академия больших данных

mail.ru group

Meta learning & Semi-supervised learning

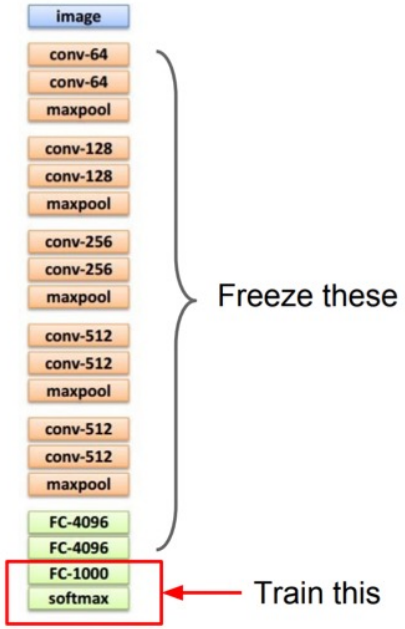
Андрей Бояров

Ведущий программист-исследователь в командах Машинного обучения почты и Машинного зрения Mail.ru, кандидат физ.-мат. наук

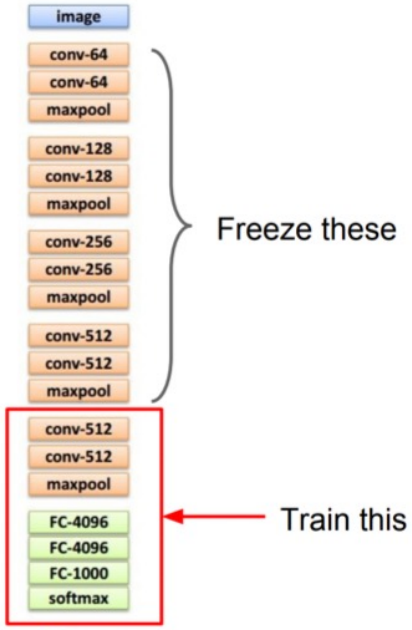
Recap: Transfer learning

1. Обучим нейронную сеть на базе ImageNet

2. Мало новых данных



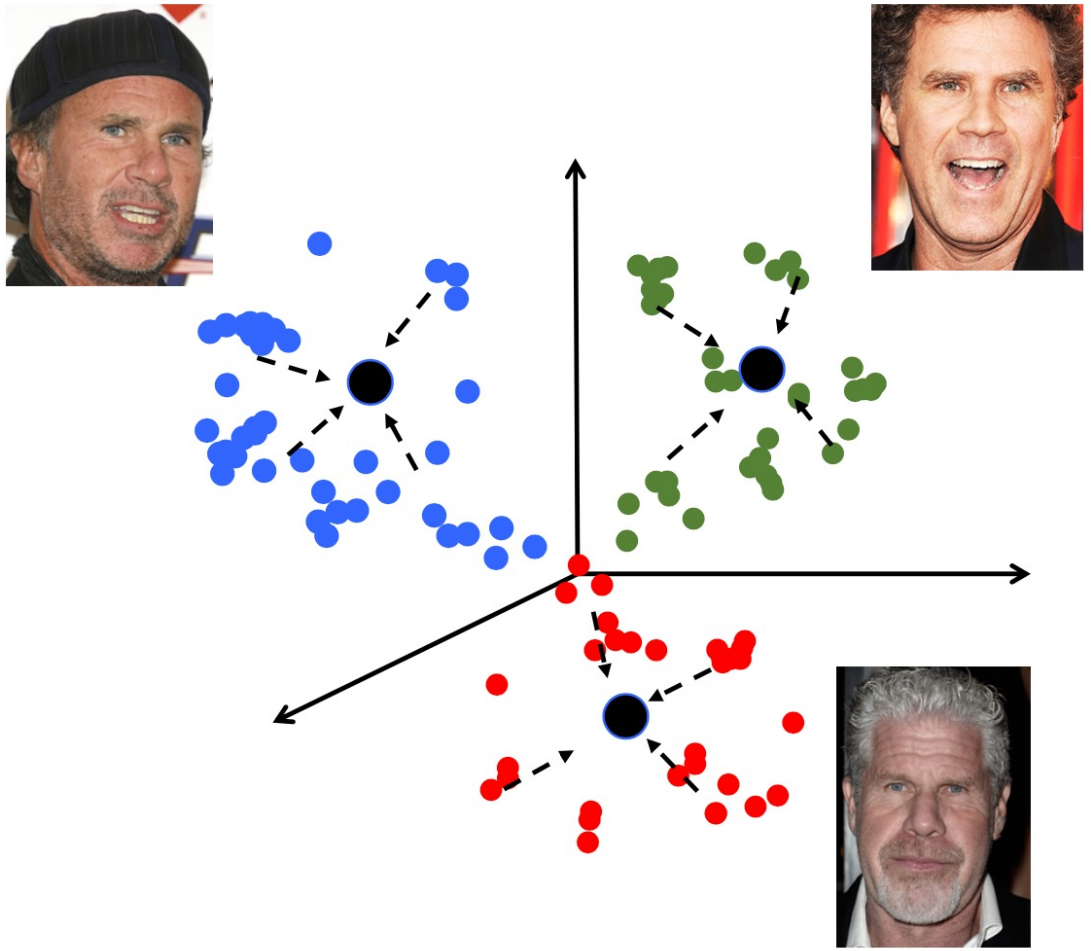
3. Среднее количество новых данных



Небольшой learning rate

http://cs231n.stanford.edu/slides/2016/winter1516_lecture11.pdf

Recap: Metric learning

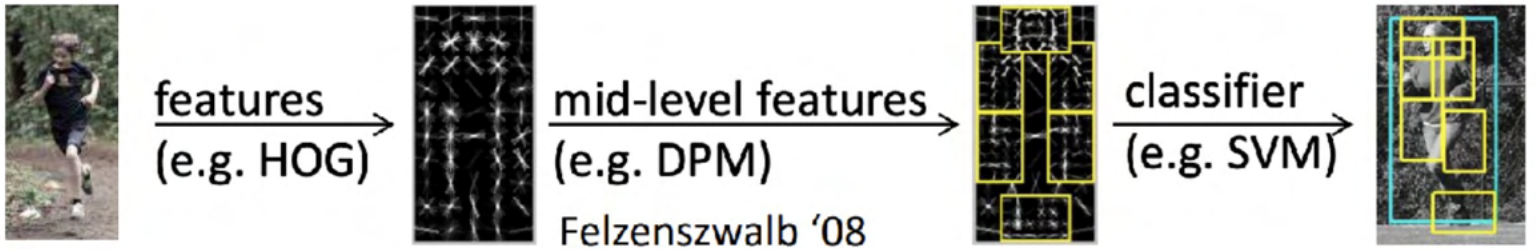


Обучение с учителем

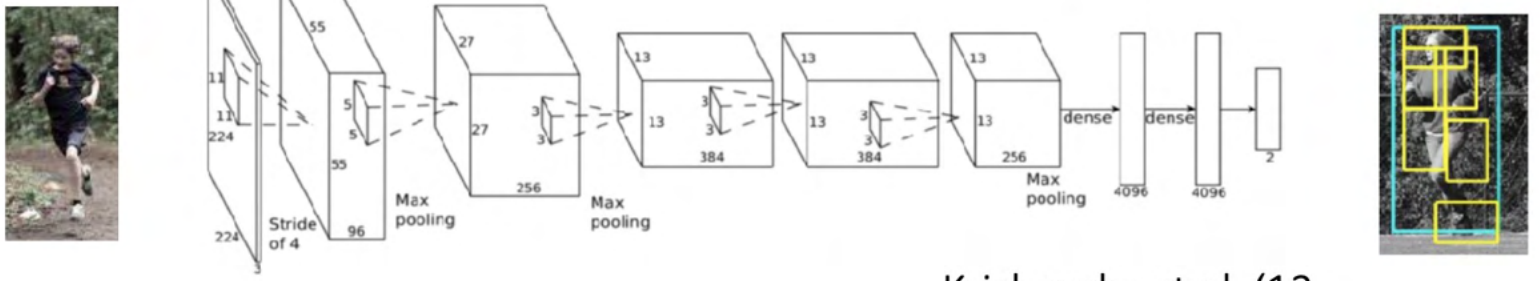
- Наиболее успешные алгоритмы DL связаны с обучением с учителем
- Задачи:
  - Computer vision
  - Text
  - Speech
  - Reinforcement Learning (с некоторой поправкой)

