

академия
больших
данных

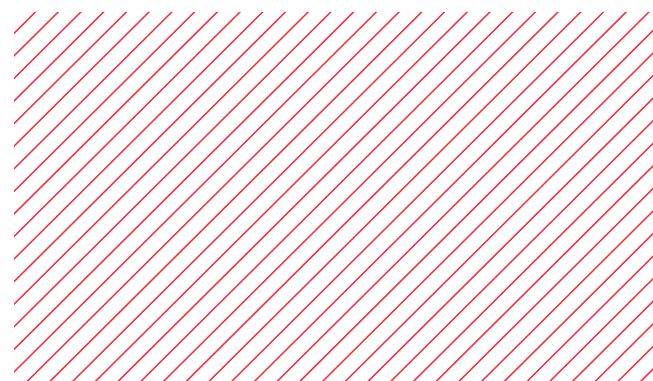


mail.ru
group

Few-shot and semi-supervised learning

Андрей Бояров

Ведущий инженер-исследователь, команда
машинного зрения



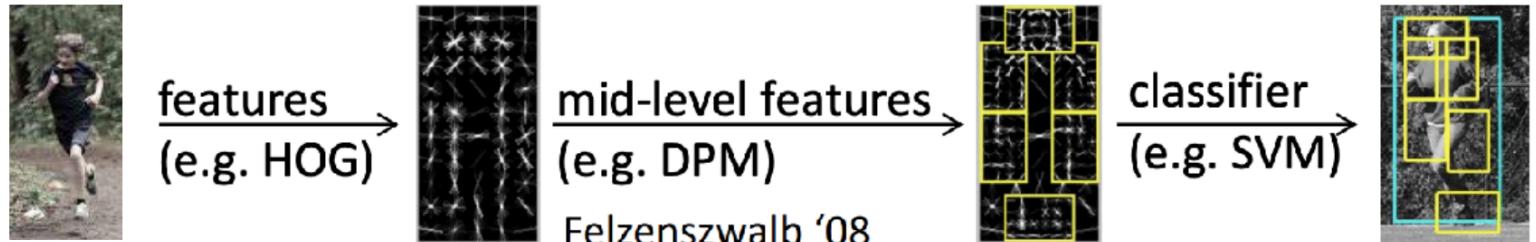


Обучение с учителем

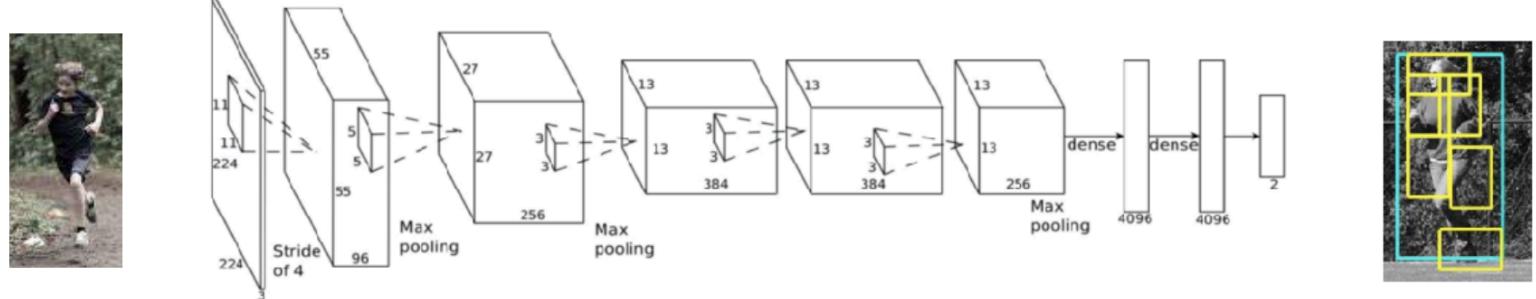
- Наиболее успешные алгоритмы DL связаны с обучением с учителем
- Задачи:
 - Computer vision
 - Text
 - Speech
 - RL

Deep Learning

Standard computer vision:
hand-designed features



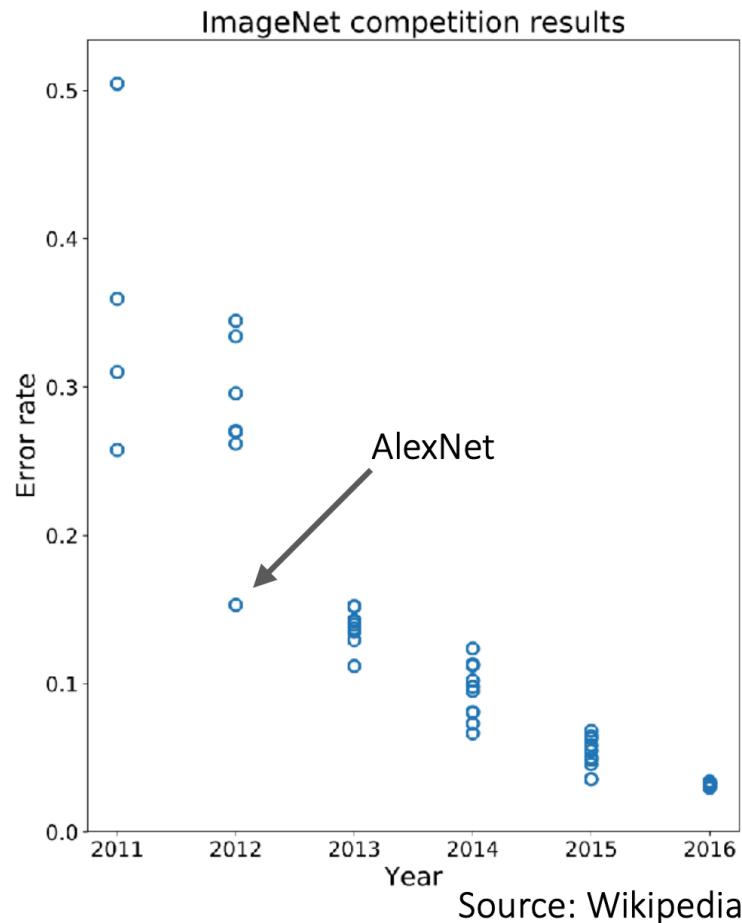
Modern computer vision:
end-to-end training



Krizhevsky et al. '12

Deep Learning

Deep learning for object classification



Deep learning for machine translation

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifeng,qvl,mnorouzi@google.com

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

Human evaluation scores on scale of 0 to 6

PBMT: Phrase-based machine translation

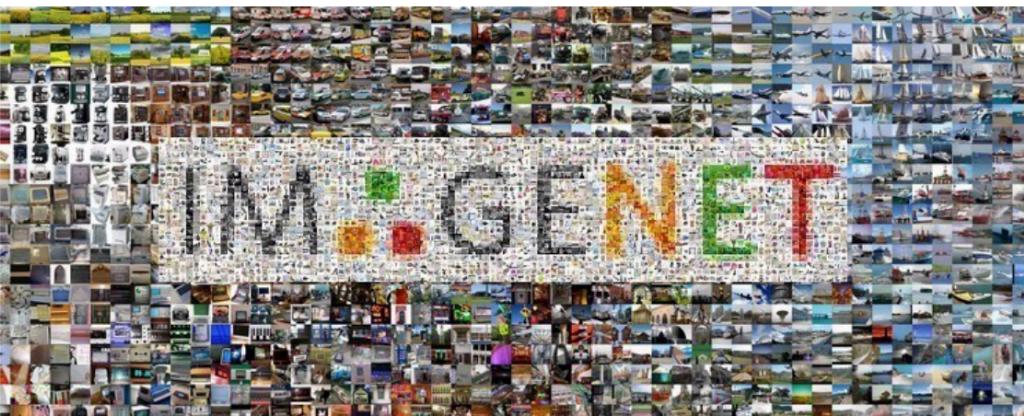
GNMT: Google's neural machine translation (in 2016)

