

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Využití neuronových sítí v BCI systémech založených na evokovaných potenciálech SSVEP

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20. července 2020

Dubina Tomáš

Poděkování

Rád bych poděkoval Ing. Lukášovi Vařekovi, Ph.D. za jeho trpělivost, ochotu a hlavně za obrovské množství času, které mi věnoval po celou dobu vypracování bakalářské práce.

Abstract

The work evaluates neural networks in BCI systems based on steady-state visually evoked potentials (SSVEP). We discuss the design of a simple BCI system based on the SSVEP and then implement a part of the classification system. Classification is based on the neural network classification of EEG data. To obtain these data, an experiment scenario was created. The data were acquired at the Department of Computer Science and Engineering at the University of West Bohemia.

Abstrakt

Cílem této práce je prozkoumat využití neuronových sítí v BCI systémech založených na ustálených vizuálně evokovaných potenciálech (SSVEP). V této práci se zabývám návrhem jednoduchého BCI systému, který je založen na SSVEP a následně implementaci části klasifikačního systému. Klasifikační systém je založen na klasifikaci EEG dat zvolenou neuronovou sítí. Aby byla klasifikace možná, byl vytvořen scénář, díky kterému jsem sbíral EEG data od dobrovolných subjektů. Data byla pořízena na katedře informatiky a výpočetní techniky KIV na Západočeské univerzitě.

Obsah

1	Úvod	8
2	Elektroencefalografie	9
2.1	Elektroencefalografie	9
2.2	Elektroencefalograf	9
2.3	Rozmístění elektrod na skalpu	10
2.4	Základní EEG aktivita	11
2.4.1	Druhy mozkových vln	11
2.5	Způsob zpracování signálu	13
2.6	Artefakty	14
3	Evokované potenciály (EP)	15
3.1	VEP - Zrakové evokované potenciály	15
3.1.1	Stimulace VEP	15
3.2	SSVEP - Ustálené vizuální evokované potenciály	17
3.3	AEP - Sluchové evokované potenciály	17
3.4	SEP - somatosenzorické evokované potenciály	18
3.5	MEP - Motorické evokované potenciály	18
3.6	ERP - kognitivní evokované potenciály	18
4	Zpracování EEG dat	19
4.1	Vzorkování signálu	19
4.1.1	Vzorkovací frekvence	20
4.2	Korekce baseline	20
4.3	Volba kanálů	20
4.4	Fourierova transformace	20
4.4.1	Diskrétní Fourierova transformace	21
4.4.2	Symetrie diskretní Fourierovy transformace	22
4.4.3	Frekvence v diskretní Fourierově transformaci	22
4.4.4	Rychlá Fourierova transformace	22
4.5	Výkonová spektrální hustota (PSD)	23
4.5.1	Parametrické metody výpočtu	23
4.5.2	Neparametrické metody výpočtu	24
5	Klasifikace dat	25
5.1	Příznakové metody rozpoznávání	25
5.1.1	Diskriminační funkce	25
5.1.2	Kritérium minimální vzdálenosti	26
5.1.3	Kritérium minimální chyby	26
5.2	Matice záměn (Confusion matrix)	27

6	BCI systémy	29
6.1	Rozdělení BCI systémů	29
6.2	Princip BCI	29
6.3	Aplikace BCI systému	30
6.4	BCI založené na VEP	30
6.4.1	Aplikace BCI systémů na signálech SSVEP	31
6.4.2	Zhodnocení BCI systémů na signálech SSVEP	31
6.5	BCI založené na vlně P300	31
7	Umělé neuronové sítě	33
7.1	Umělý neuron	33
7.2	Aktivační funkce neuronu	34
7.2.1	Aktivační funkce ReLU	35
7.2.2	Aktivační funkce Swish	35
7.2.3	Aktivační funkce Softmax	35
7.3	Struktura neuronové sítě	36
7.4	Dopředné neuronové sítě	36
7.4.1	Perceptronová síť	36
7.5	Vícevrstvá neuronová síť	37
7.6	Algoritmus Back-propagation	37
7.6.1	Princip algoritmu Back-propagation	37
7.7	Problém přeučení sítě	38
7.7.1	Callback - předčasné ukončení	38
7.7.2	Dropout	38
7.7.3	Normalizace dat	38
7.8	Konvoluční neuronové sítě - CNN	39
7.8.1	Konvoluční operace	39
7.8.2	Konvoluční vrstva	39
7.8.3	Maxpool vrstva	40
7.8.4	Plně propojená vrstva	40
7.8.5	Ztrátová funkce	40
7.8.6	Optimalizační algoritmus	41
8	Diskriminační analýza (DA)	42
8.1	Princip diskriminační analýzy	42
8.2	Lineární diskriminační analýza	43
9	Návrh experimentu	44
9.1	BCI systém	44
9.2	Stimulace	44
9.2.1	Frekvence stimulace	45
9.3	Scénář stimulace	46
9.3.1	Průběh měření	47
10	Předzpracování a klasifikace	48
10.1	Předzpracování dat	48
10.1.1	Výchozí data	48
10.1.2	Načtení dat	48
10.1.3	Předzpracování dat do výsledné podoby	50

10.2	Klasifikace	50
10.2.1	Rozdělení vstupních dat	51
10.2.2	Hodnotící metriky	51
10.3	Neuronová síť	52
10.3.1	Model neuronové sítě	52
10.3.2	Kompilace modelu	53
10.3.3	Trénování modelu	53
10.4	LDA	54
10.5	Výstupní data	54
11	Dosažené výsledky	55
12	Diskuze	58
13	Závěr	59
	Literatura	60

1 Úvod

Spolupráce mezi neurovědci a informatiky je nezbytná k tomu, aby se stala stará i nová neurovědní data dostupnější a užitečnější pro vědecké pracovníky a díky tomu, abychom dokázali porozumět činnosti mozku mnohem rychleji, než tomu bylo možné v předchozích dekadách.

Tato spolupráce je klíčová například při snaze pomoci pacientům, kteří utrpěli vážnější zranění a nemohou komunikovat s okolním světem klasickým způsobem. Tito pacienti byli v dobách předešlých odkázáni pouze na intuici svého lékaře, aby správně konstatoval potřeby pacienta a tím ho dokázal efektivně léčit. Avšak v dnešní době díky asistenčním systémům přichází konec této éry a snažíme se o to, abychom pacientovi umožnili komunikaci s okolním světem za pomoci výpočetní techniky.

Sice zatím nejsme ve fázi, kdy dokážeme pacientovi číst jeho myšlenky, ale již dokážeme využít vlastnosti mozkové aktivity k tomu, abychom se k této pomyslné hranici přiblížili. Systémy, které se zabývají problematikou komunikace skrze mozkovou aktivitu se označují jako BCI (brain computer interface) systémy. Díky nim dokážeme vytvářet experimenty, které jsou na správné cestě ke komunikaci mezi mozkem a počítačem, a tím dokázali komunikovat s postiženým pacientem nebo například ovládat robotickou paži.

Abychom dokázali navázat spojení s pacientem, je potřeba vytvořit správnou stimulaci, na kterou jeho mozek dokáže reagovat. Stimulace mozkových drah může být docílena například pomocí ustálených vizuálně evokovaných potenciálů (SSVEP). Tato stimulace je založena na principu blikajícího objektu. Tento objekt na pacienta bliká předem definovanou frekvencí, která je poté přenesena skrze odezvu neuronů do počítače.

Cílem této práce je prozkoumat využití neuronových sítí v těchto BCI systémech a tím přispět k zlepšení těchto systémů. Abychom se mohli plně věnovat této problematice, tak je potřeba si prvně projít teoretické informace a principy, které nám pomohou při realizaci našeho experimentu. Asistenční systémy jsou již v dnešní době běžně používané ve specializovaných zařízeních, ale ke svému širokému uplatnění mají ještě dlouhou cestu.

Práce je rozdělena do 14 kapitol. V kapitole 2 Elektroencefalografie je vysvětleno, co tento pojem znamená a k čemu se využívá. Následuje kapitola 3 pojednávající o evokovaných potenciálech, která obsahuje souhrnné informace o jednotlivých typech evokovaných potenciálů. V kapitole 4 je probráno zpracování EEG signálu. Je zde popsáno jakým způsobem se signál vzorkuje a jak se zpracovávají kanály. Kapitola 6 pojednává o BCI systémech a jejich využití. Poté následuje kapitola 5 popisující samotnou klasifikaci dat pomocí příznakových metod rozpoznávání. V kapitole 7 a 8 jsou rozebrány dva klasifikační algoritmy a to neuronové sítě a LDA. Realizační část bakalářské práce je rozdělena do třech kapitol. Kapitola 9 Návrh experimentu pojednává o tvorbě scénáře experimentu, poté následuje kapitola 10 Předzpracování a implementace, kde jsou informace o způsobu implementace. V poslední kapitole 11 Dosažené výsledky realizační části jsou vypsané výsledky klasifikace obou algoritmů. Následuje diskuze výsledků a samotný závěr.

2 Elektroencefalografie

2.1 Elektroencefalografie

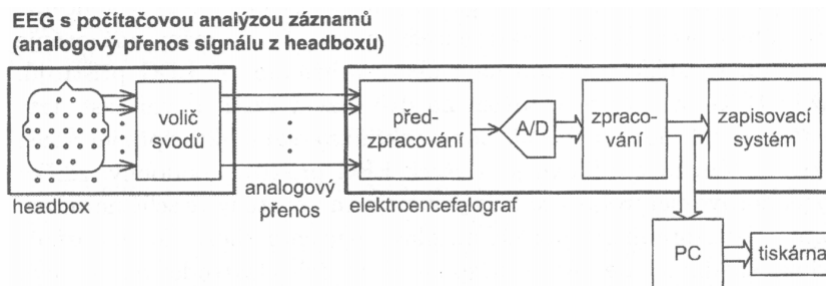
Elektroencefalografie je neurologická vyšetřovací metoda, pomocí které lze sledovat mozkovou aktivitu. Nervové buňky mezi sebou komunikují pomocí proudění nabitých iontů. Vzniká tak elektrický potenciál, šířící se v určité míře do okolí. Díky tomu se dají potenciály měřit na povrchu hlavy. Nicméně kvůli izolaci lebky a měkké tkáni jsou tyto signály velmi slabé, řádově jde o mikrovolty. Pro představu je to tisíckrát menší signál, než při vyšetření srdce pomocí EKG. [1]

Procedura se řadí mezi takzvané neinvazivní metody funkčního vyšetření elektrické aktivity centrálního nervového systému (CNS). V neurologii a psychiatrii se využívá k vyšetření poruch mozku. Touto procedurou lze sledovat stavy vědomí, projevy metabolických poruch, vlivy drog či toxických látek. Záznam, který je při vyšetření zachycen, se nazývá encefalograf a je v něm vidět mozková aktivita pacienta. [1]

2.2 Elektroencefalograf

Elektroencefalograf je přístroj pro vyšetření EEG, který musí být schopen takto slabé signály zesílit a odstranit šum. Přístroj snímá po určitou dobu elektrickou aktivitu z každé elektrody a zaznamenává je do elektroencefalografu. Elementární jednotkou generující potenciálové pole je z fyzikálního hlediska dipól. Ten je tvořen dvěma stejně velkými opačně nabitými náboji, které jsou v prostoru odděleny. Abychom mohli aktivitu dipólu registrovat povrchovými elektrodami, musí dipól generovat pole, které zasahuje do oblasti skalpu a je tam dostatečně silné. [1]

Při měření EEG využíváme biosignály, které už ve své podstatě mají elektrický charakter. Proto základní metodou pro snímání těchto signálů zůstávají elektrody. Pomocí elektrody zprostředkováváme průchod elektrického proudu mezi pacientem a aparaturou. Elektroda je umístěna bezprostředně na tělo pacienta a proto se na její materiál a konstrukci vztahují přísné požadavky. Nejčastěji se elektrody zhotovují z čistého stříbra, některé speciální bývají z nerezové oceli nebo stříbrné pokryté tenkou vrstvičkou zlata. [9]



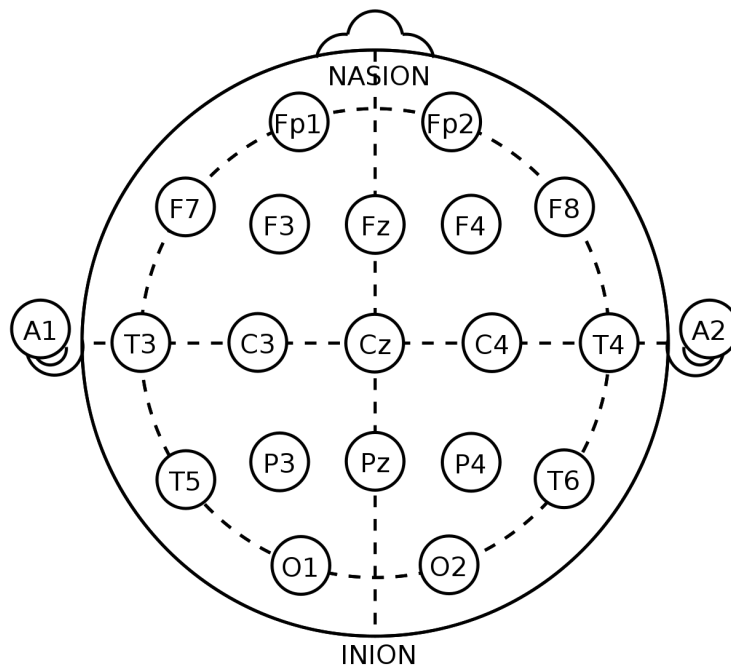
Obrázek 2.1: Elektroencefalograf s počítačovou analýzou signálu [48]

2.3 Rozmístění elektrod na skalpu

Rozmístění elektrod je možné různými způsoby, ale typicky se používá tzv. systém 10/20 (čtyřdeset – dvacet) (Obrázek 2.2), jehož název vznikl ze způsobu rozdělení elektrod. Elektrody jsou rozmístěny po obvodu hlavy v segmentech po 10% a 20%. Analogicky se rozdělí i zbylé roviny a výsledkem je síť bodů, která snímá celý povrch lebky. Následně jsou na tyto body rozmístěny snímající elektrody. Elektrody umístěné nejvíce vpředu nazýváme prefrontální (oblast za našimi očima a čelem), za nimi je rozmístěna řada elektrod frontálních, dále následují elektrody centrální, pak temenní a nejvíce vzadu jsou elektrody týlní. Po stranách umístíme elektrody temporální.[2]

Elektrody se označují:

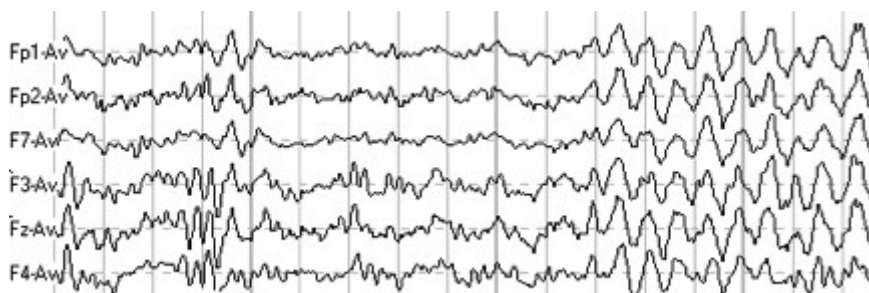
1. písmenem, které odpovídá oblasti mozku, nad kterou je elektroda umístěna
2. číslem - lichá čísla pro levou hemisféru, sudá pro pravou hemisféru



Obrázek 2.2: Rozmístění elektrod systémem 10-20 [49]

2.4 Základní EEG aktivita

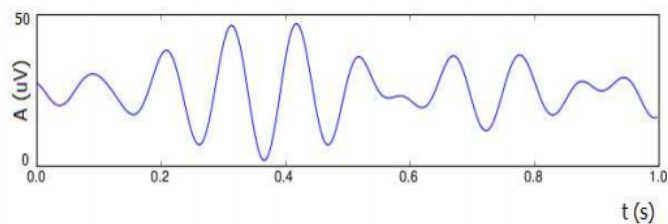
Mozková aktivita závisí na věku a aktuálním stavu vyšetřovaného během registrace (zejména z hlediska bdělosti). Nejznámější je aktivita alfa s frekvencí cca 8 - 14 Hz, kterou sledujeme u dospělých v týlní oblasti hlavy při zavřených očích. Pomalejší frekvence (theta a delta) mohou být v bdělém stavu u dospělých patologickým příznakem. Během spánku jsou naproti tomu identifikátorem různých spánkových stádií, čehož se využívá ve spánkových laboratořích; u dětí mohou být tyto frekvence měřítkem vyzrállosti CNS. Průběh signálu vidíme na Obrázek 2.3. [24]



Obrázek 2.3: Příklad EEG signálu [50]

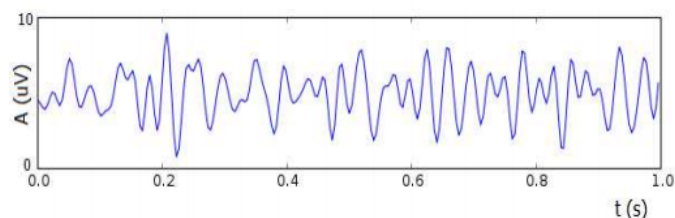
2.4.1 Druhy mozkových vln

- **Aktivita alfa (alfa rytmus)** – je nejvýraznější aktivitou EEG; pro zdravého dospělého člověka je typická v klidu, vleže, ve stavu klidného bdění při zavřených očích (otevřením očí, zvýšením pozornosti a duševní činnosti se tlumí) [24]; jedná se o rytmus o frekvenci 8 – 14 Hz; amplituda může být různá, u dospělých většinou pod $50 \mu\text{V}$ [9]; maximální amplituda tohoto rytmu se zaznamenává nad temenní a týlní oblastí mozku; alfa rytmus je korového původu. Průběh alfa vln můžeme pozorovat na obrázku 2.4.



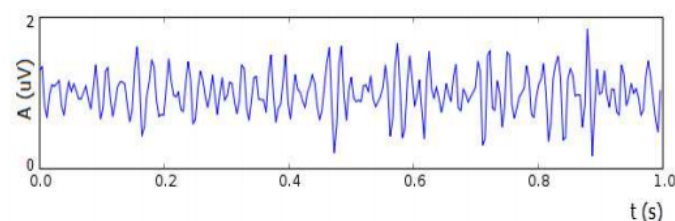
Obrázek 2.4: Průběh alfa vln [51]

- **Aktivita beta (beta rytmus)** – má frekvenci vyšší než 14Hz (rozmezí 14 – 30Hz); její amplituda nepřesahuje $20\mu\text{V}$ [9]; všeobecně se spojuje s aktivací CNS, se zvýšenou pozorností a napětím CNS; je ovlivnitelná volnými pohyby a bolestivými podněty [24]; je nejvýraznější nad frontálními laloky mozku. Průběh beta vln je znázorněn na obrázku 2.5.



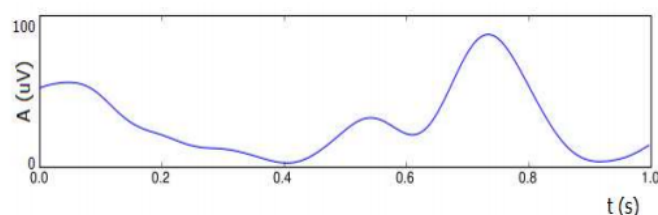
Obrázek 2.5: Průběh beta vln [51]

- **Aktivita gama (gama rytmus)** – má nejvyšší frekvenční pásmo EEG (30 – 50 Hz) [9]; je nestabilní a epizody konstantní frekvence trvají obvykle 100 – 300 ms s intervaly 15 – 30 ms; [4]. Průběh gama vln je znázorněn na obrázku 2.6.



