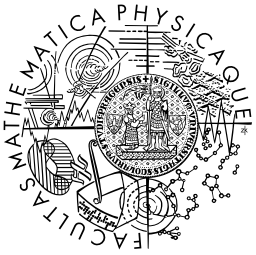


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

DIPLOMOVÁ PRÁCE

Bc. Michal Filippi

Predikce sekundární struktury proteinu pomocí hlubokých neuronových sítí

Katedra softwarového inženýrství

Vedoucí diplomové práce:	RNDr. David Hoksza, Ph.D.
Studijní program:	Informatika
Studijní obor:	Umělá inteligence

Praha 2017

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Predikce sekundární struktury proteinu pomocí hlubokých neuronových sítí

Autor: Bc. Michal Filippi

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. David Hoksza, Ph.D., Katedra softwarového inženýrství

Abstrakt: Znalost struktury, kterou proteiny zaujímají v prostoru, je klíčovým faktorem při studiu jejich funkce. Experimentální zjištění struktury je ale nákladné a časově náročné, proto jsou velmi populární predikční modely struktury. Nejvýraznějším podproblémem predikce struktury proteinů je predikce lokálního uspořádání sousedících aminokyselin určeného vodíkovými vazbami, tzv. sekundární struktury proteinů. Tato práce se zaměřuje na využití hlubokých neuronových sítí v predikci sekundární struktury. Na implementovaném predikčním modelu jsou v rámci této práce testovány různé modifikace sítě, především je pak provedeno srovnání LSTM a GRU paměťových buněk. Dále jsou zkoumány nové metody předzpracování proteinů, a to zrychlení klasické metody výpočtu PSSM a zahrnutí predikce terciární struktury mezi vstupy predikčního modelu. V poslední části práce je ověřována použitelnost vyhlazovacích metod pro modely predikující složitější osmistavové rozdělení sekundárních struktur.

Klíčová slova: bioinformatika protein sekundární struktura strojové učení neuronové sítě

Title: Protein secondary structure prediction using deep neural networks

Author: Bc. Michal Filippi

Department: Department of Software Engineering

Supervisor: RNDr. David Hoksza, Ph.D., Department of Software Engineering

Abstract: Determination of protein structure in space is a crucial part of protein function analysis. But structure determination is an expensive and time consuming process, therefore structure prediction model raised on popularity. The most notable subproblem of protein structure prediction is prediction of local conformation of the adjacent amino acids, ie. secondary structure. This thesis studies usage of deep neural networks for protein secondary structure prediction. We implemented prediction model and different modifications are evaluated. Especially compassion of LSTM and GRU memory cells was done. Furthermore, two new preprocessing methods are evaluated. Fast PSSM calculation method was proposed and prediction of tertiary structure was used as input for prediction model. Last part of this thesis examine application of filtering methods for models predicting secondary structure with eight classes.

Keywords: bioinformatics protein secondary structure machine learning neural networks

Rád bych poděkoval všem, kteří mi během celého mého studia na MFF UK byli jakýmkoliv způsobem nápomocní, v první řadě pak rodině za podporu a zázemí během studia. Současně bych ale chtěl poděkovat také vedoucímu mé diplomové práce panu RNDr. Davidovi Hokszovi, Ph.D. za odborné rady a podněty v průběhu vypracovávání.

Obsah

Úvod

Obrovskou rychlostí rostoucí množství biologických dat, které bylo potřeba zpracovávat, vedlo k nutnosti zpracovávat tato data pomocí počítačů. To dalo vzniknout oboru bioinformatika, který se snaží aplikovat různé informatické metody na problémy biologického původu. Velkou část tohoto vědního oboru tvoří strukturní bioinformatika, která se zabývá analýzou a předpovídáním struktury biologických makromolekul jako například proteinů, RNA nebo DNA (Altman a Dugan, 2003).

Právě zkoumání struktury proteinů je velmi aktivní oblastí, což je způsobeno především důležitostí proteinů pro živé buňky, jejíchž jsou proteiny jednou z hlavních komponent. V každé takové buňce se nachází až několik miliard proteinů (Milo, 2013). Důvodem pro takto obrovský výskyt v živých buňkách je široké spektrum funkcí, které v nich proteiny zastávají. Jedná se například o funkce stavební, obranné, pohybové nebo funkce řízení a katalyzování chemických reakcí.

Právě funkce a vlastnosti jednotlivých proteinů ve velké míře závisí na rozložení proteinu v prostoru, tj. jeho struktuře (Petsko, 2004). To dělá ze znalosti struktury proteinů klíčovou informací například při vývoji léků a enzymů (Whittle a Blundell, 1994; Schaffhausen, 2012). Strukturu je samozřejmě možné získat experimentálně, například pomocí metod rentgenové krystalografie, nukleární magnetické resonance nebo kryoelektronové mikroskopie. Žádná z těchto metod ale není univerzální a není ji spolehlivě možné použít pro každý protein. Navíc všechny tyto metody jsou poměrně časově i finančně náročné. Problémovost určení struktury proteinů dokládá i poměr mezi počtem známých proteinů a počtem proteinů, u nichž je známa i jejich struktura. Na začátku roku 2017 obsahovala databáze UniProt[1] přes 80 miliónů různých proteinových sekvencí, naopak databáze PDB[2] pouze přibližně 121000 proteinových struktur, což tedy odpovídá 0.15% všech známých proteinů.

To vše vytváří větší tlak na vývoj programů schopných předpovídat strukturu proteinů bez experimentálního postupu. To je ale velmi složitý problém, který se vzhledem k obrovskému počtu možných struktur pro každou proteinovou sekvenci zatím nedaří uspokojivě vyřešit. Nejlepších výsledků dosahují metody založené na znalostních databázích využívajících již známé struktury proteinů. Takové metody jsou ale aplikovatelné pouze pro predikci struktur proteinů, které jsou velmi podobné proteinům v databázi (Dorn a kol., 2014b). Jednou z mož-

[1]Databáze známých proteinů `http://www.uniprot.org/`

[2]Databáze všech známých proteinových struktur `http://www.rcsb.org/pdb/`

ností, jak výrazně snížit složitost celého problému predikce, je omezení počtu možných struktur pro daný protein. Za tímto účelem se velmi často při predikci zohledňuje předpokládaná sekundární struktura daného proteinu. Ta nepopisuje celé uspořádání proteinu v prostoru, ale pouze rozděluje protein do segmentů tvořených několika předdefinovanými tvary řetězce.

Ani sekundární strukturu proteinů není snadné získat, ale především pro svoji důležitost při predikci celé struktury proteinů se stala její predikce velmi aktivní oblastí výzkumu. Znalost sekundární struktury je ale možné využít i pro určování rychlosti (Plaxco a kol., 1998) a způsobu (Ozkan a kol., 2007) skládání proteinu nebo dokonce přímo pro předpovídání samotné funkce proteinu (Taherzadeh a kol., 2016). Stejně jako v případě predikce celé struktury je i pro predikci sekundární struktury možné využít databáze známých sekundárních struktur. Ale i zde platí, že použitelnost těchto metod pro velmi odlišné proteiny je nízká.

V této práci se proto zaměříme na predikci sekundární struktury proteinů bez využití znalostních databází, ale pomocí metody založené na hlubokých neuronových sítích. Ty v posledních několika letech stály za překonáváním dosavadních nejlepších metod napříč mnoha oblastmi výzkumu (Goodfellow a kol., 2016). Predikce sekundární struktury nezůstala výjimkou a i zde prediktory založené na hlubokých neuronových sítích posunuly dosavadní hranice. Na jednom takovém predikčním modelu založíme tuto práci a pokusíme se jeho hlavní komponenty vylepšit, otestujeme na něm nové metody předzpracovaní proteinů a také dodatečné filtrovací metody pro výstup z modelu.

Struktura práce

Práce je rozdělena do několika částí. První kapitola je věnována nezbytnému biologickému pozadí problému nutného k pochopení zbylých částí práce. Následující kapitola se zabývá formulováním problému predikce sekundární struktury a používanými metodami vyjma neuronových sítí. Těm je věnována celá navazující kapitola, ve které jsou popsány samotné neuronové sítě včetně jejich využití při predikce sekundární struktury. Dále práce obsahuje kapitolu obsahující naše základní navržení a popis experimentů, které jsou v následující kapitole provedeny a zhodnoceny. V závěru práce se nachází shrnutí všech provedených experimentů a získaných výsledků a jsou formulovány závěry z této práce.

1. Biologie proteinových struktur

V této sekci čtenáři přiblížíme, co to proteiny (bílkoviny) jsou a jaké jejich struktury zkoumáme. Ačkoliv tématem této práce je predikce sekundární struktury proteinů, na biologickou část problému nebude kladen důraz. Proto také tako kapitola představuje pouze základní informace. Pro podrobný přehled doporučujeme knihu Lehninger Principles of Biochemistry (Boyle, 2005), ze které tato kapitola, mimo jiných publikací, čerpá.

Základním biologickým pojmem, se kterým budeme pracovat, je protein. Pro vysvětlení, jak proteiny vypadají a jaké struktury vytvářejí, je nutné začít popisem aminokyselin. Ty jsou totiž základním stavebním kamenem proteinů a právě jejich řetězením proteiny vznikají.

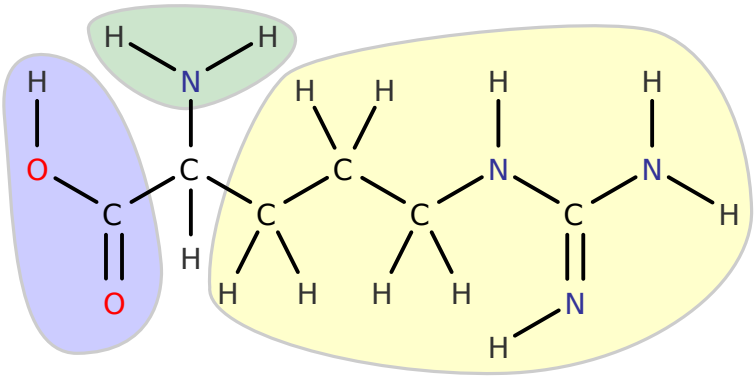
1.1 Aminokyseliny

Pod pojmem aminokyseliny se obecně uvažují molekuly obsahující karboxylovou skupinu (-COOH) a aminoskupinu ($-NH_2$). V rámci této práce, podobně jako v biochemii, si ale pod tímto pojmem budeme představovat téměř výhradně[1] α-aminokyseliny, tedy aminokyseliny, ve kterých jsou karboxylové a aminové skupiny spolu s jedním atomem vodíku (H) a postranním řetězcem propojeny jedním centrálním atomem uhlíku (C). Takovou strukturu můžeme pozorovat např. u argininu na obrázku č. 1.1. Právě postranní řetězec je hlavní faktor, podle kterého jednotlivé aminokyseliny rozlišujeme, jelikož zbytek molekuly je shodný pro všechny aminokyseliny. V kontextu bílkovin nás bude zajímat pouze dvacet různých aminokyselin, ze kterých jsou tvořeny všechny přirozeně se vyskytující proteiny procesem zvaným translace[2]. Přirozené proteiny mohou obsahovat značné množství dalších aminokyselin, ty jsou ale přidávány ex post v procesu zvaném posttranslační modifikace k již hotové struktuře proteinu. Přehled základních 20 aminokyselin je možné nalézt v tabulce č. 1.1.

Jednotlivé aminokyseliny mezi sebou mohou reagovat, a vytvářet tak mezi sebou peptidické vazby. Konkrétně aminová skupina jedné aminokyseliny reaguje s karboxylovou skupinou druhé aminokyseliny, aby vytvořily peptidickou vazbu

[1] Vyjímku tvoří pouze prolin, jehož aminová skupina je zacyklená s postranním řetězcem. Nejedná se tedy o α-aminokyselinu, ale o iminokyselinu.

[2] Aby byl výčet aminokyselin úplný, museli bychom zahrnout také další dvě aminokyseliny a to pyrolysin a selenocystein. Jejich výskyt je ale extrémně nízký, nebudeme je tedy v tomto stručném přehledu zohledňovat.



Obrázek 1.1: Struktura aminokyseliny arginin, na které můžeme vidět karboxylovou skupinu (modře), aminoskupinu (zeleně) a postranní řetězec (žlutě). Tyto tři části molekuly jsou propojeny centrálním atomem uhlíku. Postranní řetězec je unikátní pro každou aminokyselinu, zbylá část molekuly je identická ve všech aminokyselinách.

