

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD br. 2

NEURONSKE MREŽE S RANIM IZLAZOM U SVIJETU UGRADBENIH RAČUNALA

Vjeran Cvitanić

Zagreb, svibanj 2024.

Sažetak

U ovom radu obrađena je tema metode ranog izlaza kod neuronskih mreža te je objašnjena motivacija za njenu implementaciju te primjena u sustavima ograničenih resursa. Ukratko je predstavljen dosadašnji razvoj IoT i AI tehnologija s fokusom na njihovu zajedničku primjenu. Objašnjene su arhitekture implementacije AI algoritama u distribuiranim sustavima te osnovni problemi kod implementacije AI algoritama kod računalnih sustava ograničenih performansi. Kao rješenje tih problema predstavljene su neuronske mreže s ranim izlazom, koje nude pametniju raspodjelu posla te izbjegavaju nepotrebno korištenje mreže. Zaključno, dano je nekoliko primjera primjene ovih tehnologija.

Sadržaj

1.	Uvod.....	1
2.	Pregled primjene umjetne inteligencije u ugradbenim računalima	3
2.1.	Područja primjene.....	3
2.2.	Arhitekture AI u IoT-u	4
3.	Neuronske mreže s ranim izlazom	7
3.1.	Neuronske mreže	7
3.2.	Rani izlazi iz neuronske mreže	8
3.3.	Prednosti ranog izlaza	10
3.4.	Treniranje modela s ranim izlazom.....	11
4.	Moguće primjene neuronskih mreža s ranim izlazom u ugradbenim računalnim sustavima	13
5.	Zaključak.....	15
6.	Literatura.....	16

Popis oznaka i kratica

IoT	Internet stvari (engl. <i>Internet of things</i>)
AI	Umjetna inteligencija (engl. <i>Artificial intelligence</i>)
NN	Neuronska mreža (engl. <i>Neural network</i>)
EE	Rani izlaz (engl. <i>Early exit</i>)

Popis slika

Slika 1 - Arhitekture AI algoritama na raspodijeljenim računalima [2]	6
Slika 2 - Arhitektura neuronske mreže [10]	7
Slika 3 - Proces učenja neuronske mreže [11]	8
Slika 4 - Primjer neuronske mreže s dva rana izlaza [3]	9
Slika 5 - Primjeri raspodijeljenih sustava s ranim izlazima [4]	10
Slika 6 - Primjer raspodijeljenog sustava s ranim izlazima [4]	10
Slika 7 - Zajedničko učenje izlaza [6]	12
Slika 8 - Učenje izlaza zasebno [6]	12

1. Uvod

Posljednjih godina u središtu pozornosti razvoj je *Internet of things* (IoT) tehnologija. Pod pojmom IoT-a smatra se skup umreženih i samostalnih elektroničkih uređaja, koji su sposobni prikupljati podatke iz okoline, međusobno komunicirati te prilagođavati svoje djelovanje na temelju prikupljenih podataka. Ova tehnologija omogućuje distribuirano prikupljanje podataka te razmjenu i pohranu tih podataka među čvorovima u mreži, što omogućuje razvoj sustava većeg stupnja samostalnosti od dotadašnjih.

Rubni IoT senzorski uređaji, krajnji uređaji u mreži koji prikupljaju podatke iz okoline, uz jednostavne mikroprocesore najčešće sadrže senzore, aktuatore te module za komunikaciju. Zahtjevi za energetsom i vremenskom efikasnošću te malim dimenzijama rubnih IoT uređaja uglavnom rezultiraju razvojem uređaja limitiranih računalnih performansi. Pojedini rubni uređaj stoga često nije sposoban samostalno donositi odluke na temelju prikupljenih podataka pa je ideja rada IoT sustava da rubni uređaji prikupljene podatke šalju prema računalno moćnijim, centraliziranim čvorovima u mreži. Centralizirani čvorovi zatim obrađuju primljene podatke te odrađuju akcije na temelju informacija dobivenih iz podataka.

Osim IoT tehnologija, zadnjih desetak godina obilježeno je ubrzanim napretkom u razvoju sustava umjetne inteligencije (engl. *Artificial intelligence*, AI). Algoritmi umjetne inteligencije koncipirani su na način da se algoritam uči na temelju ulaznih podataka te tako s prikupljanjem podataka poboljšava svoje znanje.