Obrázek 2.6: Průběh gamma vln [51]

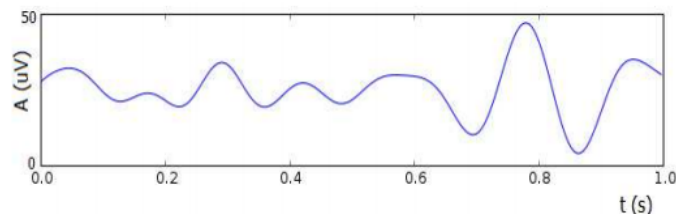
- **Aktivita delta (delta rytmus)** – je rytmus s nejnižší frekvencí (0.5 – 4Hz) vysokovoltážních vln ($20 - 200\mu\text{V}$) [9]; u dospělých osob je spojen s nehlubší fází spánku [24]; z vývojového hlediska je to elektrická aktivita mozku kojeneckého věku[4]. Průběh delta vln můžeme pozorovat na obrázku 2.7.



Obrázek 2.7: Průběh delta vln [51]

- **Aktivita theta (theta rytmus)** – má frekvenci 5 – 7 Hz, její amplituda se pohybuje v rozmezí 5 – $100\mu\text{V}$ [9]; u dospělého člověka se zaznamenává v septohipokampovém systému (součástí velkého mozku), ale do povrchového záznamu se prosazuje jen velmi málo; objevuje se v pozdních fázích spánku; theta rytmus je

typickým rytmem zrajícího mozku dětského věku (nejvýrazněji ve 2. a 3. roce života v temenní a temporální oblasti mozkové kůry) [24]. Průběh theta vln můžeme pozorovat na obrázku 2.8.



Obrázek 2.8: Průběh theta vln [51]

- **Lambda vlny** – jsou bifázické nebo trifázické vlny trvání 200 – 300 ms, nízké voltáže, většinou do $20\mu\text{V}$ [9], hrotového charakteru; lze je registrovat u bdělých osob sledujících očima nějaký předmět [24].

U dětí a starých lidí se objevují na EEG specifické odlišnosti v aktivitě mozku. U dětí dochází od narození do 6 let k výrazným změnám EEG. Teprve ve věku 15 – 20 let se elektroencefalogram stabilizuje. Alfa aktivita je dominující. Ve stáří dochází ke zpomalení rytmu alfa, zmnožení aktivity beta, začíná se objevovat a později narůstá aktivita theta a delta. [24]

2.5 Způsob zpracování signálu

Jedním ze způsobů jak zesílit nízké signály a minimalizovat vliv rušení je použití diferenčního zesilovače. Jedná se o citlivý zesilovač s velkým napěťovým zesílením a se dvěma vstupy (aktivní a referenční). Diferenční zesilovač zesiluje napěťový rozdíl neboli diferenci mezi oběma vstupy. [2]

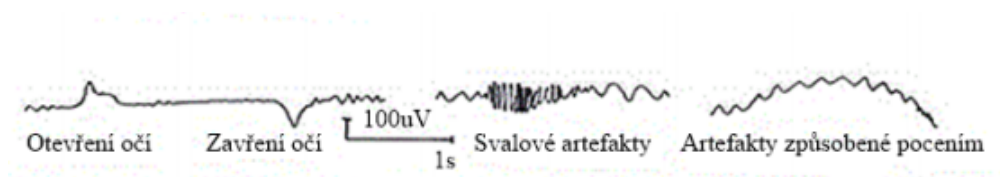
Za pomoci horního a dolního filtru vybereme pouze frekvenční pásmo, které je pro EEG podstatné, obvykle 0.5 až 70 Hz. Dále je odfiltrována rušivá síťová frekvence 50 Hz. V EEG přístrojích lze hodnoty filtru nastavit stupňovitě, buď individuálně pro jednotlivé kanály nebo současně pro všechny kanály dohromady. Dolní filtr zabraňuje přenosu frekvence pod 0.5 Hz (tedy i stejnosměrného proudu) a je řazen před diferenciální zesilovač. Horní filtr znemožňuje přenos frekvencí nad požadované pásmo (>70 Hz) a je řazen za diferenciální zesilovač. Umístění horního a dolního filtru vůči diferenciálnímu zesilovači je dáno vztahem kombinace odporu a kondenzátoru k frekvenci signálu a tedy její filtraci. Strmost filtru udává, jak je frekvenční pásmo ohraničeno (čím větší strmost, tím kvalitnější filtr). [9]

2.6 Artefakty

Za artefakty jsou považovány všechny grafoelementy (tvarové charakteristiky grafických záznamů funkcí organismu), které nemají původ v elektrické mozkové aktivitě. Můžeme je rozdělit na artefakty biologického (Obrázek 2.9) a technického původu (Obrázek 2.10). [5]

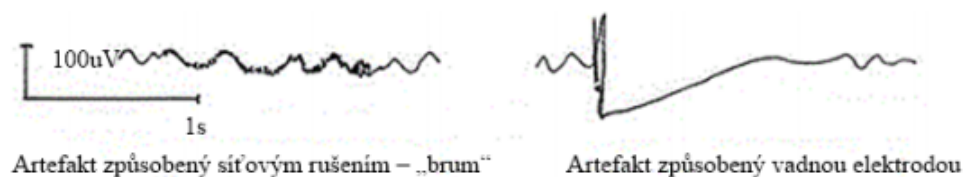
Biologické artefakty jsou například:

- Artefakt pulzový – bývá způsoben umístěním elektrody nad pulsující arterii [5]
- Artefakt EKG – představuje změny elektrického pole, ke kterým dochází při srdeční akci [5]
- Artefakty z dýchacích pohybů – jsou způsobeny pohybem elektrod a elektrodových kabelů při dýchání [5]
- Artefakty od mrkání – jsou způsobeny lehkým vytočením oční koule při mrknutí [5]
- Svalové artefakty – původ je ve stažení svalů pod elektrodou nebo v její blízkosti (nedostatečná relaxace pacienta, svírání víček, mimických svalů, zatínání zubů); jedná se vlastně o elektromyografii (EMG) [5]



Obrázek 2.9: Příklady biologických artefaktů [52]

Původem technických artefaktů jsou například vadné elektrody, špatný kontakt elektrody se skalpem pacienta. Může se stát, že pokud je přístroj napájen ze sítě, tak se nám v datech objeví vlnění o frekvenci 50Hz. Dále například zvonění mobilního telefonu, vypnutí/zapnutí osvětlení, ventilačního systému, klimatizace. [5]



Obrázek 2.10: Příklady technických artefaktů [52]

3 Evokované potenciály (EP)

Evokované potenciály jsou změny nejen elektrické aktivity mozku, ale i jiných částí nervové soustavy po působení úmyslného podnětu z vnějšího prostředí. Nejedná se tedy o spontánní biosignály vznikající v živých organizmech. Slouží nám k zhodnocení funkčního stavu příslušné nervové dráhy. Evokované potenciály testují, jak dlouho mozku trvá přijmout a interpretovat zprávy (zrakové, sluchové...). Za normálních okolností jsou reakce mozku téměř okamžité. Při poruše nervového přenosu (např. při roztroušené skleróze) trvá přenos zpráv déle. Výhodou vyšetření pomocí evokovaných potenciálů je rychlost získání výsledků. Jelikož ale musíme provádět průměrování velkého počtu epoch, došlo k výraznému rozšíření této oblasti medicíny, až s nástupem výpočetní techniky, která dokáže tyto úkony provádět automatizovaně. Další výhodou je, že se jedná o neinvazivní vyšetření a pro pacienta nenese v podstatě žádné riziko. [25]

Podle konkrétního druhu podnětu, kterým je reakce evokována, lze zaznamenávat následující EP [40]:

- VEP (zrakové EP = visual EP)
- SSVEP (ustálené zrakové EP = steady state visual EP)
- AEP (sluchové EP = acoustic EP)
- SEP (somatosenzorické EP = somatosensoric EP)
- MEP (motorické EP = motoric EP)
- ERP (kognitivní EP = event-related potential)

Každá z těchto drah je stimulována určitým podnětem, který je neměnný. Při dostatečném počtu podnětů můžeme výsledky zprůměrovat a dostat sérii výsledků, které odpovídají vyšetřované dráze CNS. [40]

3.1 VEP - Zrakové evokované potenciály

Zrakové evokované potenciály (zkr. VEP z angl. visually evoked potentials) jsou změny elektrické aktivity mozku a jiných částí nervové soustavy, vyvolané krátkým vizuálním stimulem. Zrakové evokované potenciály patří k pomocným diagnostickým metodám v neurologii, typicky se používají k diagnostice demyelinizačních onemocnění (např. roztroušená skleróza) a při poškození nervů (zejména n. opticus). [41]

3.1.1 Stimulace VEP

Existují dva základní způsoby pro stimulaci zrakové dráhy. První je stimulace zábleskem, která se nejčastěji využívá u nespolupracujících pacientů. Druhý způsob je stimulace pomocí strukturovaného podnětu, který zpravidla obsahuje tmavé a světlé plochy. Můžeme buď použít stimulaci obrazcem, který se objevuje a mizí, ale častěji se používá technika záměny inverzních vzorů. V tomto případě zůstává celkový jas stejný a podnět vyvolává

změny jasů v určitých místech obrazu. [6]

Parametry ovlivňující záznam:

- Velikost jednotlivých prvků a stimulační plochy
- Jas stimulační plochy
- Kontrast mezi prvky
- Frekvence stimulace

Velikost jednotlivých prvků a stimulační plochy ovlivňuje celkový výsledek měření. Můžeme stimulovat buď celé zorné pole (tzv. full-field stimulation) nebo polovinu zorného pole (tzv. half-field stimulation). Zvětšování stimulační plochy se mírně zvyšuje amplituda vlny P100. [6]

Pro stimulaci se nejčastěji využívá střídání buď horizontálních nebo vertikálních pruhů, popřípadě čtverců tvořících šachovnici. Velikost jednotlivých tvarů je rozhodující faktor. Při zmenšování ploch dochází ke zmenšení amplitudy a prodloužení latence vln. Pokud budeme plochy zvětšovat, může se stát, že hlavním složkou stimulace se stane jas a záznam se přiblíží spíše stimulaci pomocí záblesku. [6]

Jas stimulační plochy ovlivňuje výsledek tak, že při jeho snížení dochází ke zmenšení amplitudy a prodloužení latence vln. Snížení jasů o jednu logaritmickou jednotku dokáže zmenšit amplitudu o 18% a prodlouží latenci o 18 ms. [6]

Kontrast mezi prvky výrazně ovlivňuje záznam. Vyjadřuje se v procentech a lze jej vypočítat jako,

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} * 100$$

kde L_{max} je luminance světlých prvků a L_{min} je luminance tmavých prvků. Pokud budeme zvyšovat kontrast až do pásma saturace (20-40%), tak to vede ke zvyšování amplitud a snižování latence vln. Další zvyšování kontrastu již není tak znatelné na amplitudě. Byly provedeny i experimenty s jinou barvou než černá a bílá (např. červená-zelená, červená-černá), ale na záznam to nemělo patrný vliv. Při použití různých barev na stimulaci ustálených vizuálních potenciálů se výsledky liší (viz dále). [6]

Frekvence stimulace je určena jako počet změn stimulačního prvku za jednu sekundu a uvádí se v hertzích (Hz). Standardně se využívá frekvence 2 Hz, kdy lze vyhodnotit jednotlivé reakce na stimuly. Při zvyšování frekvence nad 5 Hz začínají vlny interferovat a vytváří tak sinusový průběh, který odpovídá frekvenci stimulace. Tento typ záznamu je označován jako steady-state VEP (SSVEP - ustálený vizuálně evokovaný potenciál). [6]

3.2 SSVEP - Ustálené vizuální evokované potenciály

Ustálené vizuální evokované potenciály (SSVEP) jsou speciálním případem VEP, kdy je sítnice buzena v periodických cyklech. Nejsou tedy evokovány jednotlivé vlny, ale vytváří se zde periodické sinusové vlnění, které má stejnou frekvenci jakou je buzena sítnice. [11]

Stimulace

Způsoby stimulace zrakové dráhy jsme si již popsali. Jelikož jsou pro stimulaci SSVEP potřeba vyšší frekvence, plynou z toho jistá omezení ze strany stimulace pomocí monitoru. U stimulace pomocí monitoru si musíme dát pozor na to, že většina monitorů má obnovovací frekvenci 60 Hz, takže z toho vyplývají omezení pro maximální frekvence. Pokud tedy pro stimulaci použijeme střídání pozadí a nějakého objektu, dosáhneme maximální frekvence 30 Hz. [11] U stimulace pomocí diody, můžeme narazit na problém s velikostí stimulační plochy.

Frekvence

Frekvence, kterými je třeba SSVEP stimulovat se nacházejí v rozsahu od 5 Hz až do 75 Hz. Přičemž jsou rozděleny do třech pásem.

- Nízké - rozmezí 5 - 12 Hz
- Střední - rozmezí 12 - 30 Hz
- Vysoké - rozmezí 30 - 75 Hz

Dle článku [46] dosahují nejvyšší amplitudy frekvence kolem 10 Hz, dále v rozmezí 16-18 Hz. Ve vysokém pásmu jsou již amplitudy nižší. Při volbě frekvence je třeba se vyhnout pásmu, ve kterém má pacient alfa vlny, neboť by mohly být zaměněny s odpovědí na stimulaci.[11]

Barvy

Jak již bylo řečeno u stimulace klasických VEP, nemá barva stimulantu žádný znatelný vliv na záznam. To ale neplatí pro SSVEP, neboť barvy mají vliv na stimulaci pro různé frekvence. Červená barva má znatelnější vliv na odezvu kolem 11 Hz, ale v okolí se rychle zmenšuje. Oproti tomu modrá barva má na 13Hz slabší odezvu než barva červená. Žlutá barva má taktéž nižší amplitudu a menší závislost na zvolené frekvenci. [11]

3.3 AEP - Sluchové evokované potenciály

Sluchové evokované potenciály slouží k diagnostice periferního sluchového nervu pomocí zvuku, generovaného z vnějšího prostředí, který směřuje pomocí sluchové dráhy ze šneka do mozkové kůry. AEP jsou podtřídou kognitivních evokovaných potenciálů.[44]

3.4 SEP - somatosenzorické evokované potenciály

Somatosenzorické evokované potenciály získáme elektrickou stimulací senzitivních vláken periferního nervu. Zjišťujeme poškození nervových drah pro vedení citlivosti a polohocitu z periferie přes zadní provazce míchy do mozkového centra. Vyšetření se provádí z horních a dolních končetin. Na horní končetině stimulujeme střední nerv v zápěstí s registrací nad pažní pletení v oblasti klíčku, nad míchou (nad 5. krčním obratlem) a nad mozkovou kůrou. Z dolních končetin SSEP získáme stimulací holenního nervu za vnitřním kotníkem s registrací nad bederní míchou a nad mozkovou kůrou. Jde o vyšetření opakovaným stimulem s frekvencí 3-7Hz v celkovém počtu 300-1000 impulsů dvakrát po sobě pro každou končetinu. Registrační elektrody jsou povrchové, přilepené ke kůži. [10]

3.5 MEP - Motorické evokované potenciály

Při vyšetření motorických evokovaných potenciálů zjišťujeme poškození motorických drah v předních a postranních provazcích míchy. Používá se stimulace magnetickou cívkou, která se přikládá nad hlavu a nad páteř. Podráždění vede ke svalové odpovědi, kterou snímáme povrchovými elektrodami nad svaly horních a dolních končetin. [10]

3.6 ERP - kognitivní evokované potenciály

ERP jsou reakce mozku, které jsou vysílány na základě nějaké "události"- v případě AEP se jedná o zvuk, resp. o jednoduché opakované cvaknutí vysílané do jednoho ucha přes sluchátka. Druhé ucho, které není vyšetřeno, je blokováno maskovacím šumem. Výhodou AEP, ve srovnání s jinými diagnostickými metodami pro stanovení poškození sluchu, spočívá právě v její větší objektivitě. [25]

4 Zpracování EEG dat

Existuje velké množství metod, kterými zpracováváme signál a většina z těchto metod má široké uplatnění i mimo EEG signál. Postupy, které si zde popíšeme, se dají využít na různorodé signály, nebo dokonce pro operace s obrazovými daty. [3]

Na začátku kapitoly se budeme zabývat zpracováním obecného EEG signálu a postupně se metody budou vztahovat na EEG signál získaný z ustálených vizuálně evokovaných potenciálů (SSVEP), které jsem se v práci rozhodl zpracovat. EEG signál bude transformován do tvaru, který již pozorovateli prozradí, zda měření bylo úspěšné a zda se v datech objevují hledané frekvence.

4.1 Vzorkování signálu

Pokud zpracováváme EEG signál pomocí výpočetní techniky, tak se vždy setkáme s diskrétními daty, neboť máme k dispozici pouze omezenou paměť a samotné zpracování dat také není nekonečně rychlé. Z tohoto důvodu je potřeba spojitý signál vzorkovat. Vzorkování je tedy proces, ve kterém se snažíme transformovat spojitý signál na posloupnost diskrétních vzorků. [3]

Perioda vzorkování se označuje T_s a označuje čas mezi jednotlivými vzorky. Převrácenou hodnotou periody je vzorkovací frekvence.

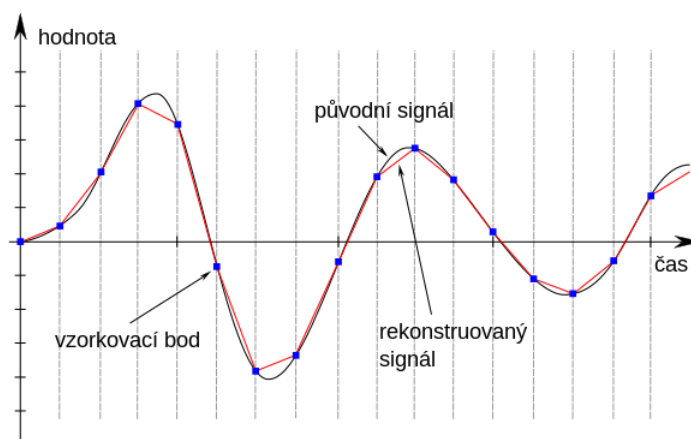
$$f_s = \frac{1}{T_s}$$

Vzorkovací frekvence se musí vhodně zvolit, protože příliš malá vzorkovací frekvence může vést ke ztrátě podstatné informace. Při vzorkování se ztrátě informace nevyhneme, ale snažíme se tyto škody minimalizovat. V případě špatně zvolené vzorkovací frekvence může docházet k aliasingu. [3]

4.1.1 Vzorkovací frekvence

Abychom správně zvolili vzorkovací frekvenci, je třeba se řídit vzorkovací Šanonovo větou, která říká, že pokud je signál $x(t)$ spojitý v čase a obsahuje složky s frekvencemi menšími než f_{max} , pak jej můžeme rekonstruovat z diskrétních vzorků $x(nT_s)$, pokud pro vzorkovací frekvence $f_s = \frac{1}{T_s}$ platí: $f_s \geq 2 * f_{max}$ [3].

Frekvence $\frac{f_{max}}{2}$ se nazývá Nyquistova frekvence (kritická vzorkovací frekvence).



