ELEKTROTEHNIŠKI VESTNIK 89(3): 117–123, 2022
IZVIRNI ZNANSTVENI ČLANEK

Energijsko učinkovito računanje s približnimi množilniki

Ratko Pilipović, Patricio Bulić, Uroš Lotrič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija
E-pošta: ratko.pilipovic@fri.uni-lj.si

Povzetek. Množenje je zelo pogosta aritmetična operacija, srečamo jo v najrazličnejših aplikacijah, na primer pri obdelavi multimedijskih vsebin in v algoritmih strojnega učenja. Gre za računsko zahtevno in energijsko potratno operacijo, ki jo izvajajo namenska vezja – množilniki. V številnih aplikacijah natančno računanje ni potrebno, zato lahko natančne množilnike zamenjamo s približnimi. Pri načrtovanju približnih množilnikov iščemo rešitev, ki ponuja zadovoljivo natančnost za aplikacijo in je energijsko čim bolj učinkovita. Poznamo logaritemske, nelogaritemske in hibridne približne množilnike, ki se razlikujejo v načinu računanja približnega zmnožka, natančnosti in porabi energije. V delu predstavimo in primerjamo novejše približne množilnike iz vseh treh skupin. Množilnike smo sintetizirali v tehnologiji Nangate 45 nm CMOS in jih uporabili za obdelavo slik in razvrščanje vzorcev z globokimi nevronskimi mrežami. Rezultati kažejo, da je za logaritemske približne množilnike značilna majhna poraba energije na račun večje računske napake, za nelogaritemske približne množilnike pa je značilna majhna računska napaka in zato višja poraba energije; hibridni množilniki po natančnosti in energijski porabi sodijo med logaritemske in nelogaritemske rešitve.

Ključne besede: Približno računanje, energijsko učinkovito računanje, aritmetična vezja, približni množilniki.

Energy-efficient computing with approximate multipliers

Multiplication is a crucial and ubiquitous arithmetic operation in various applications, such as multimedia content processing and machine learning algorithms. It is a computationally demanding and energy-consuming operation performed by dedicated circuits, i.e., multipliers. As many applications do not require accurate calculations, the exact multipliers can be replaced with the approximate ones. Therefore, a solution is needed to offer a satisfactory accuracy and to be optimally energy-efficient. There are three types of the approximate multipliers available: logarithmic, non-logarithmic and hybrid. They differ in their product approximation, accuracy and energy consumption. The paper presents and compares the state-of-the-art approximate multipliers of each type. The multipliers are synthesized in the Nangate 45nm CMOS technology and used for image processing and classification with deep neural networks to determine their advantages and weaknesses. The results show that the logarithmic approximate multipliers have a low energy consumption but a higher error, while the non-logarithmic approximate ones have a small error and a higher energy consumption. The hybrid multipliers are somewhere between the logarithmic and non-logarithmic multipliers in terms of their accuracy and energy consumption. We show that the approximate logarithmic multipliers outperform in modelling with neural networks. However, the hybrid and non-logarithmic approximate multipliers offer better results in image processing.

Keywords: Approximate computing, energy-efficient computing, arithmetical circuits, approximate multipliers.

Prejet 4. april, 2022
Odobren 17. maj, 2022

1 UVOD

V zadnjem času je postalo približno računanje (angl. *approximate computing*) priljubljen pristop pri načrtovanju energijsko učinkovitih aritmetičnih vezij. Veliko raziskav je usmerjenih v načrtovanje približnih množilnikov, saj je množenje zahtevna in energijsko potratna operacija. S približnimi množilniki lahko dosežemo občutno manjšo porabo energije na račun manjše natančnosti izračunov, ki pa je še vedno zadovoljiva za številne aplikacije [1].

Med aplikacije, pri katerih lahko veliko pridobimo s približnimi množilniki, sodijo številni postopki obdelave slik in modeliranje z nevronskimi mrežami. Pri obdelavi slik opazovalec težko opazi majhne spremembe kakovosti slik zaradi napak pri računanju. Na ta račun različni algoritmi za obdelavo slik, kot sta rezanje šivov (angl. seam carving) [2] in izgubna kompresija slik [3], poenostavljajo računanje ali zagotavljajo manjšo prostorsko zahtevnost. Globoke nevronske mreže so zelo priljubljeni modeli stojnega učenja, ki s prilagajanjem prostih parametrov ali učenjem poskušajo povezati vhodne podatke z izhodnimi. Robustni postopki učenja nevronskim mrežam omogočajo, da se do določene mere pravilno odzivajo tudi na nepoznane ali šumne vhodne podatke. Podobno lahko z množico prostih parametrov kompenzirajo tudi računske napake. Na omenjenih lastnostih temelji uporaba tehnik približnega računanja v nevronskih mrežah, kot je kvantizacija [4]. Pri naštetih in številnih drugih aplikacijah natančno računanje ne

prinese očitne izboljšave rezultata, temveč predvsem večjo porabo energije.

