

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program: Aplikovaná informatika

Obor: Kognitivní informatika

Možnosti využití umělých neuronových sítí pro mapování kognitivních funkcí mozku

DIPLOMOVÁ PRÁCE

Student : Bc. Jakub Vitvar

Vedoucí : Mgr. Ing. Miroslav Vacura, Ph.D.

Oponent : doc. PhDr. Ján Pavlík

2019

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 27. června 2019

.....

Bc. Jakub Vitvar

Poděkování

Rád bych na tomto místě poděkoval panu doktoru Miroslavu Vacurovi za trpělivost a ochotu při vedení této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá možnostmi využití umělých neuronových sítí pro mapování kognitivních funkcí mozku. Cílem práce je použít moderní metody a techniky k vytvoření umělé neuronové sítě, která bude schopná rozpoznat, zda se lidský mozek v dané chvíli nachází ve stavu klidu, nebo vykonává nějakou kognitivní funkci. Cíle bude dosaženo pomocí dostupných nástrojů a dat, tak aby tato práce mohla sloužit jako studijní pomůcka a základní přehled při studiu této oblasti kognitivní informatiky. V práci je popsáno, jakým způsobem je možné přistupovat k problematice mapování kognitivních funkcí mozku a jaké technologie jsou v současnosti pro tento účel dostupné. Dále je v práci popsáno, jaká záznamová zařízení pro sledování mozkové aktivity existují a jaká jsou jejich omezení.

Tato práce pracuje primárně s technologií TensorFlow vyvinutou společností Google pro tvorbu hlubokých neuronových sítí a technologií fMRI jako zdrojem dat o mozkové aktivitě. V praktické části práce je názorně ukázáno, jakých výsledků je možné dosáhnout za použití uvedených metod. Výstupem práce je architektura konvoluční neuronové sítě schopná klasifikovat, zda mozek zpracovává vizuální vjem.

Klíčová slova

Umělé neuronové sítě, kognitivní funkce, TensorFlow, Keras, konvoluční neuronové sítě

Abstract

This master's thesis focuses on the use of artificial neural networks to study cognitive functions of the human brain. The goal of the thesis is to use modern techniques and methods to create artificial neural network that would be able to classify whether the human brain is in passive state or whether it performs a cognitive function. The aim is to achieve this goal with only publicly available data and free tools, so that this thesis can be used as study aid or basic overview of this field of cognitive informatics. This thesis covers various approaches that can be used to study cognitive functions of the brain and what technologies are currently available for this purpose. This thesis also covers what kinds of brain activity recording devices exists and what are their limits.

This thesis focuses on technology TensorFlow, developed by Google, to create artificial neural networks and fMRI technology as a source of brain activity data. Practical part of this thesis shows what results can be achieved by using these techniques. The output of this thesis is architecture of convolutional neural network, that is able to classify, whether the brain processes visual stimulus.

Keywords

Artificial neural networks, cognitive functions, TensorFlow, Keras, convolutional neural networks

Obsah

1	Úvod	1
1.1	Cíl Práce	1
1.2	Struktura práce	1
2	Přehled současného stavu výzkumu.....	3
2.1	Rozpoznávání znaků z reálných scén pomocí neuronových sítí	3
2.2	Využití hlubokých neuronových sítí ke klasifikaci motorických činností v fMRI záznamu	3
3	Filozofický problém s explanací funkce lidského mozku	5
3.1	Explanace principu a detailní explanace	5
3.2	Absolutní mez	6
4	Neurofyzilogie mozku	8
4.1	Fyzilogie neuronu	8
4.1.1	Obecný model neuronu	8
4.2	Akční potenciál	9
4.3	Rozpoznávání vjemů z pohledu neurovědy	9
4.4	Kognitivní funkce lidského mozku	10
4.4.1	Vnímání	11
4.4.2	Paměť	12
5	Neurověda a metody zaznamenávání mozkové aktivity	14
5.1	Elektroencefalograf (EEG).....	15
5.2	Magnetoencefalograf (MEG)	15
5.3	Výpočetní tomografie	15
5.4	Pozitronová emisní tomografie.....	16
5.5	Funkční magnetická rezonance (fMRI).....	16
5.5.1	Hemodynamická odezva	17
5.5.2	Zpracování fMRI dat.....	17
6	Umělé neuronové sítě.....	18
6.1	Architektury neuronových sítí	19
6.1.1	Perceptron	19
6.1.2	Vrstvené sítě.....	20
6.1.3	Vrstva s úplným propojením	20
6.1.4	Konvoluční neuronová síť.....	20
6.2	Optimalizace hyper parametrů neuronové sítě	21
6.3	Softwarové implementace neuronových sítí	22
6.3.1	TensorFlow	23

6.3.2	Keras	23
6.3.3	DLTK	24
6.4	Možnosti práce s knihovnou TensorFlow.....	24
7	Postup implementace umělé neuronové sítě.....	26
7.1	Dostupná data	26
7.1.1	Generovaná data	27
7.2	Výběr a příprava trénovacích dat	28
7.2.1	Analýza dat pomocí korelační mapy	28
7.2.2	Finální trénovací data	30
7.3	Implementace neuronové sítě.....	31
7.3.1	Nahrání dat do cloudového prostředí	32
7.3.2	Příprava dat	33
7.3.3	Návrh architektury umělé neuronové sítě.....	33
7.3.4	Hyper parametrizace neuronových sítí	35
8	Vyhodnocení dosažených výsledků	39
8.1	Jednoduchá neuronová síť	39
8.2	Konvoluční neuronová síť	41
8.3	Porovnání výsledků	42
9	Závěr	43
9.1	Experiment	43
9.2	Diskuse	43
	Použitá literatura.....	45

1 Úvod

Na začátku vývoje umělých neuronových sítí stála inspirace v biologii a pokus vytvořit podobnou strukturu jako jsou právě biologické neuronové sítě k řešení komplexních problémů. Tato práce je pokus o aplikaci stejného principu opačným směrem, tedy použití umělé neuronové sítě k popsání toho, co bylo zdrojem původní inspirace, lidského mozku.

Tato práce propojuje všechny znalosti z oboru Kognitivní informatika a aplikuje je na řešení klasifikačního problému. V této práci se snažím pomocí umělé neuronové sítě rozhodnout, zda lidský mozek v daný okamžik vykonává nebo nevykonává kognitivní aktivitu. Tato práce vznikla na základě mé vlastní hypotézy, že s dostatečně přesným snímáním mozkové aktivity a dostatečně velkou neuronovou sítí by mělo být možné popsat všechny aktivity a činnosti, které lidský mozek provádí. I přesto, že práce vznikla na základě čistě neformální úvahy, myslím, že dobře propojuje většinu prvků kognitivní informatiky vyučovaných v tomto studijním oboru a otevírá možnosti pro další zkoumání.

V této práci přistupuji k tomuto úkolu především z pohledu informatiky a nikoli neuropsychologie. Na zdrojová data z funkční magnetické rezonance nahlížím primárně jako na holá data a nepokouším se je vysvětlovat z pohledu lidské fyziologie.

1.1 Cíl Práce

Cílem práce je zhodnotit, zda je pomocí v současnosti dostupných metod a technik možné vyhodnocovat kognitivní činnost mozku pomocí umělé neuronové sítě na základě dostupných dat zaznamenaných jednou z možných technologií. Předpokladem pro splnění tohoto cíle je představení v současnosti dostupných technologií a jejich implementací, zhodnocení dostupných dat a jejich použitelnosti pro tento účel.

Cílem této práce je na základě experimentu posoudit, zda je možné pomocí umělé neuronové sítě rozpoznat, zda lidský mozek v daný okamžik vykonává kognitivní aktivitu. Cílem experimentu není vytvoření umělé neuronové sítě, která bude dosahovat stejných, nebo lepších výsledků jako experti v této oblasti. Cílem práce je pouze zhodnotit, zda je tento způsob výzkumu realizovatelný a připravit tak prostor pro další výzkum v této oblasti.

1.2 Struktura práce

Tato práce je rozdělena do tří hlavních částí. První částí je zhodnocení akademických prací zabývajících se související tematikou a filozofické zamyšlení nad limity explanace funkce

lidského mozku. Cílem první části je vyhodnotit, jakým směrem se vyvíjí výzkum v této oblasti a dále vyhodnotit, jaké jsou limity našeho chápání funkcí lidského mozku z pohledu filozofie. Ve druhé části práce popisují především teoretické základy, ze kterých práce vychází. Cílem druhé části je definovat všechny používané termíny a jasně vymežit možnosti dostupných technologií. Poslední, třetí částí je samotná aplikace umělé neuronové sítě na vybraná data. V této části je popsáno vše, co se týká samotné implementace umělé neuronové sítě. Jsou zde uvedeny všechny potřebné knihovny, hardwarové požadavky a správné konfigurace systému, na kterém je umělá neuronová síť spuštěna.

V rámci práce se řídím normou ČSN ISO 690 a jako citační styl používám tzv. „Harvardský systém“. Seznam použité literatury je řazen abecedně podle jména prvního autora. Hlavním zdrojem literatury pro tuto práci byla doporučená literatura ze souvisejících předmětů oboru kognitivní informatika. Pokročilé vědecké články jsem čerpal z portálu Web of Science (WOS).

2 Přehled současného stavu výzkumu

Cílem této kapitoly je zmínit literaturu, které posloužila jako podnět k vytvoření této práce a zhodnotit její přínos. První prací je diplomová práce zabývající se aplikací neuronových sítí na klasifikaci obrázků a druhou prací je vědecký článek zabývající se výzkumem motorických podnětů pomocí hlubokých neuronových sítí.

2.1 Rozpoznávání znaků z reálných scén pomocí neuronových sítí

První prací, kterou jsem se rozhodl zmínit je „*Rozpoznávání znaků z reálných scén pomocí neuronových sítí*“. Práce Petra Fialy velmi dobře popisuje fungování neuronových sítí a hlubokých neuronových sítí. Při jejich aplikaci se práce zaměřuje na tři datové sety. Prvním z nich je datový set MNIST. Tento datový set se používá jako zdroj dat pro „hello world“¹ aplikace v oblasti rozpoznávání ručně psaného textu. Dále práce popisuje aplikaci neuronových sítí na datové sety ICDAR 2003 a 2011. Zde se opět jedná o rozpoznávání textové informace zachycené v obraze (Fiala, 2014).

Práce velmi dobře popisuje matematické základy fungování neuronových sítí a detaily implementace jednotlivých částí neuronových sítí. Popisuje všechny fundamentální základy, ze kterých umělé neuronové sítě vycházejí a aplikuje je na klasifikační úlohu. Tato práce mi posloužila jako základ, ze kterého vycházím při návrhu vlastních architektur umělých neuronových sítí v praktické části této práce.

2.2 Využití hlubokých neuronových sítí ke klasifikaci motorických činností v fMRI záznamu

Práce „*Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: Evaluation using sensorimotor tasks*“ se zabývá klasifikací motorických činností pomocí hlubokých neuronových sítí na základě fMRI dat. Práce nastavuje metodiku výzkumu v této oblasti, kterou se snažím replikovat v kapitole 6 a 7. Tato práce mi posloužila především jako základ metodiky, kterou jsem rozšířil o využití v cloudovém prostředí Colab (Jang, 2017).

¹ „hello world“ aplikace se používá při představování nového programovacího jazyku jako nejjednodušší možná aplikace. Datový MNIST je vhodný pro tento účel, protože se dá snadno klasifikovat i velmi jednoduchou neuronovou sítí.

Dalším přínosem této práce byl popis aplikace zobecněného lineárního modelu neboli „*general linear model (GLM)*“ na fMRI. Tento postup používám při analýze vstupních dat v kapitole 6.2.1. (Jang, 2017).

3 Filozofický problém s explanací funkce lidského mozku

Úvodem je třeba se věnovat samotným možnostem pochopení funkce lidského mozku z filozofického pohledu. Cílem práce je použít umělé neuronové sítě k rozšíření celkového chápání fungování lidského mozku. Cílem tedy je, i když povrchně, posunout možnosti zkoumání lidského mozku směrem k jeho hlubšímu pochopení. Zde narážím na filozofické omezení, které vychází z práce F. A. Hayeka, především kapitoly „*The limits of explanation*“ z knihy „*The Sensory Order*“, kde autor definuje proces chápání a na jeho základech nastavuje limit toho co je lidský mozek schopen pochopit. Proces „pochopení“ F. A. Hayek definuje jako vytváření modelů, pomocí nichž samotné chápání v mysli vzniká. Podle F. A. Hayeka dochází v mozku k vytváření modelu sledované události a tím událost nabývá významu. Podle F. A. Hayeka tedy existuje v možnostech pochopení fungování lidského mozku absolutní mez, kterou nelze překonat (Pavlík, 2010).

F. A. Hayek rozvíjí ve svém díle „*Řád smyslovosti*“ následující koncepci teoretické psychologie: „*kteřá je metodologicky založena na redukci fenoménu lidské mysli a jejích apriorních funkcí na kauzální řád fyzických událostí v neuronových sítích a která přesto, že je koncipovaná na bázi striktního, žádným heteronomním principem nedoplněného determinismu, dospívá k bezrozpornému tvrzení o neredukovatelnosti fenoménu svobodné vůle.*“ (Pavlík, 2010, s. 71)

Je zde tedy zásadní rozdíl mezi řádem fyzických událostí, tedy toho, co jsme schopni zaznamenávat a zpracovávat, a naším vědomím. I přesto, že naše vědomí z tohoto deterministického řádu vychází, zůstane vždy za hranicí možností našeho vlastního chápání. Tato práce se zabývá pouze výše zmíněným řádem fyzických událostí, které se snaží propojit s dnes již definovanými kognitivními funkcemi.

3.1 Explanace principu a detailní explanace

F. A. Hayek tak rozděluje samotný pojem explanace do dvou úrovní. Za první, explanaci principu, která vysvětluje základní fungování jednoduchých fenoménů a za druhé, úplnou explanaci. V případě, že dosáhneme v nějakém systému úplné explanace, měli bychom být schopni pro každý bod v čase předpovědět, v jakém stavu se bude systém nacházet. V některých případech existují omezení, která se projevují jako praktická mez v dalším zkoumání daného jevu. Tyto praktické meze nejsou konečné a je možné je posouvat,

například pomocí nových technologií nebo získáním nových, předtím neznámých, skutečností. Jako příklad je možné uvést například meteorologii² (Hayek, 1952).

V případě gravitace, počasí a dalších jevů, jsme schopni tento stav do určité budoucnosti aproximovat. Tuto aproximaci můžeme s rozvojem znalostí a technologií stále zlepšovat. Ale existují i systémy, kde podle F. A. Hayeka narazí možnosti lidského mozku na absolutní hranici.

3.2 Absolutní mez

Podle F. A. Hayeka je absolutní hranice dána tím, že jakýkoliv systém (počasí, gravitace, biologický systém, ...) může být vysvětlen pouze jiným systémem (klasifikačním aparátem), který má stupeň komplexity vyšší než systém, který se snaží vysvětlit. Pro lidský mozek (člověka samotného) je tak absolutní hranicí například právě vysvětlení funkce lidského mozku. Tato hranice se nevztahuje pouze na mozek a mysl jako celek, ale také na jeho jednotlivé části. V tom je právě zásadní rozdíl od explanace principu. Pokud bychom chtěli dosáhnout úplné explanace jedné části mozku, například skupiny neuronů, museli bychom vycházet z úplné explanace všech vstupů. Výsledkem tak je to, že pokud nejsme schopni dosáhnout úplné explanace celku, není možné dosáhnout ani úplné explanace části.

Stupeň komplexity klasifikačního aparátu musí být řádově vyšší než stupeň komplexity popisovaného objektu. Stupeň komplexity, tak může například být počet určujících vlastností objektu. V tom případě poté musí být stupeň komplexity klasifikačního aparátu 2^{n+1} , tak aby byl klasifikační aparát schopný klasifikovat jakýkoli objekt, který bude nebo nebude mít některou z daných vlastností (Hayek, 1952).