Deep Learning

Standard computer vision:
hand-designed features



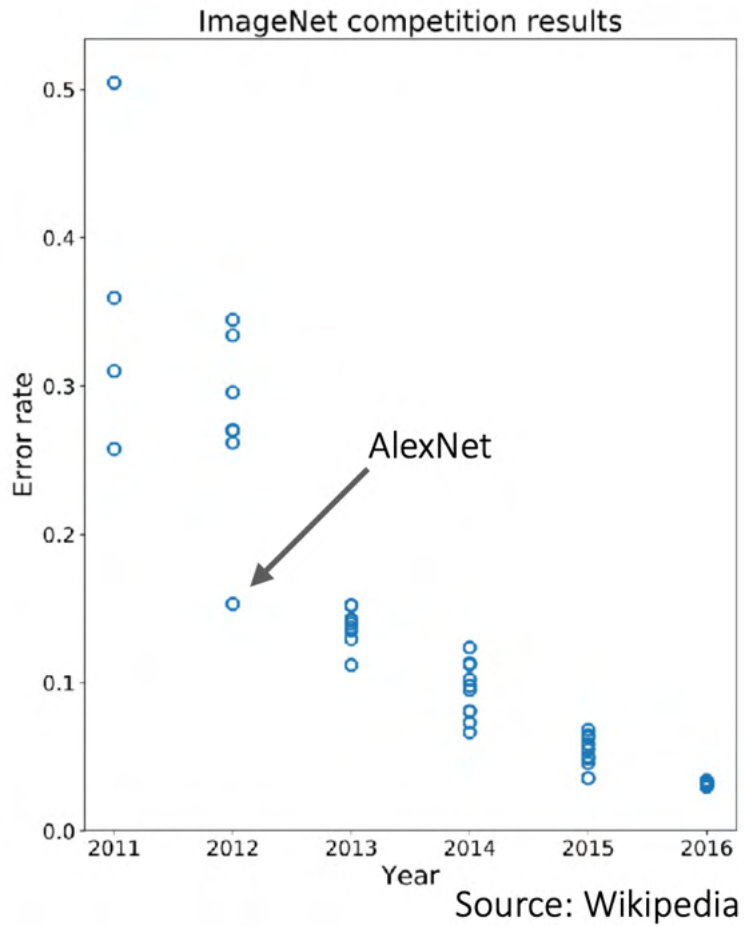
Modern computer vision:
end-to-end training



Krizhevsky et al. '12

Deep Learning

Deep learning for object classification



Source: Wikipedia

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,zhifengc,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 для успешного Deep Learning

Large, diverse data
(+ large models)

deep learning →

Broad generalization

Russakovsky et al. ‘14

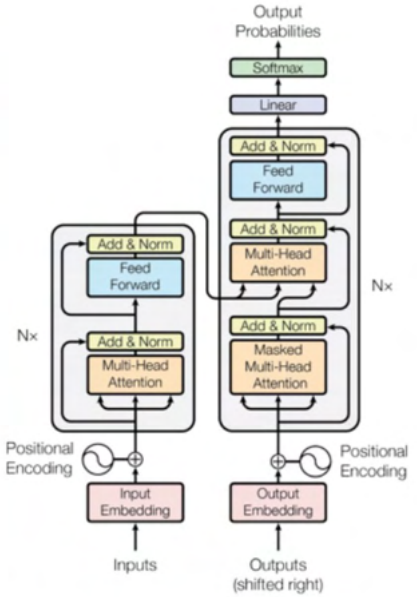


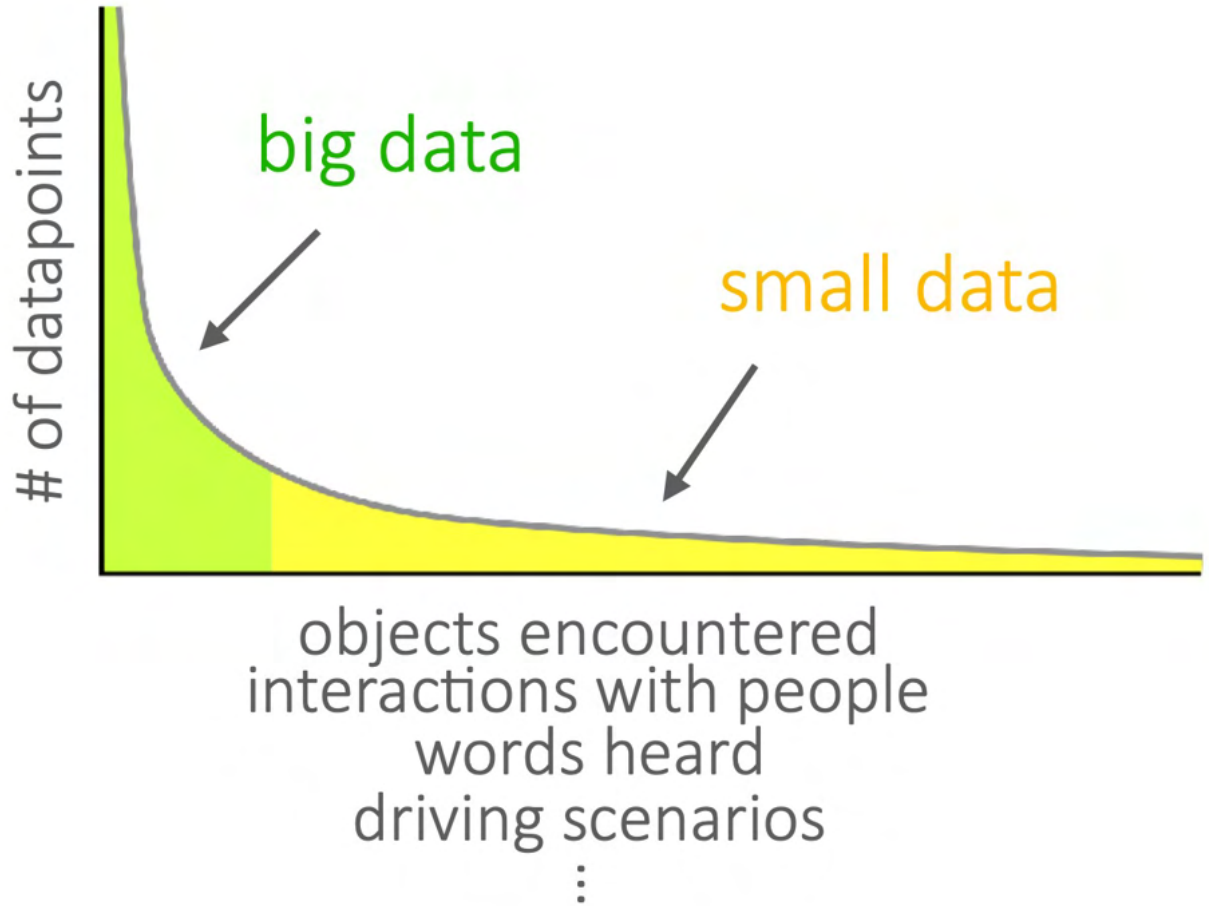
Figure 1: The Transformer - model architecture.

Vaswani et al. ‘18

GPT-2

Radford et al. ‘19

Big data для успешного Deep Learning



http://web.stanford.edu/class/cs330/

Big data vs Not so big data

ImageNet	1.2 million images and labels
WMT '14 English - French	40.8 million paired sentences
Switchboard Speech Dataset	300 hours of labeled data
Kaggle's Diabetic Retinopathy Detection dataset	35K labeled images
Adaptive epilepsy treatment with RL Guez et al. '08	< 1 hour of data
Learning for robotic manipulation Finn et al. '16	< 15 min of data

Сложности при сборе выборки

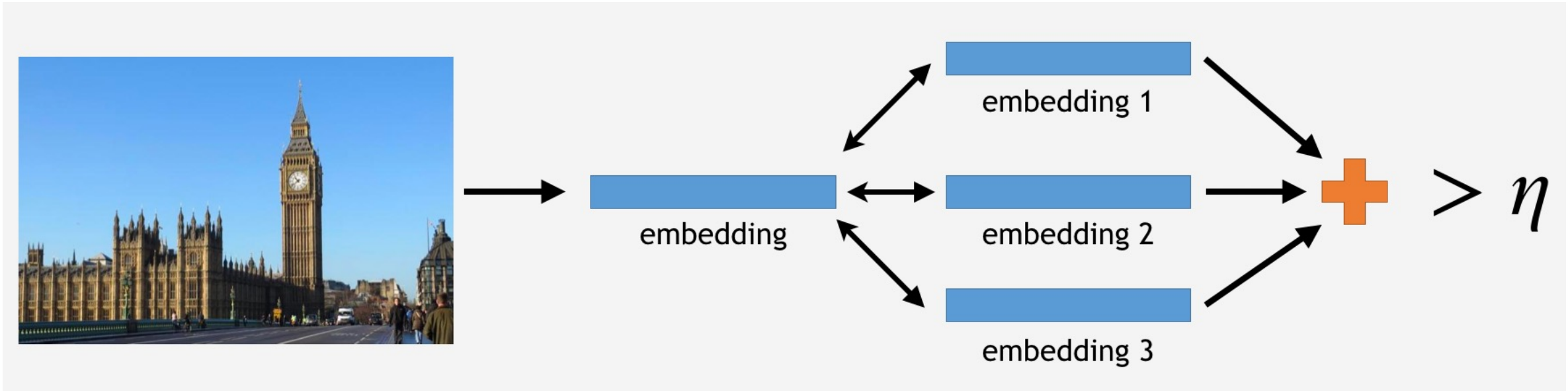
- Под каждую задачу надо собирать и размечать данные
- Сбор и расчистка данных
- Разметка данных
- Затратно по человеческим ресурсам

Вариант: автоматическая чистка

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

Boiarov and Tyantov, Large scale landmark recognition via deep metric learning, 2019

Вариант: автоматическая чистка



Boiarov and Tyantov, Large scale landmark recognition via deep metric learning, 2019

