Kraków, 27 marca 2021 r.

Marek Skomorowski (prof. dr hab. inż.)
Instytut Informatyki i Matematyki Komputerowej
Uniwersytetu Jagiellońskiego

Recenzja rozprawy doktorskiej
Pana mgra inż. Dominika Żurka
zatytułowanej
Akceleracja obliczeń algorytmów uczenia maszynowego oraz wybranych populacyjnych algorytmów inteligencji obliczeniowej ze zredukowaną precyzją danych poprzez implementację w układach GPGPU

Przewód doktorski jest prowadzony przez Radę Dyscypliny Informatyka Techniczna i Telekomunikacja Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. (RDITiT AGH). Recenzję napisano na zlecenie Przewodniczącego RDITiT AGH, Pana prof. dra hab. inż. Marka Kisiela-Dorohinickiego (pismo z dnia 26.01.2021 r.).

1. Omówienie ogólne, cel i teza rozprawy.

Problematyka rozprawy dotyczy sprzętowego przyspieszania obliczeń wybranych algorytmów sztucznej inteligencji za pomocą ich implementacji w procesorach graficznych ogólnego przeznaczenia GPGPU (*general-purpose graphics processing unit*) wykorzystując zredukowaną precyzję danych (*reduced data precision*). Stale rosnące zapotrzebowanie na moc obliczeniową uzasadnia prowadzenie badań dotyczących zastosowania sprzętowego przyspieszania obliczeń w celu zwiększania mocy obliczeniowej. Za pomocą układów GPGPU wiele obliczeń, przede wszystkim obliczeń równoległych, można przeprowadzić szybciej niż za pomocą układów CPU (*central processing unit*). Rozprawa dotyczy zatem ważnego, z praktycznego punktu widzenia, obszaru badań.

W pierwszym rozdziale przedstawiono motywację i cel podjętych badań, jak również tezę rozprawy. Motywacją było poszukiwanie rozwiązań przyspieszających obliczenia wybranych algorytmów uczenia maszynowego i populacyjnych algorytmów inteligencji obliczeniowej, a tezę rozprawy sformułowano następująco:

Implementacja algorytmów inteligencji obliczeniowej oraz uczenia maszynowego w akceleratorach GPGPU prowadzi do przyśpieszenia ich wykonania w stosunku do implementacji CPU. Użycie w implementacji sprzętowej zredukowanej precyzji danych, poprawia szybkość kart graficznych, nie zawsze prowadząc do pogorszenia jakości algorytmu.

Pierwszy rozdział zawiera również opis zawartości rozprawy.

2. Organizacja i zawartość rozprawy.

Rozprawa liczy 138 stron i składa się z siedmiu rozdziałów, spisu treści i wykazu literatury zawierającego 129 pozycji. Do rozprawy jest dołączony załącznik w postaci płyty DVD zawierającej oprogramowanie stanowiące integralną cześć rozprawy.

W drugim rozdziale przedstawiono pojęcia sztucznej inteligencji, inteligencji obliczeniowej i algorytmów ewolucyjnych. Dokonano podziału algorytmów ewolucyjnych na algorytmy genetyczne, strategie ewolucyjne, programowanie ewolucyjne i programowanie genetyczne. Opisano mechanizmy selekcji, mutacji i reprodukcji wzorowane na biologicznych procesach ewolucji. Przedstawiono pojęcie uczenia maszynowego z podziałem na uczenie nadzorowane, nienadzorowane i przez wzmacnianie, jak również pojęcie głębokiego uczenia maszynowego. Opisano pojęcie agenta, będącego podstawą agentowych metod inteligencji obliczeniowej w tym ewolucyjnego i stadnego systemu wieloagentowego. Opisano sieci neuronowe z podziałem na konwolucyjne (splotowe) i rekurencyjne. W ramach rekurencyjnych sieci neuronowych przedstawiono podejście zwane *long short-term memory* (LSTM), jak również opisano mechanizm propagacji wstecznej. Drugi rozdział nie zawiera oryginalnych osiągnięć Autora, jest jednak niezbędnym wprowadzeniem do problematyki rozprawy.

W trzecim rozdziale przedstawiono ogólną budowę układów GPGPU do obliczeń równoległych typu SIMD (*single instruction multiple data*) i typu SIMT (*single instruction multiple threads*), jak również budowę karty graficznej Nvidia Tesla V100 SXM2 32 GB. Opisano platformę obliczeń równoległych CUDA (*compute unified device architecture*). Trzeci rozdział nie zawiera oryginalnych osiągnięć Autora, jest jednak niezbędną częścią rozprawy opisującą układ GPGPA wykorzystywany przez Autora do implementacji omawianych algorytmów.

