



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

Neuroniniai tinklai mikrokontroleriuose
Neural Network for Microcontrollers

Baigiamasis bakalauro darbas

Atliko: Ernestas Velevko

VU el. p.: ernestas.velevko@mif.stud.vu.lt

Vadovas: Prof. Dr. Aistis Raudys

Vilnius

2022

Turinys

Įvadas.....	4
1. Mikrokontrolieriai	6
1.1. ESP8266 NodeMcu V3.....	7
1.2. Arduino DUE ATSAM3X8E.....	8
1.3. ESP32 WROOM 32.....	9
1.4. STM32F103C Maple Mini	10
1.5. Teensy 4.0.....	11
1.6. Micro:Bit.....	12
1.7. STM32F407VE.....	13
1.8. Raspberry Pi 4.....	14
1.9. Techninių savybių lentelė	15
2. Neuroninis tinklas.....	16
2.1. Aktyvacijos funkcijos	16
2.2. Aktyvacijos funkcijų tipai.....	17
2.2.1. Binarinė žingsnio funkcija	17
2.2.2. Linijinė funkcija.....	17
2.2.3. Sigmoido funkcija.....	18
2.2.4. Hiperbolinė tangento funkcija.....	19
2.2.5. Parametrizuota ReLU funkcija	19
2.2.6. Eksponentinė tiesinio vieneto funkcija	20
2.2.7. Švytuojanti funkcija.....	20
2.2.8. SoftMax funkcija	20
2.3. Tinklo dydžio parinkimas	21
2.3.1. Įvesties sluoksnis	21
2.3.2. Vidiniai sluoksniai	21

2.3.3. Išvesties sluoksnis.....	22
2.3.4. Išvados	22
3. Tyrimai	23
3.1. Aktyvacijos funkcijų tyrimas.....	23
3.2. Greitaveikos tyrimas	24
3.3. Tinklo dydžio tyrimas	31
Išvados	37
Summary.....	40
Literatūra	41

Ivadas

Šiomis dienomis dauguma produktų ir technologijų yra robotizuojamos. Vienas iš populiariausių algoritmų padedančių jiems tobulėti - neuroniniai tinklai. Neuroninių tinklų pavadinimas ir struktūra buvo įkvėpti žmogaus smegenų. Jie veikia kurdami ryšius tarp daug skirtingų apdorojimo elementų, kurie imituoja biologinių smegenų neuronus. Neuroninių tinklų algoritmai asocijuojami su greitą CPU ir GPU turinčiais dideliais kompiuteriais, dideliu RAM dydžiu arba algoritmų vykdymu debesyje, tačiau jie gali būti naudojami ir daug mažesnio pajėgumo įrenginiuose - mikrokontrolieriuose.

Daugelis mikrokontrolierių jų yra sukurti mobiliajai, nešiojamai elektronikai, daiktų interneto (IoT) programoms. Šio prietaiso idėja savo luste talpinti kuo daugiau elektroninių grandinių (tokias kaip „Wi-Fi“, „Bluetooth“, etc.), kad jis būtų kuo universalesnis ir jam reikėtų kuo mažiau papildomų elementų. Mikrokontrolieriai, kaip ir mums įprasti kompiuteriai, turi centrinį procesorių (CPU), kuris yra atsakingas už programos vykdymą ar jos užkrovimą iš „flash“ (laikinosios) atminties bei atsitiktinės prieigos atmintį (RAM), kurioje yra saugoma vykdymo metu naudojama informacija. Šiuolaikiniai mikrokontrolieriai pasižymi žemu energijos suvartojimu, nedidele kaina ir lankstumu, todėl yra puikiai tinkami neuroniniams tinklams. Neuroniniai tinklai mikrokontrolieriuose gali padėti sukurti naujus IoT prietaisus ar pagerinti senų veikimą gerinant jų atliekamų veiksmų tikslumą.

Šio darbo tikslas yra ištirti neuroninius tinklus skirtinguose mikrokontrolieriuose bei aktyvacijos funkcijų, tinklo architektūros ypatybių įtaką neuroninių tinklų algoritmo greitaveikai ir naudojamai atminčiai juose. Darbe tiriama, kokio dydžio tiesioginio skleidimo daugiasluoksnis (angl. „Feed-forward Multi-layer perception“) neuroninis tinklas telpa pasirinktame mikrokontrolieryje ir kokia yra greičio priklausomybė nuo tinklo dydžio, tai yra, kiek kartų kreipiamasi į tinklą per tam tikrą laiko vienetą. Be to, tiriama tinklo greitaveikos priklausomybė nuo aktyvacijos funkcijos parinkimo ir tinklo dydžio parinkimas. Pagal greitaveiką ir talpą galima nuspręsti, koks mikrokontrolieris būtų tinkamiausias pasirinktai problemai spręsti.

Šiam tikslui pasiekti reikės:

- susipažinti su skirtingų tipų mikrokontrolieriais, jų savybėmis,
- sukurti neuroninį tinklą,
- paleisti neuroninį tinklą skirtinguose mikrokontrolieriuose,
- susipažinti su skirtingų tipų aktyvacijos funkcijomis ir atlikti greitaveikos tyrimą,

- išnagrinėti tinklo dydžio parinkimo metodikas ir pateikti rekomendacijas,
- atlikti greitaveikos tyrimus su Double, Float, Float16 ir Integer duomenų tipais,
- atlikti tinklo dydžio tyrimus, ištirti Double, Float, Float16 ir Integer duomenų tipų įtaką tyrimo rezultatams,
- apibendrinti tyrimo metu gautas išvadas.

Visų pirma, darbe apžvelgiami mikrokontrolerių panašumai, aprašomas kiekvienas mikrokontroleris ir jų techninės ypatybės. Tyrimui sukuriamas neuroninis tinklas, kuris paleidžiamas mikrokontroleriuose. Vėliau darbe pateikiami greitaveikos, tinklo dydžio, slankiojo kabelio duomenų tipų, aktyvacijos funkcijų tyrimų rezultatai ir išvados.

1. Mikrokontroleriai

Atliekant šį tyrimą su skirtingais mikrokontroleriais svarbu atsižvelgti į skirtingus kiekvieno mikrokontrolerio techninius duomenis. Šiam tyrimui atlikti pasirinkti mikrokontroleriai: ESP8266 NodeMcu V3, Arduino DUE, ESP32 WROOM 32, STM32F103C, Teensy 4.0, Micro:Bit, STM32F407VE, Raspberry Pi 4.

Atliekant tyrimą svarbiausi mikrokontrolerių aspektai [Ker12]:

- „ROM - Read Only Memory“ - atmintis, kurioje surašyta informacija nekinta, tokia kaip protokolai, instrukcijos, programinė įranga (angl. „firmware“). Kai kurie mikrokontroleriai turi „EEPROM“ - „Electrically Erasable Programmable Read-Only Memory“, tai leidžia perprogramuoti mikrokontrolerio programinę įrangą;
- „RAM - Random-Access Memory“ - atmintis, kurioje vykdymo metu (angl. „runtime“) saugomas kodas ir su įdiegta užduotimi susijusi informacija;
- Procesorius matuojamas Hz (hercais), kitaip, taktiniais dažniais. Šis skaičius nusako, kiek ciklų atlieka procesorius per 1 sekundę;
- Slankaus kablelio skaičių palaikymas;
- Energijos naudojimas;
- Kaina;
- Instrukcijų rinkinys - [KGF14] nusako kaip bus bendraujama su mikroprocesoriumi: nustato registrus, pertraukimus, adresavimą, duomenų tipus. Be to, instrukcijų rinkinyje apibrėžiamos skaitymo, rašymo, duomenų manipuliavimo, aritmetinės bei visos kitos funkcijos. Visi pasirinkti mikrokontroleriai turi „RISC“ instrukcijų rinkinį;
- Architektūra - visi pasirinkti mikrokontroleriai yra „Harvard“ architektūros. [GCK+17] Jos esminis bruožas yra, kad atmintis išskiriama į kodo segmento ir duomenų segmento blokus, kurie sujungti skirtingomis (BUS) magistralėmis, todėl abu segmentai gali būti pasiekti vienu metu.

Aplinka, kurioje leidžiamas tyrimui naudojamas kodas - ArduinoIDE. Visi pasirinkti mikrokontroleriai palaiko šią aplinką.

1.1. ESP8266 NodeMcu V3

[KS16] ESP8266 yra vienas iš labiausiai naudojamų integruotų Wi-Fi lustų. Jis savyje turi anteną, galios ir mažo triukšmo stiprintuvus. Integruotas Wi-Fi tinklo sprendimas yra išsamus ir savarankiškas. Šis mikrokontrolieris turi integruotą patobulintą Tensilica L106 Diamond serijos 32 bitų procesoriaus versiją su lustine RAM. Taip pat, siekiant taupyti energiją, procesorius pasižymi ypatybe greitai persijungti tarp miego ir aktyvaus režimų, kad būtų taupoma energija. Atminties valdiklyje yra ROM ir RAM. Jį pasiekia CPU naudodamas iBus, dBus ir AHB sąsajas. Bet kuri iš šių sąsajų gali prašyti prieigos prie ROM arba RAM modulių, o atminties valdiklio arbitrai aptarnauja šias 3 sąsajas eilės tvarka.



pav. 1 ESP8266 NodeMcu V3

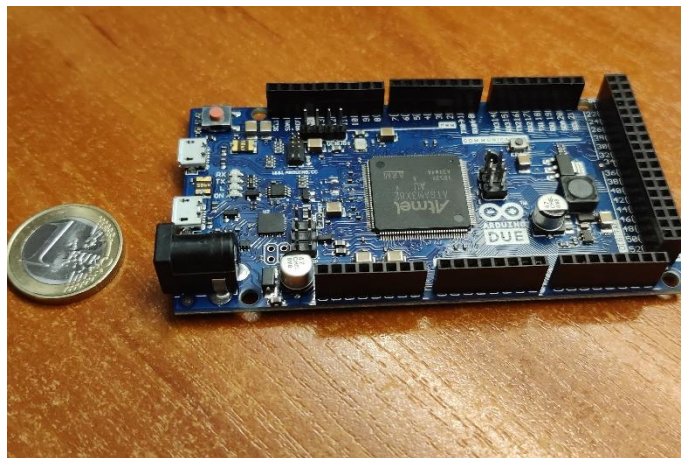
[DTH10] Šis mikrokontrolieris turi:

- Atmintis: 80 KB RAM, 64 KB ROM.
- Periferiniai įrenginiai: 2.4 GHz Wi-Fi mikroschema.
- Procesorius: ESP8266 yra integruotas su Tensilica L106 32 bitų mikrovaldikliu (MCU), kuris pasižymi ypač mažu energijos suvartojimu ir 16 bitų RSIC. CPU taktinis dažnis yra 80 MHz.
- Sąsaja: CPU sąsają (angl. „interface“) sudaro programuojama RAM/ROM sąsaja (iBus), kuri gali būti prijungta su atminties kontrolieriu ir taip pat naudojama kaip „flash“ atmintis, duomenų RAM sąsaja (dBus), kuri gali būti sujungta su atminties kontrolieriu ir AHB sąsaja bei būti naudojama registrams.
- Slankaus kablelio skaičių palaikymas: neturi slankaus kablelio skaičių palaikymo.
- Prietaisas veikia nuo: 2.5V iki 3.6V įtampos maitinimo šaltinio.
- Energijos naudojimas: max 70mA.
- Išmatavimai: 58 x 31 x 5 mm.

- Kaina: apie 13 eurų.

1.2. Arduino DUE ATSAM3X8E

[DS18] Arduino DUE ATSAM3X8E yra SMART SAM3X/A serijos mikrovaldiklis, turintis didelio našumo 32 bitų ARM Cortex-M3 RISC procesorių. Itin integruotas išorinių įrenginių rinkinys, skirtas jungtims ir ryšiams, apima eternetą, dvigubą CAN, didelės spartos USB MiniHost ir įrenginį su lustu PHY, didelės spartos SD/SDIO/MMC ir kelis USART, SPI, TWI (I2C). [KKB+17] Šiame mikrovaldiklyje taip pat yra I2S ir PWM laikmatis, trys trijų kanalų (angl. „pipeline“) bendrosios paskirties 32 bitų laikmačiai, mažos galios RTC, mažos galios RTT, 256 bitų bendrosios paskirties atsarginių kopijų registrai, 12 bitų ADC ir 12 bitų DAC. Be to, šis mikrokontroleris yra ypač gausus įvesties/išvesties kontaktais (12 analoginių ir 53 skaitmeniniai kontaktai).



pav. 2 Arduino DUE ATSAM3X8E

SAM3X/A įrenginiai turi tris programine įranga pasirenkamus mažos galios režimus: miego, laukimo ir atsarginės kopijos. Miego režimu procesorius sustabdomas, o visos kitos funkcijos gali veikti. Laukimo režimu visi laikrodžiai ir funkcijos sustabdomi. Vis dėlto, kai kuriuos išorinius įrenginius galima sukonfigūruoti taip, kad sistema pažadintų pagal iš anksto nustatytas sąlygas.

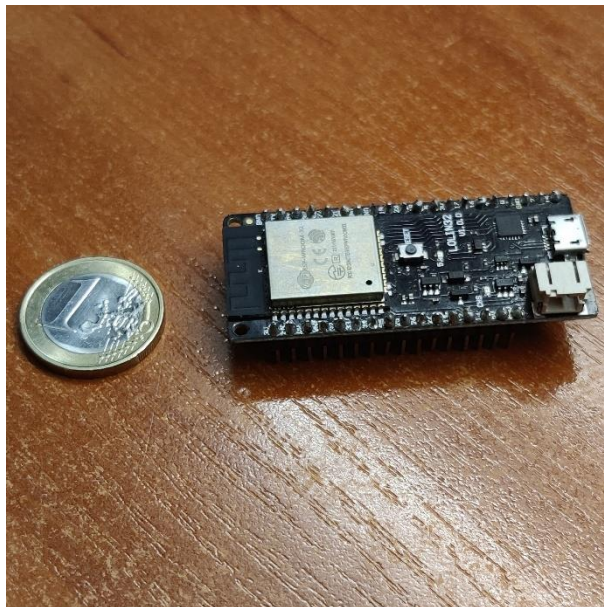
[DTH20] Šis mikrokontroleris turi:

- Atmintis: 2 x 256 ROM susidaro iš 128 baitų rašymo buferio, pasiekiamo per 32 bitų sąsają. Taip pat, atmintį sudaro 64 + 32 KB RAM.
- Procesorius: procesorius veikia maksimaliu 84 MHz greičiu.
- Slankaus kablelio skaičių palaikymas: neturi slankaus kablelio skaičių palaikymo.
- Prietaisas veikia nuo: 1.62V iki 3.6V įtampos maitinimo šaltinio.
- Energijos naudojimas: 150mA.
- Išmatavimai: 112 x 54 x 11 mm.

- Kaina: apie 45 eurus.