Решение

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

Meta Learning

Meta Learning

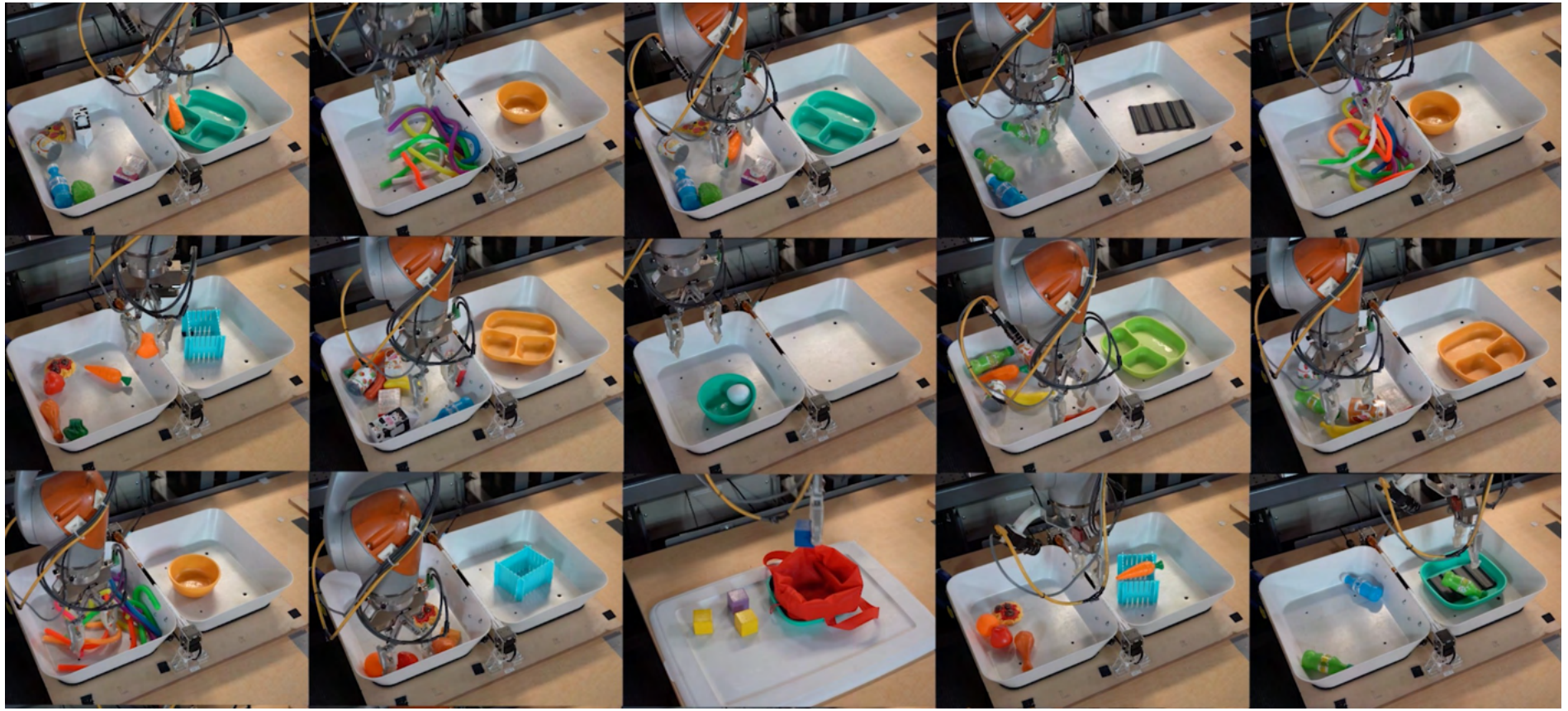
- Класс алгоритмов, которые “учатся учиться”
- Т.е. наиболее эффективно адаптироваться под новые задачи



Meta Learning

https://ai.googleblog.com/2021/04/multi-task-robotic-reinforcement.html

Meta Learning



https://ai.googleblog.com/2021/04/multi-task-robotic-reinforcement.html

“Наиболее эффективное” обучение

Адаптация под новую задачу по малому количеству тренировочных данных по ней

- Meta reinforcement learning (robotics)
- Few-shot learning (pattern recognition)

Few-shot Learning

Few-shot learning

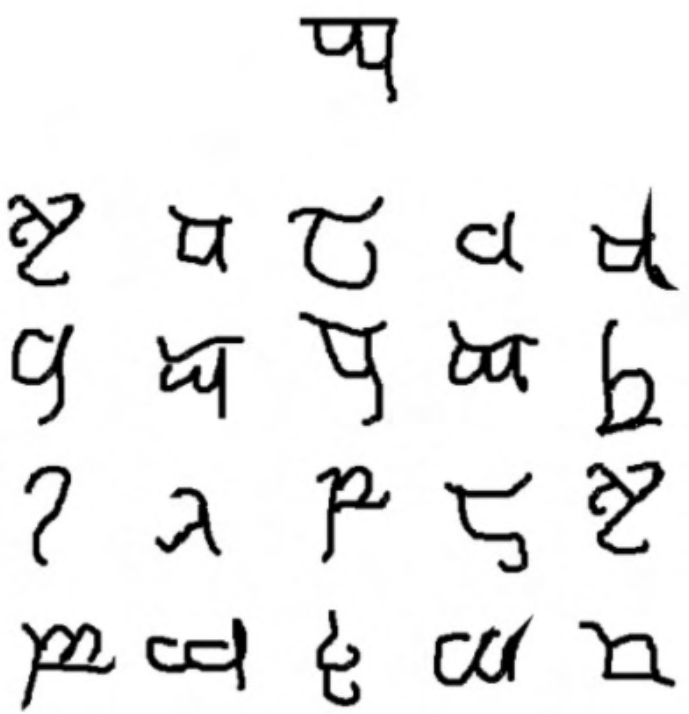
training data

Braque | Cezanne

test datapoint

By Braque or Cezanne?

Few-shot learning



Omniglot dataset: 1-shot 20-way classification.

Lake et al., Human-level concept learning through probabilistic program induction, 2015

Omniglot dataset

Lake et al., Human-level concept learning through probabilistic program induction, 2015

Omniglot dataset

1623 characters from **50 different alphabets**

Hebrew

Bengali

Greek

Futurama

...

20 instances of each character

Few-shot learning (формально)

Пусть C — количество классов, а N — количество примеров на каждый класс в наборе размеченных данных $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{CN}, y_{CN})\}$, где $\mathbf{x}_i \in \mathbb{R}^d$ является вектором-примером, а $y_i \in \{1, \ldots, C\}$ — 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 $\xi_t : (t_1, \ldots, t_M)$ 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

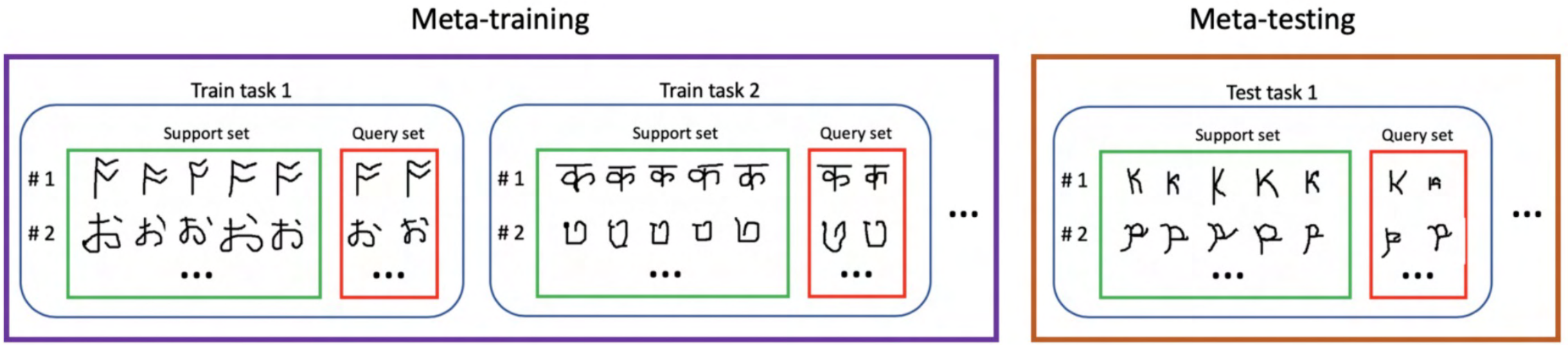
$$S_{t_i} = \{S_{t_i}^k\}_{k=1}^{N_C}, Q_{t_i} = \{Q_{t_i}^k\}_{k=1}^{N_C}, S_{t_i}^k \cap Q_{t_i}^k = \emptyset.$$