Přestože budeme i nadále rozšiřovat naše znalosti o fungování lidského mozku z pohledu základních principů pomocí stále vyspělejších technologií, nikdy nedosáhneme úrovně úplné explanace. Paradoxně i přes to F. A. Hayek tvrdí, že není logicky nemožné, že by nám právě znalost principu fungování mozku mohla umožnit vytvoření stroje, který by byl schopný odhadnout stav mozku na základě daných okolností. Jednalo by se pouze o stroj, který by na základě znalosti principu odhadoval budoucí stav. Nejednalo by se tak o dosažení úplné explanace lidským mozkem ani komplexně nadřazeným strojem.

² Meteorologie se zabývá jevem, který je velmi komplexní a obsahuje obrovské množství proměnných. V současné době jsme schopni předpovídat budoucí stav s nějakou přesností, ale nedá se mluvit o detailní explanaci. Současná praktická mez je množství vstupních dat a výpočetní výkon pro jejich zpracování. Tato mez se ale neustále posouvá a předpovědi budoucího stavu se zlepšují.

Pro F. A. Hayeka tak zůstává zásadním právě rozdělení principu fungování lidského mozku, jednotlivých fyzikálních jevů probíhajících mezi neurony, a mentálních jevů probíhajících v lidské mysli. Podporuje tím tak teorii dualismu a rozdělení fyzické a duševní stránky mozku. V opačném případě, tedy v případě úplné explanace mozku, by došlo ke ztrátě tohoto rozdělení a tím i ke ztrátě duševní svobody člověka.

4 Neurofyzologie mozku

Cílem této kapitoly je popsat a definovat základní pojmy týkající se lidského mozku, jeho stavby a jeho fungování. Tato kapitola obsahuje pouze informace důležité pro tuto práci a není možné ji považovat za kompletní přehled.

„Lidský mozek je jedním z nejdůležitějších systémů jak pro život jednotlivce, tak pro lidstvo jako celek. Současně je jednou z nejsložitějších známých soustav vůbec.“ (Novák, 1998, s. 13)

4.1 Fyzilogie neuronu

Neuron je základní funkční jednotkou mozku. Neuron v mozku zajišťuje přenos a zpracování akčního potenciálu při přenosu informace v mozku. Neuron se skládá z dendritů, těla buňky (soma) a axonu. Dendrity jsou vstupním rozhraním neuronu, přebírají vzruch z receptorů, nebo z jiných neuronů. Pro ilustraci komplexnosti sítě, kterou neurony vytváří, je důležité zmínit, že dendritů může být až 10 000 (Novák, 1992, s. 33). V rámci dendritu se vzruch šíří elektrickou energií směrem k tělu neuronu. Tělo neuronu poté na základě vzruchů ze všech dendritů vyhodnotí, zda byl překročen práh excitace. Pokud ano, tak dojde k uvolnění akčního potenciálu, který je veden přes axon k dalším neuronům. Mezi neurony se pak vzruch přenáší chemicky (Friedenberg, 2011).

Stav excitovaných neuronů je možné zaznamenat jako změny elektrického napětí na pokožce pomocí elektrod nebo jako změnu magnetického pole vyvolanou průchodem elektrické energie neuronem. Změnu magnetického pole je možné zaznamenat tzv. SQUID³ senzorem. Dalším sekundárním jevem, který doprovází excitované neurony, je zvýšení množství kyslíku v krvi v okolí neuronu a zvýšení prokrvení oblasti.

4.1.1 Obecný model neuronu

Nejjednodušší interpretací fyziologické funkce neuronu je vytvoření modelu neuronu jako matematického procesoru. Pro základní části neuronu poté můžeme na základě jejich funkcí vytvářet analogie. Funkcí dendritů je zaznamenávat vzruch a přenášet ho do těla

³ SQUID je označení pro supravodivé kvantové interferenční zařízení, které je schopné zaznamenávat biomagnetické signály. Použití samotných SQUIDů je velmi komplikované a vyžaduje několik zásadních opatření jako je například kryogenní zchlazení komponentů, utlumení okolního magnetického šumu a schopnost správně matematicky vyhodnotit nasbíraná data (Sternickel, 2006).

buňky. Jednotlivé dendrity si můžeme představit jako proměnné a přenášený vzruch jako hodnotu. Tělo buňky poté zpracovává vzruch ze všech hodnot najednou. Jako analogii, je tedy možné použít n -rozměrný vektor, který bude sloužit jako vstup pro další funkce, ke kterým dochází v těle buňky. První takovou funkcí je takzvaná agregace, také známá pod pojmem vážená sumace. Tělo buňky na základě přiřazených vah sečte všechny hodnoty zaznamenané dendrity. Jednotlivé váhy jsou analogií na uložené zkušenosti v neuronu. Výsledkem je skalární hodnota, která vstupuje do aktivační funkce. Pokud je hodnota vyšší, než hodnota prahového podnětu dojde k aktivaci neuronu a vyslání výsledného signálu, v případě biologického neuronu by se jednalo o vyslání akčního potenciálu do axonu buňky. Jednou z nejpoužívanějších aktivačních funkcí je aktivační funkce ReLU (Novák, 1998).

Modely neuronu se tak primárně liší především typem aktivační funkce. V praktické části této práce se primárně používají aktivační funkce ReLU a SoftMax.

4.2 Akční potenciál

Akční potenciál neboli vzruch vzniká změnou polarizace na membráně neuronu v dendritech a umožněním výměny nabitých iontů mezi buňkou a jejím okolím. Tímto vznikne z chemické reakce elektrický impuls, který je poté dále veden dendritem k tělu buňky. V případě, že dojde k překročení prahového podnětu, postupuje akční potenciál směrem od těla buňky k axonu (Kymplová, 2008).

Právě odezvu těchto elektrických a chemických impulsů je možné zaznamenávat a dále zpracovávat pomocí metod popsanych v kapitole 5.

4.3 Rozpoznávání vjemů z pohledu neurovědy

V dalších kapitolách bude popisováno, jakým způsobem mozek zpracovává jednotlivé vjemy z pohledu kognitivních modelů. Proto je důležité popsat, jakým způsobem jsou vjemy zpracovávány z pohledu biologických struktur mozku. Toto jsou takzvané jednoduché mozkové funkce. Patří sem například:

- vnímání vnějších podnětů (zrak, sluch, ...),
- polohový smysl,
- vnímání bolesti, a další (Novák, 1998, s. 16).

Vizuální vjemy neboli podněty zaznamenané zrakem vznikají na sítnici a jsou převedeny ze světla na vzruch, který je dále veden očním nervem do primárního vizuálního kortexu v týlním laloku (Mackenzie, 2012, s. 31). Zde dochází k prvnímu zpracování vjemu

a vzruchy jsou dále předávány do dalších oblastí. Zde dochází také k prvnímu rozdělení toku informace na dva proudy. První část se odklání směrem do temenního laloku, kde se zpracovává informace o tom, kde se sledovaný objekt nachází. A druhá část směřuje do spánkového laloku, kde se vyhodnocuje, co je sledovaným objektem (Friedenberg, 2011).

Pro účely této práce je zajímavá i teorie, že lidský mozek zpracovává vizuální vjemy primárně na základě rozpoznání jednotlivých rysů a jednoduchých struktur a až poté na základě těchto informací klasifikuje objekt. Tuto teorii dále potvrzuje i mozková porucha angl. „*apperceptive agnosia*“. Tato porucha se projevuje tak, že pacienti mají funkční zrak, jsou schopni rozpoznávat základní struktury a rysy, ale mají poškozenou část mozku, která tyto signály rozpoznává dále. Pacient tak může vidět objekt, dokáže rozpoznat jeho základní rysy, jako je barva, tvar, velikost, ale nedokáže na základě těchto rysů pojmenovat objekt (Friedenberg, 2011).

4.4 Kognitivní funkce lidského mozku

Na začátku celého studia mozku a myšlení byl takzvaný psychologický přístup. Psychologie byla první věda, která se snažila o systematické pochopení mysli. Jako nová vědní disciplína měla velmi složitý úkol, a to vysvětlit něco, co v době svého začátku nebylo možné dostatečně kvalitně a objektivně zkoumat. To bylo způsobené tím, že neexistovaly metody, jak funkce lidského mozku pozorovat. Z tohoto důvodu byla psychologie ve svém počátku založená především na nepříliš vědeckých postupech, jako je například introspekce⁴. S podobným problémem se potýkal později i behaviorální přístup, který je založen na tom, že mysl je příliš složitá na to, aby ji bylo možné měřit nebo definovat. Proto behaviorismus přistupuje k mysli jako k černé skřínce, a zkoumá pouze stimuly a reakce těmito stimuly vyvolané (Friedenberg, 2011).

Toto se změnilo až s příchodem kognitivního přístupu ke studiu mysli v padesátých letech dvacátého století (Friedenberg, 2011). Kognitivní přístup vznikl na základě tří hlavních podnětů. Prvním byl samotný neúspěch behaviorálního přístupu vysvětlit některé jevy. Druhým důvodem bylo objevení nových způsobů, jak sledovat mozkovou aktivitu. A třetím důvodem byl vznik a rozvoj počítačů, což umožnilo začít mysl přirovnávat k počítači. Tato metafora nasměrovala vědce ke zkoumání jednotlivých funkcí mozku, jako je zpracování vstupů, paměť a další (Friedenberg, 2011).

Kognitivní psychologie se zabývá studiem získávání, zpracování a reprezentace znalostí v mozku. Kognitivní přístup vychází z několika základních předpokladů. Tím

⁴ Introspekci jako metodu zkoumání jako první navrhl anglický vědec John Locke. Cílem bylo na základě sebepozorování dospět k určení elementárních prvků z nichž je možné poskládat celé lidské vědomí (Tvrdý, 2015).

nejdůležitějším předpokladem je, že mysl jedná na základě informací formovaných do tzv. kognitivních map⁵. Na základě těchto map může mysl jednat. Dalším pokrokem oproti předcházejícím metodám studia mysli je, že kognitivní psychologie nepřistupuje k mysli jako k jedné nedělitelné jednotce, ale dělí mysl na jednotlivé procesy myšlení. Toto rozdělení poté snižuje komplexitu sledovaného fenoménu a zjednodušuje další výzkum (Friedenberg, 2011).

Tyto kognitivní procesy je poté možné dále studovat a vytvářet z nich modely, pomocí kterých je možné mapovat jednotlivé kognitivní funkce na anatomickou strukturu mozku. Cílem tohoto postupu je vytvořit co nejpřesnější model, který bude přesně reflektovat biologické fungování mozku (Friedenberg, 2011).

Kognitivní funkce jsou diskrétní procesy myšlení, mozkové aktivity, které jsme schopni definovat. Kognitivní funkce vychází z modulárního přístupu, který je pro kognitivní psychologii také zásadní a je často potvrzován případy pacientů, kteří utrpěli poškození jedné části mozku, která ovlivní pouze jednu kognitivní funkci, zatímco ostatní zůstanou bez následků. Příklady kognitivních funkcí mohou být vnímání, paměť, představivost nebo řešení problémů (Friedenberg, 2011).

4.4.1 Vnímání

První kognitivní funkcí je vnímání, jedná se o získávání informací z okolí pomocí všech smyslů. V případě člověka se ve většině případů hovoří primárně o zraku, jelikož je to pro člověka smysl, kterým získává většinu, okolo 80 % informací. Nejdůležitější funkcí vnímání je rozpoznávání a identifikace objektů. V případě zraku existuje několik základních teorií, jakými způsoby lidský mozek informace zaznamenané okem zpracovává.

První teorií je takzvaná teorie porovnávání šablon, z anglického „*template matching theory*“. Tato teorie říká, že pro každý podnět zaznamenaný okem existuje v mozku šablona, která je mentální reprezentací daného objektu. Mozek poté všechny tyto šablony porovnává se získaným podnětem (Friedenberg, 2011).

Hlavním nedostatkem této teorie je množství šablon, které by bylo potřebné uchovávat v mozku, aby byla lidská mysl schopna rozpoznávat všechny možné objekty, které mohou existovat.

Druhou teorií je teorie rozpoznávání rysů, „*feature detection theory*“. Podle této teorie lidská mysl nezpracovává stimulus jako celek, ale spíše vychází z jednotlivých prvků

⁵ Kognitivní mapa neboli mentální reprezentace znalostí je vytvořena na základě minulých zkušeností a mozek se na její základě může v budoucnosti rozhodovat (Friedenberg, 2011).

a částí, ze kterých se sledovaný objekt skládá. Nejznámější model této teorie je takzvaný „pandemonium model“. Model se skládá z jednotlivých uzlů, z nichž každý detekuje jeden určitý rys. Podle toho, jaké rysy jsou zaznamenány, se potom určí, o jaký objekt se může jednat (Friedenberg, 2011). Tuto teorii podporuje i experiment provedený v roce 1962⁶. Tato teorie se také nejvíce blíží implementaci hluboké konvoluční neuronové sítě popsané v kapitole 5.1.4.

4.4.2 Paměť

Paměť je jednou z lépe definovaných kognitivních funkcí, co se týká popsání zapojených struktur mozku. Existuje několik teorií, jak je možné paměť rozlišovat. Nejčastěji je paměť rozdělována z časového hlediska, a to na krátkodobou, střednědobou a dlouhodobou paměť. Krátkodobá paměť funguje na základě dočasných bioelektrických změn, které procházejí přes thalamus a poté do korových oblastí. Tento proces je zaznamatelný například pomocí EEG. Pokud dojde k fixaci informace, k té dojde asi po 10 sekundách až po několika minutách, jedná se o paměť střednědobou. Fixace je způsobena změnou v propustnosti synapsí. Poslední typem je dlouhodobá paměť, zde se paměťová stopa váže na stabilní neuronální okruhy. Tato paměť je proto mnohem odolnější vůči vnějším vlivům jako jsou například elektrické šoky nebo působení různých farmak. Dlouhodobá paměť je zajišťována anatomickými nebo molekulárními změnami (Pstružina, 1998, s. 101). Paměť je také jednou z nejdůležitějších kognitivních funkcí, závisí na ní schopnost mozku učit se (Friedenberg, 2011).

Typy paměti můžeme charakterizovat podle toho, jak dlouho je informace v paměti dostupná, množství informací, které je možné v paměti držet a podle typu informací, které jsou v paměti obsažené (Friedenberg, 2011).

Prvním typem paměti je senzorická paměť. Tento typ paměti má nejkratší dobu trvání. Senzorická paměť uchovává všechny aktuálně vnímané podněty pouze na dostatečnou dobu k tomu, aby mohly být zpracovány. V případě ikonické paměti, z anglického „*iconic memory*“, je to mezi 250 až 300 ms. Tento interval umožňuje mozku zpracovat i stimuly, které trvaly výrazně kratší dobu, například méně než 100 ms (Friedenberg, 2011).

Dalším typem paměti je krátkodobá paměť. Krátkodobá paměť má omezenou kapacitu, ale paradoxně může mít poměrně dlouhé trvání. Záleží na tom, zda mozek s informacemi

⁶ Cílem experimentu z roku 1962, bylo odhalit jaký stimul je schopný excitovat specifické neurony v týlním laloku kočky. Experiment byl navržen tak, že kočky byly umístěny sondy do centra mozku zpracovávajícího vizuální vjemy a poté byly kočky promítány různé obrazce až do chvíle, než došlo k excitaci sledovaného neuronu. Náhodným testováním bylo objeveno, že sledované neurony excituje zobrazení šikmé přímky ve směru z levého dolního rohu do pravého horního rohu (Hubel, 1962).

v krátkodobé paměti aktivně pracuje. Proto je krátkodobá paměť někdy také označovaná jako pracovní paměť. Krátkodobá paměť slouží k uložení informací, kterým se mozek aktivně věnuje. Například bylo experimentem ověřeno, že pokud dostane subjekt za úkol zapamatování několika málo informací, bude si je pamatovat po celou dobu, co se na ně bude soustředit. Pokud dostane za úkol současně vykonávat další, i třeba jednoduchou aktivitu, klesá schopnost zapamatování si na pouhých 5 % již po 18 vteřinách (Friedenberg, 2011).

Co se týká kapacity krátkodobé paměti, bylo dokázáno, že průměrný člověk dokáže udržet v pracovní paměti 7 plus minus 2 položky (Miller, 1956).