Kako je zadaća rubnih IoT uređaja prikupljanje podataka iz okoline, obrada istih te vršenje akcija na temelju stečenog znanja, a glavna snaga AI algoritama učenje i donošenje odluka na osnovu velikog skupa podataka, poželjno je iskoristiti moć umjetne inteligencije u IoT-u. Osnovni problem implementacije složenijih algoritama umjetne inteligencije na rubnim IoT uređajima jesu već spomenute slabe performanse i nedostatak radne memorije. Jedno od mogućih rješenja je implementacija neuronskih mreža s

ranim izlazom, koje se temelje na kompromisu između preciznosti algoritma te brzine donošenja odluka.

U ovom radu obrađena je tema metode ranog izlaza kod neuronskih mreža te je objašnjena njena primjena u sustavima ograničenih resursa. U drugom poglavlju dan je pregled trenutne primjene algoritama umjetne inteligencije u ugradbenim računalima. U trećoj cjelini objašnjene su neuronske mreže te je detaljno obrađena metoda ranog izlaza, a u četvrtoj cjelini dan je pregled njenih mogućih primjena. Zaključno poglavlje sadrži kratak osvrt na temu.

2. Pregled primjene umjetne inteligencije u ugradbenim računalima

2.1. Područja primjene

Razvoj osobnih računala došao je do faze gdje su računala sveprisutna u svakodnevnicima. Susrećemo se sa sve manjim procesorima, najčešće specifične namjene poput pametnih ručnih satova, sustava pametne kuće, sustava za nadzor i kontrolu cestovnog prometa i slično. S obzirom na specifičnu funkcionalnost te konstantno prikupljanje relativno velike količine podataka, ovakvi uređaji odlična su podloga za implementaciju sustava umjetne inteligencije.

Medicina je područje koje se sve više oslanja na ugradbena računala, od sustava za nadzor pacijenata, tretiranja pacijenata (npr. *pacemaker*) do sustava za kiruršku operaciju. Zahtjevi kao što su male dimenzije uređaja, iznimna brzina odziva (*hard real time* sustavi) te besmrtnost uređaja idealni su za implementaciju ugradbenih računala. Krucijalnost brzine prikupljanja i obrade podataka u važnom području poput medicine uvelike su doprinijeli bržem razvoju kvalitetnih ugradbenih računala. Također, ogroman broj parametara koji se koriste, na primjer, u procesu nadzora pacijenata mogu se pohranjivati te koristiti kao ulazni skup podataka za učenje algoritama umjetne inteligencije. Učenjem algoritama na većem skupu podataka povećava se vjerojatnost postizanja točnije klasifikacije u budućim slučajevima procjene stanja pacijenata.

Smart city popularan je koncept koji se temelji na rasprostranjenoj IoT tehnologiji. Podrazumijeva povezivanje i komunikaciju distribuiranih rubnih uređaja koji služe kao senzori i aktuatori. Cilj je povećati autonomnost procesa koji se događaju svakodnevno u urbanom središtu, a najbolji primjer je cestovni promet. Svakodnevne prometne nesreće, zastoji i radovi na prometnicama uzimaju sudionicima prometa nezanemarivo mnogo vremena. Pomoću IoT tehnologija, moguće je poboljšati prikupljanje i dijeljenje informacija o stanju na prometnicama te tako sudionicima prometa u stvarnom vremenu sugerirati učinkovitije rute do odredišta. Izbjegavanje sudara i sličnih

nesreća moguće je korištenjem *hard real time* računalnih sustava, čija je osnovna karakteristika minimalno kašnjenje odziva, a to je najizvedivije upravo implementacijom raspodijeljenih računala. Ideja je da su računala što bliže izvoru podataka, tj. senzorima, kako bi mogli što brže reagirati. Također, analizom automatski prikupljenih podataka bilo bi moguće učiti o navikama sudionika u prometu te na temelju tih podataka prilagoditi javni prijevoz potrebama korisnika i iskorištenosti prometnica.

IoT je svoju primjenu brzo našao u poljoprivrednim djelatnostima. Kvalitetno praćenje uvjeta na velikim poljoprivrednim površinama zahtijeva postavljanje senzorskih uređaja ravnomjerno po cijeloj površini. Središnje računalo prikuplja te podatke te na osnovu njih donosi zaključke o stanju na polju, na osnovu čega informira poljoprivrednika ili čak samostalno šalje naredbe aktuatorima. Sustavi za nadzor i praćenje uvjeta u kojima se biljke uzgajaju, poput temperature, vlage, tlaka i slično, jednostavno su implementirani IoT rješenjima [5].