Big data for DL

Large, diverse data
(+ large models)

deep learning

Broad generalization



Russakovsky et al. '14

GPT-2
Radford et al. '19

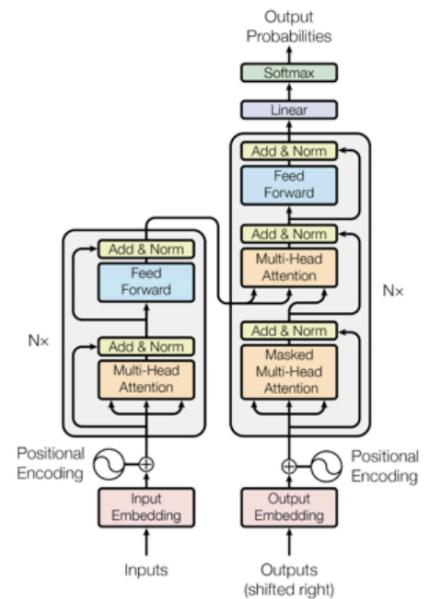
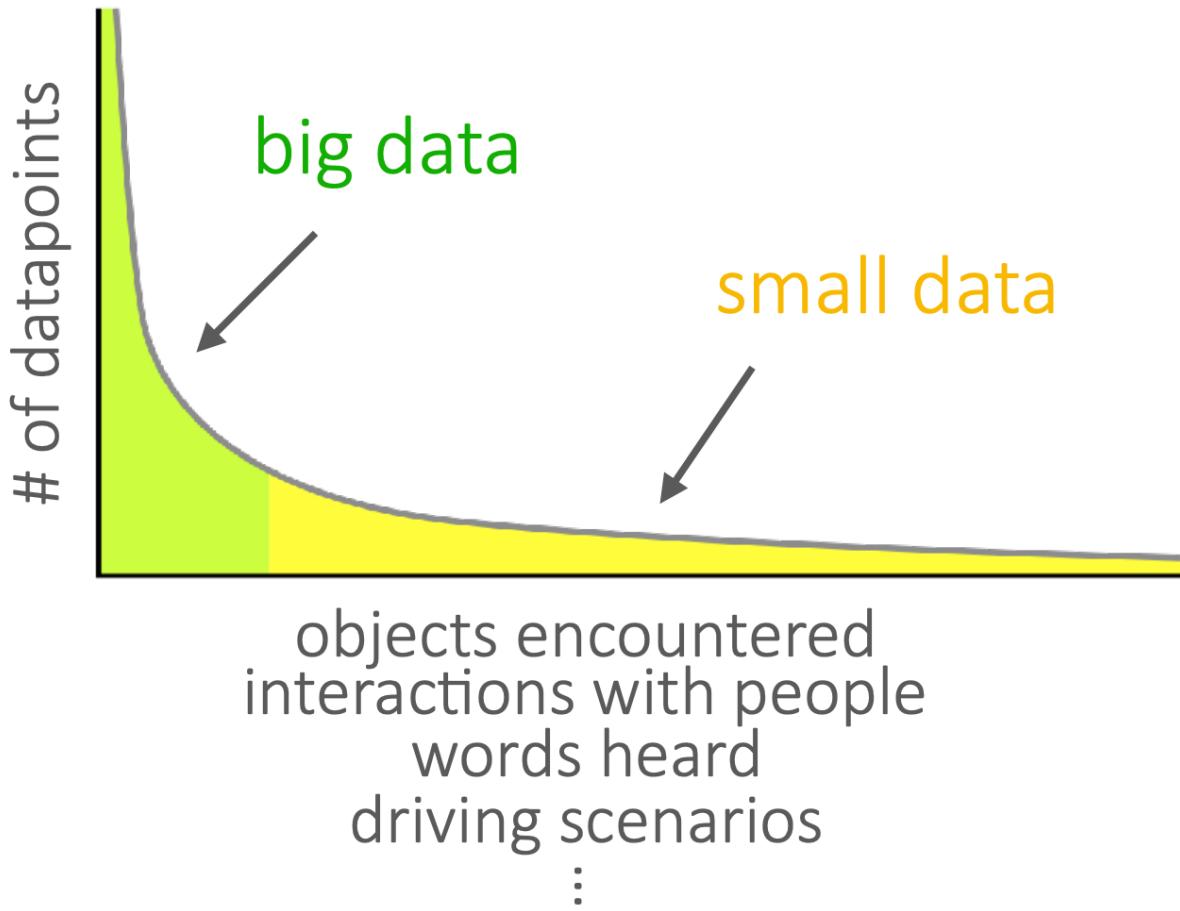


Figure 1: The Transformer - model architecture.