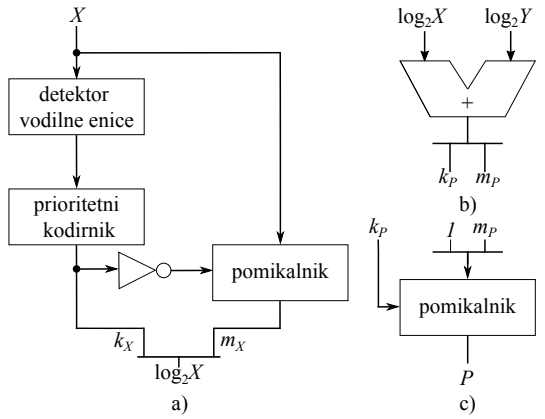
Poznamo logaritemske, nelogaritemske in hibridne približne množilnike, ki se razlikujejo v načinu računanja približnega zmnožka, natančnosti in porabi energije. Pri logaritemskih množilnikih določimo približna logaritma obeh faktorjev, ju seštejemo in antilogaritmiramo. Pri nelogaritemskih množilnikih poenostavljamo bodisi računanje delnih zmnožkov bodisi njihovo seštevanje. Hibridni množilniki združujejo elemente logaritemskih in nelogaritemskih množilnikov.

V nadaljevanju predstavimo nekaj najsodobnejših približnih množilnikov iz vsake skupine. Približne množilnike primerjamo z vidika zakasnitve, porabe prostora in energije, računske napake in obnašanja v aplikacijah. V poglavju 2 predstavimo osnovno idejo ter pregled sodobnih logaritemskih približnih množilnikov. V poglavju 3 najprej predstavimo natančni množilnik, nato pa pregledamo rešitve v sodobnih nelogaritemskih približnih množilnikih. Kratek pregled hibridnih množilnikov je podan v poglavju 4. V poglavju 5 predstavimo rezultate sinteze izbranih približnih množilnikov ter prikažemo obnašanje približnih množilnikov pri glajenju slik in razvrščanju z globokimi nevronskimi mrežami. V sklepu povzamemo glavne ugotovitve, ki lahko služijo kot osnovno vodilo pri izbiri približnega množilnika za izbrano aplikacijo.

2 LOGARITEMSKI PRIBLIŽNI MNOŽILNIKI

2.1 Osnovna zgradba

Mitchell je že leta 1962 predlagal nepredznačeni logaritemski približni množilnik [5].



Slika 1: Zgradba logaritemskih množilnikov: a) logaritemska pretvorba, b) seštevanje logaritmov in c) računanje antilogaritma.

Logaritemski približni množilniki z logaritmiranjem faktorjev množenje nadomestijo s seštevanjem, ki mu sledi antilogaritmiranje. V prvem koraku izračunamo dvojiški logaritem obeh faktorjev. Za izračun logaritma nepredznačenega števila X v bitnem zapisu določimo položaj vodilne enice, ki nam določa eksponent k_X, preostali biti na desni pa predstavljajo mantiso m_X,

$$X = 2^{k_X}(1 + m_X) \quad , \tag{1}$$

V prvem koraku izračunamo približek logaritma po enačbi

$$\log_2 X = k_X + \log_2(1 + m_X) \approx k_X + m_X \quad . \tag{2}$$

Slika 1a prikazuje ustrezno vezje. Z detektorjem vodilne enice in prioritetnim kodirnikom dobimo eksponent k_X. Mantiso faktorja dobimo s pomikanjem vhodnega števila za $k_X - 1$ bitov v levo. Ker je m_X vedno manjši od 1, lahko seštevanje v enačbi (2) nadomestimo s spenjanjem bitov v k_X in m_X.

V drugem koraku seštejemo približna logaritma faktorjev X in Y. Logaritem zmnožka $P = X \cdot Y$ lahko zapišemo kot vsoto

$$\log_2 P = k_P + m_P = k_X + m_X + k_Y + m_Y \quad , \tag{3}$$

Slika 1b prikazuje seštevalnik. Višji biti vsote predstavljajo eksponent zmnožka k_P, medtem ko preostali biti tvorijo njegovo mantiso m_P.

