MeCo

Tìm kiếm kiến trúc mạng zero-shot với một lần lan truyền tiến thông giá trị riêng nhỏ nhất của ma trận tương quan

Neural Architecture Search

Ngày 15 tháng 4 năm 2024

Mục lục

Tổng quan nghiên cứu

- Zero-shot NAS (NAS không cần huấn luyện) có thể đánh giá mạng mà không cần huấn luyện thông qua một số chỉ số cụ thể gọi là zero-cost proxy (đại diện không tốn kém).
- Mặc dù hiệu quả, các zero-cost proxy hiện có hoặc là yêu cầu ít nhất một lần lan truyền ngược hoặc phụ thuộc nhiều vào dữ liệu và nhãn.
- Để giảm nhẹ các vấn đề trên, trong bài báo này, tác giả nghiên cứu cách *ma trận tương quan Pearson của các feature map (bản đồ đặc trưng)* ảnh hưởng đến tốc độ hội tụ và khả năng tổng quát hóa của một mạng over-parameterized (mạng quá tham số).
- Dựa trên phân tích lý thuyết, đề xuất một zero-cost proxy mới gọi là **MeCo**, chỉ yêu cầu một dữ liệu ngẫu nhiên cho một lần lan truyền tiến.

Các đóng góp chính

- Phân tích lý thuyết cách ma trận tương quan Pearson của feature map ảnh hưởng đến tốc độ hội tụ trong huấn luyện và khả năng tổng quát hóa của các mạng, dựa trên đặc điểm của lớp tích chập đa kênh (multi-channel).
- Phát triển một zero-cost proxy cho zero-shot NAS, gọi là MeCo. Đồng thời đề xuất MeCo_{opt} như một phương pháp tối ưu hóa.
- Triển khai MeCo một cách nghiêm ngặt và thiết kế rộng rãi các thí nghiệm để đánh giá hiệu suất trên nhiều benchmark phổ biến với các tập dữ liệu đa dạng và mẫu dữ liệu ngẫu nhiên.
- $\Rightarrow$ Kết quả thí nghiệm cho thấy MeCo vượt trội hơn (outperform) các phương pháp hiện có, và NAS dựa trên MeCo có thể chọn mạng với độ chính xác cao nhất.

Zero-shot NAS và Zero-cost proxy

- Zero-shot NAS sử dụng các chỉ số đặc biệt (gọi là zero-cost proxy) để đánh giá mạng.
- Zero-shot NAS loại bỏ quy trình huấn luyện, do đó cải thiện đáng kể hiệu suất.
- Nhiều nghiên cứu đã được đưa ra để nâng cao chất lượng của các chỉ số zero-cost proxy, và hầu hết trong số đó dựa trên gradient.
  - Sử dụng trùng lặp kích hoạt (activation overlap) giữa các điểm dữ liệu[1]; loạt chỉ số dựa trên cắt tỉa tham số (parameter pruning)[2] gồm: `snap`, `grasp`, `synflow`, `fisher`; ZiCo[3] là phương pháp đầu tiên hoạt động tốt hơn số lượng tham số. ← cần ít nhất một lần lan truyền ngược và phụ thuộc vào nhãn dữ liệu để tính toán các chỉ số đánh giá.
  - Một hướng nghiên cứu khác dựa trên lý thuyết của học sâu như NTK (Neural Tangent Kernel), độc lập với nhãn nhưng vẫn cần lan truyền ngược[4].
  - Cuối cùng, có một vài công trình không yêu cầu lan truyền ngược hoặc nhãn dữ liệu, ví dụ Zen-score[5] thiết kế một chỉ số đại diện không tốn kém hiệu quả với các đầu vào ngẫu nhiên Gaussian. Tuy nhiên Zen-score yêu cầu nhiều lần lan truyền tiến.