Vaswani et al. '18

Big data for DL



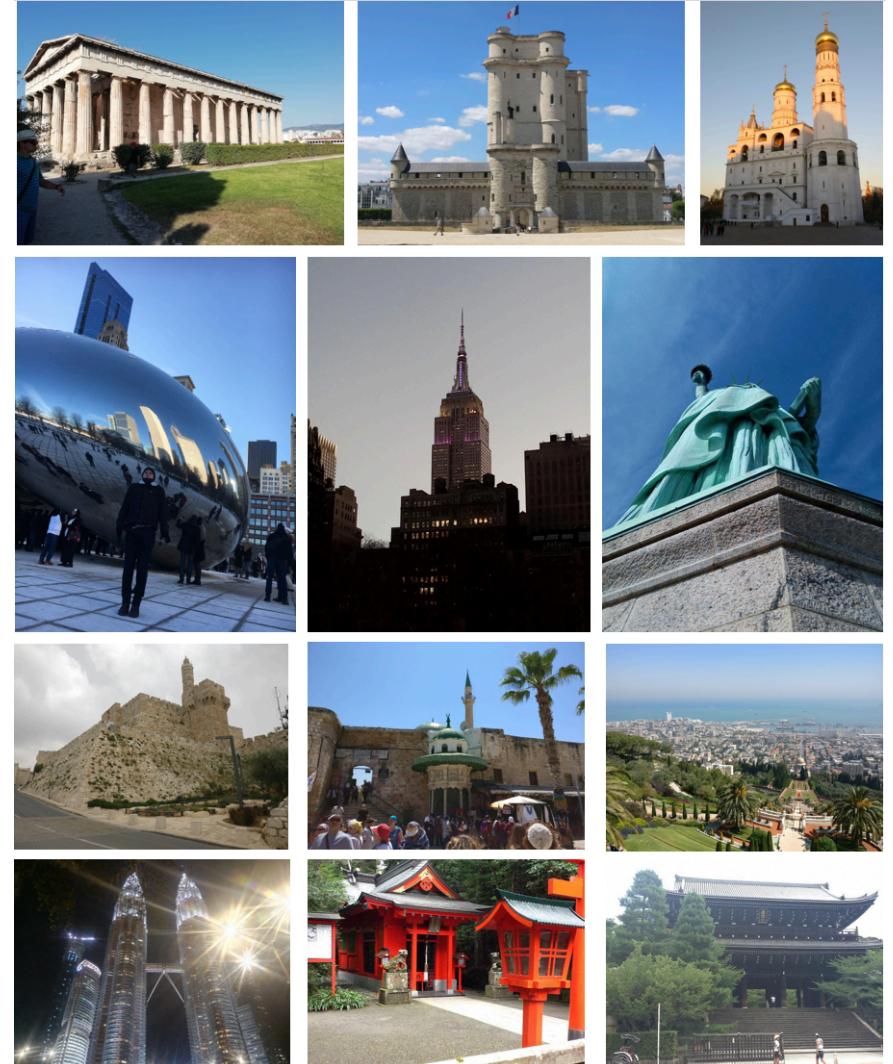


Big data

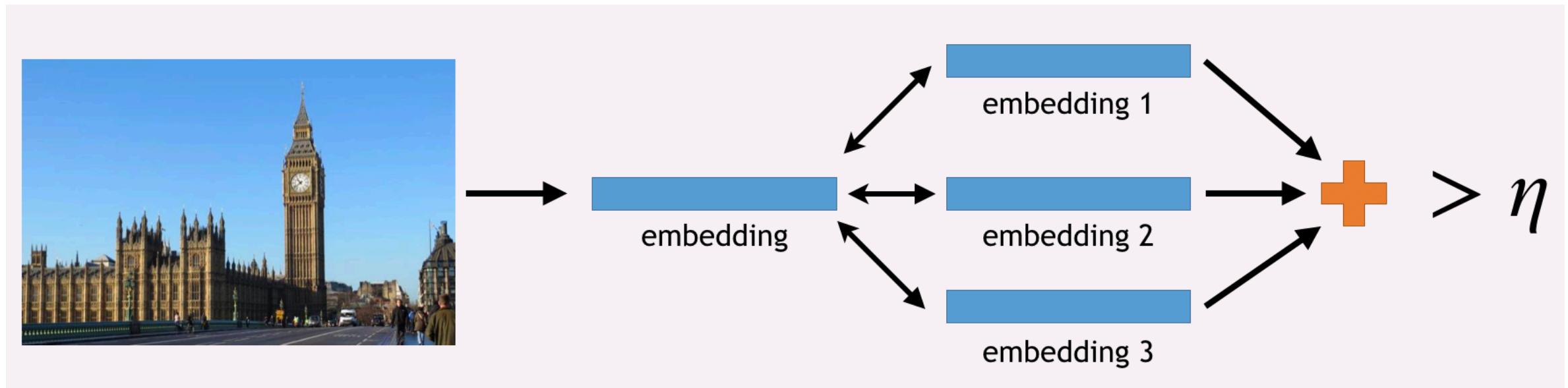
- Под каждую задачу надо собирать и размечать данные
- Сбор и расчистка данных
- Разметка данных
- Затратно по человеческим ресурсам

Вариант: автоматическая чистка

- 4 региона мира (4 этапа обучения)
 - Страна
 - Город
 - Список достопримечательностей
- Автоматическая чистка базы
 - 3 – 5 вручную проверенных «эталона» на каждую



Вариант: автоматическая чистка





Выход

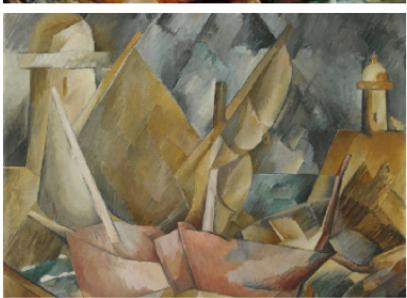
- Использовать данные с малым количеством примеров
(few-shot learning (meta-learning))
- Использовать неразмеченный данные
(semi-supervised learning)

Few-shot learning

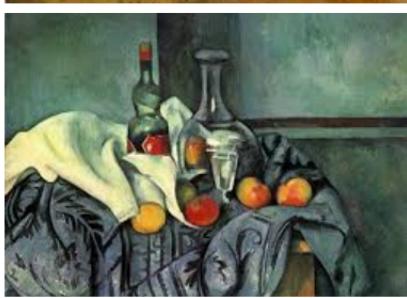
Few-shot learning

training data

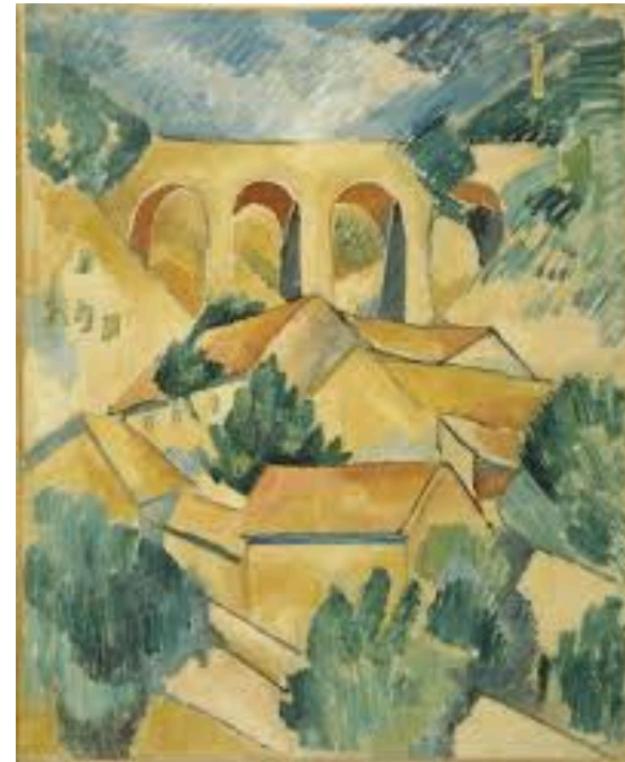
Braque



Cezanne



test datapoint



By Braque or Cezanne?

Few-shot learning

એ

એ એ એ એ એ
ગ એ એ એ એ
ર એ એ એ એ
મ એ એ એ એ

Omniglot dataset: 1-shot 20-way classification.

Few-shot learning

1623 characters from 50 different alphabets

Hebrew

וּ	טַבְדֵּלָה	בְּ	לְ	כְּ
נִ	אֲבָתָה	מְ	בְּ	מְ
לְ	גָּזָבָה	רְ	לְ	אֲ
לְ	תְּבָבָה	לְ	לְ	בְּ
לְ	בְּ	לְ	לְ	לְ

Bengali

ବ୍ରାହ୍ମିକାନ୍ତରିକ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ
କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ
କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ
କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ
କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ	କମାଳାଓଟାର୍କ୍ଷାଇ

Greek

φ	λ	β	δ	γ
μ	α	κ	χ	ν
π	θ	γ	ι	σ
ω	π	η	ο	ε
ρ	ξ	ζ	ψ	

Futurama

କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ
କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ
କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ
କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ
କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ	କୁଣ୍ଡାଳାର୍କ୍ଷାଇ

...

20 instances of each character

Few-shot learning (формально)

Пусть C — количество классов, а N — количество примеров на каждый класс в наборе размеченных данных $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{CN}, y_{CN})\}$, где $\mathbf{x}_i \in \mathbb{R}^d$ является вектором-примером, а $y_i \in \{1, \dots, C\}$ — метка класса. Тогда обозначим N_S число примеров в опорном множестве для каждого класса и N_Q — число примеров во множестве запросов, $N_S + N_Q = N$. Пусть $N_C \leq C$ — число классов в задаче. Такой процесс называется *классификацией N_C классов по N_S примерам (N_S -shot N_C -way)*.

Few-shot learning (формально)

Пусть эпизод $\xi_t : (t_1, \dots, t_M)$ состоит из M задач. Каждая задача t_i содержит опорное множество S_{t_i} и множество запросов Q_{t_i} : (S_{t_i}, Q_{t_i}) , где

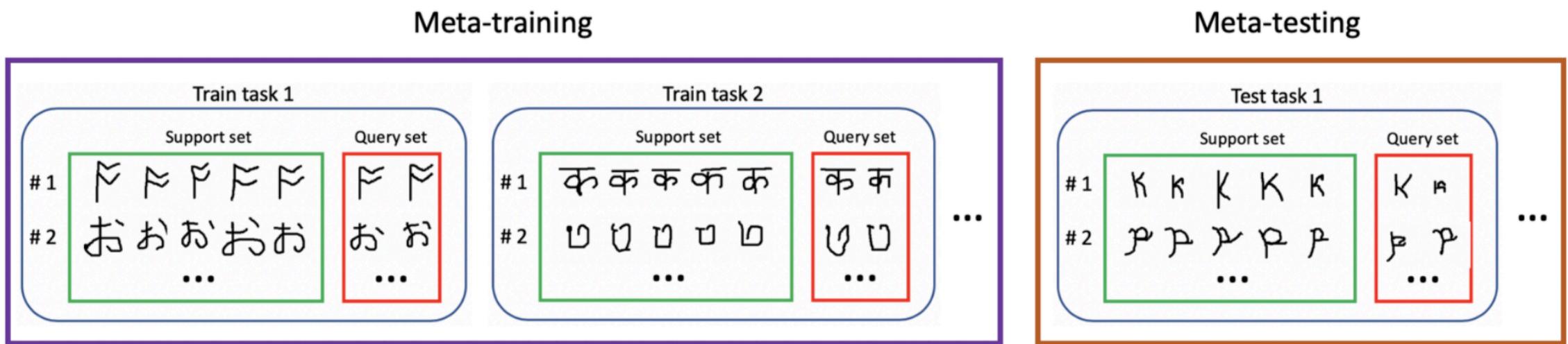
$$S_{t_i} = \{S_{t_i}^k\}_{k=1}^{N_C}, Q_{t_i} = \{Q_{t_i}^k\}_{k=1}^{N_C}, S_{t_i}^k \cap Q_{t_i}^k = \emptyset.$$

Множества

$$S_{t_i}^k = \{x_j | y_j = k\}_{j=1}^{N_S} \text{ и } Q_{t_i}^k = \{x_j | y_j = k\}_{j=1}^{N_Q}$$

случайным образом выбираются из представителей класса k .