Obrázek 4.1: Zobrazení správného vzorkování vstupního signálu [53]

4.2 Korekce baseline

Měříme-li amplitudu EEG vlny je obvykle ovlivněna průměrnou amplitudou před podnětem. Pokud tato hodnota není nulová, tak se nazývá hodnota baseline. Tato hodnota má významný vliv na výsledek všech metod zpracování EEG signálu. Kvůli tomu je nutné kompenzovat základ v každé epoše. Doporučuje se vypočítat průměrnou amplitudu v rozsahu 200 ms před začátkem stimulu a odečíst výslednou hodnotu od každé funkční hodnoty epochy. [37]

4.3 Volba kanálů

Jak již bylo řečeno, data jsou naměřena z několika elektrod a záleží na zvolené metodě vyhodnocování v jaké formě data potřebujeme. Pro účely ustálených vizuálních evokovaných potenciálů se využívají převážně elektrody $O1$ a $O2$, ale můžeme použít i elektrody Pz , $P3$ a $P4$, u kterých však hledané odezvy nejsou tolik patrné. [3]

4.4 Fourierova transformace

Díky Fourierově transformaci dokážeme převést signál do frekvenční oblasti. Tato transformace vychází z Fourierových řad, které umožňují přepsat libovolnou periodickou funkci pomocí součtu funkcí goniometrických (sinus a kosinus). Fourierovou řadou lze rozložit

i komplikovanější funkce, které by bylo jen velmi obtížné aproximovat. Fourierova transformace se dělí na spojitou a diskrétní, dle vstupních dat. V praxi se avšak setkáváme výhradně s diskrétním signálem. [3][6]

4.4.1 Diskrétní Fourierova transformace

Diskrétní Fourierova transformace (DFT) využívá Fourierovu analýzu. Jak z názvu vyplývá, očekává na vstupu diskrétní signál, což je nejčastěji vzorkovaný spojitý signál. Vstupní signál musí být konečný. Jestliže nebude tato podmínka splněna, lze pro výpočet použít pouze část vstupního signálu, který obsahuje N vzorků. Nejčastěji se v praxi využívá plovoucí okénko, které postupně načítá další data. [3][6]

Abychom vypočítali N -bodový DFT signál s N vzorky, můžeme použít tyto vzorce:

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}; k = 0, 1, \dots, N - 1$$

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x(n)[\cos(2\pi nk/N) - j \sin(2\pi nk/N)]$$

kde $x(n)$ je vstupní diskrétní signál o dané délce N , n označuje index vzorku signálu a k je index spektrální složky. Komplexní exponenciála $\exp(-j2\pi nk/N)$ tvoří bázi DFT. Výsledkem DFT je spektrum, které je složeno ze spektrálních koeficientů $X(k)$ a představuje frekvence, které obsahuje vstupní analyzovaný signál.

Zpětně transformovatelné pomocí vztahu:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_{DFT}[k]e^{j2\pi nk/N}$$

Víme, že $e^{j2\pi nk/N}$ je funkce periodická, takže je periodická i DFT. Díky tomu stačí počítat vzorky pouze přes jednu periodu. [6]

Tento vztah lze pomocí maticového zápisu přepsat do tvaru:

$$S = W \times s$$

kde S je sloupcová matice obsahující jednotlivé spektrální složky, s je vstupní signál a matice $W(n)$ obsahuje bázové funkce DFT.[6]

4.4.2 Symetrie diskrétní Fourierovy transformace

Diskrétní Fourierova transformace reálného signálu vykazuje komplexně sdruženou symetrii kolem počátku tj.:

$$X_{DFT}[-k] = X_{DFT}^*[k]$$

Díky periodicitě DFT lze napsat:

$$X_{DFT}[-k] = X_{DFT}[N - k]$$

Jestliže spočítáme spektrum ze signálu o N vzorcích, dostaneme spektrum, souměrné dle středu $N/2$. Z toho plyne, že nemusíme počítat DFT pro všechny vzorky, ale stačí pouze pro $N/2$. [6]



Obrázek 4.2: Symetrie DFT pro lichý a sudý počet vzorků [55].

4.4.3 Frekvence v diskrétní Fourierově transformaci

Když vypočítáme DFT, získáme posloupnost N prvků. Tyto prvky jsou souměrné podle středu. Četnost frekvence je v signálu udávána právě těmito čísly. Avšak musíme zjistit, která čísla odpovídají příslušné frekvenci. [6]

Jestliže známe vzorkovací frekvenci f_s signálu, je poměrně jednoduché získat frekvenci konkrétního členu posloupnosti. Pokud máme N vzorků signálu a vzorkovací frekvenci f_s bude daný prvek posloupnosti odpovídat frekvenci: [6]

$$f[k] = k \times \frac{f_s}{N}$$

Maximální frekvence pro DFT posloupnost, je polovina vzorkovací frekvence, která leží na pozici $X_{DFT}[N/2]$ pro sudý počet N . Pro lichý počet je to o prvek blíže. Může se stát, že signál bude obsahovat frekvence, které spektrum neobsahuje, jelikož se jedná o diskrétní frekvence. Může to být způsobeno tím, že frekvence ve spektru jsou celočíselné, ale signál obsahuje frekvenci, která celočíselná není. [6]

4.4.4 Rychlá Fourierova transformace

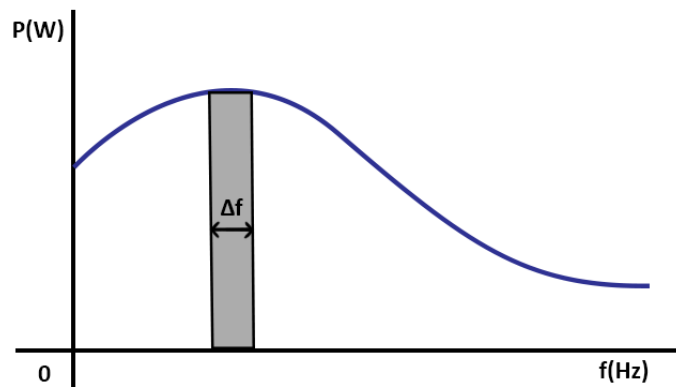
Diskrétní Fourierova transformace naráží na problém se složitostí výpočtu, který je $O(N^2)$ operací. Rychlá Fourierova transformace (FFT) optimalizuje výpočet na složitost $O(N \log N)$. S tímto algoritmem přišel poprvé James W. Cooley a John W. Tukey v roce 1965 a je to jeden z nejpoužívanějších FFT. Princip tohoto algoritmu spočívá v rozdělení signálu na lichou a sudou část, a z každé části se provádí transformace zvlášť. DFT využívající

N -bodovou posloupnost můžeme zapsat jako součet dvou $N/2$ bodových transformací, které se skládají ze sudé a liché části. [6]

Avšak vstupem do FFT algoritmu musí být vždy takový vzorek dat, který odpovídá mocnině dvou, na rozdíl od DFT, který lze vypočítat z libovolného množství. Pokud nemáme k dispozici takovou sadu dat, aby její mocnina odpovídala mocnině dvou, lze data doplnit nulami na nejbližší další mocninu dvou. [6]

4.5 Výkonová spektrální hustota (PSD)

Výkonová spektrální hustota (Power spectral density) vyjadřuje výkon obsažený v úzkém intervalu Δf spojitého spektra a je vyjádřena ve W/Hz. [42]



Obrázek 4.3: Měření výkonové spektrální hustoty

- Spojitý signál - ukazuje rozložení výkonu podél kmitočtové osy
- Diskrétní signál - z fyzikálního hlediska diskrétní signál žádný výkon nemá, pojem výkonová spektrální hustota je převzat od spojitých signálů

Pokud diskrétní signál vznikl dostatečně hustým vzorkováním signálu ve spojitém čase, může být výkonová spektrální hustota diskrétního signálu úměrná výkonu spektrální hustoty spojitého signálu. [42]

4.5.1 Parametrické metody výpočtu

Abychom mohli odhadnout výkonové spektrum pomocí parametrických metod potřebujeme vhodný vzorek signálu. Parametry takového modelu, pak popisují signál (včetně jeho spektra) úsporným způsobem a představují tak vedle analytického popisu také redukci dat. [42]

4.5.2 Neparametrické metody výpočtu

Tyto metody si popíšeme důkladněji, neboť jsem je využil při klasifikaci dat.

Metody se nazývají neparametrické, protože předem hledány vlastnosti (parametry) zpracování dat. Jsou jednoduché na výpočet a používají k výpočtu rychlé Fourierovy transformace (FFT).[42]

Mezi neparametrické metody patří:

- Bartlettova metoda průměrování periodogramů.[42]
- Welchova metoda průměrování modifikovaných periodogramů.[42]

1. Bartlettova metoda zahrnuje následující kroky. [42]

- Datová posloupnost o délce N_y je rozdělena na I nepřekrývajících se segmentů, každý segment má délku N , viz vzorec:

$$y_i(n) = y(n + iN)$$

- Pro každý segment se vypočte periodogram, viz. vzorec:

$$PSD_{yy}^i(k) = \left| \frac{1}{n} \sum_{n=0}^{N-1} y_i(n) e^{-j \frac{2\pi}{N} kn} \right|^2 = \frac{1}{N} |DFT\{y_i(n)\}|^2$$

Pro $k=0,1,\dots,N-1$; $i=1,2,\dots,I-2$.

- Periodogramy pro I segmentů jsou průměrovány podle vzorce.

$$PSD_{yy}^I(k) = \frac{1}{I} \sum_{i=0}^{I-1} PSD_{yy}^i(k)$$

2. Welchova metoda - oproti Bartlettově metodě zde Welch udělal dvě změny. [42]

- Povolil překrývání segmentů, takže data jsou interpretována dle vzorce:

$$y_i(n) = y(n + iN(1 - v))$$

- Druhou změnou bylo násobení datového segmentu časovým oknem před výpočtem periodogramu. Výsledkem je modifikovaný periodogram podle vzorce:

$$PSD_i(k) = \frac{1}{NU} |y_i(n)w(n)e^{-j \frac{2\pi}{N} kn}|^2$$

Pro $i=0,1,\dots,I-1$, kde U je normalizační koeficient.

- Normalizační koeficient, viz vzorec:

$$U = \frac{1}{N} \sum_{n=0}^{N-1} w^2(n)$$

- Welchův odhad výkonové spektrální hustoty je pak průměr dílčích modifikovaných periodogramů a je dán vztahem:

$$PSD_i^I(k) = \frac{1}{I} \sum_{i=0}^{I-1} PSD_i(k)$$

5 Klasifikace dat

Rozpoznávání a klasifikace dat je široký obor, který se využívá v mnoha odvětvích. Zjednodušeně řečeno se snažíme zařadit objekty do předem určených skupin. Tyto skupiny mají dvě základní rozdělení. V první skupině jsou předem definované (klasifikace) třídy a ve druhé skupině se třídy vytvářejí za běhu (rozpoznávání). K tomu, aby bylo možné objekty klasifikovat do skupin, potřebujeme určitou charakteristiku, dle které je možné objekty do daných tříd rozdělit. Charakteristika, kterou zvolíme pro klasifikaci nebo rozpoznávání, není vždy jednoznačná a vždy bude záležet na povaze problému. [12] Prvotní rozdělení metod, které využijeme pro klasifikaci nebo rozpoznávání je na příznakové a strukturální. [12]

- Příznakové rozpoznávání - využívá příznakový vektor, který se skládá z charakteristik objektu a na základě těchto charakteristik se rozdělují objekty do tříd.
- Strukturální metody - prvotně vytvoří primitiva pomocí kterých popisují objekty.

5.1 Příznakové metody rozpoznávání

Příznakové neboli statistické metody rozpoznávání vycházejí z popisu objektu, který je tvořen číselnými hodnotami. Tyto čísla jsou označována jako příznaky a udávají míru významnosti, nebo vlastnosti. Objekt je následně reprezentován jako soubor těchto hodnot, které jsou uspořádané v tzv. příznakovém vektoru. Úspěšnost je silně závislá na zvoleném počtu příznaků, které jsou nositeli informace o daném objektu. Bylo by mylné se domnívat, že zvyšováním počtu příznaků, se bude úměrně zvyšovat úspěšnost klasifikace. Nejen že by stouply nároky na výpočetní techniku a samotná klasifikace by trvala delší dobu, ale navíc by mohlo dojít k opačnému efektu, tj. snížení úspěšnosti klasifikace. Z tohoto důvodu by volba příznaků měla záviset na jejich přínosu pro konkrétní úlohu. Z toho vyplývá, že neexistuje obecný postup pro volbu příznaků, ale je možné využít metody, které nám umožňují vybrat příznaky z hlediska jejich informačního přínosu pro daný typ úlohy. Příznakové metody pro klasifikaci využívají tři metody, které si postupně rozebereme. [12]

5.1.1 Diskriminační funkce

Diskriminační funkce se využívá pro odvození rozhodovacího pravidla, které určuje příslušný vektor k dané třídě.

Tato klasifikace je založena na následujícím principu.

Vstupem úlohy je příznakový prostor, který obsahuje všechny vektory příznaků a následně množinu vzájemně disjunktních tříd. Třídy lze od sebe oddělit rozdělujícími nadplochami, které se definují pomocí soustavy tzv. diskriminačních funkcí.

Diskriminační funkce je vlastně libovolná skalární funkce, do které vstupuje pomocí argumentu vektor. Z toho vyplývá, že prvním krokem bude zdefinování diskriminační funkce pro každou třídu. Poté se pro každé dvě sousední třídy sestaví pomocí jejich diskriminační funkce rovnice jejich nadplochy. V momentě, kdy máme definované všechny nadplochy, je definováno i rozhodovací pravidlo a úloha klasifikace je vyřešena. Samotná klasifikace poté probíhá tak, že vybraný vektor je zařazen do třídy, jejíž hodnota diskriminační funkce je

pro daný vektor maximální.

Aby tento princip mohl fungovat, je založen na realizaci lineárního klasifikátoru za předpokladu, že třídy jsou lineárně oddělitelné. Jestliže tento předpoklad není splněn, tak jsou rozdělující nadplochy nelineární a je třeba použít nelineární klasifikátor.

5.1.2 Kritérium minimální vzdálenosti

Abychom mohli použít tuto metodu klasifikace, musí být splněny následující předpoklady. Každá třída je charakterizována vektorovým vektorem příznaků - *etalon třídy*. Rozhodovací pravidlo na jehož základě jsou objekty klasifikovány do tříd, lze definovat následujícím způsobem. Zkoumaný vektor zařadíme do dané třídy, jestliže její etalon má od tohoto vektoru nejmenší vzdálenost.[18] [12]

Jestliže v_1, v_2, \dots, v_R jsou etalony jednotlivých tříd r a x je zkoumaný vektor. Tak pravidlo, dle kterého zařadíme vektor do třídy jejíž etalon má od tohoto vektoru nejmenší vzdálenost, lze definovat dvěma způsoby v závislosti na stanovené normě vektoru příznaků. [18] [12]

1. $\|v_r - x\| = \min \|v_s - x\|$ pro $s = 1, 2, \dots, R$, kde R je počet tříd
2. $\|v_r - x\| = \min \|v_s - x\|^2$ pro $s = 1, 2, \dots, R$

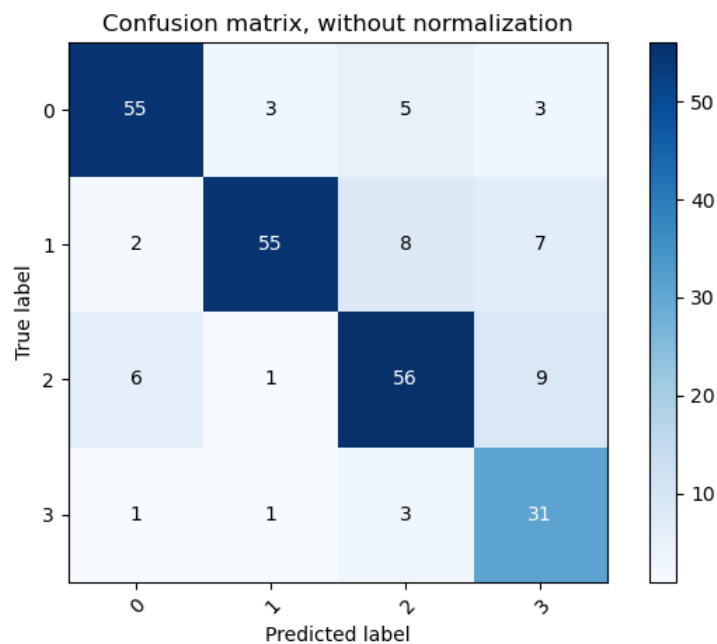
5.1.3 Kritérium minimální chyby

Jestliže zkoumaný objekt nelze jednoznačně zařadit do předem definovaných skupin, tak lze využít metodu kritéria minimální chyby. Takový případ může nastat, jestliže třídy nejsou jednoznačně oddělitelné a při jakémkoliv rozhodnutí se tedy klasifikátor dopustí určité chyby. Cílem je tedy nastavit klasifikátor tak, aby tyto chyby minimalizoval.[18] [12]

Jak již bylo řečeno, třídy nejsou jednoznačně oddělitelné a nelze o nich tedy jednoznačně rozhodnout. Kvůli tomuto důvodu je třída považovaná za náhodnou proměnou s určitými možnými hodnotami a s předem danými pravděpodobnostmi, které vyjadřují pravděpodobnost výskytu vektorů příznaků z třídy. [18] [12]

5.2 Matice záměn (Confusion matrix)

Matice záměn se často používá k popisu výkonu klasifikačního modelu na sadě testovacích dat, pro které jsou známy skutečné hodnoty. Umožňuje vizualizaci výkonu algoritmu a snadnou identifikaci záměny mezi třídami, např. Jedna třída je běžně označena jako druhá. Většina měření výkonu se počítá z matice záměn. Matice záměn nám vyobrazuje, jakým způsobem byly třídy předpověděny při klasifikaci. Dává nám to náhled nejen na chyby, kterých se algoritmus dopustil, ale především na typy chyby, která se vyskytla. Na obrázku (Obrázek 5.1) vidíme jednotlivé klasifikační třídy a jejich zařazení pomocí klasifikátoru. Na ose X jsou klasifikované klasifikační třídy a na ose Y jsou správně určené



Obrázek 5.1: Matice záměn

klasifikační třídy.

Příklad:

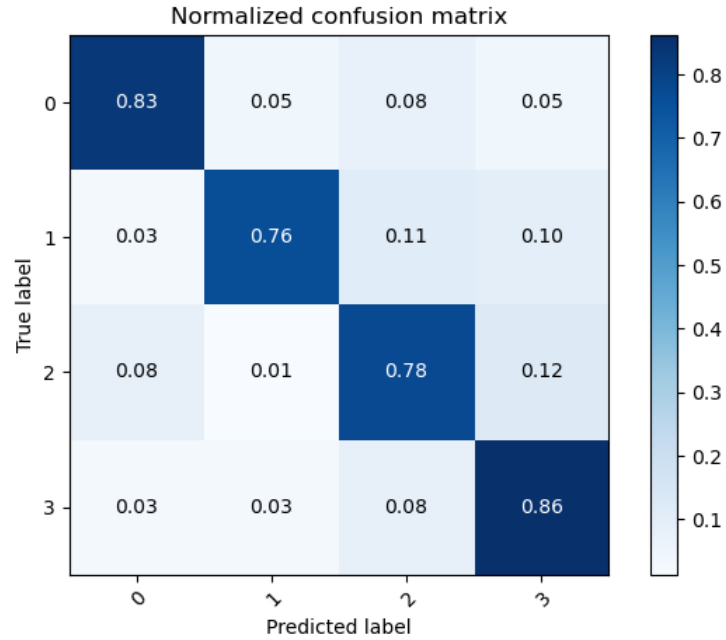
Vezmeme-li první řádek matice záměn, tak získáme hodnoty (3, 7, 5, 5), tyto hodnoty se podle osy Y vztahují k správně určené klasifikační třídě '2' (obrázek 5.1). Osa X nám naopak určuje, ke které klasifikační třídě (2-5) byla daná klasifikační třída '2' pomocí klasifikátoru přiřazena (obrázek 5.1).

To znamená, že klasifikační třída '2' byla zařazena tímto způsobem:

- 3 krát jako klasifikační třída 2 - **správně**
- 7 krát jako klasifikační třída 3 - špatně
- 5 krát jako klasifikační třída 4 - špatně
- 5 krát jako klasifikační třída 5 - špatně

Z toho vidíme, že klasifikátor správně určil klasifikační třídu '2' pouze 3 krát z 20 epoch. Průsečík osy X a Y nám udává počet, kolikrát byla daná klasifikační třída správně klasifikována

Tímto se dostáváme k normalizované matici záměn (Obrázek 5.2). Ta nám udává, že úspěšnost zařazení klasifikační třídy '2' byla 15%.