Множества

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

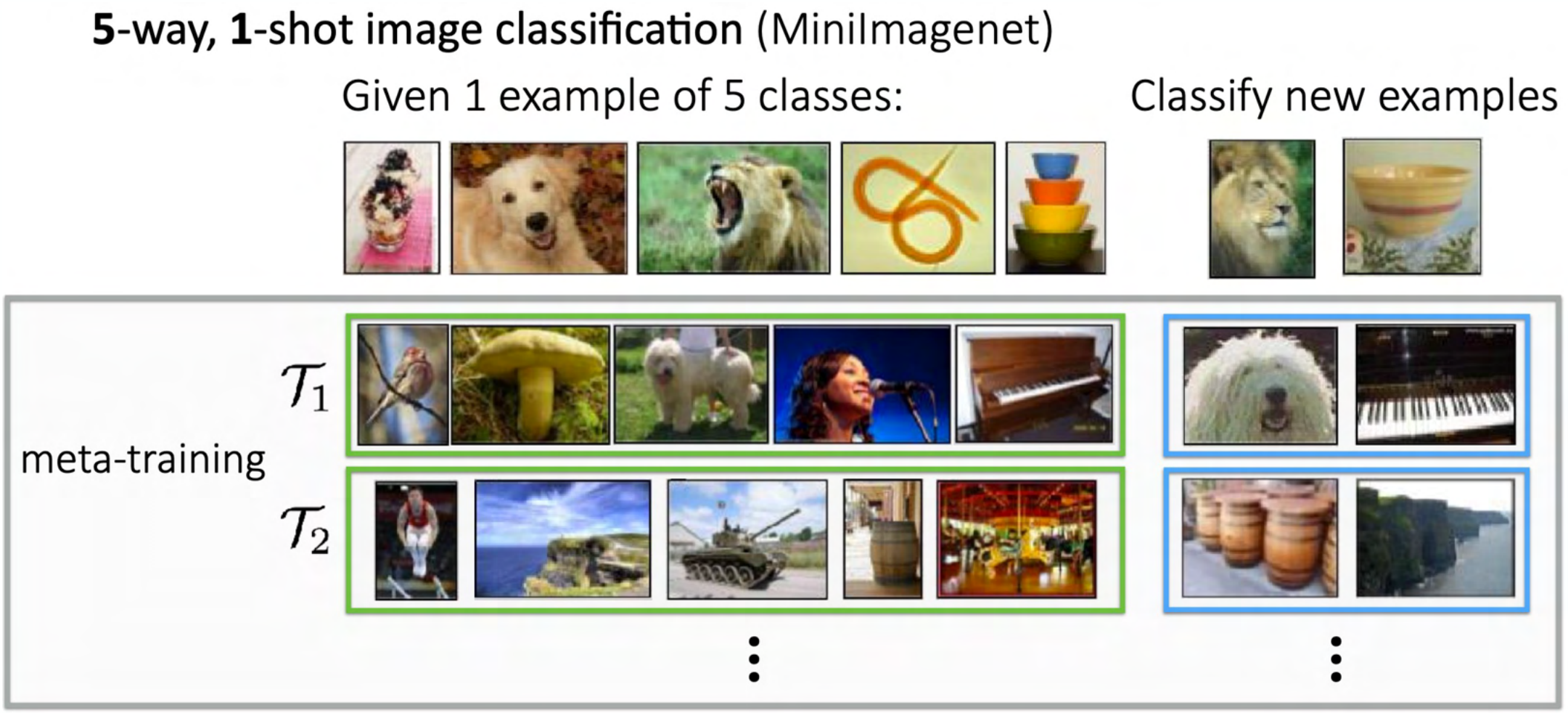
случайным образом выбираются из представителей класса k.

Few-shot learning pipeline



Boiarov et al., Simultaneous Perturbation Stochastic Approximation for Few-Shot Learning, 2020

Few-shot learning pipeline



Ravi et al., Optimization as a model for few-shot learning, 2016

Meta-parameters

Все параметры модели делятся на два типа:

1. Общие мета-параметры θ
2. Параметры, специфичные для конкретной задачи ϕ

Meta-parameters

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

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

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

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

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

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

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

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

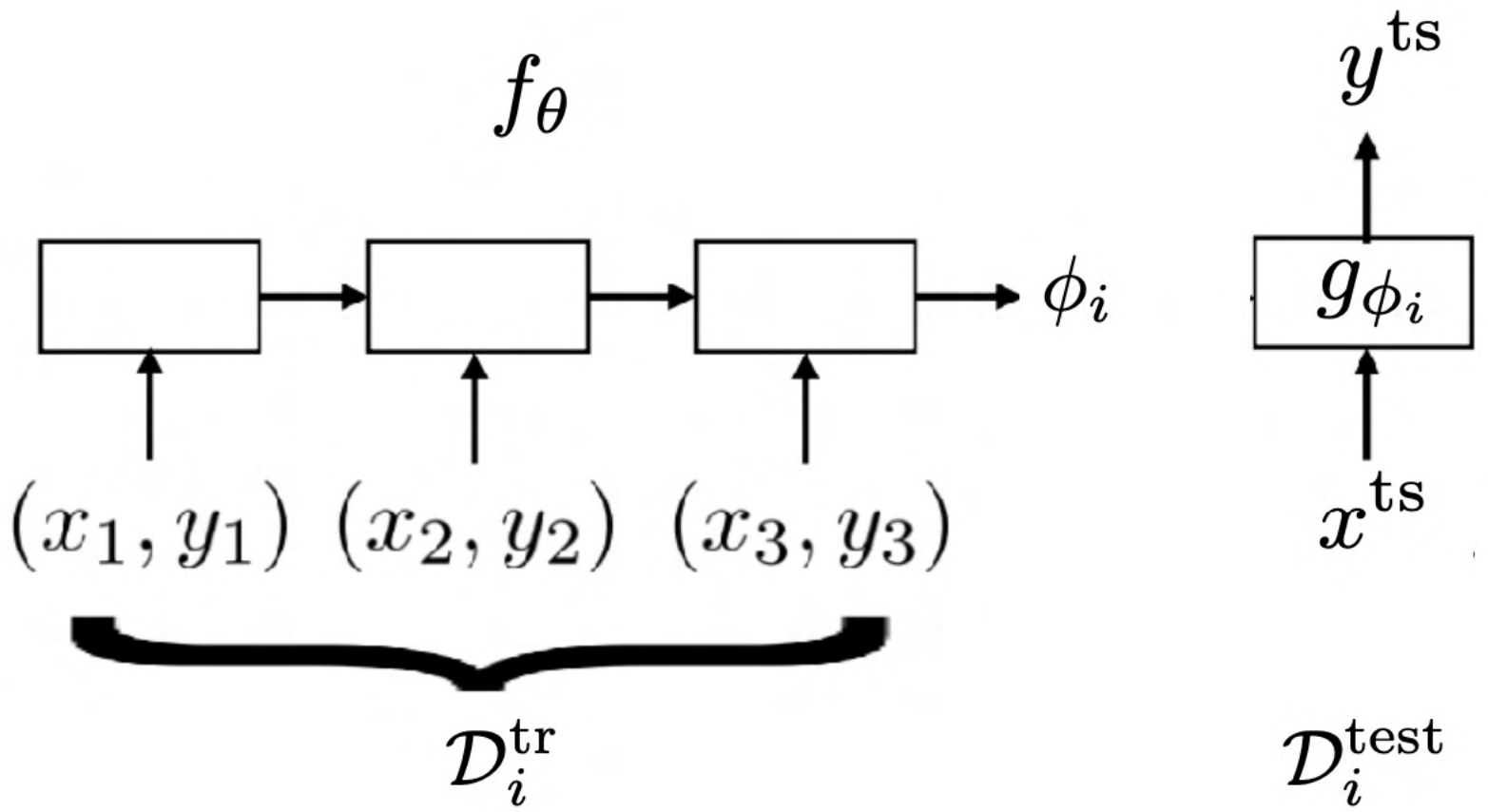
$$\text{where } \phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$$

Основные типы алгоритмов

- Black-box
- Metric-based
- Optimization based

Black-box

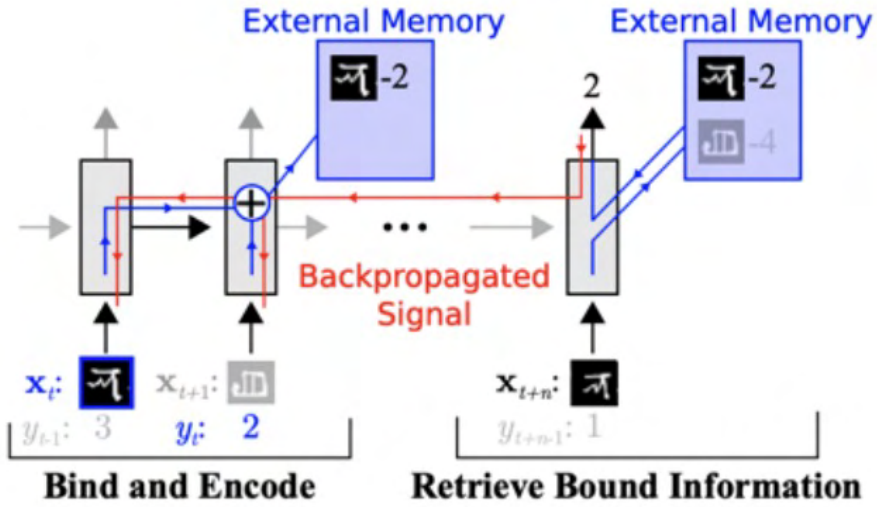
Основная идея



http://web.stanford.edu/class/cs330/

Примеры black-box

LSTMs or Neural turing machine (NTM)

