

**Recenzja rozprawy doktorskiej mgr inż. Grzegorza Rafała Deca pt.
„Sprzętowa implementacja sieci LSTM”**

Niniejsza recenzja została przygotowana w odpowiedzi na pismo RE.530/14RDITiT/2024 Pana prof. dra hab. inż. Jacka Kluski, Przewodniczącego Rady Dyscypliny Informatyka Techniczna i Telekomunikacja z dnia 27 marca 2024r zawierające informację, że Rada Dyscypliny Informatyka Techniczna i Telekomunikacja Politechniki Rzeszowskiej w dniu 21 marca 2024r. powołała mnie na recenzenta w/w rozprawy doktorskiej.

1. Zakres tematyczny i struktura rozprawy.

Tematyka recenzowanej rozprawy obejmuje sprzętową implementację sztucznej sieci neuronowej typu LSTM na układach FPGA, a w szczególności sprzętową realizację procesu uczenia takiej sieci. W typowym przypadku uczenie sieci odbywa się na komputerze PC wyposażonym w kartę GPU. Jest to proces bardzo czasochłonny i zużywający dużo energii elektrycznej. Jednocześnie, sztuczne sieci neuronowe w tym sieci głębokie oraz rekurencyjne znajdują coraz liczniejsze zastosowania we współczesnym świecie. Niestety, powoduje to powstanie negatywnego oddziaływania na środowisko związanego z dużą ilością energii elektrycznej zużywanej przez systemy komputerowe, na których takie sieci są zaimplementowane. Wszelkie próby związane z optymalizacją takiego procesu, w tym poszukiwanie nowych rozwiązań sprzętowych są zatem bardzo ważne. Doktorant jako jeden z celów rozprawy wskazał opracowanie nowej metody implementacji sieci LSTM na układach FPGA umożliwiającej uzyskanie porównywalnych lub lepszych czasów obliczeń względem GPU lub znaczącego poprawienia wydajności obliczeniowej w sytuacjach, w których zastosowanie GPU lub TPU jest niemożliwe. Badania zrealizowane w ramach pracy dotyczyły tego właśnie obszaru, a w ramach rozprawy Doktorant przedstawił prawidłowe rozwiązanie napotkanych problemów.

Tematykę rozprawy uważam za aktualną i bardzo ważną w zakresie dyscypliny Informatyka Techniczna i Telekomunikacja.

Recenzowana rozprawa doktorska obejmuje 201 stron i składa się z 5 rozdziałów oraz bibliografii zawierającej 127 pozycji. Rozdział pierwszy wprowadza w tematykę oraz

MPŁYNEŁO

18. CZE. 2024



zawiera cel i zakres pracy. Rozdział drugi obejmuje opis narzędzi i technologii, jakie zostały wykorzystane podczas realizowanych badań. Rozdział trzeci opisuje sieci LSTM oraz metody ich sprzętowej implementacji na układach FPGA. Rozdział czwarty z kolei przedstawia sposób sprzętowej realizacji procesu uczenia sieci LSTM. Podsumowanie i wnioski końcowe zawarte są w rozdziale 5.

Struktura rozprawy jest prawidłowa, podział treści jest czytelny.

2. Zawartość rozprawy i ocena najważniejszych wyników

Recenzowana rozprawa doktorska jest obszerna i obejmuje szeroki zakres badań. Wynika to między innymi z przeprowadzenia szczegółowej analizy porównawczej wielu możliwych wariantów sprzętowej implementacji całej sieci LSTM i jej elementów składowych na układach FPGA. Niektóre z zaproponowanych przez Doktoranta rozwiązań, np. implementacja procesu uczenia sieci LSTM na FPGA, należy uznać za nowatorskie. Bardzo wartościowym elementem rozprawy jest też dobrze przeprowadzona oraz dosyć drobiazgowo opisana wspomniana już analiza porównawcza, jak również przeprowadzona analiza wpływu dokładności aproksymacji funkcji aktywacji nie tylko na proces wnioskowania, ale także na proces uczenia sieci. Zagadnienie to jest niezwykle ważne, gdyż związane są z nim takie parametry jak szybkość pracy sieci i jej dokładność, zarówno procesu wnioskowania jak i uczenia oraz ilość zużywanej energii elektrycznej sprzętowej platformy cyfrowej, na której sieć LSTM jest zaimplementowana. Zgodnie z opisem przedstawionym w rozprawie doktorskiej we współczesnej literaturze zagadnienia te nie były wystarczająco dobrze zbadane lub były niedostatecznie opisane.