1.3. ESP32 WROOM 32

[MAS+17] ESP32 yra galingas mikrokontroleris, turintis du mažai energijos naudojančius Xtensa 32-bit LX6 mikroprocesorius, savyje integruotus Wi-Fi ir Bluetooth modulius. Mikroprocesoriaus mikroschema sukurta mažai energijos suvartojančia 40 nm technologija. Taip pat, šis mikrokontroleris turi mažo veikimo ciklą (angl. „low-duty cycle“), kuris yra vykdomas įvykus atitinkamai sąlygai, taip siekiant sumažinti mikroschemos sunaudojamos energijos kiekį. Visa vidinė, išorinė atmintis ir išoriniai įrenginiai yra šių procesorių duomenų ir (arba) komandų magistralės mikroprocesoriuje.



pav. 3 ESP32 WROOM 32

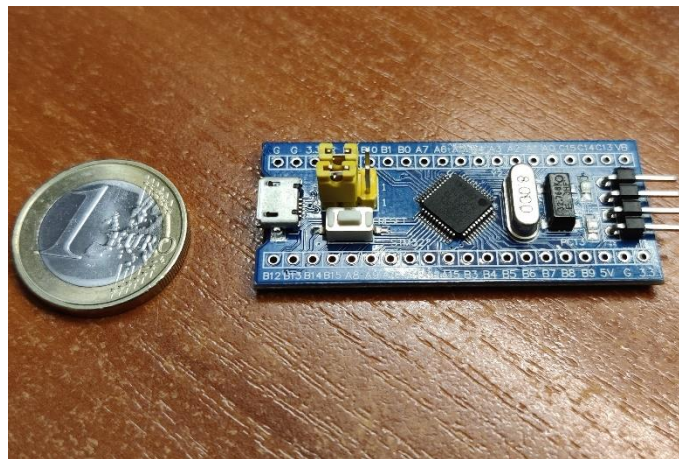
[DTH30] Pagrindinės šio mikrokontrolerio charakteristikos:

- Atmintis: 448 KB ROM, 520 KB RAM.
- Periferiniai įrenginiai: 2.4 GHz Wi-Fi ir Bluetooth mikroschemos.
- Procesorius: Xtensa 32 bitų LX6 mikroprocesoriai su 7 etapų kanalu (angl. „pipeline“) palaiko procesoriaus taktinį dažnį iki 240 MHz. 16/24 bitų instrukcijų rinkinys. Šis mikrokontroleris palaiko 32 pertraukimų (angl. „interrupt“) vektorius iš apie 70 pertraukimų šaltinių.
- Sąsaja: dviejų branduolių CPU sąsają (angl. „interfaces“) sudaro Xtensa RAM/ROM sąsaja (skirta instrukcijoms ir duomenims), Xtensa vietinės atminties sąsaja (skirta greitai įrenginių registrų prieigai), išoriniai ir vidiniai pertraukimų šaltiniai, JTAG derinimas (angl. „debugging“).
- Slankaus kabelio skaičių palaikymas: 32 bitų Float ir 64 bitų Double.
- Prietaisas veikia nuo: 2.2V iki 3.6V įtampos maitinimo šaltinio.
- Energijos naudojimas: max 170 mA.

- Išmatavimai: 58 x 25 x 5 mm.
- Kaina: apie 20 eurų.

1.4. STM32F103C Maple Mini

[AMN+21] STM32F103 vidutinio tankio našumo linijų šeima apima didelio našumo ARM Cortex-M3 32 bitų RISC branduolį. Visuose įrenginiuose yra du 12 bitų ADC, trys bendrosios paskirties 16 bitų laikmačiai ir vienas PWM laikmatis, taip pat standartinės ir pažangios ryšio sąajos: iki dviejų I2C ir SPI, trys USART, USB ir CAN.



pav. 4 STM32F103C Maple Mini

Šis mikrokontroleris buvo sukurtas siekiant užtikrinti nebrangią platformą, atitinkančią MCU diegimo poreikius, su mažesniu kontaktų skaičiumi ir mažu energijos suvartojimu, tuo pačiu užtikrinant puikų skaičiavimo našumą ir pažangų sistemos atsaką į pertraukimus.

[DTH40] ARM Cortex-M3 32 bitų RISC procesorius pasižymi išskirtiniu kodo efektyvumu, užtikrinančiu aukštą našumą, kurio tikimasi iš ARM branduolio, kai atminties dydis paprastai siejamas su 8 ir 16 bitų įrenginiais.

[DTH40] Šis mikrokontroleris turi:

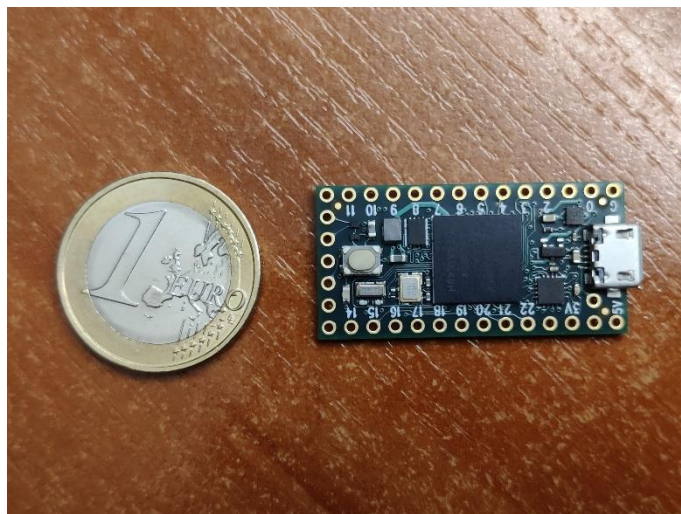
- Atmintis: 20 KB RAM ir 128 KB ROM.
- Procesorius: ARM Cortex-M3 turi 32 bitų RISC branduolį, veikiantį 72 MHz dažniu, didelės spartos įterptąsias atmintis ir daug patobulintų I/O ir periferinių įrenginių, prijungtų prie dviejų APB magistralių.
- Sąsaja: „Flash“ sudaro 128 bitų rašymo buferis, pasiekiamas per 32 bitų sąsają.
- Slankaus kablelio skaičiaus palaikymas: neturi slankaus kablelio skaičių palaikymo.
- Prietaisas veikia nuo: 2.0V iki 3.6 V įtampos maitinimo šaltinio.

- Energijos naudojimas: max 100mA.
- Kaina: apie 5 eurus.
- Išmatavimai: 53 x 22 x 5 mm.

1.5. Teensy 4.0

[LMC21] Teensy 4.0 turi ARM Cortex-M7 procesorių su NXP iMXRT1062 lustu, greičiausiu šiandien prieinamu mikrovaldikliu. ARM Cortex-M7 suteikia daug galingų procesoriaus funkcijų tikroje realaus laiko mikrovaldiklio platformoje. Pradiniai etalonai parodė, kad ArduinoIDE sugeneruotas C++ kodas paprastai pasiekia dvi instrukcijas maždaug 40–50% laiko, atliekant skaitiniu požiūriu intensyvią darbą naudojant sveikuosius skaičius ir rodykles. „Cortex-M7“ yra pirmasis ARM mikrovaldiklis, kuriame naudojamas

šakos numatymas. Naudojant M7, kelis kartus paleidus kilpą, atšakos numatymas pašalina šias papildomas išlaidas, todėl šakos instrukcija gali veikti tik vieno laikrodžio ciklo metu.



pav. 5 Teensy 4.0

[DTH50] Šis mikrokontroleris turi:

- Atmintis: 128 KB ROM, 1 MB RAM.
- Procesorius: ARM Cortex-M7 yra vieno branduolio dvigubas superskalerinis procesorius, o tai reiškia, kad jis gali vykdyti dvi komandas per laikrodžio ciklą 600 MHz dažniu. Žinoma, dviejų veiksmų vykdymas vienu metu priklauso nuo kompiliatoriaus sugeneruotų instrukcijų ir registrų.
- Slankaus kabelio skaičiaus palaikymas: neturi slankaus kabelio skaičių palaikymo.
- Energijos naudojimas: max 100mA.
- Prietaisas veikia nuo: 3.3 V įtampos maitinimo šaltinio.

- Išmatavimai: 36 x 18 x 4 mm.
- Kaina: apie 40 eurų.

1.6. Micro:Bit

[ABB+20] „Micro:bit“ yra vienos plokštės kompiuteris (SBC), kuriame yra programų procesorius su įvairiais lustiniais išoriniais įrenginiais. Šis mikrokontroleris yra labiausiai orientuotas į moksleivių supažindinimą ir robotikos mokymą. Todėl jame yra integruota nemažai periferinių įrenginių, tokių kaip, LED matrica, 2,4 GHz radijo periferinis įrenginys, naudojamas „Bluetooth“, kompasas, akcelerometras. Tad mikrokontroleriu galima atlikti nemažai funkcijų neprijungiant jokio išorinio įrenginio.



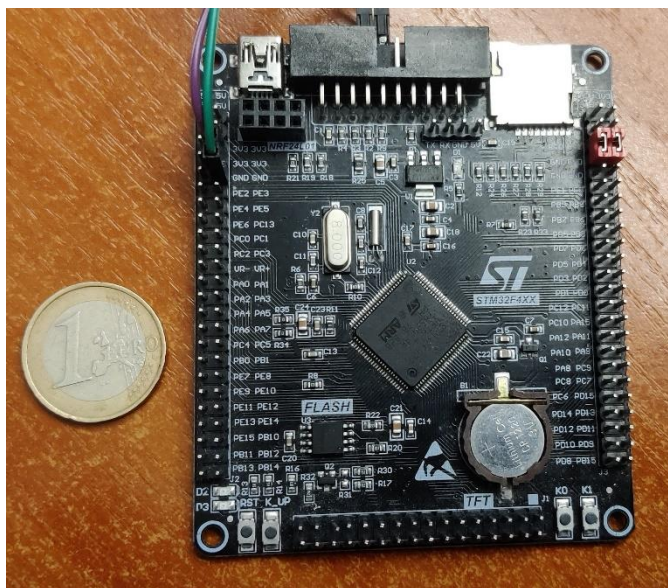
pav. 6 Micro:bit v1.3

[DTH60] Šis mikrokontroleris turi:

- Atmintis: 16 KB ROM, 256 KB „flash“, 32 KB RAM.
- Procesorius: 16 MHz 32-bit ARM Cortex-M0.
- Slankaus kablelio skaičiaus palaikymas: neturi.
- Energijos naudojimas: max 100mA.
- Prietaisas veikia nuo: 3.3 V įtampos maitinimo šaltinio.
- Išmatavimai: 52 x 40 x 10 mm.
- Kaina: apie 30 eurų.

1.7. STM32F407VE

STM32F4 yra galingesnė STM32F1 mikrokontrolerių serija su galingesniu procesoriumi Arm Cortex-M4, kuris turi FPU (slankaus kablelio palaikymą) ir daugiau įvesties/išvesties kontaktų (jų turi net 140). Be to, gausu komunikacijos protokolų sąsajų: trys I2C, keturi USART, trys SPI, du CAN ir SDIO. Į šį mikrokontrolerį galima įdėti SD kortelę, taip padidinant atmintį. Iš visų šiame darbe nagrinėjamų mikrokontrolerių būtent prie šio mikrokontrolerio gali būti prijungta daugiausia įrenginių vienu metu.



pav. 7 STM32F407VE

[DTH70] Šis mikrokontroleris turi:

- Atmintis: 1 MB „flash“, 512KB ROM, 192 KB RAM.
- Procesorius: 168 MHz 32-bit ARM Cortex-M4.
- Slankaus kablelio skaičiaus palaikymas: FPU aparatinėje įrangoje atlieka 32 bitų slankiąją Float ir 64 bitų dvigubo tikslumo Double matematinius veiksmus. Šių operacijų greitis yra maždaug toks pat kaip sveikųjų skaičių operacijų greitis.
- Energijos naudojimas: max 200mA.
- Prietaisas veikia nuo: 3.3 V įtampos maitinimo šaltinio.
- Išmatavimai: 85 x 75 x 10 mm.
- Kaina: apie 18 eurų.

1.8.Raspberry Pi 4

[MVD+14] Raspberry Pi 4 yra nepalyginamai greitesnis už aukščiau aprašytus mikrokontrolerius, bet jo lygint su mikrokontroleriais nederėtų, nes tai yra mini kompiuteris ar kitaip vienos plokštės kompiuteris (angl. „Single Board Computers“), turintis 2GB RAM ir 4 branduolių 1.5GHz procesorių su grafinių procesoriumi (GPU) kuris gali palaikyti 4k raiškos



pav. 8 Raspberry Pi 4

vaizdo įrašus. Be to, jis turi kelis USB2 ir USB3 lizdus, USB-C, microHDMI, microSD ir net 3.5mm audio jungtį. Taigi, šį prietaisą labiau derėtų lyginti su personaliniu kompiuteriu, nes prie jo galima prijungti monitorių, pelę, klaviatūrą ir į jį įrašyti operacinę sistemą. Šis įrenginys yra panašaus dydžio kaip STM32F4 ar Arduino DUE, todėl jis yra tinkamiausias, kai reikia atlikti daug sudėtingesnes užduotis, kurioms nepakanka paprasto mikrokontrolerio pajėgumo.

[DTH80] Šis mini kompiuteris turi:

- Atmintis: 2 GB RAM, nėra vidinės saugojimo atminties, todėl yra reikalinga SD atminties kortelė.
- Procesorius: Broadcom BCM2711 keturių branduolių 64-bit ARM-8 Cortex-A72 1.5 GHz.
- VideoCore VI 3D Graphics grafinis procesorius su 4Kp60 raiškos HDMI jungtimi
- Gigabit eternetu jungtis.
- Slankaus kablelio skaičiaus palaikymas: FPU aparatinėje įrangoje atlieka 32 bitų slankiąją Float ir 64 bitų dvigubo tikslumo Double matematinius veiksmus. Šių operacijų greitis yra maždaug toks pat kaip sveikųjų skaičių operacijų greitis.
- Energijos naudojimas: nuo 500mA.
- Prietaisas veikia nuo: 5 V įtampos maitinimo šaltinio.
- Išmatavimai: 89 x 59 x 18 mm. Kaina: apie 100 eurų.

1.9. Techninių savybių lentelė

lentelė 1 mikrokontrolerių savybių lentelė

	ESP8266	Due	Esp32	STM32F1	Teensy	Micro:Bit	STM32F4	Raspberry Pi 4
ROM/EEPROM	64 KB	512 KB	128 KB	128 KB	1 KB	16 KB	512 KB	-
Flash	4 MB	512 KB	4 MB	128 KB	2 MB	256 KB	1 MB	-
RAM	80 KB	96 KB	520 KB	20 KB	1 MB	32 KB	192 KB	2 GB
Branduoliai	1	1	2	1	1	1	2	4
Procesoriaus dažnis (greitis)	80 MHz	84 MHz	240 MHz	72 MHz	600 MHz	16MHZ	168 MHz	1.5 GHz
Ar turi slankaus kablelio skaičių palaikymą (FPU)	Ne	Ne	Turi	Ne	Turi	Ne	Turi	Turi
Apytikslė kaina	13 Eur	45 Eur	20 Eur	5 Eur	50 Eur	30 Eur	18 Eur	100 Eur
Maksimalus energijos sunaudojimas	70mA	150mA.	170mA.	100mA	100mA	100mA	200 mA	3A

2. Neuroninis tinklas

Šiame darbe, buvo atliekami tyrimai su tiesioginio skleidimo daugiasluoksnio (angl. „Feed-forward Multi-layer perception“) neuroninio tinklo algoritmo implementacijos modifikacijomis. Tinklo implementacijos sukurtos naudojant C++ programavimo kalbą ArduinoIDE aplinkoje. Tinklą sudaro 4 parametrai: įvesties sluoksnio (angl. „Input Layer“) neuronų skaičius, išvesties sluoksnio (angl. „Output Layer“) neuronų skaičius, vidinio sluoksnio (angl. „Hidden Layer“) neuronų skaičius ir vidinių sluoksnių skaičius. Buvo naudojama hiperbolinio tangento aktyvacijos funkcijos apytikslė funkcija ($f(x) = x / (1+|x|)$). Ši funkcija turi panašių savybių į hiperbolinio tangento funkciją, bet yra greitesnė (plačiau apie tai 3.2.4. skyrelyje). Pagal šiuos parametrus sukuriamas atitinkamo dydžio neuroninis tinklas. Tuomet padavus įvesties reikšmes yra apskaičiuojamos išvesties reikšmės ir tuo pačiu metu skaičiuojama, kiek kartų prasukamas ciklas per tam tikrą laiko vienetą. Tada didinamas vidinių sluoksnių skaičius, sukuriamas naujas tinklas ir kartojami skaičiavimai. Taip kuriamas vis didesnis tinklas tol, kol išnaudojama visa atmintis ir bandymas su mikrokontroleriu yra baigiamas.