V tretjem koraku z antilogaritmiranjem dobimo približni zmnožek

$$P = (1 + m_P) \cdot 2^{k_P} \quad . \tag{4}$$

Računanje antilogaritma v vezju (slika 1c) izvedemo s pomikanjem vrednosti $(1 + m_P)$ za k_P bitov v levo.

2.2 Pregled

Opisani Mitchellov množilnik [5] predstavlja izhodišče za načrtovanje logaritemskih približnih množilnikov. Mitchellov množilnik ima veliko računsko napako, ki narašča s številom enic v obeh faktorjih. Za zmanjšanje računske napake so Mahalingam in sodelavci [6] uporabili dekompozicijo faktorjev. Babić in sodelavci [7] so predlagali iterativni postopek računanja zmnožka, kjer z večanjem števila iteracij postopno izboljšujemo njegovo natančnost.

Z večanjem popularnosti nevronskih mrež in drugih aplikacij, odpornih proti napaki, so se zahteve po natančnem računanju zmanjšale. S tem se je povečalo zanimanje za poenostavljenje Mitchellovega množilnika. Liu in sodelavci [8] so v množilniku *ALM–SOA* uporabili približni seštevalnik. Podobno so Ansari in sodelavci [9] predlagali logaritemski približni množilnik *ILM–AA* s približnim seštevalnikom, ki doseže manjšo računsko napako. V obeh pristopih uporaba približnega seštevalnika za seštevanje logaritmov vodi do manjšega množilnika in s tem do manjše porabe energije.

Krajšanje mantise faktorjev predstavlja še en pomemben pristop pri načrtovanju logaritemskih približnih množilnikov. Krajša mantisa zmanjšuje kompleksnost vseh stopenj množilnika, ki pa se odraža v večji napaki zmnožka. Kim in sodelavci [10] so predstavili logaritemski množilnik *Mitchell–trunc*, ki odreže najmanj pomembne bite mantise. Yin in sodelavci [11] so predlagali

logaritemski približni množilnik *DR–ALM* z dinamičnim obsegom, ki ohranja le najpomembnejše bite mantise, pri čemer za kompenzacijo negativne napake nastavi zadnji bit okrajšane mantise na 1. Pilipović in sodelavci [12] so predstavili učinkovit logaritemski množilnik *TL*, ki temelji na dvojnem rezanju faktorjev – v prvem koraku skrajša sam faktor, v drugem koraku pa odreže še najnižje bite v mantisi.

3 NELOGARITEMSKI PRIBLIŽNI MNOŽILNIKI

3.1 Natančni Boothov množilnik

Natančni množilnik z Boothovim kodiranjem vključuje računanje delnih produktov ter njihovo seštevanje. Leta 1950 je Booth predlagal algoritem za kodiranje v bazi 4 [13]. Algoritem predpostavlja, da je n-bitno predznačeno število X predstavljeno v dvojiškem komplementu,

$$X = -b_{X,n-1}2^{n-1} + \sum_{i=0}^{n-2} b_{X,i}2^i \quad . \tag{5}$$

Boothovo kodiranje iz trojic bitov (slika 3.1) določi vrednosti $\hat{b}_i^{R4} = -2b_{X,2i+1} + b_{X,2i} + b_{X,2i-1}$ in tako zmanjša število delnih zmnožkov

$$P = X \cdot Y = \sum_{i=0}^{n/2-1} (\hat{b}_i^{R4} \cdot Y)4^i \quad . \tag{6}$$

Ker velja $\hat{b}_i^{R4} \in \{0, \pm 1, \pm 2\}$, lahko delne zmnožke $\hat{b}_i^{R4} \cdot Y$ tvorimo z enostavnimi operacijami, kot sta pomik v levo in negacija.

$$\overbrace{b_{15}b_{14}\underbrace{b_{13}b_{12}}_{\hat{b}_6^{R4}}\overbrace{b_{11}b_{10}}^{\hat{b}_5^{R4}}\underbrace{b_9b_8}_{\hat{b}_4^{R4}}\overbrace{b_7b_6}^{\hat{b}_3^{R4}}\underbrace{b_5b_4}_{\hat{b}_2^{R4}}\overbrace{b_3b_2}^{\hat{b}_1^{R4}}\underbrace{b_1b_0}_{\hat{b}_0^{R4}}0}^{\hat{b}_7^{R4}} \tag{7}$$

Slika 2: Kodiranje vrednosti $\hat{b}_i^{R4}$ pri Boothovem množenju 16-bitnih operandov v bazi 4.