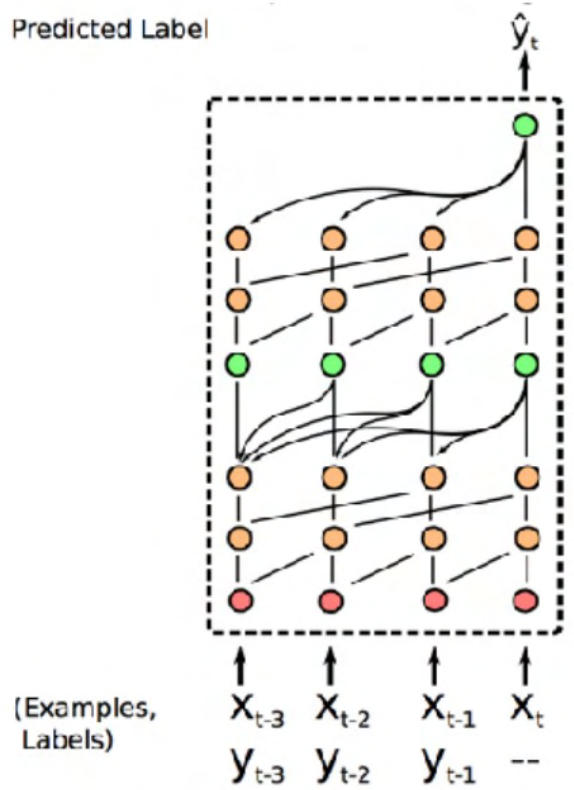


Meta-Learning with Memory-Augmented Neural Networks

Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML ‘16

Примеры black-box

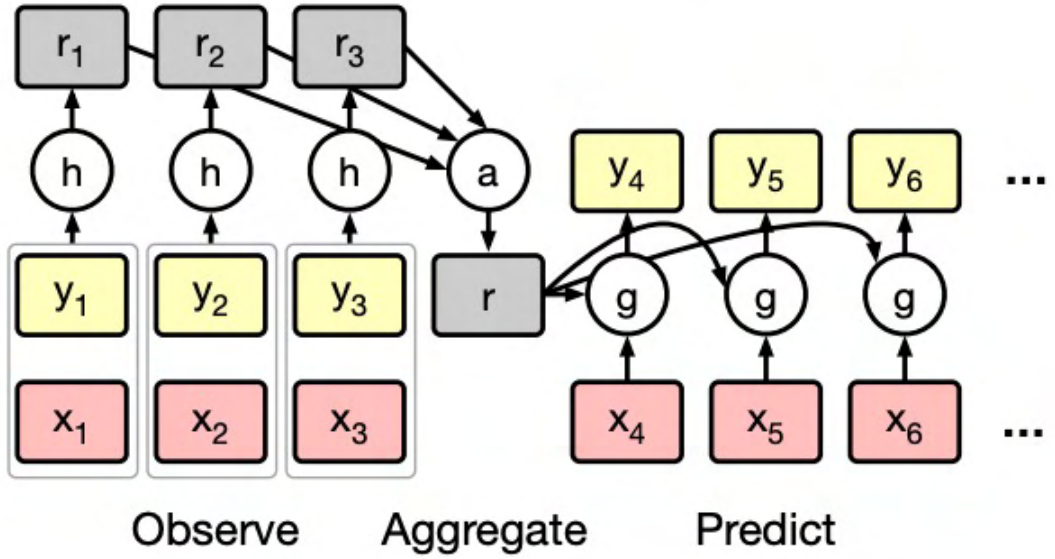
Convolutions & attention



A Simple Neural Attentive Meta-Learner

Mishra, Rohaninejad, Chen, Abbeel. ICLR ‘18

Примеры black-box



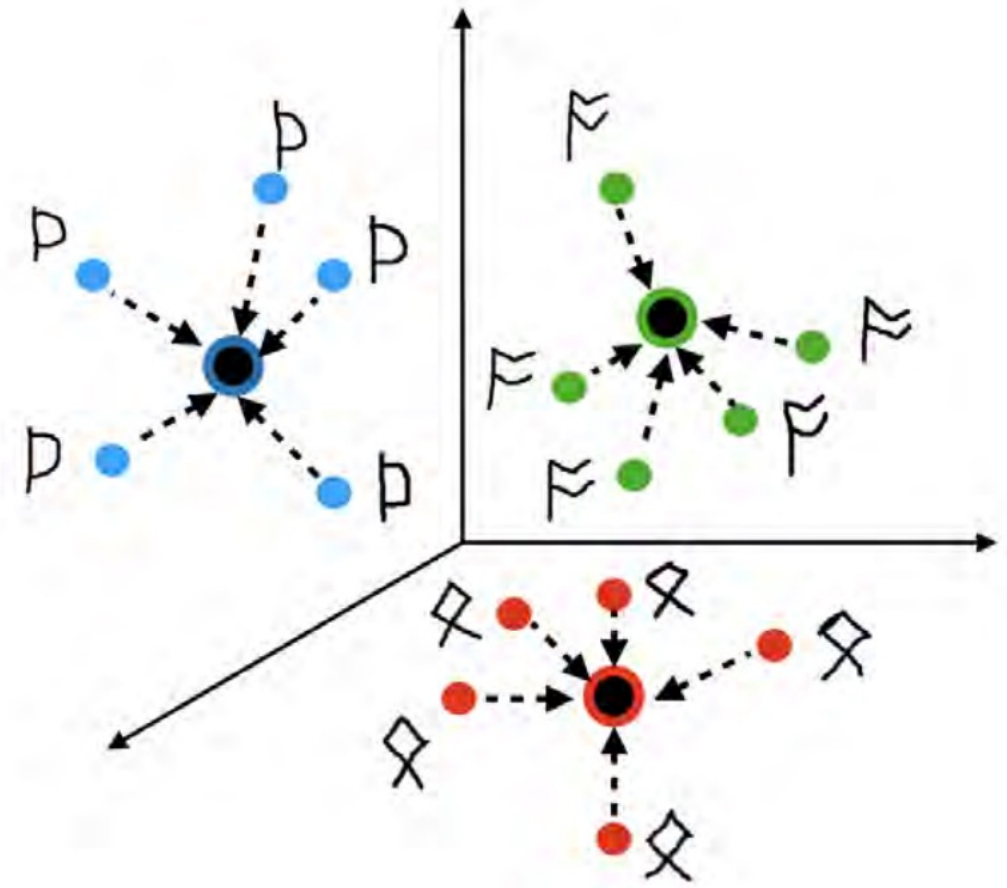
Conditional Neural Processes, Garnelo, et al., ICML'18

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 $\phi_\theta(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^n$ — 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 for Few-shot Learning, Snell et al., NIPS'17

Prototypical Networks

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

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

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

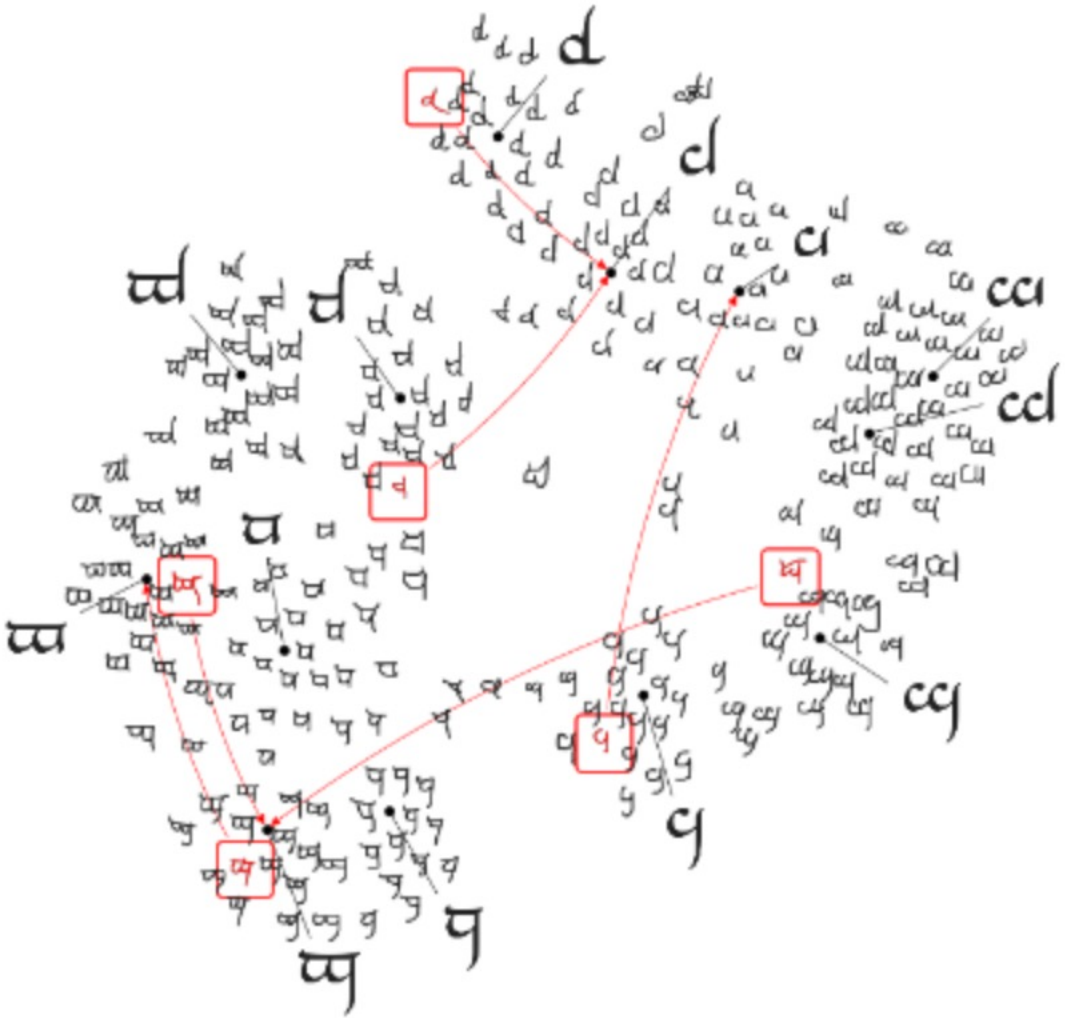
Алгоритм 1. Обучение для эпизода $\xi_t : (t_1)$.