Sukūrus tiesioginio skleidimo neuroninį tinklą buvo atliekami bandymai su skirtingais duomenų tipais: Double (64 bitų), Float (32 bitų), Float16 (16 bitų) bei Integer (32 bitų). Tinklas buvo sukurtas, kad veiktų su Double ir Float duomenų tipais, tačiau reikėjo kodo modifikacijų, kad veiktų ir su 16 bitų Float16 bei Integer duomenų tipais. Buvo panaudota Float16 biblioteka, kurioje yra sukurtas slankaus kablelio duomenų tipas Float16, kuriame skaičius saugomas 16 bitų. Kad pritaikyti šį duomenų tipą, turimame neuroniniame tinkle reikėjo šiek tiek pakoreguoti aktyvacijos funkciją, nes bibliotekoje nebuvo aprašytos reikalingos matematinės operacijos, toliau visas kodas tiko. Tuomet buvo sukurta tinklo modifikacija, kuri pritaikyta veikti su Integer (32bit) duomenų tipu. Integer tipo skaičius maksimaliai gali turėti 10 skaitmenų, tad buvo laikoma, kad iš kairės keturi skaitmenys yra sveikoji skaičiaus dalis, o likę šeši skaičiai būtų rašomi po kablelio. Taigi, intervalas nuo -1 iki 1 buvo ekvivalentus intervalui nuo -1 000 000 iki 1 000 000. Be to, buvo pakeista ir aktyvacijos funkcija, kad atitinkamai grąžintų ekvivalentaus intervalo reikšmes. Taigi, su šiomis keturiomis tinklo implementacijomis (Double, Float, Float16, Integer) buvo atliekami greitaveikos ir tinklo dydžio tyrimai.

2.1. Aktyvacijos funkcijos

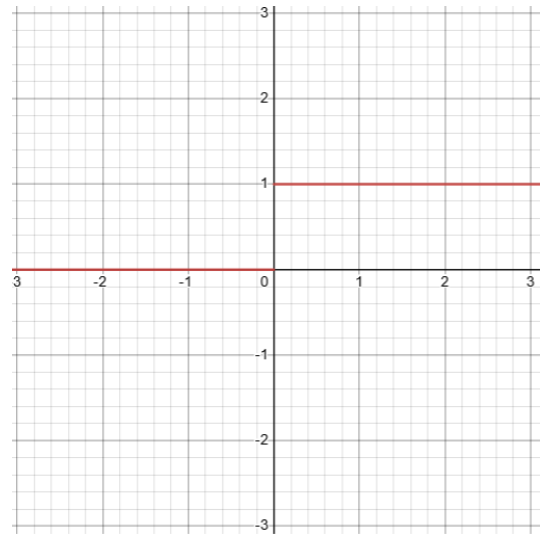
[SSA17] Aktyvacijos funkcijos yra naudojamos neuroniniame tinkle. Jų paskirtis yra apdoroti neurono įvesties reikšmes, sukuriant išvesties reikšmę, kuri yra perduodama sekančiam neuronui. Kiekvienam vidinio ir išvesties sluoksnio neuronui yra apskaičiuojami gautos įvesties reikšmės ir atitinkamų neuronų svoriai. Gaunama jų suma ir naudojantis aktyvacijos funkcija gaunama išvesties

reikšmė, kuri perduodama sekančiam neuronui ar yra išvedama kaip atsakymas išvesties sluoksnyje. Neuroninio tinklo tikslumas, greitaveika ir dydis (užimama vieta atmintyje) priklauso nuo tinklo dydžio ir aktyvacijos funkcijos. Aktyvacijos funkcijoje svarbiausias dalykas, į kurį reikia atsižvelgti, yra slenksčiu pagrįstas klasifikatorius, kuris reiškia, ar gauta reikšmė turi suaktyvinti neuroną, ar ne. Galima sakyti, kad neuronas suaktyvinamas, jei aktyvavimo funkcijos įvestis yra didesnė nei slenkstinė vertė, kitaip neuronas bus išjungtas.

2.2. Aktyvacijos funkcijų tipai

2.2.1. Binarinė žingsnio funkcija

Binarinė žingsnio funkcija – tai pati paprasčiausia aktyvacijos funkcija, kuri gražina nulį arba vienetą. Jei gaunamas teigiamas skaičius – gražinamas vienetas, jei neigiamas – nulis. Ši aktyvacijos funkcija pagrįde naudojama kuriant binarinį klasifikatorių. Kadangi iš jos gali būti gražinamos tik dvi skirtingos reikšmės, ji negali būti naudojama su daugiau nei dviem išvestimis. Be to, kai šios funkcijos gradientas yra lygus nuliui, tai gali tapti kliūtimi algoritmo apmokymui. Matematiškai šią funkciją galima apibrėžti kaip: $\{f(x)=1, x \geq 0; f(x)=0, x < 0\}$

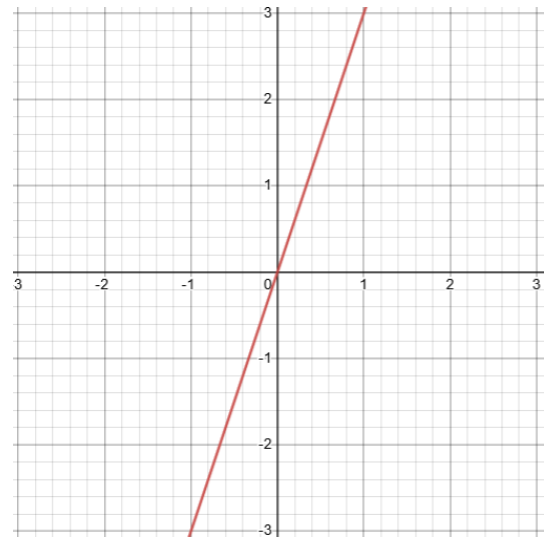


pav. 9 Binarinė žingsnio funkcija

2.2.2. Linijinė funkcija

Linijinė aktyvacijos funkcija yra tiesiogiai proporcinga įvesties reikšmei. Didžiausias binarinės žingsnio funkcijos trūkumas yra nulinis gradientas, šiam trūkumui pašalinti galima naudoti linijinę aktyvacijos funkciją.

Šios funkcijos išvestinė yra ne nulis, o lygi naudojamos konstantos reikšmei. Gradientas nėra nulis, o pastovi vertė, kuri nepriklauso nuo įvesties vertės, o tai reiškia, kad svoriai ir poslinkiai bus atnaujinami atgalinio skleidimo metu. Linijinės funkcijos naudojimas nėra labai naudingas, nes neuroninis tinklas nepagerins klaidos dėl tos pačios gradiento vertės kiekvienai iteracijai. Be to, tinklas negalės iš duomenų atpažinti sudėtingų modelių. Todėl linijinės funkcijos idealiai tinka atlikti paprastas užduotis. Matematiškai šią funkciją galima apibrėžti kaip: $f(x) = a * x$, kai a gali būti bet koks pasirinktas teigiamas skaičius.



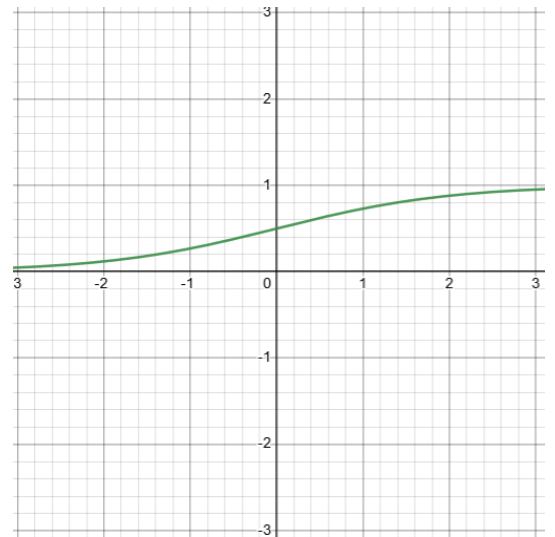
pav. 10 Linijinė funkcija

2.2.3. Sigmoido funkcija

Sigmoidinė aktyvacijos funkcija yra viena iš plačiausiai naudojamų ne linijinių aktyvacijos funkcijų. Ši funkcija transformuoja reikšmes intervale nuo 0 iki 1 ir yra nuolat diferencijuojama, sklandi S formos funkcija. Be to, sigmoidinė funkcija nėra simetriška nuliui, o tai reiškia, kad visų neuronų išvesties reikšmių ženklai bus vienodi. Šią problemą galima išspręsti sumažinus sigmoidinės funkcijos mastelį.

Matematiškai šią funkciją galima apibrėžti kaip:

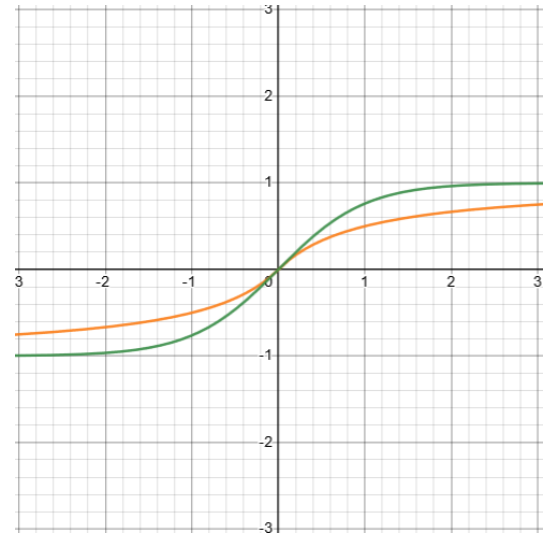
$$f(x) = 1/(1+e^{-x})$$



pav. 11 Sigmoido funkcija

2.2.4. Hiperbolinė tangento funkcija

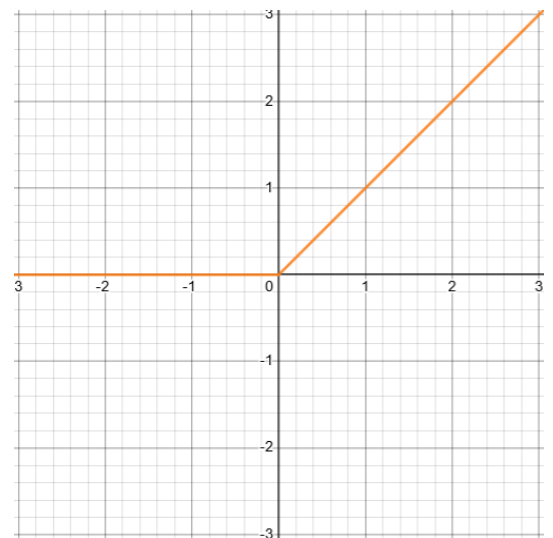
Hiperbolinė tangento ($\tanh$) funkcija yra panaši į sigmoidinę funkciją, tačiau ji yra simetriška ir transformuoja reikšmes nuo -1 iki 1. Lyginant su sigmoidine funkcija, $\tanh$ funkcijos gradientas yra statesnis. Šios funkcijos gradientas yra arti nulio; todėl galime lengvai priskirti išvesties reikšmes kaip stipriai neigiamas, neutralias arba labai teigiamas. Paprastai gradientas yra naudojamas vidiniuose neuroninio tinklo sluoksniuose, nes jo reikšmės yra nuo -1 iki 1; todėl vidinio sluoksnio vidurkis yra nulis arba labai arti jo. Tai padeda sutelkti duomenis ir palengvina mokymąsi kitam sluoksniui. Matematiškai šią funkciją galima apibrėžti kaip: $f(x) = (e^{2x}-1) / (e^{2x}+1)$ ir grafike ji yra žymima žalia spalva. Bet šiame darbe buvo naudojama greičiau apskaičiuojama šios funkcijos versija (pažymėta oranžinę spalva), $f(x) = x / (1 + |x|)$.



pav. 12 Hiperbolinė tangento ($\tanh$) funkcija

2.2.5. Parametrizuota ReLU funkcija

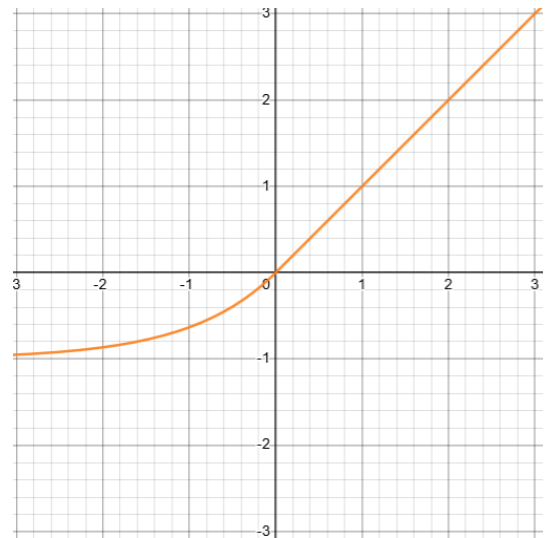
ReLU (Rectified Linear Unit) reiškia „dalimis tiesinis vienetas“ ir yra nelinejinė aktyvinimo funkcija, plačiai naudojama neuroniniame tinkle. Ši funkcija turi kelias variacijas. Paprastos ReLU pranašumas remiasi tuo, kad visi neuronai nėra aktyvuojami vienu metu. Tai reiškia, kad neuronas bus išjungtas tik tada, kai tiesinės transformacijos išvestis bus lygi nuliui. Bet trūkumas toks, kad kai kuriais atvejais gradiento reikšmė yra lygi nuliui, todėl svoriai ir poslinkiai neatnaujinami neuroninio tinklo mokymo atgalinio sklaidos žingsnio metu. Kita ReLU variacija – nesandarus ReLU (angl. „Leaky ReLU“), kurioje neigiamos x reikšmės dauginamos iš itin mažos konstantos (paprastai naudojama 0.01). Įvedant naują neigiamos funkcijos dalies parametą, t.y., nuolydį, išsprendžiama problema, kai ReLU gradientas esant neigiamoms x reikšmėms tampa nuliui. Matematiškai šią funkciją galima apibrėžti kaip: $f(x) = ax, x < 0; f(x) = x, x \geq 0, a \in \mathbb{R}^+$.



pav. 13 ReLU funkcija

2.2.6. Eksponentinė tiesinio vieneto funkcija

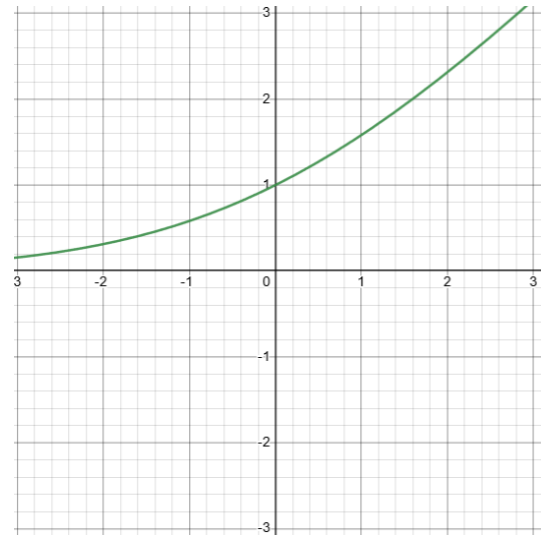
Eksponentinis tiesinis vienetas arba ELU (angl. „Exponential Linear Unit“) taip pat yra ReLU variantas. ELU įveda parametrų nuolydį neigiamoms x reikšmėms. Jis naudoja logaritminę kreivę neigiamoms reikšmėms apibrėžti. Matematiškai šią funkciją galima apibrėžti kaip: $f(x) = x, x \geq 0$; $f(x) = a(e^x - 1), x < 0, a \in \mathbb{R}^+$.



pav. 14 Eksponentinė tiesinio vieneto funkcija (ELU)

2.2.7. Švytuojanti funkcija

Švytuojanti (angl. „Swish“) funkcija yra palyginti nauja aktyvacijos funkcija, kurią atrado Google tyrėjai. Šios funkcijos skiriamasis bruožas yra tas, kad ji yra monotoniška, o tai reiškia, kad funkcijos reikšmė gali sumažėti, nors įvesties reikšmės didėja. Kai kuriais atvejais švytuojanti funkcija pranoksta net ReLU funkciją. Matematiškai šią funkciją galima apibrėžti kaip: $f(x) = x/(1 + e^{-x})$.



pav. 15 Švytuojanti funkcija (angl. „Swish“)

2.2.8. SoftMax funkcija

Softmax funkcija yra kelių sigmoidinių funkcijų derinys. Kadangi sigmoidinė funkcija grąžina reikšmes intervale nuo 0 iki 1, jas galima traktuoti kaip tam tikros klasės duomenų taškų tikimybes.