Obrázek 5.2: Normalizovaná matice záměn

Na obrázku (Obrázek 5.1) také vidíme barevné vyobrazení, které nám udává hustotu zařazení klasifikačních tříd. Čím je barva tmavší, tím více epoch bylo klasifikováno do dané kategorie. Z toho vyplývá, že chceme, aby nejtmaší barva se soustředila na diagonále, kde jsou správně zařazené klasifikační třídy.

6 BCI systémy

Rozhraní mozek - počítač je využíváno pro komunikaci mezi lidským mozkem a výpočetní technikou. BCI (Brain-computer interface) umožňuje díky nasnímaným signálům z mozku provádět vnější aktivitu. Tyto systémy jsou využívány zejména ve zdravotnictví, kde pomáhají postiženým pacientům. Přináší značnou pomoc pro pacienty, kteří trpí motorickou či komunikační bariérou. Hlavní množinu tvoří pacienti, kteří utrpěli mozkovou příhodou. Pacient, který je připojen k BCI systému, je schopen pomocí své mozkové aktivity ovládat nejen počítač, ale i celou řadu dalších externí zařízení, kde počítač hraje roli prostředníka mezi mozkem a technikou.[7]

6.1 Rozdělení BCI systémů

BCI systémy se dají teoreticky rozdělit do tří kategorií podle typu implantace do živého organismu.[7]

- **Invazivní systémy** jsou implantované přímo do šedé kůry mozkové a neobejdou se bez neurochirurgického zákroku. Tato metoda dosahuje nejlepší kvality výsledků, ale samozřejmě hlavní nevýhodou je náchylnost k rozšíření poškození na mozkové tkáni, což se může projevit zeslabením nebo úplnou ztrátou signálu. [7]
- **Částečně invazivní systémy** jsou charakteristické tím, že se implantát nachází uvnitř lebky a zbytek je vyveden mimo mozek, avšak produkují o něco horší kvalitu signálu než invazivní systémy. Jejich výhodou je ale fakt, že zde nedochází k poškození mozkové tkáně. [7]
- **Neinvazivní systémy** nejsou zaváděny přímo do mozku, ale využívá se zde speciálních EEG čepic, bohužel zde dochází k deformaci signálu, která je způsobena přechodem přes kostní tkáň. [7]

Dále také můžeme rozdělit BCI systémy dle paradigmat, kde každé paradigma se specificky využívá pro určitou problematiku. Rozděluje se to podle závislosti na vnějších podnětech a záměru využití. Například BCI systémy založených na VEP jsou příkladem vnějšího podnětu, neboť pro jeho použití je nutná stimulace subjektu. Naopak BCI systém založený na alfa vlně nemusí nutně potřebovat vnější stimulaci. [45]

Rozdělení BCI paradigmat:

- Vlna P300
- VEP signály
- Představa pohybu
- Alfa vlny (alpha neurofeedback)

6.2 Princip BCI

Snímání dat se dá zajistit v dnešní době poměrně širokou škálou metod. Jako jsou například EEG (elektroencefalografie), MEG (magnetoencefalografie), PET (pozitronová

emisní tomografie), atd. V praxi se ale nejčastěji setkáme s technologií EEG, která umožňuje měřit napěťové potenciály vycházející z mozkových neuronů. Jednotlivé elektrody bývají nejčastěji umístěny v EEG čepici, která je umístěna subjektu na skalp. [7]

Artefakty - zde se nám v signálu začínají objevovat artefakty, které jsme si popsali v sekci (2.6). Naším cílem je tyto artefakty potlačit a tím získat co nejčistší signál. [7]

Parametrizace - v tuto chvíli je snímaný EEG signál analyzován a jsou z něho odvozeny parametry, které se využijí na externím zařízení. Parametrizace je nejčastěji postavená na spektrální analýze a autoregresním modelování. [7]

Klasifikátor je zodpovědný za rozpoznávání vstupních dat dle získaných parametrů a generuje příkazy pro navazující systémy. [7]

Zpětná vazba umožňuje uživateli sledovat reakce na jeho mozkovou aktivitu a tím zlepšit spolupráci se systémem. [7]

6.3 Aplikace BCI systému

Lidi obecně fascinuje myšlenka, že by dokázali pomocí své mysli, ovládat různá zařízení. Primárně se BCI soustředí na asistenční technologie pro postižené, ale samozřejmě se BCI využívá v rozličných oborech, například i v herním průmyslu. V roce 2010 bylo aplikováno přes 220.000 kochleárních implantátů (pomoc sluchově postiženým lidem) po celém světě. Pomocí těchto systémů lze monitorovat a dokonce i detekovat některá nádorová onemocnění, projevy epilepsie, spánkové poruchy, migrény a další. [7]

6.4 BCI založené na VEP

Rozšířeným druhem evokovaných potenciálů (Kapitola 3) využívajících se v BCI systémech jsou tzv. VEP (Kapitola 3.1). Vizually evokované potenciály (VEP) jsou založené na stimulaci zrakového pole a reflexi informace, pocházející z této stimulace do mozku. Pomocí BCI systémů založených na VEP dokážeme identifikovat cíl, kterým byl subjekt stimulován. Dle použité modulace dělíme VEP do tří kategorií (Kapitola 6.4).

BCI systémy založené na signálech VEP jsou rozděleny do následujících kategorií.

- **t-VEP** - časově modulovaný VEP (Kapitola 3.1). V tomto BCI systému sledujeme předem definovanou sekvenci záblesků z různých podnětů. Tyto sekvence jsou vzájemně nezávislé. Nezávislosti docílíme tak, že se navzájem nepřekrývají tj. neprobíhají současně. Abychom dokázali v datech číst, je potřeba do EEG záznamu zavést synchronizační značky pro určení začátku stimulace. Cílem tohoto systému je zprůměrování získaných dat a následné určení maximální latence a amplitudy. [19] [20]
- **f-VEP** - Frekvenčně modulovaný VEP (Kapitola 3.1). Tyto BCI systémy se často označují jako SSVEP-BCI, neboť stimulace je způsobena blikajícími objekty, které se překrývají. Tímto vzniká periodická posloupnost ustálených vizuálně evokovaných potenciálů (SSVEP). Obvykle se rozmezí použité frekvence pohybuje mezi 5

Hz až 75 Hz. Základ frekvenčně modulovaný VEP je založen na tom, že každý objekt stimulačního scénáře má jinou frekvenci blikání. Zároveň by tyto frekvence neměly být ani násobkem již použitých frekvencí.

Jako odpověď na tuto stimulaci je periodický signál. Tento signál se vyznačuje stejnou frekvencí, popřípadě harmonickými násobky, která byla použita ve stimulačním vzoru. Hlavní výhodou *f*-VEP protokolu je rychlé učení subjektu a jednoduchá konfigurace. [19] [20]

- **c-VEP** - Pseudonáhodně kódovaně modulovaný VEP (Kapitola 3.1). Tento BCI systém generuje pseudonáhodnou sekvenci, která definuje dobu trvání stavu ON a OFF každého stimulu. Tento režim poskytuje nejvyšší komunikační rychlost. Stejně tak jako u *t*-VEP je potřeba do signálu zavést synchronizační značky. [19] [20]

Při využití stimulačních protokolů *t*-VEP a *c*-VEP vznikají klasické evokované potenciály VEP, ale při využití protokolu *f*-VEP vznikají námi požadované ustálené evokované potenciály SSVEP.[19]

6.4.1 Aplikace BCI systémů na signálech SSVEP

V experimentech založených na SSVEP je uživatel požádán, aby soustředil svůj zrak na požadovaný stimul. Díky následné extrakci dat a jejich klasifikaci, dokážeme určit, na který stimul se subjekt dívá. Tohle je základní předpoklad, neboť subjekty, pro které jsou tyto asistenční systémy navrhovány, většinou nemají možnost jiné komunikace.

Existuje velké množství asistenčních systémů, které jsou založeny na SSVEP-BCI, ale většina z nich funguje na stejném principu. Například pokud budeme chtít, aby subjekt pomocí tohoto systému ovládal svůj kolečkový vozík, tak s největší pravděpodobností mu předložíme obrazovku, kde budou 4 šipky znázorňující směr jízdy a směr otočení vozíku. Každá tato šipka má nastavenou jinou frekvenci blikání. Podle toho na kterou šipku bude subjekt upínat svou pozornost, bude vozík reagovat.[20]

Podobnou aplikaci můžeme použít například pro ovládání myši na obrazovce.

6.4.2 Zhodnocení BCI systémů na signálech SSVEP

SSVEP se ukazuje jako nejrozšířenější paradigma BCI používané pro různé aplikace. Lze jej použít pro zdravé i postižené pacienty vzhledem k tomu, že SSVEP-BCI vyžadují minimální školení subjektů. Důvodem je, že subjekt nemusí regulovat svou vlastní mozkovou činnost, aby poskytoval vstupní data. Subjekty musí zaměřit svou pozornost pouze na blikající objekty, které pak natrénovaný klasifikátor převede na příkazy, které mohou ovládat další zařízení. [20]

6.5 BCI založené na vlně P300

Paradigma P300 (P3) je založeno na odezvě, která je způsobena například vizuální stimulací. Tato odezva se nachází mezi 250 - 500 ms, avšak vrcholí kolem 300 ms poté, co mozek zpracuje informaci o náhlé nebo překvapivé události. Vlna P300 je součástí kognitivního evokovaného potenciálu (Kapitola 3.6). Experiment využívající vlnu P300 může vypadat následovně. Subjektu je zobrazena tabulka znaků a postupně jsou zvýrazňované sloupce a řádky dané tabulky. Subjekt se soustředí na určitý znak a při rozsvícení daného znaku

to v subjektu vyvolá pocit překvapení. Díky tomu vzniká v mozku vlna P300, která je pomocí BCI systému počítačem zaznamenána. Hledaný znak je poté nalezen v průsečíku řádku a sloupce. Tato metoda dosahuje velice dobrých výsledků, ale dokáže rozlišit pouze dva stavy a to, jestli se jedná nebo nejedná o vlnu P300. [21]

7 Umělé neuronové sítě

Jak už název napovídá, umělé neuronové sítě jsou inspirované biologickými neuronovými sítěmi. Ve své podstatě se snaží umělé neuronové sítě modelovat strukturu a činnosti jejich biologických vzorů. Informační systémy všech živých organismů, jsou založeny na biologických neuronových sítích. Základ teorie biologických neuronových sítích vychází z neurofyziologických poznatků. Základním stavebním blokem v neuronových sítích je nervová buňka - neuron.[8]

Umělé neuronové sítě mají své opodstatnění například v matematických úlohách, kde při řešení daného problému není možné popsat všechny vztahy a souvislosti, které ovlivňují sledovaný proces. Dále se hojně využívají v případech, kdy se nám sice podaří matematický model sestavit, ale je tak náročný, že je jeho algoritmizace téměř nemožná. Existují oblasti, kde technologie neuronových sítí přináší bezkonkurenční výhody a to například při zpracování neúplných, nepřesných, neurčitých, nebo dokonce rozporuplných informací a to hlavně v úlohách optimalizačního charakteru.[8]

Mezi hlavní výhody neuronových sítí se řadí schopnost učit sebe sama. To se dá interpretovat jako dovednost získávání znalostí učením pomocí množiny předkládaných vzorů a to bez nutnosti vytvořit algoritmus řešení.[8]

Díky tomu mohou najít umělé neuronové sítě široké uplatnění v našem životě:

- Při lékařské diagnostice signálů EEG, EKG a jejich analýze.
- Dále se uplatňují při rozpoznávání a případně i při rekonstrukci obrazců (např. otisky prstů, rozpoznání obličeje, ručně psaný text apod.).
- V řízení a modelování nelineárních procesů popřípadě v predikci časových řad a v dalším rozhodování. V praxi se s nimi můžeme setkat v prognóze vývoje počasí, ekonomii, dopravě, energetice atd.

7.1 Umělý neuron

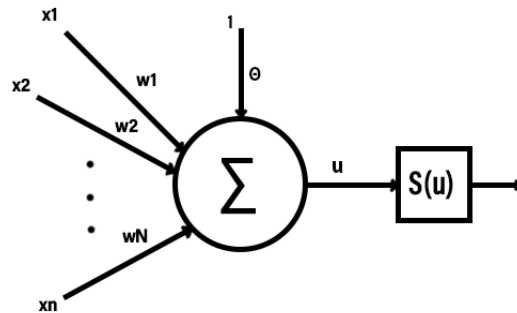
Umělý neuron je základem matematického modelu, který získáme přeformulováním zjednodušené funkce neurofyziologického neuronu do matematické řeči.

Umělý neuron (dále jen neuron) má n obecně reálných vstupů X_1, \dots, X_N , které modelují dendrity (dendrit - krátký výběžek biologického neuronu, ve kterém jsou uloženy informace). Vstupy jsou ohodnoceny obecně reálnými synaptickými váhami w_1, \dots, w_N , které určují jejich propustnost (synapse - vzájemné spojení neuronů). Ve shodě s neurofyziologickou motivací mohou být synaptické váhy záporné, čímž se vyjadřuje zpomalení daných spojů.[13]

Vážená suma vstupních hodnot představuje vnitřní potenciál neuronu:

$$u = \sum_{i=1}^n w_i X_i$$

Hodnota vnitřního potenciálu u po dosažení tzv. prahové hodnoty θ indikuje stav neuronu y , který modeluje elektrický impuls axonu. Nelineární nárůst výstupní hodnoty $y = S(u)$ při dosažení prahové hodnoty potenciálu θ je dán tzv. aktivační (přenosovou) funkcí S . [13]



Obrázek 7.1: Schéma umělého neuronu

Za pomoci formální úpravy docílíme toho, že funkce S bude mít nulový práh a vlastní práh neuronu budeme chápat jako váhu, tzv. bias (zkreslení výsledků) dalšího formálního vstupu s konstantní jednotkovou hodnotou. [13]

7.2 Aktivační funkce neuronu

Výstup z neuronu tedy můžeme přepsat do výsledného tvaru:

$$y = S\left(\sum_{i=1}^n w_i X_i + \theta\right)$$

Aktivační (přenosová) funkce neuronu je funkce, která mění vstupního signálu na signál výstupní v intervalech 0 až 1 a -1 až +1. Funkce mohou být skokové nebo spojité a musí být monotónní tzn. že změna vstupu na výstup musí být jednoznačná. Aktivační funkce je tedy velmi důležitým prvkem neuronu. Je tedy důležité, abychom pro správný chod neuronu a neuronové sítě zvolili vhodnou aktivační funkci. Aktivační funkce udává, jaká bude reakce výstupu na vstupní signál. Druhy aktivačních funkcí jsou různé a u některých obecně platí, že jejich hodnota má být v intervalu -1 až +1 a že mají být spojité nebo s nespojitostí prvního druhu (binární funkce 0-1). Správná volba aktivační funkce závisí na problému, který budeme řešit. Například pokud budeme klasifikovat objekt, kde nám stačí znát pouze jestli vyhověl a nebo nevyhověl, pak nám bude stačit binární funkce. Pokud bychom ale použili funkci spojitou, tak se musíme rozhodnout, jaká její hodnota znamená vyhověl a jaké nevyhověl (0.4, 0.8,...). [33]

Nejpoužívanější druhy aktivačních funkcí jsou:

- skoková
- lineární
- logistická

Skoková aktivační funkce nabývá pouze 2 diskrétních hodnot, a to podle toho, zda je unipolární či bipolární. Jestliže je funkce unipolární, pak nabývá hodnot 0 a 1, zatímco bipolární -1 až +1. Podobné je to i s logistickými aktivačními funkcemi. Avšak hodnoty těchto funkcí nabývají spojité hodnot. Obor hodnot z intervalu $(0, 1)$ má logistická sigmoida, hyperbolický tangens pak interval $(-1, +1)$. Lineární aktivační funkce vykazuje také spojité výstup. Její transformace je zřejmá, tedy že tutéž hodnotu na vstupu dává i na výstupu.

[33]

7.2.1 Aktivační funkce ReLU

Jedná se o velmi jednoduchou funkci, ale současně populární v posledních letech.

Matematický předpis:

$$f(x) = \begin{cases} x, & \text{pro } x \geq 0. \\ 0, & \text{pro } x < 0. \end{cases}$$

Avšak tato aktivační funkce trpí na problém umrtvení neuronu, kdy může dojít k tomu, že neuron v budoucnu již nikdy nebude aktivován a gradient proudící skrze neuron bude vždy nula (nedojde tedy k nápravě). Tento problém je patrný zejména při vyšším učícím koeficientu a snaží se jej řešit například aktivační funkce Swish, viz. dále. [17]

7.2.2 Aktivační funkce Swish

Aplikační funkce Swish je matematicky definovaná jako:

$$Swish(x) := x * sigmoid(\beta x) = \frac{x}{1+e^{-\beta x}}$$

Kde β je buď konstanta nebo parametr závislý na použitém modelu. Pro $\beta = 1$, se funkce stává ekvivalentem funkce SiL (Sigmoid-weighted Linear Unit) používané ve zpětno-vazebném učení. Pokud je $\beta = 0$ tak se z funkce stává lineární funkce $f(x) = \frac{x}{2}$. Jestliže β jde do nekonečna, tak se z aplikační funkce Swish stává aplikační funkce ReLU. Lze jí tedy považovat za vyhlazovací funkci, která nelineárně interpoluje mezi lineární a ReLU funkcí. [30]

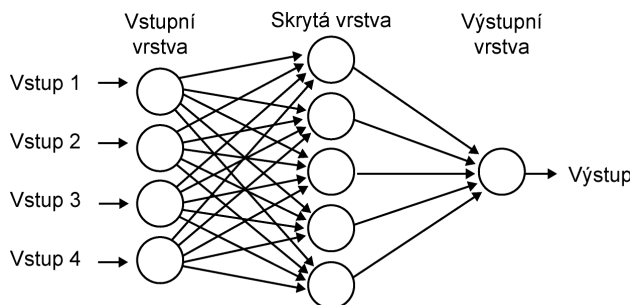
7.2.3 Aktivační funkce Softmax

Aktivační funkce Softmax se velmi často využívá u klasifikačních úloh, kdy je žádoucí, aby výstupy náležely do intervalu $(0, 1)$ a udávaly pravděpodobnost toho, že vstup patří do třídy dané výstupním neuronem. Právě takovou úpravu provádí aktivační funkce Softmax. Nejčastěji se využívá ve výstupní vrstvě neuronové sítě provádějící klasifikaci. Je dána následujícím matematickým předpisem (k výpočtu je potřeba hodnot všech K výstupních neuronů). [29]

$$f(x_i) = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}}$$

7.3 Struktura neuronové sítě

Abychom docílili větší výpočetní síly, jsou neurony uspořádány do sítí neuronů. Jsou sdružovány do struktur a tím vytvářejí specificky provázanou síť. Tato síť se většinou skládá z oblastí s podobnou funkcionalitou, tzv. vrstev. Neuronová síť je složena z jednotlivých neuronů, které jsou vzájemně propojeny a to tak, že výstup jednoho neuronu je vstupem obecně libovolnému počtu neuronů v další vrstvě. Finální topologii sítě určuje počet neuronů v jednotlivých vrstvách a princip jejich propojení. Obvykle pro topologii platí pravidlo, že každý neuron bývá spojen s každým neuronem vyšší vrstvy. [33]



Obrázek 7.2: Schéma neuronových sítí[56]

7.4 Dopředné neuronové sítě

Dopředné neuronové sítě jsou ve své podstatě vlastně orientované grafy, kde jednotlivé uzly jsou spojeny ohodnocenými orientovanými hranami. Ohodnocení hrany (váha spoje) je parametr zpracování signálu. Váhy mohou být ohodnoceny jak kladně, tak záporně, takže se neurony mohou navzájem povzbuzovat nebo potlačovat. Váha spoje vypovídá o schopnosti a intenzitě, kterou je spoj schopen přenášet informace. Signál se v dopředných neuronových sítích šíří směrem od vstupní vrstvy k výstupní. [13]

7.4.1 Perceptronová síť

Tento model neuronové sítě je jedním z prvních úspěšných modelů.

