

Designing Network Design Spaces

Петр Гринберг, Владимир Якшимамедов и Дмитрий
Поляков

Мотивация



Это я на дл пытаюсь подобрать правильную архитектуру сети



Это я после прочтения статьи

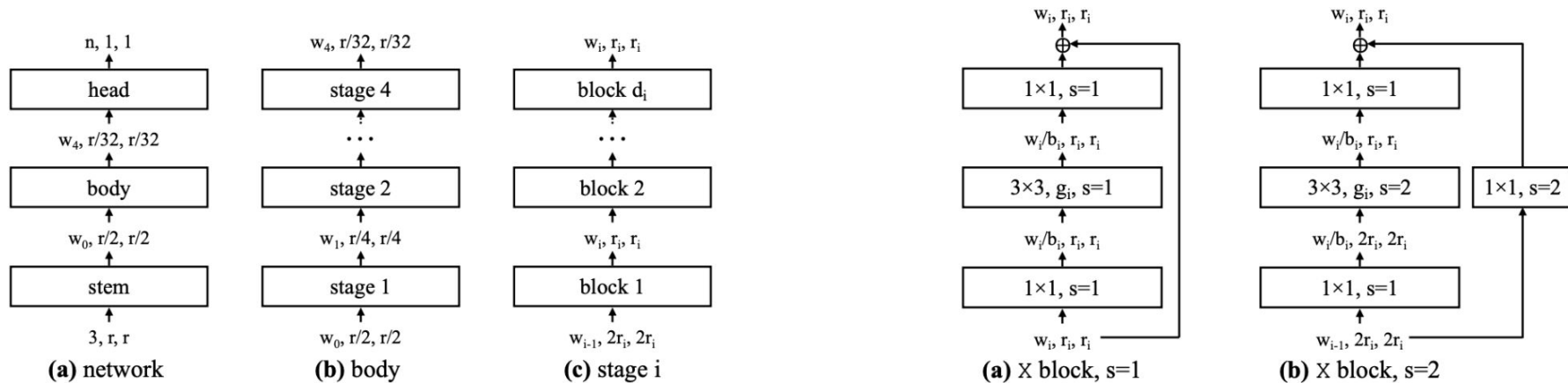
Цель

Выявить принципы дизайна архитектур нейросетей, которые позволяют получать хорошее качество, просты для понимания и имеют ограниченную сложность (не требуют больших вычислительных мощностей для обучения)

Как будем идти к цели

1. Возьмём очень широкое семейство архитектур
2. Будем его сужать, повышая качество архитектур и контролируя интерпретацию
3. Получим интерпретируемое узкое семейство с высоким качеством
4. Выявим и проверим те принципы, которые нам помогли построить узкое семейство

Очень широкое семейство архитектур (AnyNet)



Семейство определяется ограничениями на w_i, b_i, g_i, d_i

Как измерить качество семейства архитектур

Сэмплируем
 $n=500$
моделей

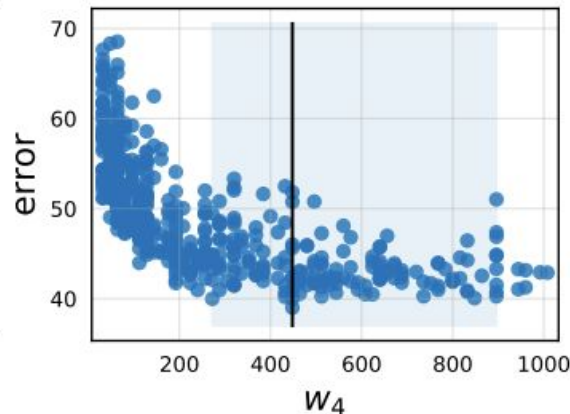
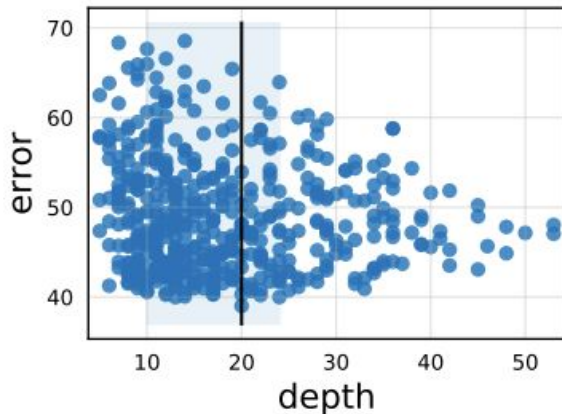
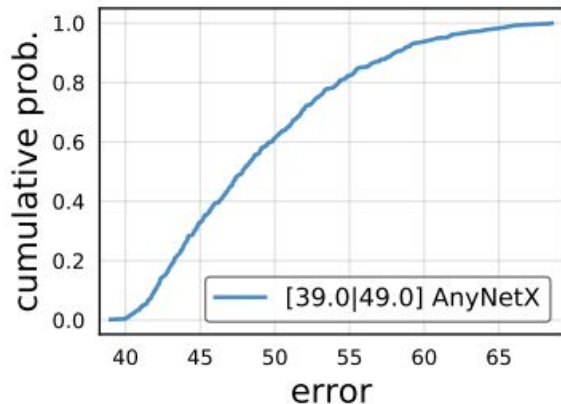