2.2. Arhitekture AI u IoT-u

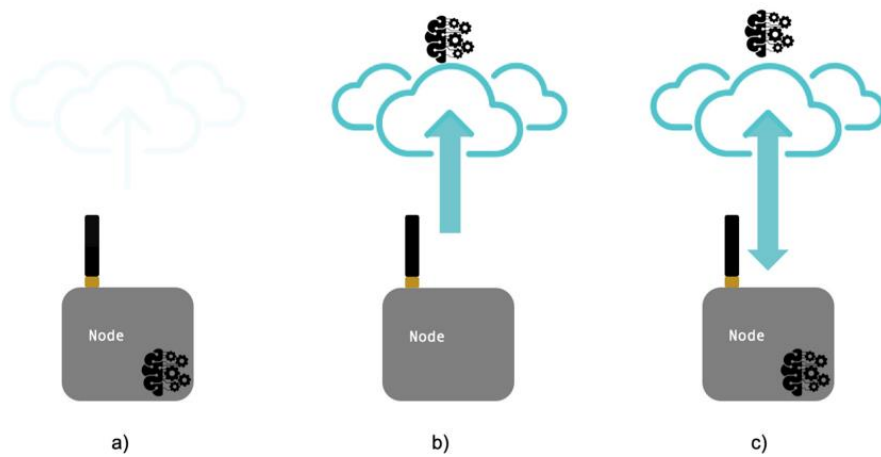
Kod implementacije složenih AI algoritama u IoT sustavima potrebno je oprezno planirati dizajn sustava zbog ograničenih resursa rubnih uređaja. Prilikom razvoja sustava umjetne inteligencije za ugradbena računala postoji potreba za postizanjem kompromisa [1]. Problem je što je, kako bi algoritam umjetne inteligencije bio učinkovit i kvalitetan, potrebno osigurati dovoljnu količinu radne memorije za pohranu podataka i modela. No glavno svojstvo ugradbenih računala je relativno mala količina dostupne radne memorije. Rubni uređaji najčešće su sposobni upogoniti jednostavnije modele umjetne inteligencije. Veliki i složeni modeli zauzimaju više memorijskog prostora zbog većeg broja parametara te zahtijevaju i veću energetska potrošnju od jednostavnijih modela [2]. Zbog toga, odabir arhitekture implementacije AI algoritama na IoT uređajima svodi se na jednu od tri prikazane na slici 1.

Prvi primjer na slici prikazuje slučaj kada se algoritam nalazi na samom rubnom uređaju, tj. svo računanje se događa lokalno. Prednost ovakvog pristupa je samostalnost uređaja te sigurnost podataka koji ne napuštaju

uređaj koji ih prikuplja. Nedostatak je očito korištenje algoritma prilagođenih performansi, najčešće neki jednostavan algoritam strojnog učenja s malim brojem parametara. Implementacija algoritama dubokog učenja moguća je uz određene kompromise na performanse modela, poput obrezivanja ponekih veza među neuronima, čime se smanjuje broj parametara, ili korištenjem manje bitova za zapis težina modela, npr. zapis težina u osmobitnom *integer* formatu umjesto 32 bitnog *float* formata. Neovisno o odabiru, neizbježan je gubitak kvalitete algoritma.

Drugi primjer na slici prikazuje scenarij u kojem rubni uređaj prikupljene podatke šalje putem mreže do moćnijeg računala u oblaku. Prednost ovakvog pristupa je neovisnost složenosti modela o računalnim mogućnostima rubnog uređaja. No komunikacija putem mreže podrazumijeva povećano kašnjenje, ranjivost privatnosti podataka te ovisnost rada sustava o mreži i udaljenom računalu u oblaku.

Treći pristup kombinira dva prethodna pristupa, tako da i rubni uređaj i računalno u oblaku imaju pristup modelu. Ideja je da rubni uređaj sadrži jednostavniji model, a računalno u oblaku puni model. U slučaju nestabilne veze ili pada rada oblaka, rubni uređaj može sam na osnovu jednostavnijeg modela donijeti odluku. Ovakva arhitektura, na kojoj se temelje neuronske mreže s ranim izlazom (engl. *early exit neural network*), rješava problem reduciranih performansi ukupnog modela, a istovremeno povećava samostalnost uređaja u slučajevima kada je to potrebno.