Dlouhodobá paměť slouží k uchovávání informací na delší dobu. Dlouhodobou paměť je možné dělit na procedurální a deklarativní paměť. Procedurální paměť je někdy také nazývaná implicitní paměť. Procedurální paměť obsahuje informace o dovednostech a schopnostech, tyto informace mozek používá bez našeho vědomí. Informace obsažené v této paměti je tak možné demonstrovat pouze vykonáním samotné aktivity (Friedenberg, 2011).

Deklarativní paměť obsahuje nabyté znalosti a zkušenosti. Je možné ji dále dělit na sémantickou paměť a epizodickou paměť. Sémantická paměť obsahuje informace, které jsme získali například ve škole, tedy z nějakého sekundárního zdroje. Epizodická paměť jsou naše zkušenosti a zážitky, které jsme získali osobně interakcí s prostředím (Friedenberg, 2011).

U dlouhodobé paměti se předpokládá, že kapacita není omezujícím faktorem, ale spíše samotné vybavení specifických vzpomínek. Předpokládá se, že je lidský mozek schopný zapamatovat si téměř všechny informace, které zaznamená, ale má pouze omezené možnosti, jak si je posléze vybavit (Friedenberg, 2011).

5 Neurověda a metody zaznamenávání mozkové aktivity

Cílem této kapitoly je popsat základní teorie neurověd a popsat technologie, pomocí nichž je možné monitorovat mozkovou aktivitu. Dalším cílem je definovat a vysvětlit základní pojmy, se kterými se bude pracovat v kapitole 8.

Neurověda se zabývá studiem nervové soustavy, primárně její strukturou a fungováním. Pro výzkum v neurovědě jsou využívány tři hlavní metody. První metodou je výzkum na jedincích, kteří utrpěli poškození mozku s následkem změny v jejich chování. Tímto způsobem je možné vysledovat, jaký vliv mělo specifické trauma na kognitivní funkce subjektu. Druhým způsobem je umělá stimulace částí mozku a sledování dopadu na kognitivní funkce jedince. Třetím způsobem je poté využití speciálních přístrojů pro zaznamenání mozkové aktivity jedince, při konání určité kognitivní funkce (Friedenberg, 2011).

První a druhou metodu zkoumání nebudu dále popisovat, protože neposkytuje data vhodná pro zpracování neuronovou sítí. První metoda není vhodná, protože se poškození mozku u každého jedince většinou liší a není tak možné generalizovat data na ostatní. U druhé metody většinou nevznikají strukturovaná data o mozkové aktivitě subjektu. Výstupem těchto experimentů je většinou pouze záznam o chování subjektu.

Poslední metodou záznamu mozkové aktivity je použití záznamového zařízení, které je schopné zaznamenat mozkovou aktivitu. Tato zařízení je možné rozdělit do skupin podle toho, jak velkou část mozku zaznamenávají nebo jakým způsobem mozkovou aktivitu zaznamenávají.

Nejpřesnější měření je možné získat pomocí přímého sledování aktivity jednoho neuronu nebo blízké skupiny neuronů. Tato metoda výzkumu se používá především u zvířat, kde je možné zavést sondu přímo do mozkové tkáně. Nejznámější studií v této oblasti je experiment provedený na kočce popsáný v kapitole 3.4.1. Kočce byla zavedena elektroda měřící aktivitu jednoho neuronu v oblasti mozku zpracovávající vizuální podněty. Poté byla kočka vystavena podnětům zobrazujícím linky v různých orientacích. Bylo zjištěno, že neuron reaguje na jeden specifický úhel náklonu linky. Tento experiment tak přispívá k potvrzení teorie, že mozek zpracovává vizuální vjemy na základě identifikace jednotlivých částí celku, tedy stejným způsobem, jakým zpracovává obraz hluboká konvoluční neuronová síť. Nevýhodou této techniky je samozřejmě to, že neposkytuje informace o fungování mozku jako celku (Friedenberg, 2011).

V následujících kapitolách jsou popsány technologie, které umožňují sledování mozkové aktivity celého mozku.

5.1 Elektroencefalograf (EEG)

První technologií je elektroencefalograf neboli EEG. EEG zaznamenává přímou elektrickou aktivitu pomocí elektrod umístěných na pokožku hlavy. Výstupem EEG jsou tedy zaznamenané hodnoty elektrického napětí na povrchu pokožky. Tato data jsou velmi hrubá, jelikož sondy zaznamenávají a sčítají hodnoty z celého svého okolí. EEG se používalo primárně pro záznam mozkové aktivity při výzkumu spánkových fází. Posléze bylo používáno i pro studium kognitivních funkcí. Hlavní výhodou je jednoduchost měření a relativně nízká pořizovací cena přístroje oproti ostatním technologiím. Nevýhodou je poté nedostatečná prostorová přesnost způsobená chybějícím prostorovým rozlišením u získaných dat. K dispozici je pouze záznam o umístění sondy. Lokalizace zdroje signálu je dopočítávána následně na základě zachycených dat z okolních sond (Sporns, 2010).

5.2 Magnetoencefalograf (MEG)

Magnetoencefalograf neboli MEG, je funkční neurovizuální metoda, která využívá extrémně citlivé sondy SQUID zaznamenávající změny v magnetickém poli vyvolané přenosem elektrického vzruchu skrze neurony. SQUID je označení pro supravodivé kvantové interferenční zařízení, které je schopné zaznamenávat biomagnetické signály. Použití samotných SQUIDů je velmi komplikované a vyžaduje několik zásadních opatření jako je například kryogenní zchlazení komponentů, utlumení okolního magnetického šumu a schopnost správně matematicky vyhodnotit nasbíraná data (Sternickel, 2006).

Data získaná touto záznamovou metodou jsou k dispozici pouze ve velmi omezeném množství.

5.3 Výpočetní tomografie

Výpočetní tomografie je nejstarší vizuální metodou, která je schopna poskytnout informace o anatomické struktuře a omezeně i funkční struktuře mozku. Výpočetní tomografie neboli CT, se skládá z lůžka pro pacienta a rentgenu, který obíhá pacienta. Je tak pořízeno množství rentgenových snímků ze všech úhlů, které jsou poté spojeny do jednoho 3D zobrazení. Hlavním nedostatkem CT je tedy omezená možnost sledování funkční aktivity mozku a používá se především pro výzkum zaměřený na biologickou strukturu mozku (Friedenberg, 2011).

5.4 Pozitronová emisní tomografie

Pozitronová emisní tomografie neboli PET, byla vyvinuta po CT. Pomocí PET je možné sledovat aktivitu mozku pomocí sledování množství krve v oblasti aktivních neuronů. K tomuto účelu jsou používány radioaktivní izotopy, které jsou injekčně vpraveny do oblasti mozku, kterou sledujeme. Takto „označená“ krev poté putuje do oblasti aktivity, kde je zaznamenána pomocí zařízení, které je schopné zachytit pozitrony vyzařované z radioaktivních izotopů. PET sken má velkou výhodu oproti CT hlavně díky, přesnému záznamu aktivních oblastí mozku. Bohužel hlavní nevýhodou je potřeba vystavení mozku radioaktivním látkám, což dělá tuto metodu nevhodnou pro dlouhodobé studie (Friedenberg, 2011).

5.5 Funkční magnetická rezonance (fMRI)

Nejvhodnější metodou pro záznam mozkové aktivity je pro účely této práce funkční magnetická rezonance. Stejně jako PET je fMRI schopná zaznamenat průtok krve v mozku, ale navíc je schopna zaznamenat i hladinu kyslíku v krvi. Zkoumaný subjekt je vystaven velmi silným magnetům, což má za následek, že pozitivně nabyté subatomární částice, pozitrony, jsou orientovány podle magnetického pole. Poté je subjekt vystaven radiovým signálům. Rádiové signály se odrazí a jsou zaznamenány. Na základě odražených signálů je možné určit, jakého typu byly atomy, od kterých se signál odrazil a je možné vytvořit 3D obraz (Friedenberg, 2011).

fMRI tak kombinuje výhodu technologie PET bez toho, aby bylo nutné používat radioaktivní látky. Má tak velmi dobré prostorové rozlišení a je schopná zaznamenat mozkovou aktivitu pomocí měření průtoku krve a množství kyslíku v krvi.

fMRI má dvě varianty, perfúzní fMRI a BOLD. Perfúzní fMRI měří průtok a množství krve v oblasti aktivních neuronů a BOLD měří množství kyslíku. Vzhledem k tomu, že se jedná o nepřímé monitorování mozkové aktivity a fMRI nezaznamenává přímo elektrickou aktivitu mozku, ale její nepřímý obraz je potřeba počítat s několika omezeními. Prvním z nich je takzvaná „*hemodynamic response function*“ neboli HRF, toto je odezva mezi mozkovou aktivitou a samotným zachycením změny magnetického pole pomocí fMRI. To je způsobeno především opožděnou reakcí vaskulárního systému mozku. Dalším je relativně malé temporální rozlišení oproti ostatním metodám, například EEG a MEG mají temporální rozlišení až 1 ms, fMRI dokáže zaznamenávat hodnoty s rozlišením okolo 2 sekund, což může být pro některé případy nedostatečné. Naopak výhodou je vyšší prostorové rozlišení (Liao, 2002).

5.5.1 Hemodynamická odezva

Hemodynamická odezva, z anglického „*hemodynamic response function*“, je reakce mozku na neuronální aktivitu. Hemodynamická odezva začíná ihned po neuronální aktivitě. V oblasti, kde dojde k aktivitě, dojde ke zvýšení spotřeby kyslíku což má za následek krátkodobý pokles síly signálu, který je způsobený rychlou deoxygenací na začátku hemodynamické reakce. Poté dojde k zvýšení objemu a průtoku krve v oblasti, čímž se zvedne poměr mezi množstvím okysličené a neokysličené krve a výrazně stoupne zachycený signál. K maximálnímu poměru mezi okysličenou a neokysličenou krví, a tedy maximálním zachyceným signálem, dojde mezi 6 až 9 sekundami po stimulu. Po skončení stimulu dojde ke snížení neuronální aktivity a postupně se oblast vrátí do normálního stavu. Návrat do normálního stavu může trvat 8 až 20 sekund (Funkční magnetická rezonance, 2008).

Vzhledem k tomu, že při použití metody fMRI nedochází k měření primární mozkové aktivity, ale pouze sekundárního projevu, kterým je zvýšení množství okysličené krve v okolí aktivních neuronů, dochází k opoždění zaznamenané aktivity vůči samotné mozkové aktivitě. Tato odchylka se pohybuje okolo 6 sekund (Liao, 2002).

5.5.2 Zpracování fMRI dat

Většina dostupných fMRI dat je k dispozici ve formátu NIfTI-2 Tento formát byl založen Národním institutem mentálního zdraví „*National Institute of Mental Health*“ a Národním institutem pro výzkum neurologických poruch a infarktu „*National Institute of Neurological Disorders and Stroke*“. Tento formát byl zaveden cca před deseti lety a nahrazuje původní formát Analyze. Zásadním nedostatkem starého formátu byl nedostatek metadat obsažených v samotném souboru. Například chyběla informace o orientaci skenu. NIfTI formát obsahuje v jednom souboru jak data (body), tak metadata (header).

Formát NIfTI-2 je možné zobrazovat pomocí programu ImageJ (<https://imagej.nih.gov/ij/>) a pro strojové zpracování je možné použít Python knihovnu Nibabel, dostupné na <https://nipy.org/nibabel/>.

Pro každý jednotlivý experiment je vytvořen samostatný datový soubor s koncovkou .nii, který je rozdělený na hlavičku a tělo. Hlavička obsahuje metainformace o záznamovém zařízení, době záznamu, minimální a maximální zaznamenané hodnotě atd. Tělo obsahuje samotná data o mozkové aktivitě subjektu v daném experimentu. Jedná se o 4-dimenzionální datové pole. Toto pole je možné převést na Numpy Array a dále zpracovávat například pomocí knihovny Numpy dostupné na <http://www.numpy.org/>.

6 Umělé neuronové sítě

Umělé neuronové sítě je možné rozdělit na dvě základní skupiny. První skupinou jsou umělé neuronové sítě biologického typu. Tyto sítě slouží k modelování reálných biologických systémů za účelem jejich dalšího zkoumání. Druhým typem jsou aplikační neuronové sítě, pro tyto sítě není primárním cílem podobnost s biologickou předlohou, ale především jejich funkce a schopnost řešení problémů (Novák, 1998). V této práci se budu zabývat pouze aplikačními neuronovými sítěmi.

Umělé neuronové sítě jsou nástroj strojového učení, který slouží především v oblastech, kde neexistuje ani přibližný algoritmus řešení nějakého problému, nebo kde je obtížné tento algoritmus matematicky formalizovat. Další vhodnou situací pro aplikaci umělých neuronových sítí je chvíle, kdy nemáme k dispozici přesná a úplná data nebo se jedná o složité signály a vícerozměrná data (Novák, 1998). Analýza zaznamenané mozkové aktivity tak splňuje všechny tyto kategorie.

„Neuronová síť je paralelní distribuovaný systém výkonných prvků modelujících biologické neurony, účelně uspořádaných tak, aby byl schopen požadovaného zpracování informací.“ (Novák, 1998, s. 144)

Umělá neuronová síť může být reprezentována orientovaným grafem. Výkonné prvky umělé neuronové sítě, umělé neurony, jsou reprezentovány uzly grafu. Umělé neurony jsou tvořeny aktivační funkcí, která zajišťuje stejnou funkci jako excitační potenciál v biologickém neuronu. Výkonné prvky jsou rozdělené na disjunktní podmnožiny zvané vrstvy (Novák, 1998).

Hrany grafu poté znázorňují jednotlivé jednosměrné spoje mezi umělými neurony. Každý umělý neuron může mít libovolný počet vstupních a výstupních spojů. Spoje mohou být laterální, tedy spoje mezi neurony v rámci jedné vrstvy. Přímovazební spoje spojují neurony mezi dvěma vrstvami ve směru od vstupu k výstupu. Tento typ spojů je nejčastější. Posledním typem spojů jsou zpětnovazební spoje. Tento typ spojů se využívá například v rekurentních neuronových sítích, které slouží především pro zpracování sekvenčních dat. Zpětnovazební spoj předává výstup jedné vrstvy jako vstup některé z předchozích vrstev. Neuronová síť je tak schopná zachytit vzor v sekvenci signálů. (Novák, 1998) Typickým příkladem rekurentní neuronové sítě je zpracování mluveného slova, tzv. „*natural language processing*“.

Umělé neuronové sítě můžeme dělit na základě jejich architektury⁷. Architektura neuronové sítě popisuje, kolik neuronů síť obsahuje, do kolika vrstev jsou neurony uspořádány a jakého jsou vrstvy typu.

⁷ Pojem architektura neuronové sítě je popsán v kapitole 6.1.

Další možností, jak dělit umělé neuronové sítě je na základě jejich softwarové implementace, zde by bylo možné uvést jako příklady: TensorFlow, Theano nebo CNTK⁸. Softwarová implementace má vliv především na to, jakým způsobem jsou prováděné operace na nejnižší úrovni. Softwarová implementace nemá vliv na konečnou umělou neuronovou síť a její fungování.

6.1 Architektury neuronových sítí

Pro účely této práce budu používat pojem architektura neuronových sítí pro popsání vnitřního uspořádání jednotlivých vrstev sítě a jejich velikosti, například se může jednat o vstupní vrstvu, konvoluční vrstvu a další. V oblasti umělých neuronových sítí se pro tento pojem používají i synonyma topologie sítě, geometrie sítě nebo struktura sítě (Novák, 1998).

Právě podle architektury neuronových sítí se umělé neuronové sítě liší a získávají speciální vlastnosti vhodné pro řešení specifických problémů. Příkladem může být konvoluční vrstva pro identifikaci objektů v obrazu nebo rekurentní vrstvy pro analýzu mluveného slova.