Обучаем их на
ImageNet,
**затрачивая
небольшое
количество
ресурсов**

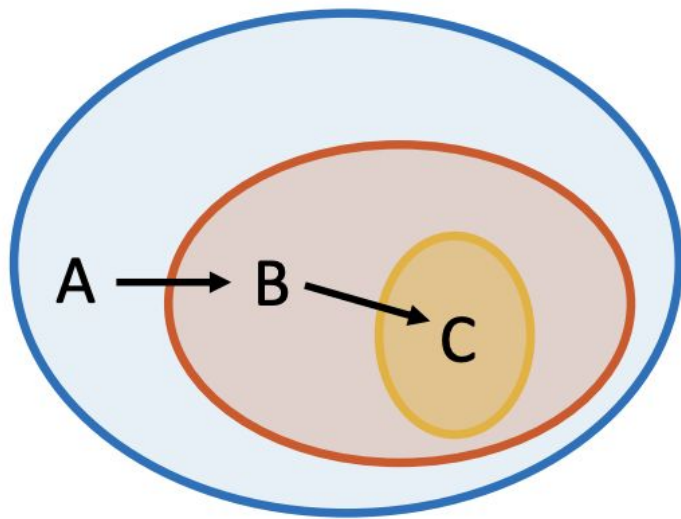


Строим эмпирическое
распределение
ошибки

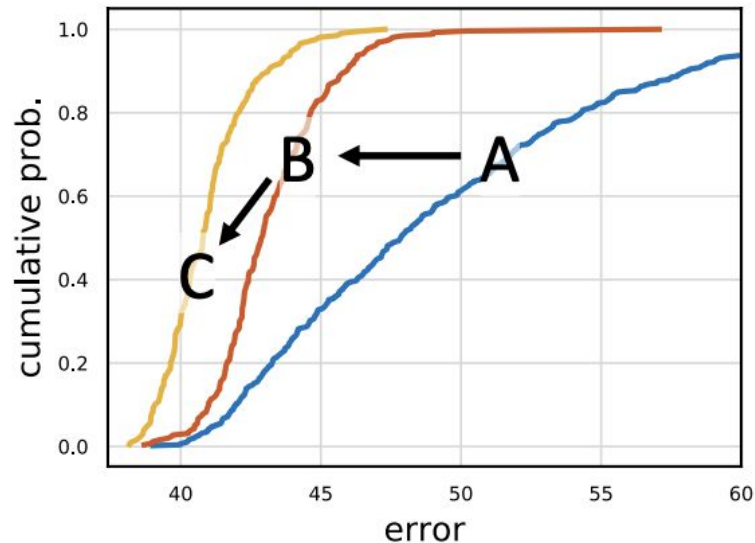
$$F(e) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[e_i < e]$$



