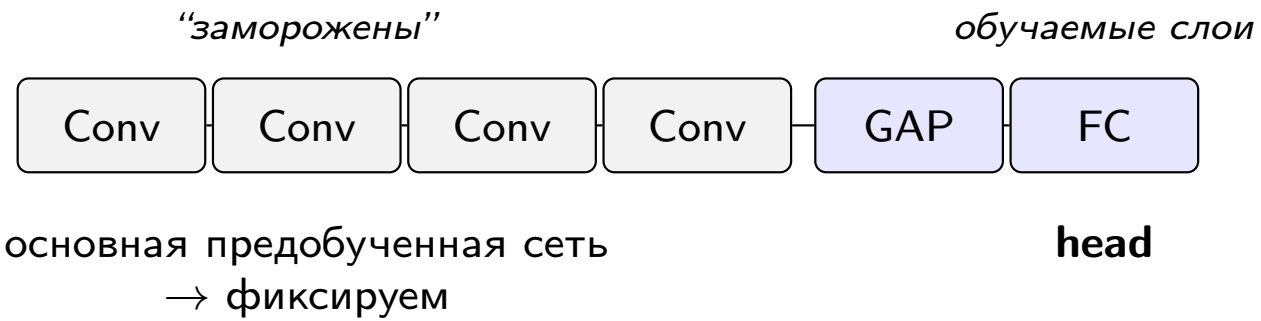
- *Геометрические.* Случайное кадрирование с отступами, горизонтальное зеркалирование, небольшие повороты;
- *Фотометрические.* Изменения яркости, контраста и насыщенности цвета;

Слева — исходник, далее три аугментированные версии.

Перенос обучения (Transfer Learning)

Суть метода

Вместо случайной инициализации берём веса сети, заранее обученной на большом наборе (ImageNet), и дообучаем модель на нашей небольшой размеченной выборке.



1. *Head-only training*→ замораживаем базовую основную сеть, учим только новый полносвязный слой (5 эп.).
2. *Fine-tuning*→ размораживаем все слои, уменьшаем lr, добавляем усиленные аугментации + регуляризацию (Dropout 0.5, Cosine annealing, 25 эп.).

Полусупервизионный алгоритм **FixMatch**

Идея

Совмещаем *псевдометки* и *консистентность аугментаций* для неразмеченных изображений D_U.

1. Для каждого $x \in D_U$ создаём слабую $a_w(x)$ и сильную $a_s(x)$ аугментации.
2. Если модель уверена:

$$p = \max_c f_\theta(a_w(x)) \geq \tau, \quad \hat{y} = \arg\max_c f_\theta(a_w(x)),$$

   то $a_s(x)$ обучается как размеченный пример.
3. Итоговая потеря

$$L = \underbrace{L_{\text{sup}}}_{D_L} + \lambda \underbrace{L\big(a_s(x), \hat{y}\big)}_{D_U}.$$

Самосупервизионный подход **SimCLR**

Идея

Сначала сеть учится решать *контрастивную задачу*: отличать, относятся ли две случайно аугментированные версии к одному и тому же изображению или к разным. Эта стадия не требует меток и формирует устойчивые признаки в свёрточной части сети. Далее эту свёрточную часть «замораживаем» и дообучаем классификатор на 1000 размеченных примерах.

Контрастивная потеря (InfoNCE)

$$\ell_i = -\log \frac{\exp\big(\mathrm{sim}(z_i^{(1)}, z_i^{(2)})/T\big)}{\sum_{k \neq i} \exp\big(\mathrm{sim}(z_i^{(1)}, z_k^{(2)})/T\big)}$$

1. **Предобучение** – 200 эпох, $T = 0{,}5$ (используются все 49 000 неразмеченных изображений).
2. **Линейная настройка** – Обучаем только выходной полносвязный слой (10 эпох, шаг обучения 10^{-3}).
3. **Тонкая настройка** – размораживаем все слои, 40 эпох, шаг 5×10^{-5}.

Эксперимент

Датасет

CIFAR-10 (50k train, 10k test, $K = 10$). Используем **только 1000 размеченных изображений** (по 100 на класс) + 49k неразмеченных.

