

Automatizovaný návrh neuronových sítí s ohledem na latenci pro edge akcelerátory

David Nepraš

Abstrakt

Klasifikátory obrazu jsou ve vestavěných systémech obvykle implementovány pomocí konvolučních neuronových sítí (CNN) urychlených specializovanými inferenčními akcelerátory. Představujeme metodu založenou na supersíti, která automatizuje návrh CNNs s cílem optimalizovat latenci na akcelerátoru Hailo-8L. Metoda využívá multiobjektivní genetický algoritmus (NSGA-II) spolu s předtrénovanou supersítí a prediktorem latence, což umožňuje rychlé vyhodnocování kandidátních architektur. Efektivita metody je ověřena na úloze klasifikace obrazu s ohledem na latenci na známých benchmarkových datasetech CIFAR-10 a CIFAR-100.

*[xnepa02@vut.cz](mailto:xnepra02@vut.cz), *Fakulta informačních technologií, Vysoké učení technické v Brně*

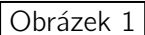
1. Úvod

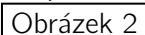
Hardware-aware Neural Architecture Search automatizuje návrh hlubokých neuronových sítí s ohledem nejen na přesnost, ale také na metriky jako latence či energetická náročnost inference [1]. Tento proces je výpočetně náročný, proto se někdy urychluje pomocí surrogate modelů.

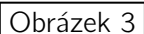
Tato práce se zabývá klasifikací obrazu pomocí *konvolučních neuronových sítí* (CNNs) ve vestavěných systémech. Cílem je optimalizovat CNN pro přesnost a latenci na akcelerátoru *Hailo-8L* [2]. Nízká latence je klíčová např. pro autonomní řízení.

Navrhujeme *latency-aware NAS* založený na multikriteriálním genetickém algoritmu, který nabízí řešení s dobrým kompromisem přesnosti a latence.

2. Návrh řešení

Navržená *NAS* metoda vyžaduje vyhodnocení přesnosti a latence kandidátních *CNN*, což je výpočetně náročné. Pro urychlení využívá metoda kombinaci dvou klíčových přístupů. Předtrénovaná supersít umožňuje efektivní hodnocení přesnosti – supersít (přeparametrizovaná *CNN*) obsahuje množství dílčích architektur (podsítí), které sdílejí váhy [3, 4, 5] . Díky tomu stačí pouze rychlá optimalizace vybrané podsítě namísto náročného trénování od nuly, což výrazně urychluje

proces *NAS*. Druhým prvkem je prediktor latence, jenž slouží jako náhrada za časově náročné přímé měření latence na akcelerátoru. Prediktor je implementován jako třívrstvá plně propojená neuronová síť (31-32-16-1). Síť přijímá jako vstup konfiguraci architektury podsítě a produkuje jediný výstup - odhadnutou latenci. 

Oba tyto přístupy jsou integrovány do vícekriteriálního evolučního algoritmu *NSGA-II* [6], který hledá řešení s optimálním kompromisem mezi přesností a latencí. 

Dále používáme dvě klíčová vylepšení pro redukcí výpočetního času:

2.1 Optimalizovaný fine-tuning

Při sdílení vah ze supersítě vyžadují podsítě pouze jemnou optimalizaci pro dosažení vysoké přesnosti. Navrhujeme hybridní přístup: první epocha trénuje celou síť, následné epochy pak pouze finální vrstvu. Tato strategie dosahuje 4× zrychlení při zachování kvality výsledků.

2.2 Redukce trénovacích dat během evoluce

Výsledky experimentů, ve kterých byla použita pouze 1/3 trénovací sady dat během evoluce demonstrují možnost výrazného urychlení *NAS* procesu bez kritického ovlivnění kvality výsledných modelů. Využitím

redukovaného trénovacího datasetu pak dochází ke $1.65\times$ zrychlení běhu algoritmu.

3. Nastavení experimentů

Pro porovnání výsledků jsou použity známé benchmarkové datasety *CIFAR-10* a *CIFAR-100*.