Построение хорошего семейства архитектур

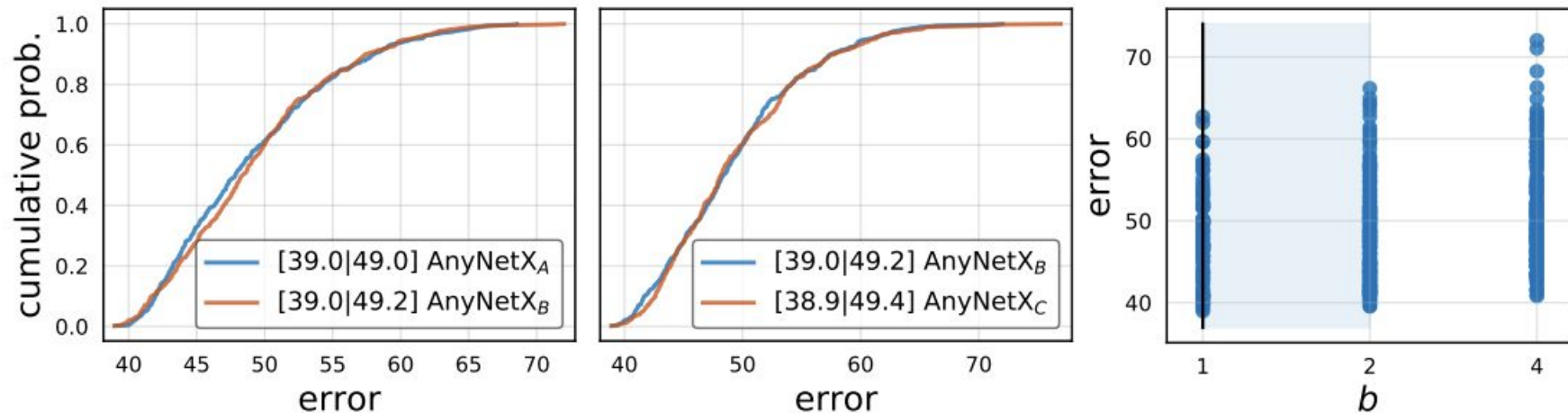


Поиск лучшего семейства архитектур



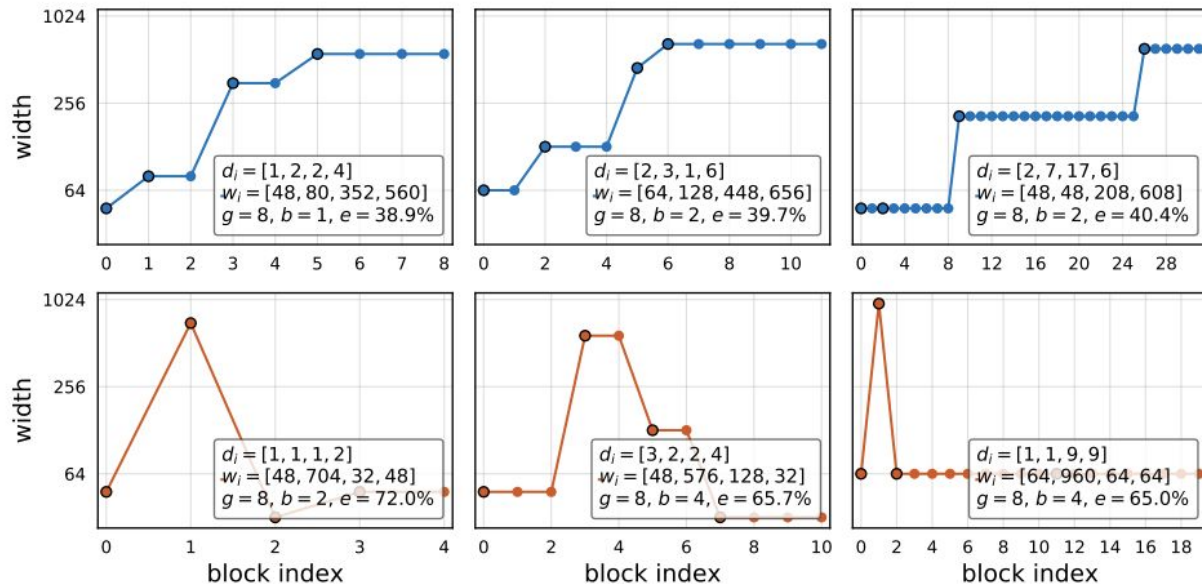
Как ведет себя распределение ошибки

Начинаем сужение семейства

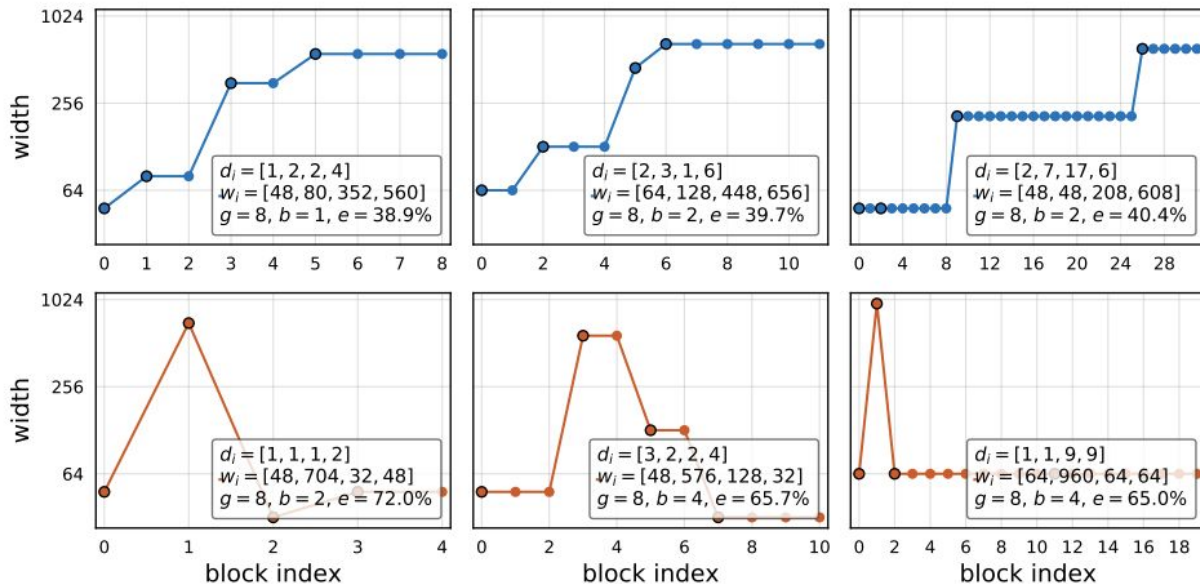


	restriction	dim.	combinations	total
AnyNetX _A	none	16	$(16 \cdot 128 \cdot 3 \cdot 6)^4$	$\sim 1.8 \cdot 10^{18}$
AnyNetX _B	$+ b_{i+1} = b_i$	13	$(16 \cdot 128 \cdot 6)^4 \cdot 3$	$\sim 6.8 \cdot 10^{16}$
AnyNetX _C	$+ g_{i+1} = g_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6$	$\sim 3.2 \cdot 10^{14}$