Jedná se o jednovrstvou síť, která má na začátku pevně specifikovanou organizační dynamiku $n - m$. To znamená, že síť má n vstupních neuronů a každý z těchto neuronů je vstupem každého z m výstupních neuronů. [13]

Způsob výpočtu funkce sítě je určen aktivní dynamikou perceptronové sítě. V tomto případě se reálné stavy neuronů ve vstupní vrstvě nastaví na vstup sítě a výstupní neurony spočítají svůj binární stav, které pak určuje výstup sítě. To znamená, že každý neuron nejprve musí spočítat svůj vnitřní potenciál jako příslušnou afinní kombinaci vstupů. [13]

$$y_j = \sum_{i=1}^n w_{ji} X_i \text{ kde } j = 1, \dots, m$$

Kde koeficienty $w = (w_{10}, \dots, w_{1n}, \dots, w_{m0}, \dots, w_{mn})$ tvoří konfiguraci sítě. Index i je číslo vstupního neuronu, n je celkový počet vstupních neuronů, j je číslo výstupního neuronu a m je celkový počet výstupních neuronů, takže váha w_{ji} znázorňuje, že tato váha je na

spojí z i -tého vstupního neuronu do j -tého výstupního neuronu. Stav perceptronu se určí z jeho vnitřního potenciálu za použití aktivační funkce, která má tvar ostré nelinearity (obrázky aktivačních funkcí).[13]

7.5 Vícevrstvá neuronová síť

Zde se rozlišují tři typy vrstev - vstupní, skrytá a výstupní. Vrstvy jsou většinou označeny číselně, kde nula odpovídá vstupní vrstvě, pak následuje libovolný počet skrytých vrstev a poslední vrstvou je vrstva výstupní. Jednotlivé neurony v sousedících vrstvách jsou mezi sebou propojeny tak, že výstup neuronu dané vrstvy je vstupem všem neuronům následující vrstvy (nelze přeskakovat vrstvy). Pokud vazba mezi neurony chybí, tak ji chápeme, jako vazbu s nulovou vahou. Díky tomu, že jsou vrstvy plně propojené, tak můžeme topologii sítě popsat pouze počty neuronů v jednotlivých vrstvách oddělených pomlčkou směrem od vstupní k výstupní vrstvě. Do počtu vrstev sítě není zahrnuta vstupní vrstva.[14]

Vstupní vrstva slouží ke vstupu signálu. Tento signál se zde pouze distribuuje na všechny neurony první skryté, anebo před distribucí na skryté vrstvy ještě dochází k normalizaci vstupního signálu.

Neurony ve skrytých vrstvách transformují vstupy z předchozí vrstvy do následující, jejich hlavním úkolem je zvýšení aproximačních vlastností neuronové sítě jako celku. Vstupní vrstva předává do okolí vstupní signály, které jsou odezvou neuronové sítě na převedené vstupní signály.[14]

Množství neuronů ve vstupní vrstvě vychází z typu řešeného problému. Počet skrytých vrstev a neuronů v jednotlivých skrytých vrstvách nelze jednoznačně určit, jejich počet je závislý na složitosti řešeného problému. Nejčastěji se počet určuje experimentálně. [14]

7.6 Algoritmus Back-propagation

Algoritmus Back-propagation, neboli algoritmus zpětného šíření (propagace) chyby je v poslední době hojně používaným algoritmem co se týká učení neuronových sítí. Tento algoritmu minimalizuje čtverce chybové funkce.

7.6.1 Princip algoritmu Back-propagation

Samotný algoritmus obsahuje tři etapy: dopředné (feed-forward) šíření vstupního signálu tréninkového vzoru, zpětné šíření chyby a aktualizace váhových hodnot na spojeních.[13]

Během dopředného šíření signálu obdrží každý neuron ve vstupní vrstvě vstupní signál a zprostředkuje jeho přenos ke všem neuronům vnitřní vrstvy. Každý neuron ve vnitřní vrstvě vypočítá svou aktivaci a pošle tento signál všem neuronům ve výstupní vrstvě. Každý neuron ve výstupní vrstvě vypočítá svou aktivaci, která odpovídá jeho skutečnému výstupu po předložení vstupního vzoru. V podstatě tímto způsobem získáme odezvu neuronové sítě na vstupní podnět daný excitací neuronů vstupní vrstvy.[13]

Během adaptace neuronové sítě metodou Back-Propagation jsou srovnávány vypočítané aktivity s definovanými výstupními hodnotami pro každý neuron ve výstupní vrstvě a pro každý tréninkový vzor. Na základě tohoto srovnání je definována chyba neuronové sítě, pro kterou je vypočítán faktor δ , jež odpovídá části chyby, která se šíří zpětně z daného neuronu ke všem neuronům předcházející vrstvy majícím s tímto neuronem definované spojení. Úprava váhových hodnot na spojeních mezi neurony konkrétní a vyšší vrstvy pak závisí na odpovídajícím faktoru δ a aktivacích neuronů ve dané vrstvě.[13]

7.7 Problém přeučení sítě

Zde nastává problém neúplné informace o průběhu učení. Jediné veličiny, které nám tuto informaci poskytují, jsou trénovací a testovací chyba. V ideálním případě by obě tyto veličiny současně konvergovaly k nule, ve skutečnosti ale ve fázi učení pravděpodobně nastane chvíle, kdy se klesající trend testovací chyby zastaví a chyba začne opět stoupat, zatímco trénovací chyba dále klesá. Tento jev se nazývá přeučení, neboli overfitting, kde se síť naučí rozeznávat příznaky, které jsou obsaženy pouze v prvcích trénovací množiny.[33]

7.7.1 Callback - předčasné ukončení

Existuje několik způsobů jak s tímto jevem bojovat. Nejjednodušším z nich je předčasné zastavení učení. Ve chvíli, kdy začíná testovací chyba růst, zastavíme učení a uložíme parametry sítě. Protože není vyloučené, že testovací chyba začne opět klesat, můžeme určit záchranné body po určitém počtu iterací, kde uložíme parametry sítě a následně vybrat ty s nejlepšími výsledky. [33]

7.7.2 Dropout

Dále s přeučením bojuje dropout. V průběhu učení přerušuje spojení mezi některými neurony dle náhodného výběru s předem určenou pravděpodobností. Díky tomu se průběžně mění váhy, přispívající do ztrátové funkce, což by mělo rovnoměrněji rozložit vliv vah, ovlivňujících výsledek predikce.[33]

7.7.3 Normalizace dat

Další možností jak omezit přeučení sítě je normalizace dat. Normalizace dat pomáhá kompenzovat rozdíly v rozsazích hodnot jednotlivých prvků vstupních dat. Pokud totiž mají přispívat všechny prvky stejnou vahou na výsledek predikce, musí být tyto rozdíly kompenzovány velikostí odpovídajících parametrů sítě, které budou mít větší vliv na vývoj gradientu a tedy i na optimalizaci parametrů sítě. Tento princip ale můžeme uplatnit na vstupy každé vrstvy a díky tomu dosáhneme efektivnějšího učení. [33]

7.8 Konvoluční neuronové sítě - CNN

Konvoluční neuronové sítě jsou podmnožinou neuronových sítí. Vyznačují se použitím konvolučních a pooling vrstev, a také algoritmu zpětného šíření (Back-propagation). Díky nim jsou schopny efektivně zpracovávat vstupy velkých rozměrů, jako jsou například obrázky, za použití mnohem menšího množství parametrů než obecná konvoluční síť. Díky tomu je snazší takovou síť naučit. Konvoluční sítě se zpravidla skládají z několika různých typů vrstev. Základní typy, použité v praktické části, tu budou popsány.[31]

7.8.1 Konvoluční operace

Konvoluční neuronové sítě obsahují konvoluční vrstvy, které tvoří jádro těchto sítí. Jak již název napovídá, využívají operace konvoluce. Dle definice by mělo před posouváním filtru po vstupu dojít k přetočení filtru, v konvolučních neuronových sítích se však často od tohoto kroku upouští - v tomto případě se operace správně nazývá cross-correlation, ale stále se používá označení "konvoluce".[31]

Tuto operaci si můžeme představit jako posouvání okna po vstupní matici a násobení překrývajících se hodnot. Tomuto oknu říkáme filtr (často také jádro, kernel nebo feature detector) a je typicky mnohem menší než vstupní data. Hodnoty uvnitř filtru se označují více výrazy, ale zde si je můžeme označit jako váhy - paralela ke klasickým neuronovým sítím. Každé posunutí okna produkuje jednu výstupní hodnotu, ze kterých je poté složena výstupní matice, která se běžně označuje jako feature map. Čím více filtrů konvoluční vrstva má, tím více rysů se naučí detekovat.[31]

Konvoluční vrstvy si můžeme představit jako neuronové sítě, kde se jeden neuron vyskytuje vícenásobně se stejnými vahami, ale je vždy připojen k jiné kombinaci vstupních neuronů (tento neuron odpovídá jednomu filtru). Zatímco počet operací se tedy nesníží, tak počet parametrů, které je třeba se naučit a uložit, se rapidně zmenší.[31]

Narozdíl od neuronových sítí se zde klade důraz na hledání lokální informace, nedíváme se na vstup jako celek, ale jako na množinu jeho částí. Konvoluční vrstvy se pak vrství za sebe, a zatímco filtry v počátečních vrstvách hledají prosté tvary, tak vrstvy na konci detekují složitější tvary. [31]

7.8.2 Konvoluční vrstva

Základním stavebním blokem konvolučních neuronových sítí jsou tzv. konvoluční vrstvy. Hlavní částí jsou třírozměrné filtry, jejichž šířka a výška je typicky malá (mnohdy menší než rozměry vstupu), ale hloubka je stejná jako u vstupu.

Předmětem učení jsou váhy filtrů. Počet filtrů, jejich výšku i šířku a krok, kterým se posouváme po vstupní matici nám udávají hyperparametry. Často je také možno určit i tzv. zero padding, který okraje dat doplní nulami, tím se zajistí, že vstup i výstup bude mít stejnou šířku a výšku.[33]

7.8.3 Maxpool vrstva

Maxpool vrstva slouží k snížení velikosti vstupů a tedy nepřímo i k zlepšení distribuce informace. Dále snižuje výpočetní náročnost učení. Každá vrstva má definovanou velikost jádra, tedy velikost podoblasti vstupní matice. Dále definujeme horizontální krok h a vertikální krok v , o které se podoblast posouvá vůči vstupní matici. Nejčastěji se setkáme s velikostí jádra $2 * 2$ a kroky o stejné velikosti jako je šířka jádra, tedy 2. vrstva vybere maximální hodnotu dané podoblasti a zapíše ji na výstup, poté se posune dále. Pro vstup o velikosti $m * n$ dostaneme výstup o velikosti $\frac{m}{v} * \frac{n}{h}$. [33]

7.8.4 Plně propojená vrstva

Plně propojené vrstvy (dense layer) jsou typicky umístěny na konci konvoluční neuronové sítě (CNN může obsahovat více takových vrstev zasebou), kde je výstup předchozích vrstev linearizován (typicky za pomoci flatten layer) do jednodimenzionálního pole a výstup poslední plně propojené vrstvy je pak celkovým výstupem konvoluční neuronové sítě. [33]

Jednoduše řečeno se jedná o obyčejnou plně propojenou neuronovou síť bez skrytých vrstev.

Obsahující následující hyperparametry:

- počet neuronů v plně propojené vrstvě
- zda využít bias neuronů nebo ne

7.8.5 Ztrátová funkce

Ztrátová funkce categorical crossentropy vypočítává ztrátu pomocí následující formule:

$$\text{Ztráta} = - \sum_{i=1}^n y_i * \log \hat{y}_i$$

Kde \hat{y}_i je i -tá skalární hodnota výstupu, y_i je odpovídající cílová hodnota a n je počet skalárních hodnot ve výstupu modelu.

Tato ztrátová funkce je dobrým měřítkem toho, jak rozlišitelné jsou dvě diskrétní rozdělení pravděpodobnosti. Znaménko mínus zajišťuje, že ztráta se zmenšuje, když se rozdělení blíží k sobě. [15]

7.8.6 Optimalizační algoritmus

Optimalizační algoritmus snaží najít takové nastavení vah, aby síť předpovídala hodnoty v co nejlepším souladu se zadanými závisle proměnnými. Tento soulad se měří součtem čtverců, jedná se tedy o metodu nejmenších čtverců, podobně jako v lineární nebo nelineární regresii. Obecně nemůže být zaručeno, že nalezené řešení je nejlepší možné. Proto se někdy může vyplatit spustit optimalizaci vícekrát a použít model s lepším reziduálním součtem čtverců (RSČ). Optimalizace začíná s náhodnými hodnotami vah, je proto přirozené, že každé jednotlivé řešení nalezené optimalizací je zcela odlišné od ostatních. I zcela rozdílné nastavení vah v síti může poskytnout prakticky identickou predikci modelu se shodnými RSČ pro zadaná data. Rozdíl se pak ovšem může někdy projevit v kvalitě predikce pro neznámá data.[33]

Adam

Optimalizační algoritmus Adam (Adaptive Moment Estimation) funguje následovně. Mějme parametry neuronové sítě Θ (například vektor vah jedné z vrstev) a dávku trénovacích dat o velikosti n . Adam uchovává exponenciálně klesající průměr z předchozích gradientů m_t a také exponenciálně klesající průměr z kvadrátů předchozích gradientů v_t , což jsou odhady středních hodnot a průměru gradientů. V nultém kroku jsou m_t a v_t inicializovány jako nulové vektory. [34]

Potom v každém kroku t :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial L_{x_i}}{\partial \Theta}$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial L_{x_i}}{\partial \Theta} \right)^2$$

m_t a v_t mají tendenci být blízko k nule na počátku učení, a to hlavně pokud jsou β_1 a β_2 blízko k jedničce. Tento problém se dá potlačit korekcí:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Takto opravené odhady jsou použité pro úpravu parametrů:

$$\Theta_t = \Theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} * \hat{m}_t$$

ϵ zajišťuje nenulovost jmenovatele. [34]

8 Diskriminační analýza (DA)

Jedná se o metodu z mnohorozměrné statistické analýzy (MSA - multivariate statistical analysis) jejímž cílem je zařadit objekty do skupin na základě jejich vlastností, které jsou popsány diskriminační funkcí nebo přiřazením nového objektu do předem známé skupiny pomocí klasifikačního pravidla.[26]

Třídění objektů se provádí pomocí tzv. Bayesovských (pracující s předpokládanými pravděpodobnostmi) nebo Fisherových (využívající projekci na směr největšího rozdílu mezi skupinami pozorování) pravidel. V tomto textu se omezíme pouze na Bayesovský přístup. [26]

8.1 Princip diskriminační analýzy

Mějme skupinu objektů, ve které je každý objekt charakterizován náhodným vektorem $X = (X_1 \dots X_p)^T$. Úkolem je tedy rozdělit objekty do dvou tříd π_1 a π_2 . Soubor objektů první třídy je populace x -ových hodnot z π_1 a obdobně hodnoty x druhé třídy tvoří populaci π_2 . Přitom předpokládáme, že objekt patřící do π_1 , (resp. π_2) se řídí spojitým rozdělením pravděpodobnosti s hustotou $f_1(x)$, (resp. $f_2(x)$). Označme Ω výběrový prostor objektů, které klasifikujeme. Z toho logicky vyplývá, že každý objekt x patří buď do třídy π_1 , nebo do třídy π_2 . [26]

Při třídění se ovšem může stát, že objekt, který ve skutečnosti patří do třídy π_1 , chybně zařadíme do π_2 . Přitom označme R_1 množinu všech objektů klasifikovaných do třídy π_1 . Množina R_2 obsahuje všechny objekty zařazené do třídy π_2 . Z čehož plyne, že sjednocením R_1 a R_2 získáme množinu všech objektů Ω , nebo-li R_1 a R_2 tvoří rozklad množiny Ω . Úkolem diskriminační analýzy je najít optimální rozklad. [26]

r Známe-li hustoty $f_1(x)$ a $f_2(x)$, potom pravděpodobnost špatného zařazení objektů patřících do třídy π_1 je podmíněná pravděpodobnost [26]

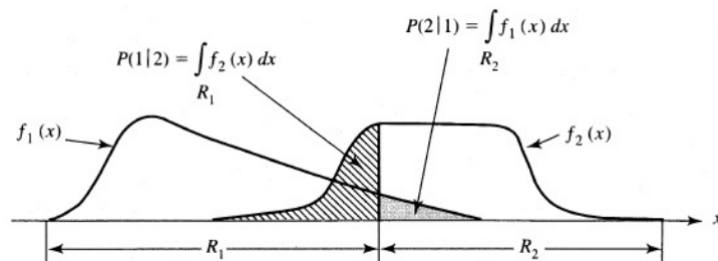
$$P(2|1) = P(X \in R_2 | \pi_1) = \int_{R_2 = \Omega R_2} \times f_1(x) dx$$

nebo naopak může nastat situace, kdy o objektu patřícím do třídy π_2 rozhodneme, že patří do třídy π_1 . Odpovídající podmíněná pravděpodobnost je

$$P(1|2) = P(X \in R_1 | \pi_2) = \int_{R_1} \times f_2(x) dx$$

Nechť p_1 je předpokládaná pravděpodobnost, která říká, že objekt pochází z třídy π_1 , p_2 pro objekt pocházející z třídy π_2 . Při splnění podmínky $p_1 + p_2 = 1$ zavádíme následující podmíněné pravděpodobnosti:

$$\begin{aligned} P(\text{správně zařazený objekt do } \pi_1) &= P(X \in R_1 | \pi_1) P(\pi_1) = P(1|1) p_1, \\ P(\text{chybně zařazený objekt do } \pi_1) &= P(X \in R_1 | \pi_2) P(\pi_2) = P(1|2) p_2, \end{aligned}$$



Obrázek 8.1: Pravděpodobnosti při špatné klasifikaci. [26]

$$P(\text{správně zařazený objekt do } \pi_2) = P(X \in R_2 | \pi_2)P(\pi_2) = P(2|2)p_2,$$

$$P(\text{chybně zařazený objekt do } \pi_2) = P(X \in R_2 | \pi_1)P(\pi_1) = P(2|1)p_1.$$

Špatné zařazení objektů pro nás představuje ztrátu. Logicky nulová ztráta nastane při bezchybné klasifikaci.[26]

8.2 Lineární diskriminační analýza

Lineární diskriminační analýza (LDA) je zobecněním Fisherovo diskriminační analýzy. Metoda je využívána ve statistice, rozpoznávání a strojovém učení ve snaze najít lineární kombinaci příznaků, které charakterizují nebo oddělují dvě nebo více tříd objektů. Výsledná kombinace může být použita jako lineární klasifikátor, nebo častěji pro redukci rozměrů před následnou klasifikací.[27]

LDA úzce souvisí s analýzou rozptylu (ANOVA) a regresní analýzou, která se také snaží vyjádřit závislou proměnnou jako lineární kombinaci dalších příznaků nebo měření.[27] Nicméně, ANOVA používá kategorické nezávislé proměnné a spojitou závislou proměnnou, zatímco diskriminační analýza má spojitě nezávislé proměnné a kategoricky závislou proměnnou (tj. Označení třídy). Logistická regrese a pravděpodobnostní regrese jsou více podobné LDA než ANOVA, protože také vysvětlují kategorickou proměnnou hodnotami spojitých nezávislých proměnných. Tyto metody jsou preferovány v aplikacích, kde není rozumné předpokládat, že nezávislé proměnné jsou normálně distribuovány, což je základní předpoklad metody LDA.[27]

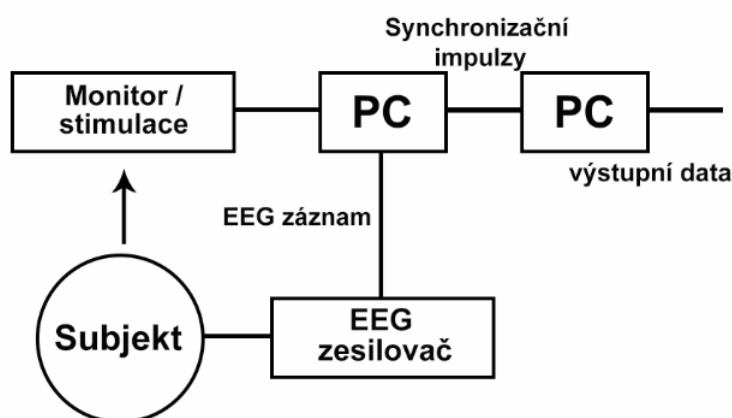
LDA také úzce souvisí s analýzou hlavních složek (PCA) a faktorovou analýzou v tom, že oba algoritmy hledají lineární kombinace proměnných, které nejlépe charakterizují data. LDA se přímo snaží modelovat rozdíl mezi třídami dat. PCA na druhé straně nebere v úvahu žádný rozdíl ve třídě a faktorová analýza vytváří kombinace funkcí založených spíše na rozdílech než podobnostech. [27]

9 Návrh experimentu

Celý tento experiment by měl sloužit jako odrazový můstek například pro asistenční systémy založené na SSVEP-BCI (Kapitola 6.4) za pomoci hlubokého učení, které by měly pomoci pacientům usnadnit komunikaci, pokud jim to jejich postižení nedovoluje běžným způsobem.