Few-shot learning pipeline



Few-shot learning pipeline

5-way, 1-shot image classification (Minilmagenet)

Given 1 example of 5 classes:



Classify new examples



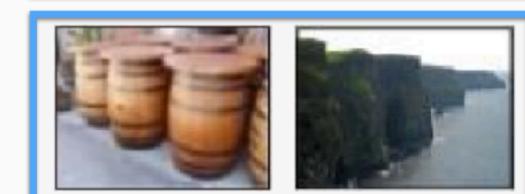
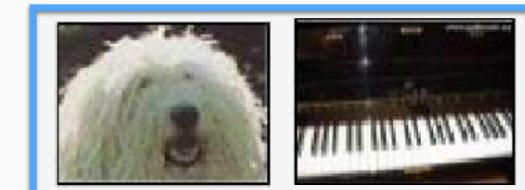
\mathcal{T}_1

meta-training

\mathcal{T}_2

⋮

⋮



Few-shot learning pipeline

learn *meta-parameters* θ : $p(\theta|\mathcal{D}_{\text{meta-train}})$



whatever we need to know about $\mathcal{D}_{\text{meta-train}}$ to solve new tasks

meta-learning: $\theta^* = \arg \max_{\theta} \log p(\theta|\mathcal{D}_{\text{meta-train}})$

adaptation: $\phi^* = \arg \max_{\phi} \log p(\phi|\mathcal{D}^{\text{tr}}, \theta^*)$



$$\phi^* = f_{\theta^*}(\mathcal{D}^{\text{tr}})$$

meta-learning: $\theta^* = \max_{\theta} \sum_{i=1}^n \log p(\phi_i|\mathcal{D}_i^{\text{ts}})$

$$\text{where } \phi_i = f_{\theta}(\mathcal{D}_i^{\text{tr}})$$

$$\mathcal{D}_{\text{meta-train}} = \{(\mathcal{D}_1^{\text{tr}}, \mathcal{D}_1^{\text{ts}}), \dots, (\mathcal{D}_n^{\text{tr}}, \mathcal{D}_n^{\text{ts}})\}$$

