

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2022

Bc. Martin Zaňko





# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

## VLIV KÓDOVÁNÍ VÝSTUPU HLUBOKÉ NEURONOVÉ SÍTĚ NA KVALITU MODELU U MULTI-LABEL KLASIFIKACE

OPTIMIZATION OF A DEEP NEURAL NETWORK LABEL ENCODING IN A MULTI-LABEL PROBLEM.

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Martin Zaťko

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Hejč

BRNO 2022



# Diplomová práce

magisterský navazující studijní program **Bioinženýrství**

Ústav biomedicínského inženýrství

**Student:** Bc. Martin Zařko

**ID:** 203694

**Ročník:** 2

**Akademický rok:** 2021/22

**NÁZEV TÉMATU:**

## **Vliv kódování výstupu hluboké neuronové sítě na kvalitu modelu u multi-label klasifikace**

**POKYNY PRO VYPRACOVÁNÍ:**

1) Seznamte se s metodami hlubokého učení pro zpracování signálů. 2) Vytvořte literární rešerši se zaměřením na konvoluční neuronové sítě a způsoby kódování výstupu modelů pomocí metody word-embedding. 3) Prostudujte dostupná elektrokardiografická data a v prostředí Python navrhnete moduly pro kódování sady diagnostických labelů s využitím konvenčního kódování a kódování metodou word-embedding. 4) Proveďte učení vybraného modelu hluboké neuronové sítě na dostupném datasetu s využitím různých typů kódování sady labelů. 5) Proveďte vyhodnocení a diskuzi dosažených výsledků.

**DOPORUČENÁ LITERATURA:**

[1] Yu Meng et al., Unsupervised Word Embedding Learning by Incorporating Local and Global Contexts, *Frontiers in Big Data*, Volume 3, Issue 1, 2020, DOI: 10.3389/fdata.2020.00009.

[2] Kanavati, F. et al. Weakly-supervised learning for lung carcinoma classification using deep learning. *Sci Rep* 10, 9297 (2020), DOI: 10.1038/s41598-020-66333-x.

[3] Sun, Li. et al. ECG Analysis Using Multiple Instance Learning for Myocardial Infarction Detection, *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 12, pp. 3348-3356, 2012, DOI: 10.1109/TBME.2012.2213597.

**Termín zadání:** 7.2.2022

**Termín odevzdání:** 20.5.2022

**Vedoucí práce:** Ing. Jakub Hejč

**doc. Ing. Radim Kolář, Ph.D.**  
předseda rady studijního programu

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



## **ABSTRAKT**

Cieľom diplomovej práce je navrhnúť metódu hlbokého učenia na klasifikáciu arytmií zo záznamov EKG a porovnať aký má vplyv kódovanie jej výstupov na celkovú kvalitu modelu. Vybraná bola 1D konvolučná neurónová sieť a boli na nej vyskúšané a porovnané metódy kódovania labelov pomocou one-hot kódovania, ordinálneho kódovania, metódy využívajúcej autoenkóder a metódy word embedding. Dosiahnuté výsledky ukazujú, že využitím metódy word embedding sa dá zvýšiť klasifikačná schopnosť navrhutej siete.

## **KLÚČOVÉ SLOVÁ**

EKG, klasifikácia, arytmie, 1D CNN, labeling, word embedding

## **ABSTRACT**

The aim of the diploma thesis is to propose a method of deep learning for the classification of arrhythmias from ECG recordings and to compare the effect of coding its outputs on the overall quality of the model. A 1D convolutional neural network was selected and methods of label coding using one-hot coding, ordinal coding, the method using an autoencoder and the word embedding method were tested and compared on it. The obtained results show that the use of the word embedding method can increase the classification capacity of the proposed network.

## **KEYWORDS**

ECG, classification, arrhythmias, 1D CNN, labeling, word embedding





## Vyhlásenie autora o pôvodnosti diela

**Meno a priezvisko autora:** Bc. Martin Zaťko  
**VUT ID autora:** 203694  
**Typ práce:** Diplomová práca  
**Akademický rok:** 2021/2022  
**Téma záverečnej práce:** Vliv kódování výstupu hluboké neuronové sítě na kvalitu modelu u multi-label klasifikace

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podpisuje iba v tlačenej verzii.



## POĎAKOVANIE

Rád bych poděkoval vedoucímu bakalářské/diplomové/disertační práce panu Ing. Jakubovi Hejčovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.



# Obsah

Úvod	17
<b>1 Teoretická časť diplomovej práce</b>	<b>19</b>
1.1 Elektrokardiograf (EKG)	19
1.1.1 Normálny srdcový rytmus	19
1.1.2 Srdcové arytmie	20
1.2 Detekcia a klasifikácia arytmií	20
1.2.1 Prístup založený na extrakcii charakteristických vlastností signálu	22
1.2.2 Prístup založený na metódach hlbokého učenia	22
1.3 Metódy kódovania kategorického výstupu	30
1.3.1 Ordinálne kódovanie	31
1.3.2 One-hot kódovanie	31
1.3.3 Word embedding	32
1.3.4 Autoenkóдеры	34
<b>2 Praktická časť diplomovej práce</b>	<b>39</b>
2.1 EKG dáta	39
2.2 Predspracovanie	39
2.3 Architektúra 1D konvolučnej neurónovej siete	41
2.3.1 Učenie siete a validácia	43
2.4 One-hot kódovanie	44
2.4.1 Architektúra	44
2.4.2 Vyhodnotenie	45
2.5 Ordinálne kódovanie	46
2.5.1 Architektúra	46
2.5.2 Vyhodnotenie	47
2.6 Autoenkóder	47
2.6.1 Architektúra	48
2.6.2 Optimalizácia	49
2.6.3 Prevod výstupu siete na kategóriu arytmie	50
2.6.4 Vyhodnotenie	51
2.7 Word embedding	51
2.7.1 Optimalizácia	53
2.7.2 Architektúra	53
2.7.3 Prevod výstupu siete na kategóriu arytmie	53
2.7.4 Vyhodnotenie	55

2.8	Výsledky a diskusia . . . . .	56
	<b>Záver</b>	<b>59</b>
	<b>Literatúra</b>	<b>61</b>

# Zoznam obrázkov

1.1	EKG krivka. . . . .	19
1.2	Zobrazenie rôznych typov arytmií na zázname EKG. . . . .	21
1.3	Chyba algoritmu znázornená ako kritériálna funkcia C. Zelená šípka znázorňuje negatívny gradientový vektor. . . . .	24
1.4	Algoritmus spätného šírenia chyby vo viacvrstvovej neurónovej sieti. . . . .	24
1.5	Rekurentná neurónová sieť. . . . .	25
1.6	Konvolúcia filtru s rozmermi 3x3 s pixelmi vstupného obrazu. . . . .	26
1.7	Veľkosť kroku 1. Filter sa posúva iba raz. . . . .	27
1.8	Max-pooling a priemerujúci pooling. . . . .	28
1.9	Nulová výplň. (Zero-padding). . . . .	28
1.10	Typy často využívaných nelineárnych funkcií v CNN. . . . .	29
1.11	AlexNet predstavená Krizhevskym v 2014. . . . .	29
1.12	1D CNN s tromi konvolučnými a dvomi plne prepojenými vrstvami. . . . .	30
1.13	Ordinálne kódovanie. . . . .	31
1.14	One-hot kódovanie. . . . .	32
1.15	Trénovací model Word2Vec. . . . .	34
1.16	Matica spoločného výskytu. . . . .	35
1.17	Vektorové vzťahy zachytené pomocou GloVe. . . . .	35
1.18	Architektúra autoenkóderu. . . . .	36
1.19	Redukcia rozmerov pomocou autoenkóderu vs PCA. . . . .	37
2.1	Príklad hlavičky nesúcej informáciu o danom signále. . . . .	40
2.2	Ukážka záznamu obsahujúca sínusový rytmus. . . . .	41
2.3	Ukážka záznamu obsahujúca fibriláciu siení. . . . .	42
2.4	Zakódovanie názvu „atrial flutter“ pomocou naučeného autoenkóderu. . . . .	48
2.5	Graf znázorňujúci závislosť senzitivity a špecificity neurónovej siete na počte dimenzií, do ktorých bol zakódovaný jej výstup. . . . .	50
2.6	Vloženie slova „atrial“ do 40-rozmerného priestoru pomocou algo- ritmu Word2Vec. . . . .	52
2.7	Vytvorenie vektorovej reprezentácie pre slovné spojenie „sinus brady- cardia“ obsahujúcej iba pozitívne hodnoty. . . . .	53
2.8	Závislosť hodnôt senzitivity a špecificity konvolučnej neurónovej siete na počte dimenzií do ktorých boli jej výstupy vložené pomocou algo- ritmu Word2Vec. . . . .	54





## Zoznam tabuliek

2.1	Zastúpenie jednotlivých typov arytmií v stiahnutom datasete. . . . .	40
2.2	Zastúpenie jednotlivých typov arytmií a ich pomenovanie vo využitom datasete. . . . .	41
2.3	Architektúra využitej konvolučnej neurónovej siete. . . . .	43
2.4	One-hot kódovanie výstupu. . . . .	44
2.5	Architektúra využitá na učenie modelov s one-hot značením výstupov. . . . .	45
2.6	5-násobná krížová validácia modelov s one-hot značením výstupov. . . . .	46
2.7	Úspešnosť siete s one-hot značením výstupov detekovať jednotlivé druhy arytmií. . . . .	46
2.8	Ordinálne kódovanie výstupu. . . . .	47
2.9	5-násobná krížová validácia modelov s ordinálnym značením výstupov. . . . .	47
2.10	Úspešnosť siete s ordinálnym značením výstupov detekovať jednotlivé druhy arytmií. . . . .	48
2.11	Architektúra využitá na učenie modelov so značením výstupov pomocou autoenkóderu. . . . .	49
2.12	5-násobná krížová validácia modelov so značením výstupov pomocou autoenkóderu. . . . .	51
2.13	Úspešnosť siete so značením výstupov pomocou autoenkóderu detekovať jednotlivé druhy arytmií. . . . .	51
2.14	Architektúra využitá na učenie modelov so značením výstupov pomocou algoritmu Word2Vec. . . . .	55
2.15	5-násobná krížová validácia modelov so značením výstupov pomocou algoritmu Word2Vec. . . . .	56
2.16	Úspešnosť siete so značením výstupov pomocou algoritmu Word2Vec detekovať jednotlivé druhy arytmií. . . . .	56
2.17	Priemerná senzitivita a špecificita každého z modelov získaná z 5-násobnej krížovej validácie. . . . .	57
2.18	Tabuľka modelov, ktoré boli najúspešnejšie pri detekcii jednotlivých druhov arytmií. . . . .	57
2.19	Rozdiely v dĺžke učenia sa jednotlivých modelov pri využití rôznych druhov značenia labelov. . . . .	58



# Úvod

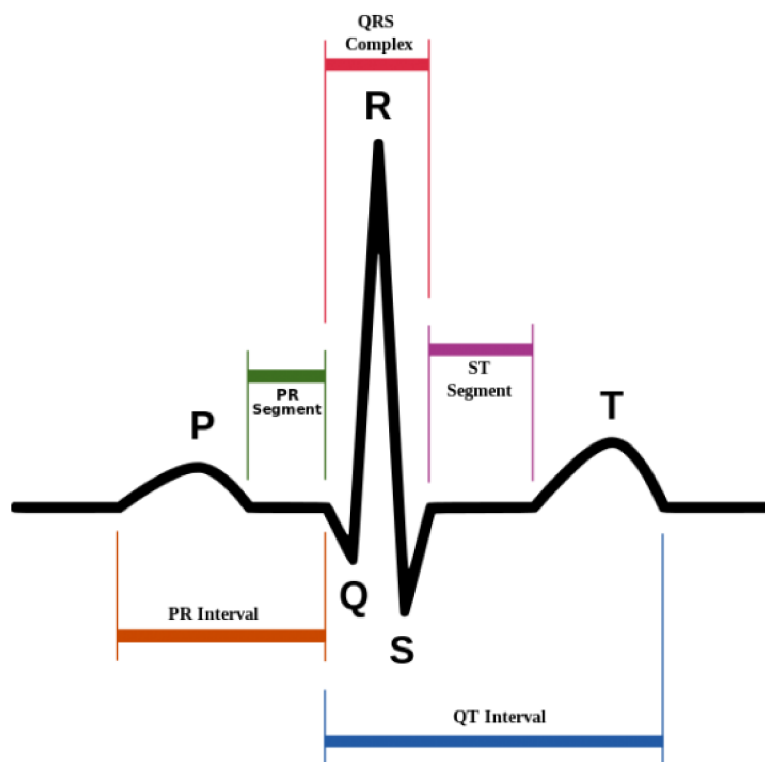
Cieľom diplomovej práce je navrhnúť metódu hlbokého učenia na klasifikáciu arytmií zo záznamov EKG a porovnať aký má vplyv kódovanie jej výstupov na celkovú kvalitu modelu. Úvodné kapitoly práce popisujú základy diagnostickej metódy EKG, spôsob tvorby a šírenia srdcových impulzov a rozdiely medzi normálnym srdcovým rytmom a srdcovými arytmiami. Ďalej sú vysvetlené metódy detekcie a klasifikácie arytmií so zameraním na metódy hlbokého učenia, presnejšie na konvolučné neuronové siete. Koniec teoretickej časti sa zameriava na metódy kódovania kategorického výstupu. Objasňuje konvenčné metódy a vysvetľuje teóriu autoenkóderov a metódy word embedding. V praktickej časti je navrhnutá architektúra 1D konvolučnej neuronovej siete a 4 metódy kódovania labelov. Konvolučná sieť je natrénovaná na dostupných EKG záznamoch a je porovnaný vplyv kódovania výstupov každého zo 4 predstavených spôsobov na efektivitu klasifikácie.



# 1 Teoretická časť diplomovej práce

## 1.1 Elektrokardiograf (EKG)

Naše nervové a svalové bunky spolu komunikujú pomocou elektrických a chemických signálov. Pravidelné elektrické signály tiež riadia náš srdcový tep. Tieto signály sú odosielané skupinou buniek v pravej srdcovej predsieni známej ako sinoatriálny uzol (SA uzol) a šíria sa cez tkanivo srdcového svalu ako drobné elektrické impulzy. To spôsobí stiahnutie predsiení a potom srdcových komôr. Spôsob, akým sa tieto signály šíria srdcom, sa dá merať aj na povrchu našej kože. EKG meria tieto zmeny elektrických signálov, presnejšie napätia, na rôznych miestach kože a zobrazuje ich ako graf. Výsledný EKG graf sa nazýva elektrokardiogram. (Obr. 1.1) [1]



Obr. 1.1: EKG krivka.[2]

### 1.1.1 Normálny srdcový rytmus

Pokojová sínusová srdcová frekvencia u dospelých je zvyčajne 60 až 100 úderov za minútu. Normálne dochádza k výrazným denným zmenám srdcovej frekvencie, s najnižšou frekvenciou tesne pred skorým ranným prebudením. Oscilácie rytmu sa