Softmax funkcija, skirtingai nei sigmoidinės funkcijos, kurios naudojamos dvejetainiam klasifikavimui, gali būti naudojama kelių klasių klasifikavimo problemoms spręsti. Softmax funkcija kiekvienam visų atskirų klasių duomenų taškui grąžina tikimybę. Kai sukuriame tinklą ar modelį kelių klasių klasifikacijai, tinklo išvesties sluoksnis turės tiek pat neuronų, kiek ir klasių.

2.3. Tinklo dydžio parinkimas

[HHP+12] Viena iš didžiausių ir svarbiausių neuroninio tinklo architektūros dalių yra tinklo dydis. Įprastai neuroninis tinklas turi 3 sluoksnius: įvesties, išvesties ir vidinius. Kiekvienas iš sluoksnių turi tam tikrą neuronų skaičių. Kad tinklas optimaliai veiktų, jis turi būti ne per didelis ir ne per mažas. Tai yra, per mažas tinklas, gali nepajėgti apskaičiuoti norimo rezultato, o per didelis – gali apsunkinti apmokymą, jis sulėtėja, atsiranda papildomo nereikalingo triukšmo. Deja, nėra konkretaus bendro būdo kaip iš karto nustatyti optimalų tinklo dydį. Vienas iš būdų yra naudojantis gairėmis nustatyti pradinį tinklo dydį ir eksperimentuojant, keičiant tinklo dydį, mokymosi konstantą, aktyvacijos funkciją ar kitus parametrus bandyti išgauti geresnį tinklo tikslumą ar mokymosi greitį. 3.3.1 – 3.3.3 skyreliuose aprašomos gairės pagal kurias kuriamas pradinis tinklas.

2.3.1. Įvesties sluoksnis

Įvesties sluoksnis yra pirmasis neuroninio tinklo sluoksnis, į kurį iš aplinkos paduodama neuroninio tinklo apdorojimui reikalinga informacija. Taigi, šio sluoksnio dydis sudaromas pagal tai, kokia informacija bus paduodama, t.y., kiek skirtingų duomenų atributų bus paduodama į tinklą. Kiekvienam skirtingam atributui priskiriamas atskiras neuronas.

2.3.2. Vidiniai sluoksniai

Vidinių sluoksnių ir neuronų kiekis priklauso nuo sprendžiamos problemos sudėtingumo. Jei problemą galima išspręsti tiesiškai, tai yra, iš pateiktų duomenų galima išvadas daryti iš karto, šio sluoksnio gali ir nebūti. Vis dėlto, įprastai sudėtingesnėms problemoms vidinis sluoksnis atlikti skaičiavimams yra reikalingas. Pagal [Pan11] šaltinį, paprastai daugumai uždavinių užtenka ir vieno vidinio sluoksnio. Daugiau nei vieno sluoksnio reikalauja sudėtingesni uždaviniai su pertrūkiais. Naudojant du vidinius sluoksnius retai pagerinamas modelis ir gali iškilti didesnė rizika užstrigti lokaliu minimumu taške, tai yra, nebus pasiektas norimas rezultatas.

Taigi, bendros gairės (siūlomi variantai) kuriant pirminį tinklo vidinio sluoksnio dydį yra:

- naudoti vieną sluoksnį,

- vidinio sluoksnio neuronų skaičius turėtų būti tarp įvesties ir išvesties sluoksnių neuronų skaičiaus,
- vidinio sluoksnio neuronų skaičius turėtų sudaryti $2/3$ įvesties ir išvesties sluoksnių neuronų kiekio,
- vidinių neuronų skaičius neturi viršyti įvesties sluoksnių skaičiaus padauginto iš dviejų.

Šios gairės yra skirtos pirminiam tinklo variantui, bet tai yra kaip pradžia, nuo kurios turėtų būti tobulinamas tinklas pasiekti geresniems rezultatams.

2.3.3. Išvesties sluoksnis

Išvesties sluoksnis yra tai, kas iš tikrųjų pateikia rezultatus išorinei aplinkai. Išvesties neuronų skaičius turėtų būti tiesiogiai susijęs su neuroninio tinklo atliekamo darbo tipu. Norėdami nustatyti neuronų, naudojamų išvesties sluoksnyje, skaičių, pirmiausia reikia nustatyti, kiek skirtingų atsakymų (rezultatų grupių, klasifikuojamų grupių) įmanoma gauti ir kiekvienam jam sukurti po neuroną. Jei neuroninis tinklas turi būti naudojamas elementams klasifikuoti į grupes, tuomet pageidautina turėti vieną išvesties neuroną kiekvienai grupei, kuriai turi būti priskirti įvesties elementai.

2.3.4. Išvados

Taigi, vien iš įvesties sluoksnio įėjimų ir išvesties sluoksnio išėjimų skaičiaus labai sunku nustatyti gerą tinklo topologiją. Ji labai priklauso nuo mokymui skirtų duomenų ir užduoties sudėtingumo.

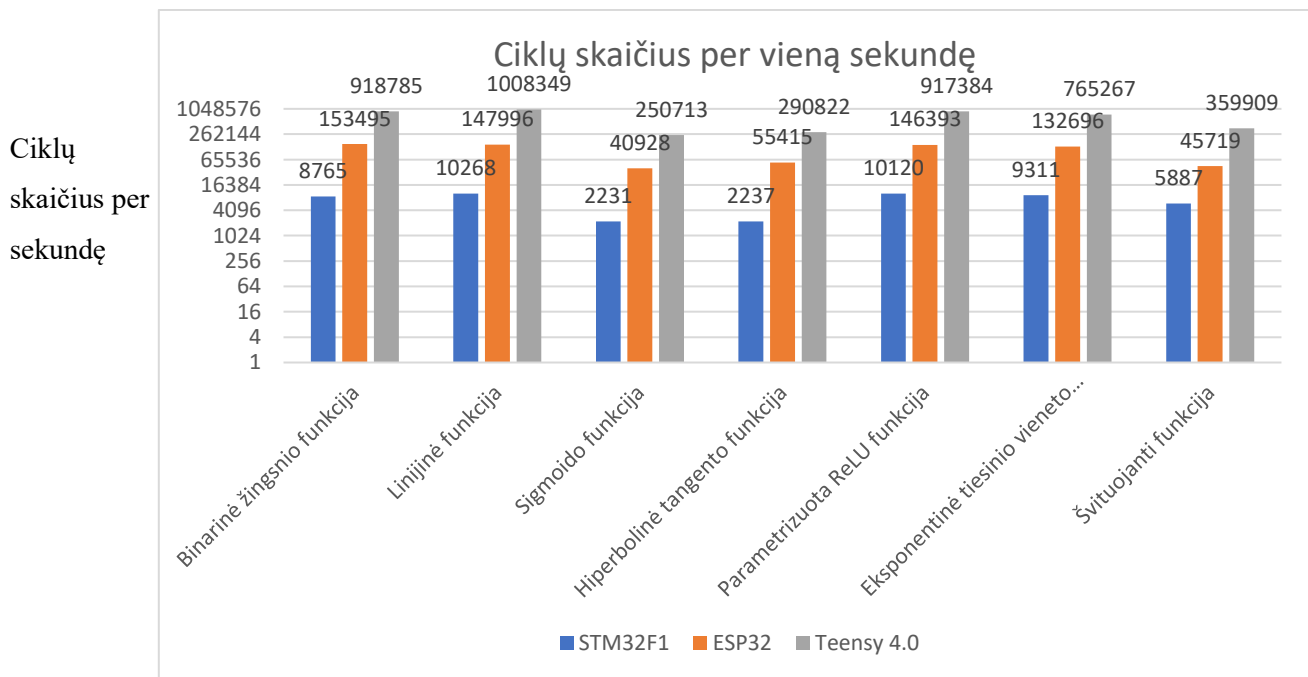
Padidinus vidinių sluoksnių skaičių iki trijų sluoksnių, tikslumas gali labai išaugti, tačiau neuroninio tinklo sudėtingumas ir mokymo laikas padidėtų daug kartų. Jei tikslumas yra pagrindinis neuroninio tinklo projektavimo kriterijus, vidinius sluoksnius galima padidinti. Vis dėlto, nereikalingas neuronų ar sluoksnio padidėjimas gali sukelti per didelio pritaikymo problemą. Taigi, labai svarbu, kad prieš kuriant neuroninį tinklą būtų išanalizuoti mokymo duomenų bazės pavyzdžiai, kad būtų galima tinkamai atspėti apytikslį neuronų ir vidinių sluoksnių skaičių.

3. Tyrimai

Šiame skyriuje aprašomuose tyrimuose visas dėmesys yra skiriamas tinklo greitaveikai, kiek ciklų per sekundę atlieka mikrokontrolerio/mini kompiuterio procesorius, ir tiriama, kokio dydžio neuroninis tinklas gali būti sutalpintas į kiekvieno mikrokontrolerio atmintį. Be to, tiriama aktyvacijos funkcijų greitaveika kiekviename mikrokontroleryje.

3.1. Aktyvacijos funkcijų tyrimas

Siekiant nustatyti kaip neuronų aktyvacijos funkcija lemia neuroninio tinklo greitaveiką, buvo atliktas tyrimas su tiesioginio skleidimo (angl. „Feed Forward“) tinklu, keičiant aktyvacijos funkcijas. Tyrimui buvo pasirinkti ESP32, STM32F103C, Teensy 4.0 mikrokontroleriai. Dėl skirtingų techninių savybių buvo pasirinkti būtent šie mikrokontroleriai. Pavyzdžiui, STM32F1 neturi slankaus kabelio skaičiaus palaikymo, ESP32 turi dalinį palaikymą, o Teensy 4.0 turi pilną slankaus kabelio skaičiaus palaikymą. Atliekant slankaus kabelio matematinius veiksmus su mikrokontroleriu, kuris turi šį koprosorių, pastebimas didelis pranašumas greitaveikoje.



grafikas 1

Iš pirmo grafiko galima pastebėti, kad greitaveikos atžvilgiu geriausi rezultatai pasiekti su binarinio žingsnio (angl. „Binary Step Function“), linijine (angl. „Linear“), parametrizuota ReLU (angl. „Parametrized ReLU“) ir eksponentinė tiesinio vieneto (angl. „Exponential Linear Unit“ - ELU) funkcijomis. Šiose funkcijose išskyrus ELU nėra vykdoma laipsnio kėlimo operacija, todėl šių

funkcijų ciklą per sekundę skaičius yra tris kartus didesnis nei sigmoido (angl. „Sigmoid“), hiperbolinio tangento (angl. „Hyperbolic Tangent“ – tanh) ar švytuojančios (angl. „Swish“) funkcijų. Be to, eksponentinėje tiesinio vieneto funkcijoje kėlimo laipsniu operacija vykdoma tik neigiamoms reikšmėms.

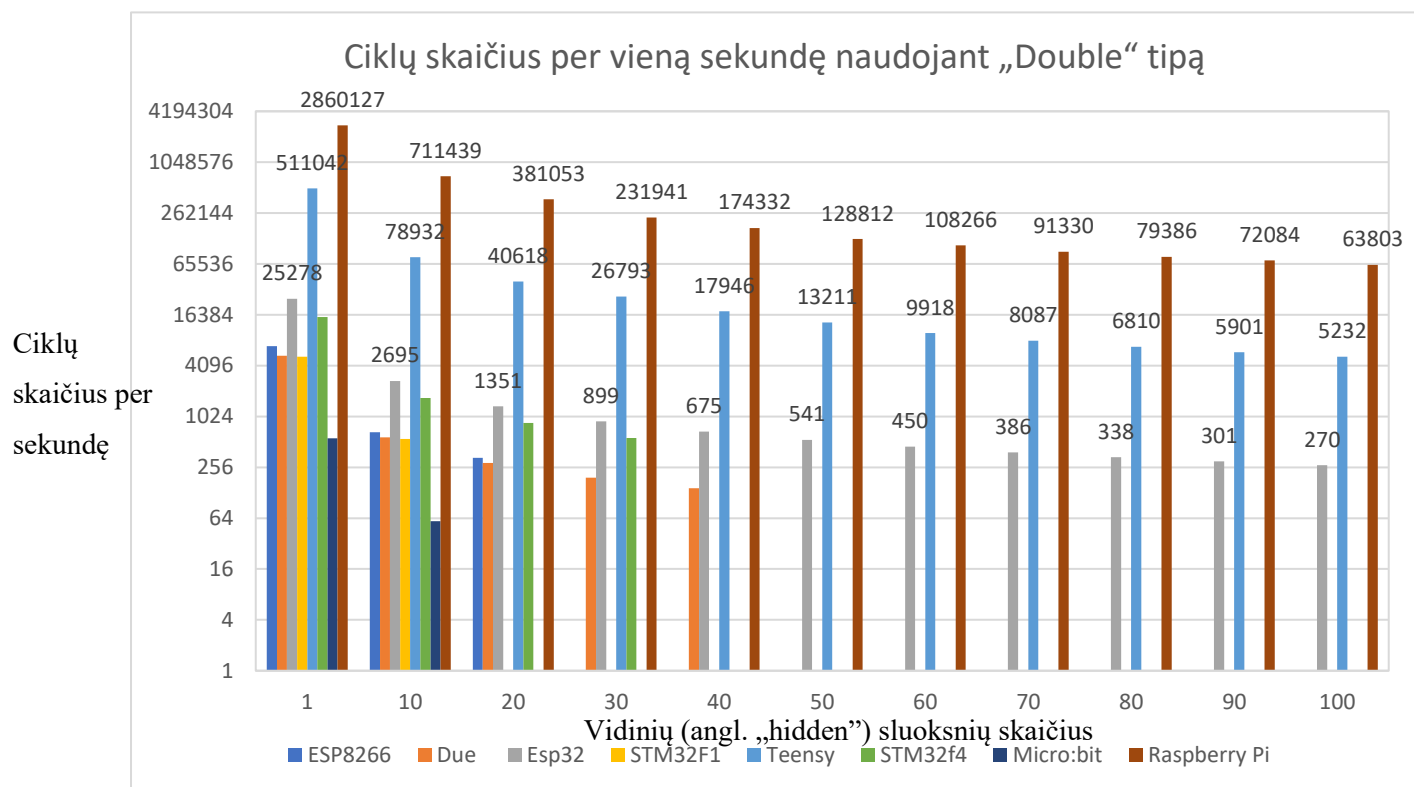
Aktyvacijos funkcijos neturi įtakos tinklo dydžiui, tai yra, galima sutalpinti tokio pačio dydžio neuroninį tinklą naudojant bet kurią iš aktyvacijos funkcijų, bet turi nemažą įtaką greitaveikai. Naudojant linijines aktyvavimo funkcijas: binarinio žingsnio, linijinę, ReLU ar ELU funkcijas galima pasiekti 3-4 kartus didesnį ciklą per sekundę skaičių nei naudojant sigmoido, hiperbolinio tangento ar swish funkcijas. [SSA17] Linijinės funkcijos naudojamos nesudėtingiems uždaviniams spręsti, kur išvestis yra tokia pati kaip pateikta įvestis su tam tikra paklaida. Bet dažniausiai neuroninis tinklas naudojamas sudėtingoms nelineinėms problemoms spręsti, kur geresnius rezultatus parodo nelineinės funkcijos.