Смотрим на характеристики моделей



Смотрим на характеристики моделей

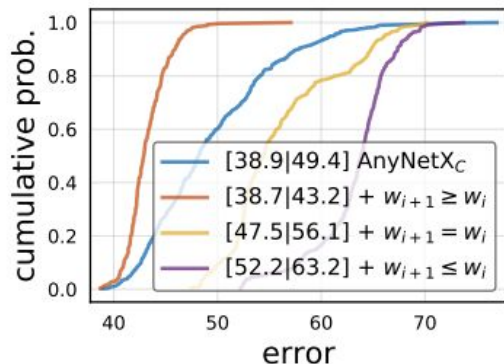


$$w_{i+1} \geq w_i$$

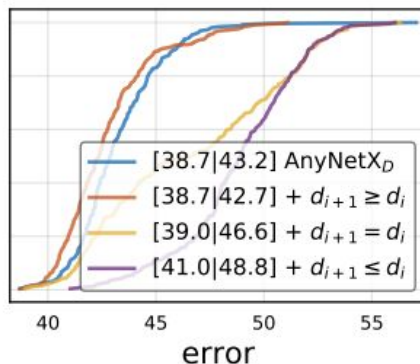


Продолжаем сужение

AnyNetX_D

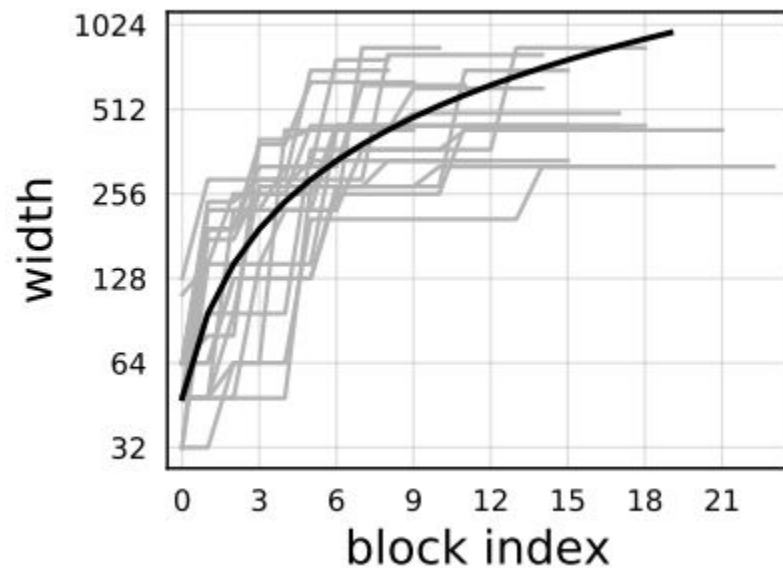


AnyNetX_E

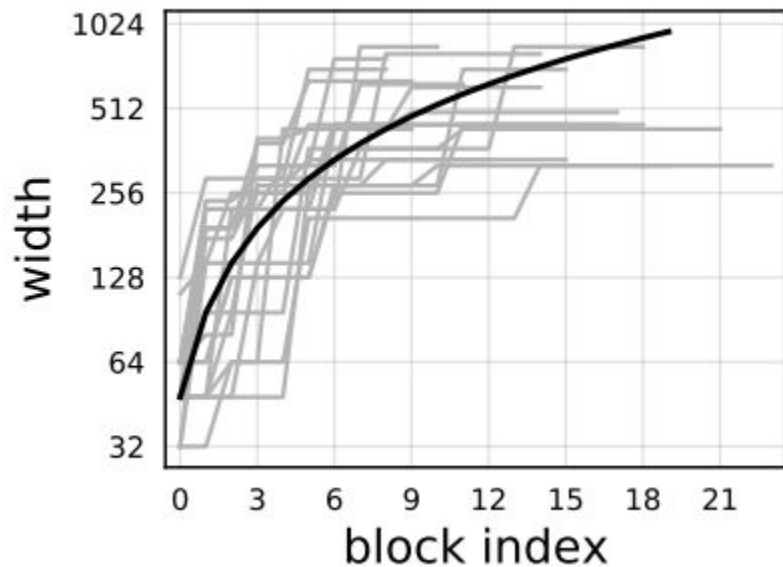


	restriction	dim.	combinations	total
AnyNetX _A	none	16	$(16 \cdot 128 \cdot 3 \cdot 6)^4$	$\sim 1.8 \cdot 10^{18}$
AnyNetX _B	+ $b_{i+1} = b_i$	13	$(16 \cdot 128 \cdot 6)^4 \cdot 3$	$\sim 6.8 \cdot 10^{16}$
AnyNetX _C	+ $g_{i+1} = g_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6$	$\sim 3.2 \cdot 10^{14}$
