

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jani Bizjak

**Analiza signalov EKG z globokimi
nevronskimi mrežami**

MAGISTRSKO DELO

MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: prof. dr. Igor Kononenko

Ljubljana, 2015

AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za matematiko in fiziko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za matematiko in fiziko ter mentorja.

©2015 JANI BIZJAK

PROGRAM DELA

Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

V magistrski nalogi preučite uporabnost globokih nevronske mreže pri analizi signalov EKG. Določite ustrezno arhitekturo nevronske mreže, ki bo sposobna prepoznati srčni ritem ter najpogostejše anomalije v signalu EKG, ki so posledica srčnih obolenj. Rezultate vaše mreže primerjajte z ostalimi state-of-the-art metodami, ki se uporabljajo za analizo signala EKG. Pri tem upoštevajte naslednje kriterije: točnost, senzitivnost, specifičnost, ploščino pod krivuljo ROC (AUC), hitrost algoritma, potrebo po pred-procesiranju in naknadnem procesiranju podatkov.

V Ljubljani, August 27, 2015

Podpis mentorja:

IZJAVA O AVTORSTVU MAGISTRSKEGA DELA

Spodaj podpisani Jani Bizjak sem avtor magistrskega dela z naslovom:

Analiza signalov EKG z globokimi nevronskimi mrežami

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom prof. dr. Igorja Kononenka,
- so elektronska oblika magistrskega dela, naslov (slovenski, angleški), povzetek (slovenski, angleški) ter ključne besede (slovenske, angleške) identični s tiskano obliko magistrskega dela,
- Fakulteti za matematiko in fiziko Univerze v Ljubljani dovoljujem objavo elektronske oblike svojega dela na spletnih straneh.

V Ljubljani, 27. avgust 2015

Podpis avtorja:

ZAHVALA

Profesorju dr. Igorju Kononenku se zahvaljujem za izzive, pomoč ter nasvete, brez katerih naloge ne bi opraviti tako uspešno kot sem jo.

Zahvaljujem se članom Odseka za inteligentne sisteme Instituta "Jožef Stefan", ki so mi z nasveti in opremo omogočili nemoteno in uspešno izvajanje eksperimentalnega dela naloge.

Za konec bi se rad zahvalil družini in ostalim, ki so me tekom študija spodbujali in motivirali.

Jani Bizjak, 2015

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

— Alan Turing

Kazalo

Povzetek	i
Abstract	iii
1 Uvod	1
2 Prepoznavanje srčnih bolezni	3
2.1 Delovanje srca	3
2.2 Elektrokardiogram - EKG	9
2.3 Srčne bolezni	11
3 Globoke nevronske mreže	23
3.1 Uvod v nevronske mreže	24
3.2 Konvolucijske nevronske mreže	34
3.3 Povratne nevronske mreže	38
4 Analiza signala EKG z globokimi nevronskimi mrežami	41
4.1 Učna množica za učenje in testiranje	41
4.2 Konvolucijske nevronske mreže	42
4.3 Povratne nevronske mreže	60
5 Zaključek	65
5.1 Nadaljnje delo	66
A Statistični podatki za vse datoteke	75

Povzetek

S hitrim razvojem računalniških zmogljivosti se strojno učenje hitro vključuje in že uporablja v vsakdanjem življenju. S povečanjem zanesljivosti in robustnosti se strojno učenje začne uporabljati tudi na občutljivih področjih, kot je medicina. Računalniški sistemi tako prevzemajo stalno spremljanje vitalnih znakov bolnika in ob relevantnih spremembah le-teh, o tem obveščajo medicinsko osebje. V magistrski nalogi sta predstavljena dva tipa globokih nevronske mreže, ki iz izpisa signala elektrokardiograma (EKG) prepoznavata anomalije, ki so posledica srčnih obolenj. V prvem delu naloge so predstavljene najpogostejše bolezni srca, ki se prepoznava iz signalov EKG. Sledi predstavitev metod strojnega učenja s poudarkom na konvolucijskih ter povratnih nevronske mreže. V zadnjem sklopu sledi opis rezultatov, kjer mreže dosežeta 99 odstotno klasifikacijsko točnost.

Ključne besede

konvolucijske nevronske mreže, povratne nevronske mreže, elektrokardiografija, levokračni blok, desnokračni blok, ventrikularna ekstrasistola, QRS, globoke nevronske mreže

Abstract

With rapid increase in computational power, machine learning methods are making huge strides into solving common societal problems. Even in medicine we recently see adaptation of computer systems designed to help medical staff with diagnosis and healthcare support, alerting them on possible risks in patient's vital signs. In this thesis two deep learning architectures are shown to be able to detect and analyse ECG signals for anomalies caused by heart diseases. First part of the thesis is dedicated to explanation of symptoms for the addressed heart diseases from ECG signal. In second part an introduction to deep neural networks is made, featuring both convolutional and recurrent neural networks. The last part is dedicated to extended analysis of networks performances, where best results show accuracy of over 99 percent.

Keywords

convolutional neural network, recurrent neural networks, electrocardiogram, left bundle branch block, right bundle branch block, premature ventricular contraction, QRS, deep neural network

Poglavje 1

Uvod

V zadnjih dvajsetih letih se je namenskost računalnikov spremenila od navadnega orodja za računanje do multimedijskega centra, namenjenega zabavi ter sproščanju. Zaradi potreb po predvajanju visoko-ločljivostnih videoposnetkov, vizualno zahtevnih računalniških iger, izdelave računalniško ustvarjenih filmov (CGI) in še veliko več, so izjemen razvoj doživele grafične kartice, katerih namen je preračunavanje velikih količin podatkov, ki jih je potrebno prikazati na zaslonu. Da to lahko poteka v realnem času, so grafične kartice sestavljene iz številnih procesorskih jeder, ki so sposobne visoke vzporednosti pri računanju.

Razvoj grafičnih kartic ni ostal neopažen s strani strokovnjakov za strojno učenje. Naenkrat je računalnik lahko obdelal tudi do 1000 krat več podatkov v enakem časovnem obdobju. To je še posebej uporabno pri umetnih nevronskih mrežah, saj so zaradi svoje arhitekture še posebej primerne za paralelizacijo.

Prišlo je do razvoja tako imenovanih globokih nevronskih mrež (glej članke [1], [2], [3]). Gre za nevronske mreže, ki namesto večanja števila nevronov na posamezni plasti, strmijo k večjemu številu plasti od tu tudi ime - globoke nevronske mreže.

V magistrskem delu smo se odločili pregledati dva tipa nevronskih mrež; konvolucijske nevronske mreže ter povratne nevronske mreže. Konvolucijske

nevronske mreže se še posebej izkažejo na področju računalniškega vida, medtem ko so povratne nevronske mreže poglavitne na področju prepoznavanja govora ter naravnega jezika.

Za testiranje metod smo se odločili za domeno srčnih bolezni, ki se prepoznavajo iz signalov EKG. To domeno smo izbrali zaradi dveh razlogov. Bolezni srca in ožilja so eden vodilnih vzrokov smrti. Več kot 30 odstotkov vseh smrti je posledica kardioloških bolezni. Zaradi tega že minimalna izboljšava metode, ki zdravnikom pomaga pri odkrivanju ter zdravljenju kardioloških bolezni, reši veliko življenj. Prav zaradi tega je to področje strojnega učenja še posebej dobro raziskano, kjer že obstajajo metode, ki dosegajo več kot 97 odstotno točnost pri odkrivanju bolezni iz signala EKG (glej članke [4],[5], [6], [7], [8], [9], [10],[11],[12]). To je tudi drugi razlog, zakaj smo se odločili za to domeno. Zaradi pomembnosti področja obstajajo številne podatkovne baze z označenimi primeri bolezni v signalu EKG, ter številni algoritmi za primerjavo klasifikacijske točnosti z našimi.

Cilj te magistrske naloge je implementacija dveh tipov globokih nevronskih mrež ter analiza njihovega delovanja pri odkrivanju srčnih bolezni.

Poglavje 2

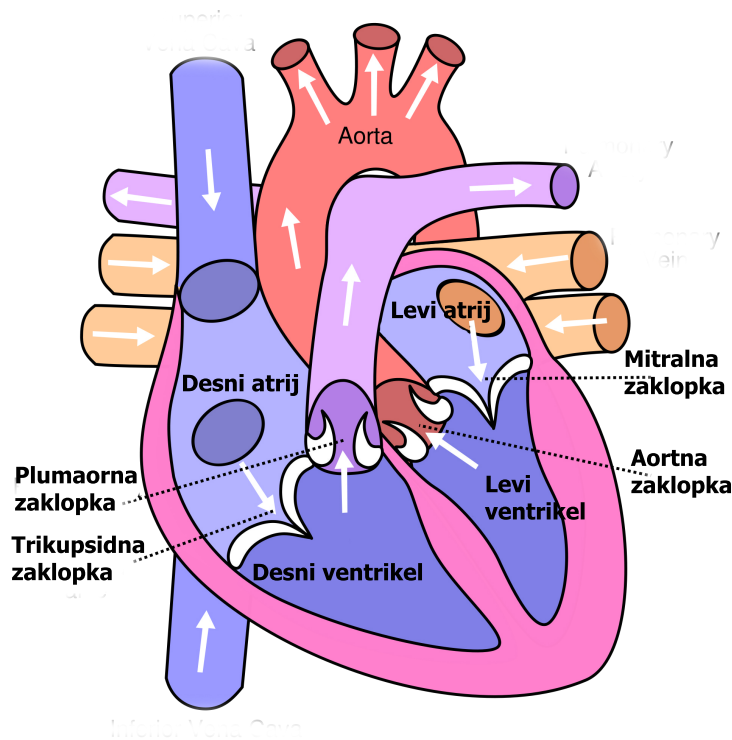
Prepoznavanje srčnih bolezni

Za odkrivanje bolezni na srcu moramo najprej poznati, kako srce deluje. V tem poglavju je predstavljena anatomija srca, tehnike merjenja elektrokardiografije (EKG) ter nekaj najpogostejših bolezni, ki se odkrivajo s pomočjo EKG.

2.1 Delovanje srca

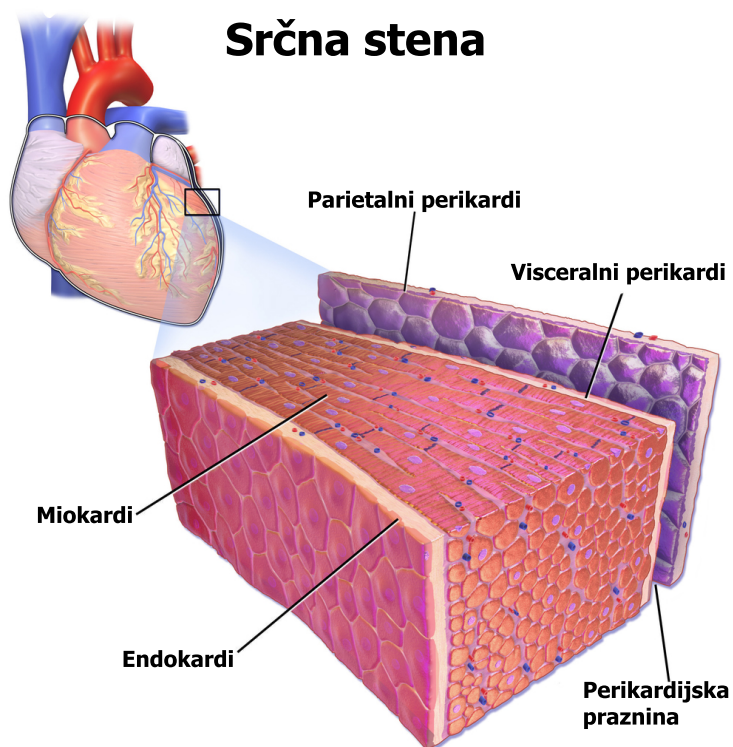
2.1.1 Anatomija

Človeško srce je sestavljeno iz dveh preddvorov (atrija) in dveh prekatov (ventrikla). Kri iz telesa priteče v desni preddvor, ki kri preko desnega prekata pošlje v pljuča. Obogatena kri s kisikom iz pljuč priteče v levi preddvor. Za levim preddvorom se nahaja levi prekat, ki kri skozi arterije črpa (potiska) v telo. Srce vsebuje štiri zaklopke, na levi mitralna ter aortna, na desni pa trikupsidalna ter pljučna. Zaklopke delujejo kot enosmerni ventil ter preprečujejo, da bi kri tekla v napačno smer. Na Sliki 2.1 lahko vidimo, da sta dvorani na desni strani srca večji (po prostornini), kot na levi. Desna polovica srca črpa kri v pljuča, ki so tik ob srcu, leva pa po celem telesu. Ker je količina krvi, ki vstopi v srce enaka količini krvi, ki iz njega izstopi, je kri v levi polovici zaradi manjše prostornine pod višjim tlakom, zaradi česar lahko potuje po celotnem telesu.



Slika 2.1: Frontalni presek človeškega srca. [13]

Zunanja srčna stena je zgrajena iz treh slojev vlaken: endokardi, miokardi in perikardi. Endokardi je notranja plast, ovojnica sestavljena iz žilam podobnih celic. Z zunanjo stranjo je povezana z miokardiom. Miokardi je mišica, ki skrbi, da se srce krči in širi. Je edina mišica v telesu, ki ne potrebuje živčnih dražljajev za svoje delovanje. Zunanjo plast srca sestavlja perikardi, njegova naloga je varovanje srca pred poškodbami ter pričvrstitev srca na ostale organe. Sestavljen je iz dveh plasti, notranje - visceralni perikardi ter zunanje - parietalni perikardi. Med obema plastema je tekočina, ki služi kot lubrikant ter s tem omogoča premikanje srca med bitjem. Na Sliki 2.2 je prikazana zgradba srčne stene.



Slika 2.2: Srčna stena je sestavljena iz treh plasti tkiv: endokardi, miokardi ter perikardi.[14]

2.1.2 Utrip srca

Utrip srca poteka v dveh fazah, krčenje (sistola) ter raztezanje (diastola). V fazi sistole se krčita ventrikla, ki črpata kri v pljuča ter telo. V fazi diastole se ventrikla sprostita, krčita pa se atrija, ki črpata kri v ventrikla.

Električni potencial

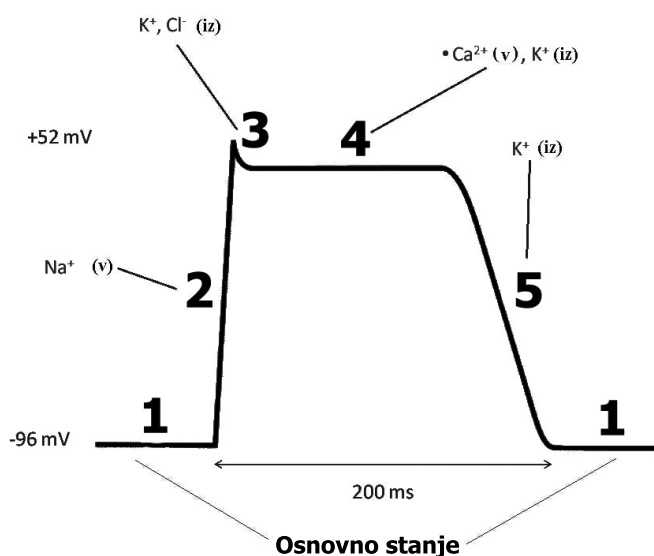
Normalna hitrost delovanja srca, pri odrasli osebi v mirovanju, je med 60 in 100 utripi na minuto. Število utripov v minuti je povezano s tem, kolikšno količino krvi lahko srce pošlje po telesu v enem utripu. Športniki imajo nižji utrip, majhni otroci pa višjega. Ob aktivnostih se število utripov na minuto poveča do 150, v ekstremnih primerih tudi do 200-220 utripov na minuto.

Povišan srčni utrip ni povezan le s fizično aktivnostjo, lahko se poveča tudi zaradi psihičnih (npr. strah) ali bolezenskih dejavnikov.

Električni signal, ki omogoča krčenje srca, se ustvarja v sinoatrijskem vozlu, gre za naravni spodbujevalnik, ki določa frekvenco utripanja srca. Celice v njem, s premikanjem elektrolitov skozi polpropustno membrano (kanale), ustvarjajo akcijski potencial, ki se širi v okolico. Delovanje akcijskega potenciala lahko razdelimo v pet korakov, prikazani so na Sliki 2.3.

1. Prvi fazi pravimo faza mirovanja. Celice so negativno nabite (-90mV). V tej fazi v celice počasi prehajajo ioni K^+ .
2. S hitrim vdorom Na^+ v celico se celica polarizira. Temu pojavu pravimo depolarizacija in se pojavlja spontano. Natrijevi ioni prehajajo skozi posebne prehode v membrani, ki prepuščajo le natrijeve ione. Ta faza je najhitrejša faza v ciklu.
3. Ko celica doseže dovolj visok električni potencial (50mV), se kanali za prehod Na^+ zaprejo. Iz celice začnejo izhajati ioni K^+ in Cl^- .
4. V tretji fazi začnejo v celico vdirati ioni Ca^{2+} . Akcijski potencial ostaja nespremenjen, saj je hitrost vhodnih ionov Ca in izhodnih ionov K enaka.
5. Zadnja faza sledi, ko se kanali za Ca^+ zaprejo, K^+ pa ostanejo odprti. Tej fazi pravimo repolarizacijska faza, saj se začne vzpostavljati osnovni potencial.

Na hitrost bitja srca vpliva avtonomno živčevje skozi simpatične in parasimpatične nevrone. Simpatični živci višajo hitrost utripa, parasimpatični pa ga nižajo. Brez parasimpatičnih živčnih signalov, ki delujejo večino časa, bi bil srčni utrip okrog 100 utripov na minuto. Simpatično živčevje povzroča sproščanje noroadrenalina, ki pospeši repolarizacijsko periodo (odpre kanale za Na in Ca) ter s tem pospeši bitje srca. Podrobneje je delovanje električnega potenciala v srcu opisano v članku [16].



Slika 2.3: Ioni s prehajanjem skozi polprepustno membrano spreminjajo akcijski potencial v celici.[15]

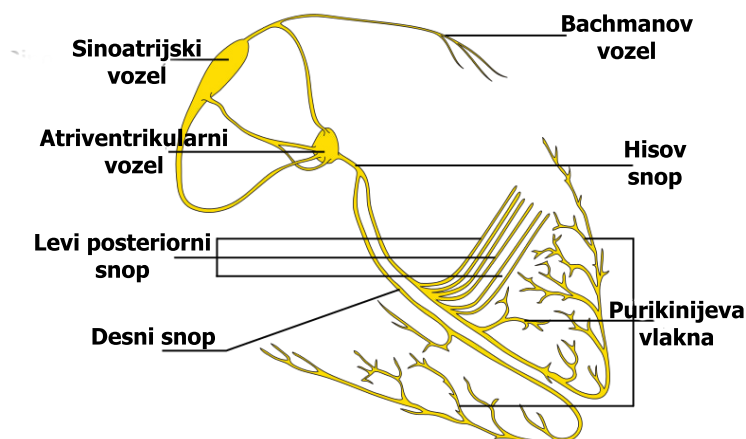
Potovanje živčnega signala po srcu

Normalen (sinusni) srčni utrip se proži v sinoatrijskem vozlu, ki leži v zgornji steni desnega atrija, glej Sliko 2.4. Sinoatrijski vozle je srčni spodbujevalnik, ki določa hitrost utripanja srca. Hitrost utripanja srca merimo v številu utripov na minuto, označujemo ga z BPM¹. Vozel tvori električni pulz, ki povzroči krčenje srčne mišice.