Do najważniejszych osiągnięć Doktoranta zawartych w recenzowanej rozprawie należy zaliczyć następujące punkty:

1. Szczegółowa analiza różnych metod implementacji funkcji aktywacji na układach FPGA, ze szczególnym uwzględnieniem dokładności i szybkości działania, a także ilości zużywanych zasobów sprzętowych.

W ramach tego punktu badano metody aproksymacji bazujące na algorytmie CORDIC, wielomianach Czebyszewa oraz na wielomianach klasycznych. Przeanalizowano też wpływ zastosowanego rzędu algorytmu aproksymacji oraz liczby odcinków, na które podzielono dziedzinę aproksymacji na szybkość, dokładność i złożoność sprzętowej implementacji. Dokonano krytycznej analizy uzyskanych wyników i zestawiono je z wartościami zawartymi w adekwatnej literaturze. Wyniki uzyskane z tej części badań okazały się być niezwykle przydatne podczas realizacji następnego punktu, tj. sprzętowej implementacji komórki LSTM.

2. Implementacja modułu komórki oraz całej sieci LSTM na układach FPGA oraz przeprowadzenie badań eksperymentalnych związanych z wykorzystaniem takiej sieci do rozwiązania rzeczywistego procesu przemysłowego, tj. procesu kucia na zimno.

Jak wykazuje Doktorant, zaproponowany w tym punkcie rozprawy sposób implementacji znacząco różni się od dotychczas spotykanych metod i pozwala na osiągnięcie bardzo dobrych rezultatów. Potwierdzeniem tego jest zestawienie uzyskanych wyników z wynikami dostępnymi w literaturze.

3. Projekt, implementacja i testy sprzętowo-programowej metody implementacji sieci LSTM na układach FPGA-SoC (ZYNQ).

W ramach tego punktu zaproponowano i szczegółowo przebadano różne warianty współpracy uniwersalnego procesora (ARM Cortex-A9) wbudowanego w układ ZYNQ (moduł PS) i logiki programowalnej (moduł PL) w ramach sprzętowo-programowej implementacji sieci LSTM. Analizowano uzyskiwaną szybkość pracy oraz ilość zużywanych zasobów sprzętowych.

4. Analiza wpływu dokładności aproksymacji funkcji aktywacji na pracę sieci LSTM zaimplementowanej na układzie FPGA.

Istotnym elementem tego punktu badań jest analiza wpływu dokładności aproksymacji nie tylko na proces wnioskowania wykorzystany do zadania klasyfikacji, ale też na proces uczenia. Zgodnie z informacją zamieszczoną przez Doktoranta w rozprawie, w literaturze znajduje się zaledwie kilka prac obejmujących taką tematykę. Tą część pracy doktorskiej z całą pewnością można uznać za nowatorską. Należy również dodać, że zagadnienie uczenia rekurencyjnych sieci neuronowych, w tym bazujące na algorytmie BPTT, jest dosyć złożone i trudne do sprzętowej implementacji. Dlatego wkład Doktoranta w tym zakresie powinien być doceniony.

5. Przeanalizowanie wpływu liczby komórek LSTM na czas obliczeń i liczbę zużywanych zasobów sprzętowych przez sieć zaimplementowaną na układzie FPGA. W tym kontekście dokonano też praktycznej analizy kosztu realizacji proponowanych w ramach rozprawy podejść.