AnyNetX _D	+ $w_{i+1} \geq w_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)$	$\sim 1.3 \cdot 10^{13}$
AnyNetX _E	+ $d_{i+1} \geq d_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)^2$	$\sim 5.5 \cdot 10^{11}$

Смотрим на лучшие модели



Смотрим на лучшие модели



Линейная
зависимость



Конструируем RegNet

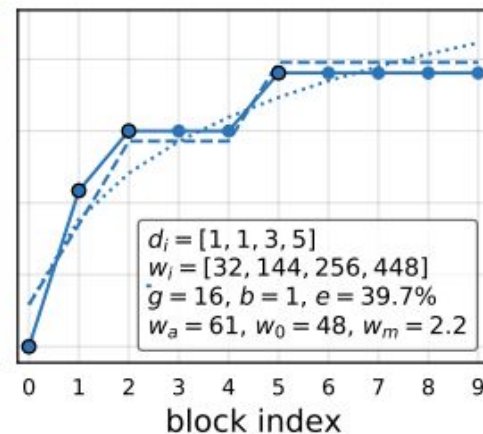
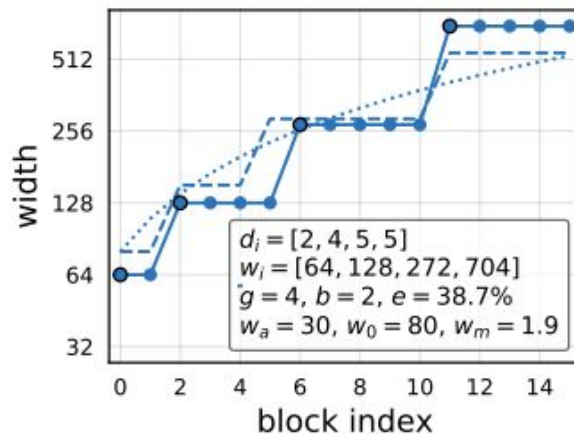
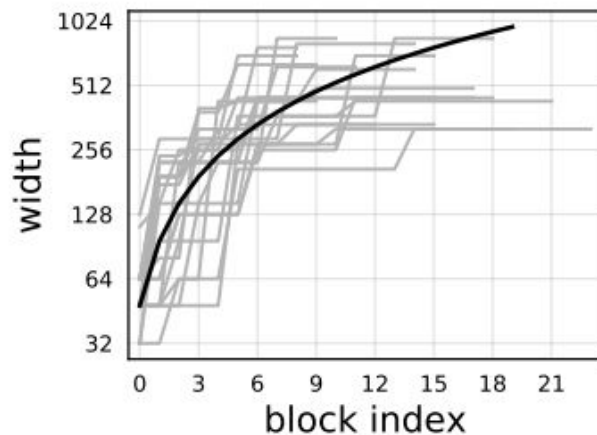
$$u_j = w_0 + w_a \cdot j \quad \text{for} \quad 0 \leq j < d$$

$$u_j = w_0 \cdot w_m^{s_j}$$

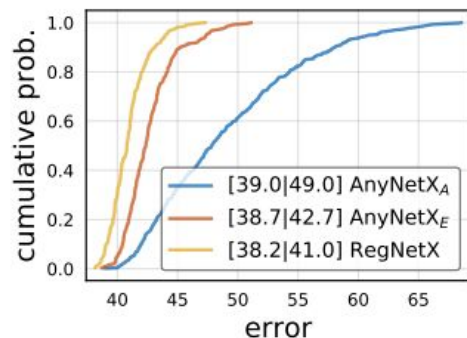
$$w_i = w_0 \cdot w_m^i \longrightarrow w_j = w_0 \cdot w_m^{\lfloor s_j \rfloor}$$