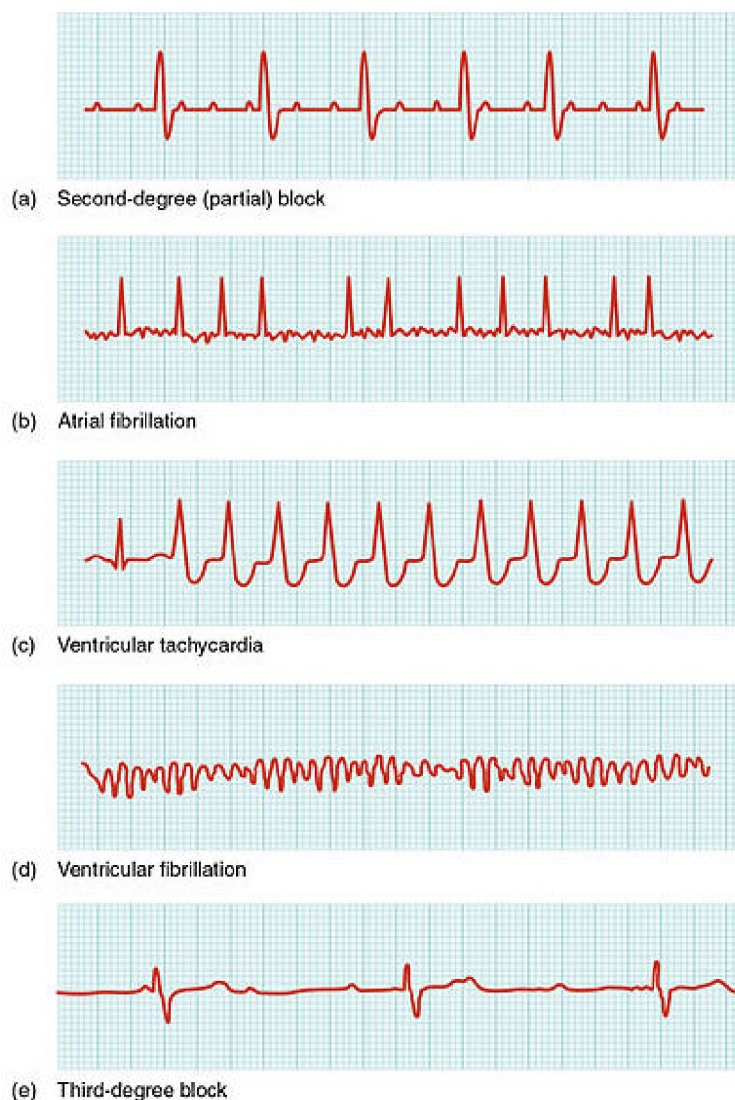
zmenšujú, ale s vekom úplne nezmiznú. Absolútna pravidelnosť frekvencie sínusového rytmu je patologická. Väčšina srdcovej elektrickej aktivity je znázornená na elektrokardiograme, hoci depolarizácia SA uzlu, AV uzlu, Hisovho zväzku a Purkyňových vlákien nezahŕňa dostatok tkaniva na to, aby bola detekovaná. Vlna P predstavuje depolarizáciu predsieni. Komplex QRS predstavuje komorovú depolarizáciu a vlna T predstavuje komorovú repolarizáciu. Interval PR (od začiatku vlny P po začiatok komplexu QRS) je čas od začiatku aktivácie predsieni do začiatku aktivácie komôr. Veľká časť tohto intervalu odráža spomalenie prenosu impulzov v AV uzle. Interval R-R (čas medzi 2 komplexmi QRS) predstavuje komorovú frekvenciu. Interval QT (od začiatku komplexu QRS do konca vlny T) predstavuje dobu trvania komorovej depolarizácie.[3]

### 1.1.2 Srdcové arytmie

Poruchy rytmu sú výsledkom abnormalít tvorby impulzov, vedenia impulzov alebo oboch. Bradyarytmie sú výsledkom zníženej funkcie pacemakeru alebo blokov vo vedení, hlavne v AV uzle alebo v His-Purkyňovom systéme. Väčšina tachyarytmií je spôsobená reentry, niektoré sú výsledkom zvýšenej normálnej automatickosti alebo abnormálnych automatických mechanizmov. Arytmie a poruchy vedenia môžu byť asymptomatické alebo môžu spôsobiť palpitácie (pocit vynechaných úderov alebo rýchlych alebo silných úderov), symptómy hemodynamického kompromisu (napr. dýchavičnosť, dyspnoe, presynkopa, synkopa) alebo zástavu srdca. Príležitostne je polyúria výsledkom uvoľnenia atriálneho natriuretického peptidu počas predĺžených supraventrikulárnych tachykardií (SVT). Palpácia pulzu a auskultácia srdca môžu určiť komorovú frekvenciu a jej pravidelnosť alebo nepravidelnosť. Vyšetrenie pulzových vln jugulárnej žily môže pomôcť pri diagnostike AV blokád a tachyarytmií. Napríklad pri kompletnej AV blokáde sa predsieni prerušovane sťahujú, keď sú AV ventily zatvorené, čím vznikajú veľké (delové) vlny v pulze jugulárnej žily. Anamnéza a fyzikálne vyšetrenie môžu odhaliť arytmie a navrhnúť možné príčiny, ale na určenie presnej diagnózy je potrebné 12-zvodové EKG alebo, menej spoľahlivý, rytmický pásik, najlepšie získaný počas symptómov, aby sa zistil vzťah medzi symptómami a rytmom. Obr.1.2 zobrazuje ukážku ako môžu vyzeráť rôzne druhy arytmii na zázname EKG.[3]

## 1.2 Detekcia a klasifikácia arytmii

Ako bolo vyššie ukázané záznam EKG je dôležitým klinickým nástrojom na detekciu srdcových porúch. Typický záznam EKG trvá od niekoľkých sekúnd (napr. počas



Obr. 1.2: Zobrazenie rôznych typov arytmií na zázname EKG.[20]

návštevy kardiológa) až po niekoľko dní pomocou Holterovho prístroja. Počet a poloha zvodových elektród sa tiež líši od jedného alebo dvoch kanálov (na nositeľnom zariadení alebo smartfóne) po väčší počet (ak je potrebné podrobnejšie zobrazenie srdcovej aktivity). Napriek tomu, že EKG je už pomerne dlho zavedenou metódou, klasifikáciu arytmiických alebo ektopických epizód vo všeobecnosti vykonávajú kardiológovia manuálne alebo poloautomaticky tak, že kontrolujú každý signál. Proces je preto nákladný, náchylný na chyby. Bolo už navrhnutých množstvo prístupov na automatizovanú klasifikáciu normálnych/abnormálnych signálov EKG. Zvyčajne používajú rôzne ručne vytvorené funkcie vrátane metrík variability srdcovej frekvencie alebo morfológických charakteristík, ako napríklad absencia P-vlny. Metódy hlbokého učenia sú čoraz populárnejšie vďaka svojej schopnosti automaticky sa učiť

funkcie na viacerých úrovniach abstrakcie (t.j. vrstvách). Tieto metódy boli úspešne aplikované v oblasti počítačového videnia, avšak v literatúre sa začali objavovať aplikácie na 1-rozmerné biomedicínske signály ako EKG.[8]

### **1.2.1 Prístup založený na extrakcii charakteristických vlastností signálu**

Pri aplikácii tohto prístupu sú obvykle signály ručne predspracované vhodnou filtráciou, ktorá umožní čo najúspešnejšiu extrakciu zvolených vlastností signálu. Medzi hlavné metódy predspracovania patrí detekcia QRS komplexov, z ktorých je možné extrahovať vlastnosti predsieňovej a ventrikulárnej aktivity pomocou HRV metrík a indexov kvality signálu. Veľmi často sa taktiež vypočítava klasická časová doména, frekvenčná doména, a nelineárne metriky HRV, ako aj nové metriky založené na zoskupovaní srdečných úderov v Poincare mapách. Okrem vlastností založených na detekcii QRS sa používajú aj morfológické vlastností, ako je prítomnosť a sila P-vlny a QT-interval. Na konci sa môžu všetky extrahované vlastnosti signálu použiť ako vstup do viacvrstvého perceptrónu (2-vrstvový, dopredný), ktorý vykoná klasifikáciu.[8]

### **1.2.2 Prístup založený na metódach hlbokého učenia**

Metódy hlbokého učenia predstavujú modely a architektúry neurónových sietí, ktoré sa skladajú z viacerých procesných vrstiev neurónov alebo blokov, ktoré sú schopné učiť sa reprezentácie údajov s viacerými úrovňami abstrakcie. Tieto metódy výrazne zlepšili súčasné možnosti v oblasti rozpoznávania reči, vizuálneho rozpoznávania objektov, detekcie objektov a mnohých ďalších oblastí spracovania a analýzy signálu. V modernej literatúre sa čoraz viac vyskytujú ich aplikácie aj na 1-rozmerné biomedicínske signály ako EKG. Hlboké učenie odhaľuje zložitú štruktúru vo veľkých súboroch údajov pomocou algoritmu spätného šírenia chyby na označenie toho, ako by mal stroj zmeniť svoje vnútorné parametre (váhy), ktoré sa používajú na výpočet reprezentácie v každej vrstve z reprezentácie v predchádzajúcej vrstve. Hlboké konvolučné siete (CNN) priniesli prelomy v spracovaní obrazov, videa, reči a zvuku, zatiaľ čo rekurentné siete (RNN) dosiahli výborné výsledky so sekvenčnými údajmi, ako je text a reč.[4] [8]

#### **Učenie viacvrstvovej architektúry**

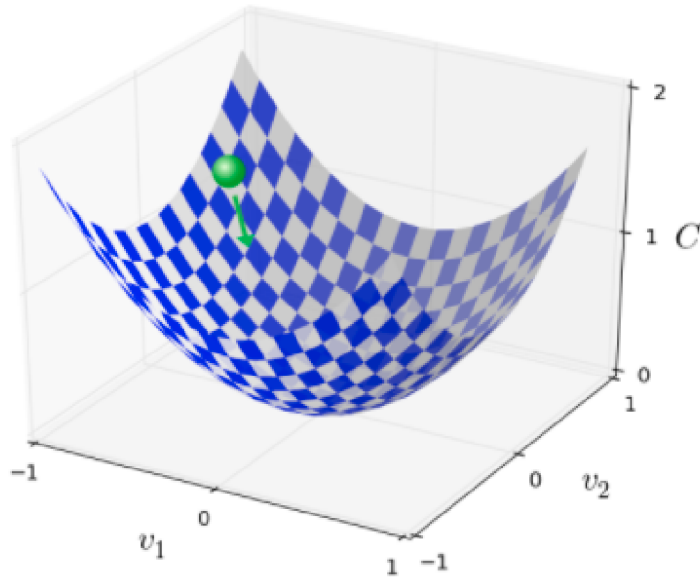
Oboje RNN a CNN, ale aj iné druhy viacvrstvových sietí sú učené spôsobom „s učiteľom“. Tento spôsob je charakterizovaný tým, že počas učenia sú siete okrem vstupov predstavené aj zodpovedajúce výstupy na základe, ktorých je sieť schopná



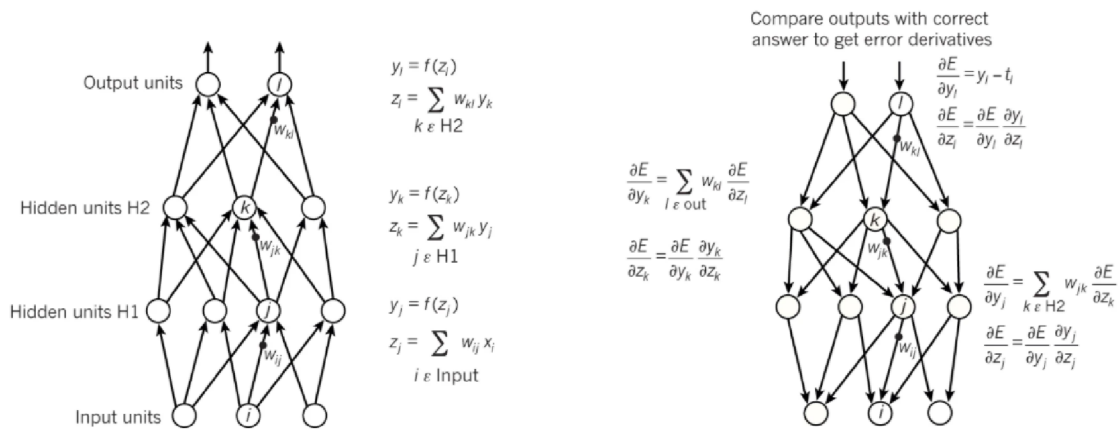
učiť sa. Na priblíženie tohto princípu je dobré si uviesť praktický príklad. Predstavme si, že chceme vytvoriť neurónovú sieť, ktorá dokáže klasifikovať obrázky, ktoré obsahujú napríklad dom, auto, osobu alebo domáce zviera. Najprv si zhromaždíme veľký súbor obrázkov domov, áut, ľudí a domácich miláčikov, pričom každý je označený svojou kategóriou. Počas tréningu sa sieti predstavia obrázok a ona vytvára výstup vo forme vektora skóre, jedno pre každú kategóriu. Chceme, aby požadovaná kategória mala najvyššie skóre zo všetkých kategórií, ale pred tréningom sa to pravdepodobne nestane. Vypočítame preto kritériálnu funkciu, ktorá meria chybu (alebo vzdialenosť) medzi výstupnými skóre a požadovaným vzorom skóre. Sieť potom upraví svoje vnútorné nastaviteľné parametre tak, aby túto chybu znížila. Tieto nastaviteľné parametre, často nazývané váhy, sú reálne čísla, ktoré definujú vstupno-výstupnú funkciu našej neurónovej siete. V typickom systéme hlbokého učenia môžu existovať stovky miliónov týchto nastaviteľných váh a stovky miliónov označených príkladov, pomocou ktorých je možné systém trénovať. (4) Aby bolo možné správne upraviť vektor váh, algoritmus učenia vypočíta gradientový vektor, ktorý pre každú váhu udáva, o koľko by sa chyba zvýšila alebo znížila, ak by sa daná váha zmenila o malé množstvo. Vektor váh sa potom upraví v opačnom smere ako vektor gradientu. Kritériálnu funkciu, spriemerovanú zo všetkých tréningových príkladov, možno považovať za akúsi kopcovitú krajinu vo vysokorozmernom priestore hodnôt váh. Obr.1.3 znázorňuje tento priestor, ako by vyzeral, keby mal 2 rozmery. Vektor záporného gradientu označuje smer najstrmšieho zostupu v tejto krajine, čím sa približuje k minimu, kde je výstupná chyba v priemere nízka. [4] Na správnu úpravu parametrov je potrebné vedieť, ako je chyba v systéme distribuovaná. Na výpočet chyby na jednotlivých neurónoch vo všetkých vrstvách sa využíva už vyššie spomenutý algoritmus spätného šírenia chyby (backpropagation). Procedúra spätného šírenia na výpočet gradientu kritériálnej funkcie vzhľadom na váhy neurónov viacvrstvovej siete nie je ničím iným ako praktickou aplikáciou reťazového pravidla pre derivácie. Kľúčovým poznatkom je, že deriváciu (deriváciou dostaneme gradient) cieľa vzhľadom na vstup siete možno vypočítať prácou naspäť od gradientu vzhľadom na výstup tejto siete (Obr.1.4). Rovnicu spätného šírenia možno použiť opakovane na šírenie gradientov cez všetky neuróny, počnúc výstupom navrchu (kde sieť vytvára svoju predpoveď) až po spodok (kde je napájaný externý vstup). Keď sú tieto gradienty vypočítané, je jednoduché vypočítať gradienty vzhľadom na váhy každého z neurónov. [4]

## **Rekurentné neurónové siete (RNN)**

Rekurentné neurónové siete sú typom NN, ktorý sa široko používa na vykonávanie procesu sekvenčnej analýzy, keďže RNN je navrhnutý na extrahovanie kontextových

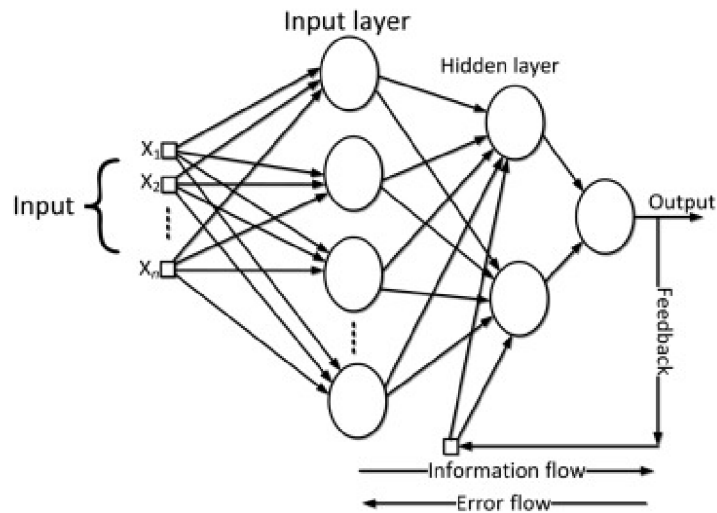


Obr. 1.3: Chyba algoritmu znázornená ako kritériálna funkcia C. Zelená šípka znázorňuje negatívny gradientový vektor. [5]



Obr. 1.4: Algoritmus spätného šírenia chyby vo viacvrstvovej neurónovej sieti. [4]

informácií definovaním závislostí medzi rôznymi časovými značkami. RNN pozostáva z mnohých po sebe nasledujúcich rekurentných vrstiev a tieto vrstvy sú postupne modelované, aby sa zmapovala sekvencia s inými sekvenciami. Obr.1.5 schématicky znázorňuje architektúru RNN s jednou rekurentnou vrstvou. RNN má silnú schopnosť zachytiť kontextové údaje zo sekvencie. Kontextové podnety v štruktúre siete sú však stabilné a efektívne sa využívajú na dosiahnutie procesu klasifikácie údajov. RNN môže prevádzkovať sekvencie s ľubovoľnou dĺžkou. [6] RNN je rozšírenie doprednej neurónovej siete s prítomnosťou slučiek v skrytých vrstvách. RNN berie