6.1.1 Perceptron

Nejjednodušším modelem neuronové sítě je jedno-neuronový perceptron. Tento neuron může mít několik vstupů. Pokud bychom například perceptrony použili pro zkoumání objektu v prostoru, mohly by vstupy být například údaje o rozměrech, barvě, teplotě atd. Každá tato hodnota musí být ve formě reálného čísla. Dále má každý vstup své vlastní ohodnocení neboli váhu, opět ve formě reálného čísla, která udává důležitost daného vstupu. Neuron je poté ohodnocen prahem. Vážené vstupy a práh neuronu je poté vstupem pro přenosovou funkci, která rozhodne o výstupu (Mařík, 1993).

Takto navržený perceptron je tedy schopný klasifikovat jednoduché vstupy do dvou tříd. Pro pokročilé možnosti klasifikace je potom nutné použít komplexnější architekturu neuronové sítě.

⁸ CNTK neboli „*Microsoft Cognitive Toolkit*“ je implementace neuronové sítě vyvinutá společností Microsoft.

6.1.2 Vrstvené sítě

Pojem vrstvené sítě označuje umělé neuronové sítě, jejichž neurony jsou uspořádané do vrstev. Vrstvené sítě se tak skládají ze vstupní vrstvy, jedné nebo více skrytých vrstev a jedné výstupní vrstvy. Hlavní výhodou vrstvených sítí je schopnost pojmut větší komplexitu zpracovávaného problému (Mařík, 1993). Vrstvené neuronové sítě uspořádané do tří a více vrstev jsou označovány jako hluboké neuronové sítě, tzv. „*deep neural networks*“.

6.1.3 Vrstva s úplným propojením

Pojem vrstva s úplným propojením⁹ označuje vrstvy neuronů, které jsou propojené se všemi neurony následující vrstvy (Novák, 1998). Vrstvy s úplným propojením se používají jako defaultní vrstvy pro konstrukci jednoduchých neuronových sítí. Vrstvy s úplným propojením budou hlavním stavebním kamenem umělé neuronové sítě navržené v kapitole 7.3.3. Jejich hlavní výhodou je právě univerzálnost. Dále se používají například jako poslední vrstvy v konvoluční neuronové síti.

6.1.4 Konvoluční neuronová síť

Název konvoluční neuronová síť vychází z anglického slova „*convolve*“, neboli vinout se. Hlavním cílem konvoluční vrstvy neuronové sítě je hledání vzorů ve vstupních více dimenzionálních datech. Základem konvoluční vrstvy je konvoluční jádro, z angl. „*convolution kernel*“, které se postupně posouvá po vstupní vrstvě a hledá ve vstupních datech vzory.

Hlavními parametry při vytváření konvoluční vrstvy je počet filtrů, velikost konvolučního jádra a velikost a tvar vstupních dat. Počet filtrů určuje, kolik různých typů vzoru bude výsledná neuronová síť schopná zachytit. Velikost konvolučního jádra poté určuje, v jak velké části vstupních dat se budou vzory hledat. Velikost a tvar slouží pouze pro inicializaci vrstvy a používá se pouze u konvoluční vrstvy, která je umístěna jako první v celé umělé neuronové síti (Keras, 2019).

Konvoluční neuronová síť vznikla na základě inspirace v přírodě. Snaží se imitovat způsob, jakým zpracovává informace lidský mozek a to tak, že jako první hledá hrubé rysy a z nich následně skládá složitější struktury. Konvoluční vrstvy umístěné na začátku neuronové sítě jsou schopné zachytit především hrubé vzory a konvoluční vrstvy umístěné hlouběji v neuronové síti zachycují detailnější struktury. Primárně se tedy konvoluční

⁹ Kromě pojmu úplné propojení se používá i pojem bohaté propojení, zdroj Mařík, 1993.
V rámci této práce budu používat pojem úplné propojení, zdroj Novák, 1998.

neuronová síť používá při klasifikaci fotografií a identifikaci objektů ve fotografiích, je ale možné ji využít téměř kdekoli.

Samotná konvoluční neuronová síť se skládá z konvoluční vrstvy a tzv. „*subsampling*“ vrstvy. Konvoluční vrstva pomocí filtru hledá vzory a jejím výstupem je ohodnocení pro každou pozici konvolučního jádra. „*Subsampling*“ vrstva poté vybere pouze největší hodnotu pro každou oblast. Jako subsamplingová vrstva se nejčastěji používá „*MaxPooling*“ vrstva, ale „*AveragePooling*“, „*GlobalAverage*“, a další. Konvoluční vrstvy mohou existovat i pro více než 2-dimenzionální pole. Knihovna Keras je omezená na maximálně 3-dimenzionální pole.

6.2 Optimalizace hyper parametrů neuronové sítě

Kromě nastavení architektury umělé neuronové sítě a jejího trénování je dalším důležitým krokem optimalizace hyper parametrů neuronové sítě. Jako hyper parametry jsou označovány všechny hodnoty, které je možné u neuronové sítě nastavovat, které nejsou nastavovány v průběhu samotného učení. Mezi hyper parametry je možné zařadit rychlost učení¹⁰, počet samotných vrstev, počet neuronů ve vrstvách, počet konvolučních filtrů, parametry aktivačních funkcí a mnoho dalších.

Základními nástroji pro optimalizaci hyper parametrů jsou dvě hodnoty. První hodnotou je přesnost, z anglického „*accuracy*“. Přesnost určuje pravděpodobnost, s jakou je neuronová síť schopná klasifikovat vstupy. V průběhu učení se přesnost počítá jak z trénovacích dat, tak i z testovacích dat. Cílem je dosáhnout maximální hodnoty přesnosti na testovacích datech, tzv. „*validation accuracy*“. Druhou hodnotou je ztráta, ztráta je výstupem tzv. ztrátové funkce, z anglického „*loss function*“. Ztrátová funkce určuje, jak konzistentní je predikce neuronové sítě. Cílem je ztrátu minimalizovat. Stejně jako v předchozím případě se ztráta počítá v průběhu učení a počítá se jak pro trénovací data, tak pro testovací data (Vasilev, 2019).

Existují dva základní způsoby, jak k optimalizaci hyper parametrů přistupovat. Prvním z nich je manuální přístup a druhým je automatický přístup. První přístup spočívá v postupném trénování a upravování hodnot hyper parametrů na základě průběžné kontroly hodnot „*validation loss*“ a „*validation accuracy*“. K tomuto účelu je možné využít například zobrazovací nástroje průběhu učení jako je knihovna TensorBoard¹¹. Tato metoda je velmi časově náročná a vyžaduje i značnou znalost práce s neuronovými sítěmi i

¹⁰ Z anglického „*learning rate*“, zdroj překladu: Autor.

¹¹ TensorBoard je doplňková knihovna ke knihovně TensorFlow, umožňuje v reálném čase zobrazování vývoje trénování modelu v prohlížeči.

samotného zkoumaného fenoménu, protože spočívá převážně v intuitivním nastavování hodnot na základě zkušeností.

Druhým přístupem je automatizovaný způsob optimalizace. Principem automatizovaného přístupu je ke každému hyper parametru vytvořit rozsah možných hodnot a poté vytvořit algoritmus, který bude testovat kombinace všech hyper parametrů a bude hledat optimální hodnoty. Pro příklad je možné uvést tři různé implementace automatické optimalizace. Prvním způsobem je takzvaný „*grid search*“, neboli vyhledávání v mřížce¹². U každého hyper parametru se rozsahu možných hodnot zvolí pravidelně rozložené hodnoty, které se následně testují každá s každou, jedná se tak o „*brute force*“ přístup k optimalizaci hyper parametrů. Dalším přístupem je hledat optimální hyper parametry náhodně v rámci možných hodnot, tato možnost sdílí stejné nedostatky jako v případě „*brute force*“ a se zvyšujícím se počtem hyper parametrů se šance na nalezení optimálních hodnot výrazně snižuje. Posledním způsobem je využít sofistikovanějšího přístupu k optimalizaci jako je například Bayesovská optimalizace, implementovaná v rámci knihovny Scikit-Optimize¹³.

Co mají všechny automatizované přístupy společné je vysoká náročnost na výpočetní výkon, protože k efektivnímu nalezení optimálních hodnot je potřeba trénovat několik modelů najednou, což není náročné pouze na rychlost, ale také na velikost dostupné operační paměti. Z tohoto důvodu jsem se pro tuto práci rozhodl používat manuální přístup.

6.3 Softwarové implementace neuronových sítí

Všechny v současnosti používané softwarové implementace umělých neuronových sítí jsou postaveny na stejném matematickém základu. Liší se tedy pouze tím, jak byly tyto fundamentální základy naprogramovány a jaké programovací jazyky k tomu byly použity.

Mezi prvními implementacemi umělých neuronových sítí byly perceptronové architektury ADALINE a MADALINE, tyto jednoduché neuronové sítě byly použity v řadě praktických aplikací, které se v některých případech používají dodnes. Po tomto rychlém rozvoji došlo k útlumu a znovu se začaly implementace neuronových sítí objevovat až v 80. letech. V této době se danou oblastí zabývalo velké množství vědeckých organizací a vznikalo velké množství samostatných proprietárních implementací (Novák, 1998).

Trendem v posledních letech je poté vytváření open-source knihoven a abstraktních rozhraní, které usnadňují práci s neuronovými sítěmi a zrychlují jejich aplikaci.

¹² Zdroj překladu: Autor.

¹³ Scikit-optimize je Python knihovna, která slouží k optimalizaci modelů použitím Bayesovské optimalizace.

6.3.1 TensorFlow

TensorFlow je systém vyvinutý americkou společností Google s primárním účelem zjednodušit, zrychlit a zefektivnit vývoj algoritmů strojového učení a práci umělými neuronovými sítěmi. TensorFlow je, stejně jako například knihovna Theano¹⁴, nízkoúrovňová knihovna primárně zaměřená na tensor¹⁵ a manipulaci s nimi. Název TensorFlow vznikl spojením slov „tensor“, a „flow“, to se dá interpretovat například jako datový tok. TensorFlow vzniklo jako softwarová knihovna pro řešení komplikovaných výpočtů, které je možné díky interpretaci pomocí tensorů a grafů řešit na distribuovaných systémech. TensorFlow je v současnosti používáno v mnoha komerčních produktech společnosti Google a je i k dispozici jako open source software (Abadi, 2016).

6.3.2 Keras

S knihovnou TensorFlow je možné pracovat přímo nebo je možné využít nadstavby, která zjednodušuje některé úkony. Jednou z takových nadstaveb je například knihovna Keras. Keras je ve skutečnosti API¹⁶ naprogramované v programovacím jazyce Python. Hlavním cílem knihovny Keras je zjednodušit použití TensorFlow, a ještě více zrychlit návrh a implementaci architektur umělých neuronových sítí.

Jednoduchost knihovny Keras je možné ukázat například na základních vrstvách umělých neuronových sítí jako jsou například: Dense, Flatten, Dropout, a další. Všechny tyto vrstvy jsou v rámci knihovny předpřipravené a stačí je do modelu přidat pomocí metody: „.add()“. Vrstva Dense je například úplně propojená vrstva, to znamená že všechny neurony této vrstvy jsou propojené se všemi neurony následující vrstvy. Pro přidání této vrstvy do modelu poté stačí pouze předat počet neuronů ve vrstvě jako jediný parametr metody. Vrstva Flatten slouží k převedení více dimenzionálního vektoru na

¹⁴ Theano je open-source knihovna pro manipulaci s daty a strojové učení, optimalizovaná pro CUDA a v omezené míře i OpenCL knihovny. Samozřejmostí je podpora pro CPU. V oblasti umělých neuronových sítí zastává stejné místo jako TensorFlow (Al-Rfou, 2016).

¹⁵ Tensor je v tomto kontextu multidimenzionální pole, obsahující data. Tak jako může být veličina určena jediným údajem, například teplota (v tomto případě se jedná o skalár), může být veličina určena i více údaji, například síla (v tomto případě se jedná o vektor). Veličiny, které jsou určeny více údaji jsou tensor¹⁵ x-tého řádu. Tensor je tedy zobecněním všech těchto možností. Tensory, tak umožňují zaznamenat stav čehokoli a dávají tak TensorFlow velkou flexibilitu s možností používat všechny základní matematické operace (Ryšavý, 1944).

¹⁶ Zkratka API, z anglického „*application programming interface*“, označuje sbírku procedur nebo protokolů, které je možné využít při integraci dvou různých systémů.

1-dimenzionální vektor. Používá se především jako vstupní vrstva nebo jako přechod mezi konvoluční vrstvou a úplně propojenou vrstvou. Dropout vrstva je speciální typ vrstvy, tato vrstva neslouží k učení modelu, ale spíše naopak. Dropout vrstva přepíše určité procento všech výstupů předchozí vrstvy na nulu, slouží tak jako ochrana proti přeučení. Vzhledem k tomu, že smazané výstupy se pokaždé mění je méně pravděpodobné, že v rámci celé neuronové sítě dojde k přeučení (Keras, 2019).

6.3.3 DLTK

Vedle univerzálních knihoven jako je Keras, Pylearn2 a další, existuje i speciální open-source knihovna DLTK, „*Deep learning tool kit*“ zaměřená především na zpracování lékařských snímků především pomocí hlubokých neuronových sítí. Tato knihovna je postavena na již zmíněné knihovně TensorFlow. DLTK vychází ze stejných poznatků, které pomohly v poslední době rychlému rozvoji v oblasti klasifikace obrazů a strojového učení a modifikuje je pro použití v lékařské oblasti. Především se jedná o použití konvolučních sítí. Zjednodušuje tak použití strojového učení na lékařských datech a zrychluje další výzkum. DLTK pomáhá především v oblastech jako je čtení dat a jejich předzpracování, definice modelu a architektury neuronové sítě a trénování a optimalizaci neuronových sítí (Pawlowski, 2017).

6.4 Možnosti práce s knihovnou TensorFlow

Všechny výše zmíněné systémy je možné aplikovat v různých prostředích v závislosti na způsobu práce s umělou neuronovou sítí. První možností je práce v nativním Python prostředí na vlastním počítači. Hlavním požadavkem je v tomto případě dostatečný výpočetní výkon, nejčastěji v podobě grafické karty podporující technologii CUDA¹⁷. To jsou v současné době pouze grafické karty od společnosti NVidia¹⁸. Kromě grafické karty je možné použít pro trénování neuronové sítě i CPU, které je oproti grafické kartě výrazně pomalejší. Například v porovnání v cloudovém prostředí Colab je grafická karta 45krát rychlejší než CPU (TensorFlow with GPU, 2019).

Práce s neuronovými sítěmi na lokálním počítači je výhodnější pouze v případě, že máte k dispozici výkonnější GPU, než je k dispozici zdarma v prostředí Colab (Tesla K80).

¹⁷ Technologie CUDA, z anglického „compute unified device architecture“, umožňuje na vybraných GPU zpracovávat vysoce paralelizované algoritmy (Cuda, 2019).

¹⁸ Karty od konkurenční společnosti AMD je možné použít pouze v kombinaci s platformou ROCm. Tuto možnost jsem netestoval.

Jako nejvhodnější způsob práce s knihovnou TensorFlow jsem vybral cloudové prostředí Colab od společnosti Google. Nejenže cloudové prostředí nabízí vyšší kapacitu operační paměti, ale je také výrazně jednodušší na konfiguraci.

7 Postup implementace umělé neuronové sítě

V této části diplomové práce aplikuji teoretické poznatky z první části diplomové práce na reálná data s cílem vytvořit architekturu umělé neuronové sítě pro klasifikaci kognitivní funkce mozku. Zdrojovými daty pro umělou neuronovou síť je 4-dimenzionální datové pole a výstupem je klasifikace pro každý jednotlivý záznam.

Prvním krokem je výběr trénovacích dat a ověření, že mají vypovídající hodnotu a obsahují informaci o aktivitě vizuálního kortexu. Pro jednoduché zobrazení dat je možné využít software ImageJ, pro složitější zobrazení je možné použít Python knihovny Nibabel a Matplotlib. Dalším krokem je příprava vstupních dat, tak aby je byla neuronová síť schopná zpracovávat. Vzhledem k tomu, že se jedná o více dimenzionální datová pole je nejjednodušší způsob zpracování opět v prostředí Python s využitím knihovny Numpy.

