Self-training with Noisy Student improves ImageNet classification

Бондаренко Наталия, БПМИ171

Проблема

До сих пор используем только обучение с учителем, для этого нужно много размеченных данных (особенно для “сложных” датасетов). При этом имеем много больше неразмеченных данных, которые бы хотели использовать для обучения тоже.

Сеть-учитель и сеть-студент

- Обучили сеть-учителя
- Использовали для генерации меток к 300 миллионам неразмеченных изображений
- Добавили их к датасету и обучили сеть-студента на нем
- Использовали сеть-студента как учителя

Шумный студент

Сеть-студент должна быть **зашумленной**, в то время как сеть-учитель должна быть **незашумленной.**

Источники шума:

- RandAugment
- дропаут
- стохастическая глубина

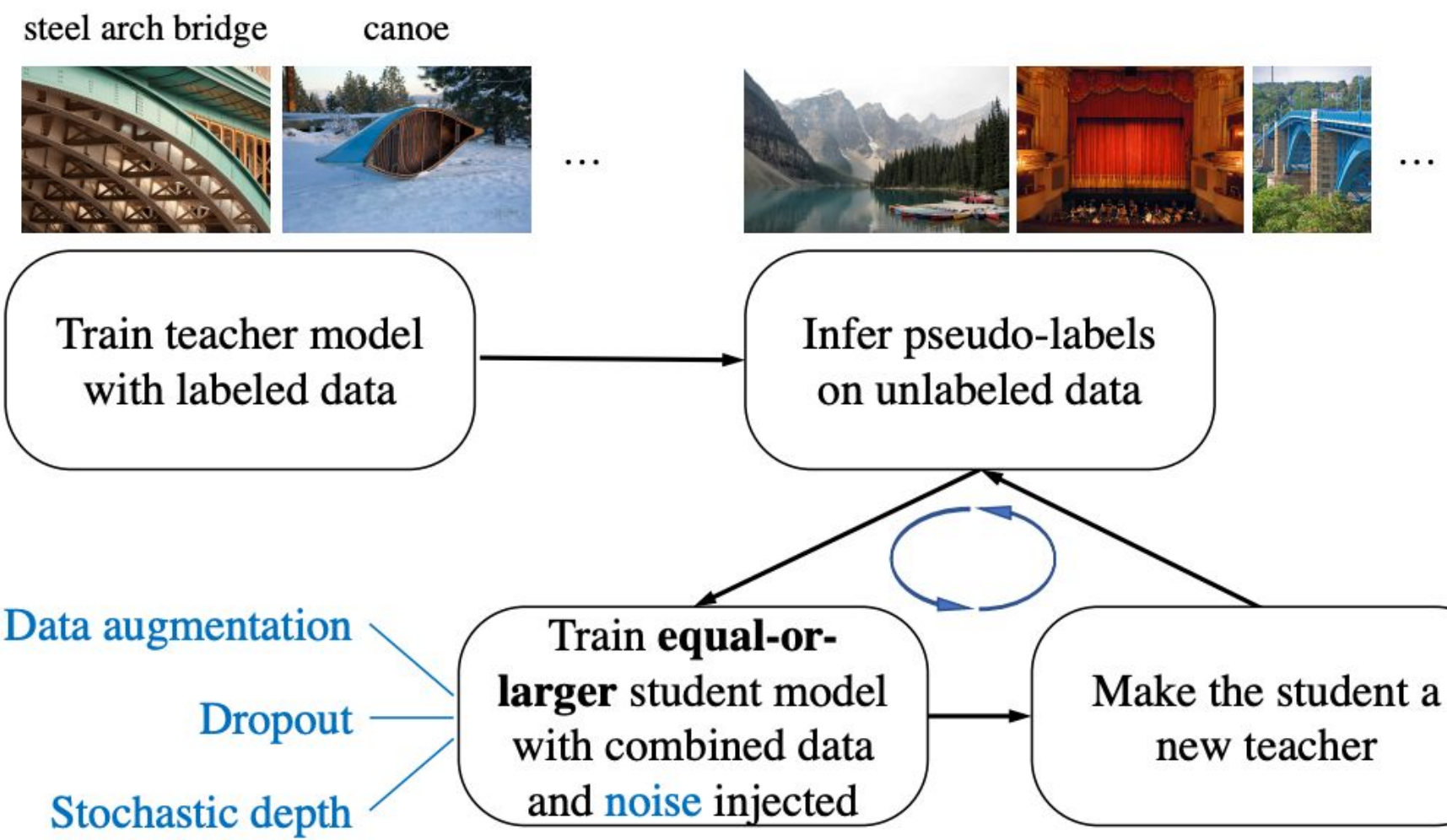
Алгоритм

- **Вход**: размеченные изображения $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ и неразмеченные изображения $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.