Obr. 1.5: Rekurentná neurónová sieť. [7]

vstup obsahujúci sekvenciu vzoriek a identifikuje časový vzťah medzi týmito vzorkami. [6] Keď sa spätné slučky RNN rozvinú v čase tak možno tieto siete považovať za veľmi hlboké dopredné siete, v ktorých všetky vrstvy zdieľajú rovnaké váhy.[5]

### LSTM (Long short-term memory) siete

Aby sa napravil problém s uchovávaním informácie, jedným z nápadov bolo rozšíriť sieť o explicitnú pamäť. Prvým návrhom tohto druhu sú siete LSTM, ktoré využívajú špeciálne skryté jednotky, ktorých prirodzeným správaním je zapamätanie si vstupov na dlhú dobu. Špeciálna jednotka nazývaná pamäťová bunka funguje ako akumulátor alebo hradlový presakujúci neurón, čo v praxi znamená, že má spojenie so sebou samým v ďalšom časovom kroku, ktoré má váhu jedna, takže kopíruje svoj vlastný skutočný stav a akumuluje vonkajší signál. Toto sebaapojenie je však multiplikatívne hradené inou jednotkou, ktorá sa učí rozhodovať, kedy vymazať obsah pamäte. Siete LSTM sa následne ukázali byť efektívnejšie ako konvenčné RNN, najmä keď majú niekoľko vrstiev pre každý časový krok, čo umožňuje zostrojiť celý systém rozpoznávania reči, ktorý ide od akustiky až po sekvenciu znakov v prepise. [5]

### Konvolučné neurónové siete (CNN)

Napriek tomu, že RNN dokážu pracovať so sekvenciami ľubovoľnej dĺžky, nedávne výskumy ukazujú, že tieto výhody v praxi takmer nevyužívame. Skôr sa stretávame s mnohými problémami, ktoré im neumožňujú pracovať s príliš dlhými sekvenciami, čo zrovna EKG signály so vzorkovacou frekvenciou 100 až 500 Hz sú. Na druhú

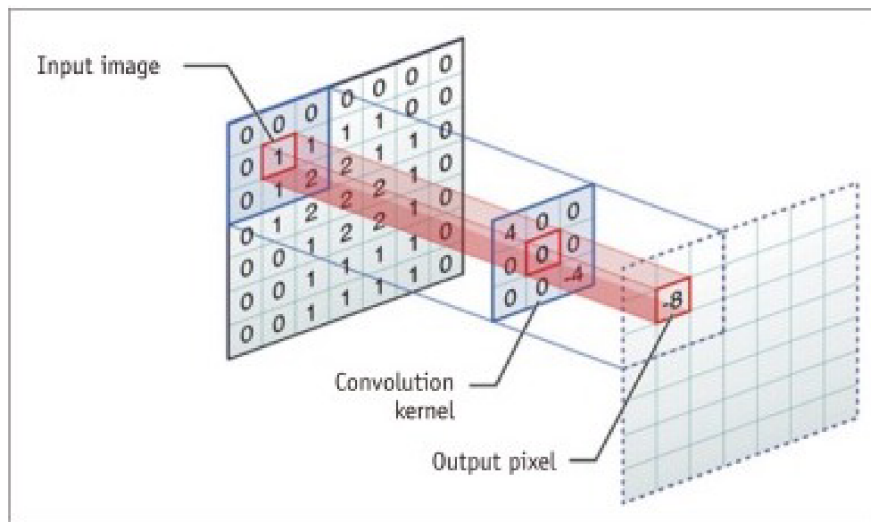
stranu CNN vykazujú skvelý výkon aj s dlhšími sekvenciami, sú rýchle, dajú sa paralelne optimalizovať a fungujú dobre pri klasifikácii a regresii, pretože ich definuje kombinácia všetkých lokálnych vzorov v časových radoch. CNN dosahujú veľmi dobré výsledky v počítačovom videní, pretože dokážu zachytiť najjemnejšie detaily (miestne vzory) v 1D a 2D signáloch alebo dokonca 3D objemových údajoch. [9] CNN v praxi využívajú viacero druhov vrstiev. Okrem konvolučných vrstiev sú to aj nelineárne, združovacie alebo plne prepojené vrstvy. Pred vysvetlením ich využitia je dobré si najskôr objasniť lineárnu matematickú operáciu konvolúcie podľa, ktorej dostali CNN svoje meno.

### Konvolučná vrstva

Každá konvolučná vrstva CNN obsahuje filtre vo forme pohybujúceho sa okna (kernelu). Konvolúciou tohto okna so vzorkami vstupného signálu dostaneme bod výstupu danej vrstvy. Ak by sa jednalo o 2D obraz, konvolúcia pre jeden pixel v ďalšej vrstve by bola vypočítaná podľa vzťahu 2.1 .

$$net(i, j) = (x * w)[i, j] = \sum_m \sum_n x[m, n]w[i - m, j - n] \quad (1.1)$$

Kde  $net$  je výstup z vrstvy,  $i$  a  $j$  sú súradnice vo výstupe,  $x$  je vstupný obraz,  $m$  a  $n$  sú súradnice vo vstupnom obraze a  $w$  je matica filtru. Obr.1.6 ukazuje ako táto operácia vyzerá v praxi.[11] V CNN sú tieto filtre inicializované a následne trénova-

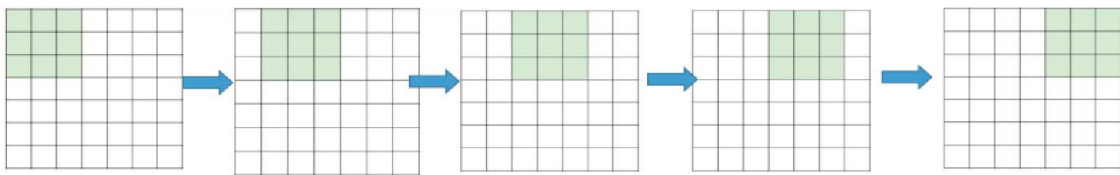


Obr. 1.6: Konvolúcia filtru s rozmermi 3x3 s pixelmi vstupného obrazu. [10]

cím postupom tvarované tak, aby boli pre danú úlohu čo najvhodnejšie. Aby bola táto metóda úspešnejšia, je možné po vstupnej vrstve pridať ďalšie vrstvy. Každá vrstva môže obsahovať rôzne filtre a z daného obrázku teda môže extrahovať rôzne

črty. Efektívnosť a rýchlosť CNN vychádza z faktu, že využitím operácie konvolúcie sa rapídne znižuje počet prepojení medzi neurónmi a taktiež počet parametrov jednotlivých vrstiev.

V skutočnosti má CNN viac možností, ktoré poskytujú ďalšie znižovanie parametrov a zároveň zníženie niektorých vedľajších účinkov. Jednou z týchto možností je veľkosť kroku (stride) s akým sa filter danej vrstvy posúva. Na Obr.1.7 vidíme okno s rozmerom  $3 \times 3$  pohybujúce sa po obrázku o veľkosti  $7 \times 7$  s posunom 1. V dôsledku veľkého prekryvu dostaneme ako výsledok takejto konvolúcie obraz s rozmerom  $5 \times 5$ . Ak však zvýšime krok na 2, výstup bude  $3 \times 3$ . Zjednodušene povedané, zníži sa nielen prekryvanie, ale aj veľkosť výstupu.[11] Ďalšou možnosťou je pooling (združovanie).

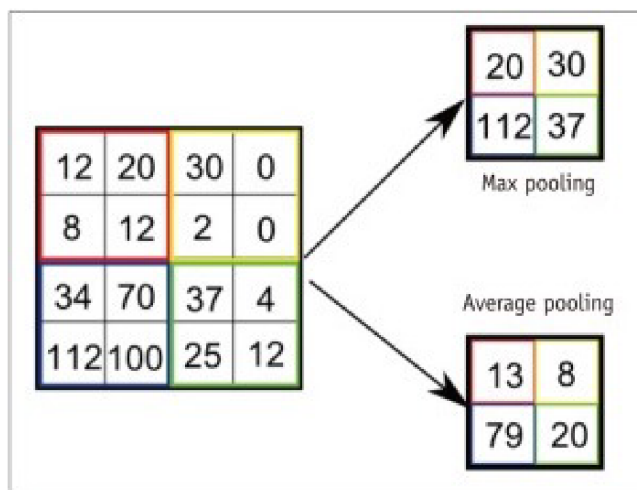


Obr. 1.7: Veľkosť kroku 1. Filter sa posúva iba raz. [11]

Hlavnou myšlienkou je odber vzoriek, aby sa znížila zložitosť pre ďalšie vrstvy. V oblasti spracovania obrazu to možno považovať za podobné znižovaniu rozlíšenia. Združovanie nemá vplyv na počet filtrov. Max-pooling je jedným z najbežnejších typov združovania. Rozdelí obrázok na podoblasti a vráti iba maximálnu hodnotu vnútra tejto podoblasti. Jedna z najbežnejších veľkostí používaných v max-poolingu je  $2 \times 2$  s krokom 2. (Obr.1.8). Aby sa zabránilo nadmernému odberu vzoriek nadol, možno použiť krok 1, čo však nie je bežné. Pri aplikácii pooling-u by sa malo vziať do úvahy, že odber vzoriek nadol nezachová pozíciu niektorých informácií. Preto by sa mal uplatňovať len vtedy, keď je dôležitá prítomnosť informácie a nie priestorové informácie. [11] Jednou z nevýhod využitia konvolúcie je strata informácií, ktoré môžu existovať na hranici obrázka. Veľmi jednoduchou, no efektívnou metódou na vyriešenie tohto problému je použitie nulovej výplne po okrajoch obrazu. Ďalšou výhodou nulovej výplne je spravovanie výstupnej veľkosti. Ak by sa na obraz na Obr.1.9 použil filter o veľkosti  $3 \times 3$  výsledok bude mať veľkosť  $5 \times 5$ . Avšak pridaním jednej nulovej výplne bude výstup  $7 \times 7$ , čo je presne to isté ako pôvodný vstup. [11]

#### Nelineárna vrstva

Ďalšou vrstvou po konvolúcii je nelineárna. Nelinearita môže byť použitá na úpravu alebo odrezanie generovaného výstupu. Táto vrstva sa používa za účelom nasýtenia výstupu alebo obmedzenia generovaného výstupu. Dlho boli sigmoid a tanh najobľúbenejšími nelineárnymi funkciami. V poslednej dobe sa však Rectified Linear Unit

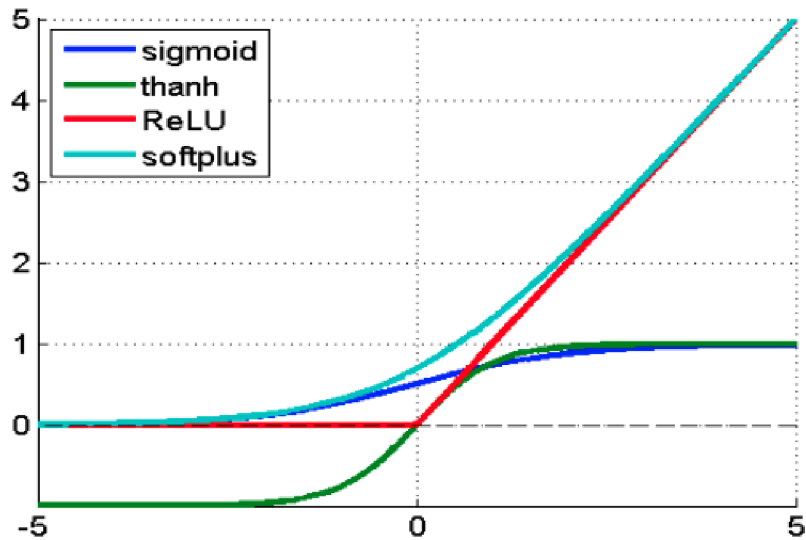


Obr. 1.8: Max-pooling a priemerujúci pooling.[10]

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Obr. 1.9: Nulová výplň. (Zero-padding). [11]

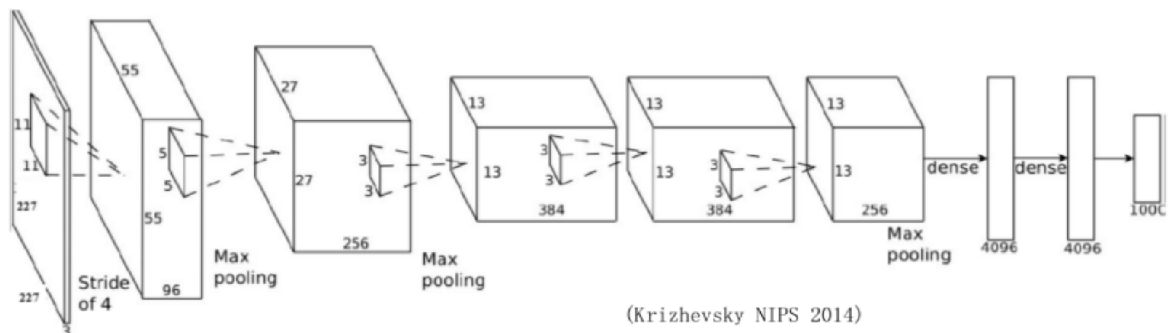
(ReLU) používa najčastejšie. Dôvodom je, že ReLU má jednoduchšie definície vo funkcii aj gradiente. Nasýtená funkcia, ako je sigmoid a tanh, spôsobuje problémy pri spätnom šírení. Keď je dizajn neurónovej siete hlbší, gradientový signál začína miznúť, čo sa nazýva „miznúci gradient“. Stáva sa to preto, že gradient týchto funkcií je takmer všade okrem stredu veľmi blízko nule. Avšak ReLU má konštantný gradient pre všetky kladné vstupy. Fakt, že funkcia nie je diferencovateľná, môže byť pri praktickej implementácii ignorovaný. ReLU vytvára redšiu reprezentáciu pretože nula v gradiente vedie k získaniu úplnej nuly. Avšak sigmoid a tanh majú vždy nenulové výsledky z gradientu, čo nemusí byť v prospech tréningu. [11] Obr.1.10 znázorňuje bežne využívané typy nelineárnych funkcií.