Následujícím krokem je pomocí knihovny Keras navrhnout základní architekturu neuronové sítě. Především se tedy jedná o to, jaké vrstvy budou v síti použité. Cílem tohoto kroku je získat síť, která bude poskytovat ne úplně primitivní závěry. Za primitivní závěr považuji například síť, která se vždy bude rozhodovat pro jednu hodnotu bez ohledu na vstup.

Posledním krokem je optimalizace hyper parametrů. Na začátku této fáze už je připravená síť, ale nedosahuje dostatečně kvalitních výsledků. Cílem této fáze je najít takové hodnoty hyper parametrů se kterými budou neuronové sítě dosahovat přesnosti alespoň 80 %¹⁹.

Celkem aplikuji na data dvě různé architektury neuronové sítě s cílem co nejlépe popsat jejich výhody a nevýhody pro tento způsob použití a v závěru vybrat nejvhodnější architekturu.

7.1 Dostupná data

Všechna data zpracovávaná v této diplomové práci jsou získaná z portálu OpenNeuro.org a jsou pod licencí Creative Commons, CC-BY. Autoři studií, ze kterých data pochází, jsou uvedeni v seznamu citací.

Pro zpracování této diplomové práce se zaměřím na data zachycená technologií fMRI. Především proto, že tato technologie je nejrozšířenější, existuje tak nejvíce dostupných dat,

¹⁹ Okolo 12 % zaznamenaných dat se nachází v přechodném stavu, kvůli vlivu hemodynamické odezvy. Vzhledem k tomu, že navržené neuronové sítě nepracují s posloupností dat pravděpodobně nebudou schopné tyto stavy správně klasifikovat.

technologie poskytuje nejlepší prostorové rozlišení a s daty se velmi snadno pracuje při dalším zpracování.

Na portálu OpenNeuro.org je možné stáhnout celé datové sety, na kterých jsou postavené různé experimenty. Pro mé účely není podstatný obsah a výsledek daného experimentu, ale pouze získaná data, protože cílem práce je vytvořit neuronovou síť schopnou data klasifikovat bez aplikace dalších znalostí z oblasti neurologie.

Každý datový set obsahuje v rámci dostupných dat informace o účastnících experimentu, subjektech, popis průběhu experimentu, popis záznamového zařízení a samotná data. Data v datovém setu jsou rozdělená podle subjektů a dále podle jednotlivých záznamů. Na každém subjektu bylo provedeno několik záznamů, během kterých subjekt vykonával aktivitu. Jeden záznam je poté v datovém setu uložen jako jeden soubor. Každý soubor obsahuje 4-dimenzionální datové pole obsahující sérii snímků s temporálním rozlišením závislejícím na možnostech záznamového zařízení.

Cílem této kapitoly je vybrat takové experimenty, které mají velmi jasně definovaný průběh experimentu a trvání alespoň 3 minuty. Jasně definovaný průběh pomáhá při vytváření trénovacích štítků²⁰, z anglického „*training labels*“, při vytváření vstupů pro umělé neuronové sítě. Větší délka experimentu poté zjednodušuje práci s přípravou dat a poskytuje větší konzistentnost dat.

Na základě těchto parametrů jsem vybral dvě studie. První datový set pochází ze studie „*Task Dependence, Tissue Specificity, and Spatial Distribution of Widespread Activations in Large Single-Subject Functional MRI Datasets at 7T*“. Jedná se o experiment provedený na třech subjektech (dvě ženy a jeden muž, věk=25+/-2,0). Každý subjekt podstupuje záznam pomocí technologie fMRI po dobu 320 sekund, během nichž vykonává mentální aktivitu v pěti 10sekundových úsecích (Gonzalez-Castillo, 2002). Druhý datový set pochází ze studie „*The neural representation of personally familiar and unfamiliar faces in the distributed system for face perception*“. V této studii je vybráno 33 subjektů (20 žen a 13 mužů, věk=23+/-3,3). Mozková aktivita každého subjektu byla zaznamenávána po dobu 316 sekund. Subjekt řešil jednoduchý úkol na základě klasifikace, zda se jedná o fotografie jednoho z jejich přátel, fotografie neznámé osoby, jeho vlastní fotografie, nebo zda mu byl ukázán soubor fotografií různých osob (Di Oleggio Castello, 2017).

7.1.1 Generovaná data

Jednou z možností, jak otestovat navrženou neuronovou síť je použít generovaná data. Tím by bylo možné zajistit, že trénovací data jsou správná a bez odchylek způsobených hemodynamickou odezvou. Ke generování umělých fMRI dat by bylo možné použít fMRI simulátor (fMRI simulátor, 2014). Pro práci s tímto generátorem neexistuje žádný manuál

²⁰ Zdroj překladu Mařík, 1993.

ani dokumentace. V případě potřeby by tak bylo nutné vytvořit vlastní generátor. Vzhledem k tomu, že se v kapitole 6.3 podařilo navrhnout neuronové sítě, které byly schopné rozpoznat aktivní a pasivní stavy i přes vliv hemodynamické odezvy, rozhodl jsem se vlastní generátor dat nepoužít.

7.2 Výběr a příprava trénovacích dat

Dvě vybrané studie velmi dobře reprezentují dostupné datové sety z pohledu metodiky experimentu, struktury zveřejněných dat a formátu dat. Z tohoto pohledu jsou všechny zveřejněné datové sety velmi homogenní. Velké rozdíly jsou v průběhu samotného experimentu, tedy aktivity, kterou provádí sledovaný subjekt.

Mým cílem bylo vybrat takové experimenty, kde sledovaný subjekt nebo subjekty vykonávají definovanou mentální aktivitu v několika úsecích oddělených úsekem odpočinku, kdy je subjekt vystaven minimu senzorických vjemů. Důvodem bylo získat data s dostatečně jednoduchou strukturou, kterou bude možné jednoduše klasifikovat za účelem vytvoření trénovacích dat. Oba vybrané datové sety tyto požadavky splňují. Zásadním rozdílem je délka jednotlivých úseků.

První datový set obsahuje pět aktivních úseků trvajících 10 sekund, oddělených čtyřmi odpočinkovými úseky trvajících 20 sekund. Druhý datový set obsahuje aktivní úseky dlouhé 1,5 sekundy, oddělené 4,5vteřinovými odpočinkovými úseky.

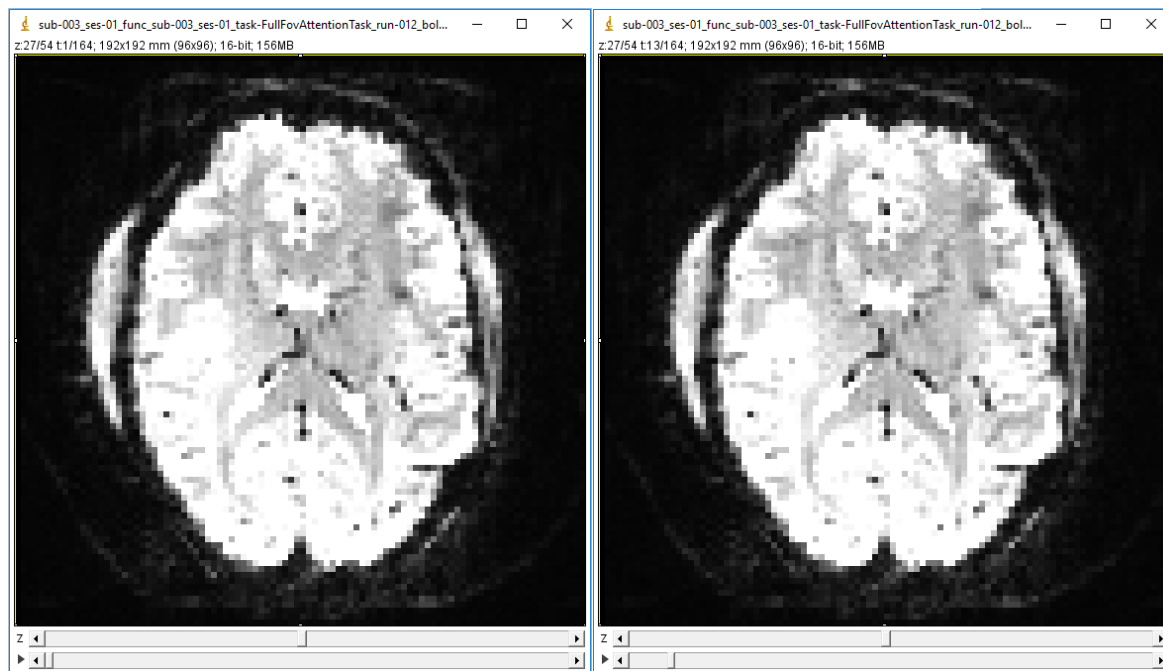
Vzhledem k průběhu hemodynamické odezvy jsem se rozhodl dále zpracovávat pouze první datový set. Průběh experimentu prvního datového setu poskytuje větší prostor pro návrat mozku do klidového stavu a poskytne tak vhodnější data pro zpracování umělou neuronovou sítí, protože bude větší rozdíl v zaznamenaných datech mezi aktivní a odpočinkovou fází.

7.2.1 Analýza dat pomocí korelační mapy

Prvním krokem, který je potřeba provést je zanalyzovat, zda data opravdu obsahují rozdíly mezi aktivní a odpočinkovou fází záznamu. Pro tento účel jsem se rozhodl použít metodu korelační mapy. Cílem je ověřit, zda vybraná data obsahují informaci o mentální aktivitě, kterou by mohla neuronová síť rozpoznat.

Pokud data tuto aktivitu obsahují, měla by existovat korelace mezi BOLD aktivitou a vizuálním stimulem. Tato aktivita by potom měla být zachycena, jako vyšší hodnoty korelačního koeficientu v oblasti vizuálního kortexu mozku, tedy v týlní části hlavy. (Mirko Novák, 1998)

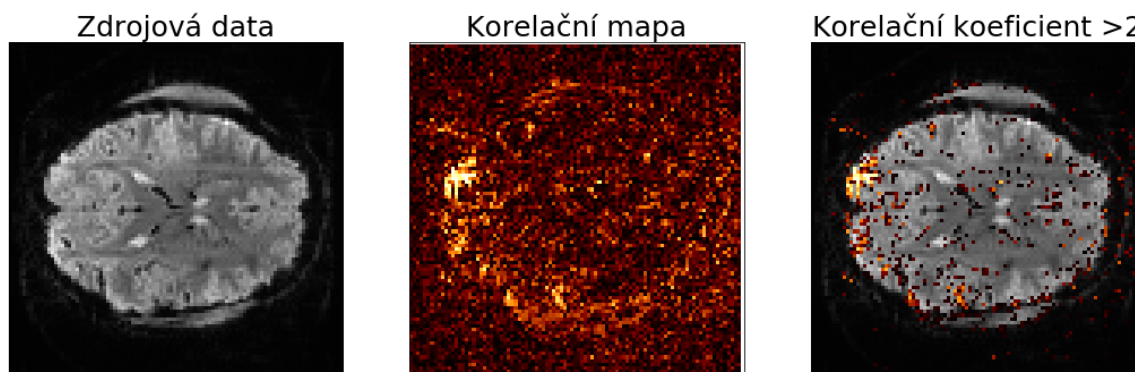
Prvním krokem je výběr dat, která budu testovat. Pro ověření správnosti dat je nejvhodnější transversální řez v oblasti okcipitálního laloku, ve kterém se nachází vizuální kortex. Pro výběr řezu je možné využít například program ImageJ, pomocí kterého je možné procházet fMRI data. Na obrázku č. 1 je zobrazen výstup programu ImageJ, transversální řez č. 27, první záznam. Na obrázku je možné rozeznat základní anatomické struktury, vizuální kortex se nachází v dolní části obrázku. Dále je na obrázku vidět, že celý mozek vykazuje výraznou BOLD aktivitu, ale není na první pohled vidět rozdíl mezi klidovou, vlevo a aktivní fází, vpravo.



Obrázek 1 – Porovnání dat z klidové fáze, vlevo, a aktivní fáze, vpravo. Transverzální řez č.27. Zobrazeno pomocí ImageJ. (Zdroj: Autor, 2019).

Dalším krokem je vytvoření datového pole, které bude odpovídat stavům, ve kterých se testovaný subjekt nacházel, tedy zda byl nebo nebyl subjekt vystaven vizuálními stimuly. Pole tedy bude obsahovat pro každý záznam jednu hodnotu, která bude nabývat hodnoty 1 nebo 0 podle toho, zda subjekt byl nebo nebyl vystaven vizuálnímu stimulu. Nyní je možné provést korelační analýzu mezi zachycenými fMRI daty a informací o stimulu. Posledním krokem je vizualizace oblastí, kde byla zachycena korelace na fMRI záznamu, tak aby bylo možné potvrdit nebo vyvrátit že došlo k zachycení aktivity v oblasti vizuálního kortexu (Jang, 2017).

Výstupem analýzy je obrázek č. 2., na obrázku je jasně patrná aktivita v oblasti vizuálního kortexu. Dostatečně sofistikovaná neuronová síť by tak měla být schopná detekovat tento typ kognitivní aktivity.

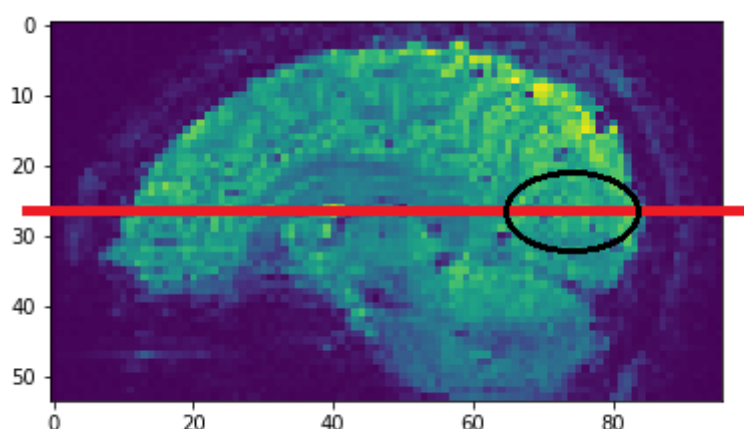


Obrázek 2 – Výstup korelační analýzy. (Zdroj: Autor, 2019).

7.2.2 Finální trénovací data

Jako finální trénovací data jsem vybral subjekt 3 ze studie „*Task Dependence, Tissue Specificity, and Spatial Distribution of Widespread Activations in Large Single-Subject Functional MRI Datasets at 7T*“. Tento subjekt byl vystaven vizuálnímu podnětu s úkolem rozpoznávat zobrazená písmena a číslce. Tento subjekt byl vystaven celkem 12 testům, kde každý test obsahoval okolo 10 záznamů. Jeden záznam poté obsahuje 164 snímků. Celkem je tedy k dispozici až 20 000 trénovacích dat (snímků).

Jako tréninková data jsem zvolil transversální řez 27, na obrázku č. 3, označený červenou horizontální linkou. Vizuální kortex je označený černou elipsou. Na těchto datech jsem v kapitole 7.2.1 potvrdil, že obsahují informaci o tom, že subjekt vykonával mentální aktivitu.



Obrázek 3 – Vizualizace transversálního řezu mozku, který je použitý jako vstupní data pro zpracování umělou neuronovou sítí. (Zdroj: Autor, 2019).

Při hledání optimálního množství dat jsem testoval tři různé sady:

1. jeden záznam, tedy 164 řezů,
2. jeden experiment, celkem přibližně 2 100 řezů,
3. všechny data subjektu č. 3, celkem 16 072 řezů.

Každou sadu jsem testoval jak jednoduchou (úplně propojené vrstvy), tak i komplexní (konvoluční vrstvy) neuronovou sítí. V prvním případě ani jedna neuronová síť nebyla schopna poskytnout vypovídající výsledky na testovacích datech. Vždy došlo k přeučení nebo neuronová síť vracela primitivní závěry. V druhém případě byly oba typy sítí schopné klasifikovat stav subjektu. V posledním případě nebyla jednoduchá síť schopná klasifikovat stav subjektu, respektive vždy vracela pouze primitivní závěry. Komplexní síť dosáhla přesnosti 69 %, ale vzhledem k rychlosti učení nebylo možné síť efektivně optimalizovat.