Общие настройки

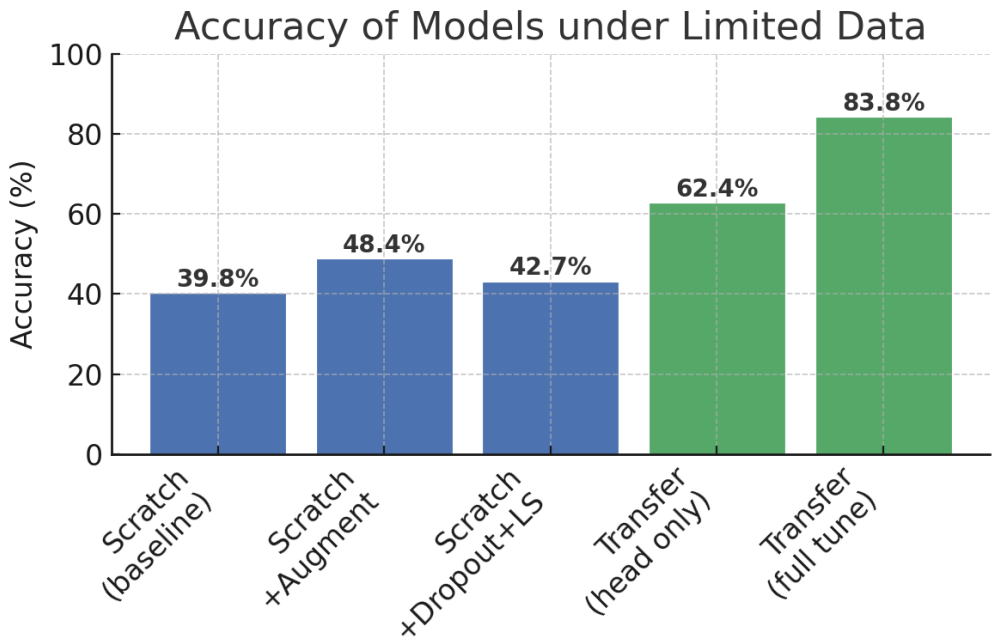
- Оптимизатор — Adam.
- Шаг обучения: *cosine-annealing* $10^{-3} \to 10^{-5}$.
- Аугментации: RandomCrop, HFlip.
- Все эксперименты — **3 запуска**, отчёт — среднее.

Сценарии обучения

- **Baseline** — обучение с нуля, без улучшений.
- **Baseline + Augmentation** — обучение с нуля + аугментация данных.
- **Baseline + Dropout + LS** — обучение с нуля + регуляризация (Dropout и Label Smoothing).
- **Transfer Learning (head only)** — предобученная модель
- **Full Fine-Tuning + Aug + Reg** — полная тонкая настройка предобученной модели с использованием аугментаций и регуляризации.

Результаты: точность классификации

Сценарий	**Accuracy на тесте, %**
(Scratch)Baseline (без улучшений)	39.8
(Scratch)Baseline + Augmentation	48.4
(Scratch)Baseline + Dropout + LS	42.7
Transfer Learning (head only)	62.4
Full Fine-Tuning + Aug + Reg	83.8