3.2. Greitaveikos tyrimas

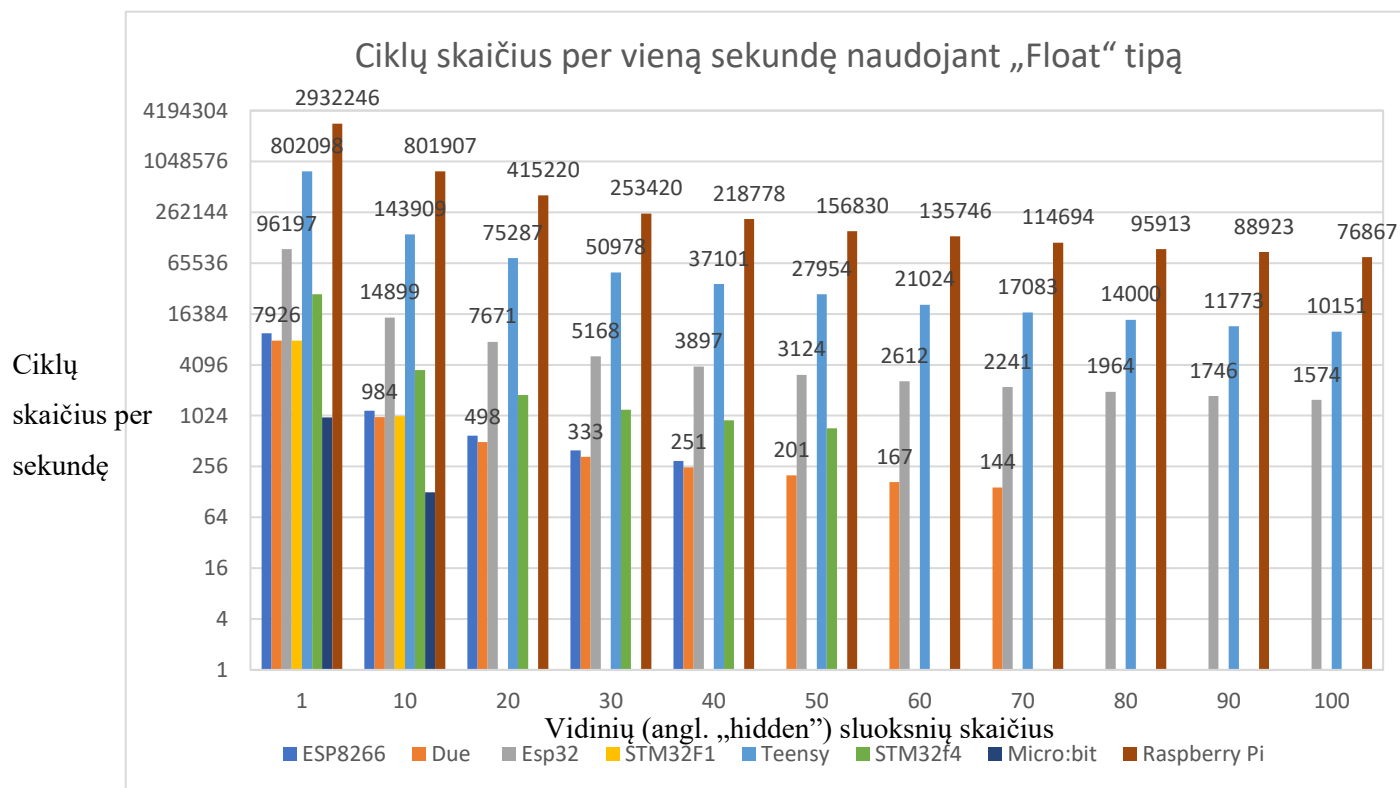
Greitaveikos tyrimo metu yra skaičiuojamas ciklą per sekundę skaičius, tai yra, kiek kartų kreipiantis į tinklą yra gaunamas rezultatas. Tiriama didinant tinklo vidinių sluoksnių (angl. „hidden layer“) skaičių nuo 1 iki 100, siekiant išsiaiškinti tinklo greitaveikos priklausomybę nuo tinklo dydžio. Buvo naudojama hiperbolinio tangento aktyvacijos funkcijos apytikslė versija, kuri turi panašių savybių, bet yra greitesnė.

Šiame tyrime buvo naudojami skirtingo duomenų tipo neuronų svoriai. Einant per kiekvieną neuroną yra atliekamos matematinės operacijos su neuronų svoriais, kai jie yra slankaus kablelio ar sveikųjų skaičių tipo.

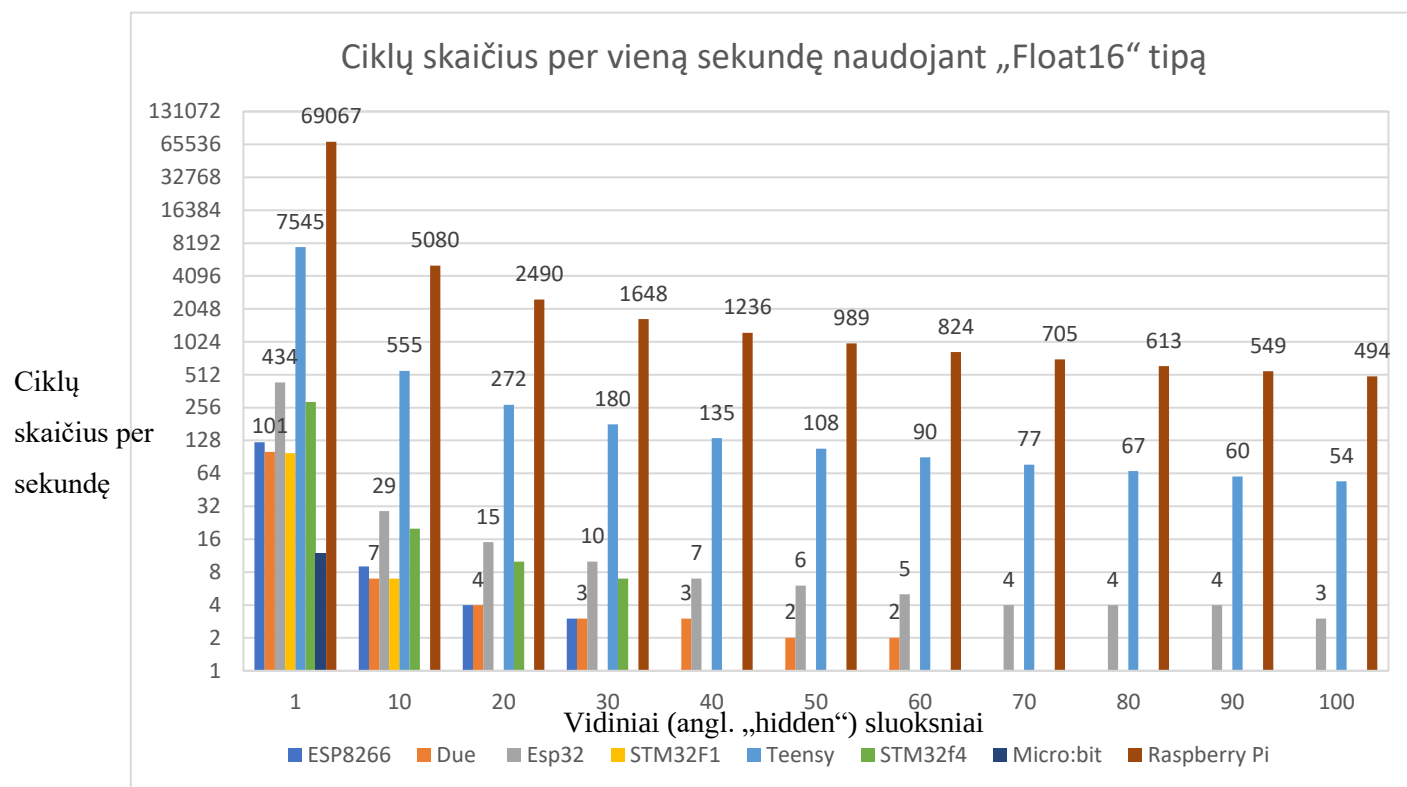
Bandymo metu nebuvo keičiamas įvesties (angl. „input layer“) sluoksnio neuronų skaičius (jis yra 4), išvesties (angl. „output layer“) sluoksnio neuronų skaičius (jis yra taip pat 4) ir vidinio sluoksnio neuronų skaičius (jis yra 5). Be to, buvo atlikti identiški bandymai, kai tinklo svoriai ir įvesties, išvesties reikšmės buvo Double, Float, Float16 ir Integer duomenų tipai. Bandymai daryti 5 kartus ir grafikuose pateikta rezultatų mediana.



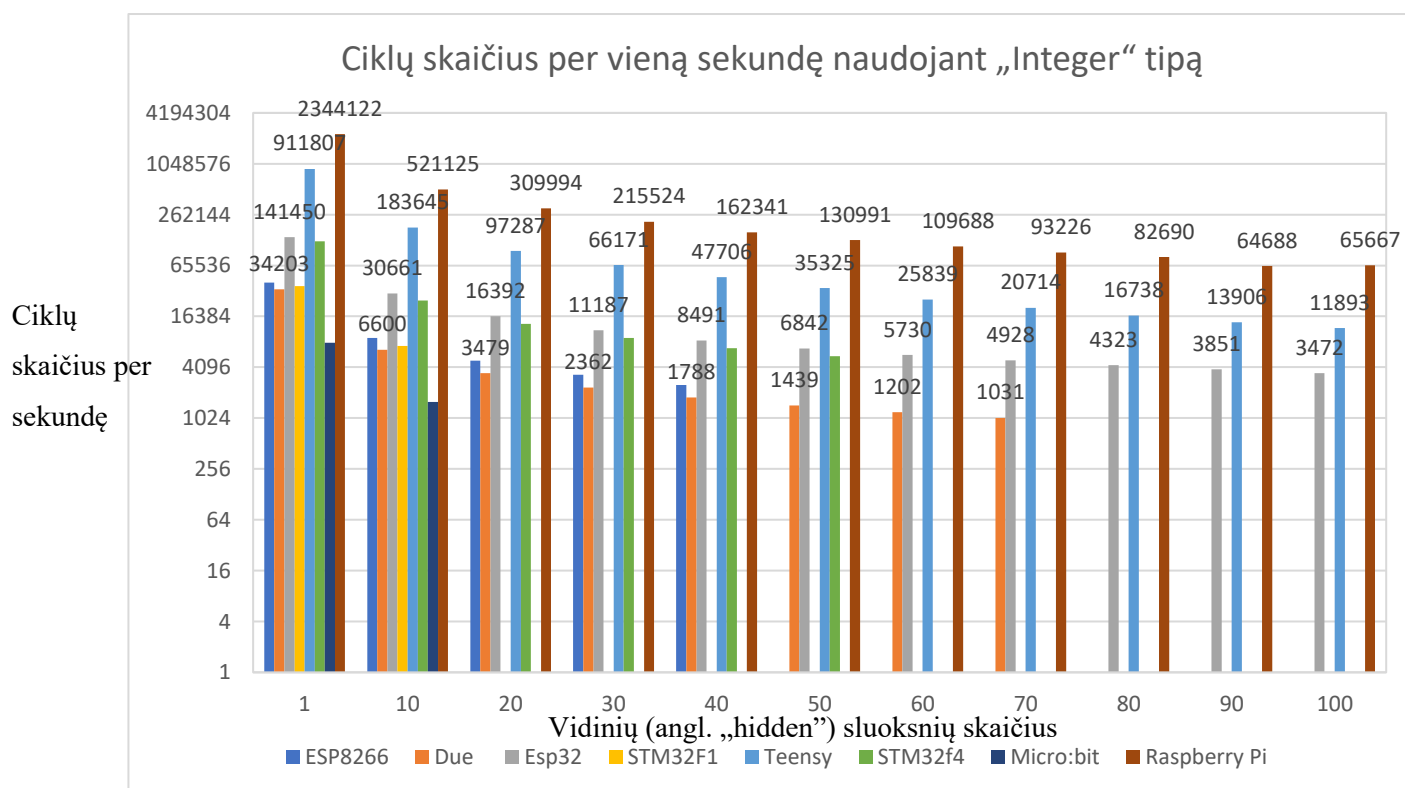
grafikas 2



grafikas 3



grafikas 4



grafikas 5

Logaritminės skalės grafikai vaizduoja ciklų skaičių per sekundę priklausomai nuo tinklo dydžio kiekvienam įrenginiui. Pačius geriausius rezultatus iš pateiktų įrenginių pasiekė mini kompiuteris Raspberry Pi. Šis įrenginys turi 4 branduolių 1.5 GHz mikroprocesorių ir 2 GB RAM atminties, todėl tai leidžia jam greičiau atlikti operacijas. Taigi, lyginat su bet kuriuo nagrinėtu mikrokontroleriu Raspberry Pi įrenginys turi daug geresnius techninius parametrus.

Nepriklausomai nuo vidinių sluoksnių skaičiaus daugiausia ciklų per sekundę iš mikrokontrolerių pasiekė Teensy mikrokontroleris. Šio mikrovaldiklio vieno branduolio procesoriaus taktinis dažnis siekia 600 Mhz. Taigi, lyginant su Raspberry Pi procesoriumi, jis yra dešimt kartų lėtesnis, bet skaičiuojant ciklų skaičių per sekundę Raspberry Pi buvo greitesnis 3.6 – 8 karto su skirtingomis tinklo konfigūracijomis. Paėmus Float tyrimo rezultatus, Raspberry Pi pasiekė 2 932 246 ciklų skaičių per sekundę, o Teensy – 802 098 ciklų skaičių per sekundę. Taip yra dėl to, kad Raspberry Pi turi savo operacinę sistemą, kuri taip pat naudoja resursus savo veikimui, todėl visi galimi resursai nėra skiriami konkrečiam uždaviniui. Teensy operacinės sistemos neturi, dėl to visas mikrokontrolerio pajėgumas yra skiriamas uždavinio sprendimui.

Taip pat daug ciklų per sekundę atliko ESP32 mikrokontroleris, turintis procesoriaus taktinį dažnį iki 240 Mhz ir STM32F4, turintis procesoriaus taktinį dažnį iki 168 MHz. Dėl slankaus kablelio skaičių palaikymo (FPU) ESP32, Teensy ir STM32F4 mikrokontroleriai greičiau susidoroja su slankaus kablelio skaičių matematinėmis operacijomis. Likusių ESP8266, Due, STM32 ir Micro:bit mikrokontrolerių rezultatai buvo mažesni ir apylygiai, nes jų procesoriaus taktinis dažnis yra iki 80 Mhz ir jie neturi FPU palaikymo.

Antrame ir trečiame grafike galima pastebėti skirtumą tarp Float ir Double duomenų tipų rezultatų. Pakitimas greitaveikoje yra nežymus, nepriklausomai ar mikrokontroleris turi slankaus kablelio skaičiaus palaikymą. Vis dėlto, galima pastebėti labai didelį greitaveikos sumažėjimą ESP32 mikrokontrolerio tyrimo rezultatuose, kai Float duomenų tipas yra pakeičiamas į Double. Pasak dokumentacijos[DTH30], šis mikrokontroleris turi slankaus kablelio skaičiaus palaikymą, t.y., kad atliekant operacijas tiek su Float, tiek su Double duomenų tipais rezultatai neturi labai stipriai keistis. Šiuo atveju greitaveika sumažėja apie 5 kartus, todėl panašu, kad šis mikrokontroleris pilnai palaiko tik 32 bitų slankaus kablelio skaičius.

