Self-training with Noisy Student

Трус Владлена, БПМИ172

Мотивация для данного метода

- Мы хотим дополнить датасет с лейблами датасетом без лейблов.
- Self-training обучает модель учителя с помощью labeled data, далее мы используем эту модель для unlabeled data.
- Так как предсказания модели учителя на unlabeled data не дают полностью хороших результатов, то нам нужен какой-то способ, чтобы работать с этой data.

Self-training

Algorithm 1 Classic Self-training

1: Train a base model $f_{\boldsymbol{\theta}}$ on $L = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{l}$
2: **repeat**
3: Apply $f_{\boldsymbol{\theta}}$ to the unlabeled instances U
4: Select a subset $S \subset \{(\boldsymbol{x}, f_{\boldsymbol{\theta}}(\boldsymbol{x}))|\boldsymbol{x} \in U\}$
5: Train a new model $f_{\boldsymbol{\theta}}$ on $S \cup L$
6: **until** convergence or maximum iterations are reached

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.

1: Learn teacher model θ_*^t which minimizes the cross entropy loss on labeled images

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

2: Use an unnoised teacher model to generate soft or hard pseudo labels for unlabeled images

$$\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \cdots, m$$

3: Learn an **equal-or-larger** student model θ_*^s which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

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

4: Iterative training: Use the student as a teacher and go back to step 2.

- В качестве шума используются: dropout, stochastic depth и augmentation
- Модель ученика должна быть больше модели учителя
- Модель ученика работает лучше, когда число картинок каждого класса без лэйблов одинаково

Какие данные

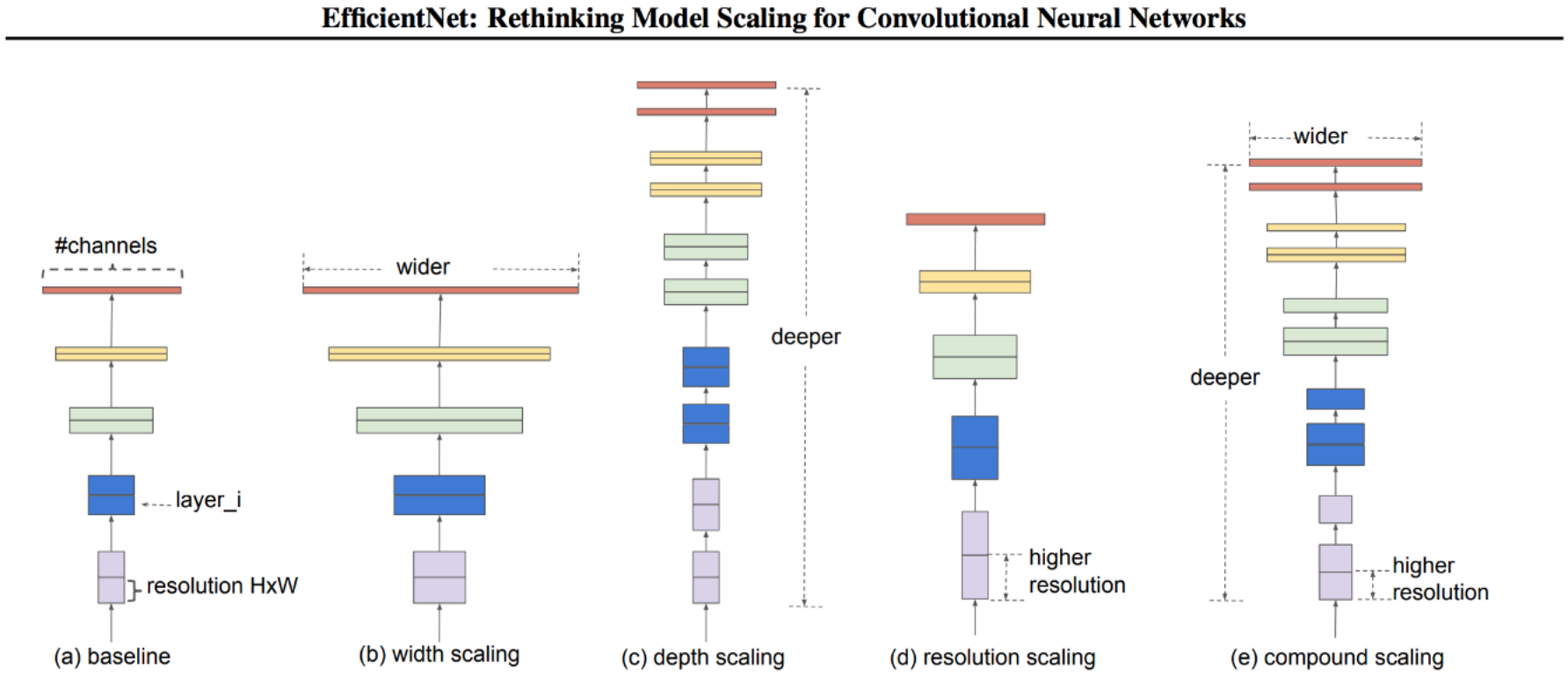
- Датасет с лейблами: ImageNet
- Датасет без лейблов: JFT, около 300 миллионов изображений, далее выбрано 130к каждого класса