[1] Mellor et al, Neural architecture search without training, ICML

[2] Abdelfattah et al, Zero-cost proxies for lightweight NAS, ICLR

[3] Li et al, Zico: Zero-shot NAS via inverse coefficient of variation on gradients, ICLR

[4] Chen et al, Neural architecture search on imagenet in four GPU hours: A theoretically inspired perspective, ICLR

[5] Lin et al, Zen-nas: A zero-shot nas for high-performance image recognition, ICCV

Mạng quá tham số (over-parameterized networks)

- Các mạng quá tham số đã nhận được nhiều sự chú ý do hiệu quả nổi bật và tính dễ tối ưu hóa của chúng.
- Jacot et al.[6] đã chứng minh rằng động lực huấn luyện của một mạng có chiều rộng (kích thước lớp ẩn) vô hạn tuân theo kernel được gọi là NTK.
- Đối với các mạng có chiều rộng hữu hạn, động lực huấn luyện có thể được minh họa bởi một ma trận gram. Du et al.[7] đã chứng minh rằng mất mát của một mạng quá tham số có thể hội tụ về một cực tiểu toàn cục, và tốc độ hội tụ huấn luyện có thể được đặc trưng bởi giá trị riêng nhỏ nhất của ma trận gram.

[6] Neural tangent kernel: Convergence and generalization in neural networks

[7] Gradient descent provably optimizes overparameterized neural networks, ICLR

Sơ bộ

- Kí hiệu $[n] = \{1, 2, \ldots, n\}$.
- $\mathbf{X} = \left\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^{d\times 1}, y_i \in \mathbb{R}, i \in [n]\right\}$ là tập dữ liệu huấn luyện.

Với một đầu vào $\mathbf{x} \in \mathbb{R}^{d\times 1}$, vector trọng số $\mathbf{w} \in \mathbb{R}^{d\times 1}$ trong ma trận trọng số $\mathbf{W} \in \mathbb{R}^{d\times m}$, và trọng số đầu ra $\mathbf{a} \in \mathbb{R}^{m\times 1}$; ta biểu thị $f(\mathbf{W}, \mathbf{a}, \mathbf{x})$ là mạng với một lớp ẩn duy nhất như sau:

$$f(\mathbf{W}, \mathbf{a}, \mathbf{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(\mathbf{w}_r^\top \mathbf{x}\right)$$

Ở đây σ là hàm kích hoạt. Trong nghiên cứu này, chủ yếu xét hàm ReLU do hiệu quả của nó. Với tập dữ liệu huấn luyện $\mathbf{X}$, mục tiêu tối ưu hóa là giảm thiểu lỗi:

$$\mathcal{L}(\mathbf{W}, \mathbf{a}) = \sum_{i=1}^{n} \frac{1}{2} \left(f(\mathbf{W}, \mathbf{a}, \mathbf{x}) - y_i\right)^2$$

Định nghĩa: Ma trận Gram

Đối với mạng đơn lớp ẩn, ma trận gram $\mathbf{H}(t) \in \mathbb{R}^{n \times n}$ được tạo ra bởi hàm ReLU trên tập huấn luyện $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ với các phần tử $[\mathbf{H}(t)]_{ij}$ xác định như sau:

$$[\mathbf{H}(t)]_{ij} = \frac{1}{m} \sum_{r=1}^{m} \mathbf{x}_i^\top \mathbf{x}_j \mathbb{I}\left\{\mathbf{w}_r(t)^\top \mathbf{x}_i \geq 0, \mathbf{w}_r(t)^\top \mathbf{x}_j \geq 0\right\}$$

trong đó $\mathbf{w}_r(t)$ là một vector phụ thuộc vào t. Tiếp xúc xây dựng $\mathbf{H}^\infty$ với các phần tử $[\mathbf{H}^\infty]_{ij}$ như sau:

$$[\mathbf{H}^\infty]_{ij} = \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})}\left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I}\left\{\mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0\right\}\right]$$

và định nghĩa giá trị riêng nhỏ nhất $\lambda_0 = \lambda_{\min}(\mathbf{H}^\infty)$.