Obr. 1.10: Typy často využívaných nelineárnych funkcií v CNN.[11]

### Plne prepojená vrstva

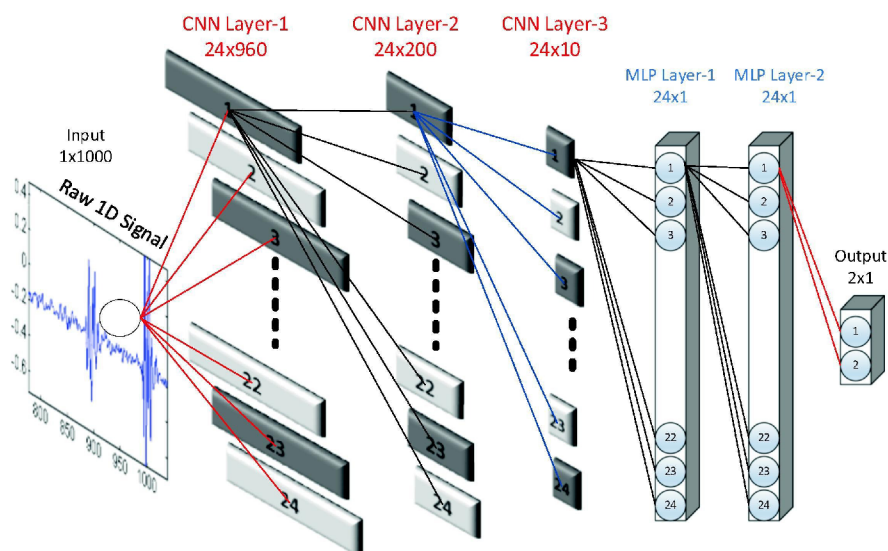
Plne prepojená vrstva funguje podobne ako klasické dopredné neurónové siete. Každý z jej neurónov je spojený s každým z neurónov predchádzajúcej a aj nasledujúcej vrstvy. Plne prepojené vrstvy sa obvykle využívajú na vykonanie koncovej klasifikácie. Hlavnou nevýhodou plne prepojenej vrstvy je, že obsahuje veľa parametrov, ktoré si vyžadujú komplexné výpočty pri procese učenia. Z toho dôvodu sa snažíme eliminovať istý počet neurónov a prepojení. Najčastejšie sa k tomu využíva technika dropout, ktorá dočasne odstraňuje neuróny a ich prepojenia zo siete. Nielenže táto metóda urýchľuje učenie, ale zároveň aj zabraňuje overfittingu. Obr.1.11 znázorňuje AlexNet ako príklad CNN so všetkými jej vrstvami. [11][12]



Obr. 1.11: AlexNet predstavená Krizhevskym v 2014. [11]

## 1D konvolučné neurónové siete

Konvenčné CNN uvedené v predchádzajúcej časti sú navrhnuté tak, aby fungovali výlučne s 2D údajmi, ako sú obrázky a videá (séria 2D obrázkov v čase). To je dôvod, prečo sa často označujú ako „2D CNN“. Ako alternatíva bola vyvinutá upravená verzia 2D CNN nazývaná 1D konvolučné neurónové siete (1D CNN) (Obr.1.12). Jednou z ich hlavných výhod je značne nižšia výpočtová zložitosť, vďaka ktorej nevyžadujú Cloudové alebo GPU služby a môžu byť kvôli tomu využité na pomerne lacné a real-time aplikácie. Štúdie preukázali vynikajúci výkon kompaktných 1D CNN v aplikáciách, ktoré majú obmedzený počet označených údajov a veľké variácie signálu získané z rôznych zdrojov (t.j. EKG, civilné, mechanické alebo letecké štruktúry, vysokovýkonné obvody atď.). Tieto dôvody a taktiež schopnosť CNN spojiť extrakciu vlastností signálu a klasifikáciu do jedného adaptívneho učiaceho sa systému majú za následok ich čoraz rozšírenejšie používanie. [13]



Obr. 1.12: 1D CNN s tromi konvolyčnými a dvomi plne prepojenými vrstvami.[13]

## 1.3 Metódy kódovania kategorického výstupu

Výkonnosť modelu strojového učenia nezávisí len od architektúry a hyperparametrov, ale aj od toho, ako spracovávame a dodávame do modelu rôzne typy premenných. Modely strojového učenia a hlbokého učenia, ako napríklad modely Keras, vyžadujú, aby všetky vstupné a výstupné premenné boli číselné. To znamená, že ak údaje obsahujú kategorické údaje, musia byť zakódované do čísel, aby bolo možné model prispôbiť a vyhodnotiť. Kategorická premenná je premenná, ktorej hodnoty patria do konečnej množiny. Premenná môže byť napríklad „farba“ a môže nadobúdať



hodnoty „červená“, „zelená“ a „modrá“. Niekedy môžu mať kategorické údaje usporiadaný vzťah medzi kategóriami, ako napríklad „prvá“, „druhá“ a „tretia“. Tento typ kategorických údajov sa označuje ako ordinálny a dodatočné informácie o ich poradí môžu byť užitočné. Dve najpopulárnejšie techniky kódovania takýchto premenných sú celočíselné kódovanie a one-hot kódovanie, hoci sa ukazuje, že novšia technika nazývaná „learned embedding“ začína byť čoraz viac využívaná vďaka jej schopnosti poskytnúť užitočný stred medzi týmito dvoma metódami.[14]

### 1.3.1 Ordinálne kódovanie

Ordinálne kódovanie zahŕňa mapovanie každej kategórie do celočíselnej hodnoty. Ako také sa niekedy označuje jednoducho ako celočíselné kódovanie. Tento typ kódovania je vhodný len vtedy, ak existuje známy vzťah medzi kategóriami. Tento vzťah môže existovať pre niektoré premenné v súbore údajov a v ideálnom prípade by sa to malo využiť pri príprave údajov. Ako príklad, Obr.1.13 ukazuje vytvorenie premennej, ktorá obsahuje kategórie reprezentujúce výšku dosiahnutého vzdelania danej osoby. [14][15]

Degree		Degree	
0	High school	0	1
1	Masters	1	4
2	Diploma	2	2
3	Bachelors	3	3
4	Bachelors	4	3
5	Masters	5	4
6	Phd	6	5
7	High school	7	1
8	High school	8	1

Obr. 1.13: Ordinálne kódovanie.[15]

### 1.3.2 One-hot kódovanie

Túto techniku kategorického kódovania údajov používame, keď sú funkcie nominálne (nemajú žiadne poradie). V one-hot kódovaní pre každú úroveň kategorického prvku vytvoríme novú premennú. Každá kategória je mapovaná binárnou premennou obsahujúcou buď 0 alebo 1. Tu 0 predstavuje neprítomnosť a 1 predstavuje

prítomnosť danej kategórie. Tieto novovytvorené binárne prvky sú známe ako fiktívne premenné. Počet fiktívnych premenných závisí na počte kategórii. Obr.1.14 ukazuje príklad, ktorý mapuje dataset s názvami miest. Keďže je miest 5, tak bolo vytvorených 5 fiktívnych premenných. [15]

	City_Delhi	City_Mumbai	City_Hydrabad	City_Chennai	City_Bangalore
<b>0</b>	1.0	0.0	0.0	0.0	0.0
<b>1</b>	0.0	1.0	0.0	0.0	0.0
<b>2</b>	0.0	0.0	1.0	0.0	0.0
<b>3</b>	0.0	0.0	0.0	1.0	0.0
<b>4</b>	0.0	0.0	0.0	0.0	1.0
<b>5</b>	1.0	0.0	0.0	0.0	0.0
<b>6</b>	0.0	0.0	1.0	0.0	0.0
<b>7</b>	0.0	0.0	0.0	0.0	1.0
<b>8</b>	1.0	0.0	0.0	0.0	0.0

Obr. 1.14: One-hot kódovanie.[15]

### 1.3.3 Word embedding

Keďže sa ukázalo, že vektorová reprezentácia je veľmi efektívna, ale stráta ordinálnej informácie, ktorá sa deje pri one-hot kódovaní je často neprehliadnutelný faktor vznikla núdza o nový prístup, ktorý by vedel tieto dve vlastnosti spojiť. S touto možnosťou prichádza word embedding. Word embeddings sú typom slovnej reprezentácie, ktorá umožňuje slovám s podobným významom mať podobnú reprezentáciu. Ide o distribuovanú reprezentáciu textu, ktorá je možno jedným z kľúčových objavov pre zlepšenie výkonu metód hlbokého učenia pri náročných problémoch spracovania prirodzeného jazyka. Jedna z výhod používania hustých a nízkorozmerných vektorov je výpočtová jednoduchosť. Väčšina súprav neurónových sietí nepracuje dobre s veľmi vysokorozmernými, riedkymi vektormi. Hlavnou výhodou hustých reprezentácií je schopnosť zovšeobecňovať. Ak sa domnievame, že niektoré vlastnosti kategórii alebo slov môžu poskytnúť podobné informácie, stojí za to poskytnúť reprezentáciu, ktorá je schopná zachytiť tieto podobnosti. Word embedding je v skutočnosti trieda techník, kde sú jednotlivé slová reprezentované ako vektory reálnych hodnôt vo vopred definovanom vektorovom priestore. Každé slovo je mapované do jedného vektora a hodnoty vektora sa učia spôsobom, ktorý pripomína neurónovú sieť, a preto je táto technika často zaradovaná do oblasti hlbokého učenia. Kľúčom k tomuto prístupu je myšlienka použitia hustej, distribuovanej reprezentácie pre každé slovo. Každé slovo

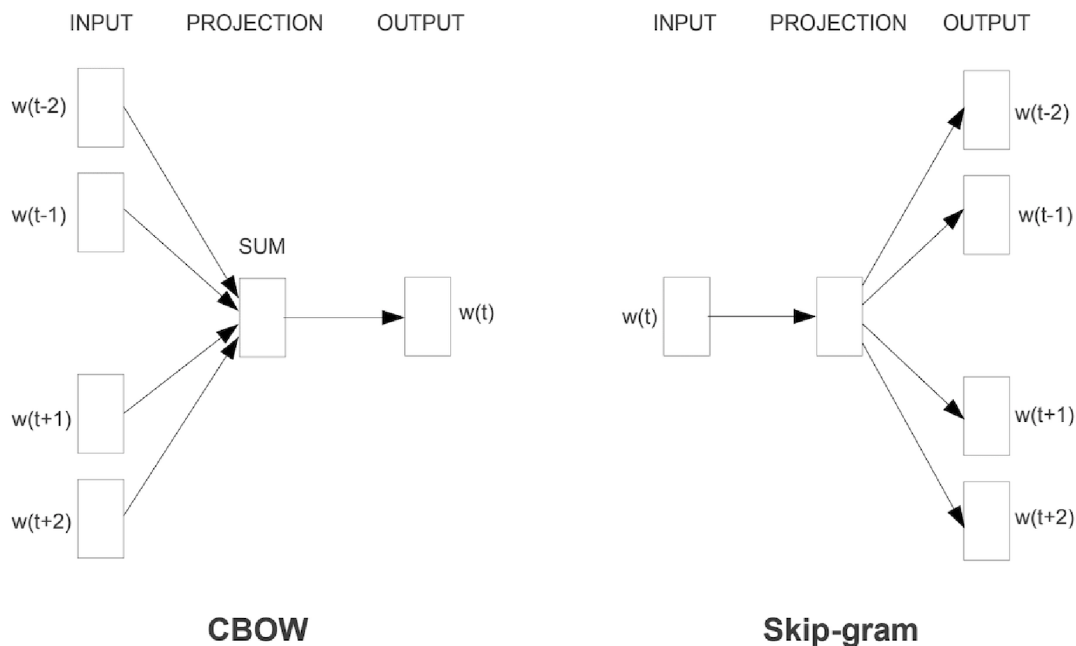
je reprezentované vektorom s často desiatkami alebo stovkami rozmerov. To je v kontraste s tisíckami alebo miliónmi rozmerov požadovaných pre jednotlivé samostatné slová, ako tomu je napríklad v one-hot kódovaní. Word embedding spája každé slovo v slovnej zásobe s distribuovaným vektorom vlastností slov. Vektor vlastností predstavuje rôzne aspekty slova. Každé slovo je spojené s bodom vo vektorovom priestore. Počet týchto vlastností je oveľa menší ako veľkosť slovnej zásoby. Distribuovaná reprezentácia sa učí na základe používania slov. To umožňuje, aby slová, ktoré sa používajú podobným spôsobom, mali podobné reprezentácie a prirodzene zachytávali ich význam. Existuje viacero metód word embeddingu, ale všetky sa naučia vektorovú reprezentáciu pre vopred definovanú slovnú zásobu pevnej veľkosti z korpusu textu. [14]

## **Word2Vec**

Word2Vec je štatistická metóda na efektívne učenie sa samostatného kódovania slov z textového korpusu. Vyvinul ju Tomas Mikolov, ako odpoveď na zefektívnenie tréningu embeddingu založeného na neurónových sieťach a odvtedy sa to stalo de facto štandardom pre vývoj vopred trénovaného word embeddingu. Bolo zistené, že tieto reprezentácie sú prekvapivo dobré pri zachytávaní syntaktických a sémantických zákonitostí v jazyku a že každý vzťah je charakterizovaný vektorovým posunom špecifickým pre daný vzťah. Tento postup umožňuje vektorovo orientované uvažovanie založené na posunoch medzi slovami. Napríklad vzťah muž/žena je automaticky naučený a s indukovanými vektorovými reprezentáciami vzťah „kráľ – muž + žena“ vedie k vektoru veľmi blízkeho „kráľovnej“. Boli predstavené dva rôzne modely učenia, ktoré možno použiť ako súčasť prístupu word2vec na učenie sa word embeddingu. (Obr.1.15) Prvý je model CBOW (Continuous Bag-of-Words), ktorý sa učí embedding predpovedaním aktuálneho slova na základe jeho kontextu. A druhý je kontinuálny Skip-Gram model, ktorý sa učí predpovedaním okolitých slov vzhľadom na aktuálne slovo. Oba modely sú zamerané na učenie sa o slovách vzhľadom na ich lokálny kontext, kde je kontext definovaný oknom susedných slov. Toto okno je konfigurovateľný parameter modelu. Veľkosť posuvného okna má silný vplyv na výsledné vektorové podobnosti. Veľké okná majú tendenciu vytvárať aktuálnejšie podobnosti, zatiaľ čo menšie okná majú tendenciu vytvárať funkčnejšiu a syntaktickejšiu podobnosť. Kľúčovým prínosom tohto prístupu je, že sa umožňuje naučiť embedding do väčších vektorov (viac rozmerov) z oveľa väčších korpusov textu (miliardy slov).[16]

## **Globálne vektory na reprezentáciu slov (GloVe)**

Klasické vektorové priestory reprezentácie slov vyvinuté pomocou techník maticovej faktorizácie, ako je latentná sémantická analýza (LSA), dobre fungujú pri použí-



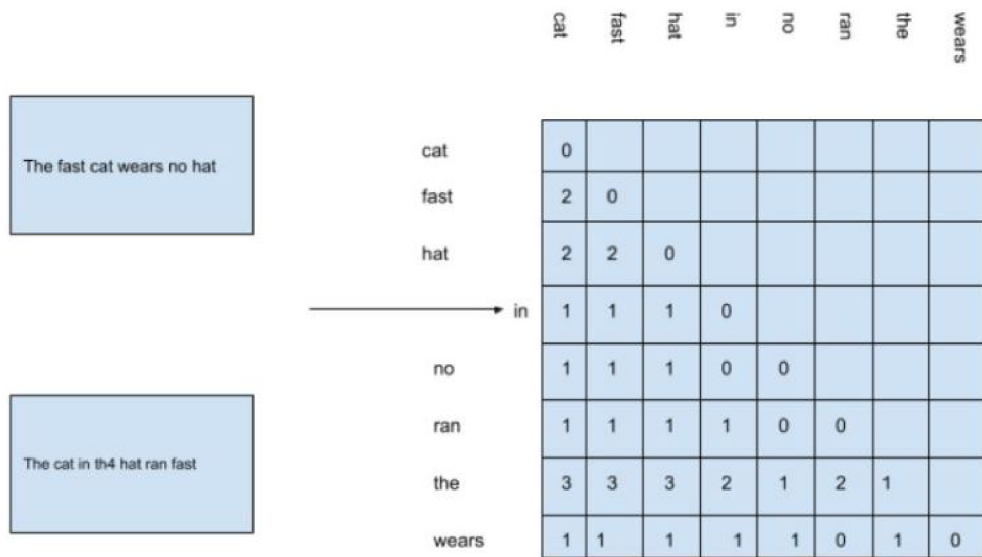
Obr. 1.15: Trénovací model Word2Vec. [16]

vaní globálnej textovej štatistiky, ale nie sú také dobré ako naučené metódy, ako word2vec, pri zachytávaní významu a jeho demonštrovaní na úlohách (napr. výpočtový príklad s „kráľom a kráľovnou“).