Ponadto przedstawione przez Doktoranta we wnioskach końcowych dalsze możliwe kierunki rozwoju są właściwe i świadczą o jego wiedzy w zakresie najnowocześniejszych rozwiązań, jakie pojawiają się na rynku. Pierwszy z wymienionych kierunków rozwoju dotyczy wykorzystania nowoczesnych układów z serii AMD Xilinx Versal ACAP, które posiadają nowatorską architekturę opartą na specjalizowanych modułach sprzętowych AI Engines, DSP Engines oraz NOC. Wymienione moduły doskonale nadają się do rozwiązań opisywanych w ramach recenzowanej rozprawy. Należy jednak dodać, że układy Versal ACAP są dostępne na rynku od niedawna i zrozumiałe jest, że Doktorant nie zastosował ich w ramach realizowanych dotychczas badań. Drugi z wymienionych

kierunków rozwoju dotyczy możliwości uwzględnienia różnych reprezentacji liczbowych do kodowania wartości przetwarzanych sygnałów oraz parametrów sieci LSTM. Jest to bardzo ważne zagadnienie, gdyż rodzaj zastosowanej arytmetyki w sposób znaczący wpływa na szybkość pracy, zużycie energii oraz złożoność obliczeniową. Trzeci z wymienionych kierunków rozwoju wydaje się być dosyć naturalny, gdyż dotyczy realizacji sprzętowej innych typów rekurencyjnych sieci neuronowych.

W trakcie realizacji tak wszechstronnych i rozbudowanych badań trudno ustrzec się przed popełnieniem drobnych błędów lub nieścisłości. W recenzowanej rozprawie błędy te nie są znaczące i nie wpływają na poprawność ogólnych wniosków sformułowanych przez Doktoranta. Poniżej przedstawiono szczegółową listę najważniejszych uwag merytorycznych i drobnych błędów edycyjnych wykrytych w rozprawie.

2.1. Uwagi merytoryczne

2.1.1. Automat stanu opisany w punkcie 3.3.2 i przedstawiony na listingu 3.1 korzysta ze zmiennoprzecinkowych bloków arytmetycznych, takich jak: komparator, sumator/subtraktor, układ mnożący oraz dzielący. Wyjaśnienie znajdujące się dopiero kilka stron później informuje, że „omawiana implementacja wykorzystywała bloki arytmetyczne wykonujące układy kombinacyjne, co pozwala na wykorzystanie wyników już w następnym taktie zegara”. Pomijając kwestię edycyjną, tj. zbyt późne umieszczenie w tekście rozprawy tak ważnej informacji, istotnej dla zrozumienia zasady działania w/w automatu stanu, takie podejście może budzić uzasadnione wątpliwości z kilku powodów.

- Po pierwsze jest ono wysoce nieadekwatne do układów arytmetycznych przetwarzających liczby zmiennoprzecinkowe. Operacje zmiennoprzecinkowe z natury rzeczy najlepiej realizuje się w co najmniej kilku kolejnych krokach, co wynika, w typowym przypadku, z konieczności dodatkowego pre-/post- skalowania jednego lub obu argumentów oraz wyniku takich operacji.
- W pracy założono, że wszystkie w/w operacje arytmetyczne (komparacja, dodawanie/odejmowanie, mnożenie i dzielenie) wykonywane są w takiej samej ilości cykli zegara. Jednak optymalna liczba tych cykli mocno zależy od rodzaju operacji. W szczególności operacja dzielenia arytmetycznego jako najbardziej skomplikowana wymaga najczęściej co najmniej kilkunastu cykli zegara, aby mogła być optymalnie realizowana w strukturze FPGA. Z kolei operacja komparacji może być z powodzeniem wykonana nawet jako logika kombinacyjna lub w ramach co najwyżej kilku cykli zegara. Doktorant nawet zaobserwował to zjawisko, gdyż jak zanotował w rozprawie „większych wartości nie rozważano ze względu na maksymalną liczbę cykli w module komparatora wynoszącą 2”. W tabeli 3.1 ten parametr oznaczono jako „liczba cykli na operację”. W tekście rozprawy jest on

określony jako „nastawa bloków arytmetycznych” i rozważano tylko trzy wartości, tj. 0, 1 i 2.