Signal iz sinoatrijskega vozla potuje proti levemu atriju v Bachmanov vozle. To povzroči sočasno krčenje atrijev ter s tem črpanje krvi v ventrikla. Signal se nato zbere v atriventrikularnem vozlu.

Atriventrikularni vozle se nahaja med levim in desnim atrijem. Pod njim je stena, ki ločuje levi in desni ventrikel, imenovana septum. V zgornjem delu septuma se nahaja Hisov snop, ki signal pošlje po levi in desni veji (srčni apeks) v oba ventrikla. Da signal ne bi prehajal iz levega v desni ventrikel, je septum obdan s posebnim srčnim tkivom, ki ne prevaja električnega toka.

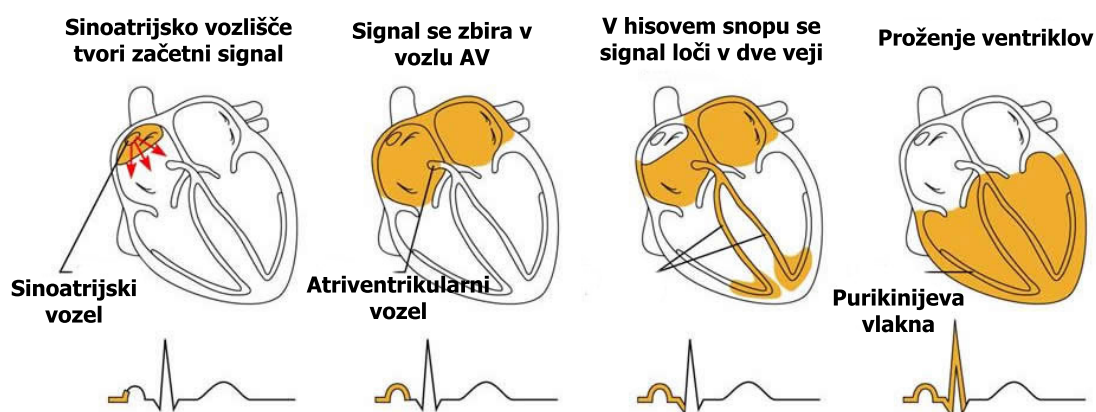
¹ang. Beats Per Minute



Slika 2.4: Prikaz živčevja, v katerem se tvorijo in po katerem potujejo električni signali, ki sprožijo krčenje srčnih mišic. [17]

V ventriklih se nahaja specializirano tkivo, imenovano Purkinijeva vlakna, ki signal prenesejo v srčne mišice. Purkinijeva vlakna so posebna vlakna, ki omogočajo zelo hitro (najhitrejšo) prevajanje električnega signala. Njihova glavna lastnost (prednost) je, da omogočajo sinhronizirano krčenje ventriklov.

Potek potovanja signala je predstavljen na Sliki 2.5.

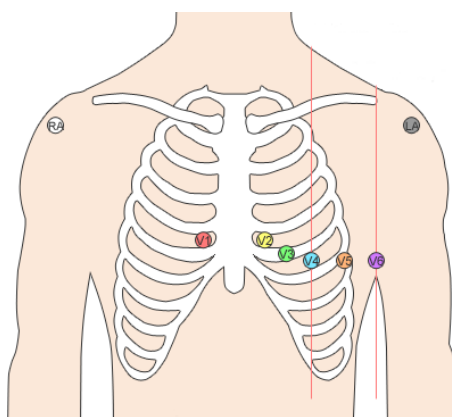


Slika 2.5: Potovanje signala skozi srce.[18]

2.2 Elektrokardiogram - EKG

Delovanje srca lahko opazujemo na več načinov, od poslušanja s stetoskopom, uporabe ultrazvoka, do merjenja električne prevodnosti z elektrokardiografijo (EKG). EKG preko elektrod, ki so pritrjene na kožo, meri električne spremembe na koži, ki nastajajo v fazi polarizacije in depolarizacije ter prehoda signala skozi srce. Meritve EKG izpisuje s standardiziranim izpisom na milimetrski papir s hitrostjo 25mm/s , 1 cm višine na grafu pa ustreza 1mV .

Odvisno od postavitve ter števila elektrod lahko merimo napetost med različnimi deli telesa. Razliki v napetosti med dvema elektrodama pravimo sled ali odvod. Standardna postavitve uporablja 10 elektrod, od tega jih je 6 pritrjenih na prsni koš, ostale pa so pritrjene na roke in noge. Takšna postavitve meri 12 odvodov. Natančen opis različnih postavitve je opisan v knjigi [19]. Postavitve elektrod, ki merijo 12 odvodov, vidimo na Sliki 2.6, kombinacijo za sledi pa preberemo iz Tabele 2.1. Različne sledi nam dajo pogled na srce iz različnih perspektiv. V tabeli so z LR, DR, LN, DN označene okrajšave za levo roko, desno roko, levo nogo ter desno nogo.



Slika 2.6: Postavitve EKG elektrod za merjenje 12 sledi. *Opomba:* Na sliki manjkata elektrodi, ki sta pritrjeni vsaka na svojo nogo. [20]

Tabela 2.1: Kombinacija elektrod pri 12 slednem EKG

Odvod	(-) Elektroda	(+) Elektroda	Pogled na srce
Odvod I	DR	LR	Stranski
Odvod II	DR	LN	Zadnji
Odvod III	LR	LN	Zadnji
aVR	LR + LN	DR	//
aVL	DR + LN	LR	Bočni
aVF	DR + LR	LN	Zadnji
V1			Septalni
V2			Septalni
V3			Sprednji
V4			Sprednji
V5			Bočni
V6			Bočni

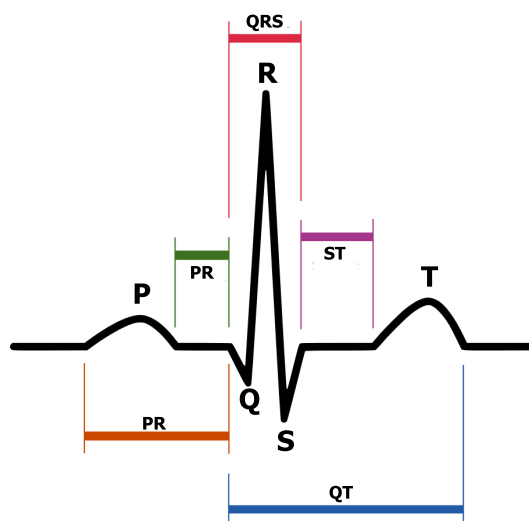
2.2.1 Normalni sinusni ritem

V prejšnjih razdelkih smo opisali, da normalni sinusni ritem izvira iz sinoatrijskega vozla. Povprečen utrip, v mirovanju, je med 60 in 100 BPM, utripi pa so med seboj enakomerno razporejeni. V tem razdelku je opisano, kako utripe prikaže EKG ter kaj velja za normalen zdrav utrip.

PQRST

Normalen srčni utrip je v EKG predstavljen kot zaporedje vrhov in dolin, imenovanih PQRST (Slika 2.7).

Srčni utrip se začne z valom P, ki predstavlja depolarizacijo atrija z začetkom v sinoatrijskem vozlu. Dolžina vala je krajša od 80ms. PR-interval je obdobje med začetkom vala P in začetkom QRS-kompleksa, predstavlja pa čas, ki je potreben da signal preide iz sinoatrijskega vozla do atriventrikularnega vozla (AV). Dolžina intervala je med 120ms in 200ms. Najbolj izstopajoči del signala predstavlja QRS-kompleks. To je kompleks, ki nastane



Slika 2.7: Srčni utrip, kot ga vidimo na posnetku EKG. Najvišji vrh R predstavlja obdobje repolarizacije, ki smo ga opisali v razdelku 2.1.2. [21]

ob prehodu signala iz AV-vozla v Purkinijeva vlakna. Ta del srca vsebuje večino mišične mase, zato ima val R veliko višjo amplitudo od preostalih delov signala. Normalen QRS-kompleks traja med 80ms in 100ms. QRS-kompleks z valom T povezuje segment ST, predstavlja pa depolarizacijo ventriklov. Val T nastane ob repolarizaciji ventriklov, traja pa 160ms. QT-interval meri dolžino od začetka QRS-kompleksa do konca vala T. Njegova dolžina je odvisna od hitrosti srčnega utripa, zato jo je potrebno normalizirati z korenomo dolžine srčnega utripa (\sqrt{RR}). Grafični prikaz signala je predstavljen na Sliki 2.7.

2.3 Srčne bolezni

V tem razdelku je opisanih nekaj najpogostejših bolezni srca, ki jih lahko odkrivamo z EKG. Te bolezni smo v naslednjih poglavjih poskušali identificirati s pomočjo strojnega učenja.

2.3.1 Sinusna tahikardija

Tahikardija je pojav, ko je srčni ritem v mirovanju hitrejši od 100BPM. Sam po sebi ni nevaren, pogosto pa vodi v ventrikularno fibrilacijo.

2.3.2 Sinusna bradikardija

Sinusna bradikardija (SBR ²) je nasprotje tahikardije. Gre za nizek srčni utrip, pod 60 utripov na minuto. Pojavlja se predvsem pri športnikih, katerih srce je dovolj možno, da lahko z manj utripi prečrpa enako količino krvi. Samo po sebi ni nevarno.

2.3.3 Levokračni blok

Do levokračnega bloka (LBBB ³) pride, ko signal zaradi blokade ne doseže levih Purkinijevih vlaken. Zaradi tega je krčenje levega ventrikla zakasnjeno. Bolezen je neozdravljiva, v večini primerov pa ne povzroča večjih težav. Pri bolniku se odraža s povišanim srčnim utripom in hitrejšo zadihanostjo (utrujenostjo) pri naporu.

Diagnoza:

- QRS-dolžina je daljša od 120ms
- Dominanten val S na odvodu V1
- Podaljšan vrh R v odvodu V5-6 (spominja na M)

V signalu EKG (Slika 2.8) lahko vidimo popačen val na odvodih V1 in V6.

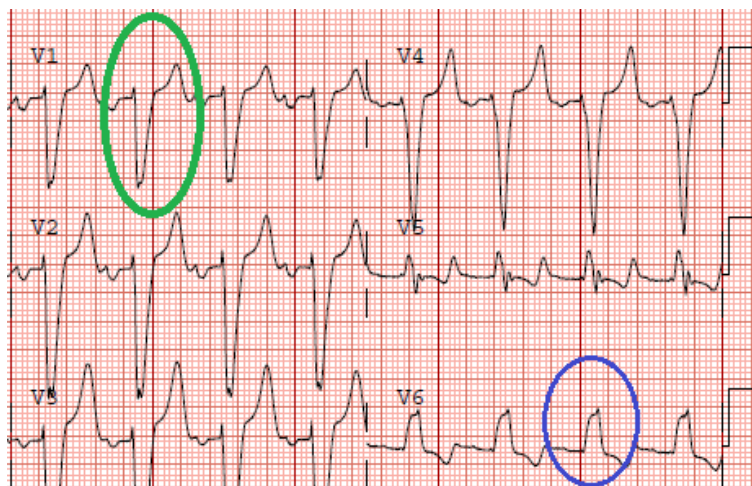
2.3.4 Desnokračni blok

Vzroki za nastanek desnokračnega bloka (RBBB ⁴) so podobni kot pri LBBB (glej razdelek 2.3.3), z razliko, da tokrat do motenj prihaja v desnem ventri-

²ang. Sinus Bradycardia

³ang. Left Bundle Branch Block

⁴ang. Right Bundle Branch Block



Slika 2.8: Z zeleno je označen močno ojačan val S, z modro pa vrh v obliki črke M. [22]

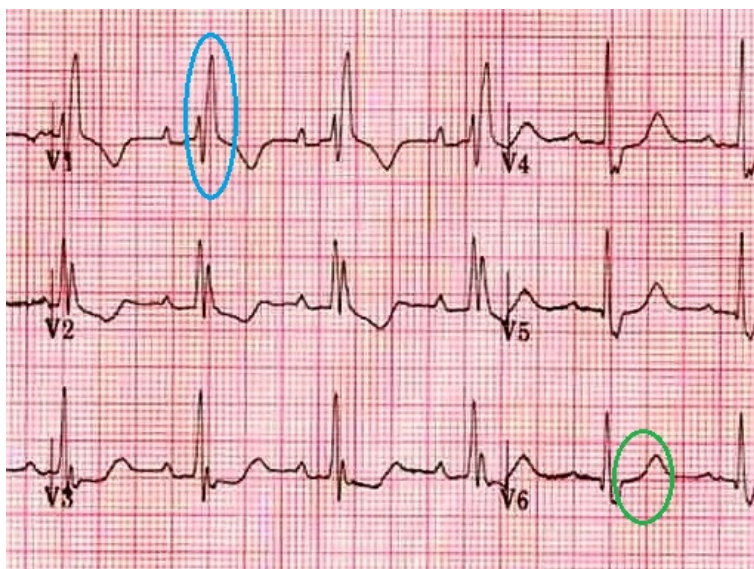
klu. Na Sliki 2.10 lahko na odvodu V1 vidimo močan vrh R, ki s predhodnim valom P spominja na črko M. Na odvodu V6 lahko opazimo podaljšan val S.

Diagnoza:

- Dolžina QRS daljša od 100ms
- val oblike RSR' na odvodu V1 (glej Sliko 2.9)
- podaljšan, nerazločen val S na odvodu V5-6 (W)



Slika 2.9: RSR oblika QRS-kompleksa značilna za RBBB. [23]



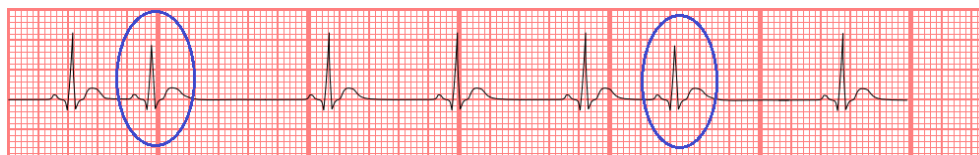
Slika 2.10: Z modro označen vzorec M na odvodu V1, z zeleno pa podaljšan val S. Oba simptoma sta značilna za RBBB. [23]

2.3.5 Atrijska ekstrasistola (prezgodnji atrijski utrip)

Do atrijske ekstrasistole (PAC ⁵) pride, ko se del srca depolarizira hitreje kot sinoatrijski vozal. Zakaj točno pride do tega še ni znano, pojavlja pa se tako pri mlajših kot starejših posameznikih (tudi tistih brez drugih srčnih težav). Na Sliki 2.11 lahko opazimo neenakomeren srčni utrip .

Diagnoza:

- Prehiter (ne-enakomeren) utrip srca.



Slika 2.11: Z modro barvo sta označena prezgodnja utripa. [24]

⁵ang. Premature Atrial Complex

Glede na pogostost pojava ločimo več vrst:

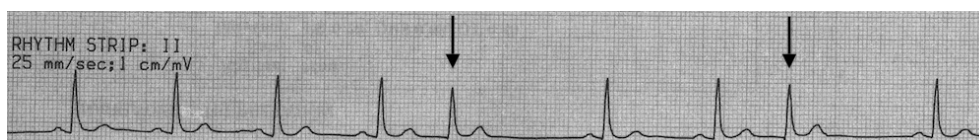
- **Bigemija** Vsak drugi utrip je tipa PAC.
- **Trigemija** Vsak tretji utrip je tipa PAC.
- **Kvadrigimija** Vsak četrti utrip je tipa PAC.
- **Dvojček** Dva zaporedna utripa tipa PAC.
- **Trojček** Trije zaporedni utripi tipa PAC.

2.3.6 Prezgodnji nodalni utrip

Prezgodnji nodalni utrip (PJC ⁶) nastane, ko se signal za utrip sproži v atriventrikularnem vozlu, namesto v sinoatrijskem. Na Sliki 2.12 v 5. in 8. utripu opazimo manko vala P ter neenakomernost utripa.

Diagnoza:

- Prezgodnji utrip QRS, pred njim pa ni vala P.



Slika 2.12: S puščico označena utripa nimata predhodnega vala P. Opazimo lahko, da sta se zgodila prezgodaj. [25]

2.3.7 Ventrikularna ekstrasistola (prezgodnji ventrikularni utrip)

Prezgodnji ventrikularni utrip (PVC ⁷) nastane, ko se signal sproži v Purkinjevih vlaknih namesto v sinoatrijskem vozlu. Pojavlja se tudi pri zdravem

⁶ang. Premature Junctional Complex

⁷ang. Premature Ventricular Contraction

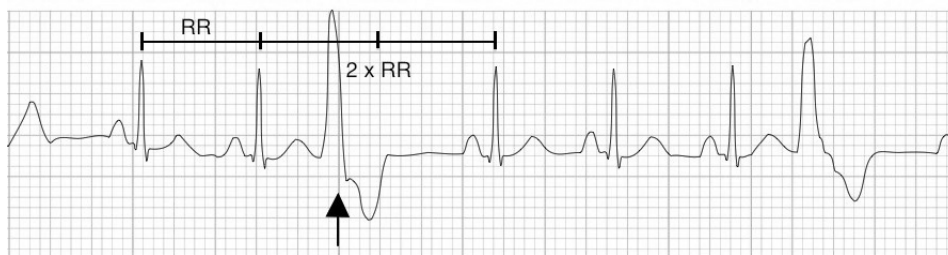
srcu. Zdravljenje ni potrebno, če se to ne pojavlja prepogosto. Problem se lahko zdravi oz. omili, z jemanjem mineralov (magnezij, kalcij). Zaradi daljšega obdobja od PVC do naslednjega (normalnega) utripa, se pacientu lahko zdi, da je srce izpustilo en utrip. Na Sliki 2.13 vidimo daljše ($> RR$) obdobje med PVC in normalnim utripom.

Diagnoza:

- QRS-kompleks daljši od 120ms
- Prezgodnje krčenje srca
- Sprememba vala ST in T

Podobno kot pri atrijskem, tudi PVC ločimo glede na pogostost pojavitve:

- **Bigemija** Vsak drugi utrip je tipa PVC.
- **Trigemija** Vsak tretji utrip je tipa PVC.
- **Kvadrigimija** Vsak četrti utrip je tipa PVC.
- **Dvojček** Dva zaporedna utripa tipa PVC.
- **Trojček** Trije zaporedni utripi tipa PVC.



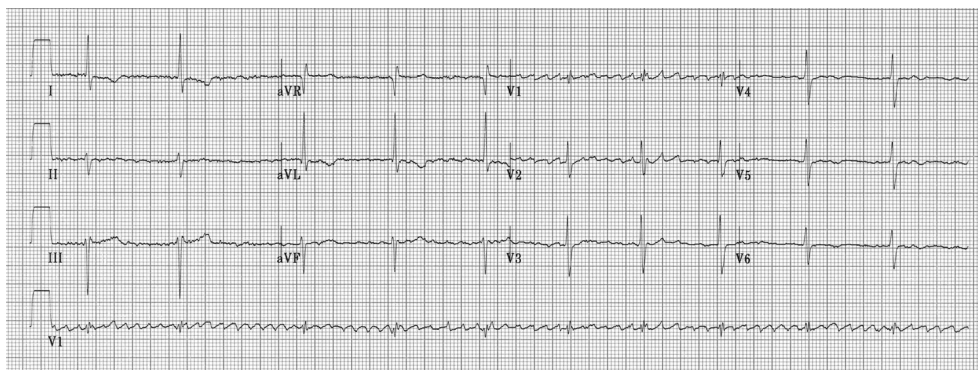
Slika 2.13: S puščico je označen PVC utrip. Vidimo, da je do naslednjega utripa daljše obdobje. [26]

2.3.8 Nodalni pobegli ritem

Junkcijski pobegli ritem (ang. Junctional Escape Rhythm) je utrip, ki se zakasni nekje v AV-vozlju. Zgodi se, ko je depolarizacija v sinoatrijskem vozlju počasnejša od atriventrikularnega vozla. Lahko se zgodi tudi, če signal iz sinoatrijskega vozla ne doseže AV-vozla zaradi blokade. Vlogo spodbujevanja tako prevzame živčevje pod AV-vozlom. Primer je na Sliki 2.14.