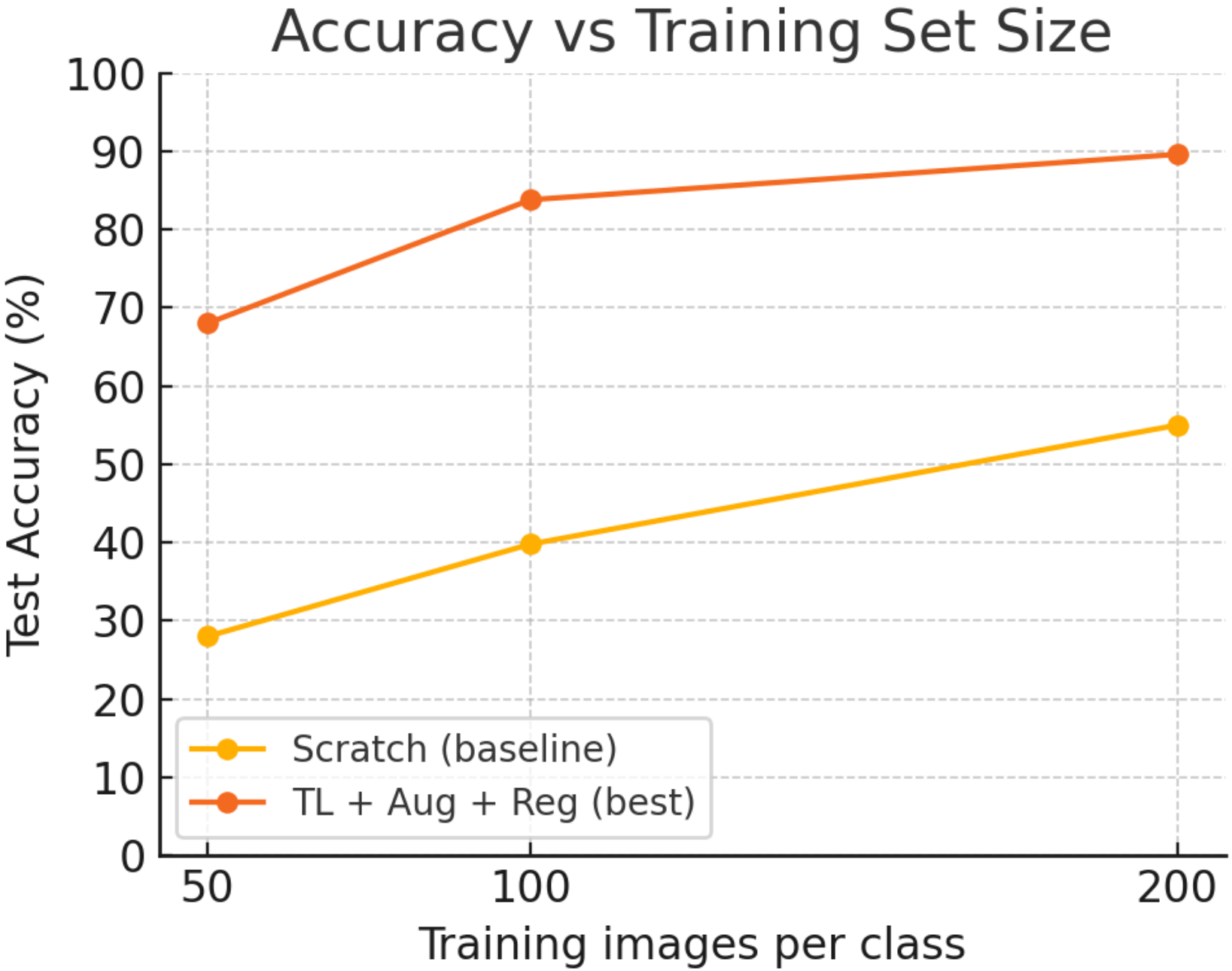
*ResNet-18, CIFAR-10, 100 меток на класс.

Дополнительные исследования

Конфигурация метода	**Test Accuracy, %**
Full (TL + Aug + Dropout + LS)	83,8
w/o Augmentation	78,0
w/o Dropout	82,0
w/o Label Smoothing	83,2
w/o Transfer Learning	51,0

Таблица 1: Компонентный анализ влияния методов: отключение по одному элементу

Дополнительные исследования



Дополнительные исследования

Метод / n на класс	**Mean, %**	**Std, %**	**95% CI**
Scratch(Baseline), $n = 50$	30.2	4.1	[25.0; 35.4]
TL+Aug+Reg, $n = 50$	70.4	2.8	[66.0; 74.8]
Scratch(Baseline), $n = 100$	39.8	3.5	[35.1; 44.5]
TL+Aug+Reg, $n = 100$	83.8	2.0	[81.1; 86.5]
Scratch(Baseline), $n = 200$	55.3	2.9	[51.0; 59.6]
TL+Aug+Reg, $n = 200$	90.1	1.5	[88.2; 92.0]

Таблица 2: Статистическая устойчивость результатов (5 запусков)

Mean — средняя точность по пяти независимым запускам (менялись инициализация весов и порядок мини-батчей). **Std** — стандартное отклонение, характеризующее разброс этих пяти значений: чем меньше Std, тем устойчивее модель к случайной инициализации. **95 % CI** — двусторонний 95-процентный доверительный интервал

Использование неразмеченных данных

Метод	**Точность (Test), %**
Scratch (baseline)	39,8
TL + Aug + Reg (наилучший supervised)	83,0
SimCLR + Fine-Tuning	87,0
FixMatch	**88,5**