Efekt nieadekwatności zastosowanego podejścia widoczny jest w ostatniej kolumnie danych w tabeli 3.1, gdzie uzyskiwana częstotliwość maksymalna tego rozwiązania jest na poziomie 33MHz. Jest to więc częstotliwość bardzo niska, kilkukrotnie niższa niż w innych podejściach implementacyjnych. Jak można przypuszczać z tego samego powodu zwiększenie parametru „nastawa bloków arytmetycznych” do wartości 1 lub 2 było niewystarczające dla optymalnego działania operacji arytmetycznego dzielenia i nie przyniosło znaczącego zwiększenia maksymalnej możliwej do uzyskania częstotliwości pracy takiego układu cyfrowego. Prawidłowym podejściem byłoby założenie, że każdy z w/w bloków arytmetycznych może pracować z inną wartością tej „nastawy”. Nieznacznie skomplikowałoby to wprowadzenie automat stanu, ale pozwoliłoby na uzyskanie znacznie lepszych efektów dla tego wariantu sprzętowej aproksymacji.

W innych wariantach aproksymacji opisywanych w rozprawie operacja dzielenia arytmetycznego nie występuje i problem ten nie został zaobserwowany. Na szczęście szybkość pracy opisanego wyżej pierwszego wariantu aproksymacji nie miała istotnego wpływu na dalszą analizę przeprowadzoną w rozprawie i poprawność wyciągniętych wniosków.

- 2.1.2. Doktorant w recenzowanej rozprawie nie zawarł wystarczająco szczegółowego opisu, w jaki sposób określana jest maksymalna uzyskiwana częstotliwość pracy (f_{max}) opisywanych układów cyfrowych. W rozprawie znajduje się tylko jedno zdanie odnoszące się do wartości tego parametru, tj. „obliczona za pomocą programu ISE Design Suite częstotliwość zegara”. Wartość f_{max} wyliczona przez środowisko projektowe dotyczy tylko danej implementacji i jest efektem próby dopasowania struktury połączeń FPGA do wymagań opisanych w plikach constraints. Jeśli te wymagania nie są zdefiniowane prawidłowo, tj. są znacznie zaniżone lub zawyżone względem możliwych do osiągnięcia, wtedy wyliczona wartość f_{max} może być zbyt niska. Ze względu na brak bardziej szczegółowego opisu trudno wywnioskować, czy wartość ta była wyznaczana przez Doktoranta w sposób prawidłowy.
- 2.1.3. Ograniczenie dziedziny aproksymacji do zakresu $\langle -6, 6 \rangle$ jest prawdopodobnie dobrym rozwiązaniem, ale w rozprawie brakuje wystarczająco precyzyjnego uzasadnienia takiego właśnie zakresu. Skoro w innym miejscu rozprawy Doktorant przyjął odgórnie, że projektowana aproksymacja powinna oferować błąd bezwzględny na poziomie 10^{-7} , to na podstawie tej wartości można przecież precyzyjnie wyznaczyć wartość argumentu, powyżej którego aproksymowana funkcja nie różni się więcej niż o wspomnianą wartość od wartości stałej, przyjmowanej odgórnie dla takiego zakresu.