Nhận xét. Ma trận Gram $\mathbf{H}(t)$ phản ánh động lực dự đoán (prediction dynamic) tại lần lặp thứ t của quá trình huấn luyện mạng. Người ta chứng minh được rằng, $\forall t \geq 0$, $\|\mathbf{H}(t) - \mathbf{H}^{\infty}\|_2 \to 0$ khi $m \to +\infty$. Hơn nữa ma trận gram $\mathbf{H}^{\infty}$ có tính chất: nếu $\mathbf{x}_i \nparallel \mathbf{x}_j$ thì $\lambda_0 > 0$.

Định nghĩa: Ma trận tương quan Pearson

Ma trận tương quan Pearson $P(\mathbf{X})$ có các phần tử $[P(\mathbf{X})]_{ij}$ định nghĩa bởi:

$$[P(\mathbf{X})]_{ij} = \rho(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbb{E}\left[(\mathbf{x}_i - \mu_{\mathbf{x}_i})(\mathbf{x}_j - \mu_{\mathbf{x}_j})\right]}{\sigma_{\mathbf{x}_i}\sigma_{\mathbf{x}_j}}$$

với $\mu_{\mathbf{x}}$ và $\sigma_{\mathbf{x}}$ là trung bình, độ lệch chuẩn tương ứng của $\mathbf{x}$.

Nhận xét. Ma trận tương quan Pearson $P(\mathbf{X})$ có liên quan mật thiết đến $\mathbf{H}(t)$. Ví dụ, giả sử $\mathbf{x}_1, \mathbf{x}_2 \sim \mathcal{N}(0, \mathbf{I})$ thỏa $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d\times 1}$ và $\mathbf{w}_r^\top(t)\mathbf{x}_i \geq 0$ với $r \in [m]$ thì $[\mathbf{H}(t)]_{12} = [P(\mathbf{X})]_{12}$.

Thiết kế MeCo

Đầu tiên ước lượng mạng CNN thành mạng quá tham số và sau đó tạo ra chỉ số đánh giá.

Phép toán tích chập. Lớp tích chập là thành phần cơ bản của CNN. Giả sử chúng ta có đầu vào $\mathbf{x}_{\text{in}} \in \mathbb{R}^{c_{\text{in}} \times w \times h}$ và bộ lọc (filter) $\mathbf{w}_{\text{in}} \in \mathbb{R}^{c_{\text{in}} \times k \times k}$. Kích cỡ bước nhảy là 1 và giữ kích thước của các bản đồ đặc trưng thông qua phép đệm:

$$[\mathbf{x}_{\text{out}}]_{i,j} = \sum_{c=1}^{c_{\text{in}}} \sum_{a=-p}^{p} \sum_{b=-p}^{p} [\mathbf{w}^c]_{a+p+1,b+p+1} \times [\mathbf{x}_{\text{in}}^c]_{a+i,b+j}$$

trong đó $\mathbf{x}_{\text{in}}^c$ biểu diễn kênh thứ c của đầu vào, $p = \dfrac{k-1}{2}, i \in [w], j \in [h]$.

Trong nghiên cứu này, ta coi như $\mathbf{x}_{\text{out}}$ và $\mathbf{x}_{\text{in}}^c$ có thể được làm phẳng thành các vector 1 chiều $\tilde{\mathbf{x}}_{\text{out}}, [\tilde{\mathbf{x}}^c]_{\text{in}} \in \mathbb{R}^{d\times 1}$ với $c \in [c_{\text{in}}], d = w \times h$. Cụ thể hơn, thông quá phép biến đổi sau:

$$\tilde{\mathbf{x}}_{\text{out}} = \sum_{c=1}^{c_{\text{in}}} \mathbf{A}_c \sigma\left((\mathbf{B}_c \odot \mathbf{W})^\top \tilde{\mathbf{x}}_{\text{in}}^c\right)$$

thỏa mãn:

- $\mathbf{A}_c \in \mathbb{R}^{d\times d_h}, [\mathbf{A}_c]_{ij} = \mathbb{I}\{j = (c-1)d + i\}$.
- $\mathbf{B}_c \in \mathbb{R}^{d_h \times d}, [\mathbf{B}_c]_{ij} = \mathbb{I}\{(c-1)d < i \leq cd\}$.
- $\mathbf{B}_c \odot \mathbf{W}$ thỏa mãn chia sẻ trọng số.

trong đó, $d_h = c_{\text{in}} \times d, \mathbf{W} \in \mathbb{R}^{d\times d_h}$ là ma trận trọng số. Như vậy, một lớp tích chập trong CNN tương đương với một lớp kết nối đầy đủ có ràng buộc. Để đơn giản, ta giảm nhẹ ràng buộc thứ hai thành $\mathbf{B}_c = \mathbf{1}_{d_h \times d}$ và bỏ qua ràng buộc cuối cùng. Kết quả là, thu được một mạng kết nối đầy đủ đa mẫu trong đó mỗi kênh được làm phẳng $\tilde{\mathbf{x}}_{\text{in}}^c$ được xem như là một mẫu dữ liệu độc lập, và tổng số mẫu là c_{in}.

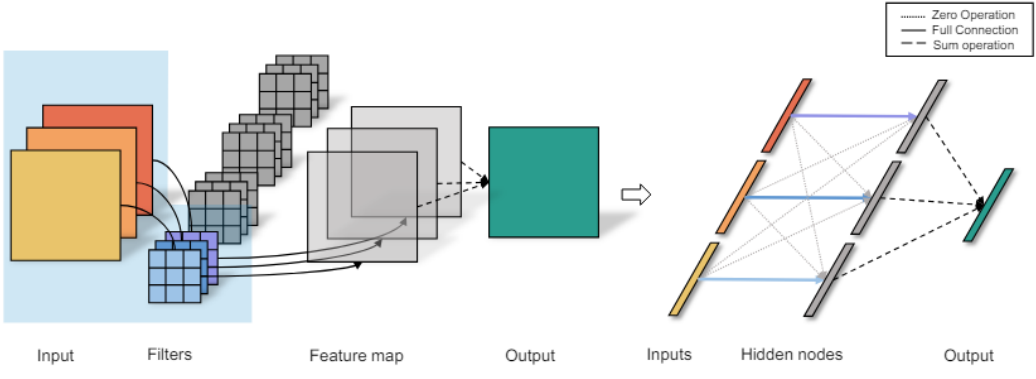

Figure 1: Conversion between the multi-channel convolution and multi-sample fully-connected operation. The input size is $3 \times w \times h$ and each filter size is $3 \times 3 \times 3$. Each input channel can be flattened with size $d \times 1$, $d = w \times h$. We collect the three flattened samples to obtain the output with constrained fully-connected operations (dot lines for zero operations and solid lines for full connection). The final output is computed by a sum operation (dashed lines) from the hidden nodes.

Mạng quá tham số. Các NN quá tham số cạnh tranh trong việc trích xuất đặc trưng phân cấp do số lượng tham số lớn. Một trong những kiến trúc điển hình là mạng với các lớp ẩn rộng, đã được chứng minh là dễ huấn luyện.

Trong phần trước, chúng ta chuyển đổi một lớp tích chập đa kênh thành một lớp kết nối đầy đủ đa mẫu với ràng buộc. Nghiên cứu lập luận rằng nếu số lượng nút ẩn trong lớp kết nối đầy đủ đã biến đổi đủ lớn, thì nó có thể được xem là một lớp NN quá tham số. Do đó, các đặc điểm của NN quá tham số có thể được chuyển giao sang CNN.