Diagnoza:

- Srčni utrip med 40-60 BPM
- Neenakomeren utrip
- Val P je lahko pred QRS-kompleksom
- Če je val P pred QRS-kompleksom, traja interval PR manj kot 0.12s
- Normalen QRS-kompleks



Slika 2.14: Valovanje pri nodalnem pobeglem ritmu. [27]

2.3.9 Ventrikularna fibrilacija (prekatno migetanje)

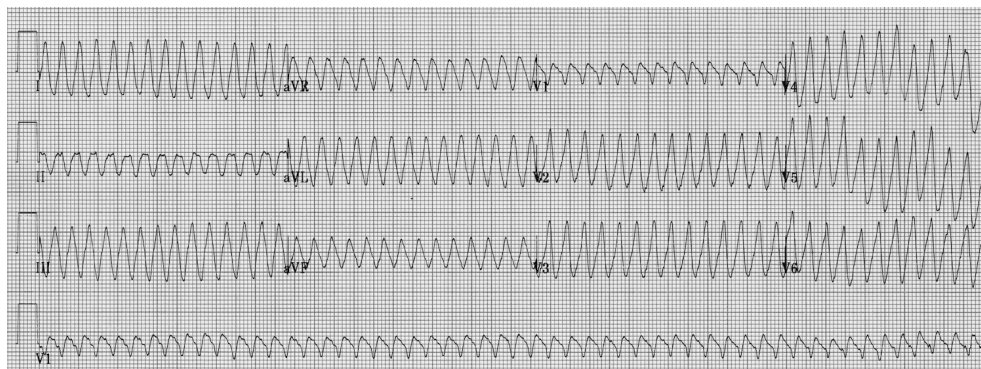
Ventrikularna fibrilacija (VFIB⁸) spada pod tahikardijo (prehiter srčni utrip), ki povzroči krčenje ventrikla hitreje od 250-300 BPM. Tahikardija se pogosto

⁸ang. ventricular fibrillation

prevesi v fibrilacijo, kar pomeni, da srce popolnoma izgubi svojo funkcijo. Fibrilacija je zelo nevarna, vsako minuto, ko je ne zdravimo, zmanjšamo možnost preživetja za 10%. Normalen sinusni ritem se vzpostavi z defibrilacijo (elektrošok). Signal EKG izgleda kot neenakomerno sinusno nihanje z visokimi amplitudami (glej Sliko 2.15).

Diagnoza:

- Neprekinjen sinusni val
- Nerazločni valovi QRS in T
- Utrip hitrejši od 200 BPM



Slika 2.15: Na posnetku EKG vidimo nerazločne sinusne vijuge z frekvenco $> 200\text{BPM}$. [28]

2.3.10 Atrijska fibrilacija

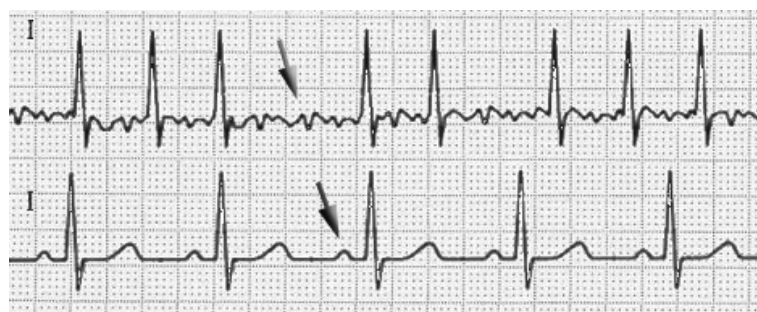
Atrijska fibrilacija (AFIB ⁹) je najpogostejša aritmija. Pogosto nastane, ko se nekaj neenakomernih utripov prevesi v daljše obdobje neenakomernega srčnega utripa. Aritmija poveča možnost srčnega zastoja (pri starejših od 40 let) za kar 25 odstotkov. Na Sliki 2.16 vidimo dva posnetka EKG prvega odvoda. Pri prvem imamo AFIB drugi pa je normalen sinusni ritem. Pri

⁹ang. Atrial fibrillation

prvem signalu lahko vidimo neenakomeren utrip ter veliko valovanja med vrhovi RR, to nastane zaradi migetanja preddvorov. Primer je prikazan na Sliki 2.16.

Diagnoza:

- Neenakomeren srčni utrip
- Brez vala P
- Povišan srčni utrip
- QRS krajši od 120ms.



Slika 2.16: Posnetek prikazuje AFIB (zgoraj) ter normalen sinusni ritem spodaj. Pri AFIB lahko opazimo neenakomerne vrhove R ter nerazpoznavno valovanje med njimi. [29]

2.3.11 Atrijska undulacija (plapolanje preddvorov)

Atrijska undulacija (AFL¹⁰) nastane zaradi kroženja (ang. re-entry) signala v desnem atriju. Navadno se to zgodi zaradi prezgodnjega proženja pulza v sinoatrijskem vozlu. Hitrost trepeta se meri v številu signalov, ki zaokrožijo po atriju. Če se signal v desnem atriju sproži 300 krat na minuto, medtem ko je utrip srca 150 krat na minuto, temu rečemo srčna blokada 2:1. Zdravila ali

¹⁰ang. Atrial flutter

druge srčne bolezni lahko povzročijo blokade 3:1 ali 4:1. Atrijska undulacija je kratkotrajen, v redkih primerih se prevesi v atrijsko fibrilacijo. Primer je prikazan na Sliki 2.17.

Diagnoza:

- Povišan srčni utrip.
- Visoka frekvenca atrijskega signala (spominja na val P)



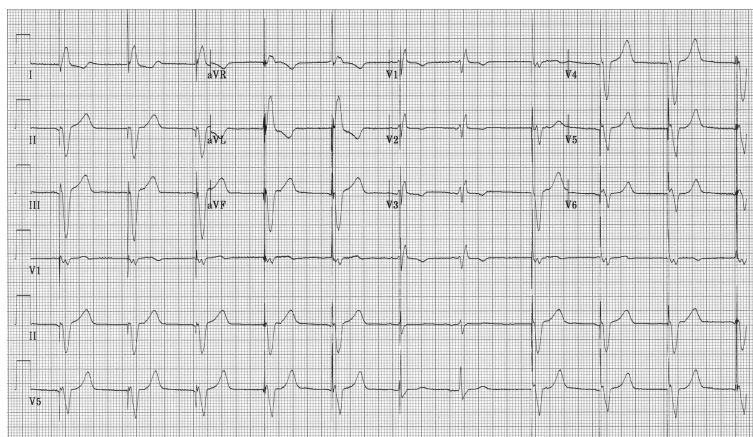
Slika 2.17: Na EKG vidimo veliko valov, ki spominjajo na val P. [30]

2.3.12 Spodbujen ritem

Spodbujen ritem (P¹¹) je ritem, ki nastane ob uporabi srčnega spodbujevalnika. Srčni spodbujevalnik lahko blaži oziroma zdravi nekatere srčne bolezni. Srčnih spodbujevalnikov je več vrst. Večina modernih spodbujevalnikov nadzoruje srčni utrip, če pa spodbujevalnik ugotovi, da do srčnega utripa ni prišlo, pošlje nizko napetostni signal v ventrikle ter s tem povzroči njihovo krčenje. Novejši spodbujevalniki lahko poleg ventriklov nadzorujejo ter prožijo tudi atrije, nekateri lahko zaznajo tudi fizično aktivnost človeka, ter temu primerno povešajo srčni utrip. Primer je prikazan na Sliki 2.18.

¹¹ang. Paced rhythm

Glede na predel srca, ki ga nadzorujejo, ter funkcijo, ki jo opravljajo, srčne spodbujevalnike ločimo na 5 kategorij. Srčni spodbujevalniki in njihovo delovanje so natančneje opisani v članku [31].



Slika 2.18: Primer EKG s srčnim spodbujevalnikom.[32]

Poglavje 3

Globoke nevronske mreže

Prvi modeli umetnih nevronskih mrež segajo v sam začetek razvoja računal. Leta 1954 sta Farley in Wesley A. Clark prva implementirala umetno nevronske mrežo na računalnik. Nato je sledil hiter razvoj naprednejših struktur, mreže so vsebovale več nivojev nevronov v t.i. večnivojski perceptron (MLP ¹). Leta 1975 je za učenje Paul Werbos prvi uporabil vzvratno razširjanje (ang. backpropagation). Razvoj umetnih nevronskih mrež se je ustavil v začetku 80 let, saj takrat računalniki niso bili zmožni preračunavati parametrov v velikih nevronskih mrežah, ki so potrebni za pridobitev rezultatov, primerljivih z ostalimi metodami strojnega učenja.

V začetku 21. stoletja, se je z razvojem računalniških grafičnih kartic, ki so omogočale visoko paralelizacijo, ponovno začelo prebujati zanimanje za nevronske mreže. Med leti 2009 in 2012 se je okrepila uporaba povratnih nevronskih mrež (RNN ²) ter globokih usmerjenih nevronskih mrež (ang. deep feedforward neural networks). Globoke nevronske mreže so začele zmagovati na tekmovanjih za prepoznavanje vzorcev ter ostalih področjih strojnega učenja.

Leta 2011 je Googlu uspelo ustvariti globoko nevronske mrežo, ki se je sama naučila prepoznavati mačke iz videoposnetkov youtube ³(glej članek

¹ang. Multilayer perceptron

²ang. Recurrent neural network

³<https://www.youtube.com/>

[33]). Za učenje so potrebovali 16.000 procesorjev, ki so se učili 3 dni. Zaradi velike potrebe po paralelizaciji se danes večino algoritmov za globoke nevronske mreže izvaja na grafičnih karticah, kjer ima ena sama kartica več kot 2000 računskih jeder. Razvitih je veliko knjižic, ki v veliki meri avtomatizirajo implementacijo paralelizacije programa, ter tega avtomatsko izvajajo na grafični kartici. Ena popularnejših knjižic, ki temelji na programskem jeziku Python, je Theano (glej članka [34],[35]).

Dandanes se globoke nevronske mreže uporabljajo na vedno več področjih, najboljše pa se trenutno obnesejo na področju strojnega vida, prepoznavanja govora ter razumevanja naravnega jezika (glej članke [36], [37], [38]). Širijo se tudi na ostala področja, letos je raziskovalcem z uporabo globokega spodbujenega učenja (ang. deep reinforcement learning) uspelo računalnik naučiti igrati arkadne igre ATARI, kjer je računalnik premagal človeka za več kot 2500% (glej članek [39]).

3.1 Uvod v nevronske mreže

V tem poglavju je opisano, kako so sestavljene ter kako delujejo nevronske mreže. Poglavje začnemo s preprostim klasifikatorjem; linearno regresijo (LR), končamo pa z MLP.

3.1.1 Logistična regresija

Linearna regresija meri odvisnost med neodvisno spremenljivko, označeno z X , in odvisnim skalarjem, označenim z Y . Cilj je poiskati takšne parametre θ , da je napaka v enačbi 3.1, čim manjša.

Za minimizacijo funkcije 3.2, je najlažje uporabiti gradientni spust (ang. gradient descend), ki je enostaven za implementacijo ter relativno hitro konvergira k lokalnemu minimumu. Ideja za metodo je, da gradient pokaže v smer razlike predvidene in resnične vrednosti, ki jo lahko nato postopoma

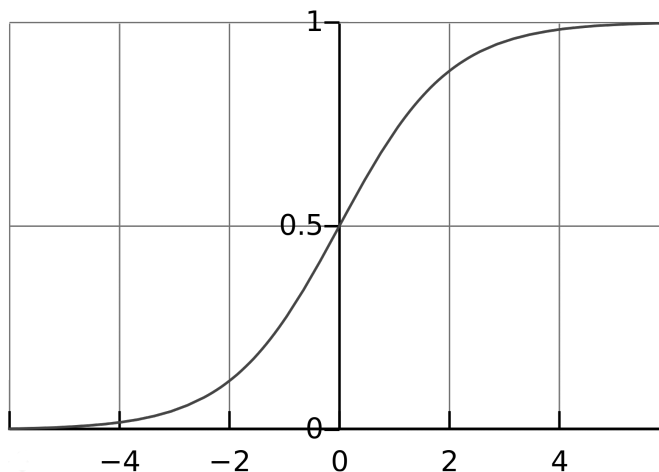
zmanjšujemo. Gradient dobimo z odvodom enačbe 3.2.

$$y = X^T\theta + \epsilon \quad (3.1)$$

$$E(\theta) = \frac{1}{2} \sum_i (X^T\theta - y)^2 \quad (3.2)$$

Če želimo spremenljivke klasificirati v dva (ali več) razredov, moramo linearno regresijo prirediti tako, da namesto zveznih vrednosti vrne (binarni) razred. Zanima nas torej, kolikšna je verjetnost, da spremenljivka spada v prvi ali drugi razred. Pomagamo si lahko s tako imenovano sigmoidno funkcijo, ki jo opisuje enačba 3.3 (glej Sliko 3.1), ki vrednosti $X^T\theta$ preslika na interval $(0, 1)$.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$



Slika 3.1: Sigmoidna funkcija vrednosti iz $(-\infty, \infty)$ preslika na interval $(0,1)$.

Z uporabo funkcije 3.1 v sigmoidi, dobimo verjetnost v enačbah 3.4 in 3.5. Želimo poiskati takšne parametre θ , ki bodo vrnilo visoko verjetnost 3.4, ko

X pripada 1. razredu in visoko verjetnost za 3.5, ko X pripada razredu 0. To dobimo s kriterijsko funkcijo 3.6. Za minimizacijo lahko ponovno uporabimo gradientni spust.

$$P(y = 1|x) = h(x) = \frac{1}{1 + e^{(-X^T \theta)}} \quad (3.4)$$

$$P(y = 0|x) = 1 - P(y = 1|x) \quad (3.5)$$

$$E(\theta) = - \sum_i (y \times \log(h(x)) + (1 - y) \log(1 - h(x))) \quad (3.6)$$

Če želimo spremenljivke klasificirati v več kot dva razreda, moramo logistično regresijo popraviti v multinomsko regresijo (ang. multinomial ali softmax). Pri logistični regresiji smo imeli $y \in \{0, 1\}$, želimo si $y \in \{0, 1, \dots, K\}$. Zanima nas torej $P(y = k|x)$, za vse vrednosti $k = 1, \dots, K$. Če to vpeljemo v verjetnostno enačbo, dobimo 3.7 (*Opomba: Vsota pod vlomkom normalizira verjetnosti, tako da se seštejejo v 1*). Enačbo 3.6 lahko generaliziramo. Vpeljemo oznako $[y = k]$, ki vrne 1, v primeru, da je vrednost y enaka k , 0 sicer. Dobimo torej kriterijsko funkcijo 3.8. Z uporabo gradientnega spusta ponovno iterativno določamo parametre.

Pri klasifikaciji preprosto vzamemo tisti razred (indeks), kateri vrne najvišjo verjetnost za določen x ($\max(h(x))$).

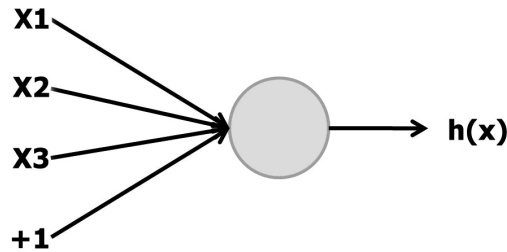
$$h(x) = \begin{bmatrix} P(y = 1|x; \theta) \\ P(y = 2|x; \theta) \\ \vdots \\ P(y = K|x; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^K e^{(X^T \theta_j)}} * \begin{bmatrix} e^{(X^T \theta^{(1)})} \\ e^{(X^T \theta^{(2)})} \\ \vdots \\ e^{(X^T \theta^{(K)})} \end{bmatrix} \quad (3.7)$$

$$E(\theta) = - \sum_i \sum_{k=1}^K [y^{(i)} = k] * \log\left(\frac{e^{X^T \theta}}{\sum_{j=1}^K X^T \theta_j}\right) \quad (3.8)$$

3.1.2 Večnivojski perceptron

Oglejmo si najbolj preprosto umetno nevronske mrežo, ki jo sestavlja en sam nevron, z n vhodnimi povezavami, pristransko (ang. bias) povezavo in izhodno hipotezo (Slika 3.2). Vsaka vhodna povezava v nevron ima svojo utež w_i , izhodno povezavo pa vrne aktivacijska funkcija na podlagi vsote $x^T W$ (enačba 3.9). Želimo si torej, da nam nevron za dane vhodne podatke X vrne približek za $y = h(x)$.

Za aktivacijsko funkcijo lahko ponovno uporabimo sigmoidno funkcijo 3.3. Ko delamo z globokimi nevronske mrežami, se izkaže, da je bolje namesto običajne sigmoidne funkcije uporabljati tangens hiperbolikus 3.10 ali usmerjevalno funkcijo (ang. rectifier) 3.11. V članku [40] avtorji primerjajo sigmoidne ter tangens hiperbolikus funkcije. Tangens hiperbolikus je razširjena verzija sigmoide, saj vrednosti preslika na interval $(-1, 1)$. Usmerjevalna funkcija je običajna linearna funkcija za vrednosti večje od 0, sicer pa 0. Na Sliki 3.3 so prikazane vse tri funkcije.

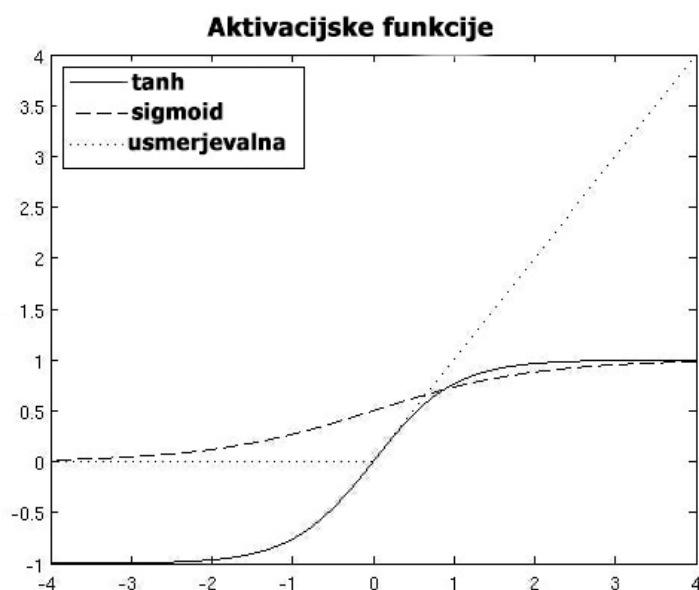


Slika 3.2: Nevron na sliki ima tri podatkovne ter eno pristransko vhodno povezavo. Na eni izhodni povezavi, pa vrne vrednost aktivacijske funkcije.

$$h_{W,b}(x) = f(x^T W) = f\left(\sum_{i=1}^n x_i W_i + b\right) \quad , \text{ kjer } f : \mathbb{R} \rightarrow \mathbb{R} \quad (3.9)$$

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.10)$$

$$f(x) = \max(0, x) \quad (3.11)$$

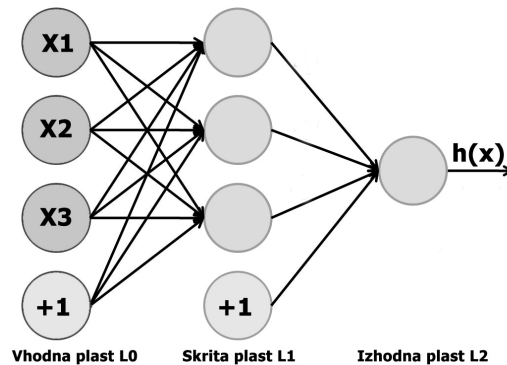


Slika 3.3: Na sliki so prikazane sigmoidna, tanh ter usmerjevalna funkcija.