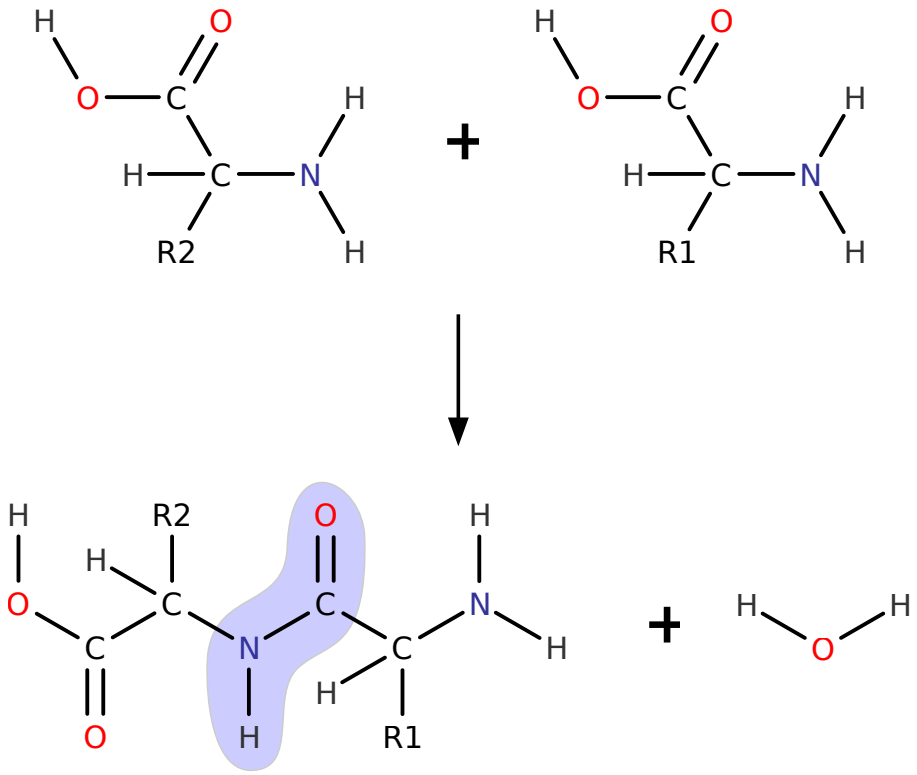
a jednu molekulu vody (H_2O). Právě peptidická vazba je základním kamenem pro řetězení aminokyselin (vytváření peptidů), a tedy i pro vznik proteinů. V takto vytvořeném peptidickém řetězci jsou původní části jednotlivých aminokyselin často nazývány rezidua. Schéma vzniku peptidické vazby reakcí dvou aminokyselin je zobrazeno na obrázku č. 1.2.

1.2 Proteiny

Proteiny jsou tedy velké molekuly tvořené řetězcem aminokyselin propojených peptidickou vazbou. Délka samotného řetězce se velmi liší a je shora omezena pouze teoretickou hranicí danou poklesem stability proteinu s rostoucí délkou řetězce. Nicméně nejdelší známý protein, titin, obsahuje okolo 33000 reziduí (Opitz a kol., 2003). Vzhledem k možné délce proteinů a počtu aminokyselin, které proteiny tvoří, je počet různých kombinací aminokyselinových řetězců, a tedy i proteinů, obrovský. Právě to také umožňuje proteinům zastávat v živých buňkách tak široké spektrum funkcí. Konkrétní funkce jsou do značné míry určeny především strukturou, kterou daný protein v prostoru zaujímá.

1.3 Struktura proteinů

Každý protein v prostoru zaujme unikátní trojrozměrnou strukturu, která je určena posloupností aminokyselin, ze kterých je protein složen (Anfinsen, 1973). Tuto strukturu zkoumáme na několika úrovních, jedná se o primární, sekundární,



Obrázek 1.2: Schéma řetězení aminokyslen, tj. vzniku peptidické vazby reakcí dvou aminokyselin s postranními řetězci $\mathtt{R}_1$ a $\mathtt{R}_2$. Vedlejším produktem reakce je molekula vody. Vzniklá peptidická vazba je zvýrazněna modře.

terciární a kvartérní strukturu, jejichž rozdělení bylo představeno již v polovině minulého století (Linderstrøm-Lang, 1952).

1.3.1 Primární struktura

Primární strukturou proteinu se rozumí sekvence aminokyselin tvořících daný protein. Standardně se tato sekvence zapisuje tak, aby aminokyselina s volnou aminoskupinou byla zapsána jako první. Primární strukturu lze tedy jednoznačně reprezentovat pomocí řetězce o délce rovné počtu aminokyselin v proteinu nad abecedou o 20 znacích, kde každý znak odpovídá jedné aminokyselině. Ustálenou abecedu pro reprezentaci aminokyselin ukazuje tabulka 1.1. Abeceda se často také rozšiřuje o znak X, který reprezentuje nespecifikovanou aminokyselinu. Právě tuto abecedu o 21 znacích budeme dále v této práci využívat pro reprezentaci primární struktury.

Primární struktura tedy žádným způsobem neřeší, jak se daný protein skládá v prostoru. Tím se zabývají až další typu struktur, počínaje sekundární.

Aminokyselina	Zkratka	Podíl [%]	Aminokyselina	Zkratka	Podíl [%]
Alanin	A	8.76	K. glutamová	E	6.32
Arginin	R	5.78	Leucin	L	9.68
Asparagin	N	3.93	Lysin	K	5.19
Cystein	C	1.38	Methionin	M	2.32
Fenylalanin	F	3.87	Prolin	P	5.02
Glutamin	Q	3.9	Serin	S	7.14
Glycin	G	7.03	Threonin	T	5.53
Histidin	H	2.26	Tryptofan	W	1.25
Isoleucin	I	5.49	Tyrosin	Y	2.91
K. asparagová	D	5.49	Valin	V	6.73

Tabulka 1.1: Seznam aminokyselin vyskytujících se běžně v proteinech, jejich jednopísmenná zkratka, která se využívá pro zápis primární struktury, a jejich procentuální zastoupení ve známých proteinechKozlowski, 2017.

1.3.2 Sekundární struktura

Sekundární struktura proteinu popisuje lokální uspořádání sousedících aminokyselin určené geometrií peptidické vazby a stabilizované nekovalentními interakcemi, zejména vodíkovými můstky. Základní rozdělení na 3 typy sekundárních struktur, α-helix, β-skládaný list (β-strand) a neuspořádaná struktura (coil), navrhl Pauling v roce 1951 (Pauling a kol., 1951).

Struktura typu α-helix odpovídá pravotočivé šroubovici s postranními řetězci aminokyselin směřujícími ven. Každá jedna otočka šroubovice je tvořena 3.6 aminokyselinami, každá aminokyselina tedy odpovídá otočce o 100°. Mezi jednotlivými aminokyselinami v sousedních patrech šroubovice vznikají vodíkové vazby mezi skupinami `C=O` a `N-H`, které drží celou strukturu pohromadě. Vodíkové vazby mezi těmito skupinami vznikají i ve struktuře β-skládaný list. Ta ale není tvořena jedním spojitým regionem řetězce, jako je tomu v případě α-helixu, ale je tvořena kombinací dvou a více rovnoběžných regionů. Právě mezi těmito rovnoběžnými regiony vznikají vodíkové můstky. Postranní řetězce aminokyselin ve struktuře β-skládaný list směřují střídavě pod a nad kolmo k rovině určené rovnoběžnými řetězci. Neuspořádaná struktura pak pouze reprezentuje oblasti proteinu, které nevykazují ani jeden z předchozích dvou vzorů.

S vývojem programu DSSP, který slouží k určování sekundární struktury z kompletního trojrozměrného obrazu molekuly, provedli autoři (Kabsch a Sander, 1983) další rozšíření původních 3 typů sekundárních struktur na 8 typů. Vyjma původní α-helixové struktury přidali autoři další dva typy helixu, a to π-helix a 3_{10}-helix. V obou případech se také jedná o pravotočivou šroubovici, hlavní rozdíl ale tvoří počet aminokyselin potřebných pro jednu otočku šroubovice. Otočka π-helixu je tvořena 4.1 aminokyselinami a otočka 3_{10}-helixu pouze 3 aminokyselinami. Helixové struktury jsou ale z definice zespoda omezeny počtem aminokyselin, které tvoří danou šroubovici. Kratší otáčející se segmenty fixované vodíkovými můstky se nazývají β-otočky (β-turn). V případě absence vodíkových vazeb se pak jedná o strukturu zvanou ohyb (higj curvature loop). V případě struktur vázaných vodíkovými můstky mezi vzdálenými aminokyselinami rozlišují autoři mezi jednou izolovanou vazbou nazvanou izolovaný β-most (β-bridge) a mezi delší sekvencí zvanou β-skládaný list (β-strand). Neuspořádaná struktura opět pouze reprezentuje oblasti proteinu, které nevykazují ani jeden z předchozích vzorů.

V závislosti na zvoleném rozdělení mluvíme tedy o třístavové (S_3) nebo osmistavové (S_8) sekundární struktuře. Běžně se mezi S_8 a S_3 používá následující převod (Zhang a kol., 2011; Kountouris a kol., 2012; Patel a Mazumdar, 2014; Feng a kol., 2014; Rashid a kol., 2016). Původní α-helix, π-helix a 3_{10}-helix pře-

vedeme na α-helix, β-skládaný list, β-most na β-skládaný list a β-otočku, neuspořádanou strukturu a ohyb převedeme na neuspořádanou strukturu. Pokud označíme každou ze sekundárních struktur jedním písmenem, bude možné celou sekundární strukturu proteinu reprezentovat podobně jako primární strukturu řetězcem znaků nad abecedou o 3 resp. 8 znacích. Všechny zmíněné typy sekundární struktury včetně písmen, jakým se obvykle označují, a vztahy pro převod z S_8 na S_3 jsou znázorněny v tabulce 1.2.

Sekundární struktura S_3	Zkratka	Sekundární struktura S_8	Zkratka
α-helix	H	α-helix	H
		π-helix	I
		3_{10}-helix	G
β-skládaný list	E	β-skládaný list	E
		izolovaný β-most	B
Neuspořádaná struktura	C	Neuspořádaná struktura	L
		Ohyb	S
		β-otočka	T

Tabulka 1.2: Seznam rozlišovaných sekundárních struktur, jejich symbolů pro zápis struktury sekvence a vztah mezi oběma rozděleními S_3 a S_8.

Se sekundární strukturou také velmi úzce souvisí tzv. přístupnost rozpouštědla. Ta popisuje, jak velká plocha daného rezidua je přístupná rozpouštědlu, v kontextu proteinů téměř výhradně vodě. A stejně jako v případě sekundární struktury se k výpočtu přístupnosti rozpouštědla využívá program DSSP. Vzhledem k propojení těchto vlastností bývají často jejich predikce spojovány, čehož také v této práci využijeme.

Sekundární struktura tedy již do jisté míry zkoumá uspořádání jednotlivých aminokyselin v prostoru, ale neposkytuje dostatečnou informaci k vymodelování celého proteinu. Jedná se pouze o jistý mezistupeň mezi primární a terciární strukturou, která zkoumá právě polohu jednotlivých atomů proteinu v prostoru.

1.3.3 Terciární a kvartérní struktura

Mimo primární a sekundární struktury, které budou předmětem této práce, lze u proteinů rozlišovat také terciární a kvartérní strukturu. Obě tyto úrovně se již zabývají celkovým rozložením proteinu v prostoru na úrovni jednotlivých atomů. Terciární struktura popisuje uspořádání jednoho proteinového řetězce. Ten, díky způsobu napojení jednotlivých aminokyselin, má pouze dva stupně rotační volnosti za každou aminokyselinu v řetězci a to úhly označované ϕ a ψ. Celou trojrozměrnou strukturu proteinu lze tedy zapsat jako posloupnost dvojic (ϕ, ψ), případně jako sérii souřadnic centrálního atomu uhlíku, na který je napojena karboxylová a aminová skupina.

Některé komplikovanější proteiny se skládají z vícero proteinových řetězců. Ty se vzájemným působením shlukují a vytváření komplexnější strukturu zvanou kvartérní struktura. Tyto proteiny obsahující několik řetězců nejsou předmětem této práce.

1.4 Určování struktury proteinů

Určování struktury proteinů reprezentuje problém hledání terciární struktury pro protein s danou primární strukturou. Tento problém lze zatím řešit pouze experimentálně, a to především pomocí rentgenové krystalografie. Ta ale vyžaduje krystalizaci proteinu, a není tak použitelná pro všechny proteiny. Ani ostatní používané metody, nukleární magnetická resonance a kryoelektronová mikroskopie, nejsou univerzální. Velmi aktivní oblastí se tak stala predikce terciární struktury na základě primární struktury proteinu a to i přes to, že se jedná o NP-úplný problém (Hart a Istrail, 1997; Crescenzi a kol., 1998).