Định lý 1

Nếu ma trận gram $\mathbf{H}^{\infty} \succ 0, \|\mathbf{x}_i\|_2 = 1, |y_i| < C$ với hằng số C nào đó, $i \in [n]$, số nút ẩn $m = \Omega\left(\frac{n^6}{\lambda_0^4 \delta^3}\right)$ và khởi tạo $\mathbf{w}_r \sim \mathcal{N}(0, \mathbf{I}), a_r \sim \mathcal{U}[-1, 1]$ với $r \in [m]$ thì với xác suất ít nhất $1 - \delta$ trên khởi tạo, bất đẳng thức sau xảy ra:

$$\|f(\mathbf{W}(t), \mathbf{a}, \mathbf{X}) - \mathbf{y}\|_2^2 \leq e^{-\lambda_0 t} \|f(\mathbf{W}(0), \mathbf{a}, \mathbf{X}) - \mathbf{y}\|_2^2$$

Nhận xét. Bất đẳng thức này cho thấy λ_0 ảnh hưởng tích cực đến tốc độ hội tụ huấn luyện của mạng. Hơn nữa, tốc độ hội tụ của mạng độc lập với nhãn $\mathbf{y}$, điều này làm cho việc đo lường hiệu suất mạng chỉ với một lượt truyền tiến là có thể.

Định lý 2

Giả sử f là một mạng NN với một lớp ẩn và hàm kích hoạt ReLU. Giả sử $\mathbf{X} \in \mathbb{R}^{d\times n}, \mathbf{w}(0) \sim \mathcal{N}(0, \mathbb{I}), P(\mathbb{X}) \succ 0, p_0 = \lambda_{\min}(P(\mathbf{X}))$ và số lượng nút ẩn $m = \Omega\left(\frac{n^6 d^2}{\lambda_0^4 \delta^3}\right)$ thì bất đẳng thức sau xảy ra với xác suất ít nhất $1 - \delta$ lần trên khởi tạo:

$$\|f(\mathbf{W}(t), \mathbf{a}, \mathbf{X}) - \mathbf{y}\|_2^2 \leq e^{-cp_0 t}\|f(\mathbf{W}(0), \mathbf{a}, \mathbf{X}) - \mathbf{y}\|_2^2$$

với c là hằng số phụ thuộc vào m và d. Lợi thế của $P(\mathbf{X})$ so với $\mathbf{H}(t)$ là việc tính toán của nó chỉ phụ thuộc vào dữ liệu. Do đó, việc ước lượng tốc độ hội tụ của mạng chỉ qua các bản đồ đặc trưng là khả thi.

Định lý 3

Cho một mạng NN quá tham số với lỗi trên tập kiểm tra được ký hiệu là $\mathcal{L}(\mathbf{W})$. Giả sử $\mathbf{y} = (y_1, \ldots, y_N)^\top$, và γ là bước của thuật toán SGD, $\gamma = \kappa C_1 \frac{\sqrt{\mathbf{y}^T(\mathbf{H}^\infty)^{-1}\mathbf{y}}}{m\sqrt{N}}$ với một hằng số tuyệt đối đủ nhỏ κ. Dưới giả định của Định lý 2, cho bất kỳ $\delta \in (0, e^{-1}]$, tồn tại $m^*(\delta, N, \lambda_0)$, sao cho nếu $m \geq m^*$, thì với xác suất ít nhất $1 - \delta$, ta có:

$$\mathbb{E}[\mathcal{L}(\mathbf{W})] \leq \mathcal{O}\left(C'\sqrt{\frac{\mathbf{y}^T\mathbf{y}}{p_0 N}}\right) + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right)$$

trong đó C, C', δ là các hằng số.

Zero-cost proxy mới: MeCo

- Một lớp CNN đa kênh có thể được xem như một lớp mạng nơ-ron quá tham số với nhiều mẫu.
- Giá trị riêng nhỏ nhất của $P(\mathbf{X})$ ảnh hưởng tích cực đến tốc độ hội tụ của quá trình huấn luyện và khả năng tổng quát hóa của mạng nơ-ron, độc lập với các nhãn.