Usmerjene večnivojske nevronske mreže

Nevronska mreža je torej mreža, sestavljena iz preprostih nevronov (glej Sliko 3.2), ki so razporejeni v plasti. Za vsako plast velja, da nevroni znotraj plasti niso povezani med sabo, na naslednjo plast pa so povezani z vsemi nevroni. Takim mrežam (plastem) pravimo tudi polno povezane nevronske mreže (plasti). *Opomba: Obstajajo tudi mreže, v katerih so nevroni med seboj povezani tudi znotraj plasti, ena takih mrež je RNN, lahko pa imamo tudi redko povezanost, kjer nevron iz ene plasti ni povezan na vse nevrone iz naslednje.*

Na Sliki 3.4 lahko vidimo enostavno nevronske mrežo, sestavljeno iz ene skrite plasti nevronov ter vhodne in izhodne plasti. Prvo plast (L_0) sestavljajo vhodni podatki, ki so nato polno povezani na drugo plast. Vsaka povezava ima svojo utež (w_i), vsak nevron pa ima tudi pristransko utež (b_i). V drugi plasti (L_1) je n (skritih nevronov), ki izračunajo odziv (enačba 3.9) na podatke iz L_0 plasti. Zadnji plasti pravimo izhodna plast (L_2).



Slika 3.4: Enostaven večnivojski perceptron, sestavljen iz vhodne, skrite ter izhodne plasti.

Če želimo izračunati rezultat, ki ga vrne zadnja plast, moramo rekurzivno izračunati izhode posameznih nevronov na vseh ostalih plasteh na enak način kot za posamezen nevron (enačba 3.9), s to razliko, da na kasnejših plasteh, namesto vhodnih podatkov, vzamemo izhodne podatke prejšnje plasti. Temu koraku pravimo razširjanje naprej (ang. feedforward propagation), saj se podatki pošiljajo (računajo) naprej iz vhodne plasti v globlje plasti nevronske mreže.

Vzvratno razširjanje

Kriterijsko funkcijo (ang. cost function) definiramo kot polovico kvadratne evklidske razdalje (enačba 3.13). Normaliziramo jo lahko z uporabo pojemajočih uteži (ang. weight decay) (enačba 3.12), ki zmanjša vpliv uteži in s tem prepreči preveliko prileganje podatkov (ang. overfitting) z uporabo parametra λ .

$$D(W) = \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \quad (3.12)$$

$$E(W, b, X, Y) = \frac{1}{k} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x) - y\|^2 \right) + D(W) \quad (3.13)$$

Za učenje mreže (računanje uteži) lahko ponovno uporabimo gradientni spust. Za zadnjo plast lahko preprosto izračunamo odvod, saj poznamo vrednost y . Problem nastane pri računanju odvodov v skritih plasteh. Z uporabo metode vzratnega razširjanja lahko napako iz zadnje plasti propagiramo proti prvim. Odvode za uteži in pristranskost lahko izračunamo z enačbami 3.14 in 3.15, edina razlika med enačbama je, da pri računanju odvoda za pristranskost ne računamo pojemaajočih uteži.

$$\frac{\delta}{\delta W_{ij}^{(l)}} E(W, b) = \left[\frac{1}{k} \sum_{i=1}^k \frac{\delta}{\delta W_{ij}^{(l)}} E(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)} \quad (3.14)$$

$$\frac{\delta}{\delta b_i^{(l)}} E(W, b) = \frac{1}{k} \sum_{i=1}^k \frac{\delta}{\delta b_i^{(l)}} E(W, b; x^{(i)}, y^{(i)}) \quad (3.15)$$

Pseudokoda za algoritem je prikazana z Algoritmom 1.

Algoritem 1 Pseudokoda za vzratno razširjanje

- 1: *Forwardpass* \leftarrow Za učne primere (x, y)
 - 2: **for** vse izhodne nevrone **do**
 - 3: $\delta_i^{(n_l)} = -(y_i - a_i^{(n_l)}) * f'(x_i^{(n_l)}) \leftarrow$ izračuna odvode za vsak nevron
 - 4: **end for**
 - 5: **for** vse plasti $l = n_l - 1, n_l - 2, \dots, 2$ **do**
 - 6: **for** vse nevrone na plasti **do**
 - 7: izračunaj parcialne odvode iz plasti $(l+1)$ v plast (l)
 - 8: **end for**
 - 9: **end for**
-

Opomba: Za nevronske mreže je pomembno, da začetne uteži niso uniformne, saj bi v tem primeru vse enako konvergirale. Potrebno jih je inicializirati z naključnimi začetnimi vrednostmi, pri tem pa se moramo zavedati, da je končna točnost nevronske mreže odvisna od začetne inicializacije uteži.

Prvi korak učenja nevronske mreže je torej vzratno razširjanje, drugi korak pa je uporaba optimizacijskih metod za posodabljanje uteži. Do sedaj smo omenili le gradientni spust. Izkaže se, da gradientni spust počasi konvergira proti rešitvi. To je še posebej razvidno v primeru, ko imamo velike

(globoke) nevronske mreže. V članku [41] avtorji pregledajo še nekaj metod, ki pohitrijo učenje z majhnim ali ničtim vplivom na končno točnost mreže.

3.1.3 Stohastični gradientni spust

Gradientni spust izračuna napako in odvod za celotno množico podatkov hkrati. To je počasno, saj v večini primerov ne moremo shraniti vseh podatkov v spomin (RAM).

Algoritem lahko nadgradimo tako, da izračunamo gradient za vsak par vhodno-izhodnih podatkov sproti, tej metodi pravimo stohastični gradientni spust (SGD ⁴). Enačbo za standardni gradientni spust 3.16 spremenimo tako, da odvod računa za vsak par podatkov (glej enačbo 3.17).

$$\theta = \theta - \alpha * \Delta_{\theta} E[J(\theta)] \quad (3.16)$$

$$\theta = \theta - \alpha * \Delta_{\theta} J(\theta; x^{(i)}, y^{(i)}) \quad (3.17)$$

Takšen način računanje gradienta je zelo občutljiv na varianco v podatkih, poleg tega je vhodnih podatkov navadno zelo veliko, še posebej pri globokih nevronskih mrežah. Posamično računanje vsakega para podatkov posebej pa je počasno, zato se v algoritmih navadno uporablja mini-blokovski stohastični gradientni spust (ang. Mini-batched Stochastic Gradient Descent).

Mini-blokovski stohastični spust združi več vhodno izhodnih podatkov ter odvod izračuna na povprečni napaki v tej podmnožici podatkov. Zaradi tega je odvod manj občutljiv na varianco v podatkih. Izkaže se, da se z uporabo paralelizacije (za navaden stohastični gradientni spust to ni mogoče) odvod hitreje izračuna. Enačba mini-blokovskega SGD je prikazana v 3.18. Velikost bloka določimo s spremenljivko m .

$$\theta = \theta - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(k)}) x^{(k)} \quad (3.18)$$

⁴ang. Stochastic Gradient Descent

Optimalna velikost bloka je odvisna od podatkov.

Hitrost učenja uravnavamo s parametrom α . V splošnem se uporablja t.i. način ohlajanja (ang. annealing), to pomeni, da je stopnja učenja v začetku visoka, ko pa se približujemo rešitvi, se stopnja učenja manjša. S tem preprečimo, da gradient skoči iz lokalnega minimuma ter začne divergirati.

Problem pri uporabi SGD je počasna konvergacija, ko se mreža bliža rešitvi, tudi če ne uporabljamo ohlajanja.

3.1.4 Momentum

V primeru, da naš optimizacijski problem spominja na sedlo, t.j. blaga dolina med dvema strmima pobočjema, bo SGD obtičal na ravnini, saj bo največji gradient usmerjen proti pobočju. To rešimo z dodajanjem momenta v SGD. V standardno enačbo vpeljemo spremenljivko $v \in (0, 1]$, ki pomaga potisniti gradient proti rešitvi (enačba 3.19). v je enake dimenzije kot θ , določa pa vpliv prejšnjih odvodov na trenutno učenje. Navadno je v v začetku učenja okrog 0.5, proti koncu pa 0.9.

$$v = \gamma v + \alpha \Delta f \quad (3.19)$$

3.1.5 ADADelta

Do sedaj opisane metode posodabljaajo stopnjo učenja in parametrov globalno. Želimo si, da bi lahko za vsak parameter posebej nastavili stopnjo učenja ter jo tekom učenja spreminjali.

ADAGrad shranjuje velikost gradienta za vsak parameter (enačba 3.20), ta se nato uporabi za normalizacijo parametrov (po elementih) (enačba 3.21). Vidimo lahko, da se parametrom z visokimi gradienti močno zmanjša stopnja učenja, obratno pa velja za parametre z majhnimi vrednostmi gradientov. Da se izognemo deljenju z ničlo, v normalizacijo dodamo ϵ .

$$v = v + (\Delta f)^2 \quad (3.20)$$

$$\theta = \theta + \alpha \frac{\Delta f}{\sqrt[2]{v} + \epsilon} \quad (3.21)$$

Problemi z uporabo take uteži se pokažejo v začetku učenja. Ker so gradienti v začetku pogosto zelo veliki, bo učenje potekalo izredno počasi. Drugi problem se pojavi ob daljšem učenju, saj se v vsaki iteraciji utež poveča, s tem pa pade stopnja učenja, dokler se v neki točki ne ustavi na 0. Ta problem rešujemo z modificirano metodo ADADelta.

ADADelta ne akumulira gradienta skozi celotno učenje, ampak le za m zadnjih izračunov. S tem prepreči (prehitro) konvergenco stopnje učenja proti 0. Na podlagi članka [42] smo implementirali našo verzijo ADADelta. Zaradi lažje implementacije smo se odločili, da bomo zadnjih m izračunov dobili z uporabo eksponentne uteži. To preprosto naredimo tako, da za m izberemo $m \in (0, 1)$. Zaradi tega ima vsak prejšnji korak eksponentno manj vpliva glede na število korakov. Za odpravo problema visokih gradientov smo vpeljali popravljeno funkcijo \tanh , kjer t predstavlja trenutno iteracijo.

$$v = v * m + \frac{1}{1 + e^{-\frac{t}{5}}} (\Delta f)^2 \quad (3.22)$$

3.1.6 RMSProp

Avtorji članka [41] kot izboljšavo predlagajo RMSProp, ki deluje na podoben način kot ADADelta, s tem da doda še parameter razpada (ang. decay), kateri pove razmerje med trenutnim in prejšnjimi gradienti. Enačbi za RMSProp:

$$v = \beta v + (1 - \beta)(\Delta f)^2 \quad (3.23)$$

$$\theta = \theta - \alpha \frac{\Delta f(\theta)}{\sqrt[2]{v} + \epsilon} \quad (3.24)$$

V naših testiranjih se je RMSProp izkazal za najboljšo optimizacijsko metodo, saj konvergira najhitreje, vpliva na končno točnost pa ni. Primerjava opisanih optimizacijskih metod je v poglavju z rezultati na Sliki 4.2.

3.2 Konvolucijske nevronske mreže

Pri strojnem učenju, še posebej računalniškem vidu, hitro naletimo na strojne omejitve velikosti spomina (RAM), katere lahko računalnik učinkovito obdelava. Pogosto se problema lotimo tako, da algoritmom namesto surovih podatkov, podajamo značilke (ang. features), podatke krčimo - aproksimiramo ter zavračamo manj informativne dele (PCA ⁵, ICA ⁶).

To pomeni, da učenje z večino algoritmov poteka v treh delih. Najprej surove podatke pred-obdelamo, sledi strojno učenje, na koncu pa še interpretacija rezultatov - post-obdelava. Predvsem pri pred-obdelavi lahko nastanejo težave. Algoritmi, ki iz surovih podatkov izluščijo značilke so v večini primerov splošni, ne ozirajo se na vrsto strojnega učenja, ki bo uporabljeno za dejansko učenje. Pogosto se zaradi pred-obdelave razbije (izgubi) notranja struktura med podatki, s tem pa tudi pomembne informacije, ki bi pomagale izboljšati končno točnost metode strojnega učenja.

V tem poglavju je opisano eno najbolj znanih in intuitivnih vrst globokih nevronskih mrež - Konvolucijske nevronske mreže (ang. Convolutional neural network). Ta vrsta nevronskih mrež je še posebej popularna na področju računalniškega vida, v naslednjem poglavju pa smo pokazali, da se odlično obnese tudi pri obdelavi signala EKG.

3.2.1 Iskanje značilk s pomočjo konvolucije

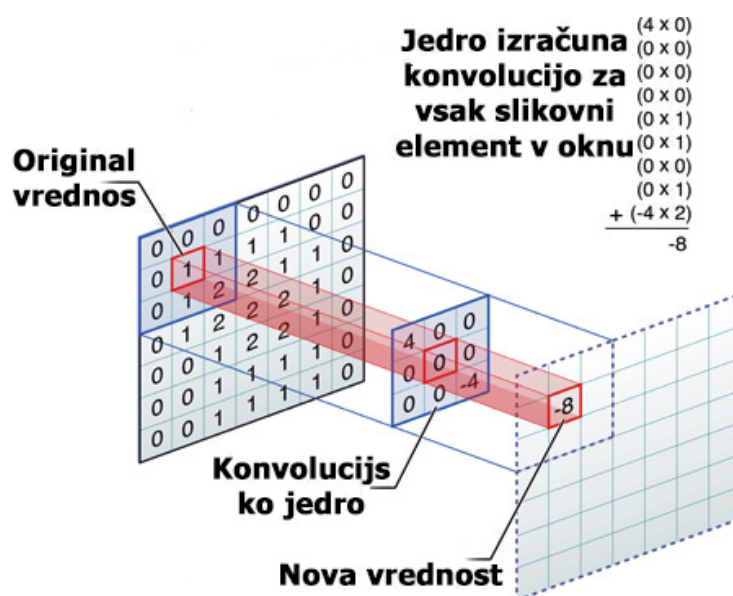
Želeli bi si, da se nevronska mreža sama nauči, kako poiskati značilne točke ter kateri tip značilk je pomemben za trenutno domeno. Konvolucijske nevronske mreže so sestavljene iz več zaporednih konvolucijskih plasti, ki so se sposobne samostojno naučiti, katere značilke so dobre za učno domeno.

Konvolucija v računalniškem vidu pomeni filtriranje slike z določenim filtrom. Ime konvolucija izhaja iz načina filtriranja, pri katerem filter množimo (konvolucija) preko vrednosti v sliki. Ker imajo naravne slike stacionarne

⁵ang. Principal component analysis

⁶ang. Independent component analysis

lastnosti, je tak filter lahko manjši kot cela slika (pogosto okrog 8×8), filter nato potuje po celi sliki, ter računa odzive posameznih oken nanj. To deluje, zaradi stacionarnih lastnosti naravnih slik, na primer, če uporabljamo filter, ki prepozna krog, se bo tak filter sprožil vsakič, ko bo šel čez predel, ki vsebuje krog, ne glede na pozicijo in število krogov v celotni sliki. *Opomba: Seveda pa to ne pomeni, da je tak filter invariatnen na velikost kroga.* Primer delovanja filtra je prikazan na Sliki 3.5.

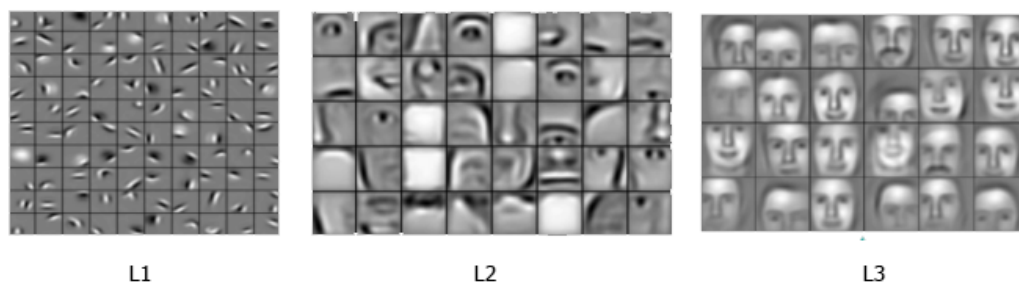


Slika 3.5: Konvolucijski filter velikosti 3×3 zmnoži in sešteje istoležne vrednosti. Vrednost predstavlja odziv filtra na izbran del podatkov. [43]

Namesto, da bi nevronske mreže jedra filtrov določili vnaprej, so se le te same sposobne naučiti, kateri filtri se najbolj odzivajo na določeni domeni podatkov. Filtre lahko tako v mrežo implementiramo kot posebno kopico nevronov, ki si delijo uteži. S tem dosežemo, da se je takšna kopica (filter) sposobna sama naučiti (uteži), na katere podatke se mora odzivati. Uteži si morajo deliti zato, da pokrijejo celotno sliko (vhodne podatke) in da je filter enak za vse podatke.

To pride še posebej do izraza pri globokih (veliko plasti) mrežah, saj so

se različne konvolucijske plasti zmožne naučiti kombinirati odzive filtrov iz prejšnjih plasti v vedno bolj in bolj kompleksne vzorce. Tako se navadno algoritem v prvi plasti nauči razpoznavati navpične in vodoravne robove (2 filtra), v naslednji plasti nato združi ta dva filtra ter se nauči prepoznavati kot. To se lahko stopnjuje do te faze, da algoritem v zadnjih stopnjah prepoznava kompleksne objekte, kot so oko, nos, usta. Primer je prikazan na Sliki 3.6.



Slika 3.6: Primer sestavljanja značilnih filtrov na različnih globinah mreže. Na L1 filter prepoznava črno bele robove, na L2 se nauči značilke iz L1 sestaviti v nos, oči, uho, itd. na zadnji plasti pa sestavi značilke iz L2 v obraz.

Na ta način dobimo filtre, ki se zelo dobro odzivajo na pomembne vzorce na določeni domeni.