- 2.1.4. W rozprawie nie podano informacji, w jaki sposób przechowywane są wartości współczynników wielomianów aproksymacyjnych w strukturze FPGA, to znaczy czy w pamięci BRAM, czy są to wartości zakodowane w postaci rozproszonej w blokach LUT. Informacja o ilości i rodzaju zasobów zużywanych przez w/w współczynniki ma istotne znaczenie ze względu na kilka aspektów. Po pierwsze, przy podziale dziedziny aproksymacji na dużą ilość, np. kilkaset przedziałów, zużycie zasobów może być istotnie większe niż przy zastosowaniu tylko kilku przedziałów. Fakt ten wydaje się być jednak sprzeczny z informacją zawartą na str. 80, gdzie napisano, że „zwiększenie liczby przedziałów znacząco polepsza dokładność przy stosunkowo niewielkim dodatkowym zużyciu zasobów sprzętowych”. Prawidłowa jest natomiast informacja zawarta na samym dole str. 92, tj. „Niesie to za sobą zwiększenie zużycia zasobów ze względu na większą liczbę współczynników, które muszą być przetrzymywane w pamięci układu”.
- 2.1.5. Na str. 107 zawarto informację, że „parametry sieci zostały więc zaszyte w poszczególnych sumatorach”. Wynika z tego brak możliwości modyfikacji tych parametrów na etapie działania takiej sieci, a więc brak również możliwości uczenia/douczenia takiej sieci w działającym systemie. Pozostaje to w sprzeczności z opisywaną w dalszych częściach rozprawy procedurą uczenia sieci zaimplementowanej na FPGA.
- 2.1.6. Na str. 116 zawarto informację, że rozważana dokładność aproksymacji funkcji aktywacji na poziomie 10^{-7} „to nieco zawyżona wartość”. Odnosząc tą informację do tekstu rozprawy zawartego na str. 175, gdzie prawidłowo napisano „koszt finansowy układów FPGA ... dla procesu uczenia sieci, jest zauważalnie większy, niż w przypadku zadania klasyfikacji”, odnosi się wrażenie, że pewne sprawy pozostały niedopowiedziane. Z przeprowadzonych badań wynika bowiem, że optymalnym rozwiązaniem byłoby zaprojektowanie dwóch sieci o różnych właściwościach. Pierwsza o dużej precyzji, wykorzystywana do procesu uczenia i testowania, druga wyłącznie do procesu wnioskowania. Ta druga sieć byłaby gotowa do zastosowania w docelowym rozwiązaniu. Potwierdzeniem tego faktu jest analiza zawarta na str. 178, z której wynika, że dla zadania wnioskowania wystarczająca jest dokładność aproksymacji funkcji aktywacji na poziomie 10^{-4} , podczas gdy dla procesu uczenia sieci minimalna wymagana precyzja w/w aproksymacji jest na poziomie 10^{-6} . Szkoda, że w rozprawie nie zawarto krytycznej, końcowej analizy również tej części uzyskanych wyników i adekwatnych wniosków.
- 2.1.7. W tabeli 3.23 zawarto zestawienie czasów klasyfikacji uzyskanych dla sieci zaimplementowanej na różnych platformach cyfrowych. Pewne wątpliwości budzi fakt, że czas obliczeń dla platformy Raspberry Pi 3 jest dłuższy niż dla platformy ZYNQ PS (ARM). Pierwsza z platform wyposażona jest w nowocześniejszy procesor

ARM Cortex-A53 pracujący z częstotliwością 1,2GHz, podczas gdy druga wyposażona jest w słabszy procesor ARM Cortex-A9, pracujący z częstotliwością ok. 0,67GHz. Sugestie Doktoranta, że ma na to wpływ system operacyjny są mało przekonujące, w szczególności biorąc pod uwagę dane zawarte w tabeli 4.6, które wyglądają już bardziej wiarygodnie. Zawarte w rozprawie wyniki budzą wątpliwości dotyczące precyzji pomiaru czasów obliczeń na platformach referencyjnych.

2.1.8. Z danych zawartych w kilku tabelach rozprawy wynika, że czasy niektórych obliczeń realizowane na FPGA są nawet kilkaset razy szybsze niż na platformach referencyjnych, w tym na GPU. Jak zauważa Doktorant, powodem mógł być zbyt mały wymiar obliczeń, który nie pozwolił na wykorzystanie w pełni możliwości kart graficznych. Zatem bardziej miarodajne wyniki dla platformy GPU mogłyby być uzyskane, gdyby wykonano próby dla kilku sieci o znacznie różniących się rozmiarach. Jednak wpływ w/w mało dokładnego pomiaru na wnioski końcowe pracy nie jest jakościowy, a jedynie ilościowy.

2.2. Uwagi edycyjne

- 2.2.1. Na str. 43 znajduje się nieprecyzyjne sformułowanie „obliczenia w obrębie trwają tak długo, jak długa jest analizowana sekwencja”. Podobne sformułowanie przewija się w rozprawie kilkukrotnie. W żadnym miejscu rozprawy nie znalazłem dokładniejszego zdefiniowania tego procesu, ani podanej wartości parametru ‘n’ zawartego na rys. 3.14.
- 2.2.2. Na str. 47 znajduje się nieprecyzyjne sformułowanie „dokładność aproksymacji ... przyjmuje się na poziomie 10^{-7} dla zakresu $\langle -6, 6 \rangle$...”. A co z zakładaną dokładnością aproksymacji poza tym zakresem?
- 2.2.3. Na str. 50 zamieszczona jest informacja o nierealnie dużej wielkości, tj. „... tablicy LUT o wielkości $2 \cdot 10^{15}$...” – prawdopodobny błąd edytorski.
- 2.2.4. Doktorant na str. 52 rozprawy przytacza wniosek z cytowanej publikacji [91] mówiący o tym, że „bezpośrednia aktywacja funkcji aktywacji może znacząco zredukować ilość wykorzystywanych zasobów oraz przyspieszyć czas obliczeń, w przeciwieństwie do implementacji obliczającej najpierw funkcję eksponenty”. Następnie bez żadnego komentarza wybiera jako pierwszy wariant implementacji rozwiązanie wskazane jako gorsze. Na szczęście kolejne trzy warianty implementacji wykorzystują już rozwiązanie bezpośrednio.
- 2.2.5. Na str. 53 znajduje się trochę niezgrabne sformułowanie „aproksymację funkcji przeprowadzono w przedziale $\langle -6, 6 \rangle$ ponieważ ... większy obszar wiązałby się z koniecznością wykonywania porównań najpierw dla $|x| > 6$ ”. Poza tym to zdanie nie