Podle Afinsenova dogmatu (Anfinsen, 1973) protein vždy zaujímá stabilní dosažitelnou strukturu, která odpovídá struktuře s minimální hodnotou potenciální energie. Toho využívají některé programy, které se při predikci snaží nalézt minimum energetické funkce definované nad strukturami proteinu. Vzhledem k astronomickému počtu možných struktur pro daný protein je ale hledání minima extrémně složitý problém. Tento přístup využívá například projekt Rosetta@home, který řeší výpočetní náročnost distribucí výpočtů na miliony zařízení poskytnutých dobrovolníky.

Extrémní výpočetní náročnost lze řešit využitím již známých struktur proteinů. Běžným přístupem je prohledání databáze proteinů se známými strukturami a nalezení proteinů s podobnou primární strukturou, jakou má proteinu, jehož strukturu je predikována. Z nalezených proteinů a jejich struktur je následně

vymodeluje predikovaná struktura v závislosti na zarovnání nalezených proteinů. Příkladem projektu využívající tuto metodu je švýcarský SWISS-MODEL (Biasini a kol., 2014). Metody založené na tomto principu obecně dosahují nejlepších predikčních výsledků. Problém ale nastává, pokud se v databázi známých struktur nacházejí pouze proteiny, jejichž primární struktura je velmi odlišná od hledané. V takovém případě se tyto metody stávají velmi nepřesné.

Tento problém se snaží odstranit poslední skupina, která nepracuje přímo s primárními strukturami proteinů. Naopak využívá skutečnosti, že počet struktur přírodně se vyskytujících proteinů je relativně nízký (Wang, 1998). Lze tedy využít menší množinu šablonových struktur, ze kterých se při predikci struktury nového proteinu vybere ta šablona, do které lze daný protein nejlépe namodelovat. Takový přístup implementuje například RaptorX (Källberg a kol., 2012).

I přes všechny používané heuristické metody ve výše zmíněných metodách je predikce terciární struktury velmi výpočetně náročný proces. Proto se velká část zmíněných metod snaží dále omezit prohledávaný prostor struktur pomocí sekundární struktury (Dorn a kol., 2014a). To z predikce sekundární struktury dělá jeden z nejaktivněji zkoumaných podproblémů predikce struktury proteinů.

2. Predikce sekundární struktury proteinů

Sekundární struktura proteinů je typicky vypočítána pomocí programu DSSP (Kabsch a Sander, 1983) z terciární struktury daného proteinu. Takový výpočet je ale možný pouze pro malý zlomek známých proteinů, protože u většiny proteinů není struktura známa. Navíc se sekundární struktura využívá ve velké míře při samotné predikci terciární struktury, a je tedy nutné hledat metody pro odhad sekundární struktury pouze na základě primární struktury bez využití terciární struktury. Takové metody lze ale využít i mimo předpovídání terciární struktury. Sekundární struktura totiž poskytuje dostatečný náhled na strukturu proteinu k odvození některých dalších vlastností (Plaxco a kol., 1998; Ozkan a kol., 2007; Taherzadeh a kol., 2016). Lze ji ale využít například i k zarovnávání proteinů (Zhou a Zhou, 2005; Deng a Cheng, 2011).

Z informatického hlediska lze problém predikce sekundární struktury popsat jako problém převodu řetězce nad abecedou o 21 znacích na řetězec stejné délky nad abecedou o 3 resp. 8 znacích podle použitého rozdělení sekundárních struktur S_3 resp. S_8. Právě pravidla pro převod těchto řetězců ale nejsou známa, proto se k jejich odvození používají různé statistické metody a metody strojové učení. Ty využívají proteiny se známou terciární strukturou a vypočítanou sekundární strukturou jako vzory, ze kterých odvodí predikční modely. Pro porovnání jednotlivých modelů se ustálily dvě základní metriky měřící přesnost predikce modelů.

2.1 Metriky pro měření kvality modelů

Predikční modely sekundárních struktur lze hodnotit na základě několika různých metrik. Základní a nejjednodušší metrikou je prostý poměr správně určených reziduí k délce proteinu, formálně tedy následovně.

$$Q_n^i = \frac{100}{R_i} \cdot \sum_{s \in S_n} M_{ss}^i \quad [\%],$$

kde R_i je celkový počet reziduí v daném proteinu i, n odpovídá počtu rozlišovaných sekundárních struktur (3 nebo 8) a M_{mn}^i je počet reziduí v proteinu i se sekundární strukturou m a predikcí n. M_{ss}^i tedy odpovídá počtu reziduí správně klasifikovaných na strukturu s.

Metriku Q_n lze také rozšířit dvěma způsoby na celou množinu proteinů.

$$Q_n^U = \frac{1}{N} \cdot \sum_{i=1}^{N} Q_n^i \quad [\%],$$

$$Q_n^W = \frac{1}{\sum_{i=1}^{N} R_i} \cdot \sum_{i=1}^{N} R_i \cdot Q_n^i \quad [\%],$$

kde N je počet proteinů v dané množině. Hodnota Q_n^U tedy reprezentuje nevážený průměr metriky Q_n napříč všemi proteiny z dané množiny a jeho hodnota popisuje průměrnou úspěšnost predikce proteinů nehledě na jejich délku. Naopak Q_n^W je vážený průměr s vahou rovnou délce jednotlivých proteinů, jeho hodnota tedy reprezentuje průměrnou úspěšnost predikce jednotlivých reziduí napříč celou množinou.

Zajímat nás také může úspěšnost predikce jednotlivých sekundárních struktur, tu můžeme spočítat následovně.

$$Q_{n,s} = 100 \cdot \frac{\sum_{i=1}^{N} M_{ss}^i}{\sum_{i=1}^{N} \sum_{m \in S_n} M_{sm}^i} \quad [\%],$$

kde s je zkoumaná sekundární struktura.

Druhou často používanou metrikou je skóre překryvu segmentů [1] (*SOV* - Segment Overlap Score) (Zemla a kol., 1999). *SOV* skóre, na rozdíl od Q_n, nepenalizuje každou chybu v predikci se stejnou vahou. *SOV* skóre upřednostňuje predikce s chybami na přechodu jednotlivých segmentů sekundárních struktur a naopak více penalizuje chyby uprostřed segmentů. Snahou je tedy preferovat predikce se správným počtem segmentů, ačkoliv jejich délka se může mírně lišit od skutečnosti.

Pro výpočet *SOV* skóre je nejdříve nutné definovat množiny T_s^i a ${T'}_s^i$ pro $s \in S_n$ a protein i, ze kterých výpočet vychází. T_s^i je množina všech dvojic segmentů (k, l) sekundárních struktur typu s v proteinu i takových, že k je segmentem ve skutečné sekundární struktuře, l je segmentem v její predikci zvoleným modelem a zároveň oba segmenty mají neprázdný průnik reziduí, které pokrývají. Množina ${T'}_s^i$ je pak množina všech segmentů k struktury s ve skutečné sekundární struktuře, které nesdílí žádné residuum s žádným segmentem struktury s v predikované struktuře. Pro protein i, množinu O_s^i segmentů typu s ve skutečné struktuře a množinu P_s^i segmentů typu s v predikci lze T_s^i a ${T'}_s^i$ popsat formálně takto.

[1] Segmentem je v tomto kontextu uvažována nepřerušená sekvence jednoho typu sekundární struktury a to buď ve skutečné struktuře nebo v její predikci.

$$T_s^i = \{(k,l); k \in O, l \in P, k \cap l \neq \emptyset\}$$

$$T'^i_s = \{k; k \in O, \forall l \in P, k \cap l = \emptyset\}$$

Samotné SOV skóre je pak definováno následovně:

$$SOV_n^i = \frac{1}{\sum_{m \in S_n} N_m^i} \cdot \sum_{s \in S_n} \cdot \sum_{(s_o,s_p) \in T_s^i} \gamma(s_o, s_p) \cdot 100$$

$$N_p^i = \sum_{(s_o,s_p) \in T_p^i} len(s_o) + \sum_{s_o \in T'^i_p} len(s_o)$$

$$\gamma(s_o, s_p) = \frac{minov(s_o, s_p) + \delta(s_o, s_p)}{maxov(s_o, s_p)} \cdot len(s_o)$$

$$\delta(s_o, s_p) = min \left\{ \begin{array}{c} maxov(s_o, s_p) - minov(s_o, s_p) \\ minov(s_o, s_p) \\ \lfloor 0.5 \cdot len(s_o) \rfloor \\ \lfloor 0.5 \cdot len(s_p) \rfloor \end{array} \right\}$$

kde $len(k)$ je délka segmentu k, $minov(k,l)$ je počet residuí pokrytých průnikem segmentů k a l, $maxov(k,l)$ je počet residuí pokrytých alespoň jedním ze segmentů k a l. Metrikou SOV pro celé množiny proteinů budeme rozumět průměr SOV přes všechny proteiny v množině, tedy následovně.

$$SOV_n = \frac{1}{N} \cdot \sum_{i=1}^{N} SOV_n^i \quad [\%],$$

V některých případech se také používá metrika zvaná SEL (Kountouris a kol., 2012), reprezentující průměr metrik Q_n a SOV.

2.2 Předzpracovaní dat

Predikce sekundární struktury vychází pouze ze znalosti primární struktury proteinu. To ale nebrání snaze extrahovat z primární struktury další informace vhodné k predikci. První jednoduché predikční modely využívaly pouze informace získané z jednotlivých reziduí proteinu (Scheraga, 1960; Finkelstein a Ptitsyn, 1971). Tento přístup ale poskytoval pouze velmi omezené predikční možnosti, jelikož nezohledňoval vliv sousedních reziduí na vytvářenou strukturu. Novější modely tedy začaly zohledňovat celé okno reziduí zvolené pevné šířky (Kabat a

Wu, 1973; Arnold a kol., 1992), což přineslo výrazné zlepšení. Poslední a také nejvýraznější změnou v předzpracování dat bylo využití informací o konzervovanosti sekvence získaných ze zarovnání homologních sekvencí (Zvelebil a kol., 1987) a z PSSM (position-specific scoring matrix) (Jones, 1999).

Právě využití PSSM je spolu se samotnou primární strukturou při predikci sekundární struktury dnes standard pro všechny běžně používané metody. PSSM se typicky získává pomocí programu PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) (Camacho a kol., 2009), který primárně slouží k vyhledává podobných proteinových sekvencí v proteinové databázi. Výstupní PSSM pro danou vstupní sekvenci je matice reprezentující rozdělení jednotlivých aminokyselin na všech pozicích zarovnaných homologních sekvencí. Matice vždy obsahuje 20 řádků odpovídající 20 aminokyselinám vyskytujícím se běžně v proteinech. Pozice i,j v matici reprezentuje logaritmus pravděpodobnosti výskytu aminokyseliny i na pozici j v zarovnaných sekvencích. PSSM tedy reprezentuje konzervovanost vstupní proteinové sekvence. Běžně se také před předáním PSSM predikčnímu modelu škálují všechny prvky matice pomocí sigmoidy[2].

V některých výjimečných případech byly během predikce zohledňovány také další informace jako například fyzikálně chemické vlastnosti jednotlivých aminokyselin (Wang a kol., 2011; Heffernan a kol., 2017), případně informace o pozici rezidua na konci proteinu (Zhou a Troyanskaya, 2014).

2.3 Metody pro vytváření predikčních modelů

Predikce sekundární struktury proteinů je aktivní oblast, které se již dlouho věnují různé vědecké skupiny po celém světě. S vývojem celé oblasti a postupem času se také výrazně měnily metody použité k predikci. První pokusy o předpovídání sekundární struktury byly založené pouze na jednodušších statistických metodách (Finkelstein a Ptitsyn, 1971; Scheraga, 1960; Chou a Fasman, 1974; Garnier a kol., 1996). Vzhledem k velmi omezeným schopnostem predikce ale tyto metody nelze nyní považovat za velmi úspěšné. S rostoucím počtem experimentálně určených struktur se začaly více využívat metody založené na využívání známých struktur homologních sekvencí (Lin a kol., 2005; Montgomerie a kol., 2006). Tyto a podobné metody dodnes dosahují nejvyšší úspěšnosti na proteinech s dostatečným počtem homologních sekvencí v PDB. Takových sekvencí je ale pouze zlomek, proto se dále zaměříme na univerzálnější metody, a to strojové učení.

[2]Sigmoida je funkce definovaná na reálných číslech jako $f(x) = \frac{1}{1+e^{-x}}$.

Právě totiž rozmach strojového učení přinesl pokrok v úspěšnosti predikce. V oblasti predikce sekundární struktury se jednalo především o markovovské modely (Aydin a kol., 2006; Malekpour a kol., 2009; Martin a kol., 2006), SVM (Ward a kol., 2003; Sun a Huang, 2006; Hua a Sun, 2001; Kim a Park, 2003; Guo a kol., 2004) a neuronové sítě. Všechny tyto tři metody pracovaly pouze s jistým okolím rezidua, jehož sekundární strukturu se snažily určit. Jednotlivá rezidua ale mohou mít vliv na sekundární strukturu reziduí na vzdálené pozici v proteinu. Na přelomu tohoto tisíciletí se k predikci začaly využívat neuronové sítě, které teoreticky umožňují zohledňovat právě tyto vztahy mezi rezidui po celé délce proteinu a nejsou tedy limitovány velikostí plovoucího okna. Neuronové sítě se tak staly nejpoužívanější a také aktuálně nejúspěšnější metodou strojového učení v predikci sekundární struktury.