- Обучение модели-учителя θ_*^t, функция потерь -- стандартная кросс-энтропия:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t))$$

- Использование незашумленной модели-учителя для генерации меток для неразмеченных изображений $\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \cdots, m$
- Обучаем **такую же или бóльшую** модель-студента с добавленным шумом, функция потерь -- кросс-энтропия:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$


steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Train **equal-or-larger** student model with combined data and noise injected
Make the student a new teacher
Data augmentation
Dropout
Stochastic depth

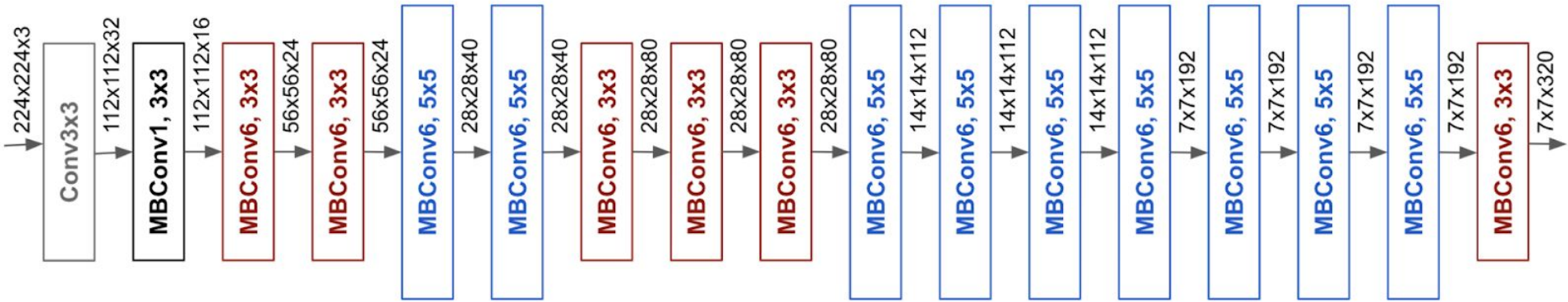
Отличия от других моделей

- Больше шума в студенте
- Модель студент больше (по крайней мере не меньше) модели-учителя