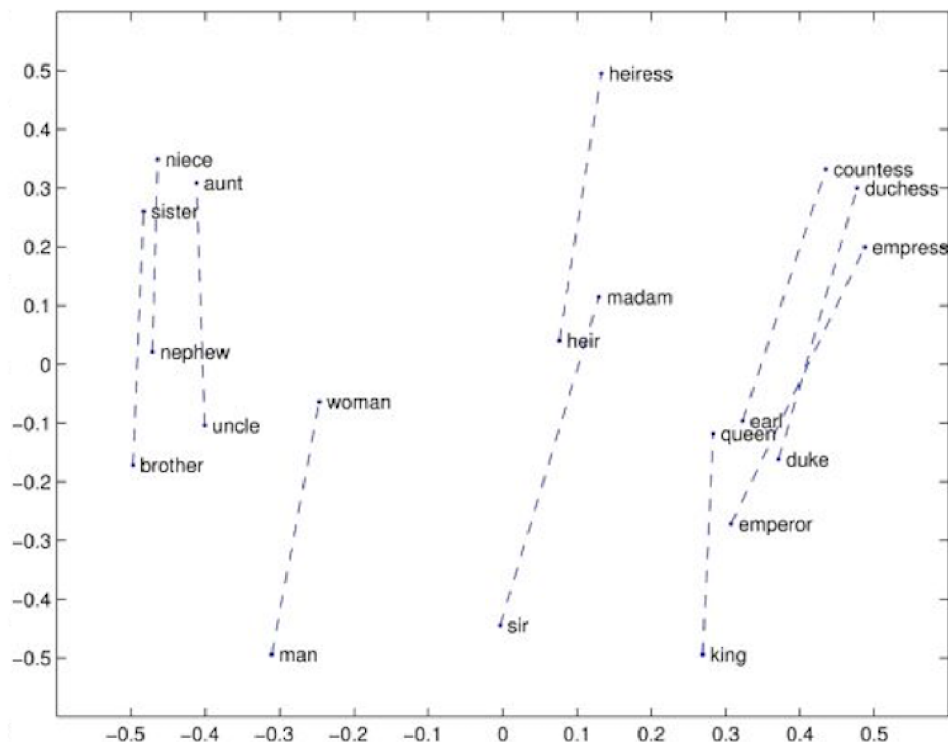
GloVe je prístup, ktorý spája globálnu štatistiku maticovej faktorizácie, ako je LSA, s lokálnym kontextovým učením vo word2vec. Namiesto použitia okna na definovanie lokálneho kontextu GloVe vytvára explicitný kontext slova alebo maticu spoločného výskytu slov (Obr.1.16) pomocou štatistík z celého tela textu. [16] Výsledkom je model učenia s vo všeobecnosti lepším word embeddingom. Sémantickú podobnosť medzi slovami pozorujeme v tomto prípade podľa vzdialenosti medzi slovami. (Obr.1.17)[17]

### 1.3.4 Autoenkódery

Poslednou technikou, ktorú si v tejto práci spomenieme na vytvorenie novej reprezentácie slov sú autoenkódery. Autoenkódery sú technikou učenia bez učiteľa, v ktorej sú využité neurónové siete na vytvorenie novej reprezentácie. Konkrétne sa navrhne architektúra neurónovej siete tak, že v sieti bude vytvorené úzke miesto, ktoré si vynúti komprimovanú reprezentáciu pôvodného vstupu. Ak by vstupné funkcie boli navzájom nezávislé, táto kompresia a následná rekonštrukcia by bola veľmi náročná úloha. Ak však v údajoch existuje nejaký druh štruktúry (t.j. korelácie me-



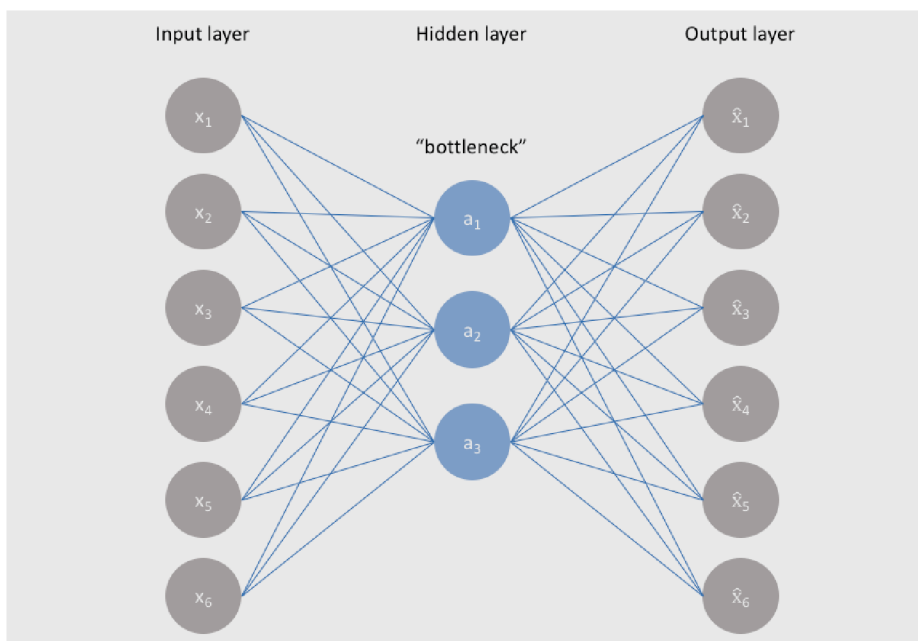
Obr. 1.16: Matica spoločného výskytu.[17]



Obr. 1.17: Vektorové vzťahy zachytené pomocou GloVe.[18]

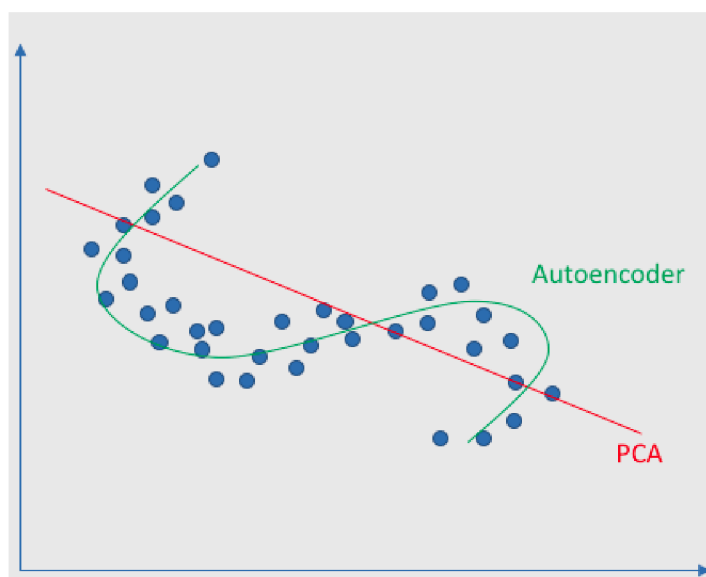
dzi vstupnými znakmi), túto štruktúru je možné naučiť a následne využiť pri pre-  
tláčaní vstupu cez úzke hrdlo siete. Obr.1.18 ukazuje, že môžeme vziať neoznačený

súbor údajov a pracovať s ním ako s problémom učenia s učiteľom, ktorého úlohou je rekonštrukcia pôvodného vstupu.[19] Túto sieť je možné trénovať minimalizovaním



Obr. 1.18: Architektúra autoenkóderu. [19]

chyby rekonštrukcie, ktorá meria rozdiely medzi naším pôvodným vstupom a následnou rekonštrukciou. Úzke miesto je kľúčovým atribútom tejto siete. Bez tohoto miesta by si sieť mohla ľahko zapamätať vstupné hodnoty tým, že tieto hodnoty iba prenesie cez sieť. Úzke miesto obmedzuje množstvo informácií, ktoré môžu prejsť celou sieťou, čo si vynúti kompresiu vstupných údajov. Najjednoduchšou architektúrou na konštrukciu autokódera je obmedziť počet neurónov prítomných v skrytej vrstve (vrstvách) siete, čím sa obmedzí množstvo informácií, ktoré môžu prechádzať sieťou. Penalizáciou siete podľa chyby rekonštrukcie sa náš model môže naučiť najdôležitejšie atribúty vstupných údajov a ako najlepšie zrekonštruovať pôvodný vstup zo „zakódovaného“ stavu. V ideálnom prípade sa toto kódovanie sieť naučí a opíše latentné atribúty vstupných údajov. Pretože neurónové siete sú schopné učiť sa nelineárne vzťahy, možno ich považovať za silnejšie zovšeobecnenie ako PCA. Zatiaľ čo PCA sa pokúša objaviť nadrovinu nižšej dimenzie, ktorá opisuje pôvodné dáta, autoenkóderi sú schopné učiť sa nelineárne rozdeľovače tohto priestoru. Rozdiel medzi týmito dvoma prístupmi je znázornený na Obr.1.19. Pre údaje vyššej dimenzie sú autokóderi schopné naučiť sa komplexnú reprezentáciu týchto údajov, ktorá sa môže použiť na opis v nižšej dimenzii a zodpovedajúcim spôsobom môže byť dekodovaná do pôvodného vstupného priestoru.[19]



Obr. 1.19: Redukcia rozmerov pomocou autoenkóderu vs PCA.[19]





## 2 Praktická časť diplomovej práce

V praktickej časti tejto práce budú najskôr predstavené dostupné EKG dáta s ich následným predspracovaním. Na riešenie tejto práce boli navrhnuté štyri spôsoby kódovania označení týchto dát. Jedná sa o dva konvenčné spôsoby, a to one-hot kódovanie a ordinálne kódovanie a dva nekonvenčné spôsoby v podobe autoenkóderu a algoritmu Word2Vec. Na porovnanie efektivity týchto štyroch druhov kódovania výstupu bola navrhnutá 1D konvolučná neurónová sieť vykonávajúca klasifikáciu dostupných EKG dát.

### 2.1 EKG dáta

EKG dáta použité v tejto práci sú časť dát poskytnutých pre PhysioNet/Computing in Cardiology Challenge 2021. Dáta v celej súťaži zahŕňajú anotované dvanásťzvodové záznamy EKG zo šiestich zdrojov v štyroch krajinách na troch kontinentoch. Tieto databázy obsahujú viac ako 100 000 dvanásťzvodových záznamov EKG. Keďže v tejto práci nejde o zostrojenie čo najvýkonnejšieho klasifikátora, ale o zistenie vplyvu kódovania výstupu na výkon neurónovej siete, do tréningovej množiny nebudú zahrnuté všetky dáta poskytnuté pre túto challenge. Namiesto toho bol vybraný len jeden zo šiestich dostupných zdrojov obsahujúci vyše 14 000 záznamov. Jedná sa o databázu zo Shaoxing People's Hospital, poskytnutú Chapman Univerzitou. Každý z jej záznamov má dĺžku desať sekúnd so vzorkovacou frekvenciou 500 Hz. Všetky údaje boli naformátované vo formáte WFDB. (Obr. 2.1) Každý záznam EKG používa binárny súbor MATLAB (.mat) pre údaje signálu EKG a súbor vo formáte obyčajného textu v tvare hlavičky WFDB na popis atribútov pacienta vrátane diagnózy a vzorkovacej frekvencie daného záznamu. Každý záznam EKG má jedno alebo viac označení pomocou kódu SNOMED-CT, ktorý popisuje srdcové abnormality (a/alebo normálny sínusový rytmus).

### 2.2 Predspracovanie

Napriek tomu, že sa v databáze nachádza vyše 14 000 záznamov, nie všetky boli v tejto práci použité. Značná časť bola označená dvomi alebo viacerými označeniami, čo znemožňovalo využitie klasických metód kódovania výstupu. Z tohto dôvodu bolo pre riešenie tejto práce využitých iba 5 druhov arytmií, ktoré boli z datasetu vybrané na základe ich množstva v datasete a ich morfológii. Z Tab.2.1 je vidieť, že dáta obsahujú príliš málo záznamov sieňového flutteru. Na vyriešenie tohto problému bol každý z týchto záznamov augmentovaný jednoduchým obrátením v časovej ose a následne bol vložený do použitej množiny, čo dvojnásobne zvýšilo zastúpenie

```

A0001 12 500 7500 05-Feb-2020 11:39:16
A0001.mat 16+24 1000/mV 16 0 28 -1716 0 I
A0001.mat 16+24 1000/mV 16 0 7 2029 0 II
A0001.mat 16+24 1000/mV 16 0 -21 3745 0 III
A0001.mat 16+24 1000/mV 16 0 -17 3680 0 aVR
A0001.mat 16+24 1000/mV 16 0 24 -2664 0 aVL
A0001.mat 16+24 1000/mV 16 0 -7 -1499 0 aVF
A0001.mat 16+24 1000/mV 16 0 -290 390 0 V1
A0001.mat 16+24 1000/mV 16 0 -204 157 0 V2
A0001.mat 16+24 1000/mV 16 0 -96 -2555 0 V3
A0001.mat 16+24 1000/mV 16 0 -112 49 0 V4
A0001.mat 16+24 1000/mV 16 0 -596 -321 0 V5
A0001.mat 16+24 1000/mV 16 0 -16 -3112 0 V6
#Age: 74
#Sex: Male
#Dx: 426783006
#Rx: Unknown
#Hx: Unknown
#Sx: Unknown

```

Obr. 2.1: Príklad hlavičky nesúcej informáciu o danom signále.

Tab. 2.1: Zastúpenie jednotlivých typov arytmíí v stiahnutom datasete.

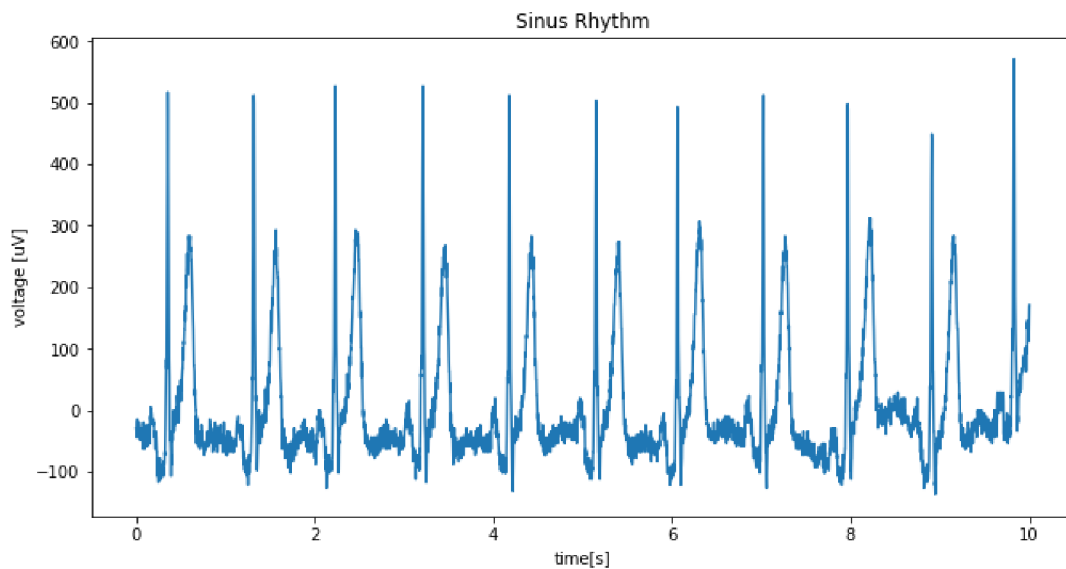
SNOMED-CT kód	Popis podľa stránky BioPortal	Množstvo
164889003	ECG: atrial fibrillation	1780
164890007	EKG: atrial flutter	445
426783006	ECG: sinus rhythm	1826
426177001	ECG: sinus bradycardia	3889
427084000	ECG: sinus tachycardia	1568
spolu		9508

tohto druhu arytmie. Pre redukciu neprimerane vysokého počtu záznamov obsahujúcich sínusovú bradykardiu, bol využitý iba každý druhý záznam. V neskoršej časti tejto práce budú využité aj algoritmy, ktoré vyžadujú slovné označenia signálov. Pre zefektívnenie výpočtov bolo z pomenovaní stránkou Bioportál odobrané prvé slovo („ECG:“) a boli použité iba holé názvy arytmíí bez veľkých písmen. Koncový počet a formu označenia záznamov možno vidieť v Tab.2.2 Keďže sa jedná o EKG signál, tak bolo potrebné vziať do úvahy možnú filtráciu, ale po vizuálnej prehliadke náhodných záznamov (ako napríklad na Obr.2.2 a Obr.2.3) bolo zistené, že signály sú už pomerne čisté a ďalšia filtrácia by bola zbytočná.

Pri práci s neurónovými sieťami je vždy dobré z výpočtových dôvodov dáta normalizovať. Táto forma predspracovania mala za následok, že použité dáta sa pohybujú v

Tab. 2.2: Zastúpenie jednotlivých typov arytmí a ich pomenovanie vo využitom datasete.