3. Neuronové sítě v predikci sekundární struktury

Umělé neuronové sítě (Kononenko a Kukar, 2007) jsou modely strojového učení inspirované biologickými nervovými systémy. Jedná se o jeden z neúspěšnějších a nejpoužívanějších modelů, který našel uplatnění v mnoha oborech, např. v klasifikaci obrázků (Rawat a Wang, 2017), zpracování hlasu (Siniscalchi a kol., 2014), asistenci lékařských diagnóz (lin a kol., 2016) nebo predikci finančních trhů (Moghaddam a kol., 2016). Umělé neuronové sítě se skládají z menších jednotek zvaných neurony. Každý neuron má libovolný počet číselných vstupů $x_1, \ldots, x_n$ a právě jeden výstup definovaný následovně.

$$F(x_1, \ldots, x_n) = \Phi\left(\sum_{i=1}^{n} w_i \cdot x_i - b\right)$$

kde parametry w_i se nazývají váhy vstupů a parametr b práh neuronu. Tyto parametry jsou různé pro každý neuron v síti. Funkce Φ se nazývá aktivační funkce neuronu a nejčastěji se jedná o funkce *sigmoid*, *tanh* nebo *ReLU* (rectified linear unit)[1]. Jednotlivé neurony jsou v síti propojeny tak, že výstup každého neuronu je propojen se vstupem libovolného množství dalších neuronů. Výjimku tvoří pouze tzv. výstupní neurony, jejichž výstup je považován za výstup celé neuronové sítě a nemusí tak být napojen na vstup dalších neuronů. Druhou významnou skupinou jsou tzv. vstupní neurony, které nemají žádné vstupy a jejich výstupy odpovídají vstupům celé neuronové sítě. Neurony, které nespadají mezi vstupní ani výstupní, nazýváme skryté neurony. Podle samotného způsobu propojení jednotlivých neuronů respektive podle orientovaného grafu, který jednotlivé neurony a jejich orientované propojení tvoří, rozlišujeme neuronové sítě na dopředné a rekurentní.

3.1 Dopředné sítě

Dopředné neuronové sítě jsou jednodušší varianta umělých neuronových sítí, kde neurony a hrany mezi nimi tvoří orientovaný acyklický graf. V takovém případě totiž můžeme neurony uspořádat do vrstev tak, aby vstupní a výstupní

[1]Funkce *ReLU* odpovídá lomené funkci $ReLU(x) = \begin{cases} 0, & pokud \;\; x < 0 \\ x, & jinak \end{cases}$.

neurony tvořily dvě samostatné vrstvy a pro každý skrytý neuron se všechny neurony, jejichž výstupy tvoří množinu vstupů daného neuronu, nacházely v některé z předchozích vrstev. Celý výpočet neuronové sítě s n vrstvami neuronů lze pak provést pomocí n iterací. V každé iteraci lze vypočítat výstupní hodnoty všech neuronů v dané vrstvě, jelikož všechny jejich vstupy byly vypočítány v předchozích iteracích. Celá neuronová síť s k vstupy a l výstupy tedy reprezentuje funkci $\Theta : \mathbb{R}^k \to \mathbb{R}^l$. Jak bylo ukázáno (Cybenko, 1989), pomocí takto sestrojené funkce pouze o třech vrstvách a konečným počtem neuronů se sigmoidální aktivační funkcí lze aproximovat libovolnou spojitou funkci na zvoleném kompaktním intervalu. Dopředné neuronové sítě jsou tedy dostatečně silný nástroj na modelování libovolného problému, který lze zapsat jako funkci.

Problém ale spočívá v nalezení vhodných vah w_i a prahů b pro všechny neurony tak, aby byla funkce dostatečně dobře aproximována. Bylo ukázáno, že hledání těchto vah je NP-těžký problém (Judd, 1990). Proces hledaní parametrů se nazývá učení sítě. To může v principu probíhat vícero způsoby, ale v rámci této práce budeme využívat pouze učení s učitelem, zaměříme se tedy výhradně na něj. Učení s učitelem se od ostatních metod učení liší využíváním informace o žádoucích výstupech pro množinu vstupních dat. Těmto datům s žádoucí výstupní hodnotou říkáme označená data. Jednou z nejvyužívanějších metod učení s učitelem je algoritmus zpětného šíření chyby (Rumelhart a kol., 1988). Ten využívá množinu označených dat $(x_1,y_1),\ldots,(x_n,y_n)$, kde y_i je požadovaný výstup sítě pro vstup x_i, k definici chybové funkce na výstupních neuronech. Typickým příkladem chybové funkce může být funkce SE (squared error).

$$SE = \frac{1}{2}\sum_{i=1}^{n} \|y_i - d_i\|^2$$

Pro fixní architekturu sítě a množinu označených dat je tedy SE funkce definovaná nad prostorem všech možných hodnot parametrů všech neuronů v síti. Učení sítě poté odpovídá minimalizaci chybové funkce. Ta probíhá pomocí gradientního sestupu, kde směr sestupu je pro každý parametr w sítě určen parciálních derivací $\frac{\partial SE}{\partial w}$. Velikost samotného kroku při gradientním sestupu je jedním z parametrů učení, ale lze jej také dynamicky měnit během učení sítě např. s délkou doby učení. Druhým často používaným principem je využití setrvačnosti při gradientním sestupu. V takovém případě je při výpočtu směru sestupu zohledněn směr předchozího sestupu.

Dopředné neuronové sítě byly poprvé použity k predikci sekundární struktury na konci 80. letech 20. století skupinami z Cambridge (Holley a Karplus, 1989) a Baltimoru (Qian a Sejnowski, 1988). Použité neuronové sítě obsahovaly jednu

skrytou vrstvu a pracovaly také s podobnými vstupy. Vstupem bylo okno 13 resp. 17 vektorů o 21 prvcích, kde každý z těchto vektorů reprezentoval příslušnou aminokyselinu na dané pozici v plovoucím okně pomocí one-hot reprezentace[2]. Ačkoliv se z dnešního pohledu jednalo o poměrně jednoduché konfigurace neuronových sítí, v obou těchto případech se jednalo o pokrok v úspěšnosti predikce oproti předchozím používaným metodám pro predikci sekundární struktury S_3.

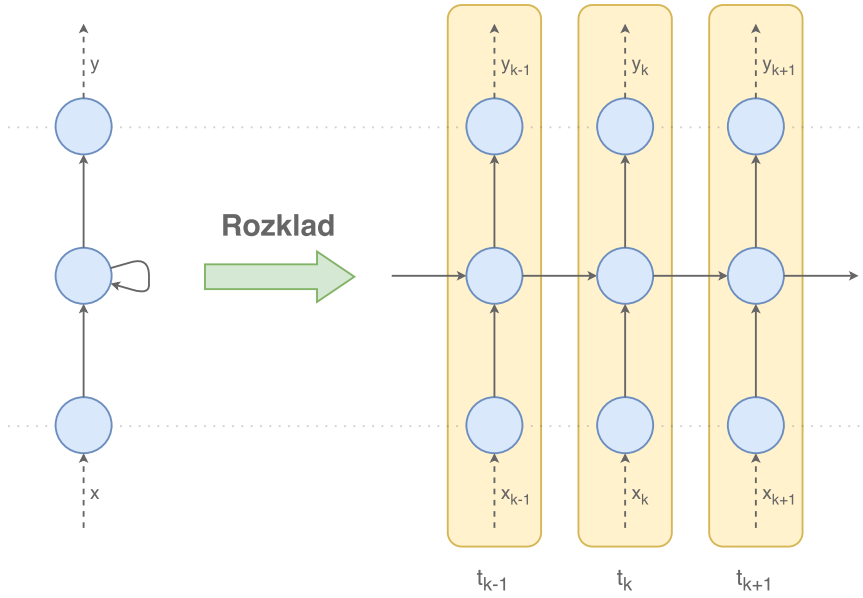
Dopředné neuronové sítě mají v kontextu predikce sekundární struktury výrazný nedostatek. A to stejný, jako další metody strojového učení zmiňované v kapitole č. 2.3, tedy zohledňování pouze jistého okolí predikovaného rezidua. Tento problém se podařilo překonat až pomocí rekurentních neuronových sítí.

3.2 Rekurentní sítě

Rekurentní sítě, oproti dopředným neuronovým sítím, dovolují orientované cykly v rámci grafu reprezentujícím jejich architekturu. S cykly v grafu sítě již není možné rozdělit neurony do vrstev stejným způsobem jako v případě dopředných sítí. Rozdělení do logických vrstev a určení jejich pořadí tedy provádí architekt sítě. Tím jsou určeny hrany v síti, které jdou směrem zpět v kontextu vrstev neuronů, případně končí i začínají ve stejné vrstvě. Takové hrany neumožňují provést výpočet stejným postupem, jako tomu bylo u dopředných sítí. Tyto hrany jsou proto zohledňovány až při zpracovávání následujícího vstupu sítě. Výpočet neuronové sítě lze tedy i v rámci rekurentních sítí provést jednosměrným průchodem po vrstvách sítě a zpětné hrany pouze zachovávají jistou míru informace o aktuálním vstupu pro zpracování následujícího vstupu. Výpočet provedený rekurentní neuronovou sítí si je možné představit i pomocí rozložení na dopřednou síť v čase. Pro sekvenci délky D a rekurentní neuronovou síť o N neuronech tak vznikne dopředná neuronová síť s $D \cdot N$ neurony, ve kterých se ale každý parametr d-krát opakuje. Rozklad neuronové sítě je znázorněn na obrázku č. 3.1. Právě rozložení rekurentní sítě podle času na dopředné sítě je přístup, jakým jsou tyto sítě učeny. Mohou pak totiž využívat klasické učící metody používané pro dopředné sítě.

Předávání informací mezi jednotlivými průchody neuronovou sítí umožňuje při výpočtu výstupu pro jeden vstup zohledňovat i předchozí vstupy sítě. To je naprosto klíčová vlastnost při práci se sekvenčními vstupy, u kterých historie vstupů hraje zásadní roli. To je případ i predikce sekundární struktury, kde se

[2]One-hot reprezentace aminokyselin je vektor o 21 prvcích se všemi hodnotami, krom právě jedné, nulovými. Každá z pozic v one-hot vektoru odpovídá jedné aminokyselině.



Obrázek 3.1: Znázornění rozkladu jednoduché rekurentní neuronové sítě s jednou cyklickou hranou na dopřednou síť. Rozklad je znázorněn pouze pro okolí sekvence v čase t.

tyto metody začaly využívat na přelomu století. Prvním pokusem o využití rekurentních neuronových sítí proběhlo na univerzitě v Kalifornii (Baldi a kol., 1999). Výstupem této práce byl, mimo jiné, také program SSPro[3], který lze v aktualizované verzi dodnes k predikci sekundární struktury použít. Základem tohoto modelu byly dvě symetrické rekurentní neuronové sítě, každá čtoucí vstupní proteinovou sekvenci z druhého konce. Výstup z obou sítí byl následně kombinován do finální predikce struktury. Samotná práce ale představovala spíše ověření konceptu a model nebyl schopen překonat ostatní používané metody. To ale autoři napravili v rámci navázání na původní práci (Pollastri a kol., 2002) s novým modelem SSPro, který již představoval novou state-of-the-art metodu.

Základní rekurentní sítě sice teoreticky mohou řešit i problémy vyžadující zohlednění dlouho trvajících závislosti, ale nalézt vhodnou sadu vah a prahů pro všechny neurony pro takový problém pomocí gradientního sestupu je velmi problematické (Bengio a kol., 1994). To vedlo k vývoji neuronových sítí které se snaží tento problém odstranit. Jednou z takových sítí je LSTM (Long Short Term Memory) (Hochreiter a Schmidhuber, 1997).

[3]Prediktor ve verzi 4.0 je aktuálně dostupný na adrese `http://download.igb.uci.edu/sspro4.html`

3.2.1 LSTM

Neuronová síť LSTM je speciálním případem rekurentní neuronové sítě navržené za účelem učení dlouho trvajících závislostí. Celý LSTM modul rozkládaný v čase se nazývá paměťová buňka. Ta disponuje vnitřním stavem, který se mění v čase v závislosti na sekvenci vstupních dat. O modifikaci vnitřního stavu se původně starala jednotka zvaná vstupní brána. LSTM byla později vylepšena o druhou bránu modifikující vnitřní stav, tzv. bránu pro zapomínání (Gers a kol., 2000). V rámci této práce budeme pracovat pouze s vylepšenou paměťovou buňkou LSTM. Buňka dále obsahuje třetí bránu nazvanou výstupní brána. Ta slouží k modifikaci vnitřního stavu před výstupem z buňky.