$$\mathcal{D}_i^{\text{tr}} = \{(x_1^i, y_1^i), \dots, (x_k^i, y_k^i)\}$$

$$\mathcal{D}_i^{\text{ts}} = \{(x_1^i, y_1^i), \dots, (x_l^i, y_l^i)\}$$

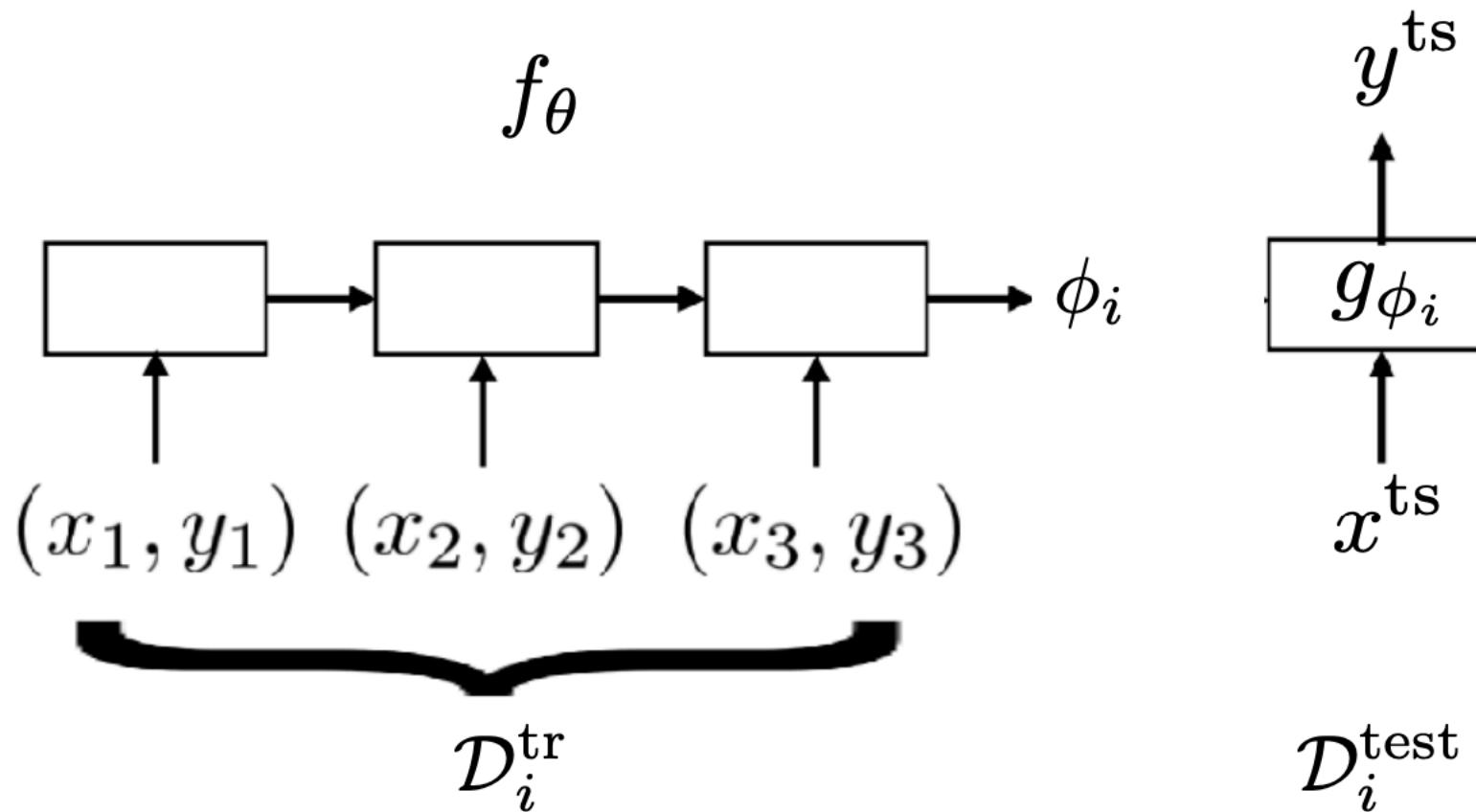


Основные типы алгоритмов

- Black-box
- Metric-based
- Optimization based

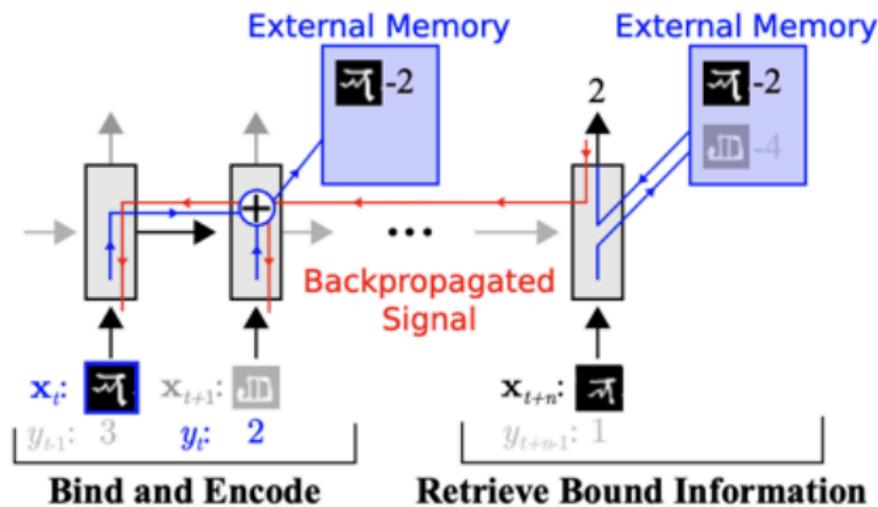
Black-box

Основная идея



Примеры black-box

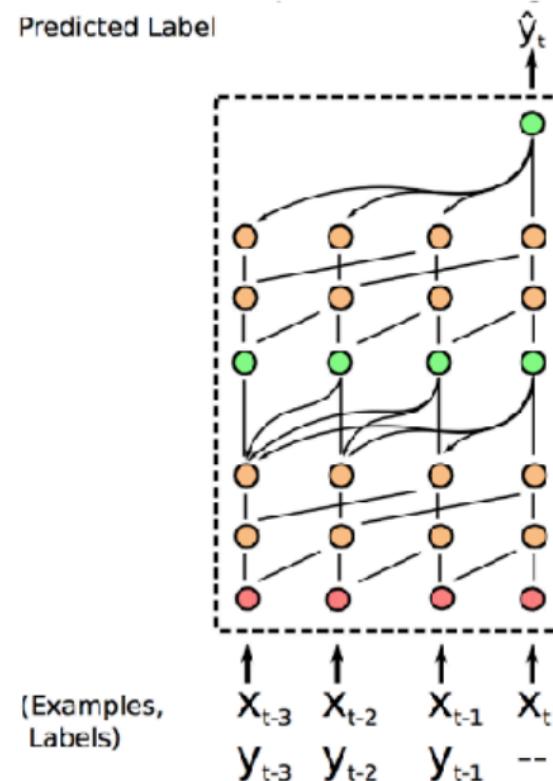
LSTMs or Neural turing machine (NTM)



Meta-Learning with Memory-Augmented Neural Networks
Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML '16

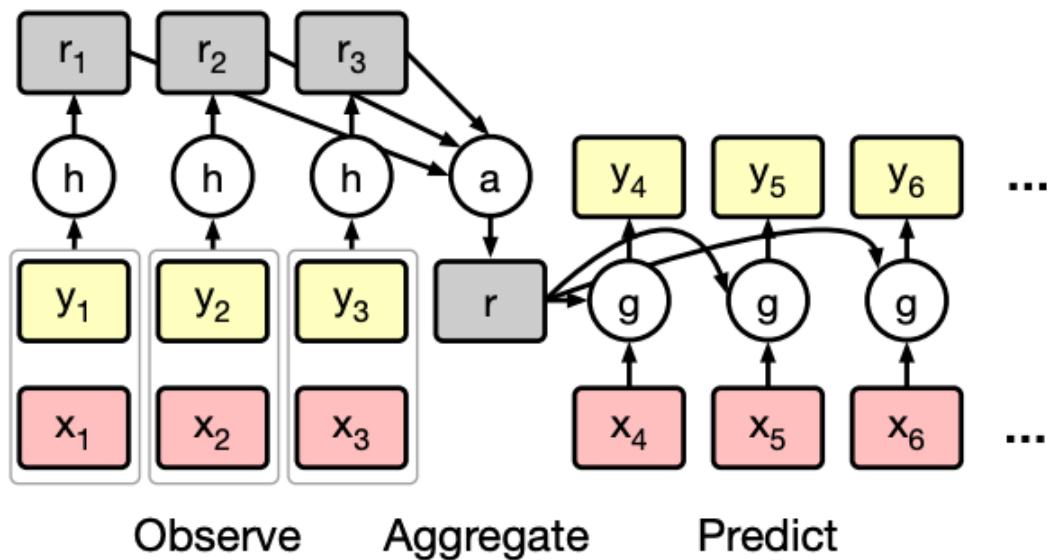
Примеры black-box

Convolutions & attention



A Simple Neural Attentive Meta-Learner
Mishra, Rohaninejad, Chen, Abbeel. ICLR '18

Примеры black-box



Conditional Neural Processes, Garnelo, et al., ICML'18

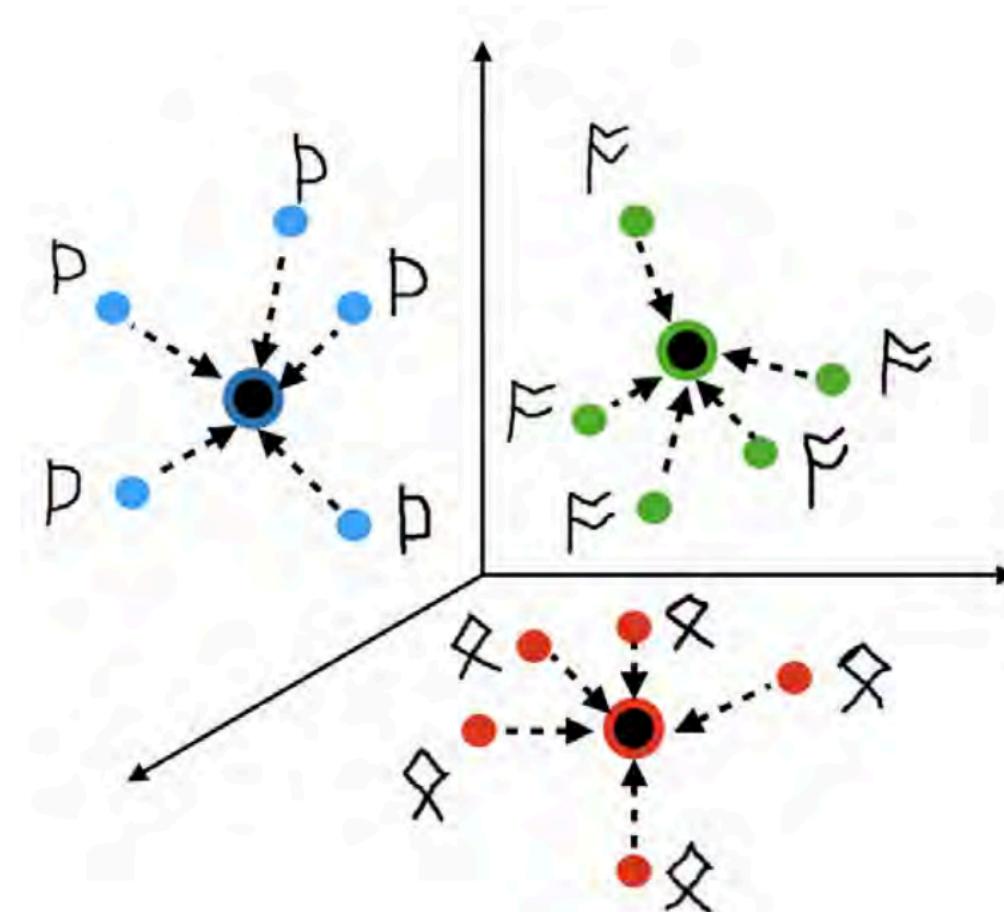


Black-box pro / contra

- Плюсы:
 - Достигают хороших результатов
 - Хорошо комбинируются с различными задачами (например, RL)
- Минусы:
 - Сложные модели
 - Порождают сложную оптимизационную задачу
 - Требуют больше данных

Metric-based

Prototypical Networks



Prototypical Networks

Пусть $\phi_\theta(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^n$ — свёрточная нейронная сеть с параметрами θ . В методе сетей прототипов для каждого класса k вычисляется представление $\mathbf{c}_{t_i}^k \in \mathbb{R}^n$, называемое прототипом. Каждый прототип является средним вектором, полученным по соответствующему опорному множеству

$$\mathbf{c}_{t_i}^k = \frac{1}{|S_{t_i}^k|} \sum_{\mathbf{x}_j \in S_{t_i}^k} \phi_\theta(\mathbf{x}_j).$$

Prototypical Networks

Функция потерь для класса k определяется как отрицательная прологарифмированная вероятность того, что элемент из запроса \mathbf{x} принадлежит классу k :

$$l_{\theta, t_i}^k(\mathbf{x}) = -\log \frac{\exp(-d(\phi_\theta(\mathbf{x}), \mathbf{c}_{t_i}^k))}{\sum_{k'} \exp(-d(\phi_\theta(\mathbf{x}), \mathbf{c}_{t_i}^{k'}))},$$

где $d(\cdot, \cdot)$ — это некоторая функция расстояния. В дальнейшем будет рассматриваться евклидово расстояние.