W czwartym rozdziale przedstawiono problem sekwencji binarnej o niskiej autokorelacji (*low autocorrelation binary sequence*) LABS. Problem ten jest przedmiotem badań od lat sześćdziesiątych dwudziestego wieku, a zainteresowanie nim bierze się stąd, że z jednej strony ma wiele zastosowań w różnych dziedzinach, na przykład w telekomunikacji, fizyce i sztucznej inteligencji, z drugiej zaś jest trudnym problemem optymalizacji kombinatorycznej (problem NP-trudny). Problem LABS sformalizowano i sformułowano w kategoriach znajdowania ciągów binarnych z minimalnymi poziomami energii, a wskaźnikiem jakości jest w tym przypadku *merit factor*. Opisano następujące dwie strategie lokalnego wyszukiwania: pierwsza z nich to SDLS (*steepest descent local search*), druga natomiast wykorzystuje przeszukiwanie *tabu search* (TS). Obie strategie przedstawiono w postaci pseudokodów i zilustrowano przykładami.

Następnie opisano równoległą implementację algorytmu SDLS w postaci pseudokodu dla układu GPGPU do problemu LABS. Obliczenia dla równoległej implementacji wykonano zarówno w układzie GPGPU (Nvidia Tesla V100 SXM2 32 GB), jak również w układzie CPU (Intel Core i7-9750H CPU 2.60 GHz). W przypadku obliczeń w układzie CPU wykorzystano bibliotekę OpenMP (*open multi-processing*). Wyniki eksperymentów porównano zarówno jeśli

chodzi o czasy obliczeń, jak również o przyspieszenia obliczeń. Z analizy przeprowadzonych eksperymentów wynika, że wszystkie obliczenia w układzie GPGPU wykonały się szybciej niż w układzie CPU. Opisano również równoległą implementację algorytmu TS w postaci pseudokodu dla układu GPGPU i wykonano obliczenia. Z analizy przeprowadzonych eksperymentów wynika że czasy obliczeń w układzie GPGPU algorytmów TS i algorytmu SDLS z równoległą implementacją są porównywalne ze sobą.

Kolejny fragment czwartego rozdziału dotyczy badania skuteczności algorytmów SDLS i TS do problemu LABS. W tym celu oba algorytmy zaimplementowano jako część aplikacji EMAS (*evolutionary multi-agent systems*), w której ewolucyjna część była wykonywana w układzie CPU, natomiast lokalna optymalizacja była wykonywana w układzie GPGPU. Wyniki przeprowadzonych eksperymentów opisano za pomocą tabel i wykresów przedstawiających liczbę kroków, najlepszą wartość, jak również przyspieszenie dla układów CPU (Intel Core i5-6600 3.30 GHz) i GPGPU (NVIDIA GeForce GTX 1060 6 GB) dla ciągów o różnych długościach. Wyniki przeprowadzonych eksperymentów pokazują, że dzięki zastosowaniu układu GPGPU było możliwe poszukiwanie większej liczby rozwiązań w tym samym czasie niż gdyby wszystkie obliczenia były wykonywane przez układ CPU, a uzyskane wyniki były lepsze ze względu na *merit factor*.