9.1 BCI systém

V zadání bakalářské práce je vytvoření scénáře pro stimulaci subjektů. Tento scénář musel být vytvořen na BCI systém, který je dostupný v laboratoři neurověd. V laboratoři se nachází odhlučněná komora s monitorem a EEG zesilovačem, který je připojen na počítač mimo odhlučněnou komoru. Na obrázku (Obrázek 9.1) můžeme vidět jak vypadalo výsledné schéma BCI systému. Subjekt byl umístěn před monitor, na kterém byla spuštěna stimulace a na hlavu mu byla nasazena EEG čepice, která byla připojena přes EEG zesilovač do počítače. Signály byly odesílány do druhého počítače, kde pomocí Brain Vision Recorder byly zaznamenány a uloženy do paměti.



Obrázek 9.1: Schéma realizace BCI systému

9.2 Stimulace

Pro vytvoření skriptu stimulace jsem zvolil programovací jazyk Python (verze 3.6). Jazyk Python jsem zvolil pro jeho jednoduchost, robustnost, ale také z toho důvodu, že obsahuje širokou nabídku modulů, které jsou předpřipravené pro práci v oblasti neuroinformatiky. Pro psaní samotného skriptu jsem měl na výběr z několika vývojových prostředí. Svou volbu jsem zúžil na vývojová prostředí Presentation a PsychoPy.

- **Presentation** - Jedná se o komerční software, který vyvinula společnost Neurobehavioral Systems. Umožňuje uživateli pomocí vestavěných jazyků naprogramovat

požadovaný scénář pro jak sluchovou, tak vizuální stimulaci. Tento software podporuje nejrůznější rozhraní pro I/O (vstup a výstup). Obsahuje také vestavěný editor pro psaní kódu scénáře, takže již není potřeba využívat jiné IDE (vývojové prostředí). Za jeho hlavní nevýhodu můžeme považovat fakt, že je komerční a pro jeho využití si musíme zakoupit licenci.

- **PsychoPy** - Jedná se o tzv. open source software, který je zdarma a na jeho zlepšování pracuje široká komunita. Tento program je vytvořený v programovacím jazyce Python a je určen pro generování experimentů v oblasti neurovědy a experimentální psychologie. Na rozdíl od většiny nabízeného softwaru, Psychopy umožňuje uživatelům volbu rozhraní. Jednak si můžeme vybrat generování experimentu pomocí grafického rozhraní, které dokáže samo vygenerovat výsledný script a nebo pomocí psaní scriptu v jazyce Python ve vestavěném editoru.

Rozhodl jsem se pro použití vývojového prostředí PsychoPy, neboť ho považuji za uživatelsky přívětivější a fakt, že se jedná o open-source, mi je bližší. Další výhodou je psaní skriptů v jazyce Python, který jsem využil také pro psaní klasifikačního skriptu.

9.2.1 Frekvence stimulace

Ke stimulaci jsem se rozhodl použít čtyři objekty, kde každý objekt měl nastavenou předem určenou frekvenci. Výběr vhodné frekvence je závislý na třech hlavních faktorech.

- Frekvence stimulace SSVEP - zde je potřeba zvolit frekvenci v rozsahu 5 Hz až 75 Hz
- Monitor - obnovovací frekvence monitoru je 60 Hz a pokud chceme stimulovat blízkajícím objektem, tak dosáhneme maximálně frekvence 30 Hz; dále je zde omezení, že určená frekvence musí dělit obnovovací frekvenci monitoru beze zbytku, aby se docílilo správného vykreslení.
- Harmonických násobné frekvence - zvolené frekvence nesmějí být násobky sebe samých, např. nelze použít 12 a 24 Hz.

Na základě těchto omezení a s využitím literatury [47] jsem zvolil frekvence **8.57, 12, 15 a 20 Hz** (viz. Obrázek 9.2)



Obrázek 9.2: Stimulační objekty, jejich markery:ID a frekvence

9.3 Scénář stimulace

Scénář je rozdělen na dvě části. V první části byl skript přizpůsoben sběru trénovacích dat a ve druhé části dat testovacích. Rozložení stimulačních objektů na obrazovce můžeme vidět na obrázku 9.2. Stimulace byla provedena opakovaným vykreslováním stimulačních objektů na obrazovku. Na začátku každé části se na obrazovku vypsaly pokyny, kterými se měl subjekt řídit.

1. **Trénovací část** - V první, trénovací části měl subjekt přesně určeno, na který stimulační objekt má soustředit svou pozornost. Toho bylo docíleno obarvením stimulačního objektu před začátkem stimulace. Díky tomu jsme provázali naměřená data s markery jednotlivých objektů. Celá trénovací část se skládala ze třech kol stimulace, mezi kterými byla pauza, aby subjekt ulevil očím. V každém cyklu se vystřídaly všechny čtyři stimulační objekty, to znamená, že jsme v trénovací fázi získali celkem dvanáct (3x4) událostí.
 - *Označení* - během označení se na obrazovku vykreslily všechny stimulační objekty, kde jeden z nich byl náhodně zvolen a obarven modře, tím jsme subjektu dali pokyn, že na tento objekt má při stimulaci soustředit svou pozornost; doba označení trvala 5 s
 - *Stimulace* - Po označení objektu následovala stimulace. Doba stimulace byla nastavena na 50 s.
 - *Pauza* - Po skončení stimulace, se na obrazovce objevil odpočet 30 s, ve kterých si měl subjekt odpočinout.
2. **Testovací část** - V této části se obdobně jako u trénovací části střídalo mezi stimulací a pauzou. Hlavním rozdílem zde bylo, že subjektu byl dopředu určen jeden objekt a na ten upínal svou pozornost po celou dobu testovací fáze. Doba stimulace zde byla nastavena na 60 s a délka pauzy byla 30 s. Opakování stimulace zůstalo zachováno, ale v každém cyklu jsme získali data pouze z jednoho stimulačního objektu. Celkem tedy tři (3x1) události.

Mezi jednotlivými částmi měl subjekt pauzu 10 minut.

9.3.1 Průběh měření

Celé měření od příchodu subjektu až po jeho odchod se dá rozdělit do několika následujících fází.

- **Administrace** - Po příchodu subjektu do laboratoře byl vyzván k vyplnění informovaného souhlasu (GDPR) a dotazníku. V dotazníku byly subjekty tázány na jejich fyzický a psychický stav, dále na konzumaci alkoholu, jaký stupeň stresu zažívají a kolik hodin denně spí. Tyto základní informace velice úzce souvisejí s psychickým stavem člověka a působí na změny v mozkové aktivitě. Tímto byl subjekt seznámen s riziky, které sebou experiment nese, hlavním rizikem byl projev epilepsie, kterou subjekt za žádných okolností nesměl být postižen, dále se jednalo o projevy klaustrofobie, neboť data byla snímána v odhlučněné komoře, aby se zamezilo vzniku ruchu. V dokumentu byl také subjekt seznámen s jednotlivými kroky měření.
- **Příprava subjektu** - Snímání dat bylo provedeno pomocí EEG čepice s připojenými elektrodami. Abychom zajistili co nejpřesnější měření, bylo potřeba očistit kůži od mazu v místech připevnění elektrod a nanesení vodivé pasty. Referenční elektroda byla připevněna doprostřed čela a uzemnění se připevnilo na ušní lalok. Poté následovalo nasazení čepice na hlavu subjektu a propojení skalpu s elektrodami v čepici pomocí elektrovedivého gelu. Samozřejmě jsme potřebovali minimalizovat odpor, který je na povrchu skalpu pod vlasy, takže bylo subjektu doporučeno umytí hlavy a tím odmaštění jejího povrchu. Tato příprava trvala přibližně 10 minut.
- **Postup měření** - Celý experiment se odehrával v laboratoři neuroinformatiky, kde je k dispozici speciálně odizolovaná komora, ve které se zamezí rušivých vlivů z okolí, takže se připraví pro subjekt ideální podmínky a naše data budou z tohoto hlediska co nejčistší. Subjekt byl posazen na křeslo a před ním byla obrazovka, na které mu byl zobrazen scénář pro měření trénovacích dat. Po skončení prvního scénáře, následovala 10 minut dlouhá pauza, ve které si měl subjekt odpočinout a odizolovaná komora byla otevřena. Po uplynutí času byl subjekt opět zavřen a byl mu spuštěn druhý scénář, ve kterém se sbírala testovací data. Celý experiment trval přibližně 30 minut.
- **Ukončení experimentu** - Po ukončení měření byly zaznamenány metadata o průběhu celého experimentu. Subjekty byly dotazovány zdali jim stimulace nezpůsobovala pocity nevolnosti, ospalosti a další nepříjemné pocity. Poté byly subjektu sundány monitorovací elektrody a poskytnuty prostředky pro vyčištění vodivého gelu z vlasů. Tímto byl celý experiment ukončen.

Během měření byla nasbírána data od 14 subjektů. V této množině byli pouze muži v průměrném věku 25 let.

10 Předzpracování a klasifikace

V této kapitole budou popsány metodiky, které byly použity pro zpracování signálu a vytvoření klasifikačního algoritmu.

10.1 Předzpracování dat

V této podkapitole si popíšeme formát vstupních dat, jejich následnou transformaci do požadované podoby a ukážeme si jak jednotlivá data vypadají.

10.1.1 Výchozí data

Data, která byla při experimentu naměřena, pochází ze softwaru BrainVision Recorder. Tato vstupní data byla naměřena na sedmi elektrodách, kterými jsou *Fp1*, *Fp2*, *P3*, *P4*, *O1*, *O2*, *Pz*. Tyto elektrody jsem vyhodnotil jako ideální pro snímání signálů SSVEP na základě literatury [3]. Nejvyšší odezvu na stimulování SSVEP mají elektrody *O1* a *O2*, neboť se nacházejí v oblasti týlního laloku, kde jsou umístěna zraková pole centrální nervové soustavy. U elektrod *Pz*, *P3* a *P4* již nejsou hledané odezvy tak vysoké a slouží jako kontrolní elektrody. Tyto elektrody jsou v datech značeny jako kanál 1-7.

Naše stimulační objekty jsou označeny číselnou hodnotou 2-5 a pro zjednodušení klasifikačních tříd jsou přejmenovány viz. následující kód(events ID).

```
1 events :
2 [522675      0      2]
3 [577771      0      5]
4 [632867      0      3]
5 [687963      0      4]
6 Channels names: ['Fp1', 'Fp2', 'P3', 'P4', 'O1', 'O2', 'Pz']
7 events ID: {'0': 2, '1': 3, '2': 4, '3': 5}
```

10.1.2 Načtení dat

Aby se data dala zpracovat, tak je potřeba je načíst a převést do podoby, aby jim rozuměl náš klasifikátor. K načtení a zpracování dat byla využita knihovna MNE [43].

Funkce která načtení dat zajistí je:

mne.io.read_raw_brainvision

Do této funkce vstupují následující parametry.

- *vhdr_fname* - vstupní *.vhdr* soubor
- *montage* - specifikujeme rozmístění elektrod na skalpu (použili jsme systém 10-20), čemuž odpovídá parametr: `mne.channels.read_montage(kind='standard_1020')`
- *preload* - zajistí, že data budou načtena hned při inicializaci

Následuje extrakce jednotlivých událostí, které jsou specifikované díky přiřazeným značkám a časovému razítku:

mne.events_from_annotations

Do této funkce vstupuje jediný parametr.

- *raw* - načtená vstupní data

Kód pro načtení raw dat a následné rozdělení do epoch:

```
1 picks = (4, 5) # Ch5=o1, Ch6=o2 - použijeme pouze data z kanalu 01 a 02
2 for i in range(INPUT_FILE_TRAINING.__len__()):
3     raw = mne.io.read_raw_brainvision(np.ravel(INPUT_FILE_TRAINING[i])[0],
4         preload=True, montage=montage)
5     events, event_id = mne.events_from_annotations(raw)
6     event_dict = {'0': 2, '1': 3, '2': 4, '3': 5}
7     epochs = mne.Epochs(raw, events=events, picks=picks,
8         event_id=event_dict, preload=True, tmin=-0.2,
9         tmax=50, baseline=(-0.2, 0))
10    loaded_data_training.append(epochs)
11    for x in range(epochs.__len__()):
12        shorter_train.append(epochs[x].copy().crop(tmin=0, tmax=10))
13        shorter_train.append(epochs[x].copy().crop(tmin=10, tmax=20))
14        shorter_train.append(epochs[x].copy().crop(tmin=20, tmax=30))
15        shorter_train.append(epochs[x].copy().crop(tmin=30, tmax=40))
16        shorter_train.append(epochs[x].copy().crop(tmin=40, tmax=50.1))
```

Jak z kódu můžeme vidět, celý proces zpracování dat do epoch je proveden v cyklu přes jednotlivé datové soubory, jejichž názvy jsou uloženy v proměnné *INPUT_FILE_TRAINING*. Poté následuje načtení dat ze souboru do proměnné *raw*, načtení eventů a markerů ze stimulace. Na šestém řádku můžeme vidět vytvoření slovníku, který přiřadí markerům jejich zástupný label 0-3, aby se s nimi lépe pracovalo.

Poté co máme všechny epochy načtené, tak následuje rozdělení epoch na dílce, neboť z pohledu klasifikace bude lepší větší počet krátkých epoch. Jedna epocha, která byla dlouhá 50 s se rozdělí na 5 epoch po 10 s. Před rozdělením epoch jsme měli $14 \times 12 = 168$ epoch (14 subjektů \times 12 událostí), ale po jejich rozdělení na 5 dílů získáme $14 \times 12 \times 5 = 840$ epoch z trénovací části experimentu. Obdobný proces následuje pro data z testovací části experimentu.

10.1.3 Předzpracování dat do výsledné podoby

Výběr příznaků je jedním z klíčových postupů předzpracování dat. Smyslem redukce dimenzionality je nejen ušetřit čas a prostor vyřazením nepodstatných částí dat, ale i zlepšit úspěšnost klasifikace či přesnosti reprezentace dat v budovaném systému potlačením vlivu šumových či jinak neinformativních příznaků.

Při výběru s tvorbě příznaků jsem postupoval následovně:

1. Vytvořil jsem cyklus přes všechny načtené epochy.
2. Nad aktuální epochou jsem zavolał metodu `psd_welch_preprocess()`.

```
1 def psd_welch_preprocess(epoch):
2     psd_norm = []
3     #labely obsazene v epose
4     labels_all = [epoch.events[:, -1] - 2]
5     # knihovna MNE, prevede signal na Power Spectral Density (PSD)
6     psd_data, freq = psd_welch(epoch, fmin=5,
7                               fmax=25, n_fft=2048)
8     # spojeni kanalu za sebou
9     channel_merge = np.ravel([psd_data[0][0], psd_data[0][1]])
10    # normalizace dat <0,1>
11    psd_norm.extend(normalize(channel_merge, order=2))
12    return np.ravel(labels_all), psd_norm
13
```

Do metody `psd_welch()` knihovny MNE vstupují následující parametry:

- `epoch` - aktuálně zpracovávaná epocha
- `fmin` - nejnižší frekvence, která nás zajímá, nastaveno na 5 Hz
- `fmax` - nejvyšší frekvence, který nás zajímá, nastavena na 25 Hz
- `n_fft` - je doporučeno, aby tato hodnota odpovídala násobkům 2; hodnota udává počet bodů, ze kterých se má FFT počítat. Naše epocha obsahuje 10000 bodů, tím že zmenšíme počet bodů, ze kterých bude výpočet proveden, získáme hladší průběh signálu, ve kterém jsou hledané frekvence výraznější a lépe se klasifikují.

3. Metoda nám vrátí pole přetransformovaných dat a pole labelů.

Tímto jsme dospěli do fáze, že máme data připravená pro klasifikaci.

10.2 Klasifikace

V tento moment již máme naše vstupní data předzpracovaná a je možné se přesunout do fáze klasifikace.

V této práci je částečně nahlíženo na klasifikační algoritmus jako na černou skříňku, neboť nepíšeme přímo zdrojový kód samotného algoritmu, ale nastavujeme daný klasifikační algoritmus pomocí vstupních argumentů a funkcí. Na základě několika konzultací bylo rozhodnuto, že v této práci porovnáme přesnost klasifikace Neuronové sítě (Kapitola 7) a LDA klasifikátoru (Kapitola 8.2).

10.2.1 Rozdělení vstupních dat

Abychom mohli klasifikační algoritmy efektivně natrénovat a otestovat, je potřeba vhodně rozdělit vstupní data do tří množin, kde jedna množina je volitelná.

K tomuto rozdělení použijeme metodu `train_test_split()` z knihovny `ScikitLearn`. Díky této funkci jsou naše vstupní data rozdělena do dvou množin, kde v jedné množině jsou testovací data a ve druhé trénovací data. Poměr, jakým jsou data rozdělena, je nastavitelný parametrem `test_size`. Například nastavením `test_size=0.25` se nám data rozdělí do těchto skupin v poměru 75:25, kde větší část dat je uložena pro trénovací množinu. Data jsou rozdělena v náhodném pořadí, abychom docílili přibližně stejného procentuálního zastoupení labelů v každé podmnožině.

Jako volitelná množina se dá považovat validační množina, kterou je možné zastoupit testovací množinou. V případě, že se rozhodneme validační množinu využít, tak se data nejčastěji rozdělují v poměru 50:25:25 nebo také 70:15:15 v zastoupení trénovací data, validační data a testovací data.

Validační množinu jsem nakonec nevyužil a místo ní jsem použil taktéž testovací množinu.

10.2.2 Hodnotící metriky

Pro otestování našeho modelu potřebujeme příslušné metriky, abychom mohli konstatovat závěry o klasifikaci. Parametr i je aktuálně zpracovávaný marker, n je počet všech klasifikovaných markerů, T_i jsou správné markery (True) a P_i jsou námi určené markery (Predicted).

- **Přesnost** (Accuracy) - pro každou instanci je definována jako podíl správně předvídaných markerů ku celkovému počtu (předpokládané a skutečné) markerů pro tuto instanci.

K tomu využijeme vzorec:

$$A = \frac{1}{n} \sum_{i=1}^n \frac{\|T_i \cap P_i\|}{\|T_i \cup P_i\|}$$

- **Určitost** (Precision) - vyjadřuje přesnost správně klasifikovaných dat ku celkovému počtu správných markerů

K tomu využijeme vzorec:

$$P = \frac{1}{n} \sum_{i=1}^n \frac{\|T_i \cap P_i\|}{\|P_i\|}$$

- **Citlivost** (Recall) - měří podíl skutečně pozitivních zařazení, které jsou jako takové správně identifikovány

K tomu využijeme vzorec:

$$R = \frac{1}{n} \sum_{i=1}^n \frac{\|T_i \cap P_i\|}{\|T_i\|}$$

- **F_1 skóre** (F_1 score) - je harmonický průměr určitosti a citlivosti

K tomu využijeme vzorec:

$$F_1 = \frac{1}{n} \sum_{i=1}^n \frac{2\|T_i \cap P_i\|}{\|T_i\| + \|P_i\|}$$

10.3 Neuronová síť

Pro klasifikaci neuronovou sítí byla využita knihovna *Keras*, která běží jako nadstavba nad knihovnou *Tensorflow*.