Pro požadovanou dimenzi l výstupního vektoru z paměťové buňky jsou jednotlivé brány tvořeny jednou vrstvou l neuronů plně propojenými se všemi vstupy. Vstupy bran v čase t jsou tvořeny konkatenací vstupního vektoru $x_t \in \mathbb{R}^k$ a vektoru $h_{t-1} \in \mathbb{R}^l$ reprezentujícího výstup buňky v předchozí časové iteraci. Aktivační funkcí neuronů ve všech branách je funkce sigmoida. Formálně můžeme výpočet výstupu z jednotlivých bran popsat následovně.

$$i_t = \sigma(W_i \cdot [x_t,h_{t-1}] + b_i)$$

$$f_t = \sigma(W_f \cdot [x_t,h_{t-1}] + b_f)$$

$$o_t = \sigma(W_o \cdot [x_t,h_{t-1}] + b_o)$$

kde i_t je výstup z vstupní brány v čase t, f_t je výstup z brány pro zapomínání v čase t a o_t je výstup z výstupní brány v čase t. Matice W_i, W_f a $W_o$jsou matice vstupních vah neuronů v branách a vektory b_i, b_f a b_o jsou vektory jejich prahů. Platí tedy, že W_i,W_f,$W_o \in \mathbb{R}^{k+l \times l}$ a b_i,b_f,$b_o \in \mathbb{R}^l$. Funkce σ je sigmoida a operace $\cdot$ je maticové násobení. Všechny výstupy z bran mají dimenzi l a všechny jejich hodnoty jsou z definice sigmoidy v rozsahu (0,1).

Výstupy ze vstupní brány a brány pro zapomínání se dále používají k modifikaci vnitřního stavu buňky. Nejdříve je vektor $C_{t-1} \in \mathbb{R}^l$ reprezentující vnitřní stav předchozího časového úseku po prvcích přenásoben s výstupem brány pro zapomínání f_t. Ta tedy řídí míru zachování předchozího vnitřního stavu. Následně je pomocí další vrstvy l neuronů s $tanh$ aktivační funkcí vypočítán vektor $\widetilde{C}_t \in \mathbb{R}^l$ reprezentující navrhovanou změnu vnitřního stavu. Vstup této sítě je stejně jako u bran buňky tvořen konkatenací vstupního vektoru $x_t \in \mathbb{R}^k$ a vektoru C_{t-1}. Vektor $\widetilde{C}_t$ je následně po prvcích přenásoben s výstupním vektorem vstupní brány,

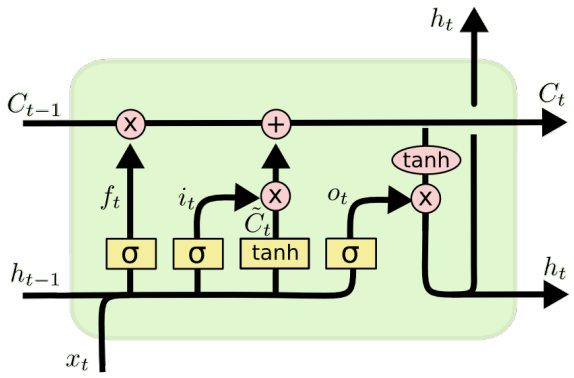
která řídí míru zápisu navrhované změny do vnitřního stavu. Součtem po prvcích vektorů vzniklých z C_{t-1} a $\widetilde{C}_t$ transformací pomocí výstupů ze zmíněných bran vzniká výsledný vnitřní stav C_t. Po naškálování stavu C_t funkcí *tanh* a přenásobením po prvcích s výstupem z výstupní brány o_t vnikne celkový výstup $h_t \in \mathbb{R}^l$ buňky v daném čase t. Formálně lze tedy výpočet nového stavu C_t a výstupu h_t popsat následovně.

$$\widetilde{C}_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c)$$

$$C_t = i_t * \widetilde{C}_t \oplus f_t * C_{t-1}$$

$$h_t = o_t * \tanh(C_t)$$

kde $*$ je operace násobení vektorů po prvcích a $\oplus$ odpovídá sčítání vektorů po prvcích. Matice $W_c \in \mathbb{R}^{k+l} \times l$ a vektor $b_c \in \mathbb{R}^l$ reprezentují matici vah vstupů neuronové vrstvy resp. prahy neuronů. Schéma paměťové buňky je znázorněno na obrázku č. 3.2.



Obrázek 3.2: Architektura paměťové buňky LSTM v čase t. Červená kolečka odpovídají operacím nad vektory. Žluté obdélníky odpovídají neuronovým vrstvám. Zdroj: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

Schopnost rozpoznat a naučit se dlouho trvající závislosti pomocí LSTM vedla k překonání výsledků tradičních rekurentních sítí v mnoha oborech. LSTM se tak stala state-of-the-art metodou oblastech rozpoznání řeči (Graves a kol., 2005a,b), rozpoznání sekvence písma psaného rukou (Graves a kol., 2008; Liwicki a kol., 2007) nebo např. kompresi textu (Sakti a kol., 2015).

Predikce sekundární struktury nezůstala výjimkou a první publikovaná práce (Sønderby a Winther, 2014) využívající LSTM se stala metodou dosahujících nejvyšších Q_8 skóre na veřejně dostupném datasetu `CB513`. Model využitý v této

práci byl složen ze tří vrstev obsahujících dvě LSTM paměťové buňky zapojené paralelně a čtoucí vstupní proteinovou sekvenci z obou stran (každá buňka z jedné strany). Výstup z poslední LSTM vrstvy byl následně předán dvěma plně propojeným vrstvám neuronů, kde proběhla výsledná klasifikace. Druhou a také aktuálně poslední publikací v oblasti predikce sekundární struktury je práce z univerzity v Brisbane (Heffernan a kol., 2017). Ta využívá velmi podobnou architekturu, odlišuje se především využitím pouze dvou vrstev obousměrných LSTM buněk. Dále publikovaný model využíval rozšířené předzpracování proteinů využívající skryté markovovské modely a tzv. iterované učení, kdy každý vstup procházel neuronovou sítí celkem čtyřikrát. Práce se ale zaměřovala pouze na predikci S_3 sekundární struktury a využívala méně rozšířená data, nelze tedy dosažené výsledky přímo srovnávat s podobnými metodami.

LSTM se stalo předlohou pro řadu podobných architektur. Jednou z aktuálně nejúspěšnějších je GRU (Gated Recurrent Unit) (Cho a kol., 2014).

3.2.2 GRU

GRU je rekurentní sítí s velmi podobnou architekturou jako LSTM. GRU se od LSTM odlišuje především sloučením vnitřního stavu s výstupem buňky, počtem bran a jejich významem. Místo tří bran disponuje pouze dvěma, a to branami pro aktualizaci stavu a zapomínání. Brána pro zapomínání funguje téměř identicky jako v případě LSTM. Výstup r_t z této brány v čase t reprezentuje míru zapomínání vnitřního stavu. Nový kandidát na vnitřní stav h_t, a tedy i výstup z buňky, je podobně jako v LSTM spočítán další skrytou vrstvou l neuronů s *tanh* aktivační funkcí. Oproti LSTM ale tato vrstva nepracuje na vstupu přímo s předchozím vnitřním stavem h_{t-1}, ale s jeho verzí po procesu zapomínání. Nový vnitřní stav a také výstup z buňky h_t v čase t je pak vypočítán váženým průměrem po složkách předchozího stavu h_{t-1} a kandidáta na nový stav $\widetilde{h_t}$. Váhy váženého průměru jsou učeny výstupem z brány pro aktualizaci stavu z_t. Formální popis výpočtu následuje.

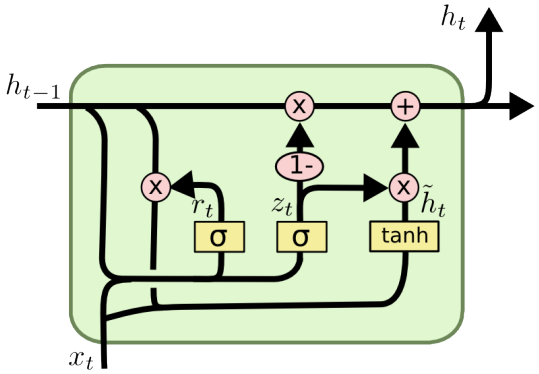
$$z_t = \sigma(W_z \cdot [x_t,h_{t-1}] + b_z)$$

$$r_t = \sigma(W_r \cdot [x_t,h_{t-1}] + b_r)$$

$$\widetilde{h_t} = \tanh(W_h \cdot [x_t,r_t * h_{t-1}] + b_h)$$

$$h_t = (z_t * \widetilde{h}_t) \oplus \left((\overrightarrow{1} \ominus z_t) * h_{t-1} \right)$$

kde $\ominus$ je operace odčítaní vektorů po složkách a $\overrightarrow{1}$ je vektor se všemi hodnotami rovnými jedné. Zbylé značení je stejné jako u popisu LSTM. Pro vstupní vektor $x_t \in \mathbb{R}^k$ a požadovanou dimenzi výstupu z buňky l pro matice vstupních vah skrytých vrstev resp. pro vektory prahů skrytých vrstev platí, že W_z,W_r,$W_h \in \mathbb{R}^{k+l \times l}$ resp. b_x,b_r,$b_h \in \mathbb{R}^l$. GRU tedy oproti LSTM vyžaduje o 25% méně parametrů, které je potřeba nalézt pomocí učení. Schéma paměťové buňky GRU je znázorněno na obrázku č. 3.3.



Obrázek 3.3: Architektura paměťové buňky GRU v čase t. Červená kolečka odpovídají operacím nad vektory. Žluté obdélníky odpovídají neuronovým vrstvám. Zdroj: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

Některé studie (Collins a kol., 2016; Chung a kol., 2014) ukázaly, že právě počet parametrů a rychlost učení je hlavní výhodou GRU oproti LSTM. Nicméně rekurentní sítě typu GRU jsou poměrně nové a skutečná šíře jejich možností není zatím dobře prozkoumána. V souvislosti s predikcí sekundární struktury byla publikována pouze jedna studie (Li a Yu, 2016). Ta je více přiblížena až v následující sekci.

3.3 Hluboké neuronové sítě

Ačkoliv neuronové sítě disponující pouze jednou skrytou vrstvou jsou teoreticky dostatečně silné nástroje na aproximaci libovolné funkce, na obrovské popularitě nabyly hlubší neuronové sítě. Těm se v posledních letech podařilo překonat dosavadní metody ve většině oborů, kde našly uplatnění také jednodušší neuronové sítě. Intuitivní ideou hlubokých neuronových sítí je představa, že každá vrstva neuronů extrahuje jinou sadu vlastností vstupu na základě výstupu

z předchozí vrstvy. Každá vrstva tedy představuje jinou úroveň abstrakce. Skládání více vrstev za sebe s sebou ale nese jisté problémy. Díky způsobu zpětné propagace chyby při tradičním učení s učitelem docházelo jen k malým úpravám nejnižších vrstev, které tedy nebylo možné spolehlivě učit. Tento problém se nazývá problémem mizení gradientu (Hochreiter, 1991). Problém se ale s narůstajícím výkonem počítačů zmenšoval. Výrazný posun také přišel přesunem učení neuronových sítí na GPU, které díky efektivnějším maticovým operacím rapidně zrychlilo dobu učení. Za rozmachem hlubokých neuronových sítí stály ale také nové způsoby učení nebo způsoby snižování počtu parametrů neuronů v jednotlivých vrstvách, což opět zvyšovalo možnosti učení. Nejpoužívanějším způsobem redukce parametrů jsou konvoluční sítě (Fukushima, 1980).

3.3.1 Konvoluční sítě

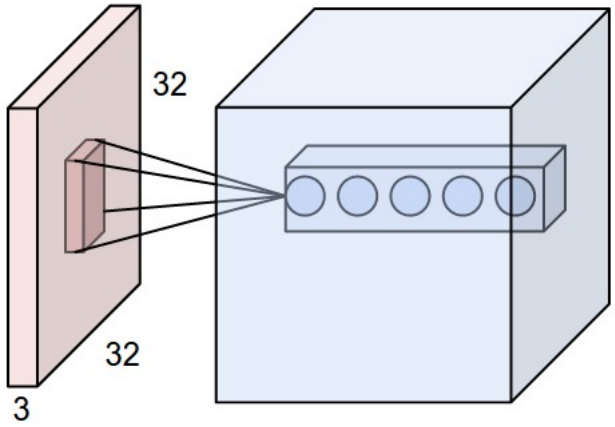
Konvoluční neuronové sítě jsou speciální případem dopředných neuronových sítí. Jejich základním principem je ale sdílení vah mezi neurony, díky čemuž je možné je trénovat standardními metodami i při použití více vrstev s rozměrným vstupem. Konvoluční sítě typicky využívají tři druhy neuronových vrstev, a to konvoluční, vzorkovací (pooling) a plně propojené vrstvy. Vstupem sítě je typicky trojrozměrné pole. Pro následující popis označme šířku, výšku a hloubku vstupu proměnnými w, h a d.