Pro finální implementaci jsem se tak rozhodl použít data pouze z jednoho experimentu. Výhodou je, že díky větší rychlosti učení je možné provést větší množství iterací při optimalizaci hyper parametrů a zároveň jsou data více konzistentní, protože se jedná o jeden experiment, a nedošlo tak ke změnám při zavádění experimentu, jako může být například změna polohy hlavy nebo jiný mentální stav subjektu.

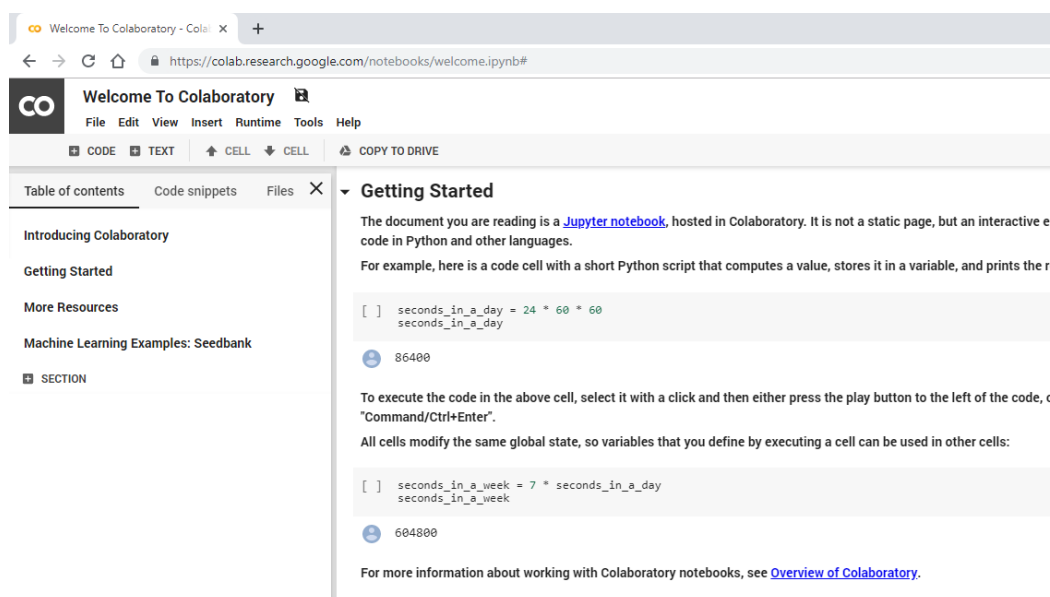
7.3 Implementace neuronové sítě

Pro samotnou implementaci jsem se rozhodl využít cloudového prostředí Google Colab. Hlavní výhody tohoto prostředí jsou popsány v kapitole 5.4. Cloudové prostředí je dostupné na <https://colab.research.google.com/>. K jeho využívání je potřebný pouze účet Google.

Cloudové prostředí Colab je založené na open source softwaru Jupyter notebook. Jupyter notebook slouží k jednoduché práci především v programovacím jazyku Python. Jupyter umožňuje sdílení kódu v rámci jednoho týmu, podporuje integraci s Big data databázemi a mnoho dalšího.

V rámci prostředí Colab se s kódem pracuje v takzvaných buňkách, z anglického „cell“. Tyto buňky je poté možné jednotlivě spouštět což je velmi výhodné právě při optimalizaci neuronových sítí, kdy není potřeba spouštět celý kód znovu, ale stačí spustit pouze upravenou část.

V příloze A je ukázka implementace finální architektury konvoluční sítě, po jednotlivých buňkách, tak jak byla síť psaná v prostředí Colab.



Obrázek 4 - Ukázka prostředí Colab. (Zdroj: Autor, 2019).

Okamžitě po otevření portálu je nutné založit nový Python sešit, k dispozici jsou obě verze, 2.7 i 3.6. Takto je cloudové prostředí připravené pro standardní práci v prostředí Python. Pro práci s hardwarovým akcelerátorem, GPU, je potřeba vybrat možnost GPU v menu „Edit“ a „Notebook settings“. Poté je možné se připojit do cloudového rozhraní pomocí tlačítka „Connect“ v pravém horním rohu. V tuto chvíli už je možné se sešitem pracovat jako s klasickým Python 3 prostředím, včetně hardwarového akcelerátoru.

Pro práci s knihovnou TensorFlow je potřeba importovat následující knihovny:

```
from __future__ import absolute_import, division, print_function, unicode_literals
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import nibabel as nib
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
```

7.3.1 Nahrání dat do cloudového prostředí

Jednou z výhod cloudového prostředí je, že není nutné stahovat trénovací data do lokálního úložiště, ale je možné data stahovat rovnou do prostředí Google Colab. Data je možné stáhnout pomocí příkazu:

```
!wget https://openneuro.org/crn/...
```

Cloudové úložiště je omezené na 350 GB, což je pro účely této práce dostačující.

7.3.2 Příprava dat

Po stažení dat do cloudového prostředí je nutné data otevřít pomocí knihovny Nibabel a převést na datový objekt Numpy Array.

```
data = nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-002_bold.nii.gz')
data = data.get_fdata()
```

Dále je potřeba data rozdělit na dvě sady, a to trénovací a testovací data. Trénovací data budou použita pro trénování umělé neuronové sítě a podle testovacích dat se bude určovat výsledná přesnost natrénovaného modelu.

Posledním krokem je normalizace dat. Cílem je, aby všechny hodnoty byly v intervalu mezi 0 a 1 včetně krajních hodnot. To je díky práci s knihovnou Numpy velmi snadné, stačí vydělit všechny hodnoty v poli data maximální hodnotou v poli.

```
data = data / np.amax(data)
```

Vstupní data, se kterými budu nadále pracovat se skládají z trénovacích obrazů a trénovacích štítků. Trénovací obrazy jsou v podobě 3-dimenzionálního pole, které bude obsahovat 1 804 snímků s rozlišením 96 x 96 pixelů. Každý pixel může nabývat hodnoty 0 až 1. Dále budu pracovat s trénovacími štítky, což bude jednodimenzionální pole, které bude obsahovat 1 804 hodnot, které mohou být 0 nebo 1. Hodnota 0 pro pasivní stav a hodnota 1 pro aktivní stav.

7.3.3 Návrh architektury umělé neuronové sítě

V této práci budu testovat dvě různé neuronové sítě, které se liší svou komplexitou. Jako první budu testovat jednoduchou neuronovou síť se dvěma skrytými vrstvami a poté komplexnější architekturu neuronové sítě, která bude využívat i konvoluční vrstvy.

První neuronová síť se bude skládat ze vstupní vrstvy, dvou skrytých vrstev a jedné výstupní vrstvy. Jako vstupní vrstva je použita vrstva „*Flatten*“, která převede 2-dimenzionální pole na vektor. Dále následuje jedna úplně propojená vrstva s 8 neurony a aktivační funkcí ReLU.

Poslední vrstvou je jednoneuronová vrstva s aktivační funkcí Sigmoid. Tato neuronová síť dosahuje přesnosti 70 % což je těsně nad hranicí primitivního závěru.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(96, 96)),
    keras.layers.Dense(8, activation=tf.nn.relu),
    keras.layers.Dense(1, activation=tf.nn.sigmoid)
])

model.compile(optimizer='adam',
```



```
loss='binary_crossentropy',  
metrics=['accuracy'])
```

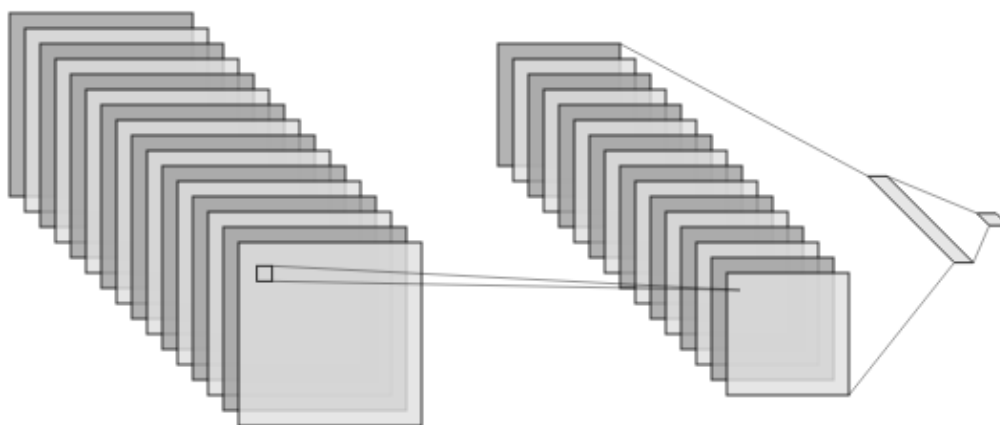
Základem druhé umělé neuronové sítě jsou konvoluční vrstvy. Cílem této umělé neuronové sítě bylo navrhnout takovou architekturu, která bude dosahovat co nejlepších výsledků a ve stejné chvíli ji bude možné efektivně trénovat na standardním stolním počítači nebo v bezplatném cloudovém prostředí. Dalším důvodem pro výběr konvoluční neuronové sítě byla její schopnost zaznamenávat vzory v 2-dimenzionálním poli. Tato vrstva by tak měla být schopná zachytit aktivované skupiny neuronů, stejným způsobem jako je schopná zachytit obrazce, hrany nebo rohy, při rozpoznávání objektů na fotografiích.

Nejjednodušší možná konvoluční neuronová síť musí obsahovat alespoň jednu konvoluční vrstvu, jednu subsamplingovou vrstvu a jednu Flatten vrstvu. Tato jednoduchá konvoluční síť by byla vhodná například ke klasifikaci psaných číslic. Jako příklad může sloužit například dataset MNIST²¹. Ukázka inicializace této jednoduché konvoluční neuronové sítě pomocí knihovny Keras:

```
model = Sequential()  
model.add(Conv2D(32, (3, 3), padding='same', input_shape=train_images.shape[1:]))  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Flatten())  
model.add(Dense(64))  
model.add(Activation('relu'))  
model.add(Dense(1))  
model.add(Activation('sigmoid'))  
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])  
model.fit(train_images, train_labels, epochs=50,  
        class_weight=class_weight, validation_split=0.3)
```

Vzhledem k tomu, že trénovací data v rámci této práce jsou poměrně složitější, bude potřeba komplexnější neuronová síť. Při návrhu architektury konvoluční neuronové sítě jsem začal s nejjednodušší možnou konvoluční sítí a postupně zvyšoval komplexitu až do chvíle, kdy neuronová síť začala vracet správné výsledky.

²¹ MNIST dataset je součástí samotné Keras knihovny, jedná se o sadu 60 000 obrázků ručně psaných číslic v rozlišení 28 x 28 pixelů. Tato sada se používá jako „Hello World“ aplikace Keras knihovny (Keras, 2019).



Obrázek 5 – Vizualizace architektury jednoduché konvoluční sítě. Vrstvy zleva: Konvoluční síť s 32 filtry o velikosti 3x3 pixely, MaxPool vrstva o velikosti 2x2 pixely, úplně propojená vrstva o velikosti 64 neuronů a výstupní vrstva o velikosti jednoho neuronu (Zdroj: Autor, 2019).

Tato jednoduchá konvoluční neuronová síť dosáhla přesnosti 69 % na testovacích datech. Základem architektury je konvoluční vrstva s 32 filtry a velikostí konvolučního jádra 2x2 pixely s ReLU aktivační funkcí a úplně propojená vrstva s 64 neurony s ReLU aktivační funkcí. Výstupní vrstvou je jeden neuron s aktivační funkcí sigmoid a optimalizační algoritmus je typu adam s defaultními hodnotami.

7.3.4 Hyper parametrizace neuronových sítí

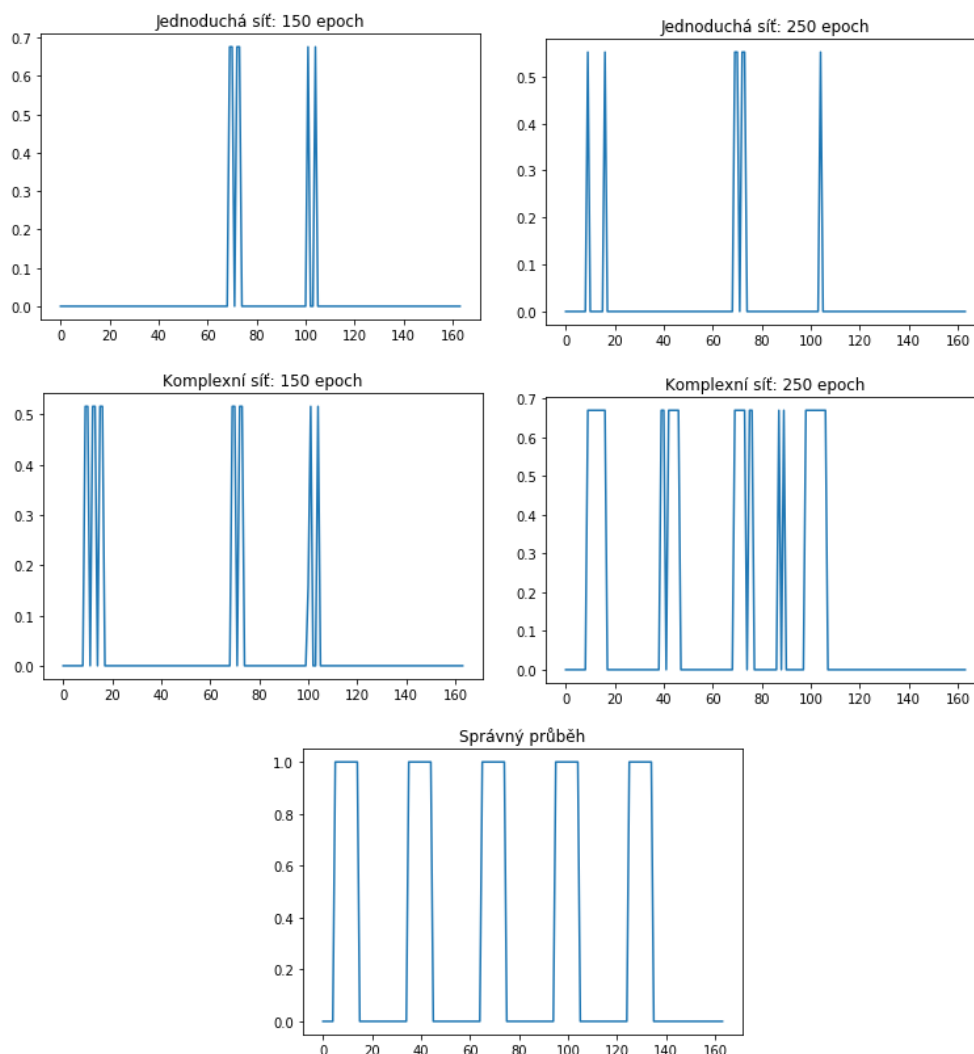
Vzhledem k jednoduchosti první neuronové sítě jsem se rozhodl využít automatického hledání optimálních parametrů. Jako nejdůležitější parametry jsem zvolil počet úplně propojených neuronových vrstev a počet neuronů v těchto vrstvách. Tyto proměnné jsem potom pomocí jednoduchého skriptu otestoval, zatímco jsem souběžně zaznamenával průběh učení mezi epochami. Vzhledem k tomu, že knihovna TensorBoard, která by umožňovala záznam učení neumožňuje automatickou optimalizaci, vytvořil jsem vlastní záznam průběhu v rámci optimalizačního skriptu.

Výsledkem tohoto testu byla následující optimální architektura:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(96, 96)),
    keras.layers.Dense(64, activation=tf.nn.relu),
    keras.layers.Dense(64, activation=tf.nn.relu),
    keras.layers.Dense(1, activation=tf.nn.sigmoid)
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Celkem bylo testováno 80 kombinací různých počtů plně propojených vrstev a množství neuronů. Bylo testováno 1 až 10 skrytých vrstev a 4, 8, 16, 32, 64, 128, 256, 512 neuronů. Počet epoch neměl výrazný vliv na učení, bez ohledu na složitost neuronové sítě vracela síť primitivní závěry do cca 100 epochy, poté došlo k učení a zlepšování výsledků na testovacích datech až do cca 250 epochy, kdy začalo docházet k přeučení.

Na obrázku č. 6 je vidět rozdíl mezi první navrženou architekturou a architekturou po optimalizaci. Velmi důležitým faktorem u architektury využívající pouze úplně propojené vrstvy je počet epoch. Hlavní nevýhodou je poté velká náchylnost na přeučení.



Obrázek 6 – V horní části obrázku je porovnání přesnosti neuronové sítě po 150 a po 250 epochách. Cílem je přiblížit se správnému průběhu zobrazenému v dolní části. (Zdroj: Autor, 2019).

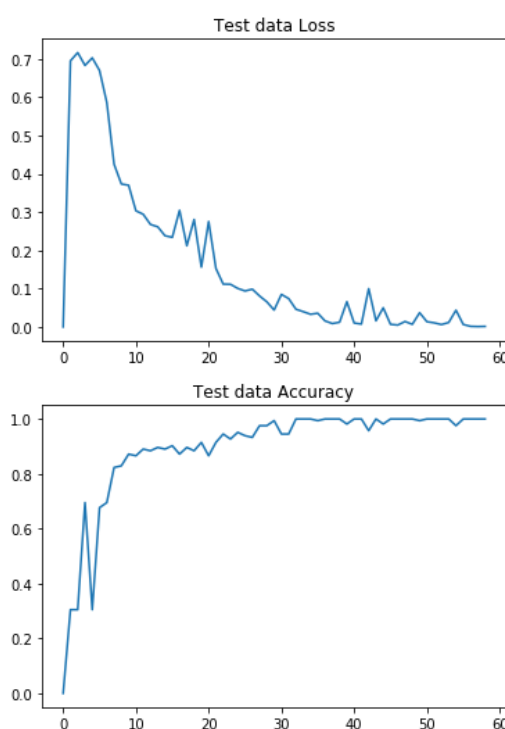
V případě konvoluční sítě jsem nejlepších výsledků dosáhl optimalizací následujících hyper parametrů:

- počet epoch,
- počet konvolučních vrstev,

- typ výstupní vrstvy.

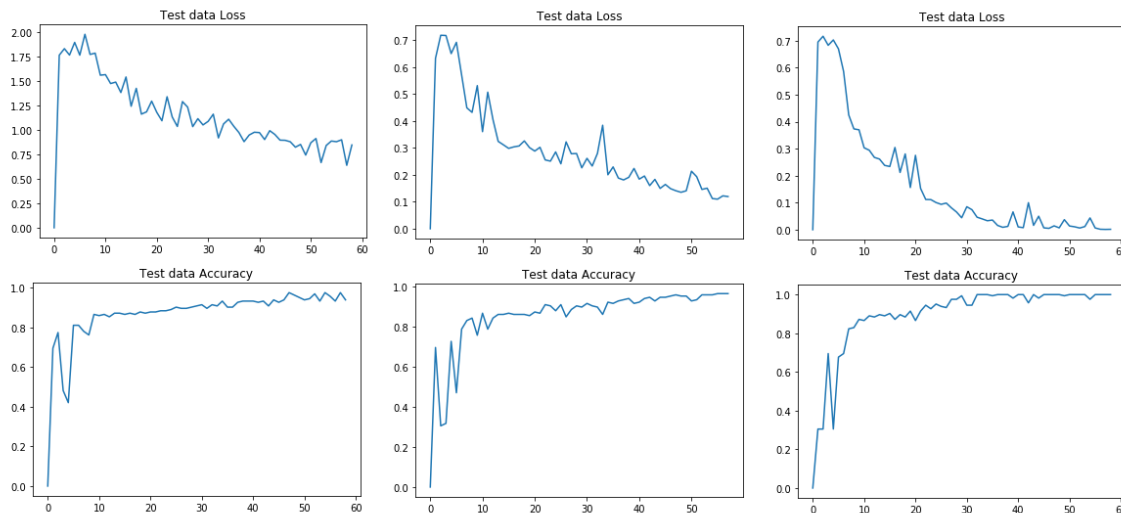
Určování vhodného počtu epoch je nejjednodušší hyper parametr k optimalizaci. Podstatou optimalizace je hledání momentu, kdy je minimální hodnota „loss“ a maximální „validation accuracy“. Hledá se tak poslední možný moment, kdy dochází k učení. Při překročení této hodnoty začne docházet k přeučení, to má za následek výrazné zvýšení přesnosti na trénovacích datech, zvýšení hodnoty „loss“ a snížení přesnosti na testovacích datech. Dochází k tomu, že síť se učí i vlastnosti trénovacích dat, která nejsou aplikovatelná na ostatní, resp. testovací, data.

Komplikací v tomto postupu je to, že změna, kteréhokoli jiného hyper parametru může mít na počet epoch vliv, proto je nutné tento postup často opakovat. Na obrázku č. 8 je vidět vývoj hodnot „loss“ a „accuracy“ v konečné verzi architektury konvoluční neuronové sítě.



Obrázek 7 – Vývoj hodnot loss, horní graf, a accuracy, dolní graf. Na ose x je znázorněn počet průběhů učení, kde každý průběh obsahoval deset epoch (Zdroj: Autor, 2019).

Počet konvolučních vrstev a jejich parametry určují schopnost neuronové sítě zaznamenávat vzory. Čím vyšší počet konvolučních vrstev a více filtrů tím rychleji probíhá učení, současně s tím se neuronová síť stává náchylnější na přeučení. Na obrázku č. 9 je vidět rozdíl mezi třemi neuronovými sítěmi, které se liší počtem konvolučních vrstev a počtem konvolučních filtrů.



Obrázek 8 – Zleva: 1. nejjednodušší konvoluční neuronová síť, 2 konvoluční vrstvy se 4 filtry. 2. Středně složitá konvoluční síť, 3 konvoluční vrstvy s 8 filtry. 3. finální verze konvoluční neuronové sítě, 3 konvoluční vrstvy s 32 filtry (Zdroj: Autor, 2019).

Dalším zajímavým faktorem je typ výstupní vrstvy. V tomto případě je cílem klasifikace do dvou tříd. Ideální výstupní vrstva by tak měla být jednoneuronová vrstva s aktivační funkcí Sigmoid. Nejlepšího výsledku během optimalizace se mi ale podařilo dosáhnout s desetineuronovou vrstvou s aktivační funkcí Softmax.

Nejlepší dosažený výsledek během optimalizace hyper parametrů byl 85% úspěšnost při testování na testovacích datech, při 70 trénovacích epochách.

Výsledná architektura konvoluční neuronové sítě:

```
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same',
                 input_shape=train_images.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(10))
model.add(Activation('softmax'))

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

8 Vyhodnocení dosažených výsledků

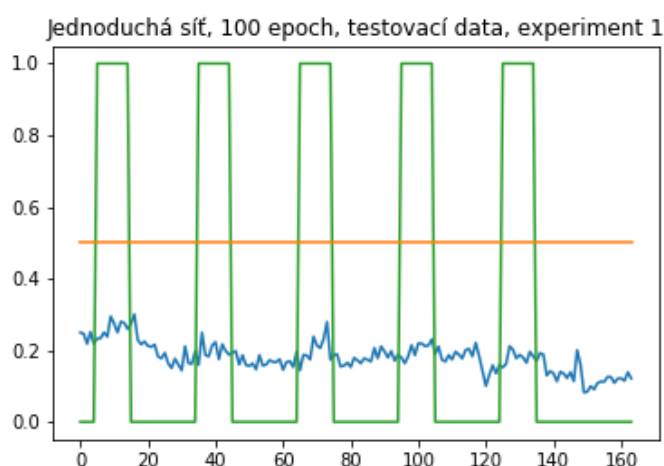
Cílem této kapitoly je zhodnotit dosažené výsledky obou navržených neuronových sítí a porovnat výsledky jednoduché umělé neuronové sítě využívající pouze úplně propojené vrstvy a komplexní umělé neuronové sítě využívající konvoluční vrstvy.

8.1 Jednoduchá neuronová síť

Jednoduchá neuronová síť dosáhla nejlepších výsledků okolo 80 % na trénovacích datech a okolo 75 % na testovacích datech. Těchto výsledků dosahovala jednoduchá neuronová síť při počtu 128 neuronů ve skrytých vrstvách při 250 trénovacích epochách.

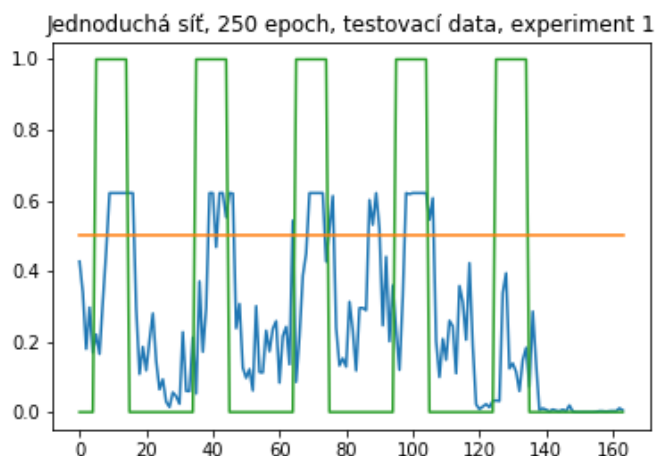
Hlavním problémem, se kterým se tato neuronová síť potýkala, bylo přeučení. Tato síť se byla schopná naučit klasifikovat oba mentální stavy, ale měla problém s aplikací tohoto modelu na ostatní data. V průběhu optimalizace hyper parametrů sice dosáhla přesnosti 75 % na testovacích datech, při pokusu aplikovat model na data z jiného experimentu se úspěšnost snížila na hranici primitivních závěrů. Dalším problémem byla hemodynamická odezva. Tedy fyziologické omezení, kdy se samotná aktivita mozku v fMRI datech zobrazuje s postupným nástupem.

Na obrázku č. 10 je vidět, že po 100 trénovacích epochách nedokáže jednoduchá neuronová síť klasifikovat zpracování vizuálního stimulu.



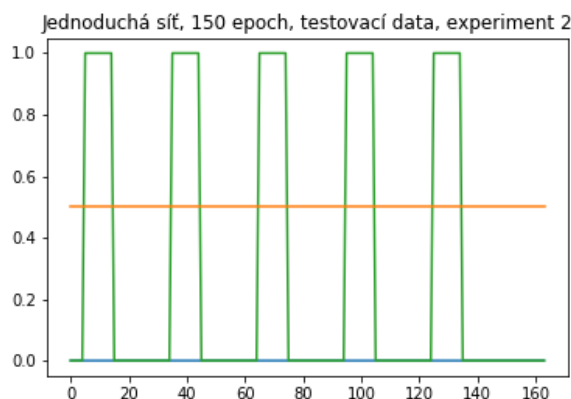
Obrázek 9 – Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).

Na obrázku č. 10 je poté vidět výstup neuronové sítě po 250 trénovacích epochách. Zde je vidět, že neuronová síť stále nezachytila první záznam aktivní fáze, ale poté už rozpoznávala stav rychleji a přesněji. Problematické oblasti jsou tak právě přechodné stavy.



Obrázek 10 - Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).

Na obrázku č. 11 je vidět, že jednoduchá neuronová síť, není schopná najít jakékoli vzory, které by byli aplikovatelné i na data mimo trénovací experiment.



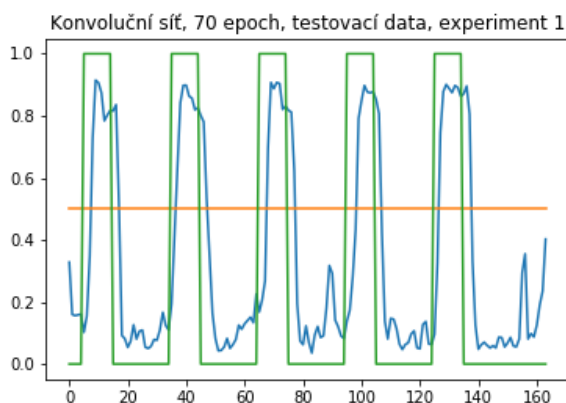
Obrázek 11 – Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).

8.2 Konvoluční neuronová síť

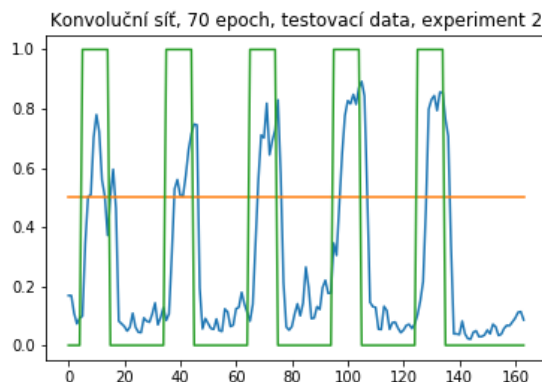
Optimalizovaná konvoluční neuronová síť dosáhla přesnosti 82 % na testovacích datech ze stejného experimentu a přesnosti 75 % na testovacích datech z jiného experimentu, potvrdila se tak schopnost zaznamenat i fyziologické změny, které opravdu provází danou kognitivní funkci.

Na obrázcích č. 12 a č. 13 je vidět klasifikace testovacích dat pomocí naučeného modelu. Zeleně je označený čas, kdy reálně docházelo k vizuálnímu stimulu. Modře je znázorněna jistota, s jakou neuronová síť klasifikuje, zda dochází k vizuálnímu stimulu. V porovnání obou obrázků je vidět, že sice model není dokonalý, protože došlo k snížení přesnosti na druhé testovací sadě i tak byl model schopný rozpoznat všechny aktivní fáze.

Dále je na grafech vidět i vliv hemodynamické odezvy, která posouvá reakci na stimul o dva až tři snímky doprava. To přímo odpovídá standardní délce hemodynamické odpovědi u lidí.



Obrázek 12 – Výstup klasifikace testovacích dat pomocí finální verze konvoluční neuronové sítě po 70 trénovacích epochách. Testovací data pochází ze stejného experimentu jako trénovací data (Zdroj: Autor, 2019).



Obrázek 13 - Výstup klasifikace testovacích dat pomocí finální verze konvoluční neuronové sítě po 70 trénovacích epochách. Testovací data pochází ze jiného experimentu než trénovací data (Zdroj: Autor, 2019).

8.3 Porovnání výsledků

Tímto testem, tak obě architektury potvrdily schopnost klasifikovat fMRI data na základě prováděné kognitivní funkce v rámci jedné sady snímků, zachycených během jednoho experimentu. Při pokusu a aplikaci naučeného modelu jednoduché sítě na data z následujícího experimentu, nebyla síť schopna vrátit žádné relevantní výsledky.

Hlavním omezení architektury s úplně propojenými vrstvami je neschopnost pracovat se vzory v rámci dvou a více dimenzionálních polí, čímž se ztrácí zásadní část informace obsažené v zaznamenaných datech. Architektura s konvolučními vrstvami byla naopak schopná tyto vzory zaznamenat, a dokonce i aplikovat na testovací data, která byla zaznamenána v rámci dalšího kola experimentu.

9 Závěr

Cílem práce bylo zhodnotit, zda je pomocí v současnosti dostupných metod a technik, možné vyhodnocovat kognitivní činnost mozku pomocí umělé neuronové sítě. Tato práce ve své podstatě kombinuje dvě odlišné oblasti, a to biologii a informatiku. Vzhledem k odlišnosti těchto oblastí bylo zapotřebí vysvětlit všechny důležité pojmy, se kterými pracuji v praktické části. Z pohledu biologie se tak jedná především o oblast neurofyzologie a metody zaznamenávání mozkové aktivity. Z pohledu informatiky se pak jedná především o popis funkce umělých neuronových sítí, jejich prvků a popis jejich v současnosti dostupných implementací.