- pasuje do kodu zawartego na listingu 3.1, gdzie jest zastosowana taka operacja arytmetycznego porównania.
- 2.2.6. W tabelach 3.1 i 3.2 znajduje się nieadekwatny odnośnik „(Rys. 2.7)” – prawdopodobnie błąd edytorski.
- 2.2.7. Na str. 71 znajduje się sformułowanie „w obu przypadkach kodowania ... część ułamkowa w słowie kodowym miała długość 31 bitów...”. Wspomniane oba przypadki kodowania to arytmetyka stałoprzecinkowa oraz zmiennoprzecinkowa. W zakresie kodowania w oparciu o arytmetykę zmiennoprzecinkową nie istnieje pojęcie „części ułamkowej”, chyba że w odniesieniu do samej mantysy, a nie do wartości kodowanej liczby.
- 2.2.8. Nie jest jasne czy tabela 3.5 prezentuje wyniki dla arytmetyki stałoprzecinkowej czy zmiennoprzecinkowej. Wydaje się, że opis w tekście (pierwsze zdanie na str. 71) oraz opis w nagłówku tabeli są sprzeczne.
- 2.2.9. W tekście na str. 71 pojawiło się określenie „kodowanie U2” bez wystarczającego wyjaśnienia. Można się tylko domyślać, że chodzi o jeden z formatów kodowania stałoprzecinkowego, ale nie dostarczono żadnych informacji o położeniu punktu ułamkowego w ramach przetwarzanego słowa binarnego. Poza tym brakuje informacji, czy w ramach implementacji stałoprzecinkowej wszystkie sygnały i parametry sieci LSTM miały taki sam format, tj. położenie punktu ułamkowego w każdym kroku algorytmu aproksymacji, czy też wartości te zostały ustalone indywidualnie w oparciu o jakieś przyjęte założenia.
- 2.2.10. W opisie do tabeli 3.6 na str. 72 zawarta jest informacja, że „w tabeli 3.6 przedstawiono wyniki implementacji dla arytmetyki zmiennoprzecinkowej zgodnej z IEEE754”. W opisie tabeli 3.5 zawartym w tekście nie występowała informacja o zgodności z normą „IEEE754”. Czy to oznacza, że te dwie arytmetyki różnią się czymś? A jeśli tabele 3.5 i 3.6 odnoszą się do takiej samej arytmetyki, to dlaczego wartości błędu maksymalnego różnią się? Jeśli pomiędzy wariantami reprezentowanymi przez tabele 3.5 i 3.6 istnieje wyłącznie różnica w ilości cykli zegara na operację arytmetyczną to wyniki powinny być identyczne.
- 2.2.11. W rozprawie jest trochę nieporządek w zakresie nagłówków tabel i podpisów do rysunków. Zazwyczaj na ich podstawie nie można jasno wywnioskować czy prezentowane na nich dane dotyczą implementacji w oparciu o arytmetykę stałoprzecinkową czy zmiennoprzecinkową, funkcji sigmoidalnej czy tangensa hiperbolicznego. Czasami prowadzi to do niejasności, np. z wykresu 3.27 wynika, że maksymalny błąd aproksymacji \tanh wynosi ok. $2,82 \cdot 10^{-7}$, natomiast z tabeli 3.6 wynika, że wartość ta wynosi $1,192 \cdot 10^{-7}$. Która wartość jest prawidłowa?