Ketvirtajame grafike buvo naudojamas Float16 duomenų tipas. Šis duomenų tipas nėra standartinis, tad buvo naudojama Arduino Float16 biblioteka. Lyginant rezultatus su Double ir Float duomenų tipais pastabėtas proporcingas labai stiprus greitaveikos nuosmukis visuose įrenginiuose

įskaitant ir mini kompiuterį Raspberry Pi. Panagrinėjus šią biblioteką buvo pastebėta, kad kiekvieną kartą atliekant matematinę operaciją Float16 tipo skaičiai yra paverčiami į Double duomenų tipą, tada atliekama matematinė operacija ir Double skaičius konvertuojamas atgal į Float16. Taigi, yra atliekama daug papildomų konvertavimo veiksmų, kurie užtrunka ilgiau nei patys skaičiavimai. Siekiant geresnių rezultatų, turėtų būti perrašoma Float16 biblioteka, kurioje nebūtų konvertavimų į kitus duomenų tipus.

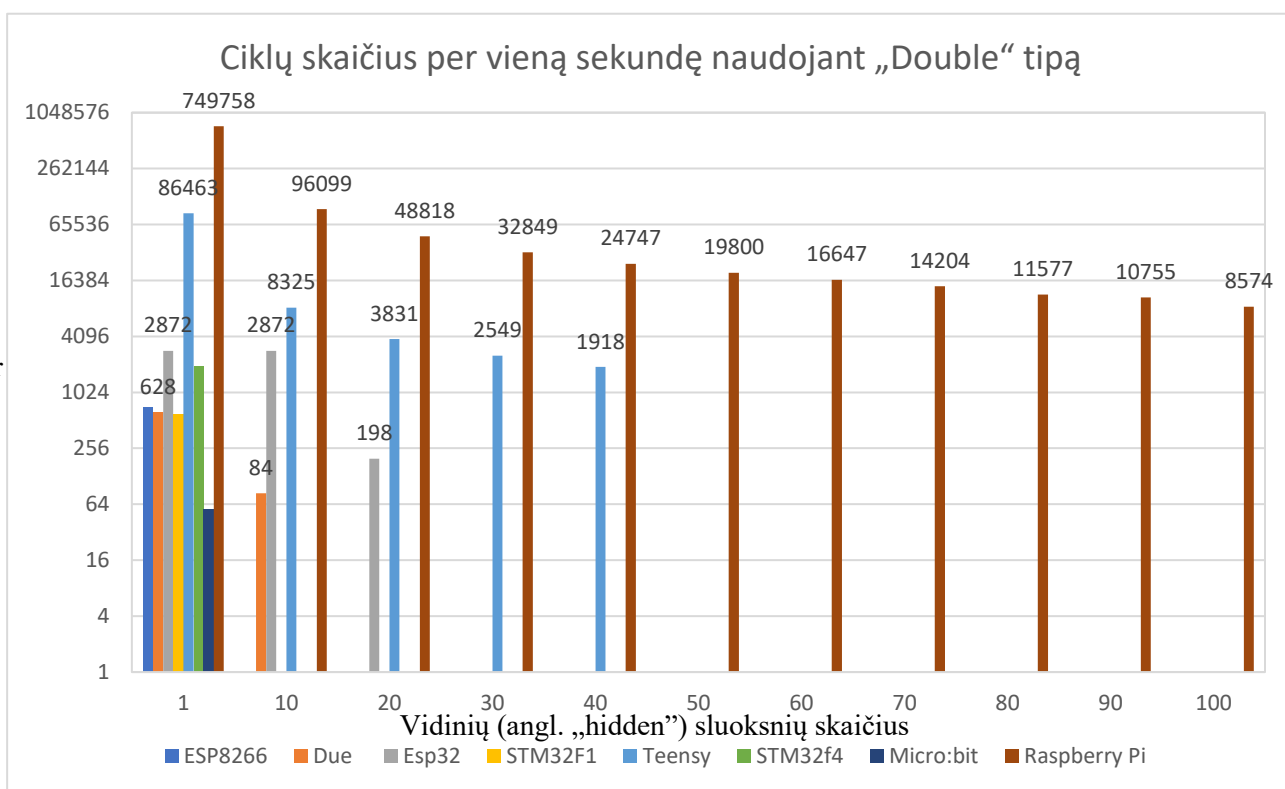
Penktajame grafike pateikiami tyrimo rezultatai naudojant sveikojo skaičiaus Integer duomenų tipą. Kad neuroninis tinklas galėtų veikti su šiuo duomenų tipu, reikėjo šiek tiek perrašyti kodą, pagrinde adaptuoti aktyvacijos funkciją. Šio tinklo variacijoje buvo laikoma, kad 1 yra ekvivalentus 1 000 000, tai yra, iš sveikojo skaičiaus, kuris daugiausia gali turėti dešimt skaitmenų, keturi skaitmenys iš kairės pusės yra laikomi sveikąja dalimi, o likę šeši skaitmenys iš dešinės pusės yra laikomi skaičiaus dešimtaine dalimi. Pavyzdžiui, 12,3456 ekvivalentus skaičiui 12 345 600.

Lyginant penktojo grafiko rezultatus su trečiajame grafike pavaizduotu Float duomenų tipo tyrimo rezultatais, pastabėtas nemažas greیتaveikos padidėjimas ypač tuose mikrokontrolieriuose, kurie neturi slankaus kablelio skaičiaus (FPU) palaikymo. ESP8266, Due ir STM32F1 mikrokontrolieriuose greیتaveika padidėjo daugiau nei 4 kartus, o Micro:bit net 8 kartus. Mikrokontrolieriuose su slankaus kablelio skaičiaus palaikymu lyginant su Float duomenų tipo tyrimo rezultatais ESP32 ir Teensy rezultatai pagerėjo nežymiai: ESP32 – 1.4 karto, Teensy – 1.14 karto. Su STM32F4 mikrokontrolieriu rezultatai pagerėjo panašiai kaip ir su mikrokontrolieriais be FPU palaikymo - 4.5 karto. Nors techninėje specifikacijoje [DTH70] parašyta, kad mikrokontrolieris turi FPU palaikymą, tačiau dėl gautų rezultatų kyla įtarimų dėl FPU koprosesoriaus nebuvimo mikrokontrolieryje. Reikėtų nuodugnesnių tyrimų, kad įsitikinti, ar tai tiesa.

Didinant tinklo dydį operacijų skaičius padidėja $10 \cdot N \cdot M$ (kai N ir M yra vidinio sluoksnio neuronų skaičius) operacijų skaičiumi. Tai geriausia matosi Teensy ir ESP32 mikrokontrolieriuose ir Raspberry Pi mini kompiuteryje. Padidinus tinklą nuo 10 iki 20 vidinių sluoksnių, greitis sumažėja maždaug 2 kartus, nuo 10 iki 100 - maždaug 10 kartų.

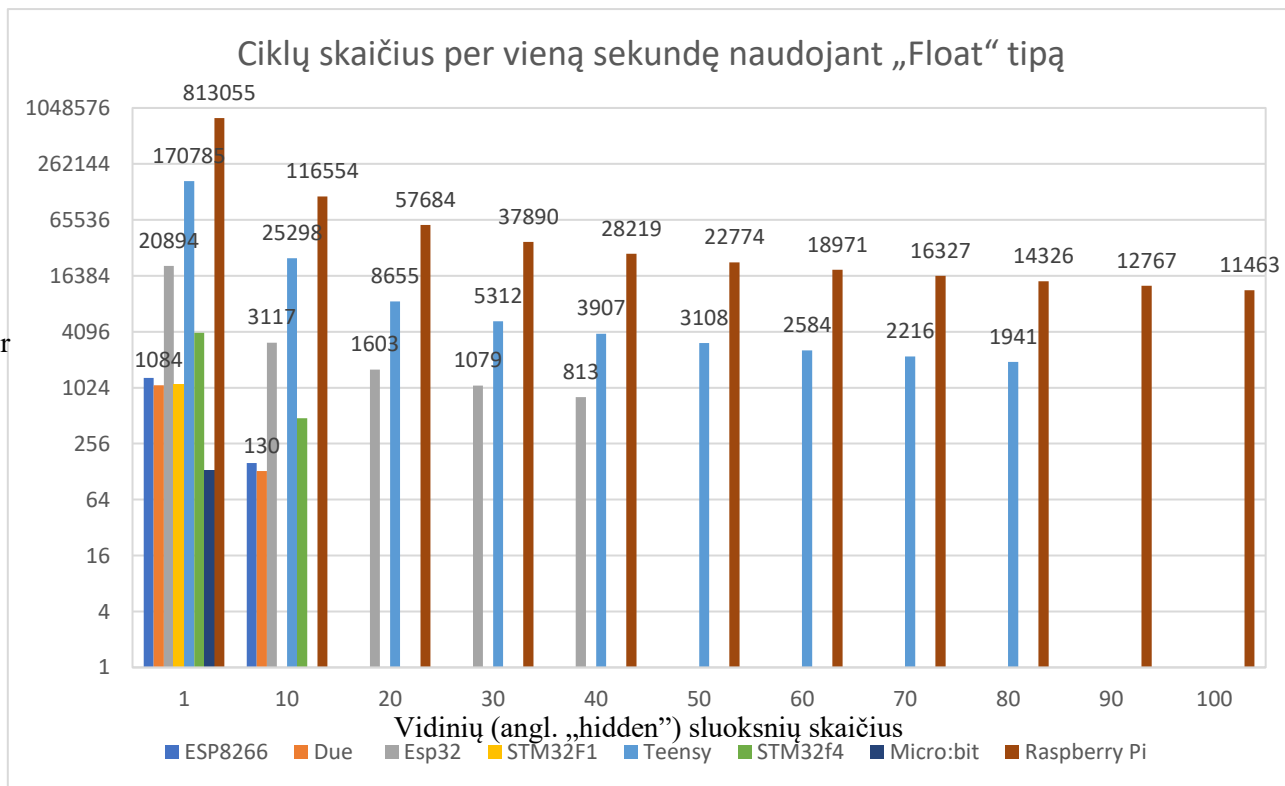
Antrojo bandymo metu buvo atliekami identiški tyrimai su Double, Float ir Integer duomenų tipais tik buvo pakeisti įvesties, vidinio ir išvesties sluoksnių neuronų skaičiai. Reikšmės buvo pakeistos į įvesties - 16, vidinio – 16, išvesties – 8.

Ciklų
skaičius per
sekundę

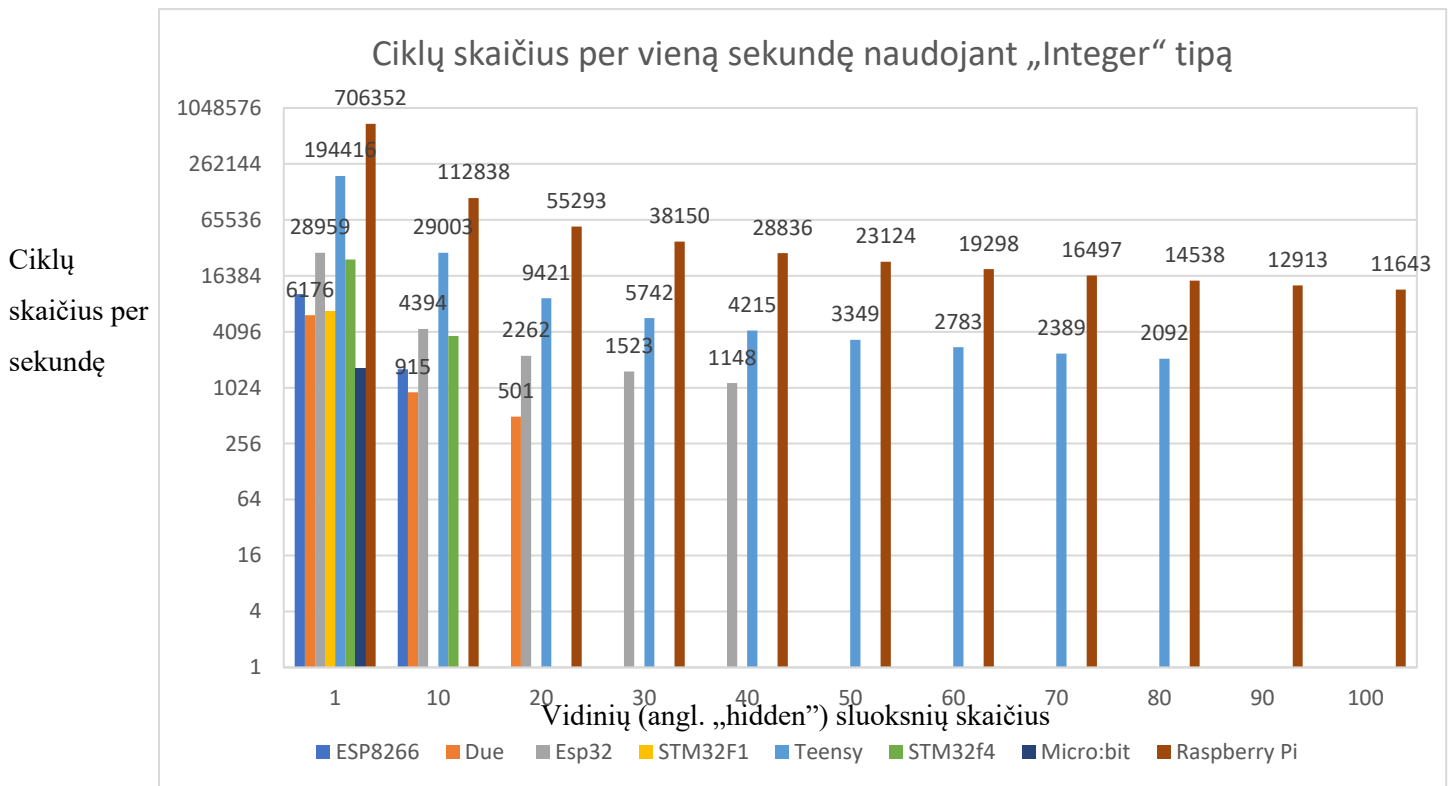


grafikas 6

Ciklų
skaičius per
sekundę



grafikas 7



grafikas 8

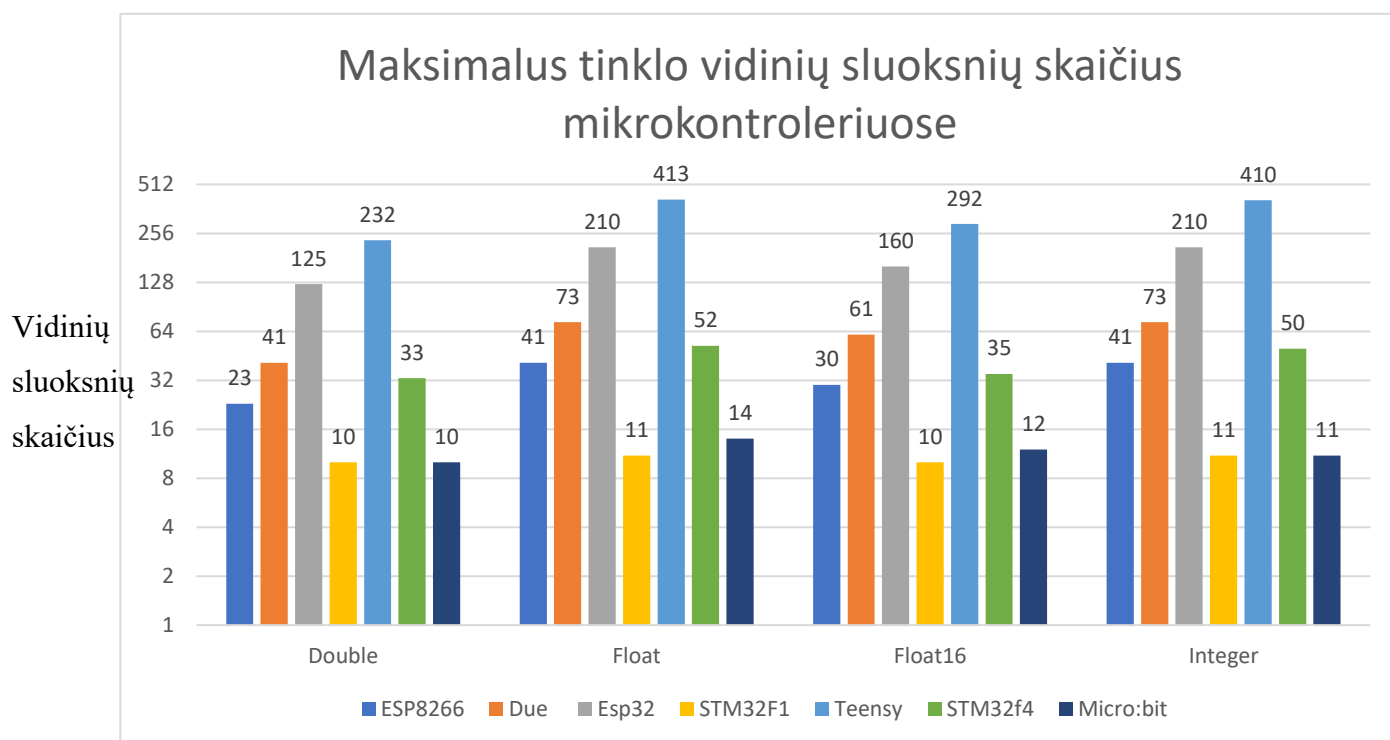
Padidinus duomenų kiekį šeštajame, septintajame ir aštuntajame grafike galima pastebėti kaip proporcingai sumažėjo greیتaveika. 4 kartus padidinus įvesties neuronų skaičių, 2 kartus išvesties neuronų skaičių ir 3,2 karto padidinus vidinių sluoksnių neuronų skaičių, yra atliekama apie septinis kartus mažiau ciklų per sekundę, priklausomai nuo tinklo dydžio.