První z nich je samotná kognitivní psychologie a neurofyzologie mozku. Tato část je popsána v prvních kapitolách této práce. V kapitolách 6 a 7 jsem navrhl a implementoval dva typy umělých neuronových sítí, které byly schopné klasifikovat data na základě kognitivní funkce, zpracování vizuálního vjemu. Hlavním závěrem vycházejícím právě z experimentu popsaném v kapitolách 6 a 7 je, že vyhodnocování fMRI dat pomocí umělých neuronových sítí je možné, ale má určitá úskalí. Nicméně experiment potvrdil, že základní analýza dat se dá provést i bez drahé výpočetní techniky, čímž byly splněny cíle stanovené v úvodu práce.

9.1 Experiment

Cílem experimentu bylo navrhnout a implementovat umělou neuronovou síť schopnou klasifikovat fMRI data. Celý experiment vychází z poznatků popsaných v kapitolách 3 až 5 a je navržen tak, aby ho mohl kdokoli replikovat pomocí běžného počítače v cloudovém prostředí Colab poskytovaného zdarma společností Google.

Výstupem experimentu jsou dvě architektury umělé neuronové sítě. Obě architektury byly schopné klasifikovat fMRI data s přesností od 75 % do 82 %. Vzhledem k omezenému výpočetnímu výkonu, považuji tyto výsledky za potvrzení hypotézy a v rámci diskuse navrhuji další možné kroky, jak by bylo možné tyto výsledky ještě zlepšit.

9.2 Diskuse

V experimentu popsaném v kapitolách 6 a 7 jsem aplikoval všechny informace z předchozích kapitol s cílem navrhnout umělou neuronovou síť, která bude schopná klasifikovat kognitivní aktivitu mozku na základě dostupných dat.

Výsledky experimentu ukazují, že je možné použít umělé neuronové sítě ke klasifikaci fMRI dat na základě vykonávané kognitivní funkce. Hlavním omezením experimentu byl především omezený výpočetní výkon, kvůli kterému nebylo možné použít k trénování všechna dostupná data ani vytvořit síť s vyšší komplexitou. Nicméně na základě porovnání jednoduché neuronové sítě, pouze úplně propojené vrstvy, a komplexní neuronové sítě, konvoluční vrstvy, se dá předpokládat, že s použitím pokročilejší techniky by bylo možné získat ještě lepší výsledky.

Hlavní prostor pro zlepšení je především v oblasti využití více dimenzionálních konvolučních sítí. Zatímco architektura neuronové sítě využívající pouze úplně propojené vrstvy nebyla schopná generalizovat model pro použití i na data z jiných experimentů. Konvoluční neuronová síť narazila především na omezení způsobené malým výpočetním výkonem.

Prvním krokem v dalším zkoumání této oblasti, by tak mělo být použití vícedimenzionálních konvolučních sítí. Z pohledu náročnosti na operační paměť by to znamenalo cca 50krát vyšší nároky vzhledem k nárůstu množství vstupních dat. Dalším krokem by mělo být zavedení rekurentních prvků. To by mělo za následek zlepšení schopností neuronové sítě rozpoznávat přechodné stavy způsobené hemodynamickou odezvou.

Z pohledu vstupních dat, by byl následující logický krok přidání dat obsahujících více různých kognitivních funkcí a tím se dále přiblížit úplnému zmapování všech kognitivních funkcí.

Použitá literatura

- [1] ABADI, Martín, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016. p. 265-283.
- [2] AL-RFOU, Rami, et al. Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688, 2016.
- [3] CUDA: Toolkit Documentation [online]. 2019 [cit. 2019-06-26]. Dostupné z: <https://docs.nvidia.com/cuda/index.html>
- [4] DI OLEGGIO CASTELLO, Matteo Visconti, et al. The neural representation of personally familiar and unfamiliar faces in the distributed system for face perception. *Scientific reports*, 2017, 7.1: 12237.
- [5] FIALA, Petr. Rozpoznávání znaků z reálných scén pomocí neuronových sítí. Praha, 2014. Vedoucí práce Lukáš Neumann.
- [6] fMRI simulátor: fMRI Brno [online]. Brno: fMRI BRNO Výzkumná skupina při LF MU v Brně, 2008, 2.4.2014 [cit. 2019-04-22]. Dostupné z: http://fmri.mchmi.com/main_index.php?strana=22
- [7] FRIEDENBERG, Jay; SILVERMAN, Gordon. Cognitive science: An introduction to the study of mind. Sage, 2011.
- [8] Funkční magnetická rezonance: fMRI Brno [online]. Brno: fMRI BRNO Výzkumná skupina při LF MU v Brně, 2008, 11.1.2008 [cit. 2019-04-22]. Dostupné z: http://fmri.mchmi.com/main_index.php?strana=12
- [9] GONZALEZ-CASTILLO, Javier, et al. Task dependence, tissue specificity, and spatial distribution of widespread activations in large single-subject functional MRI datasets at 7T. *Cerebral Cortex*, 2014, 25.12: 4667-4677.
- [10] HAYEK, Friedrich August. The sensory order: An inquiry into the foundations of theoretical psychology. University of Chicago Press, 1952, s. 184-191.
- [11] HUBEL, David H.; WIESEL, Torsten N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 1962, 160.1: 106-154.
- [12] JANG, Hojin, et al. Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: Evaluation using sensorimotor tasks. *NeuroImage*, 2017, 145: 314-328.
- [13] Keras: The Python Deep Learning library. *Home - Keras Documentation: Documentation for Keras, the Python Deep Learning library*. [online]. 2019 [cit. 2019-06-19]. Dostupné z: <https://keras.io/>
- [14] KYMPLOVÁ, Jaroslava. Katalog metod v biofyzice: Multimediální podpora výuky klinických a zdravotnických oborů. Portál 1. lékařské fakulty Karlovy Univerzity [online]. 2008, 18.1.2008 [cit. 2019-04-15]. Dostupné z: <https://portal.lf1.cuni.cz/clanek-793-katalog-metod-v-biofyzice>
- [15] LIAO, Chien Heng, et al. Estimating the delay of the fMRI response. *NeuroImage*, 2002, 16.3: 593-606.

- [16] MACKENZIE, I. Scott. Human-computer interaction: An empirical research perspective. Newnes, 2012.
- [17] MAŘÍK, Vladimír; ŠTĚPÁNKOVÁ, Olga; LAŽANSKÝ, Jiří. *Umělá inteligence*. Academia, 1993.
- [18] MILLER, George A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 1956, 63.2: 81.
- [19] NOVÁK, Mirko. *Umělé neuronové sítě: teorie a aplikace* [online]. Praha: C.H. Beck, 1998 [cit. 2019-04-24]. Učebnice informatiky. ISBN isbn80-7179-132-6.
- [20] PAVLÍK, Ján. *Spontánní geneze jazyka*. Professional publishing, 2010.
- [21] PAWLOWSKI, Nick, et al. Dltk: State of the art reference implementations for deep learning on medical images. arXiv preprint arXiv:1711.06853, 2017.
- [22] PSTRUŽINA, Karel. Svět poznávání. Nakladatelství Olomouc, Olomouc, 1998.
- [23] RYŠAVÝ, Vladimír. Vektory a tensory [online]. Praha, 1944 [cit. 2019-06-20]. Dostupné z: <http://dml.cz/dmlcz/403069>. článek. Jednota českých matematiků a fyziků.
- [24] SPORNS, Olaf. Networks of the Brain. MIT press, 2010.
- [25] STERNICKEL, Karsten; BRAGINSKI, Alex I. Biomagnetism using SQUIDS: status and perspectives. *Superconductor science and technology*, 2006, 19.3: S160.
- [26] TensorFlow with GPU: Observe TensorFlow speedup on GPU relative to CPU. *Google Colab* [online]. 2019 [cit. 2019-06-19]. Dostupné z: <https://colab.research.google.com/notebooks/gpu.ipynb>
- [27] TVRDÝ, Filip. *Nesnáze introspekce: Svoboda rozhodování a morální jednání z pohledu filozofie a vědy*. Togga, spol. sro, 2015.
- [28] VASILEV, Ivan, et al. Python Deep learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras and TensorFlow. Second edition. Birmingham: Packt publishing, 2019.

Seznam obrázků

Seznam obrázků

Obrázek 1 – Porovnání dat z klidové fáze, vlevo, a aktivní fáze, vpravo. Transverzální řez č.27. Zobrazeno pomocí ImageJ. (Zdroj: Autor, 2019).....	29
Obrázek 2 – Výstup korelační analýzy. (Zdroj: Autor, 2019).....	30
Obrázek 3 – Vizualizace transversálního řezu mozku, který je použitý jako vstupní data pro zpracování umělou neuronovou sítí. (Zdroj: Autor, 2019).	30
Obrázek 4 - Ukázka prostředí Colab. (Zdroj: Autor, 2019).....	32
Obrázek 5 – Vizualizace architektury jednoduché konvoluční sítě. Vrstvy zleva: Konvoluční síť s 32 filtry o velikosti 3x3 pixely, MaxPool vrstva o velikosti 2x2 pixely, úplně propojená vrstva o velikosti 64 neuronů a výstupní vrstva o velikosti jednoho neuronu (Zdroj: Autor, 2019)...	35
Obrázek 6 – V horní části obrázku je porovnání přesnosti neuronové sítě po 150 a po 250 epochách. Cílem je přiblížit se správnému průběhu zobrazenému v dolní části. (Zdroj: Autor, 2019).....	36
Obrázek 7 – Vývoj hodnot loss, horní graf, a accuracy, dolní graf. Na ose x je znázorněn počet průběhů učení, kde každý průběh obsahoval deset epoch (Zdroj: Autor, 2019).	37
Obrázek 8 – Zleva: 1. nejjednodušší konvoluční neuronová síť, 2 konvoluční vrstvy se 4 filtry. 2. Středně složitá konvoluční síť, 3 konvoluční vrstvy s 8 filtry. 3. finální verze konvoluční neuronové sítě, 3 konvoluční vrstvy s 32 filtry (Zdroj: Autor, 2019).....	38
Obrázek 9 – Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).	39
Obrázek 10 - Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).	40
Obrázek 11 – Graf znázorňující výstup neuronové sítě. Na ose X jsou jednotlivé snímky a na ose Y je jistota s jakou neuronová síť vyhodnotila, že mozek zpracovává vizuální stimul (Zdroj: Autor, 2019).	40
Obrázek 12 – Výstup klasifikace testovacích dat pomocí finální verze konvoluční neuronové sítě po 70 trénovacích epochách. Testovací data pochází ze stejného experimentu jako trénovací data (Zdroj: Autor, 2019).	41
Obrázek 13 - Výstup klasifikace testovacích dat pomocí finální verze konvoluční neuronové sítě po 70 trénovacích epochách. Testovací data pochází ze jiného experimentu než trénovací data (Zdroj: Autor, 2019).....	42

Příloha A: Ukázka implementace finální architektury konvoluční neuronové sítě v prostředí Colab

A.1 Import knihoven

```
#Import potřebných knihoven
from __future__ import absolute_import, division, print_function, unicode_literals
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import nibabel as nib

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
```

A.2 Stažení datového setu

```
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-002_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-003_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-004_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-005_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-006_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-007_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-008_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-009_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-010_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-011_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-012_bold.nii.gz
!wget https://openneuro.org/crn/datasets/ds001555/snapshots/1.0.0/files/sub-003:ses-02:func:sub-003_ses-02_task-FullFovAttentionTask_run-010_bold.nii.gz
```

A.3 Načtení dat

```
#Načtení dat ze souboru ve formátu NIfTI do datového pole Numpy array
#Výběr pouze transversálního řezu č.27
data = nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-002_bold.nii.gz').get_fdata().T[:,27,:,:]
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-003_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-004_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-005_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-006_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-007_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-008_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-009_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-010_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
data = np.append(data, nib.load('sub-003:ses-01:func:sub-003_ses-01_task-FullFovAttentionTask_run-011_bold.nii.gz').get_fdata().T[:,27,:,:], axis=0)
```

A.4 Vytvoření štítků

```
#Vytvoření pole štítků
active      = np.ones(10)
pasive      = np.zeros(20)
initial     = np.zeros(5)
final      = np.zeros(9)
predicted_response = np.concatenate((initial, active, pasive, active, pasive, active, pasive, active, pasive, active, pasive, final), axis=0)
predicted_response = np.concatenate(( predicted_response, predicted_response, predicted_response, predicted_response, predicted_response, predicted_response, predicted_response, predicted_response), axis=0)
```

A.5 Normalizace hodnot

```
#Normalizace hodnot v trénovacím poli
data = data / np.amax(data)
```


A.6 Příprava vstupních dat

```
#Vytvoření proměnných train_images a train_labels
#Přidání další dimenze, nutnost pro zpracování konvoluční sítí
#Založení pole s názvy stavů
train_images = data
train_labels = predicted_response
train_images = np.expand_dims(train_images,axis = 4)
train_labels = np.expand_dims(train_labels,axis = 1)
class_names = ["pasive", "active"]
```

A.7 Inicializace modelu

```
#Inicializace modelu
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same',
                 input_shape=train_images.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

A.8 Trénování modelu

```
#Nastavení váh pro každou klasifikační třídu
#Spuštění trénování modelu
class_weight = {0: 0.30487, 1: 0.69512195}
model.fit(train_images, train_labels, epochs=50,
          class_weight=class_weight, validation_split=0.3)
```

Train on 1148 samples, validate on 492 samples

```
Epoch 1/50
1148/1148 [=====] - 3s 3ms/sample - loss:
0.2952 - acc: 0.5880 - val_loss: 0.2938 - val_acc: 0.3049
Epoch 2/50
1148/1148 [=====] - 1s 604us/sample - loss:
0.2940 - acc: 0.5139 - val_loss: 0.2938 - val_acc: 0.3049
...
...
```

A.9 Vyhodnocení modelu: Trénovací data

```
#Vyhodnocení modelu podle trénovacích dat
test_loss, test_acc = model.evaluate(train_images[:200,:,:],
                                     train_labels[:200,:])

print('Test accuracy:', test_acc)
print('Test loss:', test_loss)
```

```
Output:
200/200 [=====] - 0s 788us/sample - loss: 0.3619 - acc: 0.8550
Test accuracy: 0.855
Test loss: 0.36192646622657776
```

A.10 Vyhodnocení modelu: Testovací data (stejný experiment jako trénovací data)

```
testdata_images = nib.load('sub-003:ses-01:func:sub-003_ses-01_task-
FullFovAttentionTask_run-012_bold.nii.gz').get_fdata().T[:,27,:,:]
testdata_labels = np.concatenate((initial, active, pasive, active,
                                   pasive, active, pasive, active, pasive,
                                   active, pasive, final), axis=0)

testdata_images = testdata_images / np.amax(testdata_images)
testdata_images = np.expand_dims(testdata_images,axis = 4)
testdata_labels = np.expand_dims(testdata_labels,axis = 1)
test_loss, test_acc = model.evaluate(testdata_images, testdata_labels)

print('Test accuracy:', test_acc)
print('Test loss:', test_loss)
```

```
Output:
164/164 [=====] - 0s 568us/sample - loss: 0.5760 - acc: 0.7378
Test accuracy: 0.7378049
Test loss: 0.5760075265496243
```

A.11 Vyhodnocení modelu: Testovací data (nový experiment)

```
testdata_images = nib.load('sub-003:ses-02:func:sub-003_ses-02_task-
FullFovAttentionTask_run-010_bold.nii.gz').get_fdata().T[:,27,:,:]
testdata_labels = np.concatenate((initial, active, pasive, active, pasive,
                                   active, pasive, active, pasive, active,
                                   pasive, final), axis=0)

testdata_images = testdata_images / np.amax(testdata_images)
testdata_images = np.expand_dims(testdata_images,axis = 4)
testdata_labels = np.expand_dims(testdata_labels,axis = 1)
test_loss, test_acc = model.evaluate(testdata_images, testdata_labels)

print('Test accuracy:', test_acc)
print('Test loss:', test_loss)
```

Output:

```
164/164 [=====] - 0s 262us/sample - loss: 0.4385 - acc: 0.8293
Test accuracy: 0.8292683
Test loss: 0.4385179012286954
```