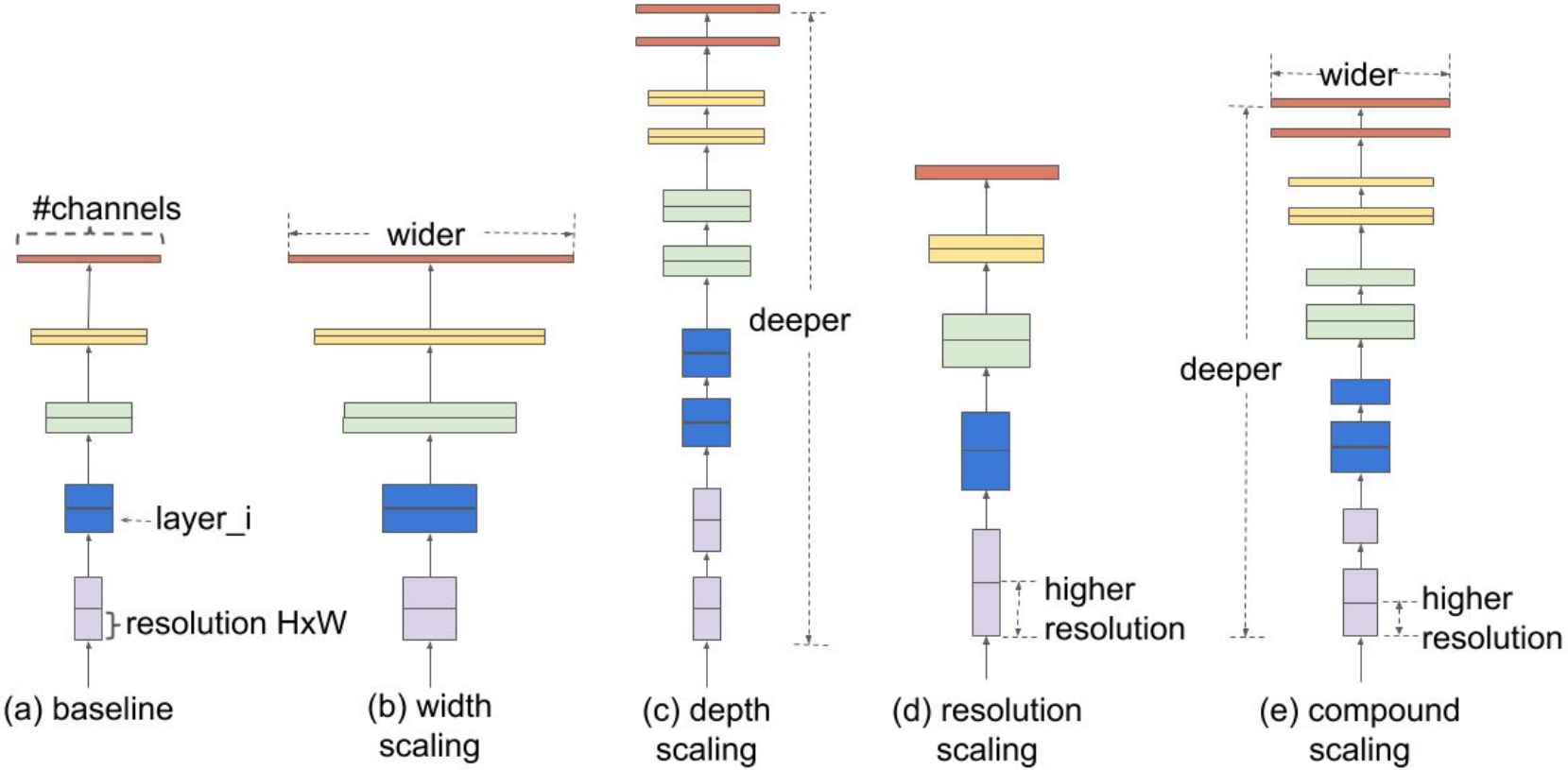
Похоже на **дистилляцию**, только дистиллируют не в меньшую сеть, а наоборот.

EfficientNets

- Основаны на архитектуре MobileNetV2 и MnasNet
- Исходя из возможностей, определяют оптимальные ширину, глубину и разрешение изображения
- Масштабируют сеть



EfficientNets



Эксперименты. Работа с данными

- Неразмеченные данные из набора JFT-300M (метки проигнорировали)
- Запустили EfficientNet-B0, обученный на ImageNet, чтобы разметить
- Выбрали изображения, в классе которых модель уверена больше, чем на 0.3
- В каждом классе -- не более 130К изображений
- Продублировали некоторые случайные изображения в классах, где их меньше 130К
- Получили 130М изображений (из них 81М уникальных)