$$d_i = \sum_j \mathbf{1}[\lfloor s_j \rfloor = i]$$

Осознаем новую параметризацию

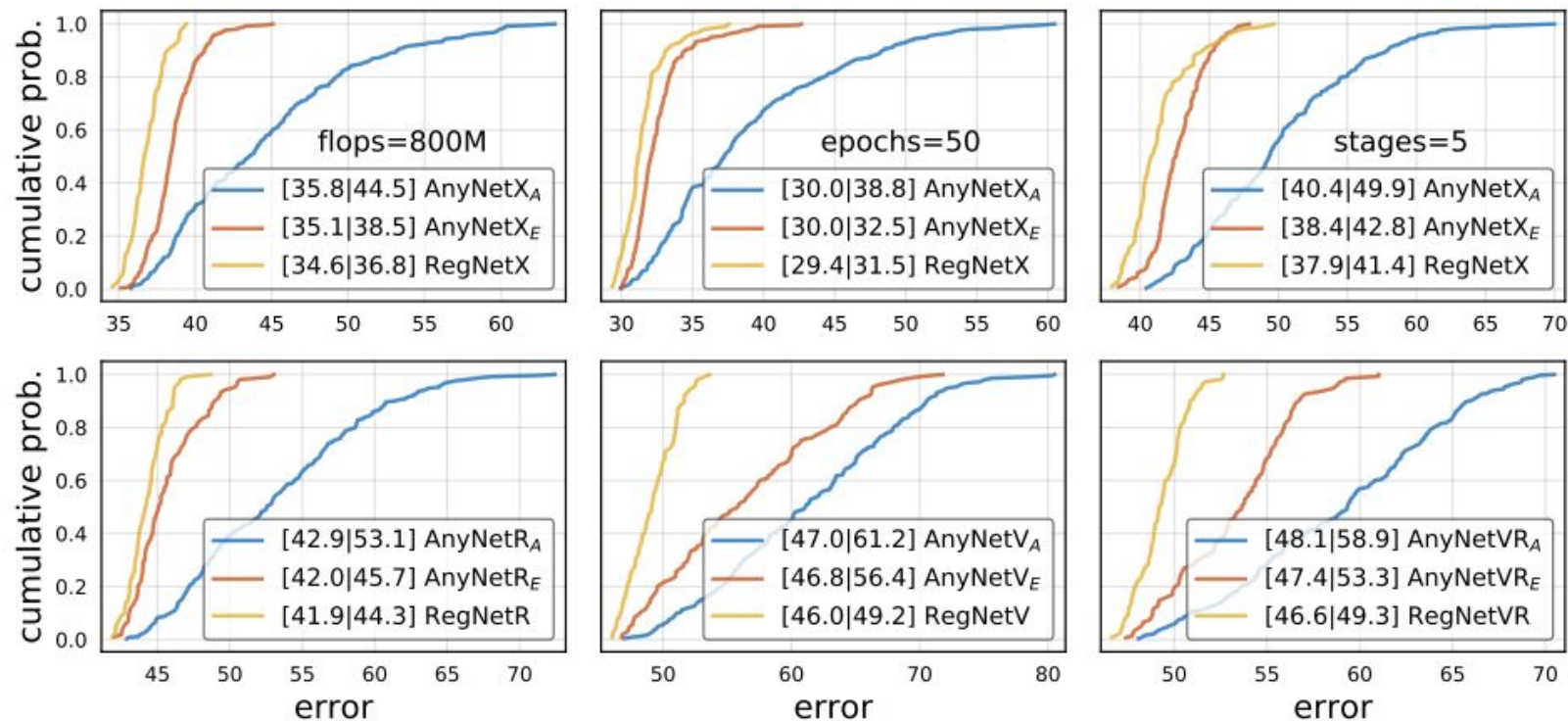


О, получилось

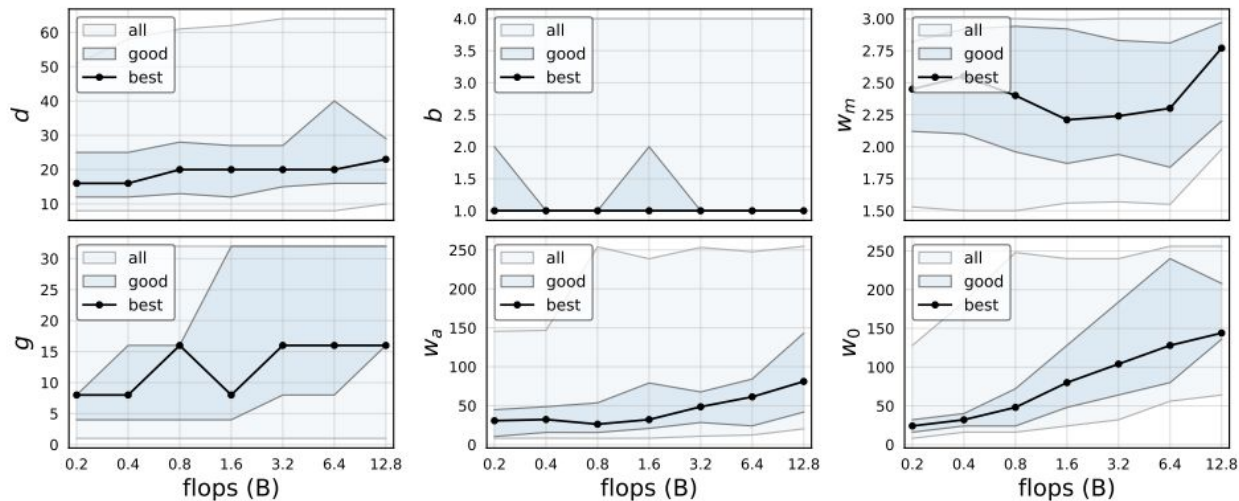


	restriction	dim.	combinations	total
AnyNetX _A	none	16	$(16 \cdot 128 \cdot 3 \cdot 6)^4$	$\sim 1.8 \cdot 10^{18}$
AnyNetX _B	$+ b_{i+1} = b_i$	13	$(16 \cdot 128 \cdot 6)^4 \cdot 3$	$\sim 6.8 \cdot 10^{16}$
AnyNetX _C	$+ g_{i+1} = g_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6$	$\sim 3.2 \cdot 10^{14}$
AnyNetX _D	$+ w_{i+1} \geq w_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)$	$\sim 1.3 \cdot 10^{13}$