Slika 1 - Arhitekture AI algoritama na raspodijeljenim računalima [2]

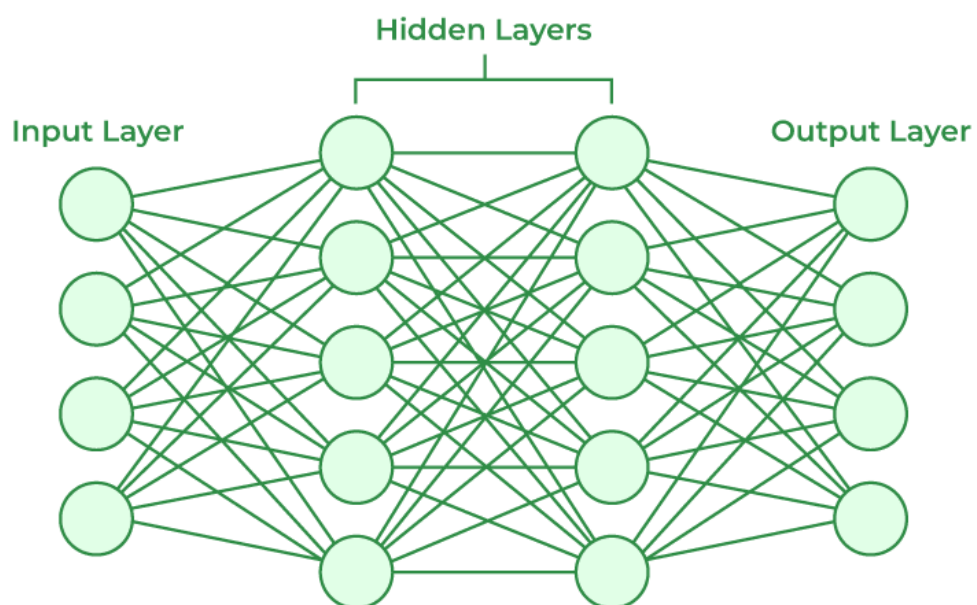
Odabir jedne od tri predstavljene arhitekture ovisi o primjeni te mogućnostima korištenih uređaja.

3. Neuronske mreže s ranim izlazom

3.1. Neuronske mreže

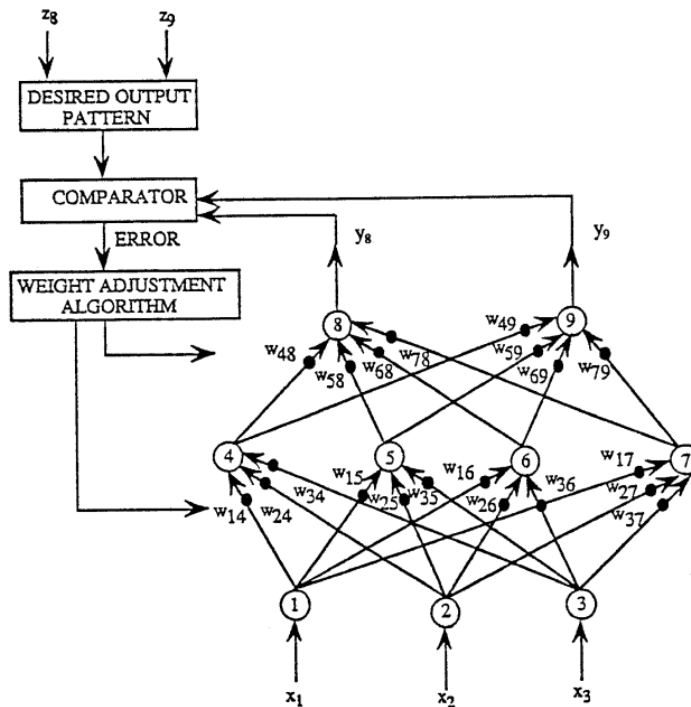
Metoda ranog izlaza temelji se na slojevitosti neuronskih mreža. Algoritam neuronske mreže vrsta je algoritma dubokog učenja koji simulira rad ljudskog mozga povezivanjem neurona u slojevitu mrežu [8]. Mreža se sastoji od tri vrste slojeva: ulazni sloj, izlazni sloj te proizvoljan broj skrivenih slojeva. Ulazni neuroni primaju input, dok izlazni neuroni predstavljaju output modela. Skriveni slojevi su povezani samo sa neposredno prethodnim i sljedećim slojem, dok neuroni koji su dio istog sloja međusobno nisu povezani. Svaki neuron jednog sloja povezan je sa svakim neuronom sljedećeg sloja, gdje je veza opisana parametrima težine i pristranosti. Aktivacija neurona u jednom sloju utječe na aktivaciju neurona u sljedećem sloju i to proporcionalno pridruženom parametru težine. Neuron se aktivira ako je suma svih težina i pristranosti veća od predefinirane granice.

Na slici 2 prikazan je model neuronske mreže sa dva skrivena sloja.