Архитектура и эксперименты

- EfficientNets -- базовая модель
- Размер батчей -- 2048 (обнаружили, что 512 и 1024 приводит к той же производительности)
- Большие модели-студенты (больше EfficientNet-B4) -- 350 эпох, модели поменьше -- 700 эпох.
- Сначала обучают с меньшим разрешением 350 эпох, потом точно настраивают модель на изображениях с большим разрешением и без аугментации 1.5 эпохи
- Наибольшая модель тренировалась 6 дней на Cloud TPU v3 Pod с 2048 ядрами, когда размер неразмеченных батчей был в 14 раз больше размера размеченных

Результаты

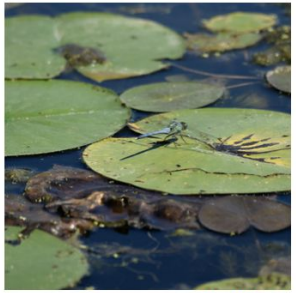
	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
Ours	**88.4%**	**83.7%**	**28.3**	**12.2**

sea lion lighthouse

submarine canoe

snow leopard electric ray

swing mosquito net



dragonfly bullfrog

starfish wreck

toaster pill bottle

gown ski

hummingbird bald eagle

basketball parking meter

parking meter vacuum

cannon television

(a) ImageNet-A

plate rack refrigerator

racing car car wheel

plate rack medicine chest

racing car fire engine

plate rack medicine chest

racing car car wheel

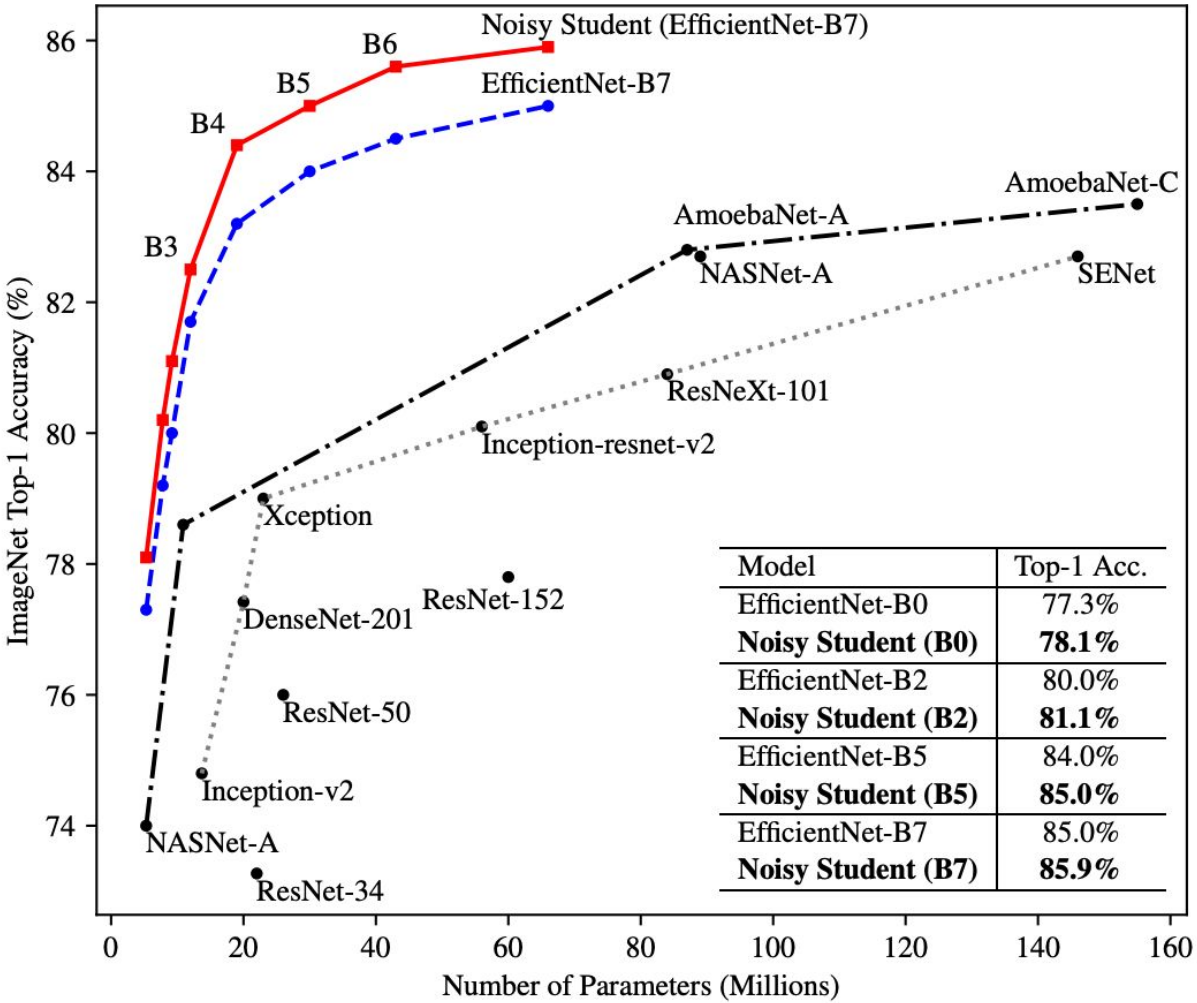
(b) ImageNet-C

(c) ImageNet-P

Method	Top-1 Acc.	Top-5 Acc.
ResNet-101 [30]	4.7%	-
ResNeXt-101 [30] (32x4d)	5.9%	-
ResNet-152 [30]	6.1%	-
ResNeXt-101 [30] (64x4d)	7.3%	-
DPN-98 [30]	9.4%	-
ResNeXt-101+SE [30] (32x4d)	14.2%	-
ResNeXt-101 WSL [51, 55]	61.0%	-
EfficientNet-L2	49.6%	78.6%
Noisy Student (L2)	**83.7%**	**95.2%**

Table 3: Robustness results on ImageNet-A.