Taigi, galima pastebėti, kad didžiausią pranašumą turi tie mikrokontroleriai, kurie savo procesoriuje turi slankaus kablelio skaičiaus palaikymą (FPU). Šis koprocesorius tokiems mikrovaldikliams leidžia atlikti matematinės operacijas su Float ar Double panašiu greičiu kaip atliekant operacijas su natūraliais skaičiais. Tyrimų atliktų su Float16 duomenų tipu rezultatai yra labai prasti, todėl būtų galima patobulinti biblioteką perrašant duomenų tipo matematinės operacijas be konvertavimo į kitą duomenų tipą. Mikrokontroleriams be FPU palaikymo greیتaveikoje pasiteisino tinklas su Integer duomenų tipu, su juo buvo pasiektas didesnis ciklų per sekundę skaičius. Aišku, su tokio tipo neuroniniu tinklu yra išsaugomi mažiau tikslūs neuronų svoriai, dėl to gali nukentėti tinklo tikslumas. Be to, daug brangesni ir galingesni mikrokontroleriai turi daug didesnę greیتaveiką dėl savo galingesnio procesoriaus.

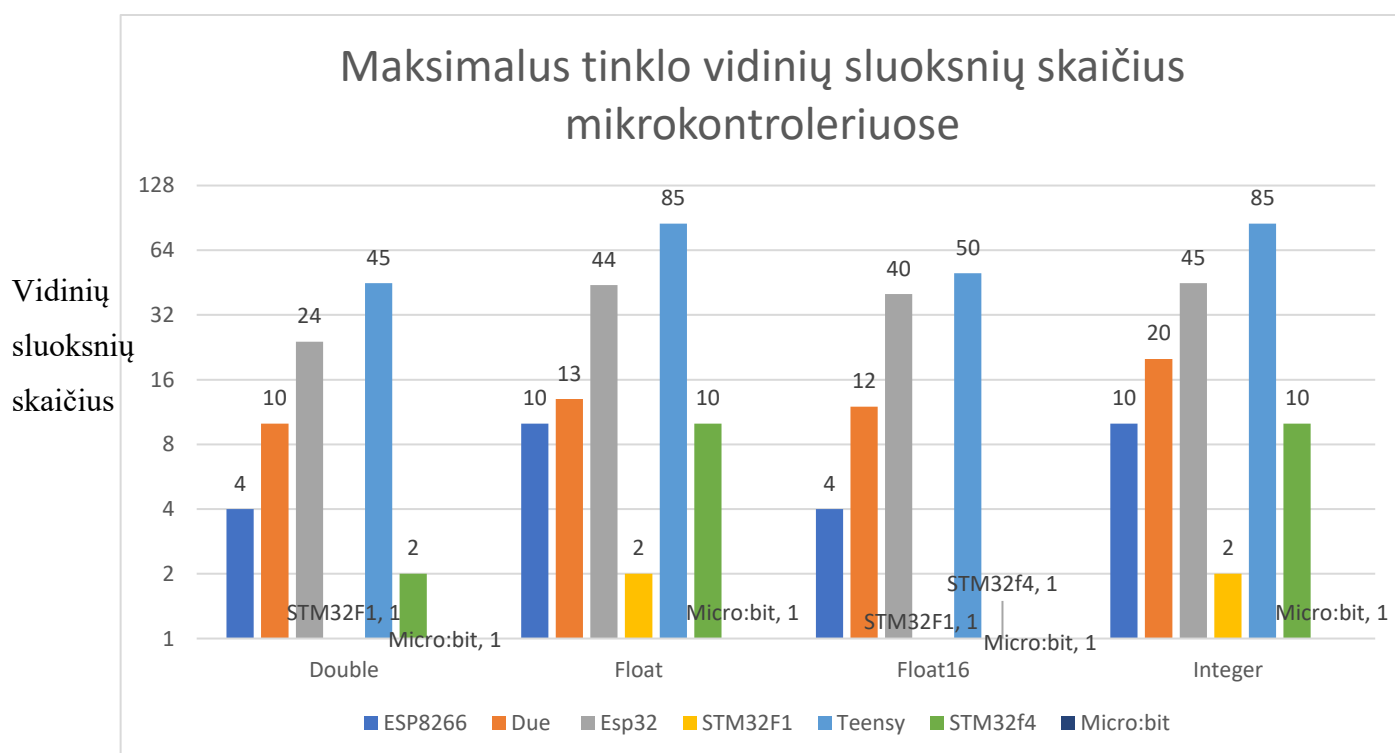
3.3. Tinklo dydžio tyrimas

Tinklo dydžio tyrimo metu yra skaičiuojama, kokį maksimalų tinklo dydį galima sutalpinti į tam tikrą mikrokontrolerį su Double, Float, Float16 slankaus kablelio ir Integer sveiko skaičiaus duomenų tipais. Nuo tinklo dydžio priklauso jo tikslumas: kuo didesnis tinklas, tuo didesnis tikslumas. Taigi, galima nesunkiai išsiaiškinti, kokio dydžio tinklą galima paleisti pasirinktuose mikrokontroleriuose ir sužinoti, kokio tikslumo iš jų galima tikėtis.

Bandymo metu buvo pasirinkti šie parametrai: įvesties (angl. „input layer“) sluoksnio neuronų skaičius, išvesties (angl. „output layer“) sluoksnio neuronų skaičius ir vidinio sluoksnio neuronų skaičius, kurie bandymo eigoje nekito. Pirmojo bandymo metu šių parametrų reikšmės buvo: įvesties - 4, išvesties - 4 ir vidinio - 5, o antrojo - įvesties sluoksnio neuronų skaičius 16, išvesties - 8 ir vidinio - 16. Tyrimo metu buvo keičiamas tik vidinio sluoksnio dydis, didinant jį nuo 1 iki kiek maksimaliai tilpo į atmintį. Be to, buvo atlikti identiški bandymai, kai tinklo svoriai ir įvesties, išvesties reikšmės buvo Double, Float, Float16 ir Integer duomenų tipai. Šis tyrimas nebuvo atliekamas su Raspberry Pi mini kompiuteriu dėl sąlyginai didelės jo RAM atminties (2GB). Nebūtų prasminga šio įrenginio lyginti su mikrokontroleriais, kurie turi tik iki 1 MB RAM atminties.



grafikas 9



grafikas 10

Kiekvieno mikrokontrolierio RAM atmintis skiriasi, dėl to skiriasi ir kokio dydžio tinklas tilpo. Todėl devintajame grafike galima pastebėti, kad, pavyzdžiui, ESP8266 mikrokontrolieriui su Double

duomenų tipu netilpo didesnis nei 23 vidinių sluoksnių neuroninis tinklas, su Float duomenų tipu – 41 vidinio sluoksnių neuroninis tinklas, su Float16 – 30 sluoksnių ir su Integer – 41 vidinio sluoksnių neuroninis tinklas. STM32F4 mikrokontroleryje su Double duomenų tipu netilpo didesnis nei 33 vidinių sluoksnių neuroninis tinklas, Float – 52, Float16 – 35, Integer – 50. To priežastis yra maža RAM atmintis: ESP8266 – 80 KB, STM32F4 – 192 KB. Į ESP32 ar Teensy mikrokontrolerį sutalpinamas gan didelis tinklas: ESP32 (Double – 125, Float – 210, Float16 – 160, Integer – 210), o Teensy (Double – 232, Float – 413, Float16 – 292, Integer – 410) vidinių sluoksnių tinklas. Toks maksimalaus talpinamo neuroninio tinklo dydžio skirtumas yra dėl to, kad palyginus su kitais aprašytais mikrokontroleriais, ESP32 ir Teensy mikrokontroleriai turi daug daugiau RAM atminties: ESP32 – 520KB ir Teensy – 1 MB.

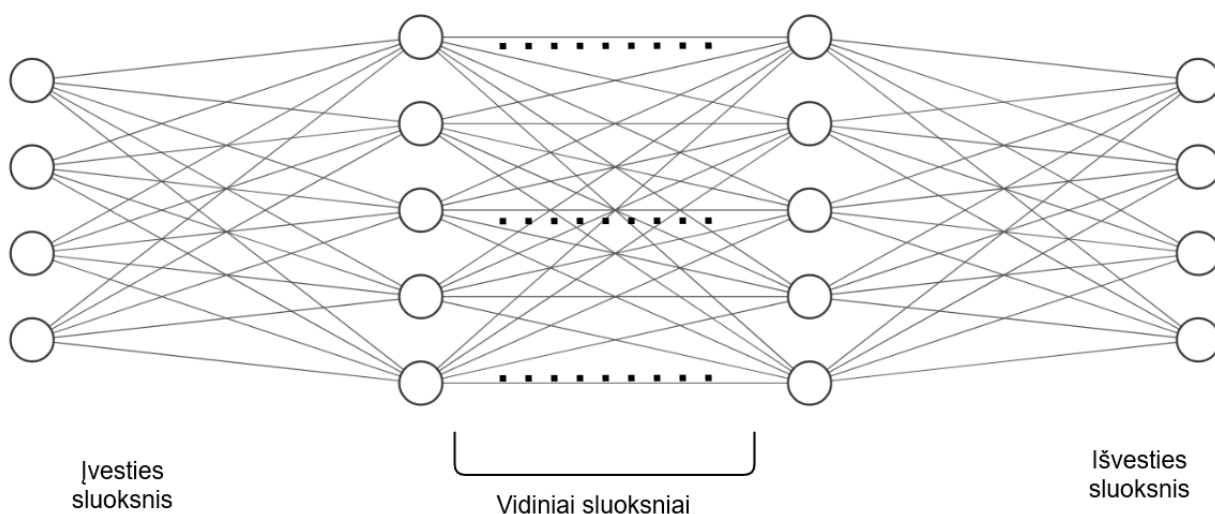
Tinklas, jo neuronai yra kuriami ir saugomi mikrokontrolerio operatyviojoje statinėje atmintyje (RAM). Būtent nuo šio parametro priklauso, kokio dydžio tinklą įmanoma sutalpinti mikrokontroleryje. Taigi, keisdami vidinių sluoksnių dydį sužinome, kokio dydžio tinklas gali tilpti mikrokontroleryje.

Šių mikrokontrolerių architektūrose slankaus kablelio duomenų tipai Float yra 32 bitų, o Double yra 64 bitų. Double duomenų tipas yra dvigubai tikslesnis nei Float, nes galima įrašyti tikslesnes reikšmes, bet taip pat užima dvigubai daugiau atminties. Tą puikiai galima pastebėti iš devintojo ir dešimtojo grafikų rezultatų. Sveikojo skaičiaus Integer duomenų tipas yra 32 bitų ir užima tiek pat atminties kiek ir Float. Tą galima pastebėti iš gautų rezultatų, pavyzdžiui, ESP32 tiek su Float, tiek su Integer gaunama 210 vidinių sluoksnių, analogiškai ir su kitais mikrokontroleriais. Analogiškai taip turėtų būti ir su Float16 duomenų tipu, tai yra, turėtų užimti per pus mažiau atminties nei Float, bet taip nėra. Devintajame grafike gaunami tokie ESP8266 rezultatai: Double – 23, Float – 41, Float16 – 30. Su Float gaunamas beveik dvigubai didesnis maksimalus vidinių sluoksnių skaičius nei su Double, o su Float16 netgi didesnis nei su Float. Panagrinėjus Float16 biblioteką galima pamatyti, kad šį duomenų tipą sudaro 2 kintamieji: UINT16_T 16 bitų sveikojo skaičiaus be ženklų duomenų tipas, kuriame saugoma skaičiaus reikšmė ir UINT8_T 8 bitų sveikojo skaičiaus be ženklų duomenų tipas, kuriame saugoma slankaus kablelio vieta.

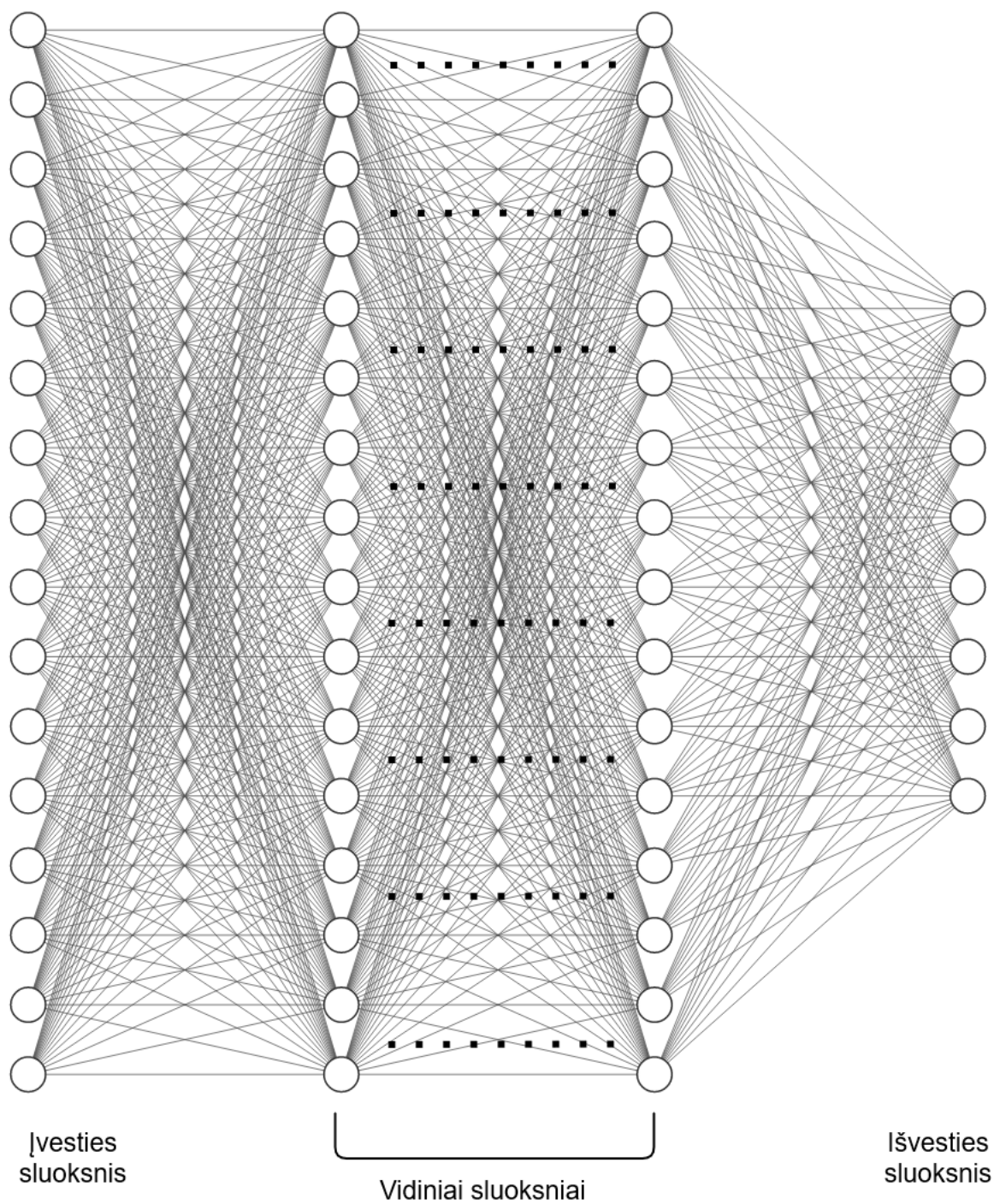
Neuronų skaičius = įvesties sluoksnių dydis + vidinio sluoksnių dydis * vidinių sluoksnių kiekis + išvesties sluoksnių dydis. Tinklo dydis apskaičiuojamas bitais su Double, Float, Float16 ir Integer duomenų tipais.