Slika 2 - Arhitektura neuronske mreže [10]

Proces treniranja modela, prikazan na slici 3, sastoji se od podjele skupa podataka na skup za treniranje i provjeru, dovođenje podataka iz skupa za treniranje na ulaz te usporedbe izlaza modela sa stvarnim pridijeljenim izlazom podataka. Zatim se na osnovu greške usporedbe ažuriraju vrijednosti neurona. Pri završetku treniranja modela, vrši se provjera sa skupom za provjeru, koji mora biti reprezentativan uzorak, čime se ugrubo saznaje točnost modela.



Slika 3 - Proces učenja neuronske mreže [11]

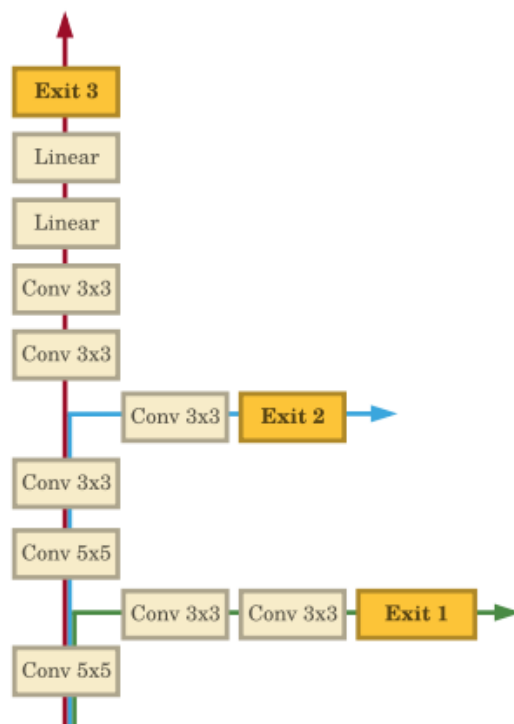
3.2. Rani izlazi iz neuronske mreže

Implementacija složenije neuronske mreže nije praktična na rubnom uređaju zbog već navedenih karakteristika takvih uređaja. No koristeći treći primjer sa slike (referenca, od prije), gdje je jednostavniji model na rubnom uređaju, a složeniji u oblaku, moguće je prilagoditi implementaciju neuronske mreže. Ideja je da se podijeli neuronsku mrežu po slojevima, tako da ulazni sloj i nekoliko skrivenih slojeva budu na rubnom uređaju, dok bi računalo u oblaku implementiralo preostale skrivene slojeve te izlazni sloj. Tada rubni

uređaj može izračunati rezultat prvih nekoliko slojeva te ga proslijediti računalu u oblaku koje bi zatim završilo izračun koristeći svoje slojeve.

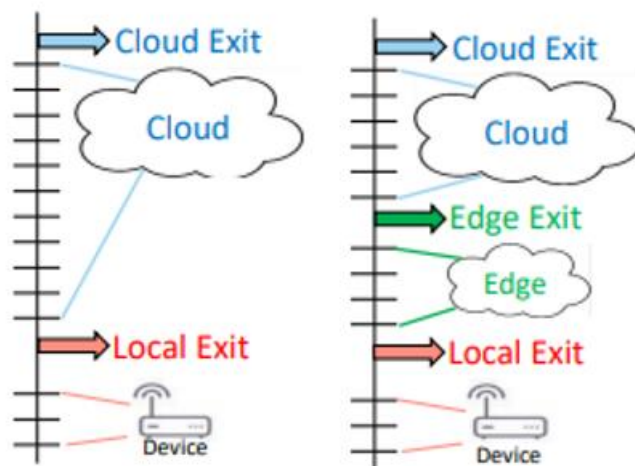
Ideja kod mreža s ranim izlazom je da tada rubni uređaj može samostalno, na osnovu prvih nekoliko slojeva vršiti klasifikaciju za jednostavne primjere, a koristiti pomoć od oblaka samo za one primjere u čiju klasifikaciju nije dovoljno siguran. Na svakom sloju koji je predviđen za prijevremeni izlaz definira se prag. Ako vrijednost izlaza na tom sloju prijeđe vrijednost praga algoritam prestaje računati te koristi prijevremeni izlaz kao konačno rješenje. Prijevremeni izlaz ne garantira uvijek optimalan izlaz modela, ali smanjuje vrijeme potrebno za klasifikaciju i zauzeće mreže za jednostavnije primjere.

Slikovit prikaz podjele neuronske mreže po slojevima, sa dva prijevremena izlaza, dan je na slici 4 [3].