Вход: N_S, N_Q, N_C

Выход: Обновлённые параметры θ

1: Случайно выбирается N_C классов
2: **for** $k \in \{1, \ldots, N_C\}$ **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** $k \in \{1, \ldots, N_C\}$ **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 обновляется с помощью стохастического градиентного спуска по $\mathcal{L}_{\theta,t_1}$

Прототипы



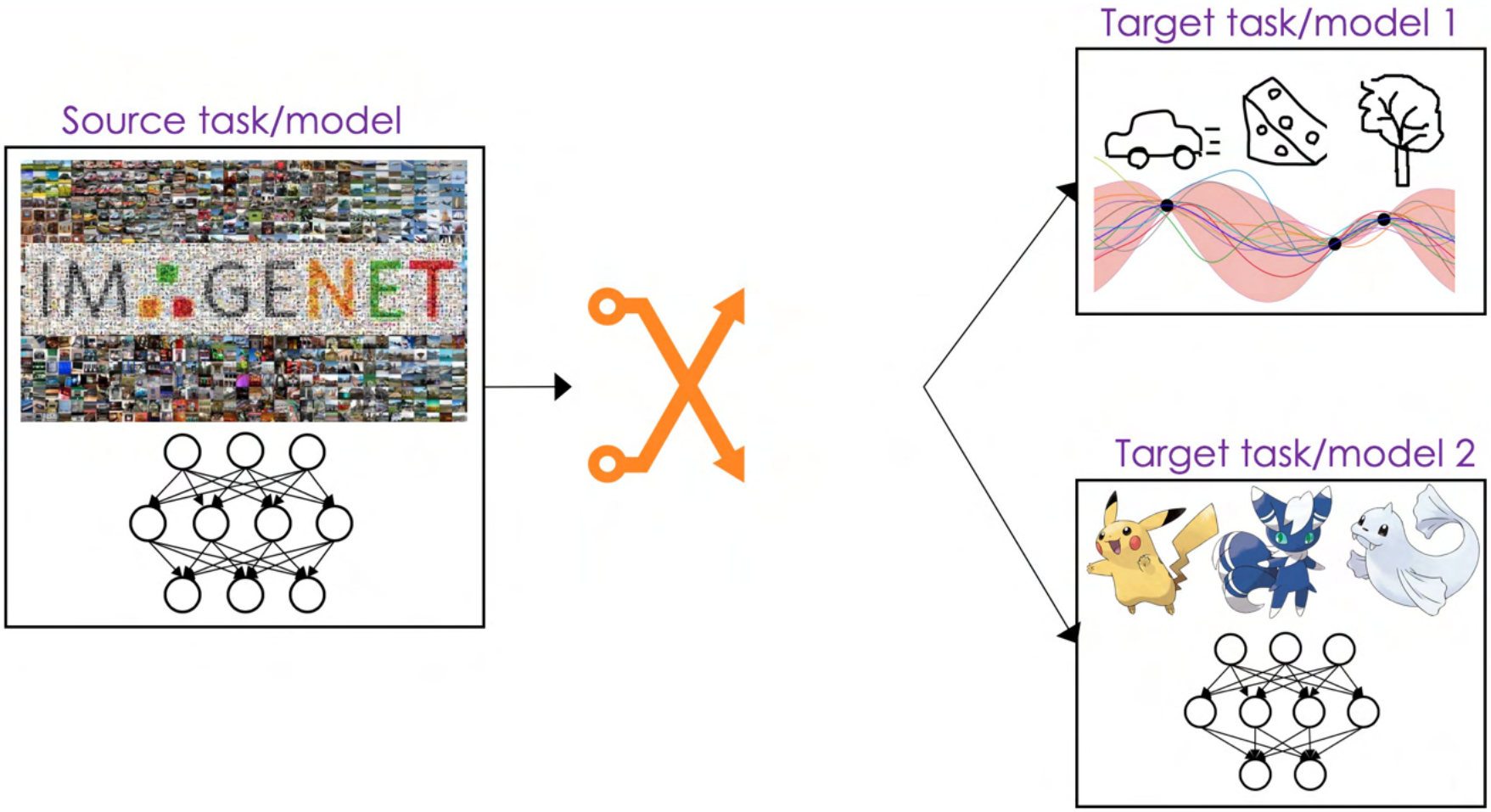
https://github.com/orobix/Prototypical-Networks-for-Few-shot-Learning-PyTorch

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

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

pre-trained parameters

training data for new task

(typically for many gradient steps)

Transfer learning based

Fine-tuning

[test-time]

pre-trained parameters

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

training data for new task

Meta-learning

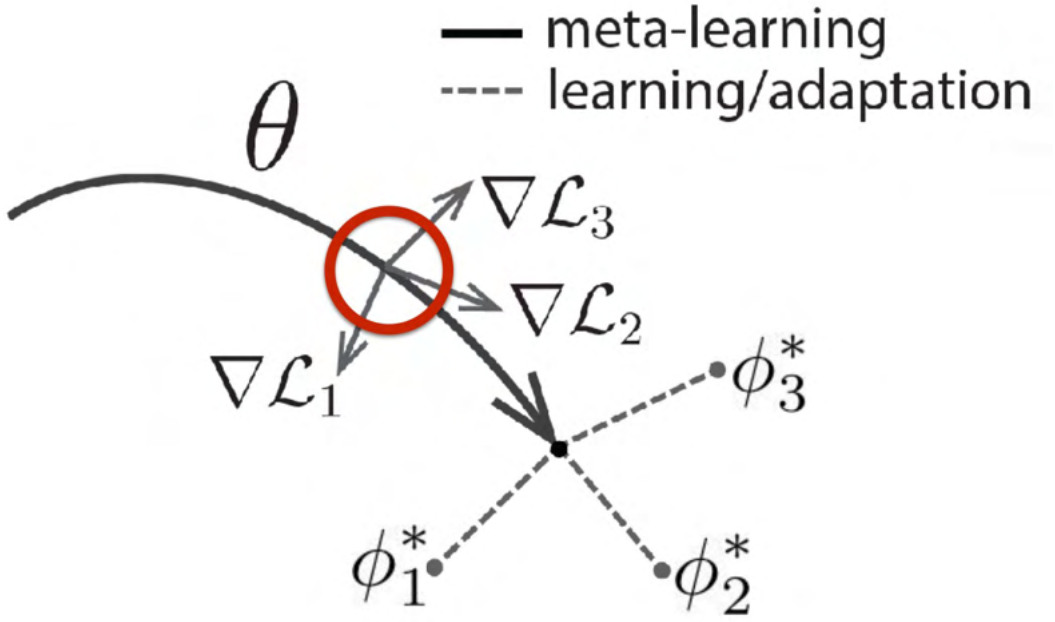
$$\min_\theta \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\mathrm{tr}}), \mathcal{D}_i^{\mathrm{ts}})$$

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



Model-Agnostic Meta-Learning, Finn et al., ICML'17

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

MiniImageNet 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

Metric-based + Optimization-based

1. Обучаем CNN как в ProtoNets
2. Используем embedding-слой из ProtoNets для инициализации нового полносвязного слоя
3. Обучаем получившуюся модель в стиле MAML
4. …
5. Proto-MAML