lentelė 2

Neuroninio tinklo architektūra				Tinklo dydis			
Ivesties sluoksnio dydis	Vidinio sluoksnio dydis	Vidinių sluoksnių kiekis	Išvesties sluoksnio dydis	Neuronų skaičius	Su Double bitais	Su Float/Integer bitais	Su Float16 bitais
4	5	1	4	13	6368	3392	1904
4	5	10	4	58	39488	20672	11264
4	5	20	4	108	76288	39872	21664
4	5	50	4	258	186688	97472	52864
4	5	100	4	508	370688	193472	104864
16	16	1	8	40	52992	27136	14208
16	16	10	8	184	361728	183808	94848
16	16	20	8	344	704768	357888	184448
16	16	50	8	824	1733888	880128	453248
16	16	100	8	1624	3449088	1750528	901248



pav. 16 4-5-...-4 dydžio neuroninis tinklas



pav. 17 16-16-...-8 dydžio neuroninis tinklas

Šešioliktoje ir septynioliktoje nuotraukose matomos šiame darbe nagrinėjamų tinklų dydžio architektūros schemas, kuriose žymimi įvesties, vidiniai ir išvesties sluoksniai. Antroje lentelėje yra matoma kai kurių darbe naudojamų neuroninių tinklų architektūros priklausomybė nuo parametrų, kiek naudojama neuronų ir kiek operatyvios atminties bitais sunaudoja skirtingų dydžių tinklai.

Išvados

Šiame darbe buvo siekiama susipažinti su mikrokontroleriais ir jų techninėmis savybėmis ir kaip jos daro įtaką neuroninių tinklų veikimui ir greitaveikai. Tikslui pasiekti buvo įvykdytos šios užduotys: sukurtas neuroninis tinklas mikrokontrolerių testavimui, susipažinta su skirtingų tipų mikrokontroleriais, jų techninėmis savybėmis, paleistos neuroninio tinklo modifikacijos skirtinguose mikrokontroleriuose, susipažinta su skirtingų tipu aktyvacijos funkcijomis, išnagrinėta tinklo dydžio parinkimo metodika, atlikti greitaveikos ir tinklo dydžio tyrimai, ištirta Double, Float, Float16 ir Integer duomenų tipų įtaka tyrimų rezultatams, apibendrintos tyrimų metu gautos išvados.

Projekto metu buvo sukurtas tiesioginio sklidimo daugiasluoksnis (angl. „Feed-forward Multi-layer perception“) neuroninis tinklas, kuris buvo modifikuojamas greitaveikos ir atminties naudojimo tyrimams kiekvienam mikrokontroleriui ir Raspberry Pi mini kompiuteriui. Tinklo greitaveika ir dydis buvo tiriami neapmokius tinklo, kadangi tinklo apmokymas nedaro įtakos greitaveikos ar dydžio eksperimentų rezultatams. Taip yra todėl, nes apmokymo metu yra keičiamos tik neuronų svorių reikšmės, bet jos užima tiek pat vietos ir sąlyginai nekeičia matematinių operacijų apskaičiavimo greičio. Taigi, tyrimo metu cikliška buvo didinamas tinklas iki tokios ribos, kai būdavo išnaudota visa operatyvi (RAM) atmintis. Šie rezultatai apipavidalinti grafikais, jie palyginami 4 skyriuje.

Taigi, tyrimams buvo atrinkti 7 skirtingi mikrokontroleriai ir mini kompiuteris Raspberry Pi, kurie palaiko ArduinoIDE aplinką. Jie pasižymi skirtingomis techninėmis savybėmis, nuo kurių priklauso jų rezultatai. Iš atliktų tyrimų akivaizdžiai matosi mikrokontrolerių turinčių slankaus kablelio palaikymą (FPU) pranašumas neuroniniuose tinkluose su slankaus kablelio skaičiaus neuronų svoriais. Jų procesoriai daug greičiau atlieka matematines operacijas su slankaus kablelio skaičiais, t.y., neuroninis tinklas greičiau apskaičiuoja rezultatą. Taip pat, mikrokontroleriai, kurie turi galingesnį procesorių, sugeba atlikti daugiau ciklų per tą patį laiką, bet tokių mikrokontrolerių kaina yra didesnė. Kadangi sveikojo skaičiaus Integer tyrimų rezultatai buvo geresni nei atlikti su slankaus kablelio Float duomenų tipu, siekiant padidinti greitaveiką mikrokontroleriuose be FPU palaikymo siūloma naudoti neuroninį tinklą su sveikojo skaičiaus (Integer) neuronų svoriais. Mikrokontroleriuose su FPU palaikymu nebuvo pastebėtas žymus greitaveikos padidėjimas. Kadangi pokytis nežymus, galima naudoti ir įprastą neuroninį tinklą su slankaus kablelio duomenų tipu.

Tinklas, jo neuronai yra kuriami ir saugomi operatyviojoje atmintyje (RAM). Būtent nuo šios atminties dydžio priklauso, kokio dydžio tinklą įmanoma sutalpinti mikrokontroleryje. Taigi, didinant vidinių sluoksnių dydį buvo išsiaiškinta, kokio dydžio tinklas gali jame tilpti. Mikrokontrolerio

tinkamumas priklauso nuo problematikos sudėtingumo, dydžio ir norimo tikslumo. Naudojant slankaus kablelio skaičiaus Float duomenų tipą galima sutaupyti dvigubai atminties ar sutalpinti dvigubai didesnę neuroninį tinklą nei naudojant Double duomenų tipą. Naudojant didesnę tinklą galima gauti tikslesnius rezultatus. Naudojant sveikojo skaičiaus (Integer) duomenų tipą galima sutalpinti tokio pačio dydžio tinklą kaip ir su Float duomenų tipu, nes jie abu yra 32 bitų.

Tiriant šiame darbe naudojamą Float16 duomenų tipą dėl prastos jo implementacijos gauti labai prasti rezultatai tiek tinklo dydžio, tiek greیتaveikos tyrimuose. Nors teoriškai lyginant Float16 su Float duomenų tipu turėjo būti atliekama beveik 2 kartus daugiau ciklų per sekundę, kaip pastebima Float (32 bitų) ir Double (64 bitų) neuroninių tinklų greیتaveikos rezultatuose. Prasti Float16 rezultatai gaunami, nes atliekant matematinės operacijas Float16 tipo skaičiai yra paverčiami į Double duomenų tipą, o tuomet atliekama matematinė operacija ir Double skaičius konvertuojamas atgal į Float16. Taigi, yra atliekama daug papildomų konvertavimo veiksmų, kurie užtrunka ilgiau nei patys skaičiavimai. Lyginant Float16 ir Float maksimalaus tinklo dydžio rezultatus buvo gauta, kad su Float16 duomenų tipu tilpo mažesnis tinklas nei su Float duomenų tipu. Teoriškai Float yra 32 bitų, o Float16 yra 16 bitų, tad su Float16 turėjo tilpti daugiau vidinių sluoksnių. Todėl esant poreikiui sutaupyti atminties nesiūloma naudoti Arduino Float16 bibliotekos. Vietoje šios bibliotekos geriau naudoti neuroninį tinklą su sveikojo skaičiaus duomenų tipu, tik vietoje Integer galima panaudoti 16 bitų INT16_T duomenų tipą, kuris yra per pus mažesnis už Integer.

Aktyvacijos funkcijų paskirtis neuroniniame tinkle yra apdoroti neurono įvesties reikšmės, sukuriant išvesties reikšmę, kuri yra perduodama sekančiam neuronui. Svarbiausias dalykas į kurį reikia atsižvelgti pasirenkant aktyvacijos funkciją yra slenksčiu pagrįstas klasifikatorius, kuris reiškia, ar gauta reikšmė turi suaktyvinti neuroną, ar ne. Šios funkcijos neturi įtakos tinklo dydžiui, tai yra, galima sutalpinti tokio pačio dydžio neuroninį tinklą naudojant bet kurią iš aktyvacijos funkcijų, bet turi nemažą įtaką greیتaveikai. Naudojant linijines aktyvavimo funkcijas (binarinio žingsnio, linijinę, ReLU) galima pasiekti 3-4 kartus daugiau ciklų per sekundę nei naudojant sigmoido, hiperbolinio tangento ar swish funkcijas.

Optimalus neuroninio tinklo veikimas priklauso nuo tinklo dydžio: jis turi būti ne per didelis ir ne per mažas. Nors konkretaus bendro būdo kaip iš karto nustatyti optimalų tinklo dydį nėra, pirminio tinklo dydį galima nustatyti remiantis gairėmis. Įvesties sluoksnio dydis sudaromas pagal į tinklą paduodamų atributų skaičių: kiek yra skirtingų atributų tipų, tiek neuronų sukuriami. Tinklo vidinių sluoksnių dydis priklauso nuo sprendžiamos problematikos sudėtingumo, bet daugeliui

uždavinių pakanka vieno sluoksnio. Vidinio sluoksnio pradinį neuronų skaičių rekomenduojama pasirinkti $2/3$ įvesties ir išvesties sluoksnių neuronų skaičiaus arba/ir vidinių neuronų skaičius neturėtų viršyti įvesties sluoksnių skaičiaus padauginto iš dviejų arba/ir vidinių neuronų skaičius neturi viršyti įvesties sluoksnių skaičiaus padauginto iš dviejų. Išvesties sluoksnio neuronų skaičius sudaromas pagal tai, kiek skirtingų atsakymų įmanoma ar tikimasi gauti ir kiekvienam galimam atsakymui sukuriamas po neuroną. Siekiant išgauti geresnį tinklo tikslumą ar mokymosi greitį, siūloma eksperimentuoti su mokymosi konstantomis, aktyvacijos funkcijomis ir kitais parametrais.

Summary

This paper examines Feed-forward multi-layer perception neural network with various data types neurons weights in microcontrollers. To achieve this goal, a number of different microcontrollers were introduced and compared by their types and properties. Also, a neural network with different data types, such as Double, Float, Float16 and Integer was created and runed in different microcontrollers. Various neural network activation functions were examined and compared by their performance in speed (network cycles per second). Investigated network selection methodologies and recommendations. A network speed (network cycles per second) and size (maximum size network that fits) studies were conducted with each microcontroller. Then their performance was compared to find suitable Feed-forward neural network for each type of microcontroller.

Literatūra

- [ABB+20] Austin, J., Baker, H., Ball, T., Devine, J., Finney, J., De Halleux, P., Hodges, S., Moskal, M. and Stockdale, G., 2020. The BBC micro: bit: from the UK to the world. *Communications of the ACM*, 63(3), pp.62-69.
- [AMN+21] Andramuño, J., Mendoza, E., Núñez, J., & Liger, E. (2021). Intelligent distributed module for local control of lighting and electrical outlets in a home. *Journal of Physics: Conference Series*, 1730(1), 012001. doi:10.1088/1742-6596/1730/1/012001
- [DS18] Damcı, E.; Şekerci, Ç. (2018). Development of a Low-Cost Single-Axis Shake Table Based on Arduino. *Experimental Techniques*, (), – doi:10.1007/s40799-018-0287-5
- [DTH10] https://components101.com/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf
- [DTH20] https://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf
- [DTH30] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [DTH40] <https://datasheetspdf.com/pdf-file/866258/STMicroelectronics/STM32F103C8/1>
- [DTH50] https://www.pjrc.com/teensy/IMXRT1060CEC_rev0_1.pdf
- [DTH60] <https://tech.microbit.org/hardware/>
- [DTH70] <https://www.st.com/resource/en/datasheet/stm32f407ve.pdf>
- [DTH80] <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>
- [GCK+17] Y. Güven, E. Coşgun, S. Kocaoğlu H. Gezici E. Yilmazlar 2017. „Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications“

- [HHP+12] Hunter, D.; Hao Yu; Pukish, M.S.; Kolbusz, J.; Wilamowski, B.M. (2012). Selection of Proper Neural Network Sizes and Architectures—A Comparative Study. , 8(2), 228–240. doi:10.1109/tii.2012.2187914 , 4(2), 251–257. doi:10.1016/0893-6080(91)90009-T
- [Ker12] Keromytis, Angelos D. (2012). [Lecture Notes in Computer Science] Financial Cryptography and Data Security Volume 7397 || Designing Privacy-Preserving Smart Meters with Low-Cost Microcontrollers. , 10.1007/978-3-642-32946-3(Chapter 18), 239–253. doi:10.1007/978-3-642-32946-3_18
- [KGF14] Khadse, R., Gawai, N., Faruk, B.M.: Overview and comparative study of different microcontrollers. Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET) 2(XII), 311–315 (2014). ISSN 2321–9653
- [KKB+17] El Abidine Kherroubi, Zine; Kermadi, Mostefa; Berkouk, El Madjid; Salam, Zainal (2017). [IEEE 2017 IEEE Conference on Energy Conversion (CENCON) - Kuala Lumpur, Malaysia (2017.10.30-2017.10.31)] 2017 IEEE Conference on Energy Conversion (CENCON) - Real time implementation of space vector pulse width modulation for three level neutral point clamped (NPC) inverter using Arduino DUE board. , (), 259–264. doi:10.1109/CENCON.2017.8262495
- [KS16] Kodali, Ravi Kishore; Mahesh, Kopulwar Shishir (2016). [IEEE 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) - Greater Noida, India (2016.12.14-2016.12.17)] 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) - Low cost ambient monitoring using ESP8266. , (), 779–782. doi:10.1109/IC3I.2016.7918788 .
- [LMC21] Lambert, J.; Monahan, R.; Casey, K. Power Consumption Profiling of a Lightweight Development Board: Sensing with the INA219 and Teensy 4.0 Microcontroller. Electronics 2021, 10, 775. <https://doi.org/10.3390/electronics10070775>
- [MAS+17] Maier, Alexander; Sharp, Andrew; Vagapov, Yuriy (2017). [IEEE 2017 Internet Technologies and Applications (ITA) - Wrexham (2017.9.12-2017.9.15)] 2017 Internet Technologies and Applications (ITA) - Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. , (), 143–148. doi:10.1109/ITECHA.2017.8101926

- [MVD+14] Maksimović, M., Vujović, V., Davidović, N., Milošević, V. and Perišić, B., 2014. Raspberry Pi as Internet of things hardware: performances and constraints. design issues, 3(8), pp.1-6.
- [Pan11] Panchal, Gaurang, et al. "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers." International Journal of Computer Theory and Engineering 3.2 (2011): 332-337.
- [Pet22] Linas Petkevičius, 2022 Dirbtinio intelekto, giliojo/mašininio mokymo sąvokų žodynėlis <https://klevas.mif.vu.lt/~linp/page/savokos.html>
- [SSA17] Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. towards data science, 6(12), pp.310-316.