Slovné označenie	Množstvo
atrial fibrillation	1780
atrial flutter	890
sinus rhythm	1826
sinus bradycardia	1944
sinus tachycardia	1568
spolu	8058

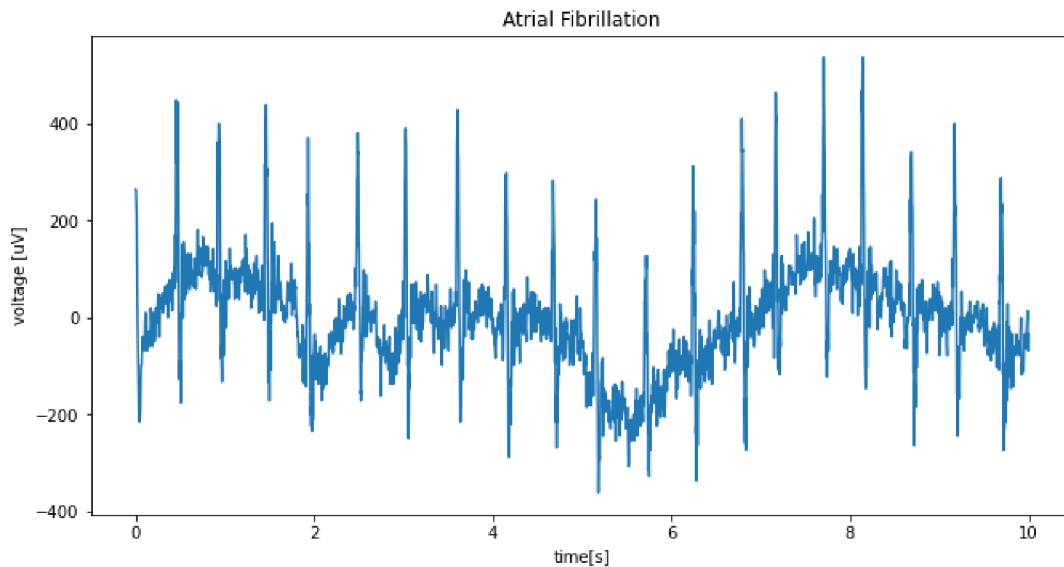


Obr. 2.2: Ukážka záznamu obsahujúca sínusový rytmus.

rozmedzí od 0 do 1. Následne už len bolo potrebné dáta nahráť do matice o veľkosti 8052(počet záznamov) x 5000(dĺžka záznamov = 10s x 500Hz) x 12(počet zvodov), ktorá môže byť použitá ako vstup do neurónovej siete.

## 2.3 Architektúra 1D konvolučnej neurónovej siete

Tak ako bolo spomenuté v úvode praktickej časti, na riešenie problému klasifikácie bola zvolená 1D konvolučná neurónová sieť, ktorá prechádza signál zo všetkých 12 zvodov naraz, ale pohybuje sa iba po jednej (časovej) dimenzii. Architektúra bola zvolená tak, aby vykonávala dostatočne úspešnú klasifikáciu na to, aby sa dal porovnať vplyv kódovania jej výstupu. Architektúra konvolučnej siete, tak isto ako aj celé prevedenie praktickej časti bolo realizované v programovacom jazyku Python.



Obr. 2.3: Ukážka záznamu obsahujúca fibriláciu siení.

Zostrojenie a učenie neurónových sietí bolo prevedené pomocou knižnice TensorFlow. Prvotný návrh architektúry bol inšpirovaný architektúrou z článku „Detection of Atrial Fibrillation Using 1D Convolutional Neural Network“ [23] a neskôr bol optimalizovaný na dataset tejto práce. Prvý spôsob čím sa zvýšil výkon bolo pridanie vrstiev „BatchNormalization“, ktoré urobili sieť rýchlejšou a stabilnejšou. Na zvýšenie generalizačnej schopnosti boli pridané ďalšie vrstvy „Dropout“. Poslednou úpravou oproti originálnemu dizajnu bolo odobranie posledných dvoch konvolučných vrstiev na zníženie komplexnosti siete.

Vstup do siete o veľkosti 5000 x 12 prechádza najskôr dvoma 1D konvolučnými vrstvami s počtom filtrov 32, kernelom veľkosti 5 a aktivačnou funkciou ReLU. Ďalej nasledujú ďalšie tri páry konvolučných vrstiev s rovnakou veľkosťou kernelu a aktivačnou funkciou. Jediný rozdiel je, že po každých dvoch vrstvách sa počet filtrov zdvojnásobí, takže posledné dve konvolučné vrstvy obsahujú 256 filtrov. Na zníženie počtu parametrov siete je po každej konvolučnej vrstve „MaxPool“ vrstva s oknom veľkosti 2. Na úplnom konci siete sa nachádzajú dve plne prepojené vrstvy, ktoré vykonávajú klasifikáciu. Prvá z nich má aktivačnú funkciu ReLU a druhá sigmoidálnu. Dohromady boli navrhnuté 4 architektúry. Jedna pre každý zo 4 druhov kódovania výstupu. Veľkosť výstupu, tak isto ako aj kritériálna funkcia sa zakaždým menili podľa spôsobu značenia výstupov (viz ďalej). Tab.2.3 znázorňuje podrobnú štruktúru využitej architektúry.

Tab. 2.3: Architektúra využitej konvolučnej neurónovej siete.

Vrstva	Parametre	Aktivačná funkcia
Conv1D	32 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	32 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	128 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	128 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Conv1D	265 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	256 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Flatten		
Dense	128	ReLU
Dense	(záleží na kódovaní výstupu)	Sigmoid

### 2.3.1 Učenie siete a validácia

Na vyhodnotenie úspešnosti klasifikácie bola použitá 5-násobná krížová validácia. To znamená, že celý dataset bol rozdelený na 5 rovnako veľkých častí, pričom každá obsahuje 1600 záznamov. Dohromady bolo natrénovaných 5 modelov pre každý z typov značenia s tým, že každý model sa učil na iných 4 skupinách záznamov a jedna vždy ostala ako testovacia množina. Množina určená na tréning siete bola ďalej rozdelená na tréningovú a validačnú v pomere 9 ku 1. „Batch size“ bol pre všetky

modely nastavený na 32 a počet epôch učenia na 50, čo však žiaden z modelov nedosiahol. Učenie bolo nastavené tak, že ak sa validačná chyba nezlepšila za posledných 10 epôch, tak sa učenie samo zastavilo a uložili sa váhy z najúspešnejšej epochy. Vo všetkých architektúrach bol pri učení využitý optimalizér „Adam“, ktorý je rozšírením metódy gradientového zostupu o adaptáciu kroku učenia počas tréningu, vďaka čomu nebolo potrebné robiť opakované skúšky učenia na nájdenie najideálnejšieho kroku učenia. Ako validačné metriky boli použité senzitivita a špecificita. Senzitivita odzrkadľuje schopnosť testu zachytiť prítomnosť sledovaného stavu. (2.1) V tomto prípade schopnosť neurónovej siete úspešne predpovedať správny výsledok a preto sa bude v priebehu práce na ňu klásť najväčší dôraz.

$$\text{Senzitivita} = TP / (TP + FN) \quad (2.1)$$

Špecificita vyjadruje schopnosť testu správne označiť prípady u ktorých skúmaný znak nenastáva. Pri klasifikácii neurónovou sieťou to znamená schopnosť siete správne určiť o ktorú arytmiu sa nejedná.

$$\text{Senzitivita} = TN / (TN + FP) \quad (2.2)$$

## 2.4 One-hot kódovanie

Ako prvý prístup pre kódovanie výstupu bolo zvolené one-hot kódovanie, ktoré sa v oblasti neurónových sietí považuje za zlatý štandard. Pri tomto druhu kódovania bolo každej z arytmií priradené označenie v podobe vektora s dĺžkou rovnou počtu možných kategórii. Táto vektorová reprezentácia je výpočtovo efektívna, ale nevystihuje vzťahy medzi jednotlivými kategóriami. Tab.2.5 zobrazuje označenie priradené jednotlivým druhom arytmií pri učení 1D CNN.

Tab. 2.4: One-hot kódovanie výstupu.

Slovné označenie	One-hot
atrial fibrillation	[1,0,0,0,0]
atrial flutter	[0,1,0,0,0]
sinus rhythm	[0,0,1,0,0]
sinus bradycardia	[0,0,0,1,0]
sinus tachycardia	[0,0,0,0,1]

### 2.4.1 Architektúra

Pre správny chod neurónovej siete musel byť jej výstup nastavený na veľkosť 5, čo odpovedá dĺžke one-hot značenia. (Tab.2.5) Ako kritériálna funkcia bola zvolená

„binary crossentropy“ keďže sa jedná o binárne značenie.

Tab. 2.5: Architektúra využitá na učenie modelov s one-hot značením výstupov.

Vrstva	Parametre	Aktivačná funkcia
Conv1D	32 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	32 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	128 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	128 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Conv1D	265 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	256 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Flatten		
Dense	128	ReLU
Dense	<b>5</b>	Sigmoid

## 2.4.2 Vyhodnotenie

Priemerná senzitivita pri one-hot kódovaní výstupov dosiahla 92% a špecificita 98% . Pritom najúspešnejší z piatich natrénovaných modelov dosiahol hodnotu senzitivity 94,7% a špecificity 98,7%. Najúspešnejší model (model 1) bol ďalej skúmaný aby bola zistená jeho citlivosť na jednotlivé druhy arytmií. Tab.2.7 ukazuje, že tento

Tab. 2.6: 5-násobná krížová validácia modelov s one-hot značením výstupov.

Model	Senzitivita	Špecificita
1	0,897	0,974
2	0,947	0,987
3	0,92	0,98
4	0,909	0,977
5	0,929	0,982
priemer	0,92	0,98
$\sigma$	0,019	0,005

model bol schopný detekovať sínusové bradykardie až so 100% úspešnosťou. Tab.2.7 taktiež ukazuje, že nižšie zastúpenie záznamov so sieňovými flutterom v tréningovej množine sa ukázalo aj na nižšej úspešnosti siete detekovať túto arytmiu.

Tab. 2.7: Úspešnosť siete s one-hot značením výstupov detekovať jednotlivé druhy arytmií.

Trieda	Senzitivita	Špecificita
atrial fibrillation	0,96	0,977
atrial flutter	0,851	0,981
sinus rhythm	0,97	0,999
sinus bradycardia	1	0,993
sinus tachycardia	0,946	0,998

## 2.5 Ordinálne kódovanie

Pri využití ordinálneho kódovania výstupov bolo prihliadnuté na druh signálu. Najnižšie označenie dostal zdravý sínusový rytmus. Po ňom nasledovali stále nepatologické arytmie vo forme spomaleného alebo zrýchleného rytmu. A najvyššie označenie dostali patologické arytmie. (Tab.2.8) Týmto spôsobom značenia bol zavedený aspoň malý vzájomný vzťah medzi arytmiami.

### 2.5.1 Architektúra

Všetky vrstvy architektúry siete pre učenie sa záznamov s ordinálnym značením boli zhodné s sieťou, ktorá využila one-hot značenie.(Tab.2.3) To znamená, že výstup mal aj napriek jednomiestnemu značeniu veľkosť 5. Knižnica TensorFlow funguje tak, že



Tab. 2.8: Ordinálne kódovanie výstupu.

Slovné označenie	Ordinálne kódovanie
sinus rhythm	0
sinus bradycardia	1
sinus tachycardia	2
atrial flutter	3
atrial fibrillation	4

v nej vytvorená architektúra po prebehnutí učenia bude aj napriek ordinálnemu značeniu produkovať predpovede vo forme vektoru (dĺžky rovnajúcej sa počte kategórii), ktorý obsahuje pravdepodobnosti s akými daná predpoveď zapadá do jednotlivých kategórii. Zmenená preto bola iba kritériálna funkcia z „binary crossentropy“ na „sparse categorical crossentropy“.

## 2.5.2 Vyhodnotenie

Využitie ordinálneho kódovanie výstupov neprineslo vo výsledkoch veľkú zmenu oproti one-hot kódovaniu. Priemerná senzitivita dosiahla 91,9% a špecificita 98% . (Tab.2.9) Najúspešnejší model bol v tomto prípade model 5, ktorý dosiahol hod-

Tab. 2.9: 5-násobná krížová validácia modelov s ordinálnym značením výstupov.

Model	Senzitivita	Špecificita
1	0,91	0,978
2	0,937	0,984
3	0,903	0,976
4	0,895	0,974
5	0,948	0,987
priemer	0,919	0,98
$\sigma$	0,023	0,005

notu senzitivity 94,8% a špecificity 98,7%. Tab.2.10 ukazuje, že aj pri jeho učení sa odzrkadlila nerovnomernosť v zastúpení dát v trénovacej množine.

## 2.6 Autoenkóder

Využitá architektúra autoenkóderu bola inšpirovaná architektúrou Yrona Vazana [5], ktorý využil LSTM vrstvy. Upravenému modelu pre túto prácu sú podané celé názvy v podobe dvojslovných strings, ktoré sú najskôr pomocou funkcie Tokenizer z

Tab. 2.10: Úspešnosť siete s ordinálnym značením výstupov detekovať jednotlivé druhy arytmií.

Trieda	Senzitivita	Špecificita
atrial fibrillation	0,904	0,979
atrial flutter	0,863	0,972
sinus rhythm	0,985	0,991
sinus bradycardia	0,979	0,997
sinus tachycardia	0,956	0,998

knižnice Tensorflow prevedené na sekvencie celých čísel o veľkosti 7, čo je počet slov použitých v slovnom pomenovaní arytmií. Tento vektor prejde bidirectional LSTM vrstvou, ktorá má výstup o veľkosti B. Táto premenná B, ďalej určuje šírku takzvaného „bottlenecku“ tohto autoenkóderu (Obr.1.18). Časť dekodéru znovu obsahuje LSTM vrstvu, ktorá nakonci predá svoj výstup do plne prepojenej vrstvy, ktorá podá výstup znovu o veľkosti 7. (Viz kód v súbore „Architectures\_AutoEncoding.ipynb“ v prílohe) Tento výstup je však použitý iba na učenie autoenkóderu. Vektor, ktorý je použitý ako nová reprezentácia označení kategórii arytmií je B miestny vektor, ktorý dostaneme vďaka prvej časti architektúry, teda enkóderu. Tento enkóder kóduje každé dvojslovné označenie arytmie do 18-miestneho vektora čísel v rozmedzí od 0 do 1. (Dôvod prečo 18 viz v podkapitole „Optimalizácia“.) Príklad takéhoto kódovania je možný vidieť na Obr.2.4. V prípade autoenkóderu použitého v tejto práci sa teda nejedná o redukciu dát pomocou enkóderu, ako tomu obvykle býva, ale o vytvorenie reprezentácie vo viacrozmernom priestore. Autoenkóder bol učný po dobu

```
['atrial flutter']
[0.          0.00504137 0.          0.00856645 0.00010146 0.00050301
 0.          0.0002954  0.0035008  0.00039415 0.00368489 0.
 0.00780227 0.0005284  0.0003341  0.          0.00624551 0.0031944 ]
```

Obr. 2.4: Zakódovanie názvu „atrial flutter“ pomocou naučeného autoenkóderu.

desiatich epóch a bola použitá kriteriálna funkcia „sparse categorical crossentropy“, keďže slovné spojenia boli tokenizované do vektorov celých čísel.

## 2.6.1 Architektúra

V architektúre pre tento typ kódovania výstupu bola pozmenená posledná vrstva tak, aby mala výstup o veľkosti 18, čo odpovedá veľkosti výstupu enkóderu.(Tab.2.11)

Ako kritériálna funkcia bola použitá „binary crossentropy“, keďže hodnoty v 18-dimenzionálnom označení sa pohybujú len od 0 do 1.

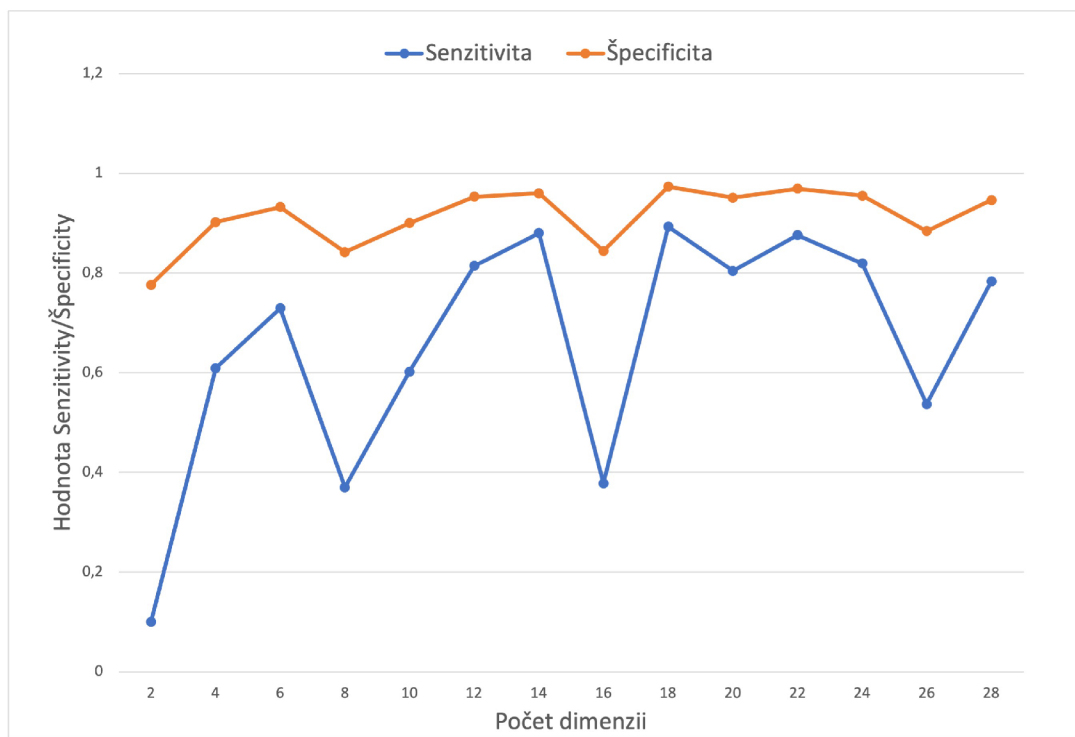
Tab. 2.11: Architektúra využitá na učenie modelov so značením výstupov pomocou autoenkóderu.