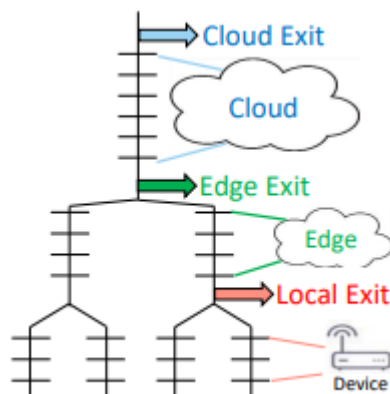
Slika 4 - Primjer neuronske mreže s dva rana izlaza [3]

Na lijevom dijelu slike 5 prikazana je arhitektura predloženog IoT sustava sa ranim izlazima, dok je na desnom prikazana arhitektura sa dodanim *edge* izlazom (uređaj u magli, engl. *fog*), koji predstavlja svojevrsan most između senzorskog uređaja u oblaka.



Slika 5 - Primjeri raspodijeljenih sustava s ranim izlazima [4]

Uređaj u magli preuzme središnji dio neuronske mreže pa se tako dodatno smanjuju zahtjevi nad rubnim uređajem i nad oblakom. Na slici 6 prikazana je ideja iza dodavanja uređaja u magli, gdje se pomoću njega povezuje više senzorskih uređaja u jednu lokalnu mrežu. Više uređaja u magli moguće je potom povezati u mrežu kako bi jedan oblak upravljao cijelom mrežom. U takvoj hijerarhiji do oblaka bi dolazili samo najsloženiji primjeri, dok bi većinu primjera koje rubni uređaj ne uspijeva klasificirati preuzeo uređaj u magli.



Slika 6 - Primjer raspodijeljenog sustava s ranim izlazima [4]

3.3. Prednosti ranog izlaza

Danim modelom podjele neuronske mreže po slojevima dolazi se do inferencije potpunog modela uz zadovoljavanje ranije navedenih zahtjeva nad

resursima i sigurnosti sustava. Osim što se ovom metodom zaobilazi problem nedovoljnih resursa rubnog uređaja, također se smanjuje sigurnosni rizik nad podacima, jer se preko mreže ne šalju čisti podaci već rezultat nižih slojeva mreže.

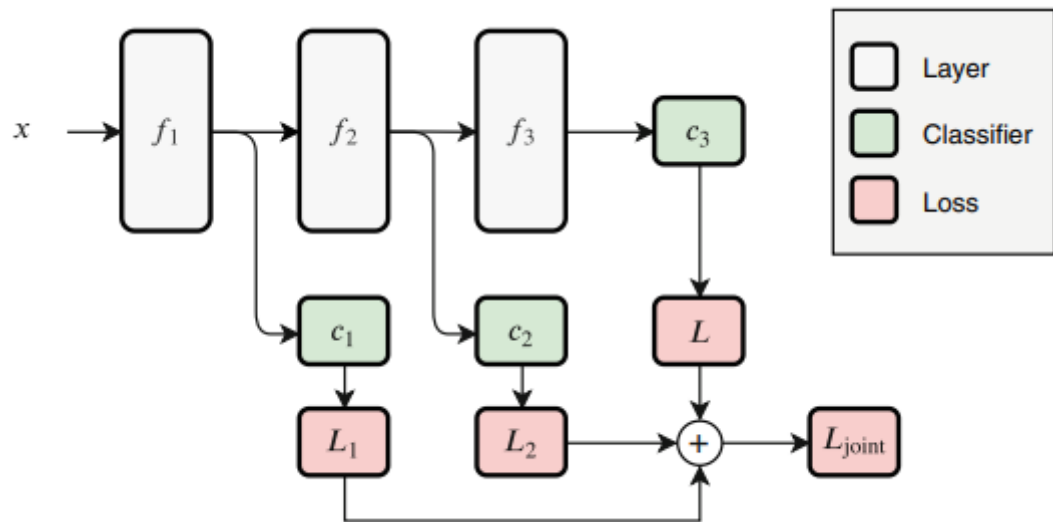
Implementacija ranog izlaza, osim što razrješava rubni uređaj dijela teškog posla, još i smanjuje prosječno vrijeme klasifikacije te sprječava negativne efekte prenaučenosti modela [6]. Ranijim izlazom iz mreže kod klasifikacije jednostavnih primjera brže se dolazi do inferencije modela te rubni uređaj niti ne traži od računala u oblaku potrebne resurse. Također, model je i dalje podložan prenaučenosti, pogotovo složeniji dio mreže, no ranim izlaskom iz mreže kod jednostavnijih primjera eliminira se utjecaj složenijeg dijela mreže, čime se smanjuje vjerojatnost pogrešne inferencije zbog prenaučenosti.