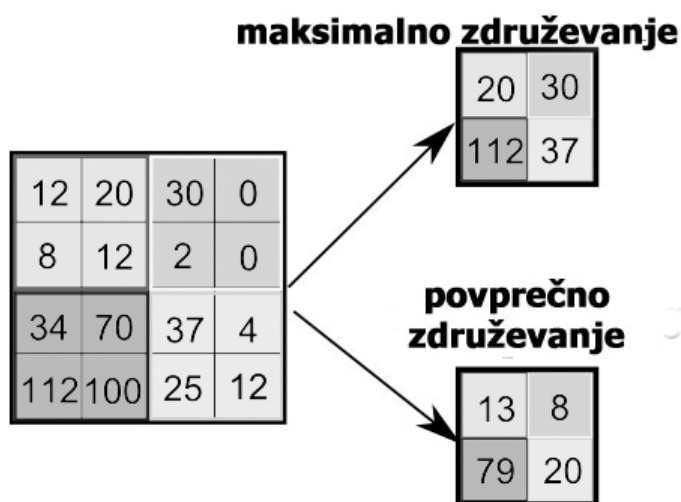
3.2.2 Združevanje

Velik problem pri uporabi računalniškega vida je sama velikost podatkov, ki jih obdelujemo. Če za primer vzamemo za današnje standarde majhno sliko velikosti 1920×1080 slikovnih pik (polna HD ločljivost), nam ena samo takšna slika da 6.220.800 (enačba 3.25) vhodnih surovih podatkov. Če želimo na takšnih podatkih uporabiti 10 filtrov za iskanje značilk, nam bo konvolucijska plast vrnila 62.208.000 izhodnih podatkov. Očitno je, da je to potrebno na

nek način zmanjšati.

$$1920 \times 1080 = 207360 \times 3(\text{RGBbarve}) = 6220800(\text{Byte}) \quad (3.25)$$

Združevanje (ang. pooling) je metoda, ki za neko področje slike izračuna statistiko iz odkritih značilk. To lahko naredi na več načinov, s povprečenjem (ang. mean-pooling), maksimiziranjem (ang. max-pooling) ali na kakšen drug način. Zaradi statičnih lastnosti slik je ta način tudi invarianten za majhne translacije (če se maksimalna značilka, znotraj okna, premakne za nekaj slikovnih pik, bo še vedno v enakem oknu). Tudi izguba informacije je zaradi tega minimalna. V splošnem se izkaže, da združevanje preprečuje preveliko prileganje podatkom in s tem povečuje klasifikacijsko točnost metode. Velikost združevalnega filtra je navadno 2×2 . Že tako majhen filter zmanjša število značilk za $4 \times$, pri zares velikem številu značilk pa se lahko uporabi filter velikosti 4×4 , ki zmanjša število značilk za $16 \times$. Primer združevanja je prikazan na Sliki 3.7.



Slika 3.7: Metoda združi vrednosti iz enega okna. Posledica tega je $4 \times$ manjša slika.

3.3 Povratne nevronske mreže

Konvolucijske nevronske mreže so dobre za prepoznavanje elementov (značilke) na slikah, saj so neodvisni med seboj po lokaciji in velikosti. Ko analiziramo signal EKG, to velja le v primerih, ko se sam signal popači (npr. LBBB), zelo slabo pa se izkaže v primeru, ko za pravilno klasifikacijo niso pomembni le trenutni podatki na vhodu, ampak tudi podatki izpred nekaj sekund (npr. PAC).

Pri običajnih nevronskih mrežah povezave prehajajo iz nevronov na eni plasti v nevrone na naslednji plasti. Povratne nevronske mreže (RNN) pa povežejo tudi nevrone znotraj ene plasti med seboj (Slika 3.8), ustvarijo lahko tudi cikle, s tem ko je izhodna povezava nevrona hkrati njegova vhodna. Na ta način omogočijo, da mreža pri učenju upošteva tudi podatke na sosednjih nevronih, ter podatke, iz prejšnjega učenja.

RNN pridejo do izraza pri domenah kjer so podatki odvisni med seboj. Zelo dobro se obnesejo pri prepoznavanju govora, procesiranju naravnega jezika, prepoznavanju in ustvarjanju glasbe.

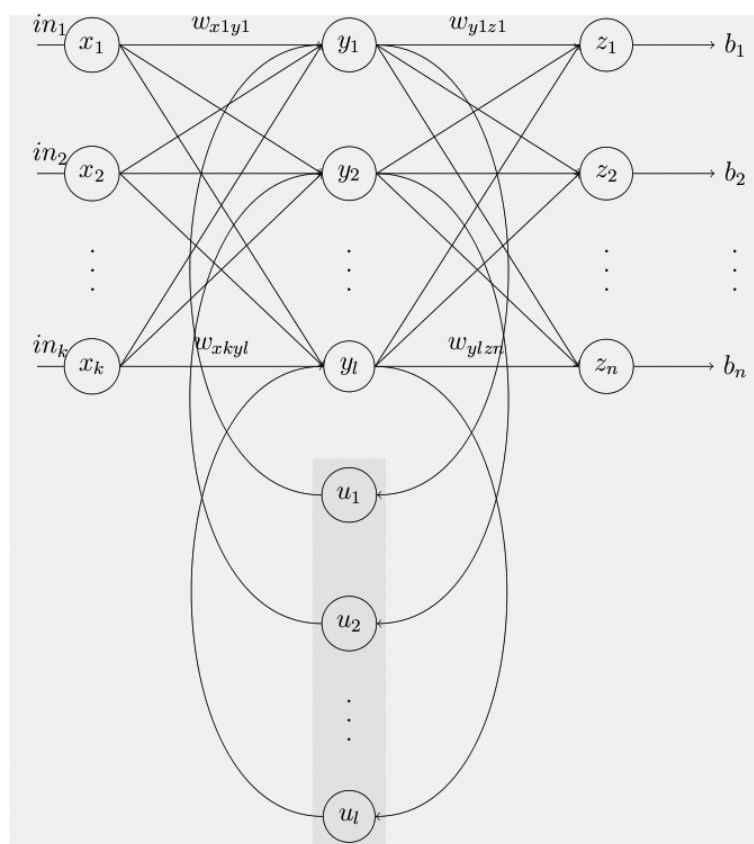
Zaradi arhitekture, je RNN podvržen problemu izginjajočega (ang. vanishing) ali eksplozivnega (ang. exploding) gradienta. To se zgodi, ko se zaradi globine mreže in napak pri zaokroževanju vrednost gradienta približa ničli, hkrati pa se poveča napaka, kar onemogoči učenje takšnih mrež. RNN zato ni zmožen učenja (uporabe) podatkov, ko so le ti preveč narazen. Ta problem rešuje tako imenovana kratko-dolgo ročno spominska mreža (LSTM⁷).

3.3.1 Kratko-dolgo ročna spominska mreža

Kratko-dolgo ročno spominsko mrežo sta leta 1997 zasnoval Sepp Hochreiter in Jurgen Schmidhuber. Mreže LSTM se dobro obnesejo pri odvisnih podatkih, tudi če so ti, časovno, daleč narazen.

LSTM je sestavljena iz tako imenovanih spominskih blokov (nevronov)

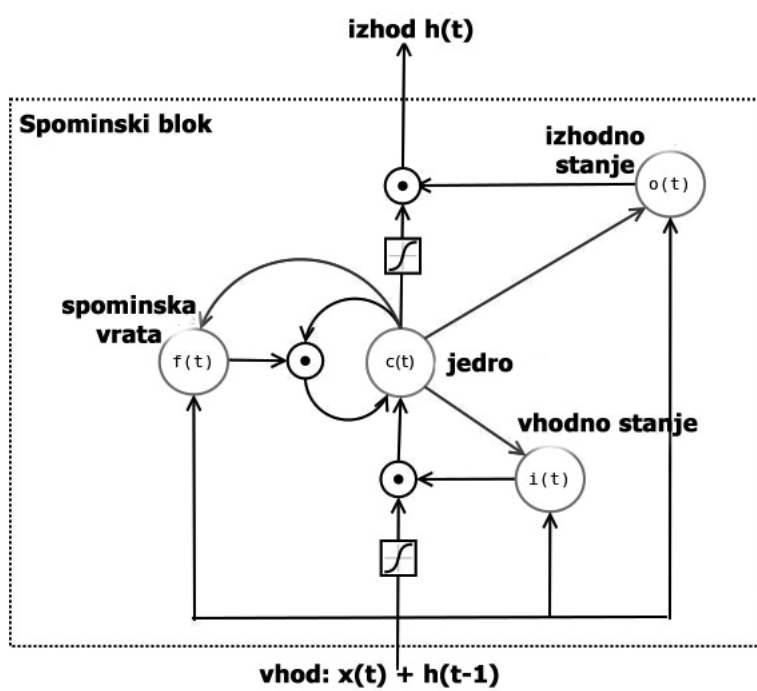
⁷ang. Long Short Term Memory Network



Slika 3.8: V RNN so nevroni na isti plasti povezani med seboj. [44]

(Slika 3.9). Ti nevroni so sposobni ohraniti informacijo poljubno dolgo časa. Tak blok je sestavljen iz enot, ki določijo, kdaj je informacija dovolj pomembna, da si jo nevron zapomni, ter kdaj informacijo lahko pozabi.

Za učenje takšnih mrež se uporablja metoda, imenovana vzvratno razširjanje skozi čas (ang. back propagation through time). Ko algoritem napako prenaša nazaj, se napaka ujame v spominskem bloku. Spominski blok nato podaja napako sam sebi (v spominski celici), dokler se ne nauči, kdaj mora ta podatek (napako) pozabiti.



Slika 3.9: Spominska celica mreže LSTM.[45]

Poglavje 4

Analiza signala EKG z globokimi nevronske mrežami

4.1 Učna množica za učenje in testiranje

Algoritme smo učili in testirali na podatkih iz podatkovne zbirke PhysioNet[46], natančneje na podatkih MITBIH [47]. Podatkovna baza vsebuje 48, 30 minutnih, posnetkov EKG. Posneti so bili na 47 različnih ljudeh, med letom 1975 in 1979 v sklopu laboratorija za aritmijo BIH. 23 posnetkov je izbranih naključno, ostalih 25, iz enake skupine ljudi, pa tako, da so bili izbrani redkejši primeri aritmij. Baza je sestavljena iz približno 110.000 označenih primerov, z resolucijo 10mV. Frekvenca merjenja je 30Hz. Aparat, ki so ga uporabljali za merjenje EKG, je bil dvokanalen, zato so v podatkih le kanala V1 in odvod II.

Statistične podatke, ki nas zanimajo, smo predstavili v Tabeli 4.1. Podatki za vse posnetke pa so prikazani v dodatku A.1. Vidimo lahko, da je pri večini posnetkov večinski razred okno brez utripa. *Opomba: Število oken v posnetku je odvisno od velikosti oken. V omenjenih tabelah so bila uporabljena okna velikost 59.* Opazimo lahko tudi, da ima navadno ena oseba le eno vrsto aritmije, nekateri pacienti pa uporabljajo srčni spodbujevalnik (oznaka SPO).

Tabela 4.1: Statistika učnih podatkov

Oseba	Brez utripa	Normalen utrip	PAC	LBBB	RBBB	PVC	Vse	Komentar
100	7453	2238	33	0	0	1	34	
101	8227	1860	3	0	0	0	3	
102	312	99	0	0	0	4	4	spo
103	8061	2082	2	0	0	0	2	
104	226	163	0	0	0	2	2	spo
105	7091	2526	0	0	0	41	41	
106	7757	1507	0	0	0	520	520	
107	0	0	0	0	0	59	59	spo
108	8064	1739	4	0	0	17	21	
109	7478	0	0	2492	0	38	2530	
111	8054	0	0	2123	0	1	2124	
112	7584	2537	2	0	0	0	2	
113	8318	1788	0	0	0	0	0	
114	8093	1820	10	0	0	43	53	
115	8133	1953	0	0	0	0	0	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
Skupaj	320.268	75.048	2.546	8.075	7.259	7.130	25.010	

V prvi fazi testiranja smo iskali normalen srčni utrip (QRS-kompleks), takšnih primerov je tudi največ v podatkih. Naše metode smo preizkusili tudi pri iskanju srčnih bolezni.

4.2 Konvolucijske nevronske mreže

Analiziranje in klasifikacija bolezni s pomočjo računalnika sega v same začetke umetne inteligence. Po pričakovanjih je analiza srčnih bolezni še posebej popularna, saj smrti zaradi kardioloških vzrokov predstavljajo okrog 30 odstot-

kov vseh smrti na leto.

Zaradi popularnosti domene in pomembnosti odkritij obstajajo baze številnih označenih podatkov [46] ter tekmovanja, kjer strokovnjaki primerjajo rezultate svojih algoritmov. Strokovnjaki so se iskanja QRS-kompleksov ter prepoznavanja bolezni iz EKG lotili na številne načine, v zadnjih letih je tako klasifikacijska točnost za to domeno dosegla več kot 95% točnost, za najbolj pogoste vzroke bolezni. Naše rezultate smo primerjali na eni najpopularnejših baz podatkov [47] na način, ki zagotavlja regularno primerjavo rezultatov algoritmov.

V razdelku 2.3 smo ugotovili, da je večina bolezni povezanih ali z enakomernostjo srčnega utripa ali z obliko QRS-kompleksa. Zato smo se odločili, da bomo najprej poizkusili prepoznati QRS-kompleks, kar nam bo omogočilo nadaljno analizo signala.

Ob pogledu na EKG lahko ljudje enostavno prepoznamo obliko QRS-kompleksa ter z nekaj štetja določimo čas med dvema vrhoma RR. Glede na to, da ljudje na vizualen način (ni potrebe po štetju, računanju ipd.) prepoznavamo QRS-komplekse, smo se odločili, da se bomo problema prepoznavanja QRS-kompleksa lotili s konvolucijskimi nevronskimi mrežami (KNN).

4.2.1 Predobdelava podatkov

Večina algoritmov za strojno učenje ne deluje, če kot vhodne podatke podamo surove podatke iz meritev. Za večino algoritmov je potrebno surove podatke predelati z algoritmi, ki znajo iz njih izluščiti značilke (ang. features). Algoritmi za strojno učenje se nato učijo na značilkah. Zanimalo nas je, ali se KNN lahko učinkovito in zanesljivo uči kar iz surovih podatkov.

4.2.2 Velikost oken

Podatke smo najprej razrezali na okna različnih velikosti ter jih kot vhodne podatke podali v konvolucijsko mrežo. Zanimalo nas je, ali se mreža lahko nauči, ali ima v oknu QRS-kompleks ali ne.

Velikost okna smo merili kot odstotek dolžine povprečnega QRS-kompleksa. Največje okno je bilo 4 krat daljše, najmanjše pa 3 krat krajše kot povprečen QRS-kompleks. Večjih oken nismo testirali, saj so bila že skoraj tako velika kot RR-interval in poslednično skoraj vsa vsebovala QRS-kompleks. Manjša okna tudi nima smisla preverjati, saj je očitno, da se algoritem ne more naučiti oblike signala, če vidi le 1% tega. Izkazalo se je, da se algoritem najbolje obnese, ko je okno rahlo daljše kot QRS-kompleks. Ob razmisleku se to zdi smiselno, saj vsebuje najvišjo koncentracijo informacij, ki nas zanimajo, brez dodatnega šuma.

Učenje je potekalo na datotekah 100, 101, 103, 105, 106 in 108, pri validaciji in testiranju pa na datoteke 112, 113, 114. Okna smo naključno razdelili med validacijsko in testno množico v razmerju 1 : 1. Število učnih in testnih primerov je odvisno od velikosti okna, za okno velikosti 100% učna množica vsebuje 147.000 učnih primerov, od katerih jih je 12.200 pozitivnih, testna in validacijska množica pa vsebujeta vsaka po 42.000 primerov, od katerih jih je 3.050 pozitivnih. Obe množici sta torej vsebovale primere signala EKG iz vseh štirih posnetkov, a iz različnih časovnih intervalov. Število oken je pri vseh treh množicah odvisno od velikosti okna. Rezultati meritev so prikazani v Tabeli 4.2.

Tabela 4.2: Vpliv velikosti oken na klasifikacijsko točnost.

Velikost okna (%)	Napaka $\times 10^2$	Velikost okna (%)	Napaka $\times 10^2$
50	0,58	180	0,225
80	0,36	230	0,675
100	0,14	280	0,192
120	0,175	350	0,226
150	0,231	400	0,26

Preverili smo tudi, ali se je mreža sposobna naučiti, kje znotraj okna se nahaja najvišji vrh (R). Izkazalo se je, da ta problem ni rešljiv s KNN. Ko nas je v prihodnje zanimala točna lokacija vrha R, smo s pomočjo po-

stprocesiranja poiskali maksimum v oknu, kjer je bil zaznan QRS-kompleks. Reševanje tega problema je enostavno, zato je točnost iskanja vrha R enaka iskanju QRS-kompleksa.

4.2.3 Arhitektura mreže

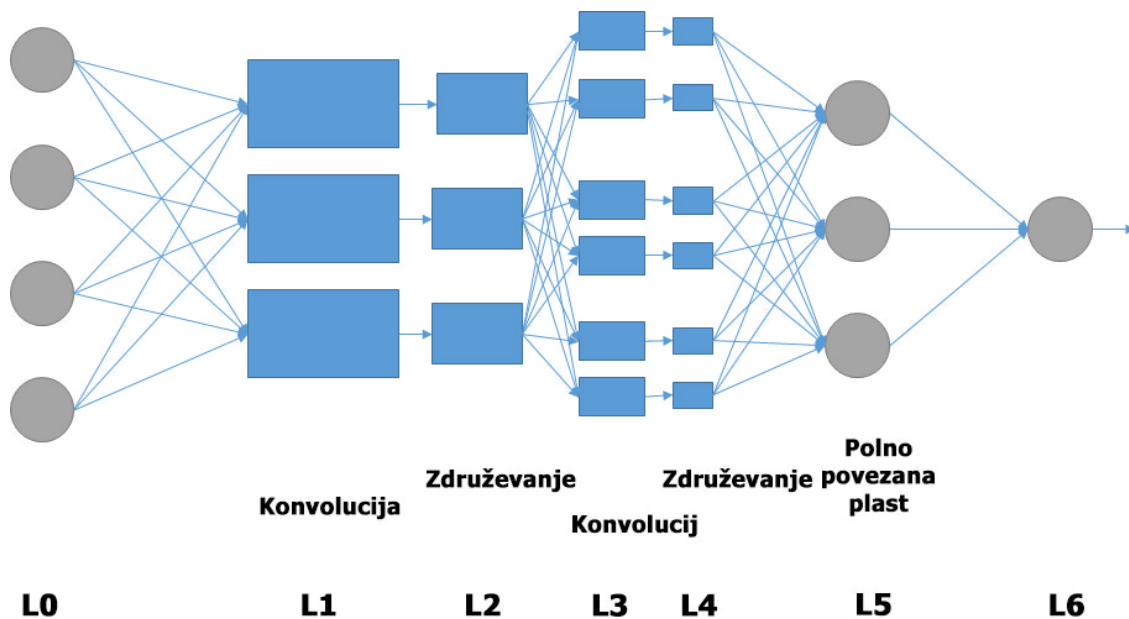
Zanimalo nas je, kako velikost in sama arhitektura mreže vplivata na točnost klasifikacije. Testirali smo tri velikosti mrež, s petimi, s sedmimi in z devetimi plastmi.

Vhodna plast je vedno velikosti okna, to pomeni, da ima toliko nevronov, kot je signalov v enem oknu. Sledi konvolucijska plast. Širino te plasti smo spreminjali od 10 do 100 (filtrov), velikost filtra pa je od 1×3 do 1×5 . Na konvolucijsko plast je povezana združitvena plast, ki ima enako število nevronov kot konvolucijska plast. Pri globlji mreži se ta par plasti ponovi. Predzadnja plast mreže je polno povezana mreža nevronov. Izhodni nevroni te plasti pa so povezani z multinomsko regresijo, ki nato klasificira podatke. Iz arhitekture je razvidno, da globoka nevrnska mreža išče značilke v podatkih, klasifikacija pa nato poteka z multinomsko regresijo, s tem pohitrimo čas učenja. Shema mreže, ki jo opišemo kot [3,6,3] je prikazana na Sliki 4.1.

Velikost mreže

Izkaže se, da je za dobro učenje potrebna globina vsaj sedmih plasti. Pri večjih (globljih) mrežah pogosto pride do problemov z izginjajočim gradientom ter dolgim časom učenja, končna klasifikacijska točnost pa ni veliko boljša. V naših testiranjih se je sicer izkazalo, da je najgloblja mreža dosega najboljše rezultate, a smo se zaradi časovne potratnosti in majhne izgube točnosti odločili, da bomo večino nadaljnjih testiranj opravili na mreži globine 7.