Vrstva	Parametre	Aktivačná funkcia
Conv1D	32 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	32 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	128 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	128 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Conv1D	265 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	256 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Flatten		
Dense	128	ReLU
Dense	<b>18</b>	Sigmoid

## 2.6.2 Optimalizácia

Pri použití autoenkóderu sa vyskytla otázka, do koľkých dimenzií je potrebné zakódovať kategórie, tak aby sieť podávala čo najlepšie výsledky. Z tohto dôvodu bolo

vyskúšaných 16 verzií, pre ktoré bola vždy vypočítaná senzitivita a špecificita. Graf na Obr.2.5 ukazuje, že najvyššie hodnoty senzitivity a špecificity sa stretli pri 18-miestnom kódovaní a preto bola táto hodnota zvolená aj pre ďalšie výpočty.



Obr. 2.5: Graf znázorňujúci závislosť senzitivity a špecificity neurónovej siete na počte dimenzii, do ktorých bol zakódovaný jej výstup.

### 2.6.3 Prevod výstupu siete na kategóriu arytmie

Prepočet výstupu tejto neurónovej siete na finálnu kategóriu arytmie nie je tak jednoduchý, ako napríklad pri one-hot kódovaní kedy je výsledok rovný indexu s najvyššou hodnotou pravdepodobnosti. Keďže výstupom siete je vektor s hodnotami od 0 po 1 bez bližšie známych návazností bolo potrebné prísť na nový spôsob prevodu. Vieme, že označenia sú zakódované do 18-rozmerného priestoru a že výstup siete je bod posadený niekde do tohto priestoru. Za výsledok teda môžeme brať označenie, ktoré je v tomto priestore najbližšie k predpovedi našej siete. Tento výsledok bol dosiahnutý výpočtom Euklidovskej vzdialenosti medzi predikciou siete a označeniami arytmií.

## 2.6.4 Vyhodnotenie

S kódovaním výstupu do 18-rozmerného priestoru pomocou autoenkóderu bola sieť schopná dosiahnuť priemernú senzitivitu 90,8% a špecificitu 97,7%. Tab.2.13 uka-

Tab. 2.12: 5-násobná krížová validácia modelov so značením výstupov pomocou autoenkóderu.

Model	Senzitivita	Špecificita
1	0,874	0,968
2	0,945	0,986
3	0,927	0,982
4	0,893	0,973
5	0,899	0,975
priemer	0,908	0,977
$\sigma$	0,028	0,007

zuje, že najúspešnejší model (model2) dosiahol priaznivé výsledky pri klasifikácii väčšiny arytmií, ale taktiež ukazuje, že tento daný spôsob značenia výstupov je najviac náchylný na nerovnomernosti v zastúpení jednotlivých kategórií v tréningovej množine.

## 2.7 Word embedding

Na vytvorenie reprezentácie pomocou word embeddingu bol využitý algoritmus Word2Vec z knižnice Gensim. Tento algoritmus vie využívať už dopredu naučenú slovnú zásobu, avšak tieto slovné zásoby nemusia vždy obsahovať medicínske termíny a navyše kódujú slová do zbytočne príliš vysokorozmerného priestoru. Pre potreby tejto diplomovej práce stačilo slovnú zásobu simplifikovať na počet slov použitých v

Tab. 2.13: Úspešnosť siete so značením výstupov pomocou autoenkóderu detekovať jednotlivé druhy arytmií.

Trieda	Senzitivita	Špecificita
atrial fibrillation	0,948	0,971
atrial flutter	0,694	0,989
sinus rhythm	0,971	0,99
sinus bradycardia	0,982	0,982
sinus tachycardia	0,947	0,996

slovných pomenovaniach arytmií. Bol otestovaný aj prístup pri ktorom bola sieť vykonávajúca embedding naučená na rozsiahlejšom texte opisujúcom každú z arytmií v snahe vytvoriť algoritmus, ktorý by bol schopný zachytiť vzťahy medzi slovami ešte efektívnejšie, avšak tento prístup iba miatol použitú neurónovú sieť, ktorá vytvárala predikcie vo forme slovných spojení, ktoré nedávali zmysel.

Algoritmus bol pri učení nastavený tak, aby kategórie vkladal do 40 miestneho vektora s tým aby bola maximálna dĺžka medzi jednotlivými slovami 1 a aj minimálny počet slov, ktoré má brať v úvahu je tiež 1. Toto nastavenie parametrov znamená, že bude vkladat aj jednotlivé slová do vektora, nie iba viac slovné pomenovania. Obr.2.6 ukazuje príklad vloženie slova „atrial“ do 40-rozmerného priestoru algoritmom Word2Vec. Ako bolo v teoretickej časti vysvetlené, algoritmus Word2Vec

```
['atrial']
[-2.4008876e-02  1.2518233e-02 -2.1898966e-02 -1.0979563e-02
-8.7749955e-05 -7.4045657e-04 -1.9153100e-02  2.4036855e-02
 1.2455141e-02  2.3082858e-02 -2.0394797e-02  1.1239493e-02
-1.0342693e-02  2.0613372e-03  2.1246547e-02 -1.1155444e-02
 1.1293751e-02 -1.6967405e-02 -8.8712219e-03  2.3496270e-02
-3.9441348e-03  8.0342888e-04 -1.0351574e-02 -1.9206721e-02
-3.7700236e-03  6.1744871e-03 -2.2200705e-03  1.3834154e-02
-6.8574427e-03  5.6501627e-03  1.3639485e-02  2.0864880e-02
-3.6343515e-03 -2.3020357e-02  1.0926378e-02  1.4294625e-03
 1.8604767e-02 -2.0332099e-03 -6.5960349e-03 -2.1882523e-02]
```

Obr. 2.6: Vloženie slova „atrial“ do 40-rozmerného priestoru pomocou algoritmu Word2Vec.

umožňuje vektorovo orientované uvažovanie založené na posunoch medzi slovami, čo v praxi znamená, že ak chceme slovnému spojeniu „atrial fibrillation“ vytvoriť vektorovú reprezentáciu vo vytvorenom 40-dimenzionálnom priestore, tak nám stačí jednoducho sčítať vektory reprezentujúce slová „atrial“ a „fibrillation“. Poslednou úlohou pri tvorbe nového značenia kategórii bolo vysporiadať sa s negatívnymi hodnotami, ktoré word embedding vytvoril, keďže sieť je schopná akceptovať iba reprezentácie pozitívnymi číslami. Prvotným nápadom bolo dať všetky negatívne hodnoty rovné nule, čím sa ale stratilo príliš veľké množstvo informácie potrebné pre správny chod kategorizácie. Použitým prístupom bolo nakoniec vytvorenie vektorov dvojnásobnej veľkosti toho pôvodného, čiže dĺžky 80, s tým že v prvej polovici vektoru sa nachádzajú hodnoty, ktoré boli pozitívne od začiatku a v druhej polovici sa nachádzajú absolútne hodnoty tých negatívnych.(Obr.2.7) Čiže ak od prvej polovice tohto novovytvoreného vektoru odčítame jeho druhú polovicu tak dostaneme náš pôvodný 40 miestny vektor.

```

['sinus', 'bradycardia']
[0.00453037 0.          0.0337302  0.          0.          0.
 0.0038151  0.03342794 0.          0.00737003 0.04336338 0.
 0.          0.00214603 0.          0.00242652 0.02441914 0.01773242
 0.0031332  0.          0.03802459 0.          0.          0.00101778
 0.00812762 0.01122237 0.01248043 0.01055728 0.          0.
 0.00076497 0.          0.02333147 0.          0.          0.01611007
 0.04272054 0.00151402 0.          0.00739167 0.          0.01070651
 0.          0.00212223 0.00634577 0.01050598 0.          0.
 0.01688751 0.          0.          0.01473979 0.01283988 0.
 0.00252334 0.          0.          0.          0.          0.0004385
 0.          0.00479811 0.00599543 0.          0.          0.
 0.          0.          0.01502669 0.00536516 0.          0.02610058
 0.          0.00962493 0.00818186 0.          0.          0.
 0.00166615 0.          ]

```

Obr. 2.7: Vytvorenie vektorovej reprezentácie pre slovné spojenie „sinus bradycardia“ obsahujúcej iba pozitívne hodnoty.

### 2.7.1 Optimalizácia

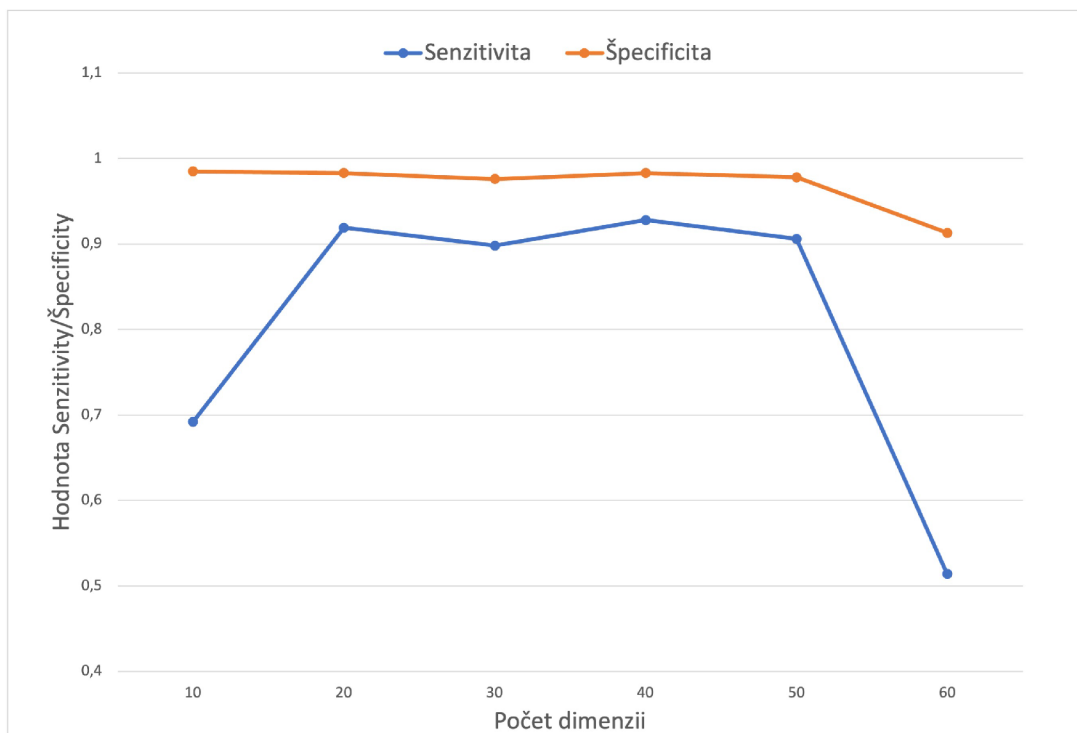
Podobne ako pri metóde kódovania pomocou autoenkóderu bolo potrebné prísť na optimálny počet dimenzií do ktorých budú slová vkladané, tak aby bola klasifikácia konvulčnou sieťou čo najefektívnejšia. Narozdiel od autoenkóderu zvýšenie dimenzií iba o 2 nemalo zjavný vplyv na výkon siete. Navyšovanie preto prebehlo po desiatkách. Graf na Obr.2.8 znázorňuje závislosť hodnôt senzitivity a špecificity konvulčnej siete na počte dimenzií do ktorých boli jej výstupy kódované. Je vidno, že najvyššie hodnoty sieť dosiahla pri počte 40.

### 2.7.2 Architektúra

Podobne ako pri ostatných druhoch kódovania bola využitá rovnaká architektúra siete so zmenou v poslednej vrstve, kde musela byť veľkosť výstupu siete zmenená na hodnotu 80, čo odpovedá dvojnásobku dimenzionality priestoru do ktorého sú názvy arytmií vložené.(Tab.2.14) Kriteiálna funkcia použitá v tomto prípade je „binary crossentropy“.

### 2.7.3 Prevod výstupu siete na kategóriu arytmie

Predikcie konvulčnej siete majú tvar vektoru dĺžky 80. Aby sme z tohto vektoru boli schopní dostať interpretovateľnú predikciu, tak musíme najskôr od prvej polovice tohto vektoru odčítať druhú polovicu, čím dostaneme zasadenie predpovede



Obr. 2.8: Závislosť hodnôt senzitivity a špecificity konvolučnej neurónovej siete na počte dimenzií do ktorých boli jej výstupy vložené pomocou algoritmu Word2Vec.

do 40-rozmerného priestoru vytvoreného pomocou algoritmu Word2Vec. Tento algoritmus z knižnice Gensim taktiež obsahuje funkciu „most\_similar“, ktorá vypočíta podobnosti, vo forme kosínovej podobnosti všetkých slov z naučenej slovnej zásoby modelu a predikcie vytvorenej konvolučnou sieťou. Následne sme schopný si vybrať počet najbližších slov ku našej predpovedi. Logicky môžeme uvažovať, že nám stačí nájsť dve najbližšie slová a dostaneme tak výslednú predikciu. Po zobrazení výsledkov predpovede bolo zistené, že konvolučná sieť často predpovedá dvojslovné pomenovania v opačnom poradí a taktiež že mnoho razí predpovie slovné spojenie „sinus atrial“. Pre opravu tejto chyby bol navrhnutý algoritmus, ktorý si nechá v prípade takejto chybnej predpovede vypočítať ešte tretie najbližšie slovo a pokiaľ sa jedná buď o slovo „fibrillation“ alebo „flutter“, tak mu automaticky priradí slovo „atrial“ a ak sa jedná o slová „rhythm“, „bradycardia“ alebo „tachycardia“, tak im priradí slovo „sinus“. Následne algoritmus prejde ešte všetky ostatné predpovede a uistí sa, že dvojslovné pomenovania sú v správnom poradí.



Tab. 2.14: Architektúra využitá na učenie modelov so značením výstupov pomocou algoritmu Word2Vec.

Vrstva	Parametre	Aktivačná funkcia
Conv1D	32 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	32 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Conv1D	64 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Conv1D	128 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	128 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Conv1D	265 filtrov, kernel=5	ReLU
BatchNormalization		
MaxPool	okno=2	
Dropout	0,2	
Conv1D	256 filtrov, kernel=5	ReLU
MaxPool	okno=2	
Dropout	0,2	
Flatten		
Dense	128	ReLU
Dense	<b>80</b>	Sigmoid

## 2.7.4 Vyhodnotenie

Kódovaním výstupu metódou word embedding sa podarilo dosiahnuť 93% senzitivitu konvolučnej neurónovej siete. Jej špecificita dosiahla hodnoty 98,3%. (Tab.2.15) Hodnoty o efektívnosti najlepšieho modelu (model 1) predpovedať jednotlivé druhy arytmií sú zhrnuté v Tab.2.16. Je z nej zjavné, že model bol náchylnejší na nerovnomerné zastúpenie údajov v tréningovej množine viac ako modely využívajúce

Tab. 2.15: 5-násobná krížová validácia modelov so značením výstupov pomocou algoritmu Word2Vec.

Model	Senzitivita	Špecificita
1	0,945	0,987
2	0,911	0,978
3	0,942	0,986
4	0,929	0,984
5	0,921	0,981
priemer	0,93	0,983
$\sigma$	0,014	0,004

konvenčné značenie kategórií, ale zároveň menej model využívajúci autoenkóderu.

Tab. 2.16: Úspešnosť siete so značením výstupov pomocou algoritmu Word2Vec detekovať jednotlivé druhy arytmií.