- *Keras* - přímo nepodporuje algoritmy hlubokého učení sama o sobě, ale je to spíše nadstavba, která využívá další knihovny neuronových sítí jako například *Tensorflow*.
- *Tensorflow* - je volně dostupná knihovna vyvinutá společností Google jako alternativa pro knihovnu *Theano*. *Tensorflow* má na rozdíl od *Theana* výrazně rychlejší běh a obsahuje podporu i pro jiné metody hlubokého učení, než jen pro neuronové sítě, jako je tomu u *Theana*.

Pro klasifikaci pomocí neuronové sítě, jsem se rozhodl pro využití konvolučních neuronových sítí, které jsou mimo klasifikaci obrazových dat vhodné i na klasifikaci EEG signálu (viz. studie [32]).

Ještě před vstupem dat do neuronové sítě jsou labely převedeny na kategorizované vektory. Toho je docíleno pomocí funkce *to_categorical()* knihovny *Keras*.

Výsledné labely vypadají následovně:

```
0 -> [1,0,0,0]
1 -> [0,1,0,0]
2 -> [0,0,1,0]
3 -> [0,0,0,1]
```

10.3.1 Model neuronové sítě

1. Vrstva *Conv1D* - první vrstva našeho modelu neuronové sítě je konvoluční vrstva (viz. kapitola 7.8). Jedná se o jednodimenzionální variantu, neboť EEG signál má pouze jednu dimenzi.

```
1 model.add(Conv1D(filters=8, kernel_size=2, padding="same",
2   activation='swish', input_shape=(np.shape(x_train)[-2:])))
3
```

Parametry konvoluční vrstvy:

- *filter* - nastaveno na 8; určuje počet posuvných okének
 - *kernel_size* - nastaveno na 2; velikost posuvného okénka
 - *padding* - nastaveno na *same*; nastaví okraje dat tak, aby výstup měl stejnou délku jako vstup
 - *activation* - nastaveno na *swish*; jedná se o druh aktivační funkce
 - *input_shape* - dimenze vstupu; nastavuje se pouze u první vrstvy
2. vrstva *BatchNormalization* - normalizuje data ve skupinách po aktivační funkci z předchozí vrstvy

3. vrstva Dropout - v této vrstvě se určí část výstupů z předešlých skrytých vrstev a zahodí se (nastaví se 0) a díky tomu předejme přeučení sítě.
4. vrstva MaxPooling1D - snižuje velikost a komplexnost vstupu a to tak, že extrahuje dominantní vlastnosti
5. vrstva Flatten - redukuje dimenzi dat z předchozích vrstev, aby se výstup dal použít v plně propojené vrstvě
6. vrstva Dense - jedná se o plně propojenou vrstvu neuronové sítě, tato vrstva obsahuje 15 neuronů a je zde povolený bias vektorů dat
7. Následuje opakování vrstev *BatchNormalization*, *swish* aktivace a *dropout*
8. Vrstva Dense - výstupní vrstva se čtyřmi neurony (jejich počet odpovídá velikosti kategorizovanému vektoru) a jako aktivační funkce je zde nastavena funkce *Softmax*

10.3.2 Kompilace modelu

Poté co model obsahuje požadované vrstvy, tak můžeme celý model zkompilevat a tím ho připravit na trénování. Kompilaci modelu zajišťuje funkce:

```

1 model.compile(loss=keras.losses.categorical_crossentropy,
2               optimizer=keras.optimizers.adam(),
3               metrics=['categorical_accuracy'])

```

Do které vstupují následující parametry:

- *loss='categorical_crossentropy'* - nastavení ztrátové funkce *categorical_crossentropy*
- *optimizer='adam'* - nastavení optimalizačního algoritmu *Adam*
- *metrics=[categorical_accuracy]* - Metrika je funkce, která slouží k posouzení výkonu našeho modelu. Metrická funkce je podobná ztrátové funkci s tím rozdílem, že výsledky hodnocení metriky nejsou použity při tréninku modelu. Funkci ztráty lze využít i jako metrickou funkci. [38]

V této chvíli je náš model neuronové sítě funkční, ale stále nedokáže klasifikovat data, neboť nebyl natrénován.

10.3.3 Trénování modelu

Pro natrénování zkompilevaného modelu využijeme následující funkci:

```

1 model.fit(x_train, y_binary_train,
2          batch_size=64,
3          epochs=n_epochs,
4          validation_data=(x_test, y_binary_test), shuffle=True)

```

Do které vstupují následující parametry:

- *x = x_train* - pole trénovacích dat (pokud model má jeden vstup), nebo seznam polí (pokud má model více vstupů) - v našem případě se jedná o seznam polí
- *y = y_binary_train* - pole cílových markerů (label) (pokud model má jeden výstup), nebo seznam polí (pokud má model více výstupů) - v našem případě se jedná o seznam polí, protože jsou markery kategorizované

- `batch_size = 64` - počet trénovacích vzorů, které se mají zpracovat najednou
- `epochs=n_epochs` - udává počet epoch; 1 epocha je 1 iterace přes všechna vstupní data; v našem případě nastaveno `n_epochs=500`
- `shuffle = True` - zda mají být zamíchána trénovací data před každou epochou

Nyní máme náš model neuronové sítě natrénovaný a je možné jej použít pro klasifikaci dat.

10.4 LDA

Pro klasifikaci pomocí lineárně diskriminační analýzy je použita knihovna `sklearn.discriminant_analysis` a její funkce `LinearDiscriminantAnalysis`. K vytvoření modelu LDA klasifikátoru použijeme funkci:

LinearDiscriminantAnalysis()

Do které vstupuje následující parametr:

- `solver='lsqr'` - řešení za pomoci nejmenších čtverců (Least squares solution) [39]

Tímto vytvoříme model LDA klasifikátoru. Následuje jeho natrénování na vstupních datech pomocí funkce `fit(x_train, y_train)`, do které vstupují trénovací data, náš model je tímto natrénovaný a připravený ke klasifikaci.

10.5 Výstupní data

Kategorizovaný vektor získaný po klasifikaci jako předpověděný label je převeden pomocí funkce `numpy.argmax()`, knihovny `Numpy`, zpátky na celočíselnou podobu.

```
[1,0,0,0] -> 0
[0,1,0,0] -> 1
[0,0,1,0] -> 2
[0,0,0,1] -> 3
```

Poté se přidá do pole všech předpověděných labelů a to je výstupem našeho klasifikačního algoritmu. Tímto způsobem jsou získány dvě pole labelů, kde jedno pole je pro klasifikaci pomocí CNN a druhé pro LDA. Následně se provede průnik s polem pravdivých labelů a tím získáme procentuální úspěšnost klasifikace.

11 Dosažené výsledky

Implementované klasifikátory konvoluční neuronové sítě (CNN) a lineární diskriminační analýzy (LDA) jsem využil při analýze dat nasbíraných během testovací fáze scénáře. Celý experiment získání průměrných hodnot klasifikace napříč jednotlivými subjekty byl proveden následovně. Vytvořil jsem cyklus přes jednotlivé subjekty, ve kterém byl další cyklus s počtem opakování rovno pěti. V malém cyklu jsem natrénovával model na zamíchaných datech z trénovací fáze experimentu a poté jsem klasifikoval data z testovací fáze experimentu daného subjektu. Výsledek každého měření jsem uložil do pole pro pozdější vyhodnocení. Pole, ve kterém byly výsledky klasifikace uloženy mělo tvar 14x5. Celý tento proces trval 2h, z čehož se 95% času strávilo na učení modelu a 5% na následné klasifikaci.

Tímto jsem získal 70 výsledků klasifikace pro všechny subjekty a přes jednotlivé subjekty byl proveden průměr daných pěti hodnot. Naměřené výsledky jsou zobrazeny v tabulkách 11.1, 11.2 a 11.3. Je zde vypsána nejvyšší úspěšnost klasifikace daného subjektu, poté jeho střední hodnota a medián. V poslední je celkový průměry všech výše vypsanych hodnot a reflektuje výsledek klasifikace na všech subjektech najednou.

Subjekt	1	2	3	4	5
Přesnost CNN: max [%]	72.22	44.44	55.56	77.78	94.44
Přesnost CNN: střední hodnota [%]	58.89	34.44	33.33	63.33	87.78
Přesnost CNN: medián [%]	55.56	33.33	27.78	66.67	88.89
Přesnost LDA: max [%]	55.56	55.56	61.11	72.22	72.22
Přesnost LDA: střední hodnota [%]	52.22	51.11	55.56	66.67	71.11
Přesnost LDA: medián [%]	50.00	50.00	55.56	66.67	72.22

Tabulka 11.1: Tabulka dosažených výsledků 1

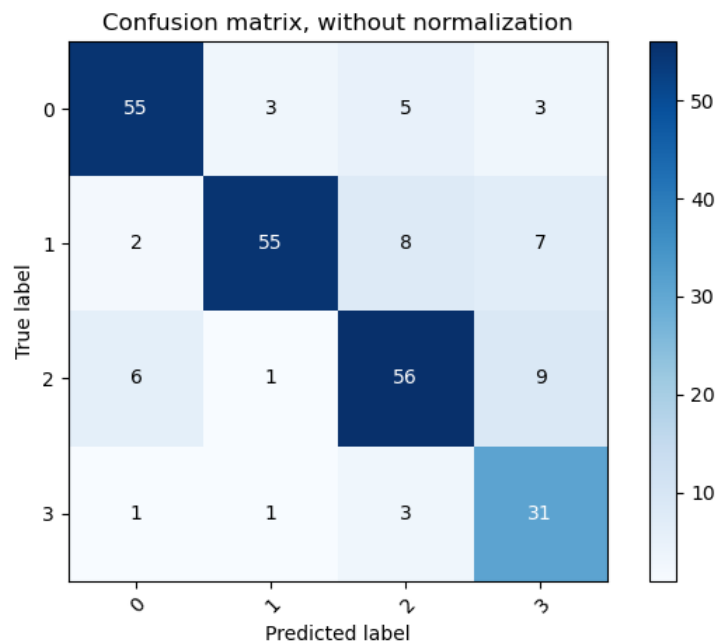
Subjekt	6	7	8	9	10
Přesnost CNN: max [%]	94.44	94.44	94.44	94.12	100
Přesnost CNN: střední hodnota [%]	84.44	94.44	87.78	92.94	95.3
Přesnost CNN: medián [%]	88.89	94.44	88.89	94.12	94.12
Přesnost LDA: max [%]	88.89	94.44	77.78	100	100
Přesnost LDA: střední hodnota [%]	81.11	93.33	76.67	100	97.65
Přesnost LDA: medián [%]	77.78	94.44	77.78	100	100

Tabulka 11.2: Tabulka dosažených výsledků 2

Subjekt	11	12	13	14	Všechny subjekty
Přesnost CNN: max [%]	100	100	70.59	76.47	Ø81.44
Přesnost CNN: střední hodnota [%]	98.82	100	56.47	70.59	Ø75.11
Přesnost CNN: medián [%]	100	100	52.94	70.59	Ø83.33
Přesnost LDA: max [%]	100	100	58.82	82.35	Ø79.92
Přesnost LDA: střední hodnota [%]	95.55	76.67	95.3	100	Ø73.16
Přesnost LDA: medián [%]	100	94.12	47.06	64.71	Ø72.22

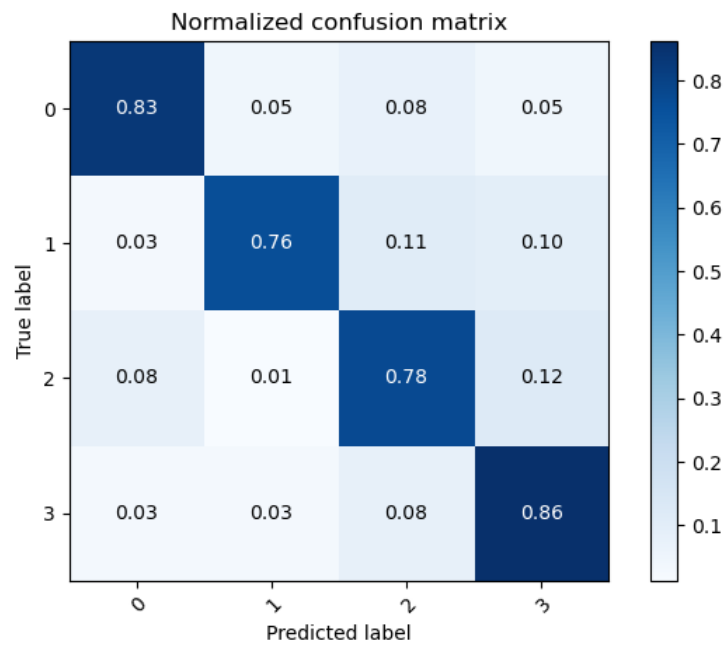
Tabulka 11.3: Tabulka dosažených výsledků 3 (Ø - znak průměru)

Z výsledků klasifikace je jasné, že oba klasifikační algoritmy dosahovaly podobných hodnot. Průměrná přesnost CNN klasifikátoru byla 75% a LDA klasifikátoru 73%. Dále jsem vygeneroval matici záměn výsledků klasifikace přes všechny subjekty najednou.



Obrázek 11.1: Matice záměn - 246 epoch

V matici 11.1 jsou vypsané počty správně a špatně určených labelů a v matici 11.2 je normalizovaná hodnota, která po vynásobení stem odpovídá procentuálnímu poměru klasifikace dat.



Obrázek 11.2: Normalizovaná matice záměn - 246 epoch

Dále jsem nechal vypsát zprávu (tabulka 11.4) o výsledku klasifikace konvoluční neuronovou sítí, kde jsou uvedeny hodnotící metriky (viz. kapitola 10.2.2).

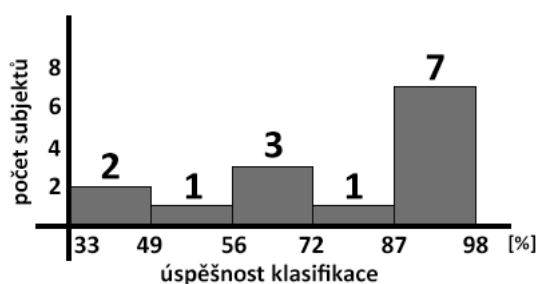
Labely	Určitost	Citlivost	F1-skóre	Počet epoch
0	0.86	0.83	0.85	66
1	0.92	0.76	0.83	72
2	0.78	0.78	0.78	72
3	0.62	0.86	0.72	36
Přesnost			0.80	246
Klouzavý průměr	0.79	0.81	0.79	246
Vážený průměr	0.82	0.80	0.80	246

Tabulka 11.4: Zpráva o klasifikaci všech subjektů

12 Diskuze

Při sběru dat od jednotlivých subjektů se nevyskytl žádný vážný problém, pouze jeden subjekt trpěl nesnášenlivostí jehel, a proto byla aplikace vodivého gelu náročnější a u jednoho subjektu se projevil mírné klaustrofobické stavy. Avšak devět subjektů si stěžovalo, že stimulace je nepříjemná pro oči a z toho usuzují, že čas stimulace by nebylo vhodné o moc prodlužovat. U jednoho subjektu byly hledané frekvence velmi slabé a je možné, že to zapříčinily dioptrické brýle. Toto tvrzení nemohu nijak potvrdit a je to pouze má domněnka.

V předchozí kapitole jsou vypsané výsledky klasifikace jednotlivých subjektů pomocí konvoluční neuronové sítě a lineární diskriminační analýzy. Je zde vidět, že oba klasifikační algoritmy dosahovali přibližně stejných průměrných hodnot a to 75% úspěšné klasifikace. Pokud se ale podíváme na následující histogram (Obrázek 12.1) úspěšnosti klasifikace.



Obrázek 12.1: Histogram úspěšnosti klasifikace

Můžeme zde pozorovat trend, že data od poloviny subjektů byla klasifikována s úspěšností větší než **87%**. Dokonce data od pěti subjektů dosáhly úspěšnosti klasifikace větší než **92%**. Pouze data od tří subjektů dosahovala nižší úspěšnosti klasifikace. Tyto rozdílné hodnoty úspěšnosti klasifikace nejsou způsobeny špatným klasifikátorem, nýbrž samotnými daty, která jsem získal při experimentu. Již po naměření dat jsem provedl jejich analýzu pomocí nástroje EEGLAB a zjistil jsem, že některé subjekty na stimulaci nereagují požadovanou intenzitou a tím je zhoršena možnost klasifikace. Jedná se hlavně o subjekty číslo dva a tři. Například subjekt číslo pět vykazuje enormní odezvu na alfa vlny (8 - 12 Hz) a tím dokonce zhoršil viditelnost odezvy na frekvenci 15 Hz, ale i přes to se jeho data podařilo klasifikovat s úspěšností **85%**.

Pokud bychom z našeho výčtu odstranili subjekty, které nevykazují požadovanou odezvu (subjekt č. 1, 2, 3 a 13), tak by se průměrná úspěšnost klasifikace vyšplhala na **87%**. Tato hodnota je dle mého názoru velice dobrá, vzhledem k tomu, že se klasifikují SSVEP signály, na které subjekty mohou, ale nemusí mít dostatečnou odezvu.

13 Závěr

V této práci jsem se věnoval problematice využití neuronových sítí v BCI systémech založených na odezvě ustálených vizuálně evokovaných potenciálů (SSVEP). Proto jsem se musel seznámit s problematikou týkající se elektroencefalografie (EEG), evokovaných potenciálů, konkrétně ustálených VEP, asistenčních systémů, BCI systému na bázi f-VEP a v poslední řadě s klasifikačními algoritmy.

Práce je zaměřena v oboru neuroinformatiky, což je věda na pomezí medicíny a informatiky. Musel jsem prostudovat literaturu z obou těchto oborů, a proto začátek práce pojednává spíše o lékařské problematice. Bez těchto informací bychom se ale neobešli ve fázi přípravy experimentu.

Cílem této práce bylo navrhnout jednoduchý BCI systém, který je založen na f-VEP stimulačním protokolu a následně implementovat část klasifikačního systému. Klasifikační systém je založen na klasifikaci EEG dat zvolenou neuronovou sítí. Pro získání dat na klasifikaci byl navrhnout a implementován scénář, díky kterému jsem sbíral EEG data celkem od 14 dobrovolných subjektů. Pro implementaci scénáře jsem zvolil aplikaci PsychoPy, která obsahuje editor pro tvorbu experimentů z oblasti neurověd. Samotný experiment je tvořen formou skriptů psaných v programovacím jazyce Python.

Celý experiment probíhal bez větších problémů. Většina subjektů si nestěžovala na případné obtíže, jen jednou se stalo, že subjekt trpěl nesnášenlivostí jehel, a proto byla aplikace vodivého gelu do EEG čepice pomocí jehly náročnější. Dále se u subjektu projevil pocit klaustrofobických stavů, protože měření probíhala v uzavřené zvukotěsné komoře.

Poté co jsem od prvních subjektů získal trénovací a testovací sadu dat, tak jsem mohl začít s návrhem a s následnou implementací klasifikačního algoritmu. Povedlo se mi implementovat konvoluční neuronovou síť a LDA algoritmus pro klasifikaci naměřených dat. Následně jsem provedl natrénování obou algoritmů na datech z trénovací fáze a poté jsem je otestoval na datech z testovací fáze scénáře. Klasifikace probíhala v režimu off-line, při kterém se nejprve nasbírají data, poté se předzpracují a následně klasifikují.

Klasifikátor pomocí konvoluční neuronové sítě dosahuje dle mého názoru velmi dobré úspěšnosti klasifikace a to 87,5% napříč všemi subjekty, velmi podobné úspěšnosti dosahuje i algoritmus LDA, který klasifikuje data s 73% úspěšností.