Za hitrejše seštevanje delnih zmnožkov se običajno uporabljajo drevesa seštevalnikov (npr. Wallaceovo drevo). To so kombinacijska vezja, ki omogočajo hkratno seštevanje več delnih zmnožkov. Ta drevesa namesto polnih seštevalnikov, ki seštejejo le dva bita, uporabljajo 4-2 paralelne števnike (angl. *4-2 compressor*) za seštevanje štirih bitov naenkrat.

3.2 Pregled

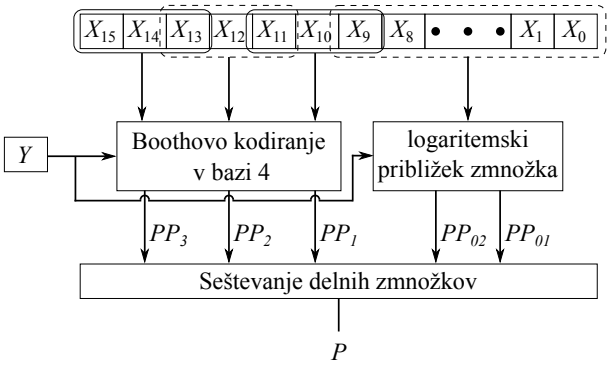
Nelogaritemski približni množilniki poenostavljajo računanje delnih zmnožkov ali pa njihovo seštevanje. Poenostavitev računanja delnih zmnožkov daje večje prihranke prostora in energije kot seštevanje. Zaradi tega se v pregledu osredotočimo na tovrstne pristope.

Jiang in sodelavci [14] so predlagali približni množilnik z Boothovim kodiranjem v bazi 8, ki približno računa delne zmnožke za vrednosti $\hat{b}_i^{R8} \in \pm 3$. Liu in sodelavci [15] so poenostavili vezje za približno računanje Boothovega kodiranja v bazi 4. Da so omejili napako, so približno kodiranje uporabili samo za nekaj spodnjih delnih zmnožkov. V [16] so predlagali približni množilnik *RAD1024* z Boothovim kodiranjem, pri katerem je en faktor razdeljen na višji in nižji del. Bite iz višjega dela operanda kodirajo natančno v bazi 4, medtem ko bite iz nižjega dela kodirajo približno v bazi 1024. Podobno so Waris in sodelavci [17] predlagali približni Boothov množilnik *HLR–BM2*, ki za višje bite uporablja natančno Boothovo kodiranje v bazi 4, za nižje pa približno kodiranje v bazi 8.

4 HIBRIDNI PRIBLIŽNI MNOŽILNIKI

Hibridni množilniki predstavljajo najmlajšo skupino približnih množilnikov. Nastali so v želji po zmanjšanju računske napake logaritemskih približnih množilnikov. Kljub temu, da so logaritemski približni množilniki majhna in energijsko učinkovita vezja, lahko precejšnja računska napaka omeji njihovo uporabnost. Z vključevanjem Boothovega kodiranja v bazi 4 v logaritemske približne množilnike dobimo natančnejša in energijsko nekoliko bolj potratna vezja.

Slika 3 prikazuje prvi hibridni približni množilnik *LOBO* [18]. Množilnik združuje natančno Boothovo kodiranje v bazi 4 in logaritemsko kodiranje. Podobno kot nelogaritemski približni množilniki poskuša množilnik *LOBO* narediti čim manj delnih zmnožkov. Zato za bolj pomembne bite v faktorju X uporablja natančno Boothovo kodiranje v bazi 4, z logaritemskim množilnikom, ki je predstavljen v [7] pa zmnoži manj pomembne bite faktorja X s faktorjem Y. Dodatno je z zmanjševanjem širine podatkovnih poti množilnik *LOBO* malenkost manj natančen, vendar zato energijsko bolj učinkovit.



Slika 3: Zgradba množilnika LOBO.

Hibridni približni množilnik *HRALM* [19] je zasnovan na podobni ideji, vendar ponuja enostavnejši dizajn. Množilnik z Boothovim kodiranjem v bazi 4 ustvari le dva delna zmnožka, zaradi česar lahko kompleksne drevesne sheme nadomesti z enim samim seštevalnikom. Vse preostale delne zmnožke oceni z logaritemskim

približkom, ki vključuje tudi rezanje manj pomembnih bitov v mantisi.