AnyNetX _E	$+ d_{i+1} \geq d_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)^2$	$\sim 5.5 \cdot 10^{11}$
RegNet	quantized linear	6	$\sim 64^4 \cdot 6 \cdot 3$	$\sim 3.0 \cdot 10^8$

Проверяем обобщающую способность RegNet



Ищем подходящие параметры RegNet



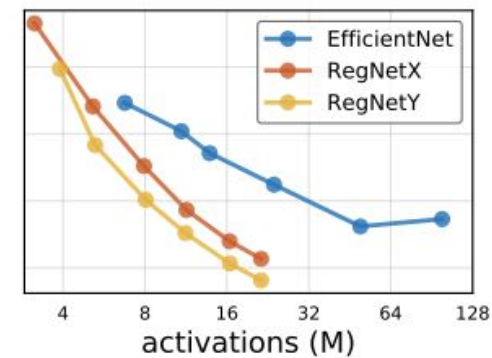
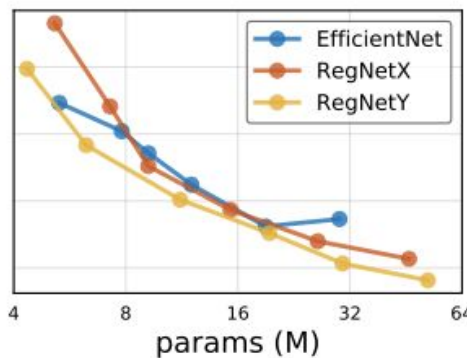
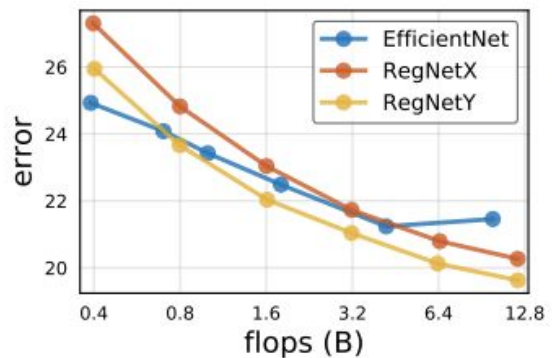
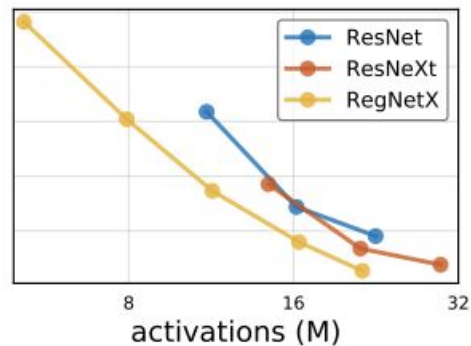
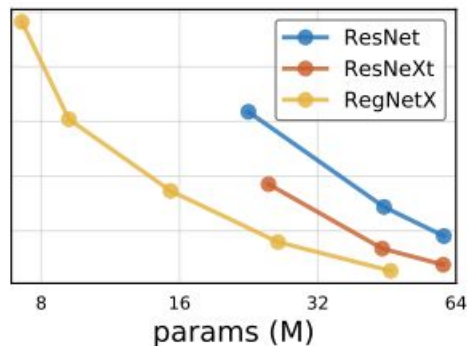
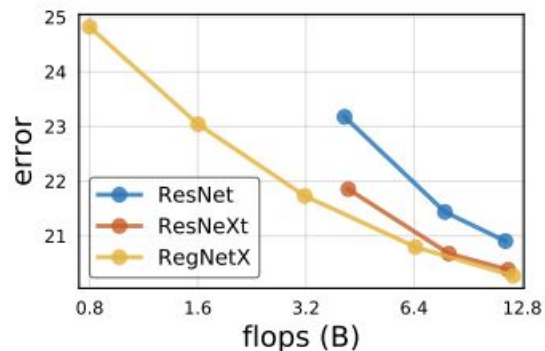
Инсайты