Literatura

- [1] HRAZDIRA, Ivo a Vojtěch MORNSTEIN. *Lékařská biofyzika a přístrojová technika. 1. vydání.* Brno, Neptun, 2001. ISBN 80-902896-1-4.
- [2] HEŘMAN, Petr. *Biosignály z pohledu biofyziky. 1. vydání.* Praha, Petr Heřman – DÚLOS, 2006.
- [3] ESMAILDOKHT MAMAGHANI, Amir Hossein. *Analýza a zpracování EEG signálu [online].* Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, Brno, 2012. Dostupné z:
<http://hdl.handle.net/11012/17200>
- [4] TROJAN, Stanislav. *Lékařská fyziologie* GRADA Publishing, září 2008. Dostupné z:
<https://books.google.cz/books?id=Kwr8Mc2Wg4MCpg=PA6731pg=PA673v=onepageqf=false>
- [5] Antti Savelainen *An introduction to EEG artifacts* Mat-2.4108 Independent research projects in applied mathematics Dostupné z:
<https://pdfs.semanticscholar.org/5cd2/3372a19187f7dfb61c1443879f392473e250.pdf>
- [6] KAŇOVSKÝ, P. *DISKRÉTNÍ TRANSFORMACE - Matematika pro inženýry 21. století* (reg. č. CZ.1.07/2.2.00/07.0332), Vysoká škola báňská – Technická univerzita Ostrava a Západočeská univerzita v Plzni
- [7] Jaromír ŠVEJDA, Roman ŽÁK, Roman JAŠEK *Zpracování Mozkové Aktivity V Bci Systémech* Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, nám. T.G. Masaryka 5555, 760 05 Zlín, Dostupné z:
http://trilobit.fai.utb.cz/zpracovani-mozkove-aktivity-v-bci-systemech_e498d494-fd80-4948-a8d6-f5f3720bba2d
- [8] Podyablonskiy A. *Umělé neuronové sítě a jejich aplikace* Praha, Katedra matematiky, statistiky a informačních technologií, Dostupné z:
https://is.ambis.cz/th/ml6jt/130426_BP_Podyablonskiy.pdf
- [9] MIŠUREC, J., CHMELARĚ, M. *Elektroencefalografie. 1. vydání.* Brno, IDV SZP, 1990, 194s. ISBN 80-701-3065-2.
- [10] Nuwer, Marc R. "Fundamentals of evoked potentials and common clinical applications today" (February 1998). *Electroencephalography and Clinical Neurophysiology*. 106 (2): 142–148. doi:10.1016/S0013-4694(97)00117-X
- [11] Ding J, Sperling G, Srinivasan R. "Attentional modulation of SSVEP power depends on the network tagged by the flicker frequency". *Cereb. Cortex.* (2006): 1016–29. doi:10.1093/cercor/bhj044. PMC 1880883. PMID 16221931. Dostupné z:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1880883/>
- [12] Václav Matoušek, Pavel Král. *4. Klasifikace, rozpoznávání a shlukování* Umělá inteligence a rozpoznávání, LS 2015 Dostupné z:
<http://home.zcu.cz/~pkral/uir/pr6-materialy/FThema4-select2.pdf>

- [13] ŠÍMA, Jiří, NERUDA, Roman. *Teoretické otázky neuronových sítí. 1. vyd.* Praha : MATFYZPRESS, 1996. 390 s. Dostupný z <http://www2.cs.cas.cz/sima/kniha.pdf>
- [14] MAŘÍK V., ŠTĚPÁNKOVÁ O., LAŽANSKÝ J. a kol. *Umělá inteligence 4. 1.* Academia, 2003. 476 s. ISBN 80-200-1044-0. [cit. 2010-10-29]
- [15] <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss->
- [16] A. Lukášová, J. Šarmanová *Metody shlukové analýzy* SNTL, Praha 1985
- [17] Du, S. S.; Lee, J. D.; Tian, Y.; *Gradient Descent Learns One-hidden-layer CNN: Don't be Afraid of Spurious Local Minima. 2017.*
<https://arxiv.org/pdf/1712.00779.pdf>
- [18] UIR/KRO *Klasifikace a rozpoznávání, výukový program KRO KIV/ZČU* Dostupné z:
<https://www.kiv.zcu.cz/studies/predmety/uir/KR02/>
- [19] Guangyu Bin, Xiaorong Gao, Yijun Wang, Bo Hong, and Shangkai Gao *VEP-Based Brain-Computer Interfaces: Time, Frequency, and Code Modulations [online].* Tsinghua University, China Dostupné z:
<https://sccn.ucsd.edu/yijun/pdfs/IEEEECIM09.pdf>
- [20] Rajesh Singla *SSVEP-Based BCIs [Chapter 6][online].* Dostupné z:
<http://dx.doi.org/10.5772/intechopen.75693>
- [21] Front. Hum. Neurosci, 04 January 2019 Dostupné z:
<https://doi.org/10.3389/fnhum.2018.00524>
- [22] Camron Godbout *What Killed the Curse of Dimensionality?[online].* Dostupné z:
<https://hackernoon.com/what-killed-the-curse-of-dimensionality-8dbfad265bbe>
- [23] Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, Tuan Anh Nguyen *A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations[online].* Dostupné z:
<https://arxiv.org/abs/1901.10854>
- [24] HONDROU, C., CARIDAKIS, G. *Affective, Natural Interaction Using EEG: Sensors, Application and Future Directions [online]. Image, Video and Multimedia Systems Lab, NTUA, 2012.* Dostupné z:
<http://www.image.ece.ntua.gr/papers/758.pdf>
- [25] BAREŠ, Martin *Kognitivní evokované potenciály [online].* Cesk Slov Neurol N 2011; 74/107(5): 508-517 Dostupné z:
<http://www.csnn.eu/ceska-slovenska-neurologie-clanek/kognitivni-evokovane-potencialy-36052>
- [26] Hebák, P. *Vícerozměrné statistické metody (1)*, 2. vydání, Praha: Informatorium, 2007
- [27] Aleix M. Martinez *PCA versus LDA*, Dostupné z:
<http://www2.ece.ohio-state.edu/aleix/pami01.pdf>

- [28] Rectified Linear Units (ReLU) in Deep Learning | Kaggle. Kaggle: Your Home for Data Science [online]. Copyright © 2019 Kaggle Inc. Dostupné z: <https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning>
- [29] Uniqtech, 30.1.2018 Dostupné z: <https://www.medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>
- [30] Prajit Ramachandran, BarretZoph, Quoc V. Le *SEARCHING FOR ACTIVATION FUNCTIONS* Google Brain;arXiv:1710.05941v2 [cs.NE] 27 Oct 2017 Dostupné z: <https://arxiv.org/pdf/1710.05941.pdf>
- [31] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: *ImageNet Classification with Deep Convolutional Neural Networks*. In *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., 2012, . Dostupné z: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deepconvolutional-neural-networks>
- [32] Nik Khadijah Nik Aznan†, Stephen Bonner, Jason D. Connolly‡Noura Al Moubayed and Toby P. Breckon *On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks* arXiv:1805.04157v2 [cs.HC] 2 Aug 2018 <https://arxiv.org/pdf/1805.04157.pdf>
- [33] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *ImageNet classification with deep convolutional neural networks*. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25* Curran Associates, Inc., 2012.
- [34] Kingma, D. P. a Ba, J. (2015). *Adam: A method for stochastic optimization*. In Proceedings of the 3rd International Conference on Learning Representations (ICLR)
- [35] TensorFlow Core r1.14, TensorFlow. TensorFlow Dostupné z: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adamax
- [36] Jason Brownlee, 30.1.2019, Better Deep Learning Dostupné z: <https://www.machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>
- [37] T. Řondík. *Methods for Detection of ERP Waveforms in BCI Systems* Západočeská universita v Plzni, 2012.
- [38] Dokumentace Keras, Dostupné z: <http://www.faroit.com/keras-docs/1.2.0/metrics/>
- [39] Dokumentace scikit-learn [online]. Dostupné z: https://www.scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html
- [40] MLČOCH, Zbyněk *Evokované potenciály - definice, způsob provedení (vyšetření), využití, k čemu slouží [online]*. Dostupné z: <https://www.zbynekmlcoch.cz/medicina/neurologie-nemoci-vysetreni/evokovane-potencialy-definice-zpusob-provedeni-vysetreni-vyuziti-k-cemu-slouzi>
- [41] ŠRÁMKOVÁ, Pavla *Speciální vyšetřovací metody v oftalmologii [online]*. Dostupné z: <https://zdravi.euro.cz/clanek/sestra/specialni-vysetrovaci-metody-v-oftalmologii-301150>
- [42] SMĚKAL, Zdeněk. *Číslíkové zpracování signálů*. 2009. 203 s. Vysoké Učení Technické v Brně.

- [43] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann *MEG and EEG data analysis with MNE-Python [online]* Frontiers in Neuroscience, 2013, 10.3389/fnins.2013.00267, Dostupné z: <https://www.frontiersin.org/article/10.3389/fnins.2013.00267>
- [44] POKORNÝ, Jan. *Evokované potenciály, Přednáška, [online]*. Dostupné z: <https://docplayer.cz/21771845-Evokovane-potencialy-doc-mudr-jan-pokorny-drsc.html>
- [45] (PDF) Bacteria Hunt: A multimodal, multiparadigm BCI game. ResearchGate | Find and share research [online]. Copyright © 1977 International Federation for Medical [cit. 24.07.2019]. Dostupné z: https://www.researchgate.net/publication/41035463_Bacteria_Hunt_A_multimodal_multiparadigm_BCI_game
- [46] ZAFER İşcan, NIKULIN VV *Steady state visual evoked potential (SSVEP) based brain-computer interface (BCI) performance under different perturbations [online]* January 23, 2018: Xu Lei, School of Psychology, CHINA, Dostupné z: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191673>
- [47] Han, Chengcheng et al. “*Highly Interactive Brain-Computer Interface Based on Flicker-Free Steady-State Motion Visual Evoked Potential.*” Scientific reports vol. 8,1 5835. 11 Apr. 2018, doi:10.1038/s41598-018-24008-8
- [48] Jan Havlík, Elektroencefalograf s počítačovou analýzou signálu, [cit.16.7.2019] BAM31LET Lékařská technika, Katedra teorie obvodů, xhavlikj@fel.cvut.cz, [online] Dostupné z: https://cw.fel.cvut.cz/b181/___media/courses/bam31let/11a_elektroencefalografie.pdf
- [49] Elektroencefalografie – WikiSkripta, ISSN 1804-6517, e-mail: info@wikiskripta.eu.[cit.16.7.2019] [online] Dostupné z: <https://www.wikiskripta.eu/w/Elektroencefalografie>
- [50] Ingrid E. Scheffer, Příklad EEG signálu, Brain (1995), 118, str. 61-73, [cit.16.7.2019] [online] Dostupné z: <https://pdfs.semanticscholar.org/803d/e4d45bb82207a44e6d3144be86f3ab07f635.pdf>
- [51] Szilvia Linnert, Eötvös Loránd University, Faculty of Education and Psychology, ERP analysis in NetStation, [cit.16.7.2019] [online] Dostupné z: https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0018_07_erp_analysis_netstation/ch01s03.html
- [52] MUDr. Zbyněk Mlčoch, Příklady technických artefaktů, [cit.16.7.2019] [online] Dostupné z: <https://www.zbynekmlcoch.cz/medicina/neurologie-nemocivysvetreni/artefakty-na-eeg-zaznamu-obrazky-deleni-eeg-artefaktu>
- [53] Pulzně kódová modulace, [cit.16.7.2019] [online] Dostupné z: https://www.wikiwand.com/cs/Pulzn%C4%9B_k%C3%B3dov%C3%A1_modulace
- [54] File:Aliasingplot.png - Wikimedia Commons., [cit.16.7.2019] [online] Dostupné z: <https://commons.wikimedia.org/wiki/File:Aliasingplot.png>
- [55] Ing. Pavel Mautner, Ph.D., Analýza a zpracování signálů, Symetrie DFT pro lichý a sudý počet vzorků., [cit.16.7.2019] [online] Dostupné z: http://www.kiv.zcu.cz/mautner/Azs/Azs5_Fourierova_transformace.pdf

[56] Schéma neuronových sítí, [cit.16.7.2019]
[online] Dostupné z: <http://www.portal.matematickabiologie.cz/res/image/Umela%20inteligence/obr-4-4-usporadani-neuronu-do-vrstev.png>

[57] Schéma neuronu, [cit.16.7.2019]
[online] Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471

Seznam obrázků

2.1	Elektroencefalograf s počítačovou analýzou signálu [48]	9
2.2	Rozmístění elektrod systémem 10-20 [49]	10
2.3	Příklad EEG signálu [50]	11
2.4	Průběh alfa vln [51]	11
2.5	Průběh beta vln [51]	12
2.6	Průběh gamma vln [51]	12
2.7	Průběh delta vln [51]	12
2.8	Průběh theta vln [51]	13
2.9	Příklady biologických artefaktů [52]	14
2.10	Příklady technických artefaktů [52]	14
4.1	Zobrazení správného vzorkování vstupního signálu [53]	20
4.2	Symetrie DFT pro lichý a sudý počet vzorků [55].	22
4.3	Měření výkonové spektrální hustoty	23
5.1	Matice záměn	27
5.2	Normalizovaná matice záměn	28
7.1	Schéma umělého neuronu	34
7.2	Schéma neuronových sítí[56]	36
8.1	Pravděpodobnosti při špatné klasifikaci. [26]	43
9.1	Schéma realizace BCI systému	44
9.2	Stimulační objekty, jejich markery:ID a frekvence	45
11.1	Matice záměn - 246 epoch	56
11.2	Normalizovaná matice záměn - 246 epoch	57
12.1	Histogram úspěšnosti klasifikace	58

Seznam tabulek

11.1 Tabulka dosažených výsledků 1	55
11.2 Tabulka dosažených výsledků 2	55
11.3 Tabulka dosažených výsledků 3 (\emptyset - znak průměru)	56
11.4 Zpráva o klasifikaci všech subjektů	57

Seznam použitých zkratek

- BCI - Brain-Computer Interfaces (Rozhraní mozek - počítač)
- EEG - Electroencephalography (Elektroencefalografie)
- EP - Evoked Potentials (Evokovaný potenciál)
- VEP - Visual Evoked Potentials (Zrakově evokované potenciály)
- SSVEP - Steady State Visually Evoked Potential (Ustálené zrakově evokované potenciály)
- LDA - Linear Discriminant Analysis (Lineární diskriminační analýza)
- NN - Neural Network (Neuronové sítě)
- MNE - MEG + EEG Analysis Visualization
- BAEP - Sluchové kmenové evokované potenciály
- SEP - Somatosenzorické evokované potenciály
- MEP - Motorické evokované potenciály
- ERP - Endogenní kognitivní evokované potenciály
- f-VEP - Frekvenčně modulovaný VEP
- t-VEP - Časově modulovaný VEP
- c-VEP - Pseudonáhodně modulovaný VEP
- PSD - Power spectral density

Obsah CD

Příložené CD obsahuje následující soubory.

- **rozcestnik.txt** - Struktura souborů na CD
- **GDPR_dokument.pdf** - obsahuje formulář, který musel každý subjekt vyplnit před začátkem experimentu
- **python** - interpret python3.6 + knihovny
- **Program**
 - **data** - složka s data sety
 - **lda.py** - kód LDA klasifikace
 - **main.py** - hlavní kód programu
 - **convolutional_neural_network.py** - kód klasifikace pomocí neuronových sítí
 - **plot_confusion_matrix.py** - kód výpisu confusion matrix
 - **requirements.txt** - požadované knihovny pro spuštění
 - **RUN.cmd** - spouštěcí soubor
 - **trained_model.h5** - natrénovaný model CNN
- **RUN** - zástupce spouštěcího souboru
- **Bakalářská_práce.pdf**
- **SSVEP.zip** - obsahuje kompletní dokument bakalářské práce v LaTeXu

Uživatelská dokumentace

Program je na příloženém CD, které je formátováno jako tzv. *aktivní systém souborů*, takže je možné z CD zkopírovat data do svého PC a poté je pohodlně spustit.

Program se spouští pomocí souboru **RUN.cmd**.

Pokud se při spuštění vyskytnou problémy s knihovnami nebo Pythonem, je potřeba dané náležitosti doinstalovat.

Python 3.6 -> <https://www.python.org/downloads/release/python-368/>

MNE-Python -> https://www.martinos.org/mne/stable/install_mne_python.html

Scikit-learn -> <http://scikit-learn.org/stable/install.html>

Keras -> <https://keras.io/installation>

Tensorflow -> https://www.tensorflow.org/install/install_windows

Program je vytvořen pouze jako konzolová aplikace.

Poté co program spustíme, tak začne automaticky načítat vstupní data ze složky *program\data\train* a *program\data\test* (viz. výpis z konzole).

```
1 C:\A15B0024P>CALL python\Scripts\activate.bat
2 Using TensorFlow backend.
3
4 Number of training datasets: 14
5 Number of testing datasets: 14
6 Training data extraction...
7 Testing data extraction...
8 Training data file 0 - 100 preprocessing...
9 Training data file 100 - 200 preprocessing...
10 Training data file 200 - 300 preprocessing...
11 Training data file 300 - 400 preprocessing...
12 Training data file 400 - 500 preprocessing...
13 Training data file 500 - 600 preprocessing...
14 Training data file 600 - 700 preprocessing...
15 Training data file 700 - 800 preprocessing...
16 Training data file 800 - 900 preprocessing...
17 Number of preprocessed training epochs: 840
18 Testing data file 0 - 100 preprocessing...
19 Testing data file 100 - 200 preprocessing...
20 Testing data file 200 - 300 preprocessing...
21 Number of preprocessed testing epochs: 246
22
23 Choose between three modes.
24 1 = using already trained model to classify testing data sets
25 2 = train new model and then classify testing data sets
26 3 = for-loop across all subjects (14x5) and return average precision (2.5
    hours run)
27 Choose one option: 1 / 2 / 3
```

Můžeme si vybrat z následujících možností:

- **1** - použije se již natrénovaný model a provede se klasifikace dat z testovací části experimentu
- **2** - natrénuje se nový model a provede se klasifikace dat z testovací části experimentu
- **3** - Tato možnost spustí kód běžící v cyklu, kde se každý subjekt klasifikuje 5x a poté se vypíší průměrné hodnoty. Doba běhu je 2,5h.

Zde je zjednodušený příklad výpisu pro možnost číslo 2.

```
1 Choose one option: 1 / 2 / 3
2 > 2
3 Train on 672 samples, validate on 168 samples
4 Epoch 1/500 - 1s - loss: 1.7308 - categorical_accuracy: 0.2485
5 Epoch 2/500 - 0s - loss: 1.6084 - categorical_accuracy: 0.2935
6 Epoch 3/500 - 0s - loss: 1.5241 - categorical_accuracy: 0.3199
7 .
8 .
9 Epoch 500/500 - 0s - loss: 0.7781 - categorical_accuracy: 0.6890
10 Model validation accuracy: 67.86 % Test loss: 0.87
11
12 Subject no.1 - 14
13
14 MN classification report:
15           precision    recall  f1-score   support
16
17      0             0.77       0.77       0.77         66
18      1             0.94       0.71       0.81         72
19      2             0.79       0.72       0.75         72
20      3             0.53       0.89       0.67         36
21
22      accuracy                   0.76         246
23      macro avg                   0.76         246
24      weighted avg                 0.79         246
25
26 Confusion matrix, without normalization
27 [[51  1  7  7]
28  [ 3 51  5 13]
29  [11  1 52  8]
30  [ 1  1  2 32]]
31 Normalized confusion matrix
32 [[0.77 0.02 0.11 0.11]
33  [0.04 0.71 0.07 0.18]
34  [0.15 0.01 0.72 0.11]
35  [0.03 0.03 0.06 0.89]]
36
37 CNN accuracy           : 75.61 %
38 LDA accuracy           : 73.58 %
```