W dalszej część czwartego rozdziału opisano zaproponowane przez Autora dwie oryginalne modyfikacje algorytmu SDLS do problemu LABS. Pierwsza z nich to algorytm nazwany SDLS-2, w którym oprócz poszukiwania najlepszego rozwiązania w sąsiedztwie oddalonym o 1 bit w stosunku do wejściowego ciągu, poszukuje się także rozwiązania w obszarze oddalonym od niego o 2 bity. Poszukiwanie to odbywa się w dwóch krokach, przy czym przez jeden krok rozumie się w tym przypadku jeden obieg zewnętrznej pętli algorytmu (SDLS-2 bazuje na dwóch pętlach: wewnętrznej i zewnętrznej). Przedstawiono w postaci pseudokodów zarówno wersję sekwencyjną, jak również równoległą (implementacja w GPGPU) algorytmu SDLS-2. W przypadku drugiej z zaproponowanych przez Autora modyfikacji, nazwanej algorytmem SDLS przeszukującym w głąb, poszukiwanie najlepszego rozwiązania w sąsiedztwie oddalonym o 1 bit i o 2 bity w stosunku do wejściowego ciągu odbywa się w jednym kroku, przy czym przez jeden krok rozumie się również i w tym przypadku jeden obieg zewnętrznej pętli algorytmu (SDLS przeszukujący w głąb bazuje na dwóch pętlach: wewnętrznej i zewnętrznej). Przedstawiono w postaci pseudokodu zarówno wersję sekwencyjną, jak również równoległą (implementacja w GPGPU) algorytmu SDLS przeszukującego w głąb. Opisano badania dotyczące skuteczności algorytmu SDLS-2 i algorytmu SDLS przeszukującego w głąb, w porównaniu do algorytmu SDLS do problemu LABS. Z analizy przeprowadzonych obliczeń wynika, że algorytm SDLS przeszukujący w głąb jest najskuteczniejszy dla wszystkich długości ciągów, a skuteczności algorytmów SDLS i SDLS-2 są porównywalne ze sobą.

Czwarty rozdział zawiera oryginale osiągnięcia Autora, a do najważniejszych z nich zaliczam:

- Implementację algorytmów SDLS i TS w układach GPGPU (dotychczas algorytmy te były implementowane w układach CPU i FPGA (*field programmable gate array*).

- Opracowanie dwóch metod nazwanych SDLS-2 i SDLS z przeszukiwaniem w głąb, do problemu LABS i ich implementację w układach GPGPU.

W piątym rozdziale przedstawiono problem przetwarzania języka naturalnego (*natural language processing*). Opisano następujące modele reprezentacji tekstu używane w przetwarzaniu języka naturalnego i wyszukiwaniu informacji: *bag of words*, TD IDF (*term frequency inverse document frequency*), *N-gram* i wektoryzacja tekstu. W ramach wektoryzacji tekstu przedstawiono następujące techniki: *Word2Vec*, *GloVe* (*global vectors for word representation*), *FastText*, ELMo (*embeddings from language models*), *Transformer* i BERT (*bidirectional encoder representations from transformers*). Przedstawiono ideę maszyny wektorów nośnych SVM (*support vector machine*), opisano zarówno liniowy, jak również nieliniowy klasyfikator SVM i proces uczenia klasyfikatora dwuklasowego. Przedstawiono dwa następujące podejścia do klasyfikacji wieloklasowej za pomocą SVM: *one-vs-all* (OvA) i *one-vs-one* (OvO).

Dalsza część piątego rozdziału, dotyczy problemu kwantyzacji, co w przypadku niniejszej rozprawy oznacza przedstawianie danych wejściowych ze zredukowaną precyzją, to znaczy reprezentowanie N-bitowych danych wejściowych za pomocą M-bitów, gdzie M < N. Korzyści wynikające z używania formatów liczb ze zredukowaną precyzją są następujące: wymagają mniej pamięci i mniejszej przepustowości pamięci, a operacje matematyczne są wykonywane szybciej, zwłaszcza w procesorach graficznych z rdzeniami *tensor core*. Należy podkreślić, że oprócz wymienionych korzyści używanie formatów liczb ze zredukowaną precyzją, to znaczy z mniejszą dokładnością, może pogorszyć skuteczność implementowanych algorytmów. W związku z tym Autor założył, że dopuszczalny spadek skuteczności implementowanych w ramach rozprawy algorytmów nie może być większy niż 1 % (w przybliżeniu). Następnie przedstawiono standard arytmetyki zmiennoprzecinkowej IEEE 754.

W kolejnej części piątego rozdziału Autor zaproponował dwie oryginalne metody kwantyzacji: pierwszą z nich nazwał *max magnitude dynamic fixed-point*, drugą natomiast *min-max dynamic fixed-point*. Obie metody opisano formalnie, a ich działanie zilustrowano przykładami. W celu wykazania słuszności następującego fragmentu tezy rozprawy: *Użycie w implementacji sprzętowej zredukowanej precyzji danych, poprawia szybkość kart graficznych, nie zawsze prowadząc do pogorszenia jakości algorytmu*, zaproponowane metody kwantyzacji zaimplementowano i wykorzystano w procesie uczenia (trenowania) SVM dokonującej klasyfikacji wieloklasowej, do realizacji której wykorzystano opisane wcześniej podejście OvA. Jako zbiory treningowe wykorzystano: *Reuters dataset-r8* i *WebKB*. Przedstawiono szczegóły implementacji i wykonano obliczenia zarówno w układzie CPU (Intel Xeon CPU E5 2630 v3 2.4 GHz), jak również w układzie GPGPU z rdzeniami *tensor core* (Nvidia Tesla V100 SXM2 32 GB). Z analizy przeprowadzonych obliczeń wynika, że jest możliwe uczenie SVM za pomocą danych ze zredukowaną precyzją z zachowaniem przyjętego w rozprawie założenia, że dopuszczalny spadek skuteczności SVM nie może być większy niż 1 % (w przybliżeniu).