Prototypical Networks

Модель в сетях прототипов обучается с помощью стохастического градиентного спуска путём минимизации функции потерь для тренировочной задачи t_i

$$\mathcal{L}_{\theta,t_i}(Q_{t_i}) = \frac{1}{N_C} \sum_{k=1}^{N_C} \frac{1}{N_Q} \sum_{\mathbf{x}_j \in Q_{t_i}^k} l_{\theta,t_i}^k(\mathbf{x}_j).$$

Prototypical Networks

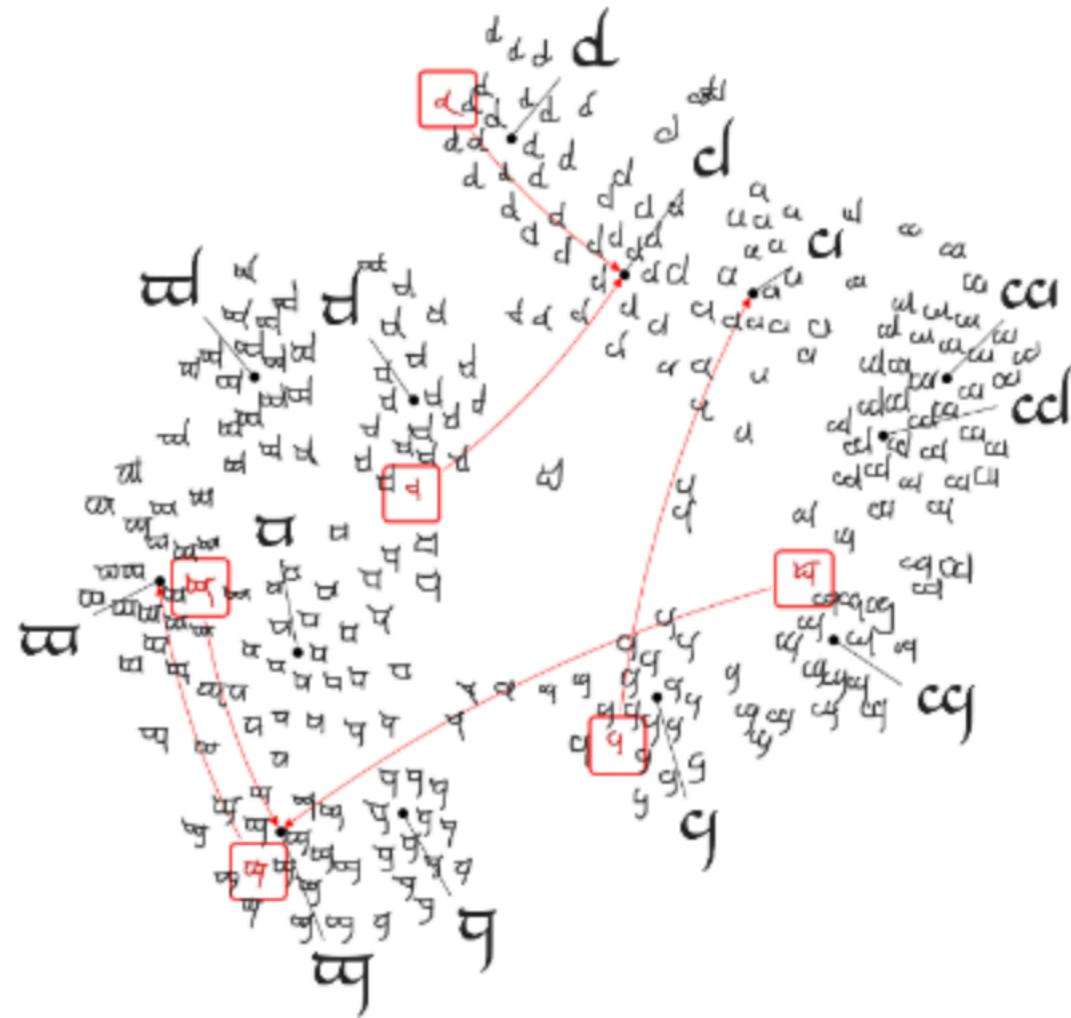
Алгоритм 1. Обучение для эпизода $\xi_t : (t_1)$.

Вход: N_S, N_Q, N_C

Выход: Обновлённые параметры θ

- 1: Случайно выбирается N_C классов
- 2: **for** $k \in \{1, \dots, N_C\}$ **do**
- 3: Случайным образом набираются элементы в $S_{t_1}^k$
- 4: Случайным образом набираются элементы в $Q_{t_1}^k$
- 5: Вычисляется $\mathbf{c}_{t_1}^k$ согласно (1)
- 6: **end for**
- 7: $\mathcal{L}_{\theta, t_1} = 0$
- 8: **for** $k \in \{1, \dots, N_C\}$ **do**
- 9: **for** $(\mathbf{x}, y) \in Q_{t_1}^k$ **do**
- 10: $\mathcal{L}_{\theta, t_1} = \mathcal{L}_{\theta, t_1} + \frac{1}{N_C N_Q} l_{\theta, t_1}^k(\mathbf{x})$
- 11: **end for**
- 12: **end for**
- 13: Параметры θ обновляются с помощью стохастического градиентного спуска по $\mathcal{L}_{\theta, t_1}$