Konvoluční vrstva se skládá z tzv. filtrů, které se vstupem pracují vzájemně nezávisle. Filtr transformuje každý sloupec vstupu o rozměru $1 \times 1 \times d$ na jednu hodnotu. Výstupem každého filtru je tedy dvojrozměrné pole o rozměru $w \times h$ nazývané aktivační mapa filtru. Ten při transformaci nepracuje se samotným sloupcem, ale s celým okolním kvádrem o rozměru $r \times r \times d$, kde r je parametr konvoluční vrstvy nazývaný zorné pole filtrů. Každý filtr je reprezentován jedním neuronem o $r \times r \times d$ vstupech. Transformace každého kvádru tedy probíhá pomocí standardního výpočtu neuronů.

$$F(x_1, \ldots, x_{r \cdot r \cdot d}) = \Phi\left(\sum_{i=1}^{r \cdot r \cdot d} w_i \cdot x_i - b\right)$$

kde Φ je aktivační funkce a hodnoty $x_1, \ldots, x_{r \cdot r \cdot d}$ jsou prvky kvádru $r \times r \times d$. Aby byl rozměr výstupu filtru skutečně $w \times h$ je nutné rozšířit vstup vrstvy na rozměr $w+r-1 \times h+r-1 \times h$ například vkládáním nulových hodnot. Při zarovnání nulovými hodnotami a počtem filtrů f je výstupem z konvoluční vrstvy trojrozměrné pole o rozměru $w \times h \times f$. Ideou konvoluční vrstvy je využívání stejného filtru, a tedy i stejné matice vah vstupů a prahu, po celém vstupu. Nezáleží tedy

na pozici vstupu, nad kterou se filtr nachází. To oproti klasické dopředné neuronové sítí přináší obrovskou redukci parametrů vrstvy. Schéma práce konvoluční vrstvy je znázorněna na obrázku č. 3.4.

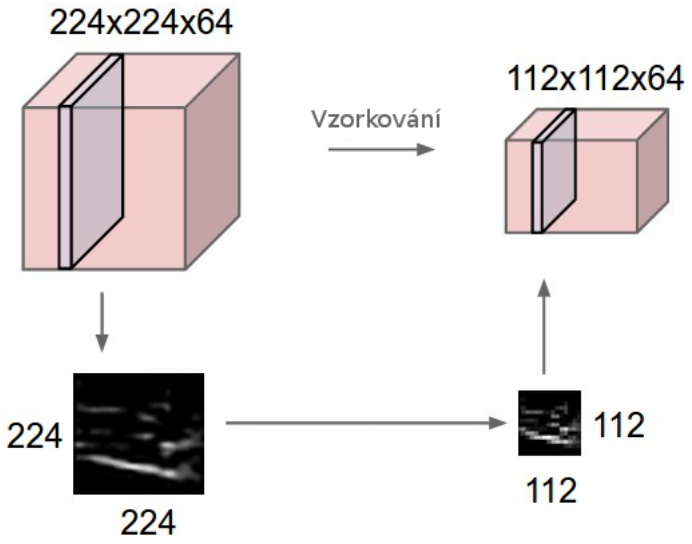


Obrázek 3.4: Schéma práce konvoluční vrstvy v konvolučních neuronových sítích. Jednotlivé filtry jsou zde reprezentovány modrými kolečky. Zdroj: `http://cs231n.github.io/convolutional-networks/`.

Vzorkovací vrstva slouží k redukci šířky a výšky průchozích dat. Vrstva pracuje s každou hladinou vstupu ve směru hloubky nezávisle. Pro zvolený rozsah r a posun s je každá hladina vstupu rozdělena na čtvercové segmenty $r \times r$ postupným posouváním čtverce ve směru šířky a výšky o s jednotek. Každá hladina je tedy rozdělena na $(\frac{(w-r)}{s} + 1) \cdot (\frac{(h-r)}{s} + 1)$ čtverců. Každá ze čtvercových matic je pak transformována na jediného číselného reprezentanta. Typicky používanými transformačními metodami je např. průměr hodnot nebo jejich maximum. Pro vstup o rozměru $w \times h \times d$ je výstup ze vzorkovací vrstvy kvádr o rozměrech $\frac{(w-r)}{s} + 1 \times \frac{(h-r)}{s} + 1 \times d$. Vzorkovací vrstva je tedy definována parametry r, s a funkcí transformace čtvercových matic a neobsahuje žádné parametry, které je nutné určit pomocí učení. Způsob práce vzorkovací vrstvy je zobrazen na obrázku č. 3.5.

Konvoluční neuronové sítě se nejčastěji spojují s analýzou obrázků, kde také dosahují nejlepších výsledků. Podařilo se například dosáhnout rekordních výsledků na datasetu ručně psaných číslic MNIST (Ciresan a kol., 2012), kde konvoluční sítě dosahují výsledků srovnatelnými s rozlišovacími schopnostmi lidí nebo také v klasifikaci velkého množství rozměrných obrázků (Krizhevsky a kol., 2017).

Konvoluční sítě ale v posledních 2 letech našly i výrazné uplatnění v oblasti predikce sekundární struktury a staly se v této oblasti jednou z nejúspěšnějších metod. Jedna z prvních prací využívající konvoluční sítě popisovala predikční model DeepCNF (Wang a kol., 2016). Pro tento model byly proteiny před vstupem



Obrázek 3.5: Schéma práce vzorkovací vrstvy v konvolučních neuronových sítích. Vstup o rozměru $224 \times 224 \times 64$ je vzorkovací vrstvou s rozsahem r i posunem s rovným 2. Zdroj: `http://cs231n.github.io/convolutional-networks/`.

do sítě transformovány do pole rozměrech $n \times 1 \times k$, kde n je délka proteinu a k je délka vektoru popisujícího dané reziduum. Konkrétně v případě DeepCNF se bylo k rovno 42, kde celé pole odpovídalo spojení matic PSSM a one-hot reprezentace aminokyselin. Každé vstupní proteinové pole pak procházelo 5 konvolučními vrstvami o 100 filtrech. Finální modelování sekundárních struktur z výstupu z poslední konvoluční vrstvy bylo provedeno pomocí podmíněnými neuronovými poli (Conditional Neural Fields) (Peng a kol., 2009). DeepCNF na veřejném datasetu `CB513` dosáhl v té době nejlepších výsledků v rámci Q_3, Q_8 i SOV_3 skóre.

Další publikovanou prací využívající konvoluční vrstvy byla práce z Univerzity v Hongkongu (Li a Yu, 2016). Té se na stejném datasetu podařilo DeepCNF překonat a dosáhnout vyššího Q_8 skóre. Model pracoval se stejným formátem vstupu jako DeepCNF, ale využíval pouze tři konvoluční vrstvy a to dokonce pouze paralelně zapojené. Výstupy z těchto vrstev byly předány rekurentnímu bloku využívající tři vrstvy oboustranných GRU jednotek. Výstup z rekurentního bloku byl dále předán dvou plně propojeným neuronovým sítím, které zajišťovaly finální klasifikaci. Právě ale GRU byly podle autorů hlavním zdrojem zlepšení.

Ještě lepší výsledek na stejném datasetu byl na konci roku 2016 dosažen v rámci projektu Google Brain Residency (Busia a kol., 2016). Základní model byl postaven především na sérii dvou bloků skládajících se ze tří paralelně zapojených konvolučních vrstev a jedné sériově zapojené konvoluční vrstvě. Síť byla opět zakončené dvěma plně propojenými vrstvami neuronů. Výsledek se podařilo dále vylepšit pomocí složeného modelu využívajícího podmíněné neuronové pole

jako v případě DeepCNF.

V posledních několika letech zaznamenala predikce sekundární struktury výraznou proměnu požívaných metod především přechodem na metody hlubokého učení. To samozřejmě přineslo zkvalitnění predikcí, nicméně všechny tyto metody mají stále poměrně daleko k teoretickému maximu přesnosti predikce, které je odhadováno přibližně na 90% v případě Q_3 i SOV_3 skóre (Rost, 2001; Rost a kol., 1994). Proto se tato práce zaměřuje na vylepšení jedné z těchto metod. Navíc se oproti velké většině předchozích prací zaměřuje tato práce také na studii vstupů a předzpracování proteinů.

4. Výzkumná část

Výzkumná část této práce bude složena ze čtyř oddílů. V první části implementujeme vybranou architekturu hlubokých neuronových sítí pro predikci sekundární struktury. Na tomto výchozím modelu budou postavené zbylé tři části. Ve druhé části se pokusíme navrhnout a implementovat možná vylepšení pro klíčové komponenty výchozího modelu. Následně se ve třetí části zaměříme na předzpracování primární struktury před vstupem do neuronové sítě. Poslední část se naopak zaměří na zpracování výstupů z neuronové sítě, a to vyhlazováním výstupu.

4.1 Implementace výchozího modelu

Jako výchozí architekturu neuronové sítě pro tuto práci jsme zvolili architekturu DCRNN (Li a Yu, 2016) vytvořenou na Univerzitě v Hongkongu v roce 2016. Klíčovým kritériem výběru byla skutečnost, že podle všech nám dostupných informací se jednalo o model dosahující nejvyšších Q_n přesností predikce. V současné době se ale již nejedná a state-of-the-art metodu, jelikož na konci roku 2016 byla překonána inženýry z programu Google Brain Residency (Busia a kol., 2016).

Model DCRNN pracuje s klasickou sadou vstupů. Každé reziduum proteinu je reprezentováno spojením dvou vektorů. První vektor o 21 prvcích reprezentoval danou aminokyselinu pomocí tzv. one-hot kódování[1]. Druhý vektor znovu o délce 21 prvků byl tvořen odpovídajícím sloupcem PSSM získané pomocí 3 iterací programu PSI-BLAST nad databází UniRef90 (Suzek a kol., 2015) s E-hodnotou 0.001. Všechny prvky této matice jsou před použitím ještě naškálovány do intervalu (0,1) pomocí funkce sigmoida. Pro protein o n reziduích je tvořen vstup pro predikční model maticí o rozměru $[n \times 42]$.

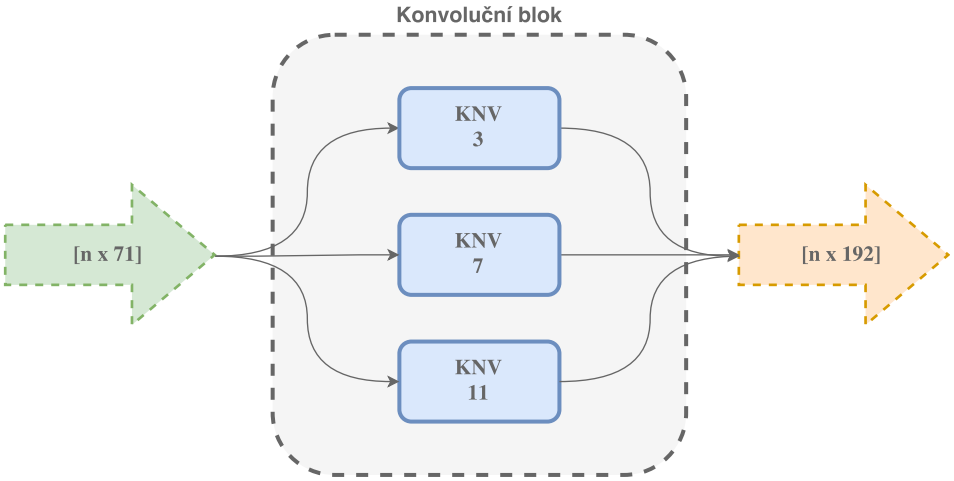
Výstupem z DCRNN není pouze predikce sekundární struktury S_8, ale také predikce přístupnosti rozpouštědla k danému reziduu. To je vlastnost rezidua, která velmi úzce souvisí právě se sekundární strukturou. Proto je také jejich predikce často spojována do jednoho predikčního modelu, což vede k rychlejšímu učení a lepší úspěšnosti predikce (Qi a kol., 2012).

[1] One-hot vektor je vektor obsahující právě jeden prvek s hodnotou 1, zbylé prvky jsou 0.