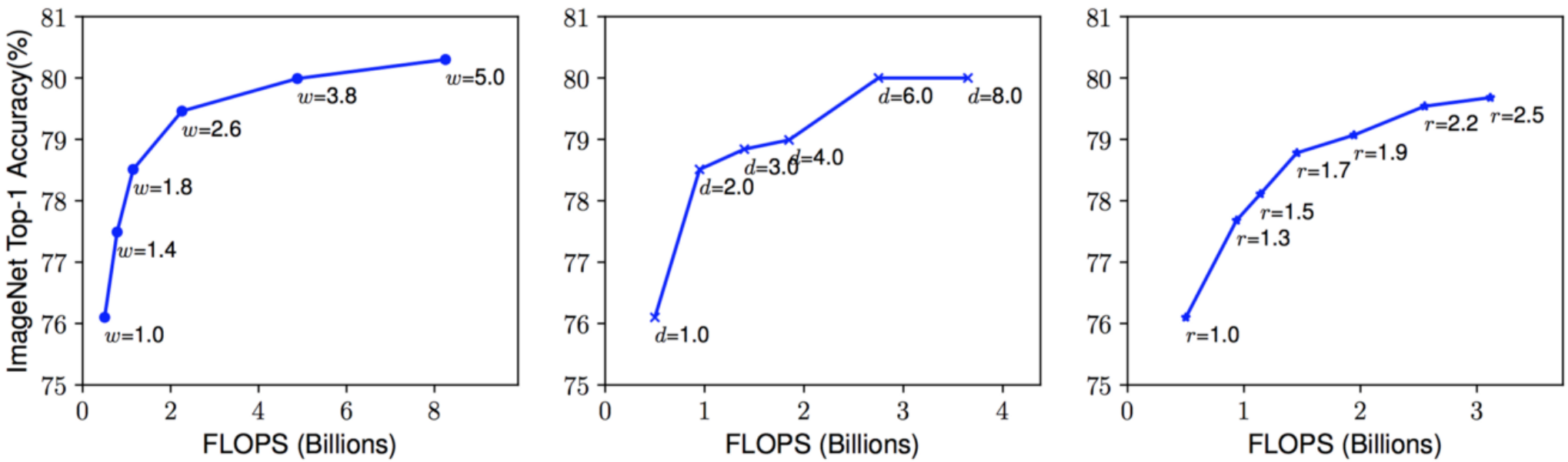
EfficientNets

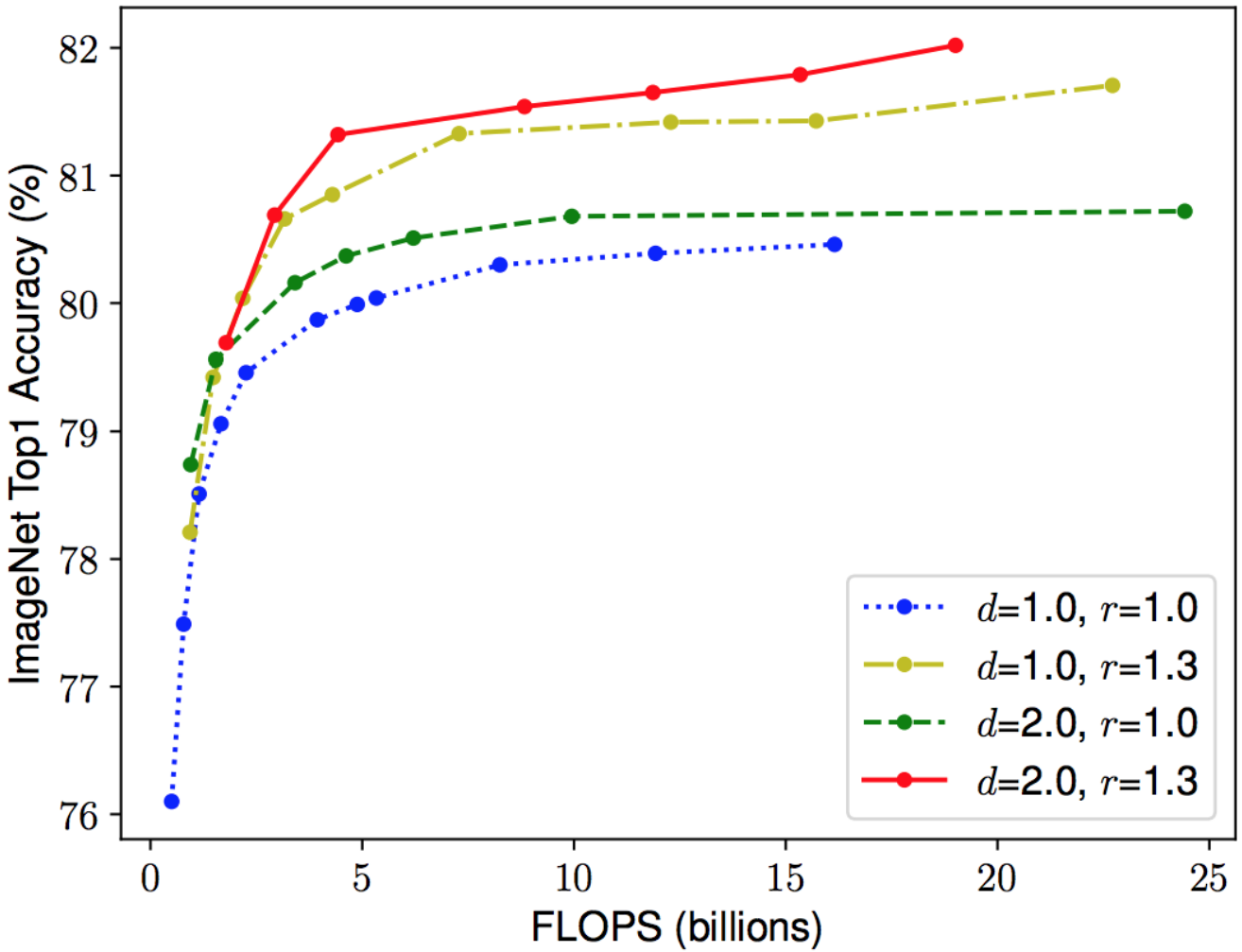
- Baseline создан с помощью NAS

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Масштабирование




ImageNet Top-1 Accuracy(%)
81
80
79
78
77
76
75
0
2
4
6
8
FLOPS (Billions)
w=1.0
w=1.4
w=1.8
w=2.6
w=3.8
w=5.0
1
3
d=1.0
d=2.0
d=3.0
d=4.0
d=6.0
d=8.0
r=1.0
r=1.3
r=1.5
r=1.7
r=1.9
r=2.2
r=2.5


82
81
80
79
78
77
76
ImageNet Top1 Accuracy (%)
0
5
10
15
20
25
FLOPS (billions)
d=1.0, r=1.0
d=1.0, r=1.3
d=2.0, r=1.0
d=2.0, r=1.3

Комбинированное масштабирование

$$\text{depth: } d = \alpha^{\phi}$$

$$\text{width: } w = \beta^{\phi}$$

$$\text{resolution: } r = \gamma^{\phi}$$

$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

- Фиксируем $\varphi = 1$, выполняем маленький grid search для α, β и γ

(для базовой B0 у нас вышли $\alpha = 1.2$, $\beta = 1.1$ и $\gamma = 1.15$)