Прототипы



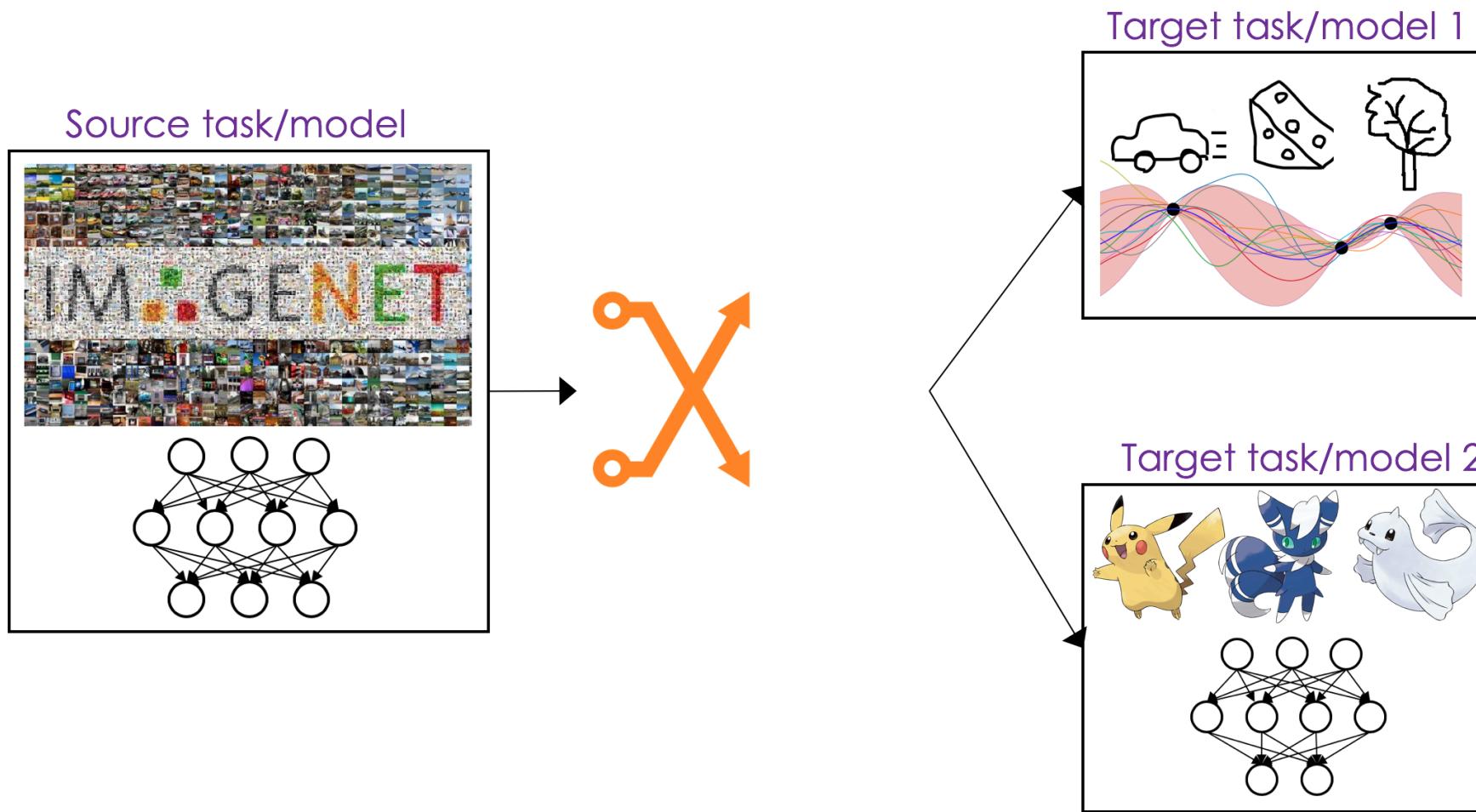


Metric-based pro / contra

- Плюсы:
 - Лёгкость модификаций
 - Не зависит от выбора архитектуры сети
 - Хорошая обобщающая способность
- Минусы:
 - Результаты обычно хуже, чем у optimization-based методов

Optimization-based

Transfer learning based



Transfer learning based

Fine-tuning

$$\phi \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

(typically for many gradient steps)

pre-trained parameters

training data
for new task

Transfer learning based

Fine-tuning [test-time]

$$\phi \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

pre-trained parameters

training data
for new task

Meta-learning

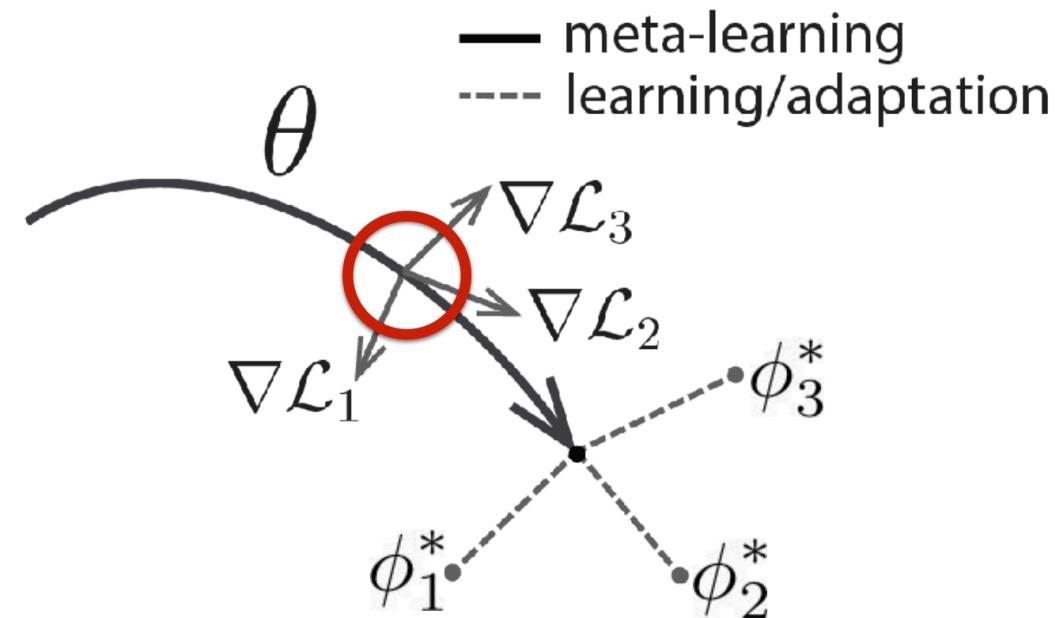
$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

Model Agnostic Meta Learning (MAML)

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

θ parameter vector
being meta-learned

ϕ_i^* optimal parameter
vector for task i





Optimization-based pro / contra

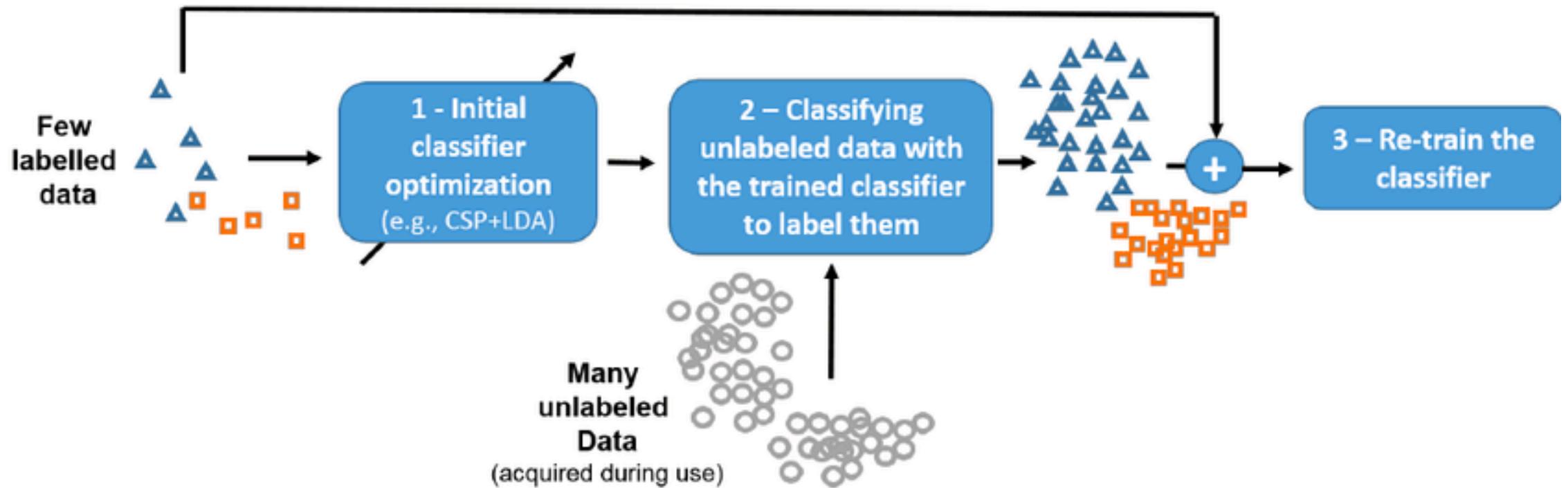
- Плюсы:
 - Показывают лучшие результаты
 - Хорошо комбинируются с различными задачами (например, RL)
 - Не зависит от выбора архитектуры сети
 - Хорошая обобщающая способность
- Минусы:
 - Требуют вычисления градиента второго порядка
 - Требуют много вычислительных ресурсов и памяти