4.1.1 Architektura DCRNN

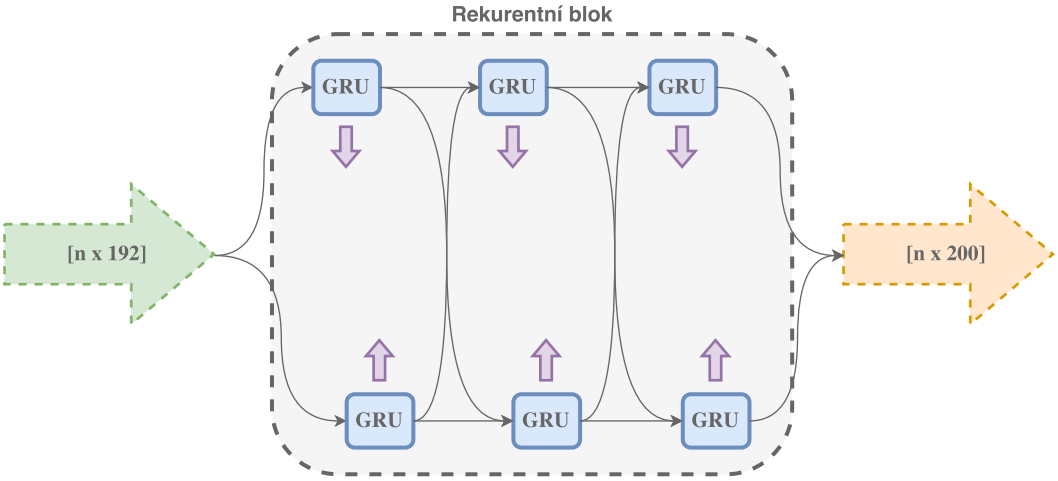
Model DCRNN se skládá z čtyř různých logických bloků neuronových vrstev. Těmi jsou blok pro zakódování vektorů reprezentujících aminokyseliny a dále konvoluční, rekurentní a klasifikační blok. Schéma posledních tří bloků je znázorněno na obrázcích 4.1, 4.2 a 4.3. Vrstva pro zakódování vstupu zajišťuje transformaci one-hot vektoru reprezentujícího jednu z 21 aminokyselin na dané pozici v proteinu do hustého vektoru o 50 prvcích. To je provedeno jednou plně propojenou vrstvou obsahující 50 neuronů bez prahových hodnot a s lineární aktivační funkcí. Tato praktika je převzata z oblasti zpracování přirozeného jazyka, kde se ukázala jako velmi přínosná k zakódování jednotlivých slov (Mesnil a kol., 2015).

Konvoluční blok obsahuje tři různé paralelní konvoluční vrstvy. Každá z konvolučních vrstev na vstup přijímá výstup z bloku pro zakódování aminokyselinového vektoru spojený s druhou vstupní maticí reprezentující PSSM. Celkově má tedy vstupní matice rozměr $[n \times 71]$. Tuto matici každá z konvolučních vrstev modifikuje pomocí 64 filtrů. Vrstvy je liší velikostí plovoucího okna, které berou v potaz. Autoři zvolili okna velikostí 3, 5 a 7 sousedících reziduí o aktivační funkci $ReLU$. Každá z těchto vrstev tedy bere v potaz pouze několik málo sousedících reziduí a slouží k extrakci pouze lokální kontextu jednotlivých reziduí. Výstupy z jednotlivých vrstev jsou spojeny za sebe a vytvářejí tak pole o rozměru $[n \times 192]$.



Obrázek 4.1: Schéma konvolučního bloku DCRNN.

Výstup z konvolučního bloku je předán rekurentnímu bloku, který obsahuje tři rekurentní neuronové vrstvy. Každá z těchto vrstev je složena ze dvou GRU, které čtou vstupní sekvenci, každá z opačného konce. Každá z jednotek GRU obsahuje celkem 300 skrytých neuronů, výstup každé z těchto jednotek má tedy dimenzi 100. Výstupem z každé vrstvy je spojení výstupů obout jednotek, výstupem celého bloku je výstup poslední rekurentní vrstvy, tedy matice $[n \times 200]$.



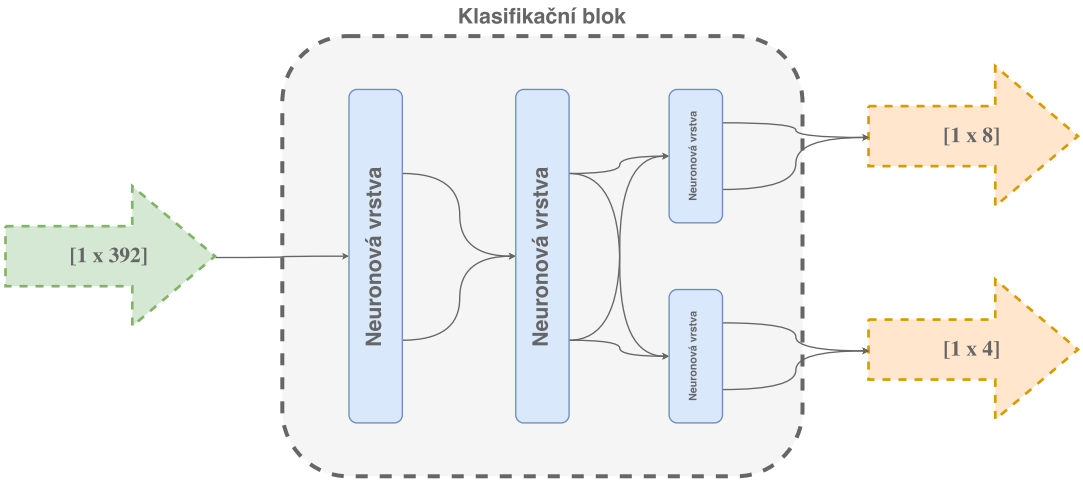
Obrázek 4.2: Schéma rekurentího bloku DCRNN.

Na výstupní matici z rekurentní vrstvy je aplikováno vynulovávání náhodné poloviny ze všech prvků. To pochopitelně komplikuje proces učení sítě, ale jedná se o jednoduchý a účinný způsob, jakým lze omezit přeučování velkých neuronových sítí (Srivastava a kol., 2014). Vynulování náhodně vybraných prvků matice probíhá ale pouze ve fázi učení sítě, během skutečné predikce se stává tato vrstva neaktivní.

Posledním blokem modelu je klasifikační blok, jehož vstup je tvořen spojením výstupních matic z konvoluční a rekurentní vrstvy (matice o rozměru $[n \times 392]$). Se vstupem blok již nepracuje jako s celkem, ale pouze na úrovni jednotlivých vektorů vztahujícím se k jednotlivým reziduím. Tedy vektorům o délce 392 prvků. Blok obsahuje dvě plně propojené vrstvy neuronů s *ReLU* aktivační funkcí, na které je napojeno celkem 12 výstupních neuronů. Výstupní sada neuronů je rozdělena do dvou skupin, čtyři pro predikci přístupnosti rozpouštědla a osm pro predikci sekundární struktury. Aktivační funkcí jednotlivých výstupních skupiny je funkce softmax. Ta zajišťuje korektní pravděpodobnostní rozdělení na výstupu z obou skupin.

4.2 Úprava modelu

Model získaný reimplementací architektury DCRNN využijeme jako výchozí model k dalšímu postupu. Konkrétně se provedením několika modifikací architektury DCRNN pokusíme dosáhnout zlepšení v Q_8^U nebo SOV_8 skóre. Zaměříme se především na modifikaci dvou klíčových komponent, tedy bloků pro extrakci lokálního a globálního kontextu. Jak bylo popsáno výše, lokální kontext je získáván pomocí tří paralelních konvolučních vrstev s různými velikostmi zorných



Obrázek 4.3: Schéma klasifikačního bloku DCRNN.

polí. Ale v rámci jiných publikací (Wang a kol., 2016; Li a Yu, 2016; Busia a kol., 2016) bylo ukázáno, že velmi dobrých výsledků lze dosáhnout využitím několika sériově zapojených vrstev. Takovým způsobem se také snadno rozšíří rozsah, ze kterého je lokální kontext získáván. Pro n sériově zapojených konvolučních vrstev se zorným polem filtrů r je celkový rozsah roven r^n. Tato možnost tedy bude v rámci této práce prozkoumána.

Především se ale zaměříme na modifikaci rekurentního bloku, který slouží k extrakci globálního kontextu. Výchozí model v tomto bloku využívá sérii GRU. My se pokusíme v rekurentním bloku využít větší paměťové buňky typy LSTM. Podle studie publikované na Univerzitě v Montreálu (Chung a kol., 2014) nelze v obecné rovině rozhodnout, která z paměťových buněk dosahuje lepších výsledků. Nicméně bylo také ukázáno (Cravens a Probert, 2016), že neuronové sítě využívající LSTM dosahovaly lepších výsledků při predikci α-helixové struktury než ekvivalentní síť využívající GRU. V kontextu predikce celé sekundární struktury proteinů byla LSTM již několikrát úspěšně využita. Žádná z předchozích prací ale nevyužívala doplňující prvky k extrakci lokálního kontextu před vstupem do LSTM.

4.3 Vstupy modelu

Mimo architektury modelu je také možné zkoumat přínosnost metod předzpracování primární struktury před vstupem do predikčního modelu. Právě tomu se budeme věnovat v této části, a to ze dvou hledisek. Nejdříve se zaměříme na úpravu stávající metody předzpracování proteinu. Stávající metoda je totiž časově řádově náročnější než samotný výpočet provedený neuronovou sítí, pokusíme

se tedy o její zrychlení. V druhé části prozkoumáme možnost rozšíření stávající sady vstupů. Ta se pro predikční modely téměř výhradně ustálila na zakódované proteinové sekvenci a PSSM. My prozkoumáme možnost zahrnutí mezi vstupy také predikci terciární struktury.

4.3.1 Zrychlení předzpracovaní dat

Mimo samotné zakódované primární struktury proteinu se jako vstup do predikčních modelů nejčastěji přidává informace o konzervovanosti v podobě PSSM získané ze zarovnání homologních sekvencí. K tomu se tradičně využívá několik iterací programu PSI-BLAST, který PSSM přímo využívá pro hledání homologních sekvencí v proteinové databázi. Pro omezení počtu nalezených sekvencí, a tedy i rychlosti výpočtu jejich zarovnání, se nejčastěji využívají proteinové databáze s nízkou redundancí sekvencí. Typickým příkladem mohou být databáze UniRef (Suzek a kol., 2015), které nahrazují skupiny podobných sekvencí pouze jedním reprezentantem. Tyto databáze ale stále obsahují desítky milionů proteinů. To dělá výpočet PSSM velmi obtížný a na běžném počítači tak může tento výpočet trvat i přes 12 hodin. Samotný výpočet provedený neuronovou sítí lze snadno provést během jedné vteřiny, výpočet PSSM je tedy naprosto klíčový faktor pro rychlost predikce.

Jednou z možností, jak tento výpočet urychlit je snížit počet iterací programu PSI-BLAST. Vyšší počet iterací programu PSI-BLAST umožňuje nacházet v databázi i evolučně vzdálenější proteiny v databázi, tedy snížením počtu iterací bude výsledná PSSM hůře popisovat konzervovanost daného proteinu. Druhou možností jak urychlit výpočet PSSM je využití menších proteinových databází. V takovém případě ale hrozí, že v menší databázi nebude nalezeno dostatek homologních sekvencí a výsledná konzervovanost bude touto skutečností velmi zkreslena.

Kombinace obou zmíněných postupů je využita v projektu cal_protein_conservation[2], který slouží k výpočtu konzervovanosti pomocí Jensen-Shannonovy divergence (Capra a Singh, 2007). Projekt ale nevyužívá pouze jednu databázi, ale pracuje se dvěma různými databázemi, mezi kterými přepíná podle nutnosti. Nejdříve probíhá hledání pomocí programu PSI-BLAST v menší databázi, nalezené sekvence se naklastrují pomocí programu CD-HIT (Fu a kol., 2012) s hranicí podobnosti 90%. Pokud je počet klastrů vytvořených z výsledků průchodu malou databází menší než 50, spustí se hledání ve velké databázi. Tímto je minimalizována doba výpočtu i pravděpodobnost, že PSSM bude vytvořena pouze

[2]Volně šiřitelný software pod licencí GPLv3 `https://github.com/jendelel/calc_protein_conservation`

z minimálního počtu proteinů.

Zmíněný projektu využijeme k naprogramování našeho vlastního nástroje k rychlému výpočtu PSSM. Toto zrychlení ale bude dosaženo na úkor přesnosti konzervovanosti, která je PSSM reprezentována. Pomocí výchozího modelu tedy také ověříme, do jaké míry má zrychlení výpočtu skutečný vliv na úspěšnost predikce.