- Потом фиксируем α, β и γ, которые получили на прошлом шаге, и экспериментируем с разными φ (в итоге получили EfficientNets B1-B7)

Архитектура

- Расширяют EfficientNet-B7 и получают три разные сети: EfficientNet-L0, L1 и L2
- EfficientNet-L0 шире и глубже и шире, чем EfficientNet-B7, но с меньшим разрешением
- EfficientNet-L1 расширяется от EfficientNet-L0 за счет увеличения ширины
- С помощью комбинированного масштабирования из EfficientNet-L1 получили EfficientNet-L2

Итеративное обучение

1. Точность EfficientNet-B7 сначала повышается за счет использования его как *модели учителя,* так и *модели ученика.*
2. Улучшенный EfficientNet-B7 теперь используется как учитель, а EfficientNet-L0 - как модель ученика.
3. EfficientNet-L0 теперь используется в качестве учителя, а EfficientNet-L1, который шире L0, используется в качестве модели ученика.
4. EfficientNet-L1 теперь используется в качестве учителя, а EfficientNet-L2, который является самой большой моделью, используется в качестве ученика.
5. Efficient-L2 теперь используется как учитель, а также как модель ученика.

Результаты Noisy Student

Самая большая модель, EfficientNet-L2, обучалась в течение 3,5 дней на Cloud TPU v3 Pod с 2048 ядрами

Method	# Params	Extra Data	Top-1 Acc.	Top-5 Acc.
ResNet-50 [23]	26M	-	76.0%	93.0%
ResNet-152 [23]	60M	-	77.8%	93.8%
DenseNet-264 [28]	34M	-	77.9%	93.9%
Inception-v3 [67]	24M	-	78.8%	94.4%
Xception [11]	23M	-	79.0%	94.5%
Inception-v4 [65]	48M	-	80.0%	95.0%
Inception-resnet-v2 [65]	56M	-	80.1%	95.1%
ResNeXt-101 [75]	84M	-	80.9%	95.6%
PolyNet [83]	92M	-	81.3%	95.8%
SENet [27]	146M	-	82.7%	96.2%
NASNet-A [86]	89M	-	82.7%	96.2%
AmoebaNet-A [54]	87M	-	82.8%	96.1%
PNASNet [39]	86M	-	82.9%	96.2%
AmoebaNet-C [13]	155M	-	83.5%	96.5%
GPipe [30]	557M	-	84.3%	97.0%
EfficientNet-B7 [69]	66M	-	85.0%	97.2%
EfficientNet-L2 [69]	480M	-	85.5%	97.5%
ResNet-50 Billion-scale [76]	26M	3.5B images labeled with tags	81.2%	96.0%
ResNeXt-101 Billion-scale [76]	193M		84.8%	-
ResNeXt-101 WSL [44]	829M		85.4%	97.6%
FixRes ResNeXt-101 WSL [71]	829M		86.4%	98.0%
Noisy Student (L2)	480M	300M unlabeled images	**87.4%**	**98.2%**

Почему важно добавить шум

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

Сравнение с Knowledge Distillation

- Различия: модель ученика такая же/больше модели учителя
- Делаем псевдометки
- Используем unlabeled data

Выводы

Self-training with Noisy Student является хорошим методом, который показал, что возможно использовать данные без лейблов для значительного повышения точности и надежности современных моделей, предсказывающих класс изображения.

Добавление шума к ученику помогает ему учиться за пределами знаний учителя.

В итоге, Noisy Student и EfficientNet могут достичь точности 88,4% что на 2,9% выше, чем без Noisy Student.

Этот результат на 2% лучше, чем предыдущий лучший метод.

Ссылки на источники

- https://arxiv.org/pdf/1911.04252.pdf
- https://arxiv.org/pdf/1905.11946.pdf
- https://arxiv.org/pdf/1503.02531.pdf
- https://medium.com/@nainaakash012/efficientnet-rethinking-model-scaling-for-convolutional-neural-networks-92941c5bfb95
- https://medium.com/@nainaakash012/self-training-with-noisy-student-f33640edbab2