Rozdział piąty zawiera oryginale osiągnięcia Autora, a do najważniejszych z nich zaliczam:

- Opracowanie dwóch metod kwantyzacji nazwanych *max magnitude dynamic fixed-point* i *min-max dynamic fixed-point* i ich implementację do treningu SVM.

W szóstym rozdziale przedstawiono następujące modele konwolucyjnych sieci neuronowych: VGG16 (*visual geometry group*), ResNet (*residual neural network*) i CNN-non static (*convolutional neural network*). Opisano obliczanie konwolucji w układach GPGPU metodą bezpośrednią, metodą mnożenia macierzy, za pomocą szybkiej transformaty Fouriera i za pomocą algorytmu Winograd. Przedstawiono metodę *weight pruning* (przycinanie wagowe), zmniejszającą rozmiar sieci neuronowej za pomocą zerowania wag o wartościach bliskich zeru. Fakt, że wytrenowana sieć neuronowa posiada wiele wag, których wartości są bliskie lub równe zeru był motywacją do podjęcia przez Autora udanej próby sprzętowego przyspieszania obliczania konwolucji za pomocą operacji na macierzach rzadkich wykonywanych w układach GPGPU. Zaproponowane podejście, nazwane konwolucją rzadką, zmniejsza liczbę operacji na macierzach rzadkich. Przeprowadzono również obliczenia, których celem było zbadanie wpływu zastosowania zredukowanej precyzji danych na szybkość obliczeń zarówno za pomocą algorytmu konwolucji rzadkiej, jak również za pomocą biblioteki cuDNN (NVIDIA CUDA *deep neural network*) dla opisanych wcześniej modeli konwolucyjnych sieci neuronowych: VGG16, ResNet i CNN-non static. Dokonano analizy przeprowadzonych obliczeń.

Rozdział szósty zawiera oryginale osiągnięcia Autora, a do najważniejszych z nich zaliczam:

- Opracowanie i implementację efektywnej metody obliczania konwolucji, za pomocą operacji na macierzach rzadkich w układach GPGPU, nazwanej konwolucją rzadką.

- Wskazanie przypadków, dla których obliczanie konwolucji za pomocą metody konwolucji rzadkiej jest bardziej efektywne od metod, których dostarcza biblioteka cuDNN.

- Przeprowadzenie badań dotyczących wpływu zastosowania zredukowanej precyzji danych na szybkość obliczania konwolucji zarówno za pomocą metody konwolucji rzadkiej, jak również metod, których dostarcza biblioteka cuDNN.

Siódmy rozdział stanowi podsumowanie rozprawy.

3. Omówienie wyników rozprawy.

Na oryginalny naukowy dorobek przedstawiony w rozprawie w zakresie problematyki sprzętowego przyspieszania obliczeń składają się następujące osiągnięcia Doktoranta:

1. Implementacja algorytmów SDLS i TS w układach GPGPU (dotychczas algorytmy te były implementowane w układach CPU i FPGA).

2. Opracowanie dwóch metod nazwanych SDLS-2 i SDLS z przeszukiwaniem w głąb, do problemu LABS i ich implementacja w układach GPGPU.

3. Opracowanie dwóch metod kwantyzacji nazwanych *max magnitude dynamic fixed-point* i *min-max dynamic fixed-point* i ich implementacja do treningu SVM.

4. Opracowanie i implementacja efektywnej metody obliczania konwolucji, za pomocą operacji na macierzach rzadkich w układach GPGPU, nazwanej konwolucją rzadką.

5. Wskazanie przypadków, dla których obliczanie konwolucji za pomocą metody konwolucji rzadkiej jest bardziej efektywne od metod, których dostarcza biblioteka cuDNN.