- Лучшие результаты получили, когда повторили цикл **три** раза
- Шум помог больше, чем увеличение модели
- На “сложных” датасетах (ImageNet-A, ImageNet-C, ImageNet-P) также хорошие результаты



Model	Top-1 Acc.
EfficientNet-B0	77.3%
Noisy Student (B0)	**78.1%**
EfficientNet-B2	80.0%
Noisy Student (B2)	**81.1%**
EfficientNet-B5	84.0%
Noisy Student (B5)	**85.0%**
EfficientNet-B7	85.0%
Noisy Student (B7)	**85.9%**

Важность шума

Model / Unlabeled Set Size	1.3M	130M
EfficientNet-B5	83.3%	84.0%
Noisy Student (B5)	**83.9%**	**84.9%**
student w/o Aug	83.6%	84.6%
student w/o Aug, SD, Dropout	83.2%	84.3%
teacher w. Aug, SD, Dropout	83.7%	84.4%

Iteration	Model	Batch Size Ratio	Top-1 Acc.
1	EfficientNet-L2	1:14	87.6%
2	EfficientNet-L2	1:14	88.1%
3	EfficientNet-L2	1:28	88.4%

Выводы

- Большая производительная модель-учитель приводит к лучшим результатам
- Для повышения производительности нужно много неразмеченных данных
- Мягкие псевдо-метки работают лучше жестких
- Большая модель-студент важна для того, чтобы позволить студенту изучить более мощную модель
- Балансировка данных полезна для небольших моделей
- Совместное обучение по маркированным данным и немаркированным данным лучше предварительной подготовки с немаркированными данными, а затем тонкой настройки на размеченные данные
- Использование большого соотношения между немаркированными размерами батча и маркированный размерами батча позволяют моделя достигать более высокой точности
- Обучать студента с нуля лучше, чем инициализировать с помощью учителя