5 REZULTATI

5.1 Sinteza množilnikov in ocena napake

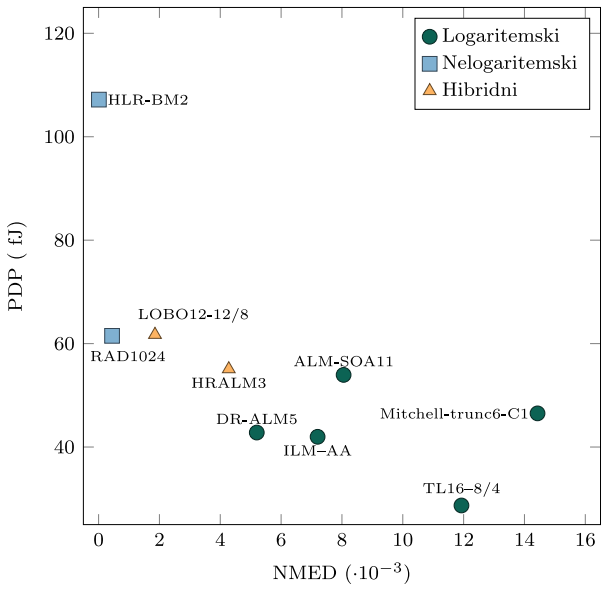
Pri izbiranju najustreznejšega množilnika za izbrano aplikacijo so pomembne strojne značilnosti vezja in napaka. Med prvimi bomo opazovali zakasnitev signala skozi vezje, velikosti vezja, dinamično moč in porabo energije z mero PDP (angl. *Power-Delay-Product*). Za oceno napake bomo uporabili normalizirano povprečno razdaljo NMED (angl. *Normalized Mean Error Distance*), ki jo povprečimo preko vseh možnih faktorjev in normaliziramo z največjim zmnožkom natančnega množilnika.

V primerjavo smo vključili približne množilnike, predstavljene v predhodnih poglavjih: logaritemske ALM–SOA11 [8], ILM–AA [9], Mitchell–trunc6 [10], DR–ALM5 [11] in TL16–8/4 [12], nelogaritemske RAD1024 [16] in HLR–BM2 [17] ter hibridne LOBO12–12/8 [18] in HRALM3 [19]. Vse približne množilnike smo opisali v jeziku Verilog in jih po potrebi prilagodili za računanje s predznačenimi števili. Za sintezo smo uporabili odprtokodno orodje OpenROAD [20] s knjižnico Nangate 45 nm CMOS. Za oceno napake z mero NMED smo množilnike simulirali v programskem jeziku C.

Lastnosti množilnikov so zbrane v tabeli 1 in ponazorjene na sliki 4. Logaritemski približni množilniki se odlikujejo v fizikalnih lastnostih: so manjša vezja, s krajšimi zakasnitvami in manjšo porabo energije, njihova računska napaka pa je večja kot pri preostalih približnih množilnikih. Kot taki so odlični za aplikacije, kjer je nizka poraba pomembnejša od natančnosti računanja. Nelogaritemska množilnika se odlikujeta z nizkim NMED in zato z bistveno večjo natančnostjo od logaritemskih. Večja natančnost pa se odraža v slabših fizikalnih lastnostih in manjših prihrankih v primerjavi z natančnim množilnikom – vezja so večja, zato tudi zakasnitve in poraba energije. Zaradi majhne napake so nelogaritemski množilniki izvrstni kandidati za aplikacije, ki dovoljujejo približno računanje, vendar prevelika nenatančnost pri računanju pomembno vpliva na kakovost rezultata. Hibridna množilnika se po fizikalnih lastnostih in napaki uvrščata med obe skupini in sta primerna za aplikacije, ki zahtevajo energijsko varčna vezja, vendar ne prenesejo precejšnje računske napake logaritemskih približnih množilnikov.

Tabela 1: Primerjava 16 bitnih množilnikov. Znak * označuje množilnike, prilagojene za računanje s predznačenimi števili.

Množilnik	**Zakasnitev** [ns]	**Moč** [μW]	**Velikost** [μm^2]	**PDP** [fJ]	**NMED** [10^{-3}]
Natančni	1,74	69,20	1.576,58	120,41	0
Logaritemski					
ALM–SOA11* [8]	1,47	36,70	952,01	53,95	8,06
ILM–AA* [9]	1,51	27,80	780,18	41,98	7,20
Mitchell–trunc6 [10]	1,44	32,30	840,83	46,51	14,43
DR–ALM5 [11]	1,31	32,70	831,78	42,80	5,27
TL16–8/4 [12]	1,13	25,40	702,24	28,70	11,84
Nelogaritemski					
RAD1024 [16]	1,50	41,00	1.008,67	61,50	0,44
HLR–BM2 [17]	1,76	60,90	1.312,18	107,18	0,01
Hibridni					
LOBO12–12/8 [18]	1,71	36,10	904,93	61,73	1,85
HRALM3 [19]	1,70	32,40	842,42	55,08	4,28



Slika 4: Povezava med napako NMED in porabo energije PDP za izbrane približne množilnike.