- 2.2.12. Na str. 74 znajduje się niezbyt fortunne sformułowanie „... stosunkowo dużo argumentów osiąga wartości powyżej połowy ...”. Prawdopodobnie chodziło o wartości błędu aproksymacji dla pewnych zakresów argumentów.
- 2.2.13. Na dole str. 75 rozprawy zamieszczona jest informacja, że na podstawie 10 najstarszych bitów sygnału wejściowego wybierana jest wartość współczynników wielomianu. W listingu 3.4 nie znalazłem adekwatnego kodu tego algorytmu wybierającego te współczynniki.
- 2.2.14. W punkcie 3.34 rozprawy nie napisano, za pomocą jakiego algorytmu (Taylora, funkcji minimalizacji normy błędu, itp.) obliczano wartości współczynników wielomianu. Informacja ta nie ma wprawdzie bezpośredniego wpływu na uzyskiwany czas obliczeń i sposób implementacji, ale może mieć wpływ na wartość uzyskiwanego błędu aproksymacji.
- 2.2.15. W tabeli 3.15 brak jest kolumny z czasem obliczeń poszczególnych wariantów implementacji, co zmusza czytelnika do samodzielnego i kłopotliwego przetwarzania w pamięci poszczególnych wartości liczbowych.
- 2.2.16. Na stronie 87 znajduje się niejasne sformułowanie „ponowne użycie bloków nie wprowadza dodatkowego opóźnienia”.
- 2.2.17. Na stronie 93 znajduje się niezrozumiałe sformułowanie „wykonanie szeregu obliczeń w komórce zauważalnie wpłynęło na osiąganą dokładność, która ... spadła o rząd wielkości”. Czy chodziło o „wykonanie szeregowo obliczeń” i efekt nawarstwiania się błędu aproksymacji?
- 2.2.18. W jakich jednostkach wyrażone są wartości na przebiegu widocznym na rys. 3.34 i jak mają się te wielkości (rzędu 1000, 2000) do sformułowania zawartego na str. 115 rozprawy, gdzie mowa o wartościach rzędu 0.49 i 0.50?
- 2.2.19. Na str. 158 zawarto raczej nieprawidłowe sformułowanie „dotyczących minimalnego czasów narastania sygnału zegara”.

3. Wniosek końcowy

Przeprowadzona przez Doktoranta w sposób właściwy krytyczna analiza rozwiązań opisywanych w literaturze świadczy o tym, że **prezentuje on wymaganą ogólną wiedzę teoretyczną osoby ubiegającej się o nadanie stopnia doktora**. Z kolei przeprowadzenie w sposób prawidłowy wszechstronnych, a przy tym dosyć szczegółowych badań dotyczących tematyki pracy wskazuje, że **Doktorant wykazuje się umiejętnością samodzielnego prowadzenia pracy naukowej, co jest wymaganiem stawianym osobie ubiegającej się o nadanie stopnia doktora**. Rozwiązania zawarte w rozprawie, w tym w szczególności zaproponowana sprzętowa realizacja procesu uczenia sieci LSTM,

przedstawiają **oryginalne rozwiązanie problemu naukowego**.

Oceniając rozprawę doktorską pt. „Sprzętowa implementacja sieci LSTM” stwierdzam, że spełnia ona wymagania określone w ustawie Prawo o szkolnictwie wyższym i nauce z dnia 20 lipca 2018 roku z późniejszymi zmianami. Rozprawa spełnia wszystkie wymagania ustawowo przyjęte w polskim środowisku naukowym stawiane rozprawom doktorskim i wnoszę o jej przyjęcie oraz dopuszczenie do dalszych etapów postępowania o uzyskanie stopnia naukowego doktora w dziedzinie nauk technicznych, w dyscyplinie Informatyka Techniczna i Telekomunikacja.

Jednocześnie, biorąc pod uwagę wysoki poziom naukowy recenzowanej rozprawy doktorskiej oraz ważność i aktualność poruszanej w niej tematyki składam wniosek o jej wyróżnienie.

A handwritten signature in blue ink, reading "Andrzej Przytycki". The signature is written in a cursive style with a horizontal line under the last part of the name.