Na základě předběžných experimentů s ohledem na dostupné výpočetní zdroje a požadovanou kvalitu výsledků bylo zvoleno toto nastavení pro *NSGA-II*:

- Velikost populace: 28 jedinců
- Počet generací: 50
- Pravděpodobnost mutace: 0.35
- Pravděpodobnost křížení: 0.9

Všechny experimenty byly provedeny na výpočetním serveru (2× *Intel Xeon* - 16 jader, 256 GB RAM, 4× GPU: *NVIDIA RTX A5000*)

4. Výsledky

4.1 Výsledky predikce latence

Prediktor latence byl podroben důkladnému vyhodnocení. [Obrázek 4](#) znázorňuje korelaci mezi predikovanou a skutečně naměřenou latencí pro sítě z trénovací (modré body), testovací (zelené body) a validační (červené body) množiny. Pearsonův koeficient je 0.899 pro trénovací data a 0.946 pro validační data. Tyto výsledky potvrzují, že navržený prediktor poskytuje dostatečně přesné odhady pro potřeby metody NAS.

4.2 Návrh CNN pro CIFAR-10

Vybrali jsme tři CNN architektury (SCF10-1, SCF10-2 a SCF10-3) z finální Pareto fronty získané navrženou metodou a provedli jejich dodatečný trénink po dobu 5 epoch na celém trénovacím datasetu. I tento krátký trénink vedl k významnému zlepšení přesnosti o 4–6%, zatímco latence zůstala prakticky nezměněna. Při porovnání s běžně používanými architekturami [\[7\]](#) (viz. [Tabulka 1](#)) implementovanými pro *Hailo-8L* je zřejmé, že evolucí získané sítě nejsou přímo konkurenceschopné. Hlavním důvodem je to, že původní supersítě byla natrénována pro ImageNet, což vede k nadměrné komplexitě pro CIFAR-10.

4.3 Návrh CNN pro CIFAR-100

S využitím metody vyladěné na CIFAR-10 jsme aplikovali navržený NAS přístup na náročnější CIFAR-100. Vybrali jsme tři nejzajímavější architektury (SCF100-1, SCF100-2 a SCF100-3) z finální Pareto fronty a

provedli jejich dodatečný trénink po dobu 5 epoch na celém trénovacím datasetu CIFAR-100. Tento relativně krátký trénink vedl k výraznému zlepšení přesnosti o 12–14% [\[Tabulka 2\]](#). Přesnosti těchto navržených modelů překonaly i ručně navržené CNN architektury z [\[7\]](#). Tento úspěch však přichází za cenu vyšší latence, což koresponduje s našimi dřívějšími pozorováními na CIFAR-10.

5. Závěr

V této práci jsme představili novou metodu pro automatický návrh konvolučních neuronových sítí (NAS) optimalizovaných pro nízkou latenci na akcelerátoru *Hailo-8L*. Klíčovým přínosem je vyvinutí efektivního prediktoru latence, který umožňuje nahradit časově náročné měření na reálném hardwaru, čímž výrazně urychluje celý proces návrhu. Naše řešení je univerzální - nevyžaduje žádné specifické znalosti o architektuře akcelerátoru, a může být tedy snadno aplikováno na různé typy hardwarových platform.

Poděkování

Tímto bych rád poděkoval svému vedoucímu práce, panu prof. Ing. Lukáši Sekaninovi, Ph.D., za odborné vedení, konzultace a podnětné připomínky při tvorbě této práce. Dále děkuji panu doc. Ing. Jiřímu Jarošovi, Ph.D. za umožnění přístupu k výpočetnímu serveru.

Literatura

- [1] Hadjer Benmeziane, Kaoutar El Maghraoui, Hamza Ouarnoughi, Smail Niar, Martin Wistuba, and Naigang Wang. Hardware-aware neural architecture search: Survey and taxonomy. In *Proc. of the 30 Int. Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4322–4329, 2021. Survey Track.
- [2] Mas Nurul Achmadiyah, Afaroj Ahamad, Chi-Chia Sun, and Wen-Kai Kuo. Energy-efficient fast object detection on edge devices for iot systems. *IEEE Internet of Things Journal*, pages 1–15, 2025.
- [3] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proc. of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, pages 4092–4101. PMLR, 2018.

- [4] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentiveness: Improving neural architecture search via attentive sampling. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6414–6423, 2021.
- [5] Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernets with alpha-divergence. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 10760–10771. PMLR, 2021.
- [6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [7] Chenyaofu. PyTorch CIFAR models, 2025. Last accessed 25 April 2025.