- $d \sim 20$
- $b \sim 1$
- $w_m \sim 2,5$
- g, w_a, w_0 возрастают

Сравнение с другими архитектурами: Mobile

	flops (B)	params (M)	top-1 error
MOBILENET [9]	0.57	4.2	29.4
MOBILENET-V2 [25]	0.59	6.9	25.3
SHUFFLENET [33]	0.52	-	26.3
SHUFFLENET-V2 [19]	0.59	-	25.1
NASNET-A [35]	0.56	5.3	26.0
AMOEBANET-C [23]	0.57	6.4	24.3
PNASNET-5 [17]	0.59	5.1	25.8
DARTS [18]	0.57	4.7	26.7
REGNETX-600MF	0.60	6.2	25.9 \pm 0.03
REGNETY-600MF	0.60	6.1	24.5 \pm 0.07

Сравнение с другими архитектурами: Full Regime



Выводы

Пространство RegNet (в сравнении с AnyNet)

1. Упрощенно как с точки зрения размеров, так и с точки зрения сетевых конфигураций
2. Содержит более высокую концентрацию хороших моделей
3. Лучше поддается анализу и интерпретации.

Авторы статьи представили новую парадигму проектирования архитектур нейросетей. Их результаты показывают, что проектирование семейств архитектур является многообещающим направлением для будущих исследований.

Авторы статьи

- ~~Facebook~~ Meta AI Research

Ilija Radosavovic, Justin Johnson, Saining Xie, Wan-Yen Lo - CV (Object Detection, Video Recognition)

Piotr Dollar - CV + NAS (Neural Architecture Search)

- Ilija Radosavovic, Piotr Dollar - On Network Design Spaces for Visual Recognition

Предшествующие и конкурирующие работы

AutoML: A Survey of the State-of-the-Art (2021) - не совсем конкурент, но...

- Разбирает полностью автоматизированный пайплайн обучения нейронки, включая обработку данных (пока с пропусками)
- Одной из частей является разбор различных алгоритмов поиска NAS (RL, Evolutionary, Cell-Based and so on)
- Фокусируется только на Auto-ML

Немного об алгоритмах

Evolutionary: Играем в жизнь: пытаемся эволюционировать лучшими сетками

Selection - оставляем часть структуры, чтобы сохранить общую хорошую генетику.

Cross-over - скрещиваем двух родителей в ребенка, который забирает по половине информации от родителей в свободных узлах.

Мутация - Случается рандом. Рандомные веса, новый слой, изменение слоя и т.д. (На самом деле, он немного задается исследователями)

Немного об алгоритмах

Cell-Based - Учим super-network, которая генерирует архитектуры блоков (cells) из которых мы строим network для задачи.

Weight-Sharing Cell-Based - super-network вместе с архитектурой блоков, учит pretrained веса для этих блоков.

Проблемы:

Как вообще нормально учить super-network?

Сходство блоков.

Очень нестабильное обучение.

Предшествующие и конкурирующие работы

On Network Design Spaces for Visual Recognition (2019) - вводит EDF и предлагает метод сравнения распределений

Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap (2020) - предлагает модификации к Weight-Sharing NAS для стабильности обучения супер-сетки.

Последующие работы и возможности

- 569 цитирований, но нет прямых последователей.

В этих статьях используется EDF, сравниваются сетки, полученные разными способами NAS или поднимается вопрос: насколько вообще достаточно оптимизировать CNN только на ImageNet.

- В целом идею данной статьи можно использовать не только в CV, но и в других областях ML.

Список литературы

Designing Network Design Spaces - <https://arxiv.org/pdf/2003.13678.pdf>

On Network Design Spaces for Visual Recognition - <https://arxiv.org/pdf/1905.13214.pdf>

AutoML: A Survey of the State-of-the-Art - <https://arxiv.org/pdf/1908.00709.pdf>

Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap
- <https://arxiv.org/pdf/2008.01475.pdf>