Število (novih) nevronov na plasti smo pustili za vse plasti enako. To pomeni, da so globlje plasti veliko širše, saj za vsak filter iz prejšnje plasti naredijo n novih nevronov. Opazimo, da pri prvi mreži z večanjem števila nevronov povečujemo točnost pri globljih pa dosežemo najboljšo točnost pri



Slika 4.1: Konvolucijska nevronska mreža s petimi skritimi plastmi.

manjšem številu nevronov. To nakazuje na preveliko prileganje (ang. overfitting) podatkov pri velikem številu nevronov na globokih mrežah.

V Tabeli 4.3 so vidni rezultati, kako globina in širina mreže vplivata na končno točnost. Učenje in testiranje je potekalo na podoben način kot v razdelku 4.2.2. Mreža se je učila iz podatkov 100, 101, 103, 105, 106 in 108, velikost oken pa je bila fiksirana na 59 signalov. Učna množica je vsebovala 11952 pozitivnih primerov in 46653 negativnih primerov oken. Okna za validacijsko in testno množico so bila naključno izbrana iz posnetkov 111, 112, 113 in 114, tako da je vsaka množica dobila polovico od 8098 pozitivnih ter 32128 negativnih primerov oken.

Za aktivacijsko funkcijo smo preizkusili tako sigmoido, hiperbolični tangens in funkcijo maks. Razlika v končni točnosti je bila minimalna, zanemarljiva. V nadaljevanju smo pri testiranjih uporabljali *tanh*.

Tabela 4.3: Povezanost med klasifikacijsko točnostjo in velikostjo mreže

Št. plasti	Št. novih nevronov na plast	Preci- znost	Senziti- vnost	Specifi- čnost	Točnost	Napaka $\times 10^2$
5	10	0,9968	0,9979	0,9991	0,9988	0,112
5	20	0,9966	0,9957	0,9990	0,9983	0,168
5	40	0,9960	0,9971	0,9988	0,9985	0,15
5	60	0,9979	0,9973	0,9994	0,999	0,1
5	80	0,9968	0,9991	0,9991	0,9991	0,0875
5	100	0,9991	0,9966	0,9997	0,9990	0,0937
7	10	0,9985	0,9980	0,9995	0,9992	0,075
7	20	0,9953	0,9987	0,9987	0,9986	0,131
7	40	0,9982	0,9991	0,9995	0,9994	0,0562
7	60	0,9985	0,9994	0,9996	0,9995	0,043
7	80	0,9970	0,9973	0,9992	0,9988	0,118
7	100	0,9985	0,9982	0,9995	0,9993	0,068
9	10	0,9951	0,9977	0,9986	0,9984	0,15
9	20	0,9976	0,9979	0,9993	0,9990	0,0937
9	40	0,9977	0,9991	0,9993	0,9993	0,068
9	60	0,9973	0,9979	0,9992	0,999	0,1
9	80	0,9971	0,9991	0,9992	0,9991	0,081
9	100	0,9971	0,9983	0,9991	0,999	0,1

Optimizacijske metode

Zaradi enostavnosti implementacije smo najprej zgradili ter implementirali optimizacijo SGD. Čas učenja je bil, kljub visoki paralelizaciji (1000 jeder obremenjenih 100% ves čas), izredno dolg, od nekaj ur za manjše mreže in majhno podmnožico učnih primerov, do nekaj dni za velike mreže. Najdaljše učenje je potekalo 10 dni. Da bi pohitrili učenje, smo implementirali ADAGrad, ADADelta in RMSProp.

Pri ADAGrad smo opazili, da je v začetku učenje tako zadušeno, da se

algoritem ustavi pri visoki napaki. Metodo ADAGrad smo nadgradili, kot smo opisali v razdelku 3.1.5, v našo verzijo ADADelta. Razlika je občutna, ADADelta v začelih fazah konvergira k rešitvi enako hitro kot SGD. Za konec smo implementirali še RMSProp. Izkazalo se je, da je ta metoda najboljša za našo vrsto učenja.

Učenje in testiranje je tokrat potekalo na vseh posnetkih pacientov, razen tistih kjer ima celoten posnetek manj kot 100 normalnih QRS-kompleksov. Vse tri množice so vsebovale posnetke iz različnih intervalov vseh datotek. Okna so bila med množice razdeljena naključno, polovica za učenje (180.000 oken), polovica za testiranje (90.000 oken) in validacijo (90.000 oken). Točno število oken se zaradi paralelizacije in lažje implementacije rahlo spreminja v odvisnosti od velikosti bloka, ki je uporabljen pri učenju.

V tabeli 4.4 je prikazano, kako se zmanjšuje napaka v prvih 15 iteracijah učenja z različnimi metodami optimizacije. Na Sliki 4.2 vidimo napako za prvih 120 iteracij. Zaradi velike napake in ustavitve učenja smo ADAGrad predčasno ustavili. ADADelta se obnese veliko bolje, skoraj tako dobro kot RMSProp. Iz slike je razvidno, da metode v veliki meri vplivajo na hitrost konvergence, na samo klasifikacijsko točnost pa nimajo velikega vpliva. RMSProp in ADADelta dosežeta enako (končno) točnost v 80 iteracijah.

4.2.4 Podatki, predstavljeni kot slika

Opazili smo, da naš algoritem kljub visoki točnosti ne dosega rezultatov ostalih state-of-the-art algoritmov.

V splošnem je KNN zelo dober za prepoznavanje slik. Odločili smo se, da bomo preverili, kako se obnese v primeru, ko okna predstavimo kot dvodimenzionalne slike. Sestavili smo okna kvadratne oblike, kjer smo po višini signal normalizirali, tako da je povprečen R vrh dosegel vrh okna. V primeru, da je bil signal višji od te točke, smo ga odrezali, tako da je dosegel vrh okna. Točke znotraj okna smo interpolirali, tako da smo dobili zvezen signal od začetka, do konca okna.

Opazili smo, da se zaradi dodatne dimenzije eksponentno zmanjša velikost

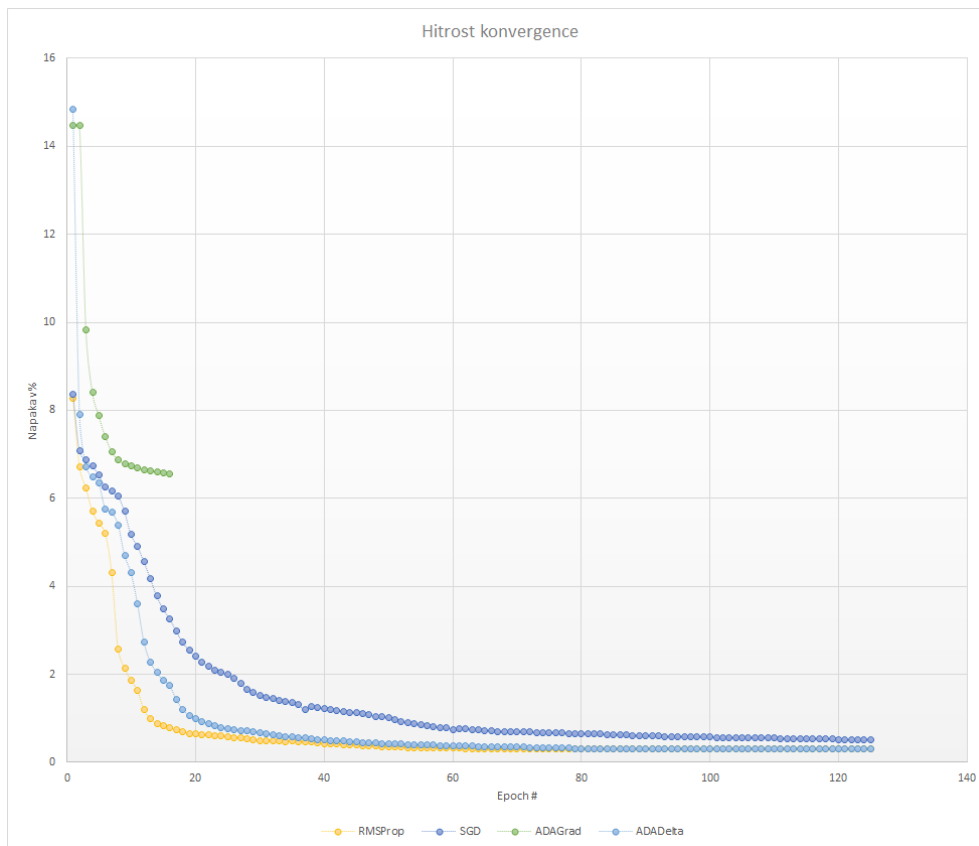
Tabela 4.4: Napaka v posameznih iteracijah z različnimi metodami posodabljanja gradienta.

Iteracija	SGD	RMSProp	ADAGrad	ADADelta
1	8,372	8,282	14,48	14,85
2	7,073	6,723	14,48	7,91
3	6,866	6,246	9,83	6,72
4	6,745	5,71	8,41	6,49
5	6,528	5,441	7,88	6,34
6	6,268	5,215	7,41	5,76
7	6,156	4,309	7,07	5,68
8	6,051	2,566	6,88	5,4
9	5,705	2,139	6,79	4,7
10	5,176	1,86	6,73	4,303
11	4,915	1,633	6,686	3,61
12	4,57	1,199	6,648	2,736
13	4,185	0,987	6,62	2,284
14	3,779	0,886	6,603	2,038
15	3,49	0,829	6,583	1,862

(širina) mreže, ki jo program lahko izračuna v enakem časovnem obdobju. Poleg tega so bili rezultati presenetljivo slabši. V povprečju je bila napaka okrog 30%. Ob premisleku, je to smiselno. V podatke smo prinesli dodaten šum (interpolacija med točkami), iz podatkov smo zaradi normalizacije in rezanja vrhov izgubljali informacije in kot zadnje, zaradi manjše mreže se mreža ni dovolj dobro naučila prepoznavanja podatkov.

4.2.5 Popravljeni podatki

Med testiranjem smo opazili, da se točnost med validacijo in testiranjem razlikuje več, kot je to običajno. Kot smo prej omenili, je učenje potekalo na treh disjunktnih množicah podatkov. Kljub temu, da so bili podatki



Slika 4.2: RMSProp se že v prvih 20 iteracijah bolj približa rešitvi kot SGD v 120.

snemani laboratorijsko s kar se da malo šuma, pri meritvah EKG že majhno odstopanje od pravilne postavitve elektrode močno vpliva na signal, ki ga dobimo. Poleg tega se med daljšimi meritvami elektrode včasih premaknejo (zaradi gibanja človeka), zaradi česar se cel signal od takrat naprej poveča ali zmanjša (npr. je višji za 1mV).

Problema smo se lotili tako, da smo vse podatke iz učne domene obdelali. Posnetke vsakega pacienta smo razdelili na vnaprej določeno dolžino oken ter jih označili, ali vsebuje QRS-kompleks ali ne. Nato smo združili vsa okna, vseh pacientov v skupno množico, katero smo naključno razdelili na učno, validacijsko in testno, v razmerju 2 : 1 : 1. V učni množici smo tako imeli

180.000 učnih primerov, v testni in validacijski pa v vsaki 90.000. Ker so bila okna izbrana naključno, so vse tri množice vsebovale okna iz vsakega posnetka pacienta, a drugega časovnega intervala. To smo lahko naredili zato, ker nas zanima sama oblika signala v določenem oknu, ta oblika pa (s strani našega algoritma) ni odvisna od oblike v ostalih oknih.

Da to res drži, si oglejmo analogni, bolj razumljiv, primer iz domene prepoznavanja števil. Imamo množico slik, na katerih so narisane številke. To množico naključno razdelimo na tri podmnožice, na katerih testiramo in učimo algoritem. Ne glede na to, kako smo izbrali množico za določeno številko, to ne vpliva na končno točnost metode. Če slike števk, iz tega primera, nadomestimo s slikami iz EKG oken, torej QRS-kompleks, dobimo enak način učenja in testiranja na domeni EKG.

Pri tem se moramo zavedati, da bo algoritem imel nekaj primerov istega posnetka (ob različnih časih) v vseh treh množicah. Pri primeru s številkami bi to pomenilo, da bi različne številke, ki jih je ena oseba napisala, razdelili v vse tri množice. Algoritem bi tako v učni množici videl stil (rokopis) te osebe. Menimo, da to ne zmanjša vrednosti našega rezultata, tudi človek se pri reševanju klasifikacijskih problemov sklicuje na predhodno (že videno) znanje, če je nov primer preveč različen od že videnih, ga ne bo znal pravilno klasificirati. Boljša rešitev bi bila prečno preverjanje (ang. cross-validation), rezultati takega načina preverjanja so predstavljeni v naslednjem razdelku. Zaradi dolgih časov učenja, tako velike (kompleksne) mreže ni bilo mogoče testirati s prečnim preverjanjem.

Ta pristop se je izkazal za izjemno uspešnega. Validacijska in testna točnost sta bili zelo blizu. Zaradi dodatnega šuma v podatkih je učenje sicer trajalo veliko dlje. Opazili smo, da so rezultati v tem primeru boljši, če je mreža globlja in širša.

S to obliko mreže smo dosegli najboljšo klasifikacijsko točnost za prepoznavanje QRS-kompleksov. Dosegli smo klasifikacijsko točnost 99,95%. Izrisali smo si napačno klasificirane primere (Slika 4.3), opazili smo, da so nekateri med njimi napačno označeni. Sklepamo lahko, da je naš algoritem

dosegel točnost, ki je vsaj tako dobra, kot tista, ki jo dosežeta dva zdravnika, ki ročno označujeta te podatke. V tabeli 4.5 so prikazani rezultati najboljših testov. Širina plasti označuje velikost parov konvolucijske in združitvene plasti.

Tabela 4.5: Rezultati najboljšega učenja

Širina plasti	Iteracije	Čas učenja	Validacijska napaka $\times 10^2$	Testna napaka $\times 10^2$	Absolutna napaka
[500, 46, 23]	2000	70min	0,127	0,120	//
[23, 46, 500]	2000	280min	0,087	0,0844	152 / 180k
[230, 460, 500]	2000	4495min	0,057	0,0478	93 / 190k

Na Sliki 4.4 je predstavljena površina pod krivuljo ROC ¹. Dobili smo jo tako, da smo pri klasifikaciji, v zadnji plasti, spreminjali verjetnost, ki mora veljati za klasifikacijo v razred 1. Na Sliki 4.5 pa je predstavljeno razmerje med različnimi meritvami točnosti, ko tako spreminjamo verjetnost za klasifikacijo v razred 1.

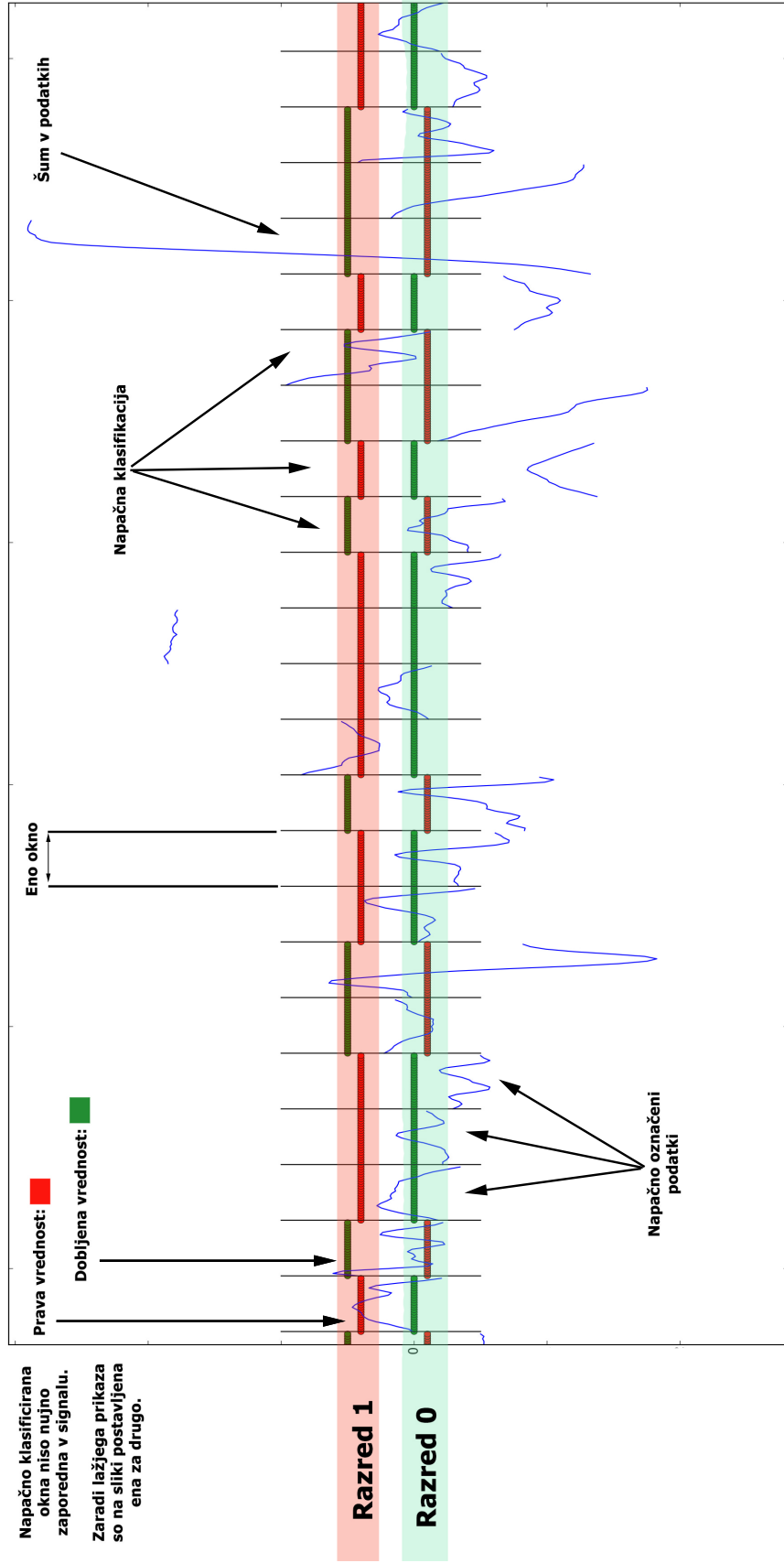
4.2.6 Dodajanje šuma

V literaturi [48] smo prebrali, da se z dodajanjem šuma točnost algoritma lahko še poveča. Odločili smo se, da dodamo dve vrsti šuma. Vsako okno, smo za $\pm 200\%$ (po povprečni višini okna) naključno premaknili po osi y . Ta šum preprečuje, da bi se algoritem naučil enostavno pravilo za klasifikacijo, na podlagi višine vrhov R (vsi vrhovi R pred šumom so bili višji od 0.5). Druga vrsta šuma je bil šum samega signala. Vsako vrednost znotraj okna smo za $\pm 1\%$, premaknili po osi y , s tem smo dodali še nekaj dodatnega šuma k meritvam ter rahlo popačili oblike signalov.

Izkazalo se je, da so bili rezultati malenkost slabši, a še vedno boljši od ostalih state-of-the-art metod. Z dodatnim šumom v podatkih smo dosegli klasifikacijsko napako 0.101%, kar je minimalno slabše od čistih podatkov.