5.2 Ocena uporabnosti množilnikov v aplikacijah

V predhodnih raziskavah [18], [19], [12] smo ocenili uporabnost sodobnih približnih množilnikov v različnih aplikacijah, na primer pri obdelavi slik in razvrščanju z globokimi nevronskimi mrežami. V tem prispevku povzemamo glavne rezultate in razpravljamo o uporabnosti izbranih približnih množilnikov.

5.2.1 Glajenje slik: Glajenje je eden od osnovnih postopkov pri obdelavi slik. Gre za postopek, pri katerem z rahlo prerazporeditvijo barv v sliki zgladimo ostre prehode in fine detajle [21]. Jedro algoritma predstavlja konvolucija vhodne slike s konvolucijsko masko velikosti 3×3. Slike, zglajene z različnimi približnimi množilniki, primerjamo s sliko, zglajeno z natančnim množilnikom. Razliko med slikama ovrednotimo s standardnima merama MSSIM (angl. *Mean Structural Simmilarity Index*) in PSNR (angl. *Peak Signal to Noise Ratio*).

Tabela 2 prikazuje rezultate glajenja na izbranih slikah iz slikovne baze TESTIMAGES [22]: *building, cards, flowers, snails in wood game*. Množilniki z izjemo DR–ALM5 se odlikujejo z visoko vrednostjo mere MSSIM, ki ocenjuje subjektivno kakovost slike. Do večjih

Tabela 2: Meri MSSIM in PSNR za glajenje slik.

Približni množilnik	**building**		**cards**		**flowers**		**snails**		**wood game**	
	MSSIM	PSNR[dB]	MSSIM	PSNR[dB]	MSSIM	PSNR[dB]	MSSIM	PSNR[dB]	MSSIM	PSNR[dB]
Logaritemski										
ALM–SOA11 [8]	0,99	39,56	0,99	39,17	0,99	39,78	0,99	39,78	0,99	39,45
ILM–AA [9]	0,99	39,75	1,00	39,24	0,99	40,08	0,99	40,08	1,00	40,04
Mitchell–trunc6 [10]	0,99	36,63	1,00	35,30	0,99	36,90	0,99	36,90	0,99	36,32
DR–ALM5 [11]	0,94	36,25	0,97	34,31	0,94	33,75	0,94	32,45	0,96	34,15
TL16–8/4 [12]	0,98	37,44	0,98	36,55	0,99	38,02	0,98	37,52	0,98	36,35
Nelogaritemski										
RAD1024 [16]	1,00	40,70	1,00	40,65	0,99	40,71	0,99	40,71	1,00	40,63
HLR–BM2 [17]	1,00	54,62	1,00	53,06	1,00	56,38	1,00	54,54	1,00	51,19
Hibridni										
LOBO12–12/8 [18]	0,99	40,69	0,99	40,42	0,99	40,99	0,99	40,67	0,99	40,38
HRALM3 [19]	1,00	41,15	1,00	41,27	1,00	41,41	1,00	41,34	1,00	41,28

a) Natančni b) DR-ALM5 c) HLR-BM2 d) HRALM3

Slika 5: Rezultati glajenja slike *building* z izbranimi množilniki. Na sliki b), ki smo jo zgladili z logaritemskim približnim množilnikom DR–ALM5, opazimo posterizacijo v okolici stropne luči.

razlik prihaja pri meri PSNR. Logaritemski množilniki zaradi velike napake NMED dosegajo nizko razmerje med signalom in šumom. Nasprotno pa nelogaritemska množilnika z majhno napako NMED dosegata zelo visok PSNR. Oba hibridna množilnika dajeta podobne rezultate kot nelogaritemski približni množilnik RAD1024, predvsem množilnik HRALM3 s precej manjšim in energijsko učinkovitejšim vezjem. Na sliki 5 je pri glajenju z logaritemskim množilnikom DR–ALM5 opazna posterizacija, do katere pa ne pride pri glajenju z nelogaritemskim približnim množilnikom HLR–BM2 in hibridnim približnim množilnikom HRALM3.