Trieda	Senzitivita	Špecificita
atrial fibrillation	0,973	0,967
atrial flutter	0,76	0,992
sinus rhythm	0,932	0,996
sinus bradycardia	0,976	0,992
sinus tachycardia	0,993	0,988

## 2.8 Výsledky a diskusia

Na začiatku práce bola navrhnutá 1D konvolučná neurónová sieť, vytvorená na klasifikáciu 10-sekundových záznamov EKG zo všetkých dvanástich zvodov. Aby bolo možné porovnať vplyv rôznych techník kódovania labelov na klasifikačnú schopnosť siete, architektúra sa už počas testov nemenila. Pri vyhodnocovaní EKG záznamov je dôležitejšie zachytiť danú arytmiu a správne ju zaradiť, ako schopnosť povedať o akú arytmiu sa nejedná. Z tohto dôvodu sa v priebehu práce pri optimalizácii parametrov, ale aj vyhodnocovaní výsledkov kládol väčší dôraz na senzitivitu. Najsmerodatnejším parametrom pri porovnávaní vplyvu kódovania labelov na celkový výkon konvulčnej siete je priemerná senzitivita každého z modelov získaná z 5-násobnej krížovej validácie.

Z Tab.2.17 je zjavné, že metóda word embeddingu skutočne má pozitívny vplyv na výkon neurónovej siete. Vďaka použitiu tejto metódy bolo možné zvýšiť priemernú senzitivitu siete o jedno celé percento oproti konvenčným metódam. Jedno percento

Tab. 2.17: Priemerná senzitivita a špecificita každého z modelov získaná z 5-násobnej krížovej validácie.

Druh kódovania labelov	priemerná senzitivita	$\sigma$
one-hot	0,92	0,019
ordinálne	0,919	0,023
autoenkóder	0,908	0,028
word embbeding	0,93	0,014
	priemerná špecificita	$\sigma$
one-hot	0,98	0,005
ordinálne	0,98	0,005
autoenkóder	0,978	0,007
word embbeding	0,983	0,004

nevyzerá ako veľa pri veľkosti datasetu aký bol použitý v tejto práci, ale pri značne väčších datasetoch, ktoré by sa použili v reálnej praxi by sa jednalo o výrazné zlepšenie. Ďalším faktorom na ktorý je potrebné poukázať je veľkosť smerodajnej odchýlky, ktorá je tiež najnižšia pri metóde word embbedingu. Tento fakt značí o najväčšej robustnosti spomedzi modelov. Tabuľka ďalej ukazuje, že využitie navrhnutého autoenkóderu nemalo na sieť pozitívny vplyv. Ten naopak znížil výkon siete o jedno percento a dosiahol najväčšiu smerodajnú odchýlku spomedzi modelov. Modeli konvenčného kódovania dosiahli takmer rovnaké výsledky. Je zaujímavé, že pridanie ordinálnej hodnoty do kódovania labelov siete nepomohlo a znížilo to jej výkon o desatinu percenta. Tieto hodnoty sa však menia ak analyzujeme bližšie najúspešnejšie modely z krížovej validácie. Práve model s ordinálnym spôsobom kódovania labelov sa delí s modelom využívajúcim word embbeding o najvyššie zastúpenie v Tab.2.18. Tab.2.18 znázorňuje, ktoré modely boli najúspešnejšie v detekcii jednotlivých arytmií. Hodnota, ktorá z tabuľky vyčnieva je 100% senzitivita a 99,3% špecificita modelu s one-hot značením výstupov pri detekcii sínusovej bradykardie.

Tab. 2.18: Tabuľka modelov, ktoré boli najúspešnejšie pri detekcii jednotlivých druhov arytmií.

Slovné označenie	Druh kódovania labelov	Senzitivita	Špecificita
sinus rhythm	ordinálne	0,985	0,991
sinus bradycardia	one-hot	1	0,993
sinus tachycardia	word embbeding	0,993	0,988
atrial flutter	ordinálne	0,863	0,972
atrial fibrillation	word embbeding	0,973	0,967

Skutočnosť, že kategória „atrial flutter“ bola v použitom datasete najmenej zastúpená, umožnila skúmať vplyv nerovnomerného zastúpenia kategórii pri procese učenia na modely s rôznym spôsobom značenia výstupov. Modely využívajúce značenie autoenkóderom nielenže produkujú najslabšie výsledky, ale sa aj javia najnáchyľnejšie na túto skutočnosť. Na druhú stranu modely s ordinálnym značením vykazujú, že im tento fakt zavádza najmenšiu chybu do ich klasifikačného procesu. Pri výbere, ktorú možnosť značenia využiť pri učení siete je nutné brať v úvahu viacero faktorov. Za prvé je komplexnosť riešeného problému. Ak sa jedná o jednoduchú klasifikáciu na menšom datasete, je zbytočne komplikované vymýšľať, optimalizovať a aplikovať spôsob word embeddingu, keďže komplexnosť tohto predspracovania môže prekročiť komplexnosť celého riešeného problému. Tab.2.19 ďalej ukazuje, že použitím rôznych metód kódovania sa môže predĺžiť aj čas učenia. Ak sa však jedná

Tab. 2.19: Rozdiely v dĺžke učenia sa jednotlivých modelov pri využití rôznych druhov značenia labelov.

<b>Druh kódovania labelov</b>	<b>Počet epóch učenia</b>
ordinálne	10-22
one-hot	8-19
ordinálne	19-29
word embedding	25-37

o komplexnejší problém a jednotlivé kategórie, ktoré sú v tomto probléme klasifikované majú medzi sebou vzťahy, ktoré by sa mohli pri učení neurónovej siete odzrkadliť a čas učenia spolu s prístupom ku výkonnej výpočtovej technike nie je problém, je výhodné siahnuť po metóde word embeddingu. Práca ukázala, že táto metóda jednoznačne umožňuje zachytiť vzťahy medzi jednotlivými slovnými pomenovaniami kategórii, čo sa aj vo výsledku prejaví v klasifikačnej schopnosti použitej neurónovej siete.

## Záver

V teoretickej časti diplomovej práce bolo vysvetlené na čo slúži elektrokardiograf, ako vznikajú elektrické impulzy, ktoré meria a ako sú distribuované elektrickým prevodovým systémom srdca. Práca tiež ukazuje aký je rozdiel medzi normálnym srdcovým rytmom a srdcovými arytmiami a čo môže tieto arytmie spôsobiť. Práca ďalej vysvetľuje prečo je dôležité včasne detekovať srdcové arytmie a aké sa na to používajú v dnešnej dobe metódy. Bližšie sa zameriava na metódu hlbokého učenia pri ktorej objasňuje spôsob učenia viacvrstvových neurónových sietí a ich rôzne druhy. Detailne sú vysvetlené princípy konvolučných neurónových sietí spolu s možnosťami ich optimalizácie. V poslednej časti sú vysvetlené možnosti značenia výstupov neurónových sietí pomocou konvenčných metód a bližšie pomocou metódy word embedding a autoenkóderov.

V praktickej časti sú predstavené dostupné EKG dáta a spôsob ich predspracovania. Ďalej je v práci navrhnutá 1D konvolučná neurónová sieť, ktorá je optimalizovaná na klasifikáciu dostupných EKG záznamov. Sú implementované štyri metódy značenia labelov v podobe one-hot značenia, ordinálneho značenia, autoenkóderu a metódy word embedding, konkrétne algoritmus Word2Vec. Na vyhodnotenie výsledkov bola použitá 5-násobná krížová validácia. Týmto spôsobom tak bolo dohromady natrénovaných 22 neurónových sietí. 5 modelov pre každý z typov kódovania labelov plus jeden autoenkóder a jeden model Word2Vec. Natrénované modely boli následne porovnané na základe ich priemernej senzitivity a špecificity, spolu s ich smerodajnými odchýlkami. Ďalej na základe ich schopnosti klasifikovať každú z kategórií arytmií osobitne a na základe dĺžky učenia modelov. Diplomová práca ukazuje, že metóda word embedding jednoznačne zachytáva vzťahy medzi slovnými pomenovaniami jednotlivých arytmií a že dokáže tieto vzťahy preniesť aj učebného procesu siete. S využitím tejto metódy bolo možné dosiahnuť najvyššiu priemernú senzitivitu a špecificitu spomedzi modelov a taktiež sa podarilo zvýšiť robustnosť navrhnutej neurónovej siete. Najvyššia dosiahnutá priemerná hodnota senzitivity je 93% a špecificity 98,3%. Táto hodnota senzitivity je o celé jedno percento vyššia ako druhá najlepšia hodnota, ktorá bola získaná modelom využívajúcim one-hot kódovanie. Sieť dosahujúca najnižšiu senzitivitu (90,8%) bola sieť využívajúca kódovanie labelov pomocou autoenkóderu. Diplomová práca dokázala, že ak sa jedná o klasifikačný problém pri ktorom sa využíva neurónová sieť a existujú vzťahy medzi jednotlivými kategóriami, word embedding je užitočný spôsob akým sa tieto vzťahy dajú zachytiť, preniesť do procesu učenia siete a zvýšiť tak celkovú klasifikačnú schopnosť danej neurónovej siete.



# Literatúra

- [1] *InformedHealth.org [Internet]. Cologne, Germany: Institute for Quality and Efficiency in Health Care (IQWiG); 2006-. What is an electrocardiogram (ECG)? 2019 Jan 31. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK536878/>*
- [2] *HABÁSKO, Jan. MYOKARDITIDA Webový průvodce o nemoci myokarditida, její diagnostice a léčbě.: Elektrokardiogram (EKG) [online]. 2016, 5.2. [cit. 2022-01-02]. Dostupné z: [https://myokarditida.cz/cs\\_CZ/diagnostika-a-lecba/diagnostika/elektrokardiogram-ekg/](https://myokarditida.cz/cs_CZ/diagnostika-a-lecba/diagnostika/elektrokardiogram-ekg/)*
- [3] *MITCHELL, L.Brent. Overview of Arrhythmias [online]. Libin Cardiovascular Institute of Alberta, University of Calgary, 2021 [cit. 2022-01-02]. Dostupné z: <https://www.msmanuals.com/professional/cardiovascular-disorders/arrhythmias-and-conduction-disorders/overview-of-arrhythmias>*
- [4] *LECUN, Yann, Yoshua BENGIO a Geoffrey HINTON. Deep learning. Nature [online]. 2015, 521(7553), 436-444 [cit. 2019-12-24]. ISSN 0028-0836. Dostupné z: [doi:10.1038/nature14539](https://doi.org/10.1038/nature14539)*
- [5] *NIELSEN, Michael A. Neural Networks and Deep Learning [online]. Determination Press, 2015 [cit. 2020-03-12]. Dostupné z: <http://neuralnetworksanddeeplearning.com/chap1.html>*
- [6] *KUMARASWAMY, Balachandra. 6 - Neural networks for data classification. Artificial Intelligence in Data Mining [online]. Academic Press, 2021, s. 109-131 [cit. 2022-01-02]. ISBN 9780128206010. Dostupné z: <https://doi.org/10.1016/B978-0-12-820601-0.00011-2>*
- [7] *GUPTA, Tarun Kumar a Khalid RAZA. Machine Learning in Bio-Signal Analysis and Diagnostic Imaging: Chapter 7 - Optimization of ANN Architecture: A Review on Nature-Inspired Techniques [online]. Academic Press, 2019, s. 159-182 [cit. 2022-01-03]. ISBN 9780128160862. Dostupné z: <https://doi.org/10.1016/B978-0-12-816086-2.00007-2>*
- [8] *ANDREOTTI, Fernando, Oliver CARR, Marco A F PIMENTEL, Adam MAHDI a Maarten DE VOS. Comparing Feature Based Classifiers and Convolutional Neural Networks to Detect Arrhythmia from Short Segments of ECG [online]. In: . 2017-9-14, s. - [cit. 2022-01-03]. Dostupné z: [doi:10.22489/CinC.2017.360-239](https://doi.org/10.22489/CinC.2017.360-239)*

- [9] HONCHAR, Alexandr. *Deep learning: the final frontier for signal processing and time series analysis?* Medium [online]. 2018, 10. November [cit. 2022-01-03]. Dostupné z: <https://alexrachnog.medium.com/deep-learning-the-final-frontier-for-signal-processing-and-time-series-analysis-734307167ad6>
- [10] LEE, June-Goo, Sanghoon JUN, Young-Won CHO, Hyunna LEE, Guk Bae KIM, Joon Beom SEO a Namkug KIM. *Deep Learning in Medical Imaging: General Overview*. Korean Journal of Radiology [online]. 2017, 18(4) [cit. 2020-03-11]. DOI: 10.3348/kjr.2017.18.4.570. ISSN 1229-6929. Dostupné z: <https://synapse.koreamed.org/DOIx.php?id=10.3348/kjr.2017.18.4.570>
- [11] ALBAWI, Saad, Tareq Abed MOHAMMED a Saad AL-ZAWI. *Understanding of a convolutional neural network*. 2017 International Conference on Engineering and Technology (ICET) [online]. IEEE, 2017, 2017, , 1-6 [cit. 2019-12-30]. DOI: 10.1109/ICEngTechnol.2017.8308186. ISBN 978-1-5386-1949-0. Dostupné z: <https://ieeexplore.ieee.org/document/8308186/>
- [12] SRIVASTAVA, Nitish, Geofrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER a Ruslan. *Dropout: a simple way to prevent neural networks from overfitting*. The Journal of Machine Learning Research. 2014, 15(1), 1929–1958. ISSN 1532-4435.
- [13] *Mechanical Systems and Signal Processing: 1D convolutional neural networks and applications: A survey* [online]. 2021. 2021 [cit. 2022-01-04]. ISSN 0888-3270. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0888327020307846>
- [14] BROWNLEE, Jason. *3 Ways to Encode Categorical Variables for Deep Learning*. Machine Learning Mastery [online]. 2019, 19. November [cit. 2022-01-05]. Dostupné z: <https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/>
- [15] SAXENA, Shipra. *Here's All you Need to Know About Encoding Categorical Data (with Python code)*. Analytics Vidhya [online]. 2020 [cit. 2022-01-04]. Dostupné z: <https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/>
- [16] BROWNLEE, Jason. *What Are Word Embeddings for Text?* Machine Learning Mastery [online]. Deep Learning for Natural Language Processing, 2017 [cit. 2022-01-05]. Dostupné z: <https://machinelearningmastery.com/what-are-word-embeddings/>



- [17] VERMA, Yugesh. *Word2Vec vs GloVe – A Comparative Guide to Word Embedding Techniques*. *Analytics India Mag* [online]. DEVELOPERS CORNER, 2021, 19. Október [cit. 2022-01-05]. Dostupné z: <https://analyticsindiamag.com/word2vec-vs-glove-a-comparative-guide-to-word-embedding-techniques/>
- [18] *An overview of word embeddings and their connection to distributional semantic models*. *Aylien* [online]. [cit. 2022-01-05]. Dostupné z: <https://aylien.com/blog/overview-word-embeddings-history-word2vec-cbow-glove>
- [19] JORDAN, Jeremy. *Introduction to autoencoders*. [online]. *Data Science*, 2018, 19. Marec [cit. 2022-01-05]. Dostupné z: <https://www.jeremyjordan.me/autoencoders/>
- [20] *OpenStax College. Cardiac Arrhythmias*. *Wikipedia commons* [online]. 2013 [cit. 2022-01-05]. Dostupné z: [https://commons.wikimedia.org/wiki/File:2024\\_Cardiac\\_Arrhythmias.jpg](https://commons.wikimedia.org/wiki/File:2024_Cardiac_Arrhythmias.jpg)
- [21] VAZAN, Yron. *Training an AutoEncoder to Generate Text Embeddings* [online]. 2019, 28. September [cit. 2022-01-06]. Dostupné z: <http://yaronvazana.com/2019/09/28/training-an-autoencoder-to-generate-text-embeddings/>
- [22] PARDASANI, Rohit a Navchetan AWASTHI. *Classification of 12 Lead ECG Signal Using 1D-Convolutional Neural Network With Class Dependent Threshold* [online]. *General Electric Healthcare, Bangalore, Karnataka, India , Massachusetts General Hospital, Harvard University, Boston, Massachusetts, USA* [cit. 2022-01-06]. Dostupné z: <https://www.cinc.org/archives/2020/pdf/CinC2020-277.pdf>
- [23] HSIEH, Chaur-Heh, Yan-Shuo LI, Bor-Jiunn HWANG a Ching-Hua HSIAO. *Detection of Atrial Fibrillation Using 1D Convolutional Neural Network*. *Sensors* [online]. 2020, 20(7) [cit. 2022-05-18]. ISSN 1424-8220. Dostupné z: [doi:10.3390/s20072136](https://doi.org/10.3390/s20072136)