Posudek oponenta bakalářské práce

Autor/autorka práce: **Matěj Černý**

Název práce: **Neuronové sítě – porovnání výkonnosti knihovny založené na PyTorch v Pythonu a C++**

Obsah práce

Práce se zabývá adaptací obalové knihovny pro framework PyTorch z jazyka Python do jazyka C++ a následným porovnáním těchto dvou verzí. Text práce je členěn na poměrně krátkou a stručnou teoretickou část a následnou rozsáhlejší část praktickou, ve které autor shrnuje i poznatky z výkonnostní analýzy obou verzí knihovny. V přílohách pak autor uvádí uživatelskou příručku, kde popisuje sestavení a spouštění knihovny a dodaných testů.

Kvalita řešení a dosažených výsledků

V textu práce je teoretická část věnující se problematice redukována do poměrně malého počtu stran. Autor již od počátku práce používá specifické termíny, z nichž nemalé množství ponechává nevysvětlených (GPU, CUDA, *bias*, aktivační funkce, tenzor, aj.). Teoretická část o neuronových sítích popisuje v podstatě pouze jeden druh neuronové sítě (vícevrstvý perceptron), přičemž další, mnohem významnější druhy ponechává pouze ve formě zmínky (RNN, CNN). Ty jsou pak z nějakého důvodu povrchně popsány v kapitole 7, která se věnuje testování. Rovněž v teoretické části postrádám analýzu toho, co by mohlo mít vliv na výkon, tj. motivaci přepisu knihovny do C++.

Kapitola 4 se pak věnuje rozdílům jazyka Python a C++ za účelem adaptace knihovny. Tento rozbor je přibližně dvakrát delší, než zbytek teoretické části.

Kapitola 5 pak operuje s nevysvětlenými termíny, které měly být zmíněny v teoretické části – metrika, *loss* funkce, *batch size*, *epoch count* a jiné.

Kapitola 6 zmiňuje architekturu Unet, která je blíže popsána až v kapitole 7. Podkapitola 6.1 z nějakého důvodu popisuje vlastnosti jazyka C++17 – tento rozbor by patřil spíše do kapitoly 4.

V kapitole 7, jak již bylo zmíněno, je spousty textu, které patří spíše do kapitol 2, 3 a 4. Doporučení o volbě počtu pracovníků není citováno, ani není diskutováno, proč by mohlo mít vliv na výkon, když podstatná část programu běží na GPU.

Programové řešení je vyhotoveno v uspokojivé kvalitě. Narazil jsem pouze na minimální množství prohřešků co se použití jazyka C++ týče. Kód jinak odpovídá modernímu pojetí jazyka C++ a obsahuje i konstrukce novějších standardů (C++17).

Formální úroveň

V textu jsou četně využívány anglicismy i přesto, že existují ustálené české varianty (např. *templaty* – šablony, *loss* funkce – ztrátová funkce, *macro* – makro, aj.). Text obsahuje minimální množství překlepů a typografických chyb. Hlavním jazykem textu je čeština, grafy a tabulky jsou však popsány v anglickém jazyce.

Až na nepříliš šťastné strukturování zmíněné v předchozím bodě je formální úroveň v pořádku.

Práce s literaturou

Práce cituje 29 zdrojů, z nichž 7 jsou vědecké publikace a knihy a zbytek vybrané webové stránky a dokumentace technologií a knihoven. Zdroje jsou citovány korektně.

Splnění zadání

Teoretická část je poměrně strohá a splňuje bod zadání č. 1 pouze minimalisticky, byť by k uspokojivé míře stačilo práci trochu jinak strukturovat. I přesto tento nedostatek považuji zadání za splněné.

Dotazy k práci

1. Jaká byla úvodní očekávání co do výkonu obou řešení? Bylo očekáváno, že C++ verze bude výrazně rychlejší, než je skutečnost změřená v následné implementaci?
2. Jaký je reálný vliv počtu workerů na výkon? Pokuste se vysvětlit, proč např. C++ verze nejrychleji trénuje s počtem n=4 a Python verze s n=1.
3. Pokuste se stručně popsat kroky, které by bylo nutné podniknout k podpoře AMP ve Vaší implementaci (za předpokladu, že by tuto funkci měla i podlehlá knihovna).

Navrhuji hodnocení známkou **velmi dobře** a práci doporučuji k obhajobě.

V Plzni 22. 5. 2023 Ing. Martin Úbl