6. Przeprowadzenie badań dotyczących wpływu zastosowania zredukowanej precyzji danych na szybkość obliczania konwolucji zarówno za pomocą metody konwolucji rzadkiej, jak również metod, których dostarcza biblioteka cuDNN.

7. Przeprowadzenie eksperymentów w celu weryfikacji wszystkich zaproponowanych rozwiązań i interpretacja ich wyników.

4. Uwagi natury redakcyjnej.

- Zamiast „ … algorytmy genetyczne można podzielić na: …” (s. 14), powinno być „ … algorytmy ewolucyjne można podzielić na: …”.

- Zamiast „ … dla wybranego α_1, poszukuje α_1” (s. 93) powinno być „ … dla wybranego α_2, poszukuje α_1”.

- Następujące podrozdziały nie zostały uwzględnione w spisie treści: 5.1.2.1. Model *N-gramów* (s. 75); 5.1.3.1. Algorytm *Word2Vec* (s. 76); 5.1.3.2. Model GloVe (s. 78); 5.1.3.3. FastText (s. 79); 5.1.4.1. Model ELMo (s. 80); 5.1.4.2. Model Transformes (s. 82, powinno być „Transformer” jak w publikacji [117]); 5.1.4.3. Model Bert (s. 84).

- Brak konsekwencji w niektórych zapisach, na przykład: Softmax (s. 20), *Softmax* (s. 30), **Softmax** (s. 77), softmax (s. 80).

- Brak pełnych danych bibliograficznych w niektórych pozycjach bibliografii, na przykład:

  [1] Karel Adámek i in. GPU Fast Convolution via the Overlap-and-Save Method in Shared Memory. Paź. 2019.

  [37] Gongde Guo i in. “KNN Model-Based Approach in Classification” (sierp. 2004).

  [88] Alec Radford, Luke Metz i Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks” (list. 2015).

- Występują pomyłki językowe, na przykład: „internatu” (s. 8) zamiast „Internetu”; „zimniejsza się” (s. 14) zamiast „zmniejsza się”; „strategi” (s. 14) zamiast „strategii”; „koncepci” (s. 60) zamiast „koncepcji”; „tagże” (s. 78) zamiast „także”.

5. Podsumowanie.

Podsumowując, stwierdzam że:

1. Cel rozprawy został osiągnięty, a teza rozprawy została wykazana.

2. Rozprawa stanowi oryginalne rozwiązanie problemu naukowego z zakresu sprzętowego przyspieszania obliczeń i praktycznego zastosowania go do wybranych algorytmów sztucznej inteligencji, na co składają się osiągnięcia Doktoranta wymienione w trzecim rozdziale niniejszej recenzji, w punktach od 1 do 7.

3. Dobór literatury i jej przegląd dowodzą, że Doktorant zna problematykę będącą przedmiotem rozprawy. Wykaz literatury zawiera 6 pozycji z zakresu tej problematyki, których współautorem jest Autor rozprawy, co oznacza, że Doktorant ma już osiągniecia naukowe w tym zakresie.

4. Rozprawa dowodzi również, że Doktorant wykazał się umiejętnością samodzielnego prowadzenia badań naukowych.

Ustawa z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. z 2003 r. nr 65, poz. 595, art. 13, ust. 1) stanowi:

Rozprawa doktorska, przygotowywana pod opieką promotora, powinna stanowić oryginalne rozwiązanie problemu naukowego lub artystycznego oraz wykazywać ogólną wiedzę teoretyczną kandydata w danej dyscyplinie naukowej lub artystycznej, a także umiejętność samodzielnego prowadzenia pracy naukowej lub artystycznej.

Na podstawie punktów 1, 2, 3 i 4 podsumowania niniejszej recenzji stwierdzam, że rozprawa doktorska **spełnia warunki Ustawy** o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki. W związku z tym wnioskuję o przyjęcie rozprawy doktorskiej i dopuszczenie Pana mgra inż. Dominika Żurka do publicznej jej obrony w dziedzinie nauk inżynieryjno-technicznych w dyscyplinie informatyka techniczna i telekomunikacja.

Marek Skomorowski

Elektronicznie podpisany przez Marek Tadeusz Skomorowski
DN: c=PL, sn=Skomorowski, givenName=Marek Tadeusz, cn=Marek Tadeusz Skomorowski, serialNumber=PNOPL-51100102813
Data: 2021.04.01 21:08:26 +02'00'