Meta-dataset

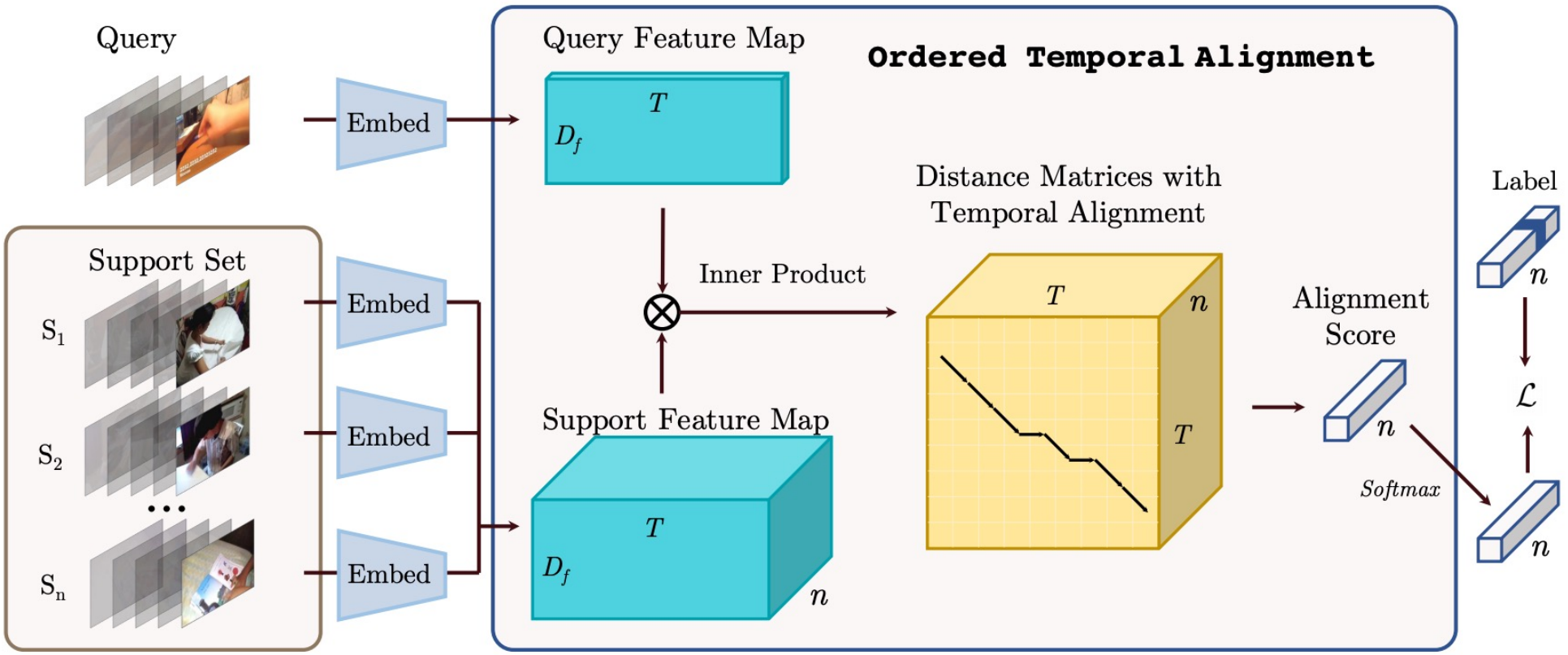
Table 1: Few-shot classification results on META-DATASET using models **trained on ILSVRC-2012 only (top)** and **trained on all datasets (bottom)**.

Test Source	k-NN	Finetune	MatchingNet	ProtoNet	fo-MAML	RelationNet	fo-Proto-MAML
ILSVRC	41.03	45.78	45.00	**50.50**	45.51	34.69	**49.53**
Omniglot	37.07	60.85	52.27	59.98	55.55	45.35	**63.37**
Aircraft	46.81	**68.69**	48.97	53.10	56.24	40.73	55.95
Birds	50.13	57.31	62.21	**68.79**	63.61	49.51	**68.66**
Textures	66.36	**69.05**	64.15	66.56	**68.04**	52.97	66.49
Quick Draw	32.06	42.60	42.87	48.96	43.96	43.30	**51.52**
Fungi	36.16	38.20	33.97	**39.71**	32.10	30.55	**39.96**
VGG Flower	83.10	85.51	80.13	85.27	81.74	68.76	**87.15**
Traffic Signs	44.59	**66.79**	47.80	47.12	50.93	33.67	48.83
MSCOCO	30.38	34.86	34.99	41.00	35.30	29.15	**43.74**
Avg. rank	5.7	2.9	4.65	2.65	3.7	6.55	**1.85**

Test Source	k-NN	Finetune	MatchingNet	ProtoNet	fo-MAML	RelationNet	fo-Proto-MAML
ILSVRC	38.55	43.08	36.08	44.50	37.83	30.89	**46.52**
Omniglot	74.60	71.11	78.25	79.56	83.92	**86.57**	82.69
Aircraft	64.98	72.03	69.17	71.14	**76.41**	69.71	75.23
Birds	66.35	59.82	56.40	67.01	62.43	54.14	**69.88**
Textures	63.58	**69.14**	61.80	65.18	64.16	56.56	**68.25**
Quick Draw	44.88	47.05	60.81	64.88	59.73	61.75	**66.84**
Fungi	37.12	38.16	33.70	40.26	33.54	32.56	**41.99**
VGG Flower	83.47	85.28	81.90	86.85	79.94	76.08	**88.72**
Traffic Signs	40.11	**66.74**	55.57	46.48	42.91	37.48	52.42
MSCOCO	29.55	35.17	28.79	39.87	29.37	27.41	**41.74**
Avg. rank	5.05	3.6	4.95	2.85	4.25	5.8	**1.5**

Triantafillou et al., Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples, 2020

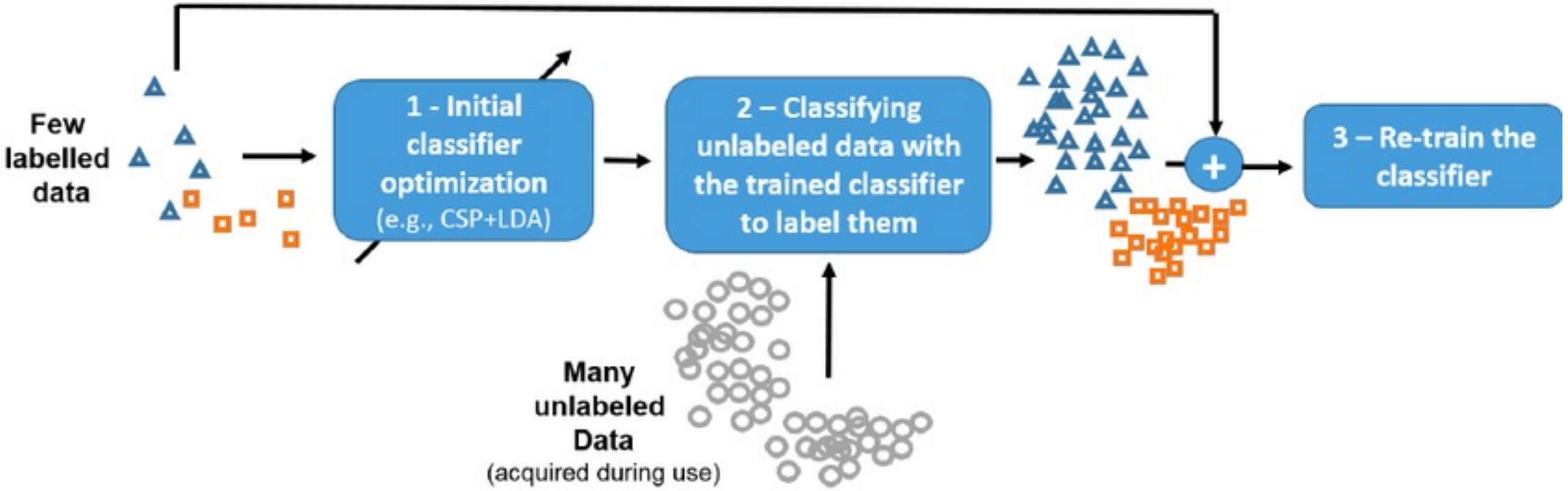
Few-shot learning for video



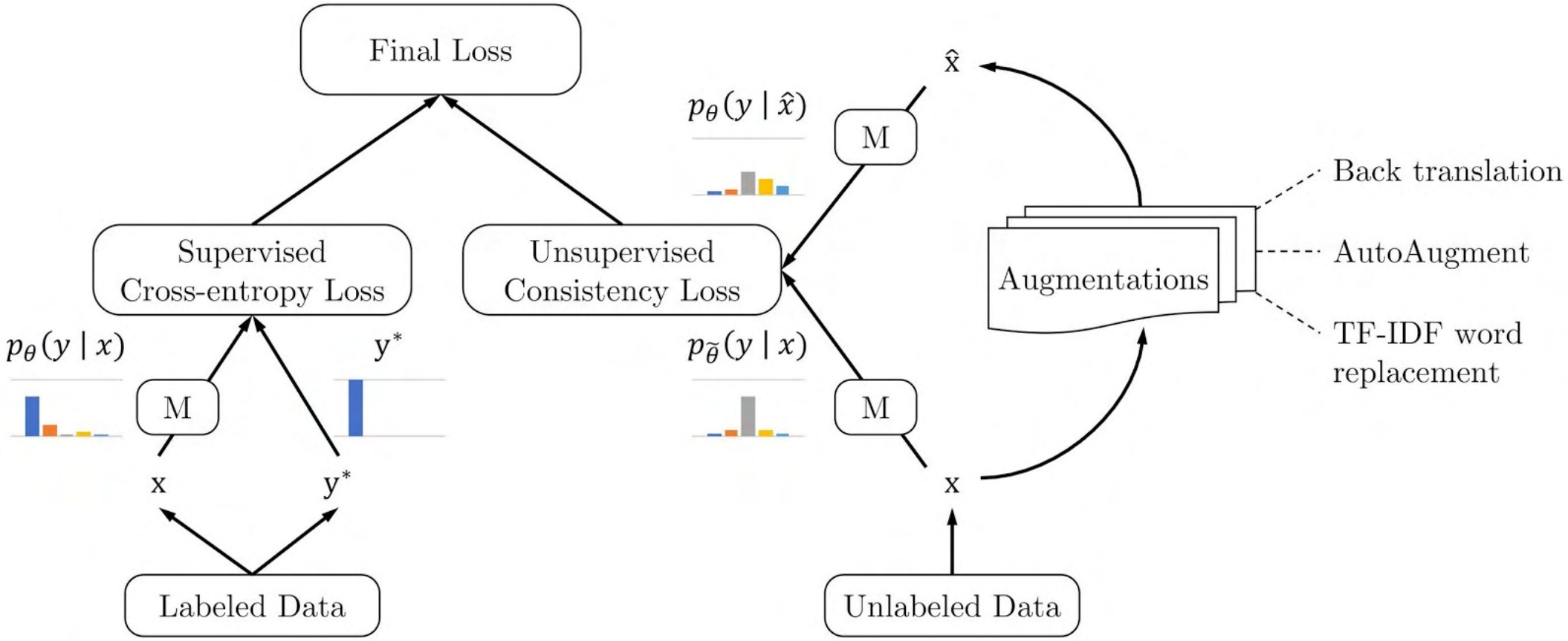
Cao et al., Few-Shot Video Classification via Temporal Alignment, 2020