Da bi se maksimizirale prednosti metode s ranim izlazom potrebno je promisliti gdje postaviti izlaze. U slučaju korištenja modela na jednom računalu, ako je cilj modela spriječiti prenaučenost tada su već 2 izlaza postavljena na trećinama mreže dovoljna. Ako je cilj efikasnost računanja i energetske potrošnje tada je ovo kombinatorni problem [6]. U slučaju korištenja mreže u distribuiranom sustavu, izlazi su određeni brojem i računalnim sposobnostima rubnog uređaja i uređaja u magli.

3.4. Treniranje modela s ranim izlazom

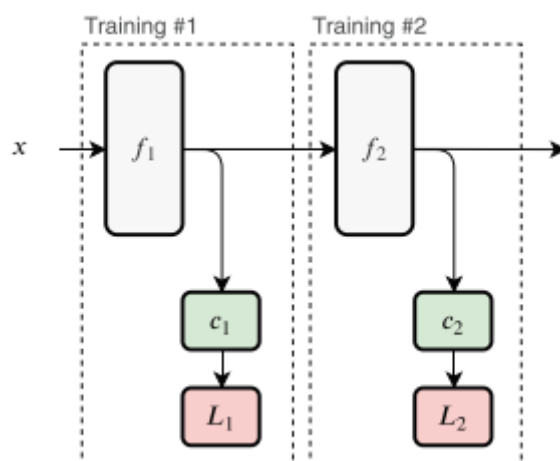
Treniranje mreže s ranim izlazima razlikuje se od treniranja mreže s jednim izlazom po tome što je potrebno istrenirati više izlaznih klasifikatora. Kako svaki rani izlaz daje svoju klasifikaciju na osnovu obrade značajki na slojevima koji prethodne tom sloju, tako će često pojedini izlazi davati različite klasifikacije za iste ulazne primjere. Prilikom treniranja potrebno je računati funkciju gubitka, koja govori koliko se predviđanja modela razlikuju od stvarnih vrijednosti. Kada bi se za funkciju gubitka koristili samo rezultati krajnjeg izlaza modela, rani izlazi ne bi bili kvalitetno trenirani. Stoga svaki rani izlaz računa svoju funkciju gubitka, na osnovu koje popravljaju svoja predviđanja.

Optimizaciju parametara mreže s ranim izlazom moguće je raditi na razini cijelog modela i na razini pojedinog sloja. Na slici 7 prikazana je shema učenja potpunog modela na temelju sume pogrešaka svih izlaza. Ovim postupkom model želi minimizirati kolika je ukupna pogreška svih izlaza te uči i koliko bi trebao vrednovati vjerodostojnost pojedinog izlaza.



Slika 7 - Zajedničko učenje izlaza [6]

Alternativa, prikazana na slici 8, je treniranje svakog izlaza pojedinačno, što pomaže izbjegavanju problema kod regularizacije modela. Ovim pristupom pojedini rani izlazi bolje uče na osnovu dotada obrađenih značajki.



Slika 8 - Učenje izlaza zasebno [6]

4. Moguće primjene neuronskih mreža s ranim izlazom u ugradbenim računalnim sustavima

Neuronske mreže s ranim izlazom, iako korisne i kod nedistribuiranih računalnih sustava, primarno se koriste u distribuiranim sustavima, s naglaskom na heterogene sustave s jasnom hijerarhijom. Potreba za hijerarhijom proizlazi iz arhitekture mreže s ranim izlazima, gdje su poželjne karakteristike središnjeg uređaja visoke performanse i mogućnost obrade veće količine podataka, dok se od rubnih uređaja očekuje blizina izvoru podataka te mali odziv. U nastavku je dano nekoliko primjera uporabe ovakvih neuronskih mreža u IoT okolini.

Kod IoT sustava koji se sastoje od velikog broja rubnih uređaja moguće je da jedno središnje računalo nije sposobno istovremeno primiti i obrađivati podatke sa svih uređaja. Kako bi se izbjegla potreba za korištenjem više središnjih računala, a zadržale performanse, implementacijom neuronske mreže s ranim izlazom smanjuju se zahtjevi nad sustavom. Primjeri ovakvih sustava su složeni sustavi s izrazitim zahtjevima nad brzinom sustava, poput autonomnih vozila koja si ne mogu priuštiti zastoje prilikom obrade podataka u centralnom računalu [9]. Mnogo bolje rješenje od jednog prezaposlenog računala je implementacija više rubnih uređaja, koji su u blizini senzora, čija je zadaća primiti podatke sa senzora te procijeniti mogu li sami donijeti odluku na temelju podataka ili moraju zatražiti pomoć od moćnijeg računala. Kod primjera autonomnih automobila, brzina obrade nekih podataka je od ključne važnosti za ispravan rad sustava. Slanje previše takvih podataka izravno na središnje računalo potencijalno bi dovelo do zastoja, zbog čega pati ispravnost sustava.