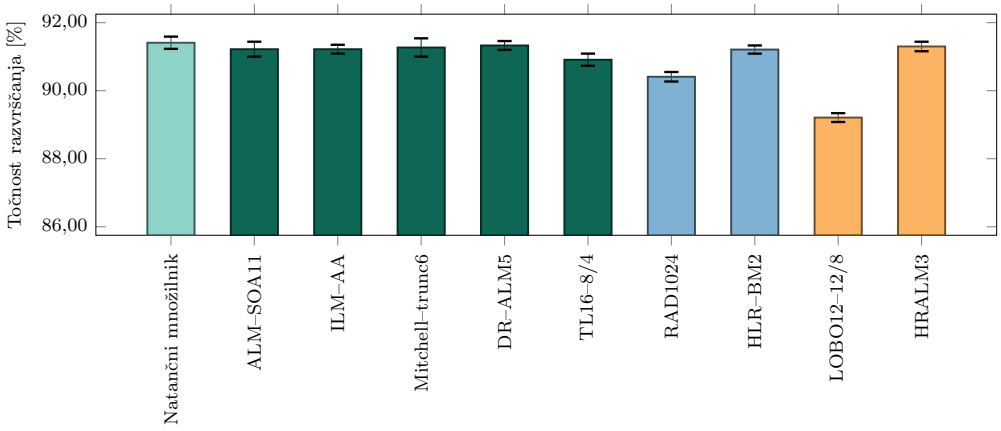
5.2.2 Razvrščanje s konvolucijskimi nevronskimi mrežami: Konvolucijske nevronske mreže so globoke nevronske mreže, ki so primerne za razvrščanje slik [23]. Nevronske mreže so računsko zahtevni modeli, ki so sposobni tolerirati napake v vhodnih podatkih in se prilagoditi tudi napaki pri računanju. Kovolucijske nevronske mreže vključujejo množico množenj in so zato zelo primerne za testiranje približnih množilnikov.

V poskusih smo uporabili globoko nevronsko mrežo ResNet-20 [24] in podatkovno bazo CIFAR10 [25]. Delali smo s knjižnico Caffe [26], v kateri smo med izvajanjem modela natančna množenja nadomestili s približnimi. Za vsak približni množilnik smo nevronsko mrežo učili desetkrat, vsakič z naključno začetno nastavitvijo uteži in z vnaprej določeno učno in testno množico [12]. Pri izvajanju modela smo uporabljali približno množenje, pri učenju pa natančno množenje v fiksni vejici.

Slika 6 predstavlja povprečno točnost razvrščanja slik iz podatkovne zbirke CIFAR10 s konvolucijsko nevronsko mrežo ResNet-20. Večina približnih množilnikov, ne glede na napako NMED, pri razvrščanju slik dosega točnost, ki je primerljiva s točnostjo natančnega množilnika. Slabši rezultat pri nekaterih množilnikih lahko pripišemo njihovi zasnovi, ki jim zaradi velike napake pri množenju majhnih števil ne uspe zaznati manjših razlik v vhodnih podatkih. Nevronski mreži z njeno robustno arhitekturo uspe kompenzirati večjo napako logaritemskih množilnikov. Ti so zaradi majhnosti in energijske učinkovitosti odlična izbira pri modeliranju z nevronskimi mrežami.

6 SKLEP

V prispevku smo predstavili najsodobnejše približne množilnike. Logaritemski približni množilniki so majhni in energijsko učinkoviti, vendar pri množenju lahko naredijo precejšnjo računsko napako. Nelogaritemski približni množilniki so veliko večji in energijsko bolj potratni, vendar naredijo veliko manjšo napako pri množenju. Načrtovanje hibridnih približnih množilnikov sloni na idejah nelogaritemskih in logaritemskih množilnikov, zato tudi po fizikalnih lastnostih in računski napaki zasedajo prostor med nelogaritemskimi in logaritemskimi



Slika 6: Točnost razvrščanja slik iz podatkovne množice CIFAR10 z nevronsko mrežo ResNet-20 in približnimi množilniki.

množilniki.

Pokazali smo, da so logaritemski približni množilniki odlična izbira za modeliranje z nevronskimi mrežami, ki uspešno kompenzirajo njihovo večjo računsko napako. Nelogaritemski in hibridni približni množilniki ponujajo boljše rezultate od logaritemskih približnih množilnikov pri obdelavi slik, kjer so sprejemljive le manjše napake pri računanju.

Približni množilniki so manjši, hitrejši in energijsko učinkovitejši od natančnih množilnikov. Zaradi energijske učinkovitosti so po eni strani odlična izbira za aplikacije, ki zahtevajo dolgo avtonomijo, po drugi strani pa njihovo hitrost in majhnost lahko izkoristimo za povečanje računskih kapacitet visokozmogljivih računalniških sistemov. Pri izbiranju najprimernejšega približnega množilnika je treba upoštevati zahteve aplikacije. Običajno želimo najmanjši, najhitrejši ali energijsko najučinkovitejši množilnik, ki ima za pravilno delovanje aplikacije še sprejemljivo računsko napako.