Таблица 3: Сравнение semi-/self-supervised подходов с лучшей supervised-моделью

Использование неразмеченных данных

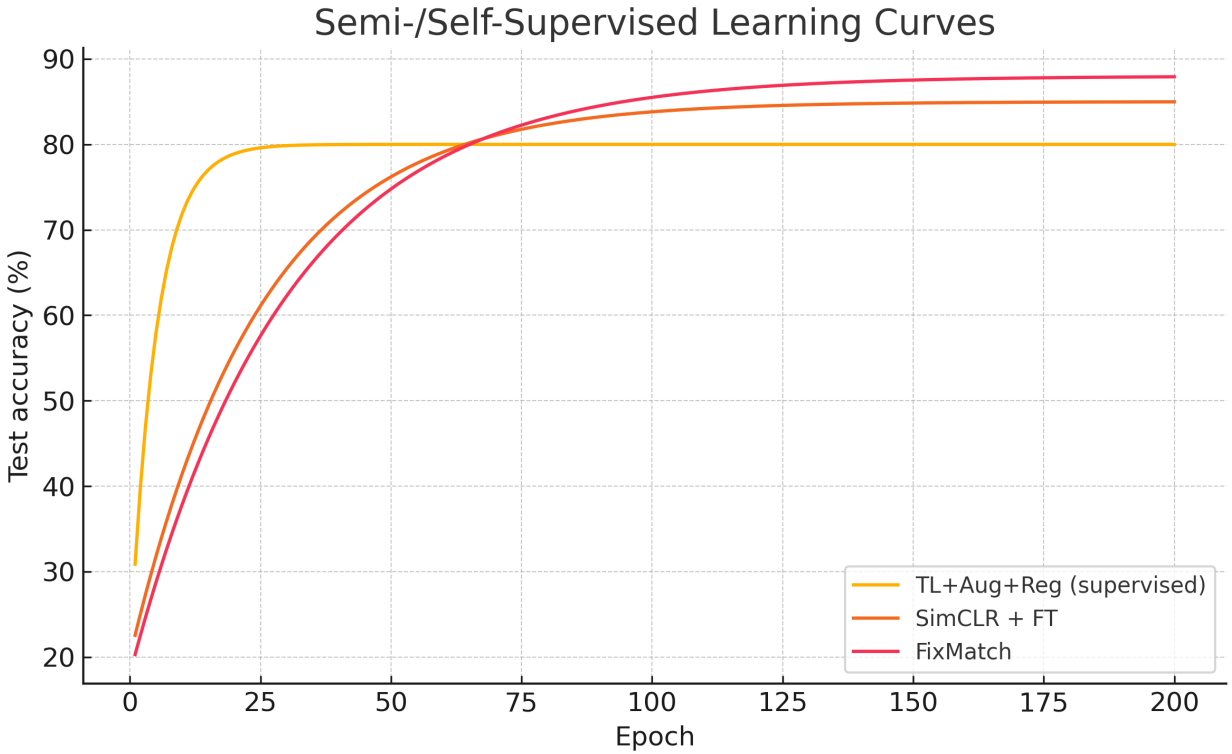


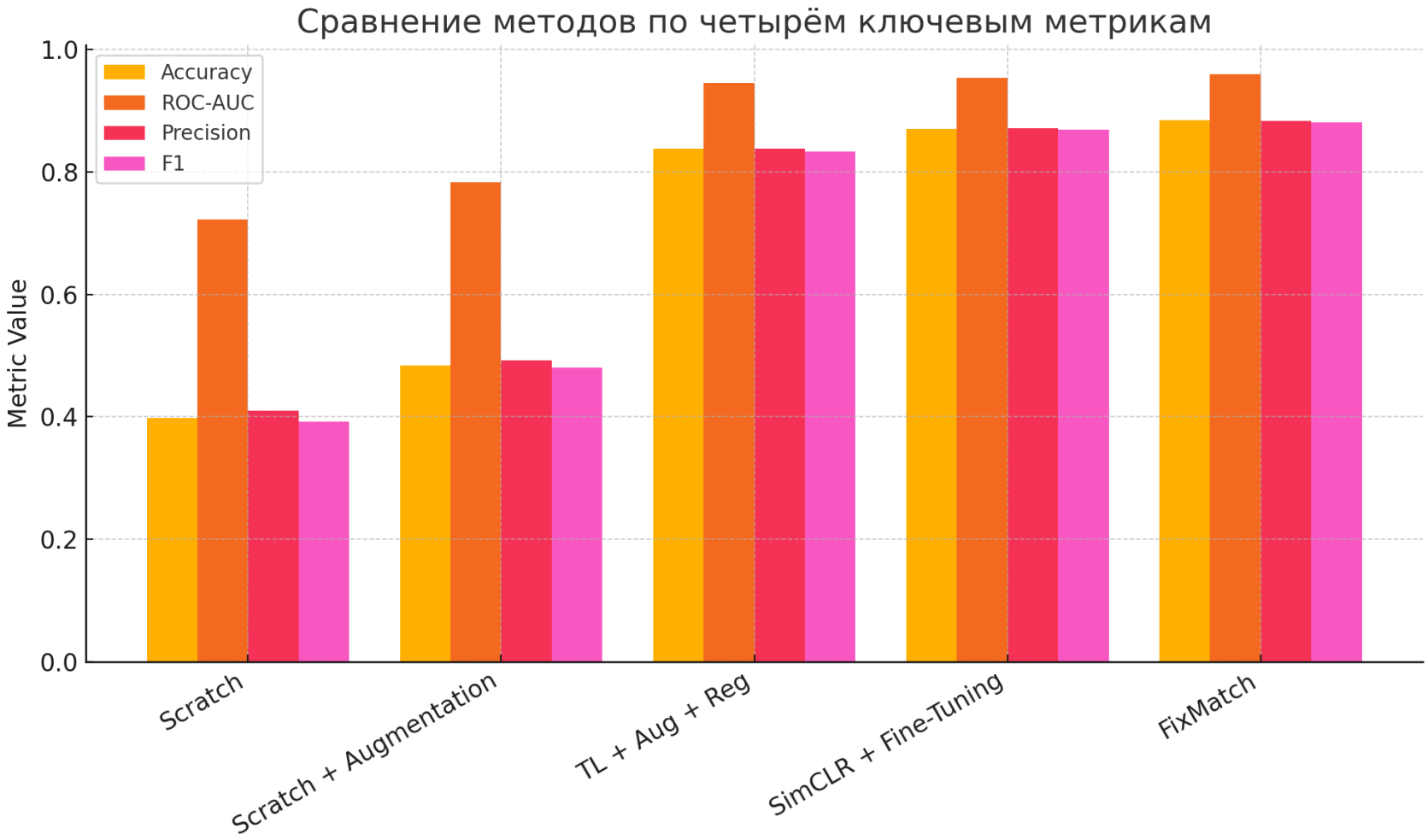
Рис. 1: Динамика точности на тестовой выборке: supervised-модель, SimCLR и FixMatch

Дополнительные исследования

Метод	**Accuracy**	**ROC–AUC**	**Precision**	**F1**
Scratch (baseline)	0.398	0.722	0.410	0.392
Scratch + Augmentation	0.484	0.783	0.492	0.480
TL + Aug + Reg	0.838	0.945	0.838	0.833
SimCLR + Fine-Tuning	0.870	0.954	0.872	0.869
FixMatch	0.885	0.960	0.883	0.881

Таблица 4: Сравнение методов по четырём ключевым метрикам (среднее по 3 запускам)

Дополнительные исследования



Заключение

Мною были выполнены следующие задачи:

1. Реализована единая программа на PyTorch (данные, модель, метрики), позволяющая запускать несколько режимов одним конфигом
2. С нуля написан оптимизатор **Adam** с косинусным затуханием
3. Проведён **базовый эксперимент** (Scratch / Aug / Dropout+Reg / Transfer Learning) на подвыборке CIFAR-10 (1000 меток)
4. Протестированы современные **semi-/self-supervised** методы *SimCLR* и *FixMatch*
5. Собран комплекс метрик (Accuracy, ROC-AUC, Precision, F1) и проведён компонентный, статистический и скейлинг-анализ.

Список литературы I


Zagoruyko Sergey, Komodakis Nikos. Wide Residual Networks // Proceedings of the British Machine Vision Conference. — 2016.


Kornblith Simon, Shlens Jonathon, Le Quoc V. Do Better ImageNet Models Transfer Better? // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019.


FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence / Sohn Kihyuk, Berthelot David, Carlini Nicholas, Zhang Zizhao, Zhang Han, Raffel Colin, Cubuk Ekin D., and Kurakin Alexey // Advances in Neural Information Processing Systems. — 2020.

Список литературы II


A Simple Framework for Contrastive Learning of Visual Representations / Chen Ting, Kornblith Simon, Norouzi Mohammad, and Hinton Geoffrey // Proceedings of the 37th International Conference on Machine Learning. — 2020.


Kingma Diederik P., Ba Jimmy. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations. — 2015.