Từ đó, nhóm tác giả khai thác giá trị riêng nhỏ nhất của ma trận tương quan Pearson trên từng lớp của các bản đồ đặc trưng để tạo ra đại diện chi phí không mới của chúng tôi được gọi là MeCo, theo cách sau:

Định nghĩa (MeCo). Giả sử mạng nơ-ron $f(;\theta)$ có tổng cộng D lớp, thì MeCo được định nghĩa là:

$$\text{MeCo} = \sum_{i=1}^{D} \lambda_{\min}(P(f^i(\mathbf{X};\theta)))$$

Algorithm 1 MeCo-based NAS

Require: A_0: An untrained supernet; $\mathcal{E}$: The set of edges in search cells; $\mathcal{N}$: The set of nodes in search cells; $\mathcal{O}$: The set of candidate operations; N: the number of candidate networks.

Ensure: The best network A_{best}.

// Stage 1: Architecture Proposal

$C = \emptyset$;

for $i = 1; i \leq N; i++$ **do**

  for $j = 1; j \leq |\mathcal{E}|; j++$ **do**

    Randomly choose an un-discretized edge e_t

    Choose the best edge from the supernet, s.t.

$$e_{t,best} = \arg\min_{1 \leq k \leq |\mathcal{O}|} \mathsf{MeCo}(A_0 / e_{t,k})$$

    Use operation $e_{t,best}$ to substitute e_t

  end for

  $A_{|\mathcal{E}|}$ consists of $\{e_{t,best} | 1 \leq t \leq |\mathcal{E}|\}$

  for $j = 1; j < |\mathcal{N}|, j++$ **do** ▷ prune the edges of the obtained architecture $A_{|\mathcal{E}|}$

    Randomly select an unselected node $n \in \mathcal{N}$

    for $k = 1; k < |\mathcal{E}(n)|; k++$ **do**

      Calculate MeCo of the architecture $A_{|\mathcal{E}|} / e_n^k$

    end for

    Retain edges e_n^1, e_n^2 with the 1st and 2nd best MeCo value, and remove the other edges

  end for

  Get the candidate networks A_i that consist of $\{e_{t,best} | 1 \leq t \leq \mathcal{E}\}$, and append it to the set C

end for

// Stage 2: Architecture Validation

Get the best network:

$$A_{best} = \arg\max_{1 \leq i \leq N} \mathsf{MeCo}(A_i),\ s.t.\ A_i \in C$$

Table 1: Spearman correlation coefficients ρ of proxies on NATS-Bench-TSS and NATS-Bench-SSS

Approach	NATS-Bench-TSS			NATS-Bench-SSS		
	CIFAT-10	CIFAR-100	ImageNet16	CIFAT-10	CIFAR-100	ImageNet16
grasp	0.39	0.46	0.45	-0.13	0.01	0.42
fisher	0.40	0.46	0.42	0.44	0.55	0.47
grad_norm	0.42	0.49	0.47	0.51	0.49	0.67
snip	0.43	0.49	0.48	0.59	0.62	0.76
synflow	0.74	0.76	0.75	**0.81**	0.80	0.57
#Param	0.72	0.73	0.69	0.72	0.73	0.84
NWOT	0.77	0.80	0.77	0.45	0.43	0.42
jacov	0.73	0.70	0.70	0.30	0.13	0.30
NTK	0.76	0.75	0.72	0.34	0.29	0.28
zen	0.38	0.36	0.40	0.69	0.71	0.87
KNAS	0.20	0.35	0.42	0.25	0.12	0.32
NASI	0.44	0.43	0.63	0.17	0.04	0.20
GradSign	0.77	0.79	0.78	0.21	0.16	0.04
ZiCo	0.80	0.81	0.79	0.73	0.75	**0.88**
MeCo **(Ours)**	**0.894±0.003**	**0.883±0.005**	**0.845±0.004**	-0.79±0.01	**-0.87±0.01**	-0.86±0.02
MeCo_{opt} **(Ours)**	**0.901±0.002**	**0.890±0.003**	**0.850±0.003**	**0.89±0.002**	**0.83±0.004**	**0.89±0.003**