Semi-supervised Learning

Semi-supervised learning idea



Unsupervised Data Augmentation



Xie et al, Unsupervised Data Augmentation for Consistency Training, 2019

SimCLR

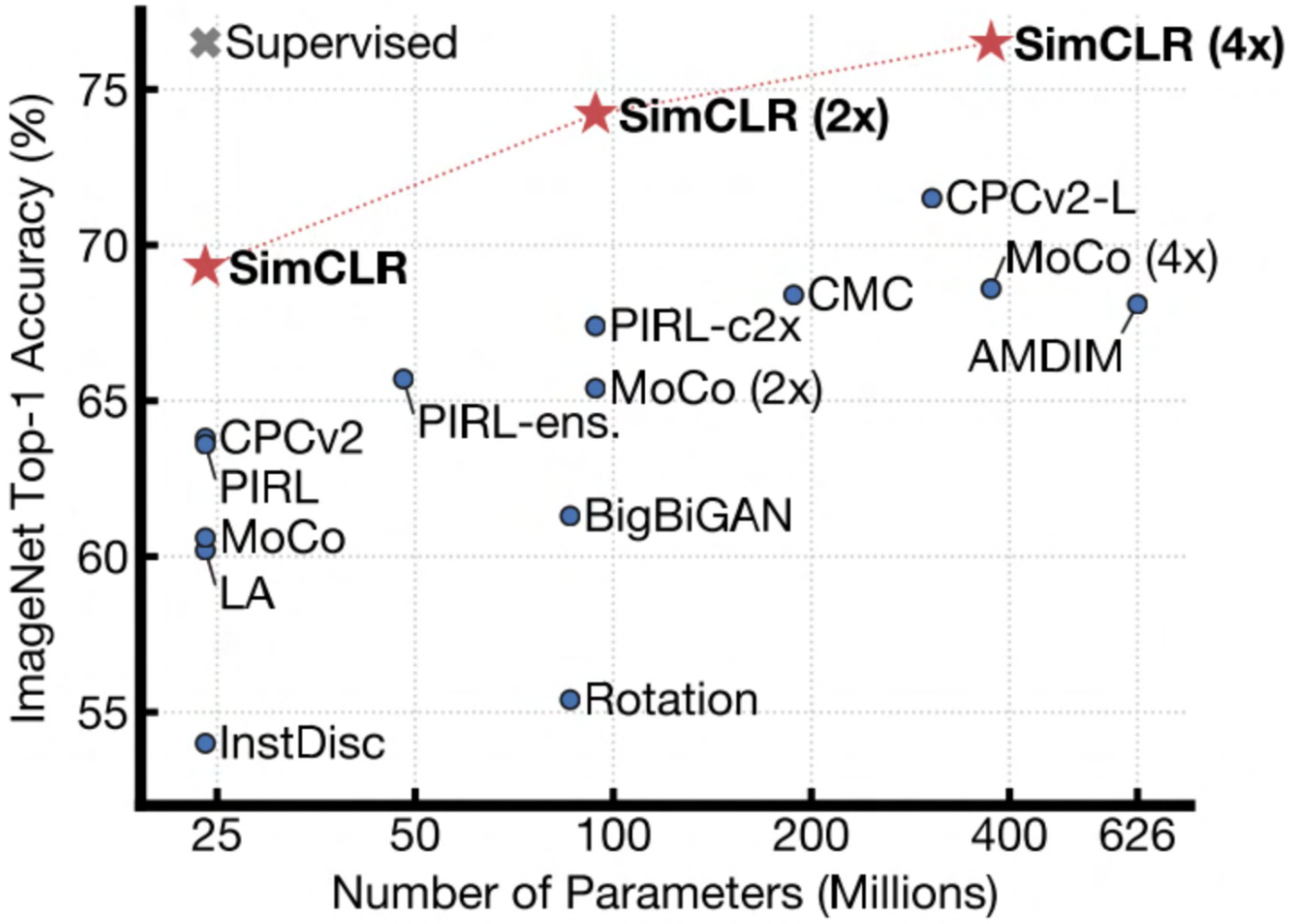
https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html

SimCLR: Noise Contrastive Estimator Loss

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

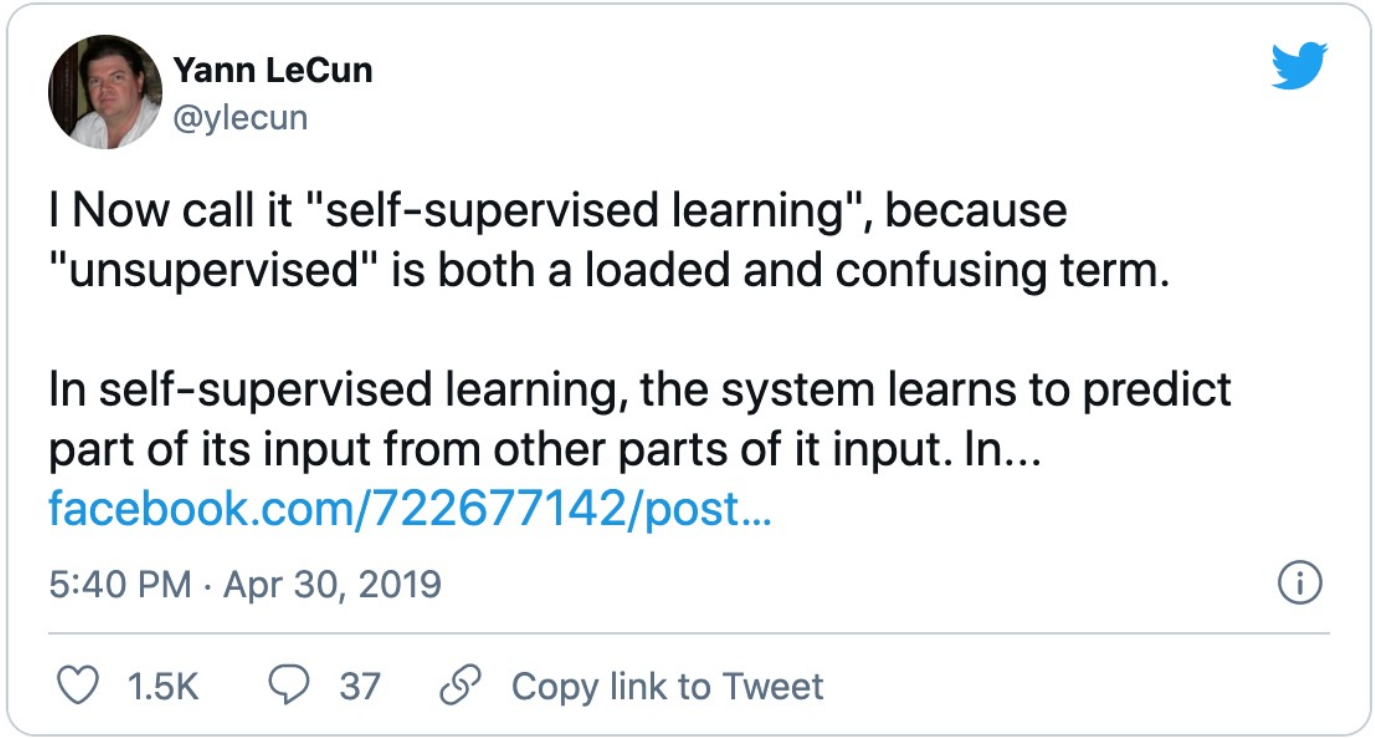
Chen et al, A Simple Framework for Contrastive Learning of Visual Representations, 2020

SimCLR Performance



Semi-supervised vs Self-supervised

- Semi-supervised: есть размеченный и неразмеченный наборы данных
- Self-supervised: разметка генерируется автоматически



Заключение

- Во многих задачах мало размеченных данных
- Если будут добавляться новые классы, то Few-shot Learning
  - Black-box
  - Metric-based
  - Optimization-based
- Если есть неразмеченный данные
  - Semi-supervised Learning

Курсовые проекты

- Проект 2020: Deep Multi-Task Few-Shot Learning
- Boiarov et al., Simultaneous Perturbation Stochastic Approximation for Few-Shot Learning, 2020
- В топ-3 всех проектов за 2020 год
- Почта: a.boiarov@corp.mail.ru, Discord