MinilmageNet 5-way results

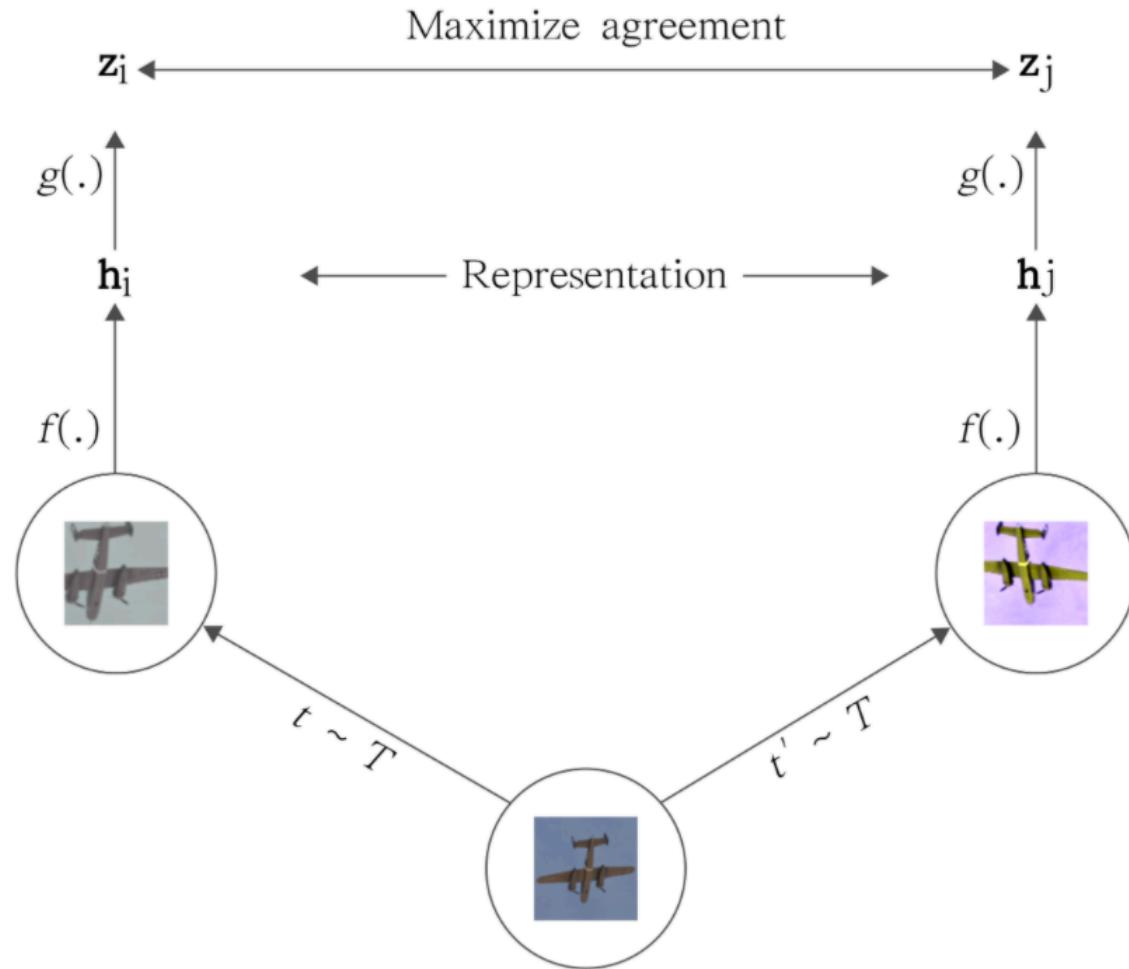
Model	Backbone	1-shot	5-shot
MAML (Finn et al., 2017)	ConvNet-4	51.67 ± 1.81	70.30 ± 1.75
Prototypical Networks* (Snell et al., 2017)	ConvNet-4	53.31 ± 0.89	72.69 ± 0.74
Relation Networks* (Sung et al., 2018)	ConvNet-4	54.48 ± 0.93	71.32 ± 0.78
LEO (Rusu et al., 2019)	WRN-28-10	66.33 ± 0.05	81.44 ± 0.09
MetaOptNet (Lee et al., 2019)	ResNet-12	65.99 ± 0.72	81.56 ± 0.53

Semi-supervised

Semi-supervised learning idea



SimCLR



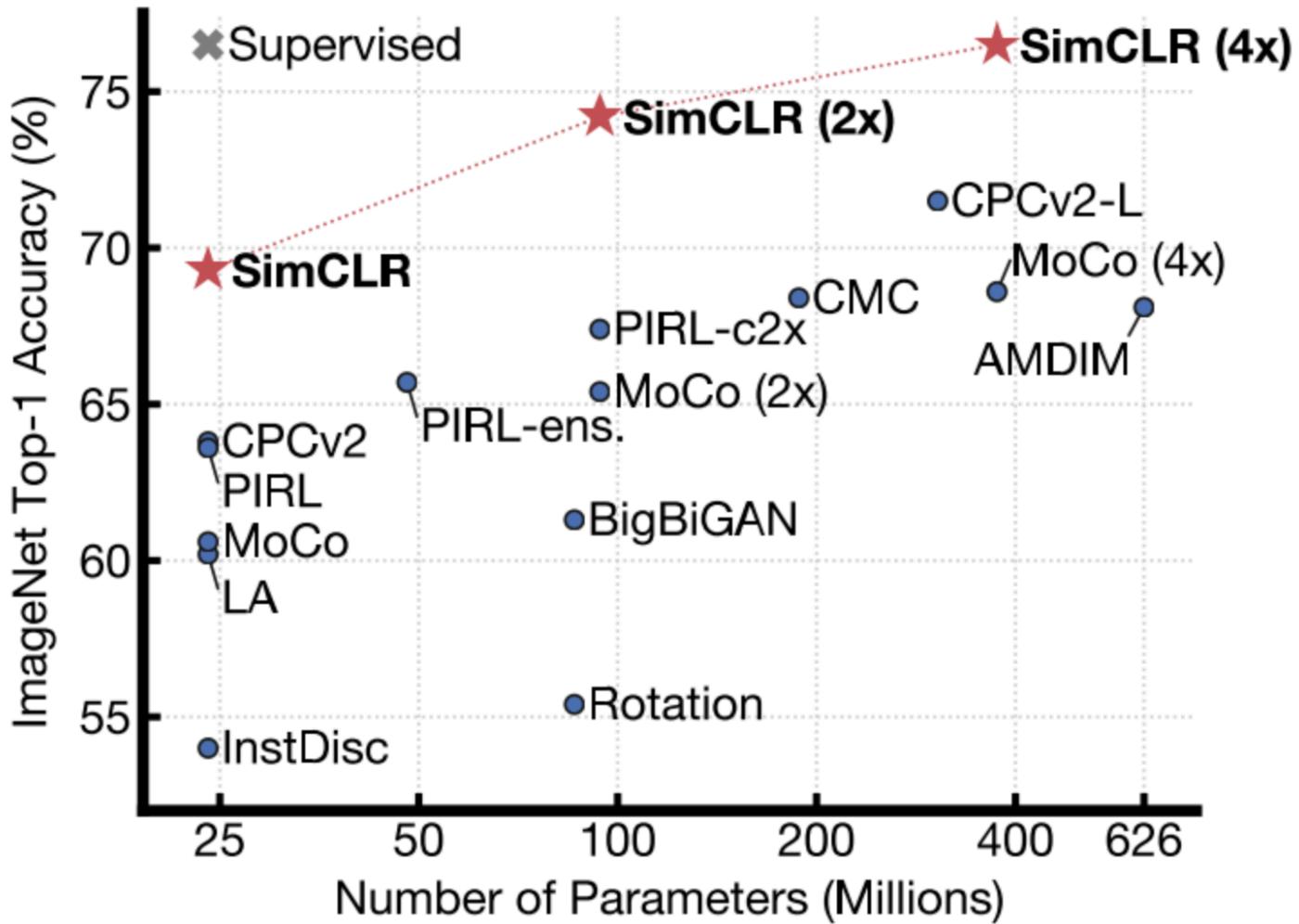
A Simple Framework for Contrastive Learning of Visual Representations, Chen et al., 2020



SimCLR: Noise Contrastive Estimator loss

$$NCE_{Loss} = -\log \frac{\exp(\text{sim}(g(\mathbf{x}), g(\mathbf{x}^+)))}{\exp(\text{sim}(g(\mathbf{x}), g(\mathbf{x}^+))) + \sum_{k=1}^K \exp(\text{sim}(g(\mathbf{x}), g(\mathbf{x}_k^-)))}$$

SimCLR





Заключение

- Во многих задачах мало размеченных данных
- Если будут добавляться новые классы, то FSL
 - Black-box
 - Metric-based
 - Optimization-based
- Если есть неразмеченные данные
 - Semi-supervised learning