Table 4: Comparisons of MeCo-based NAS with baselines using DARTS-CNN and CIFAR-10

Approach	Test Error (%)	Search Cost (GPU Days)	Params (M)	Method
AmoebaNet-A [56]	3.34 ± 0.06	3150	3.2	MS
PNAS [57]	3.41 ± 0.09	225	3.2	MS
DARTS (1st) [6]	3.00 ± 0.14	1.5	3.3	OS
DARTS-PT[58]	2.61 ± 0.08	0.8	3.0	OS
SDARTS-RS [59]	2.67 ± 0.03	0.4	3.4	OS
SGAS (Cri.1)[60]	2.66 ± 0.24	0.8	3.7	OS
Eigen-NAS[26]	7.4	-	-	ZS
TE-NAS [24]	2.63 ± 0.064	0.05	3.8	ZS
Zero-Cost-PT$_{\text{synflow}}$[22]	2.96 ± 0.11	0.03	5.1	ZS
Zero-Cost-PT$_{\text{fisher}}$[23]	3.12 ± 0.16	0.05	2.5	ZS
Zero-Cost-PT$_{\text{grasp}}$ [21]	2.73 ± 0.10	0.1	3.3	ZS
Zero-Cost-PT$_{\text{jacov}}$[7]	2.88 ± 0.15	0.04	3.5	ZS
Zero-Cost-PT$_{\text{snip}}$ [20]	2.90 ± 0.03	0.04	4.0	ZS
Zero-Cost-PT$_{\text{NTK}}$ [24]	2.89 ± 0.09	0.21	4.1	ZS
Zero-Cost-PT$_{\text{ZiCo}}$[9]	2.80 ± 0.03	0.04	5.1	ZS
Zero-Cost-PT$_{\text{MeCo}}$(Ours)	**2.69± 0.05**	**0.08**	4.2	ZS

Tối ưu MeCo_{opt}

- Sự tương quan tiêu cực trên NATS-Bench-SSS và một số nhiệm vụ khác phản ánh rằng MeCo rất nhạy cảm với số lượng kênh.
- Mặc dù hiện tượng này không mâu thuẫn với kết quả lý thuyết hoặc làm giảm hiệu quả của MeCo, nó có thể gây ra vấn đề khi $p_0 = 0$.
- Để giải quyết điều này, nhóm tác giả trình bày một phương pháp tối ưu hóa trên cơ sở MeCo để giảm bớt tính nhạy cảm với kênh.

Cụ thể, đối với các lớp tích chập, $p_0 > 0$ nếu $\forall i \neq j : \mathbf{x}_i \not\parallel \mathbf{x}_j$ và $c < w \times h$, trong đó c là số lượng kênh, $w \times h$ là kích thước của đầu vào.

Do đó, trong các ứng dụng thực tế, MeCo có thể mất hiệu quả trong quá trình giảm mẫu. Để giải quyết vấn đề này, thay vì làm phẳng tất cả các kênh của bản đồ đặc trưng, chúng ta lấy mẫu ngẫu nhiên một số lượng kênh cố định và làm phẳng chúng thành ma trận $\mathbf{P}'$. Sau đó, tính kết quả cuối cùng bằng cách nhân giá trị riêng nhỏ nhất của $\mathbf{P}'$ với trọng số kênh, theo công thức:

$$\text{MeCo}_{\text{opt}} = \sum_{i=1}^{D} \frac{c^{(i)}}{n} \lambda_{\min}(\mathbf{P}')$$

với $c^{(i)}$ là số lượng kênh ở lớp thứ i và n là số lượng lấy mẫu cố định. Bằng cách này, thay vì tính toán ma trận Pearson kích thước lớn, ta chỉ việc tính toán và lấy mẫu các ma trận kích thước nhỏ hơn.