Primjer IoT sustava koji je pogodan za implementaciju neuronske mreže s ranim izlazom je sustav kamera za nadzor i prepoznavanje sumnjivih aktivnosti. Implementacija više uređaja s kamerom koji svaki zasebno analizira svoju sliku ima ili negativne posljedice na brzinu odziva sustava ili zahtijeva nepotrebno moćno sklopovlje na svakom rubu. Dodavanje nekoliko središnjih računala u oblaku uz korištenje jednostavnijeg sklopovlja na pojedinim

kamerama i implementacija mreže s ranim izlazom i efikasnijom raspodjelom posla vjerojatno će imati pozitivan utjecaj na odziv sustava te trošak sklopovlja.

Zanimljiv koncept primjene mreža s ranim izlazom dan je u radu [7]. Autori su primijetili da je kod obrade videozapisa moguće primijeniti metodu ranog izlaza iz mreže prilikom obrade svakog pojedinog kadra, ali i u vremenskoj domeni. Naime, ideja je imati dva modela, jedan koji kao input uzima pojedini kadar iz videozapisa, obrađuje ga te na izlazu daje informacije o tom pojedinačnom kadru. Ovaj model bi se koristio ranim izlazom na razini sloja u mreži. Drugi model na temelju informacija iz dotad obrađenih kadrova na izlazu daje informaciju o videozapisu. Rani izlaz se ovdje koristi u vremenskoj domeni, jer bi drugi model mogao na temelju prvih nekoliko kadrova prepoznati o čemu se radi u videozapisu te bi tako brže došao do inferencije.

5. Zaključak

Algoritmi umjetne inteligencije sposobni su obraditi i učiti se na temelju velike količine podataka. Stoga su vrijedan resurs u IoT sustavima, distribuiranim sustavima za prikupljanje i obradu podataka te aktuaciju na osnovu prikupljenih informacija. Neizostavna komponenta svakog IoT sustava su rubni uređaji, računala smanjenih dimenzija i performansi čija je osnovna uloga prikupljanje podataka iz okoline putem senzora. Ograničene performanse ovakvih uređaja otežavaju implementaciju složenijih algoritama umjetne inteligencije na njima pa je potrebno iskoristiti računalnu moć središnjeg računala u sustavu. Prebacivanjem cijelog modela na središnje računalo sustav se u potpunosti oslanja na ispravan rad i besmrtnost komunikacijske mreže te središnjeg računala te se riskira sigurnost podataka. Rješenje koje zadovoljava i implementaciju modela zadovoljavajućih performansi i izbjegava postojanje takozvanog *single point of failurea* koristi se modelom neuronske mreže čiji su niži slojevi na rubnom uređaju, a viši na središnjem. Rezultati obrade podataka na nižim slojevima šalju se središnjem računalu na daljnju obradu. Prilikom obrade na rubnom uređaju moguće je da izlaz nižih slojeva klasificira ulaz sa zadovoljavajućom vjerojatnošću, nakon čega nije ni potrebno tražiti računanje od ostatka mreže te se odmah ide na klasifikaciju ulaza. Ova metoda se naziva metodom ranog izlaza iz mreže, a prednosti su smanjeno vrijeme računanja te sprječavanja utjecaja prenaučivosti modela. Primjene metode ranog izlaza pogodne su u raznim IoT sustavima koji imaju velike zahtjeve nad središnjim računalom te kod sustava gdje je prioritet na brzom donošenju odluka.

6. Literatura

1. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7979979>
2. <https://www.mdpi.com/1424-8220/20/9/2533>
3. <https://ieeexplore.ieee.org/abstract/document/7900006>
4. <https://ieeexplore.ieee.org/abstract/document/7979979>
5. <https://www.mdpi.com/2079-9292/9/2/319>
6. <https://link.springer.com/article/10.1007/s12559-020-09734-4>
7. <https://ieeexplore.ieee.org/document/10176276>
8. <https://www.ibm.com/topics/neural-networks>
9. <https://www.mdpi.com/1424-8220/23/4/1963>
10. <https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/>
11. <https://ieeexplore.ieee.org/document/483329>