¹ang. Receiver operating characteristic

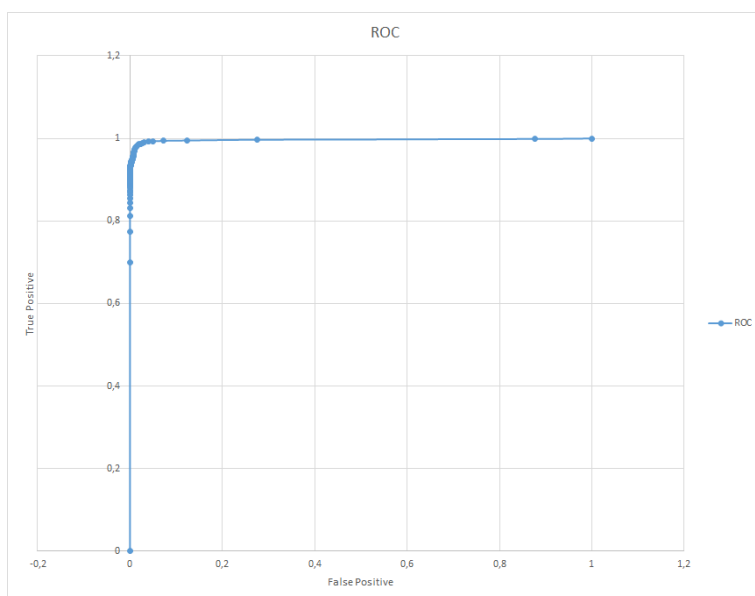
Napačno klasificirani primeri



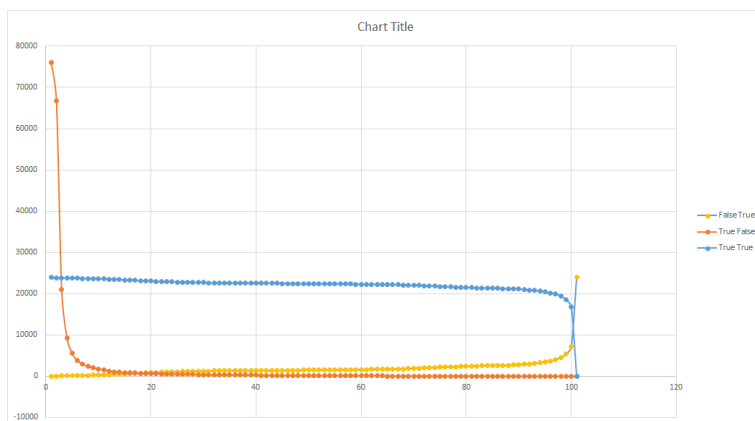
Napačno klasificirana okna niso nujno zaporedna v signalu. Zaradi lažjega prikaza so na sliki postavljena ena za drugo.

Slika 4.3: Del napačno klasificiranih oken. Vidimo lahko, da je veliko napačno klasificiranih oken narobe označenih.

Opomba: Okna so združena na eno sliko zaradi prikaza, v podatkih so med njimi še ostala okna.



Slika 4.4: Ploščina pod krivuljo ROC za našo mrežo RNN. Dobili smo jo tako, da smo spreminjali mejno verjetnost, da je primer označen kot pozitiven razred.



Slika 4.5: Razmerje med različnimi merami točnost, ko spreminjamo mejno verjetnost, da je primer označen kot pozitiven razred.

4.2.7 Klasifikacija bolezni

Poleg zaznavanja QRS-kompleksa nas zanima, kako se algoritem obnese pri prepoznavanju bolezni, ki jih lahko razberemo iz EKG. Spremenili smo označevanje oken iz binarnega na 6 razredov (0-5). Pri tem smo merili dve točnosti, samo zaznavanje QRS-kompleksa, tudi če je ta popačen, ter prepoznavanja bolezni, razvidne iz kompleksa. Pri tem velja omeniti, da neenakomernega utripa ni mogoče ugotoviti na ta način, je pa vseeno vštet v našo napako. Izbira učnih primerov je potekala na enak način kot v razdelku 4.2.5.

V Tabeli 4.6 opazimo, da so rezultati zaznavanja QRS-kompleksa celo boljši, kot tisti, predstavljeni v prejšnjem razdelku. Samo prepoznavanje bolezni je nekoliko slabše, a še vedno zelo dobro.

Tabela 4.6: Rezultati iskanja bolezni v signalu EKG.

Širina plasti	Čas učenja	Napaka bolezni $\times 10^2$	Napaka QRS $\times 10^2$	Število primerov
Okno velikosti 49				
[23,46,500]	280min	0,469	//	120k
[49,50,500]	413min	0,195	//	120k
[100,250,500]	1960min	0,193	//	120k
Okno velikosti 59				
[59, 59, 500]	482min	0,213	0,038	100k
[259,259, 500]	2263min	0,188	0,033	100k
[259, 259, 500]	5dni	0,17	0,027	100k

4.2.8 Primerjava rezultatov z ostalimi metodami

Učno množico MIT-BIH smo izbrali tudi zaradi tega, ker obstaja veliko člankov, ki so na tej množici testirali svoje algoritme za iskanje QRS-kompleksov ter prepoznavanja bolezni, razvidnih iz signala EKG. Pri primerjavi rezultatov smo se potrudili, da smo imeli testno množico enako.

Zaradi dolgotrajnosti postopka testiranja in ponovnega učenja za vsak posnetek posebej smo uporabili relativno majhno mrežo [59, 60, 60, 40, 40, 20]. Učenje in validacija sta potekali na 20% (80.000) in 5% (20.000) naključno izbranih disjunktnih oken, ki smo jih dobili iz vseh posnetkov razen testnega. Učna in validacijska množica sta tako vsebovali okna iz različnih časovnih intervalov enakih pacientov, testiranje pa je potekalo na še ne videnem novem pacientu. Takemu tipu testiranja pravimo prečno preverjanje leave-one-patient-out. Število oken za posamezen posnetek je zapisano v Tabeli A.1. Učenje smo prekinili pri 200. iteraciji. Model smo testirali samo na testnem posnetku. Rezultati testov so prikazani v Tabelah 4.7 in 4.8.

Iz rezultatov lahko razberemo, da smo kljub skrajšanemu času učenja dosegli podobno dobre rezultate kot v prejšnjem razdelku. Pri nekaterih posnetkih lahko opazimo občutno manjšo točnost (primer posnetek 113), v teh podatkih je več šuma ter narobe označenih primerov, če bi mrežo učili več časa, bi bila tudi tam točnost višja. *Opomba: Posnetke pacientov, ki uporabljajo srčni spodbujevalnik, nismo upoštevali v učenju ter testiranju. Menimo, da ti posnetki niso zanimivi iz vidika strojnega učenja, saj so vsi kompleksi v njih umetno ustvarjeni. Podobno so storili avtorji člankov, s katerimi primerjamo rezultate.*

V članku [49] avtor s pomočjo podobnosti valov in dolžino intervala RR klasificira 6 različnih tipov utripa. Za učenje uporablja tudi prvih 5 minut vsakega posnetka, zato se število utripov med njegovimi in našimi primeri rahlo razlikuje. Naše učenje poteka na vseh datotekah razen tisti, na kateri testiramo klasifikacijsko točnost. Druga razlika med našim in avtorjevim testiranjem je, da mi nismo prepoznavali neenakomernih utripov tipa PAC, ampak smo jih označili kot normalen utrip. Avtor je dosegel klasifikacijsko točnost 97,5%, v članku je opisana tudi točnost za vsak posamezen posnetek.

Z uporabo nevronske mreže so QRS-kompleks poskušali prepoznati avtorji članka [50]. Za učenje so uporabili 20 posnetkov, za testiranje pa ostalih 10. Katere posnetke so uporabili za kaj, ni jasno napisano. Dosegli so povprečno točnost 99,01%.

Tabela 4.7: Iskanje QRS-kompleksa v posameznih posnetkih. [1/2]

Posnetek	Preciznost	Senzitivnost	Specifičnost	Točnost	Napaka
100	1,0	1,0	1,0	1,0	0,0
101	0,9989	0,9994	0,9997	0,9997	0,0003
103	1,0	1,0	1,0	1,0	0,0
105	0,9968	0,9976	0,9989	0,9986	0,0014
106	0,9980	0,9971	0,9994	0,9990	0,0010
108	0,9984	0,7400	0,9997	0,9552	0,0448
109	0,9996	1,0	0,9998	0,9999	0,0001
111	1,0	0,9544	1,0	0,9905	0,0095
112	1,0	1,0	1,0	1,0	0,0
113	0,8440	1,0	0,9602	0,9673	0,0327
114	1,0	0,8441	1,0	0,9730	0,0270
115	1,0	1,0	1,0	1,0	0,0
116	1,0	1,0	1,0	1,0	0,0
117	1,0	0,9479	1,0	0,9924	0,0076
118	0,9918	1,0	0,9973	0,9980	0,0020
119	1,0	0,9063	1,0	0,9810	0,0190
121	1,0	0,5761	1,0	0,9232	0,0768
122	1,0	1,0	1,0	1,0	0,0
123	0,9986	1,0	0,9997	0,9998	0,0002
124	1,0	0,9885	1,0	0,9982	0,0018
Povprečje	0,9913	0,9475	0,9977	0,9887	0,0112
Mediana	1	0,9997	1	0,9988	0,0012

V člankih [51] in [52] so avtorji ločeno testirali iskanje LBBB, RBBB ter PVC, zato smo v Tabelah 4.9, 4.10 in 4.11 prikazali točnost naše metode za tiste datoteke, ki so vsebovale vsaj 100 primerov iskanih bolezni. Pri tem velja omeniti, da smo morali zmanjšati velikost bloka na 10, v nasprotnem primeru se mreža zaradi majhnega števila primerov ni znala naučiti pravilne

Tabela 4.8: Iskanje QRS-kompleksa v posameznih posnetkih. [2/2]

Posnetek	Preciznost	Senzitivnost	Specifičnost	Točnost	Napaka
200	0,9879	1,0	0,9948	0,9964	0,0036
201	1,0	1,0	1,0	1,0	0,0
202	1,0	1,0	1,0	1,0	0,0
203	0,9922	0,9940	0,9960	0,9954	0,0046
205	1,0	0,9985	1,0	0,9996	0,0004
207	0,9950	0,9449	0,9987	0,9872	0,0128
208	1,0	0,9887	1,0	0,9962	0,0038
209	0,9957	1,0	0,9979	0,9986	0,0014
210	0,9963	0,9963	0,9986	0,9980	0,0020
212	0,9979	1,0	0,9991	0,9994	0,0006
213	0,9968	1,0	0,9980	0,9988	0,0012
214	0,9965	0,9982	0,9989	0,9988	0,0012
215	0,9982	1,0	0,9990	0,9994	0,0006
219	1,0	1,0	1,0	1,0	0,0
220	1,0	1,0	1,0	1,0	0,0
221	1,0	1,0	1,0	1,0	0,0
222	1,0	0,8954	1,0	0,9724	0,0276
223	0,9992	1,0	0,9997	0,9998	0,0002
228	0,9943	0,9888	0,9984	0,9964	0,0036
230	1,0	1,0	1,0	1,0	0,0
231	1,0	1,0	1,0	1,0	0,0
Povprečje	0,9976	0,9907	0,99903	0,9969	0,00303
Mediana	0,9992	1	0,9997	0,9994	0,0006

klasifikacije.

Pri iskanju LBBB je imela naša metoda težave z ločevanjem LBBB in PVC. Signala anomalije sta na sledeh, uporabljenih v bazi MIT-BIH, zelo podobni, predvidevamo, da bi večja mreža bila sposobna bolj pogosto ločevati

ta dva primera.

Tabela 4.9: LBBB

Posnetek	Preciznost	Senzitivnost	Specifičnost	Točnost	Napaka
109	0,9853	1,0	0,9950	0,9963	0,0037
111	0,9979	0,9524	0,9994	0,9897	0,0103
207	0,9073	0,9888	0,9779	0,9798	0,0201
214	0,8891	0,9305	0,9700	0,9620	0,0380
Povprečje	0,9449	0,9679	0,9856	0,9819	0,0180
Mediana	0,9463	0,9706	0,9864	0,9847	0,0152

Tabela 4.10: RBBB

Posnetek	Preciznost	Senzitivnost	Specifičnost	Točnost	Napaka
118	0,9952	0,9933	0,9985	0,9973	0,0026
124	0,9880	0,5013	0,9988	0,9219	0,0780
212	0,9989	0,9956	0,9997	0,9989	0,0010
231	1,0	0,9672	1,0	0,9936	0,0063
Povprečje	0,9955	0,8643	0,9992	0,9779	0,0220
Mediana	0,9970	0,9803	0,9993	0,9954	0,0045

V članku [51] so avtorji s pomočjo nevronske mreže ter metode ICA, iskali normalen, LBBB, RBBB in PVC utrip. Iz posameznega posnetka so vzeli od 50 do 100 oken enega tipa utripa, tako za učno kot testno množico. Kot smo že omenili, mi za učenje nismo uporabili nobenega okna pacienta, pri katerem smo testirali klasifikacijsko točnost. Za enake tipe bolezni, kot smo jih testirali mi, so avtorji dosegli povprečno točnost 98.7%.

V članku [52] ni jasno razvidno na katerih posnetkih so avtorji testirali svoj algoritem. Dosegli so klasifikacijsko točnost 95,69%.

Tabela 4.11: PVC

Posnetek	Preciznost	Senzitivnost	Specifičnost	Točnost	Napaka
106	1,0	0,1295	1,0	0,9664	0,0336
119	1,0	0,0049	1,0	0,9598	0,0402
200	0,9972	0,9280	0,9997	0,9942	0,0058
203	0,2500	0,7098	0,8855	0,8766	0,1234
207	0,9333	0,1386	0,9997	0,9824	0,0176
208	1,0	0,9566	1,0	0,9938	0,0062
213	0,9457	0,7820	0,9985	0,9918	0,0082
214	0,9500	0,6834	0,9989	0,9902	0,0098
221	1,0	0,0120	1,0	0,9508	0,0492
223	0,6730	0,2083	0,9964	0,9700	0,0300
228	1,0	0,8837	1,0	0,9950	0,0050
Povprečje	0,8863	0,4942	0,9890	0,9700	0,0299
Mediana	0,9972	0,6834	0,9997	0,9824	0,0176

4.3 Povratne nevronske mreže

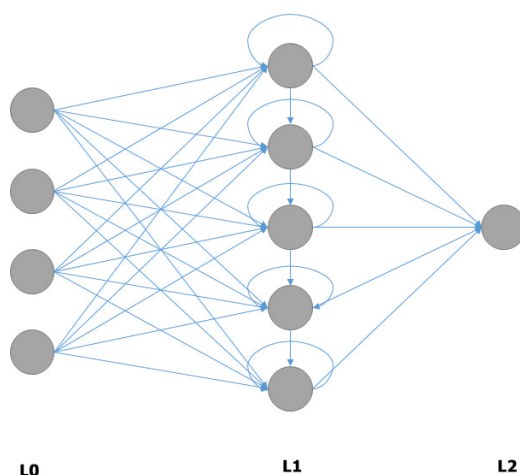
V prejšnjem razdelku smo pokazali, da s konvolucijskimi mrežami dosegamo dobre rezultate na domeni EKG, a le v primeru, ko je signal v določenem delu tako popačen, da ga mreža lahko razlikuje od normalnega signala. Takšna mreža pa ne bo nikoli sposobna zaznati neenakomernega srčnega utripa ter ostalih bolezni, ki so povezane s časovno odvisnostjo elementov v signalu.

V razdelku 3.3 smo povedali, da so povratne nevronske mreže še posebej dobre za analiziranje signala, kjer so podatki odvisni med seboj in je potrebno vedeti, kakšen je ta signal bil pred nekaj sekundami.

4.3.1 Arhitektura

Mrežo smo sestavili iz treh plasti: vhodna, izhodna ter skrita. Od ostalih nevronskih mrež se loči po tem, da njena skrita plast vsebuje povezave znotraj

plastii. Na ta način smo dobili povratno nevronske mrežo. Za aktivacijsko funkcijo smo ponovno uporabili \tanh . Shema mreže je prikazana na Sliki 4.6.

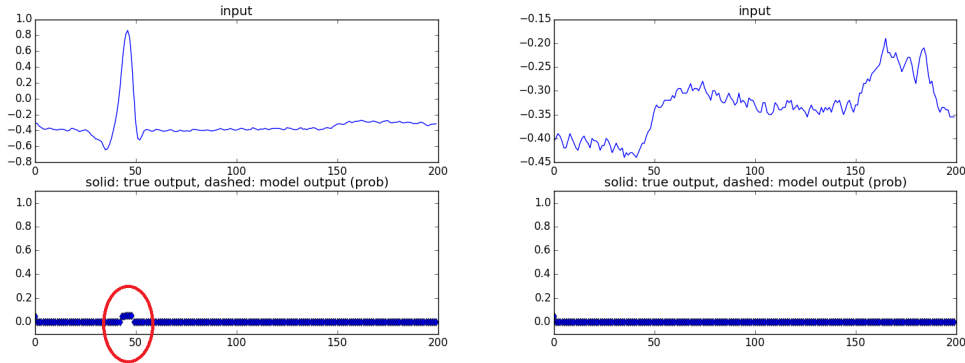


Slika 4.6: Shema naše povratne nevronske mreže s 5 skritimi nevroni.

4.3.2 Iskanje QRS-kompleksa

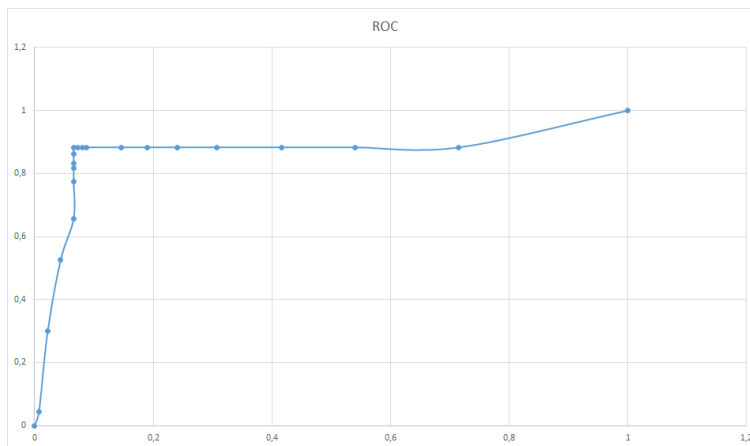
Podatke smo podobno kot pri KNN razdelili na okna. Tokrat smo morali biti pozorni, da okna med sabo nismo premešali, saj imajo prejšnja okna vpliv na vrednosti v trenutnih. Učne podatke smo izbrali iz prvih 24 posnetkov, testirali pa smo na naključnih petih datotekah med 200 in 230. Ob dovolj velikem številu nevronov se je mreža uspela naučiti prepoznavati QRS-komplekse.

Učenje je bilo počasnejše kot pri KNN, končni rezultat pa rahlo slabši. Dosegli smo povprečno klasifikacijsko točnost 97%, senzitivnost 0,88 ter specifičnost 0,98. Na Sliki 4.7 sta predstavljena pravilno klasificiran QRS-kompleks ter signal vmes. Graf ploščine pod krivuljo ROC (AUC) je predstavljen na Sliki 4.8, dobili smo ga tako, da smo spreminjali verjetnost, kdaj je signal označen za pozitivni razred. Verjetnost je prikazana na Sliki 4.7a, kot višina spodnjega signala (označen z rdečo).



(a) Pravilno ugotovljen QRS-kompleks. (b) Pravilna pozitivna klasifikacija.

Slika 4.7: Pravilna klasifikacija

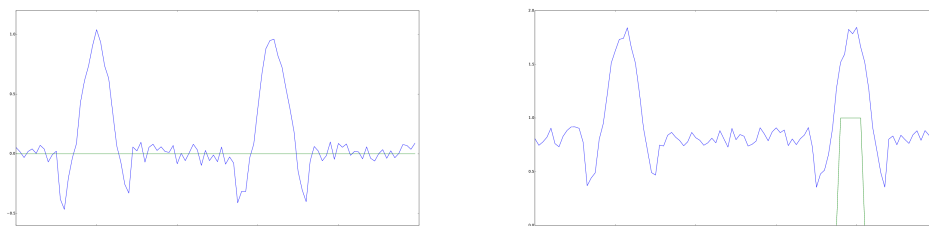


Slika 4.8: Ploščina pod krivuljo ROC (AUC) za RNN.