Približni množilniki so velik potencial za energijsko učinkovito računanje, ki je vedno bolj aktualno na najrazličnejših področjih. Glede na trenutni trend porasta aplikacij, primernih za približne množilnike, verjamemo, da bodo približni množilniki postali nenadomestljivi elementi modernih računalniških sistemov.

ZAHVALA

Raziskavo je omogočilo Ministrstvo za visoko šolstvo, znanost in tehnologijo Republike Slovenije v okviru programa P2-0359 - Vseprisotno računalništvo in P2-0241 Sinergetika kompleksnih sistemov in procesov. Hvala Tijani Milaković za vloženi trud pri lektoriranju prve različice prispevka.

LITERATURA

[1] W. Liu, F. Lombardi, and M. Schulte, "Approximate computing: From circuits to applications [scanning the issue]," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2103–2107, 2020.

[2] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM SIGGRAPH 2007 Papers*, ser. SIGGRAPH '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 10–es. [Online]. Available: https://doi.org/10.1145/1275808.1276390

[3] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[4] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[5] J. N. Mitchell, "Computer multiplication and division using binary logarithms," *IRE Transactions on Electronic Computers*, vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[6] V. Mahalingam and N. Ranganathan, "Improving accuracy in mitchell's logarithmic multiplication using operand decomposition," *IEEE Transactions on Computers*, vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[7] Z. Babić, A. Avramović, and P. Bulić, "An iterative logarithmic multiplier," *Microprocessors and Microsystems*, vol. 35, no. 1, pp. 23 – 33, Jul. 2011.

[8] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 9, pp. 2856–2868, Sep. 2018.

[9] M. S. Ansari, B. F. Cockburn, and J. Han, "An improved logarithmic multiplier for energy-efficient neural computing," *IEEE Transactions on Computers*, pp. 1–1, 2020.

[10] M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient mitchell's approximate log multipliers for convolutional neural networks," *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 660–675, Dec. 2019.

[11] P. Yin, C. Wang, H. Waris, W. Liu, Y. Han, and F. Lombardi, "Design and analysis of energy-efficient dynamic range approximate logarithmic multipliers for machine learning," *IEEE Transactions on Sustainable Computing*, vol. Early access, pp. 1–1, Jun. 2020.

[12] R. Pilipović, P. Bulić, and U. Lotrič, "A two-stage operand trimming approximate logarithmic multiplier," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–11, 2021.

[13] A. D. Booth, "A signed binary multiplication technique," *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 4, no. 2, pp. 236–240, 1951.

[14] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance

operation," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, Aug. 2016.
[15] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017.
[16] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 421–430, Nov. 2018.
[17] H. Waris, C. Wang, and W. Liu, "Hybrid low radix encoding-based approximate booth multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3367–3371, Feb. 2020.
[18] R. Pilipović and P. Bulić, "On the design of logarithmic multiplier using radix-4 booth encoding," *IEEE Access*, vol. 8, pp. 64 578–64 590, Apr. 2020.
[19] U. Lotrič, R. Pilipović, and P. Bulić, "A hybrid radix-4 and approximate logarithmic multiplier for energy efficient image processing," *Electronics*, vol. 10, no. 10, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/10/1175
[20] S. Reda, "Overview of the openroad digital design flow from rtl to gds," in *2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2020, pp. 1–1.
[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. USA: Prentice-Hall, Inc., 2006.
[22] N. Asuni and A. Giachetti, "TESTIMAGES: A large data archive for display and algorithm testing," *Journal of Graphics Tools*, vol. 17, no. 4, pp. 113–125, Feb. 2013. [Online]. Available: https://doi.org/10.1080/2165347X.2015.1024298
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Lake Tahoe, NV, USA: Curran Associates, Inc., Dec. 2012, pp. 1097–1105.
[24] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 1398–1406.
[25] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Tech. Rep., Apr. 2009.
[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 675–678. [Online]. Available: https://doi.org/10.1145/2647868.2654889

Ratko Pilipović je leta 2021 doktoriral s področja računalništva in informatike na Univerzi v Ljubljani. Je asistent na Fakulteti za računalništvo in informatiko. Njegovo področje raziskovanja obsega načrtovanje aritmetičnih vezij za približno računanje, strojno učenje in vgrajene sisteme.

Patricio Bulić je diplomiral na Fakulteti za elektrotehniko Univerze v Ljubljani ter magistriral in doktoriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer poučuje več predmetov s področja računalniških sistemov.

Uroš Lotrič je izredni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno dela na področju informacijske teorije in visokozmogljivega računanja. Omenjeni področji pokriva tudi v pedagoškem procesu.