4.3.3 Prepoznavanje neenakomerne utripa

Izkazalo se je, da so razdalje med dvema vrhovoma R predolge, da bi se jih RNN še lahko zapomnil in uspešno naučil. Poleg tega smo imeli težave, saj nismo imeli dovolj učnih podatkov. Odločili smo se, da bomo to vrsto podatkov sintetizirali. Naredili smo umetne QRS-komplekse, ki so bili med seboj oddaljeni približno polovico razdalje med pravimi. Razdaljo med dvema kompleksoma smo določili naključno, znotraj 40% (umetnega) intervala RR. Primer normalnega sintetiziranega utripa je prikazan na Sliki 4.9a.

Na drugi Sliki 4.9b je primer zakasnelega utripa. Z zeleno črto je označena verjetnost, da gre za neenakomeren utrip. Pri drugem utripu je ta verjetnost 1, kar predstavlja tudi klasifikacijski razred.

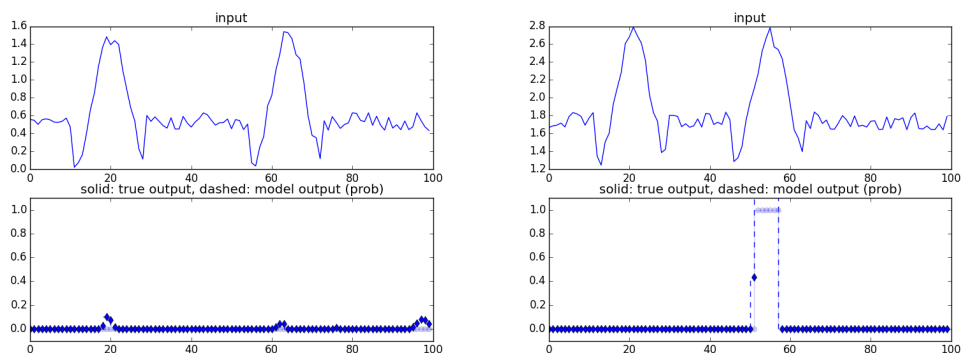


(a) V tem primeru imamo normalen utrip. (b) V tem primeru imamo zapozneli utrip, zato je drugi utrip razreda 1.

Slika 4.9: Zelena črta predstavlja klasifikacijski razred.

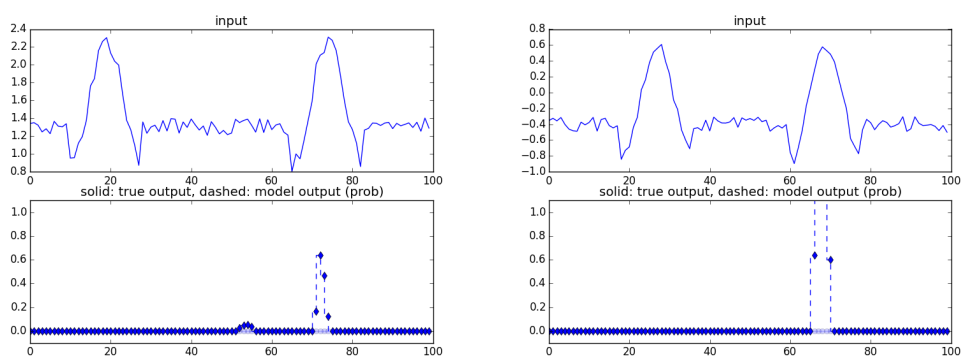
V primeru, ko sta dva QRS-kompleksa preblizu ali predaleč (20% okrog robov naključnega intervala), vrh označimo kot neenakomeren utrip. Izkazalo se je, da se je RNN sposoben naučiti najprej prepoznavati QRS-kompleks, nato pa še, ali je razdalja med dvema sosednjima kompleksoma pravilna. Na Slikah 4.10 ter 4.11 so prikazani primeri pravilne ter napačne klasifikacije. V spodnjem delu slike je prikazana verjetnost, ali gre za neenakomeren utrip ali ne.

V tabeli 4.12 so prikazani rezultati pri prepoznavanju neenakomernega utripa. Opazimo večjo razliko med senzitivnostjo ter specifičnostjo, kot smo jo navajeni iz prejšnjih primerov. Najboljšo točnost doseže mreža z 200 skritimi nevroni.



(a) Pravilna negativna klasifikacija. (b) Pravilna pozitivna klasifikacija.

Slika 4.10: Pravilna klasifikacija



(a) Napačna negativna klasifikacija. (b) Napačna pozitivna klasifikacija.

Slika 4.11: Napačna klasifikacija

Tabela 4.12: Rezultati detekcije neenakomernega utripa na umetno ustvarjenih signalih.

Št. nevronov	Senzitivnost	Specifičnost	Klas. točnost
150	0,928	0,965	0,929
200	0,945	0,974	0,947
400	0,918	0,960	0,918

Poglavje 5

Zaključek

Cilj magistrske naloge je bil pregled področja globokih nevronske mreže v strojnem učenju ter primerjava, kako se obnesejo proti ostalim metodam strojnega učenja. Učenje in testiranje je potekalo na domeni odkrivanja srčnih bolezni, kjer globoke nevronske mreže še niso bile preizkušene.

Reševanja problema smo se lotili v dveh korakih. Najprej smo z uporabo konvolucijskih nevronske mreže iskali QRS-komplekse, ki so pomembni za določanje ritma utripa srca. Veliko bolezni srca je povezanih s problemi prevažanja električnih signalov v srcu, zaradi česar jih je veliko mogoče določiti že iz QRS-kompleksa. Ugotavljali smo, kako se globina ter širina mreže odražata v končni klasifikacijski točnosti. Zaradi kompleksnosti mreže smo implementirali različne optimizacijske metode, s katerimi smo skrajšali čas učenja.

Naša najboljša mreža je dosegla končno klasifikacijsko točnost, primerljivo s strokovnjaki iz področja (zdravniki). Poleg detekcije QRS-kompleksov smo iskali tudi bolezni, ki so razvidne iz QRS-kompleksa (LBBB, RBBB). Klasifikacijska točnost pri teh primerih je bila nekoliko slabša kot pri iskanju QRS-kompleksov, a vseeno za odstotek boljša od ostalih state-of-the-art algoritmov, s katerimi smo primerjali naše rezultate.

V drugem delu testiranja smo iskali bolezni, ki se izražajo z neenakomernim srčnim utripom. Implementirali smo povratne nevronske mreže, ki lahko

prepoznajo časovno odvisnost med podatki. Tudi ta tip mreže se je uspešno naučil prepoznavanja QRS-kompleksov. Pri iskanju neenakomernih srčnih utripov se je izkazalo, da so le ti preveč narazen, da bi se jih tak tip mreže uspel naučiti. Algoritem smo zato preizkusili na sintetiziranih podatkih, kjer so bili tipi valov na krajšem časovnem intervalu. Izkazalo se je, da se jih RNN lahko nauči prepoznavati. Klasifikacijska točnost je bila okrog 94%.

5.1 Nadaljnje delo

Iz rezultatov sodeč lahko sklepamo, da so globoke nevronske mreže med najboljšimi tipi strojnega učenja, ki ga lahko uporabimo za klasifikacijo srčnih bolezni iz signala EKG. Kljub vsemu nam v naši implementaciji ni uspelo doseči zelene klasifikacijske točnosti za vse iskane bolezni.

Reševanje problema z dolgoročnim spominom v RNN smo že opisali v razdelku 3.3.1. Na ta način se lahko odpravi pomanjkljivost RNN. V delu smo pokazali, da se globlje mreže bolje odnesejo od kratkih a širokih, poleg tega smo pri KNN pokazali, da so značilke, ki se jih mreža sama nauči prepoznavati, boljše od tistih, katere poda človek.

Predlagamo združitev KNN ter LSTM v eno globoko mrežo. V prvem delu mreže bi plasti KNN iz podatkov izluščile dobre značilke, v drugem delu pa bi z uporabo LSTM (RNN) prepoznavali časovno odvisnost med značilkami.

Literatura

- [1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [2] G. Hinton, L. Deng, D. Yu, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. G. Dahl, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition, *IEEE Signal Processing Magazine* 29 (6) (2012) 82–97.
- [3] D. Palaz, R. Collobert, M. Magimai.-Doss, Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks, eprint arXiv:1304.1018.
- [4] J. G. Pucer, Avtomatsko razlikovanje patoloških in nepatoloških sprememb EKG signala, Doktorska disertacija, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2013.
- [5] B.-U. Kohler, C. Hennig, R. Orglmeister, The principles of software QRS detection, *Engineering in Medicine and Biology Magazine, IEEE*.
- [6] K. So, H.H.and Chan, Development of QRS detection method for real-time ambulatory cardiac monitor, *Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*.
- [7] J. Pan, W. J. Tompkins, A real-time QRS detection algorithm, *IEEE Transactions on biomedical engineering*, vol. bme-32, no. 3.

- [8] N. M. Arzeno, Z.-D. Deng, C.-S. Poon, Analysis of first-derivative based qrs detection algorithms, Biomedical Engineering, IEEE Transactions on 55 (2) (2008) 478–484.
- [9] J. G. Kemmelings, A. C. Linnenbank, S. L. Mulwijk, A. SippensGroenewegen, A. Peper, C. Grimbergen, et al., Automatic qrs onset and offset detection for body surface qrs integral mapping of ventricular tachycardia, Biomedical Engineering, IEEE Transactions on 41 (9) (1994) 830–836.
- [10] R. Silipo, C. Marchesi, Artificial neural networks for automatic ecg analysis, Signal Processing, IEEE Transactions on 46 (5) (1998) 1417–1425.
- [11] N. V. Thakor, Y.-S. Zhu, Applications of adaptive filtering to ecg analysis: noise cancellation and arrhythmia detection, Biomedical Engineering, IEEE Transactions on 38 (8) (1991) 785–794.
- [12] R. V. Andreão, B. Dorizzi, J. Boudy, Ecg signal analysis through hidden markov models, Biomedical Engineering, IEEE Transactions on 53 (8) (2006) 1541–1549.
- [13] W. Commons, Diagram of the human heart (2006) [obiskano 20-8-2015].
URL [https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_\(cropped\).svg](https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg)
- [14] B. Blaus, Diagram of the human heart, blausen.com staff. Blausen gallery 2014. Wikiversity Journal of Medicine. DOI:10.15347/wjm/2014.010. ISSN 20018762 (2013).
- [15] W. Commons, Basic cardiac action potential (2010) [obiskano 20-8-2015].
URL https://commons.wikimedia.org/wiki/File:Action_potential_ventr_myocyte.gif

- [16] H. T. Shih, Anatomy of the action potential in the heart., Texas Heart Institute Journal 21.1 (1994): 30–41. Print.
- [17] Electrical conduction system of the heart [obiskano 20-8-2015].
URL <http://en.academic.ru/pictures/enwiki/67/ConductionsystemoftheheartwithouttheHeart.png>
- [18] W. Commons, A graphical representation of the electrical conduction system of the heart (20013) [obiskano 20-8-2015].
URL <https://en.wikipedia.org/wiki/File:ConductionsystemoftheheartwithouttheHeart-en.svg>
- [19] F. G. Yanowitz, Introduction to ECG interpretation v8.0, 2012.
- [20] Emtresource, Plaques of lambda phages on e. coli x11-blue mrf, file: 12-lead-ecg-placement.png [obiskano 20-8-2015].
URL <http://www.emtresource.com/resources/ecg/12-lead-ecg-placement/>
- [21] W. Commons, Schematic diagram of normal sinus rhythm for a human heart as seen on ECG (2007) [obiskano 20-8-2015].
URL <https://commons.wikimedia.org/wiki/File:SinusRhythmLabels.svg>
- [22] LBBB ECG [obiskano 20-8-2015].
URL <http://rebelem.com/wp-content/uploads/2013/11/LBBB-with-normal-postassium.png?380a45>
- [23] E. Burns, Righth bundle branch block, file: rbbb3.png [obiskano 20-8-2015].
URL <http://lifeinthefastlane.com/ecg-library/basics/right-bundle-branch-block>
- [24] W. Commons, A drawn image of premature atrial contraction on ii-lead ECG [obiskano 20-8-2015].
URL <https://commons.wikimedia.org/wiki/File:PAC.png>

- [25] E. Burns, Premature junctional contraction, file: pjc.jpg [obiskano 20-8-2015].
URL <http://lifeinthefastlane.com/ecg-library/basics/right-bundle-branch-block>
- [26] E. Burns, Premature ventricular complex, file: multifocal-PVC.jpg [obiskano 20-8-2015].
URL <http://lifeinthefastlane.com/ecg-library/basics/pvc/>
- [27] E. Burns, Junctional escape rhythm, file: regularised-AF.jpg [obiskano 20-8-2015].
URL <http://lifeinthefastlane.com/ecg-library/basics/pvc/>
- [28] E. Burns, Ventricular flutter, file: v-flutter4.jpg [obiskano 20-8-2015].
URL <http://lifeinthefastlane.com/ecg-library/ventricular-flutter/>
- [29] J. Heuser, Scheme of atrial fibrillation (2005) [obiskano 20-8-2015].
URL https://commons.wikimedia.org/wiki/File:Afib_ecg.jpg
- [30] Flutter [obiskano 20-8-2015].
URL <http://sitemaker.umich.edu/ecgtutorial/files/flutter3.jpg>
- [31] A. D. Bernstein, J.-C. Daubert, R. D. Fletcher, D. L. Hayes, B. Luderitz, D. W. Reynolds, M. H. Schoenfeld, R. Sutton, The revised NASPE/BPEG generic code for antibradycardia, adaptive-rate, and multisite pacing., in: Pacing and Clinical Electrophysiology, 2002, pp. 25: 260-264. doi: 10.1046/j.1460-9592.2002.00260.x.

- [32] Paced rhythm (2012) [obiskano 20-8-2015].
URL <http://cdn.lifeinthefastlane.com/wp-content/uploads/2012/01/Ventricular-pacing-with-capture-beats.jpg>
- [33] Q. V. Le, Building high-level features using large scale unsupervised learning, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 8595--8598.
- [34] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio, Theano: new features and speed improvements, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop (2012).
- [35] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010, oral Presentation.
- [36] G. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 20 (1) (2012) 30--42. doi: 10.1109/TASL.2011.2134090.
- [37] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, Signal Processing Magazine, IEEE 29 (6) (2012) 82--97. doi:10.1109/MSP.2012.2205597.
- [38] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks

- with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, ACM, New York, NY, USA, 2008, pp. 160--167. doi: 10.1145/1390156.1390177.
URL <http://doi.acm.org/10.1145/1390156.1390177>
- [39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, *Nature* 518, doi:10.1038/nature14236.
- [40] Performance analysis of various activation functions in generalized MLP ARCHITECTURES OF NEURAL NETWORKS, author=Karlik, Bekir and Olgac, A Vehbi, journal=Int. J. Artificial Intell. Expert Syst, volume=1, number=4, pages=111--122, year=2011.
- [41] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks for Machine Learning 4.
- [42] M. D. Zeiler, ADADELTA: an adaptive learning rate method, CoRR abs/1212.5701.
URL <http://arxiv.org/abs/1212.5701>
- [43] Convolutional kernel [obiskano 20-8-2015].
URL https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/Art/kernel_convolution.jpg
- [44] W. Commons, Elman srnn (2009) [obiskano 20-8-2015].
URL https://commons.wikimedia.org/wiki/File:Elman_srnn.png

- [45] Long short term memory cell [obiskano 20-8-2015].
URL <http://christianherta.de/lehre/dataScience/machineLearning/neuralNetworks/pics/memory-cell.png>
- [46] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, *Circulation* 101 (23) (2000 (June 13)) e215--e220, *circulation Electronic Pages*: <http://circ.ahajournals.org/cgi/content/full/101/23/e215> PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
- [47] M. GB, M. RG, The impact of the MIT-BIH arrhythmia database, *IEEE Eng in Med and Biol* 20(3):45-50 (May-June 2001) PMID:11446209.
- [48] N. Morales, L. Gu, Y. Gao, Adding noise to improve noise robustness in speech recognition., in: *INTERSPEECH*, 2007, pp. 930--933.
- [49] A. K. Ka, ECG beats classification using waveform similarity and RR interval, arXiv: 1101.1836v1.
- [50] M. Hasan, M. Ibrahimy, M. Reaz, NN-Based R-peak detection in QRS complex of ECG signal, *Biomed 2008, Proceedings* 21, pp.217-220.
- [51] S.-N. Yu, K.-T. Chou, Integration of independent component analysis and neural networks for ecg beat classification, *Expert Systems with Applications* 34 (4) (2008) 2841--2846.
- [52] M. S. Aware, V. V. Shete, Neural network based heart arrhythmia detection and classification from ECG signal,

ITSI Transactions on Electrical and Electronics Engineering
(ITSI-TEEE)ISSN, 2320-8945, Volume -2, Issue -3.

Dodatek A

Statistični podatki za vse datoteke

Tabela A.1: Statistika učnih podatkov

Oseba	Brez utripa	Normalen utrip	PAC	LBBB	RBBB	PVC	Vse	Komentar
100	7453	2238	33	0	0	1	34	
101	8227	1860	3	0	0	0	3	
102	312	99	0	0	0	4	4	Spodbujevalnik
103	8061	2082	2	0	0	0	2	
104	226	163	0	0	0	2	2	Spodbujevalnik
105	7091	2526	0	0	0	41	41	
106	7757	1507	0	0	0	520	520	
107	0	0	0	0	0	59	59	Spodbujevalnik
108	8064	1739	4	0	0	17	21	
109	7478	0	0	2492	0	38	2530	
111	8054	0	0	2123	0	1	2124	
112	7584	2537	2	0	0	0	2	
113	8318	1788	0	0	0	0	0	
114	8093	1820	10	0	0	43	53	
115	8133	1953	0	0	0	0	0	
116	7247	2302	1	0	0	109	110	
117	8975	1534	1	0	0	0	1	
118	7618	0	96	0	2166	16	2278	
119	7689	1543	0	0	0	444	444	
121	8376	1861	1	0	0	1	2	
122	7396	2476	0	0	0	0	0	
123	8750	1515	0	0	0	3	3	
124	8326	0	2	0	1531	47	1580	
200	6628	1743	30	0	0	826	856	
201	6766	1625	30	0	0	198	228	
202	7585	2061	36	0	0	19	55	
203	6349	2529	0	0	0	444	444	
205	7142	2571	3	0	0	71	74	
207	6655	0	107	1457	86	105	1755	
208	5148	1586	0	0	0	992	992	
209	6189	2621	383	0	0	1	384	
210	6849	2423	0	0	0	194	194	
212	6856	923	0	0	1825	0	1825	
213	4938	2640	25	0	0	220	245	
214	7470	0	0	2003	0	256	2259	
215	6510	3195	3	0	0	164	167	
217	581	244	0	0	0	162	162	Spodbujevalnik
219	6694	2082	7	0	0	64	71	
220	7978	1953	94	0	0	0	94	
221	7251	2031	0	0	0	396	396	
222	5685	2062	208	0	0	0	208	
223	7181	2029	72	0	0	473	545	
228	7650	1688	3	0	0	362	365	
230	6906	2255	0	0	0	1	1	
231	4893	314	1	0	1254	2	1257	
232	8172	0	1382	0	397	0	1779	
233	6050	2230	7	0	0	831	838	
234	6914	2700	0	0	0	3	3	
Skupaj	320.268	75.048	2.546	8.075	7.259	7.130	25.010	