4.3.2 Zahrnutí terciární struktury

Výchozí model pracuje s typickými vstupy pro většinu publikovaných metod v oblasti predikce sekundární struktury. V této části se pokusíme prozkoumat možnost rozšíření sady vstupů, konkrétně prozkoumáme možnost zahrnutí predikce samotné terciární struktury. Většina metod pro predikci terciární struktury během svého výpočtu ale využívá prediktory sekundární struktury. Ty většinou také využívají metody strojového učení a informace o datech, na kterých byly tyto metody naučeny, typicky nejsou dostupné. Využití těchto prediktorů terciární struktury by tedy nemuselo být korektní. Metody založené na konstrukci terciární struktury pomocí nalezených homologních sekvencí v PDB také použít nelze, jelikož by bylo nekorektní testovat takový přístup na proteinech, která již strukturu v PDB mají. To je ale předpoklad pro znalost sekundární struktury nutné k otestování této metody. Řešením by bylo využití predikcí založených pouze na minimalizaci energetické funkce. Tyto výpočty jsou ale extrémně náročné, a tak pro nás není možné využít ani tuto metodu. Výše zmíněné problémy jsou pravděpodobně hlavními příčinami, proč je zapojení predikce terciární struktury při predikci sekundární struktury proteinů neprozkoumaná oblast. Podle všech nám dostupných informací se této možnosti nikdo nevěnoval.

Místo pravé predikce terciární struktury je ale možné využít zjednodušený problém. Jedním takovým je například HP mřížkový model struktury proteinů. Tento model pracuje pouze s pozicemi centrálních uhlíků, která navíc umisťuje do předdefinované mřížky. To omezuje počet možných struktur proteinu a urychluje predikci struktury. To ovšem také způsobuje omezení přesnosti predikce. Za účelem minimalizace této chyby použijeme FCC (face-centered-cubic) mřížkový model, který nabízí největší volnost predikce ze všech používaných mřížkových modelů. FCC model pro každé reziduum nabízí 12 různých pozic oproti předchozímu reziduu[3], způsob rozmístění těchto pozic je znázorněn na obrázku č. 4.4. Bylo ukázáno (Park a Levitt, 1995), že FCC model je dostatečně komplexní model

[3]Pozic je během samotné predikce k dispozici pouze 11, jelikož 12. pozice je obsazena reziduem před předchozím reziduem.

k alespoň částečné reprezentaci sekundární struktury. Dalším zjednodušením, se kterým budeme pracovat, je zjednodušená energetická funkce (Jacob a kol., 2007). Ta pracuje pouze s binární informací o jednotlivých aminokyselinách v daném řetězci, a tou je polarita resp. hydrofobnost aminokyseliny. Příslušnost aminokyselin k těmto dvou skupinám je znázorněna v tabulce č. 4.1.

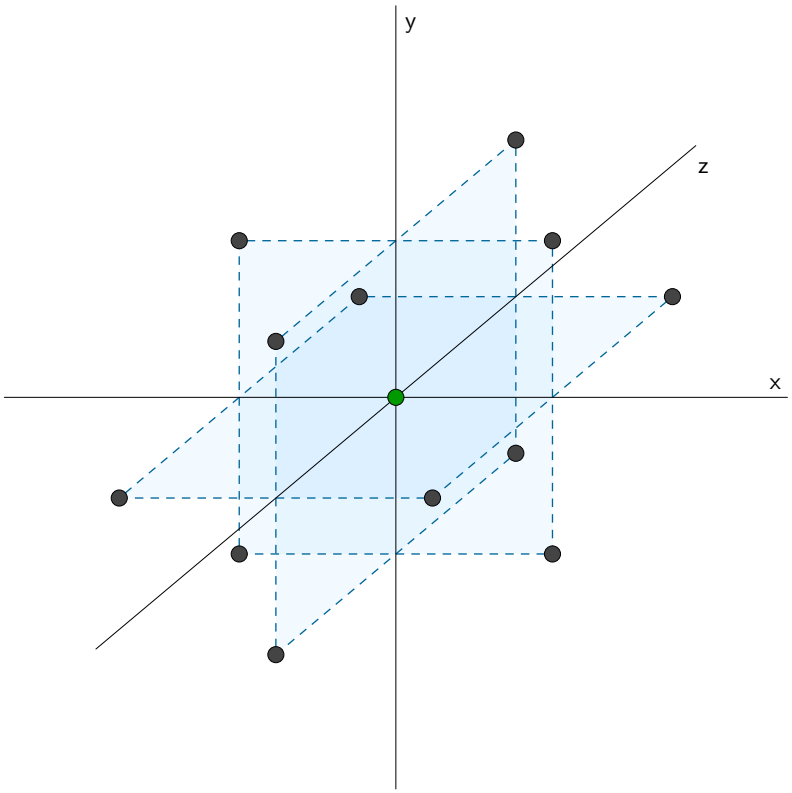
Aminokyselina	H/P	Aminokyselina	H/P	Aminokyselina	H/P
Alanin	H	Histidin	P	Prolin	H
Arginin	P	Isoleucin	H	Serin	P
Asparagin	P	K. asparagová	P	Threonin	P
Cystein	P	K. glutamová	P	Tryptofan	P
Fenylalanin	H	Leucin	H	Tyrosin	P
Glutamin	P	Lysin	P	Valin	H
Glycin	H	Methionin	P		

Tabulka 4.1: Seznam aminokyselin a jejich příslušnost mezi polární (P) resp. hydrofobické (H) aminokyseliny.

Tyto dvě zjednodušení mají již dostatečný vliv na snížení komplexity tohoto problému k tomu, aby bylo možné vypočítat predikci struktury pro dostatečné množství proteinů k naučení a otestování na výchozím modelu. K samotnému výpočtu HP mřížkové struktury použijeme program HPstruct (Mann a kol., 2008). Ten pomocí programování s omezujícími podmínkami a sadou předpočítaných šablon vypočítá FCC mřížkovou strukturu proteinu s minimální zjednodušenou energetickou funkcí. Program vypočítanou strukturu reprezentuje jako posloupnost hran mezi centrálními uhlíky reziduí, kde každá hrana je reprezentována jedním z 12 předdefinovaných absolutních směrů. Takový formát je ale nevhodný jako vstup do neuronové sítě, proto nejprve definujeme jednu z možných konverzí.

Konverze terciární struktury

Cílem této konverze bude převést strukturu do formátu, který neobsahuje žádné absolutní pozice či směry. Zároveň je potřeba převést posloupnost $n-1$ hran mezi jednotlivými centrálními uhlíky na posloupnost délky n tak, aby jednotlivé prvky posloupnosti reprezentovaly vztah jednotlivých reziduí k okolní



Obrázek 4.4: FCC model mřížkové struktury. Pro každé reziduum (zelený bod) existuje 12 různých pozic pro následující reziduum (šedé body). Možné nové pozice lze také reprezentovat vektory (1,1,0), $(-1,1,0)$, $(1,-1,0)$, $(-1,-1,0)$, (1,0,1), $(-1,0,1)$, $(1,0,-1)$, $(-1,0,-1)$, (0,1,1), $(0,-1,1)$, $(0,1,-1)$, $(0,-1,-1)$.

struktuře proteinu. Při konverzi vyjdeme z posloupnosti $p_0, p_1, \ldots, p_n$ souřadnic jednotlivých centrálních uhlíků reziduí. Tu snadno získáme umístěním prvního centrálního uhlíku na pozici (0,0,0) a výpočtem dalších pozic se směrů hran. Tuto posloupnost přepočítáme na finální posloupnost $(\alpha_1, \mu_1, d_1^\mu, \nu_1, d_1^\nu)$, $(\alpha_2, \mu_2, d_2^\mu, \nu_2, d_2^\nu)$, $\ldots$, $(\alpha_n, \mu_n, d_n^\mu, \nu_n, d_n^\nu)$ následovně:

Pro reziduum i reprezentuje α_i úhel sevřený hranami vedoucími z a do daného rezidua. Formálně lze tedy α_i definovat takto.

$$\alpha_i = \begin{cases} 0, & pokud \;\; i \in \{1,n\} \\ \arccos\left(\frac{(p_i - p_{i-1}) \odot (p_{i+1} - p_i)}{\|p_i - p_{i-1}\| \cdot \|p_{i+1} - p_i\|}\right)/\pi, & jinak \end{cases}$$

kde funkce $\odot$ je skalární součin vektorů.

Pro úplný popis struktury související s daným reziduem je ale také potřeba přidat rotaci hran vedoucích z a do daného rezidua. Rotace hrany je reprezentována dvojicí μ_n a d_n^μ. μ_n odpovídá úhlu sevřeného rovinami $\rho_i = \overline{p_{i-2}p_{i-1}p_i}$ a $\rho_{i+1} = \overline{p_{i-+}p_ip_{i+1}}$ a d_n^μ pak směru této rotace (pravotočivá rotace -1, levotočivá 1, jinak 0).

$$n_{\rho_i} = (p_{i-1} - p_{i-2}) \otimes (p_i - p_{i-1})$$

$$\mu_i = \begin{cases} 0, & pokud \;\; i \in \{1,2,n\} \\ \arccos\left(\frac{n_{\rho_i} \odot n_{\rho_{i+1}}}{\|n_{\rho_i}\| \cdot \|n_{\rho_{i+1}}\|}\right)/\pi, & jinak \end{cases}$$

$$d_i^\mu = \begin{cases} 0, & pokud \;\; i \in \{1,2,n\} \\ sign(n_{\rho_i} \odot p_{i+1} - n_{\rho_i} \odot p_i), & jinak \end{cases}$$

kde $\otimes$ je operace vektorový součin. Zde je potřeba ošetřit dva speciální případy, a to pokud velikost jednoho z normálových vektorů n_{ρ_i} nebo $n_{\rho_{i+1}}$ vyjde nulová. To odpovídá případu dvou sousedních hran se stejným směrem. V případě nulové $n_{\rho_{i+1}}$ budeme rotační úhel i směr považovat také za nulový. V případě nulového normálového vektoru n_{ρ_i} najdeme první předcházející pozici $p_j,\ j < i-2$ centrálního uhlíku, který neleží na přímce určené body p_i a p_{i-1}. Normálový vektor n_{ρ_i} předefinujeme pomocí nalezeného centrálního uhlíku na $n_{\rho_i} = (p_{i-1} - p_j) \otimes (p_i - p_{i-1})$ a dále pokračujeme podle předchozích rovnic. V případě, že žádná taková pozice p_j neexistujeme, budeme rotační úhel i směr považovat opět za nulový.

Rotace hrany vedoucí z rezidua je identická s rotací hrany vedoucí do následujícího rezidua. Lze tedy definici ν_n a d_n^ν snadno zapsat pomocí μ_n a d_n^μ následovně.

$$\nu_i = \mu_{n-i+1}$$

$$d_i^{\nu} = d_{n-i+1}^{\mu}$$

V tomto výpočtu není zohledněna různá délka hran, jelikož v FCC mřížkovém modelu jsou všechny hrany stejně dlouhé. Ale Rozšíření tohoto formátu o délku hran je samozřejmě možné. Tento formát má několik výhod. Neobsahuje žádné absolutní pozice nebo směry, všechny hodnoty jsou v rozmezí $\langle -1; 1 \rangle$ a vektor hodnot zastupující jednotlivá rezidua poskytuje symetricky informace o okolní struktuře proteinu v obou směrech. Ačkoliv poslední vlastnost je vyvážena redundancí dat v podobě ν_n a d_n^{ν}.

Tato konverze je inspirována reprezentací terciární struktury proteinů sekvencí dvou rotačních úhlů (ϕ, ψ). Tu ale nebylo možné přímo použít, jelikož predikce mřížkové struktury pracuje s aminokyselinami jako s dále nedělitelnými jednotkami. Nelze tedy rotační úhly (ϕ, ψ) spočítat, jelikož k tomu je vyžadována znalost pozice všech atomů v peptidické kostře.

4.4 Vyhlazovaní

Samotné predikce modelů bývají často předány další logické vrstvě, která zajišťuje funkci vyhlazování výstupu (Salamov a Solovyev, 1995; Wood a Hirst, 2005; Chen a Chaudhari, 2007). To může být jednoduchým řešením pro odstranění některých chyb v predikci, jako je například predikce α-helixové struktury o délce jednoho rezidua. Taková struktura není z principu, jakým je α-helix tvořen, možná a vyhlazovací metody mohou takovou chybu snadno odhalit a odstranit.

Nedávno publikovaná studie vyhlazovacích metod pro predikci sekundárních struktur (Kountouris a kol., 2012) ukázala, že některé metody vyhlazovaní mohou vylepšit Q_3 skóre prediktoru, ale především velká část testovaných metod zvýšila SOV_3 skóre. Nejlépe z tohoto srovnání vyšly metody SVM (Support Vector Machine), logistická regrese a neuronová síť. Mimo jiné studie zkoumala také kombinaci metod strojového učení a empirických pravidel, což vedlo k dalšímu mírnému vylepšení jednotlivých metrik oproti samotným výše zmíněným metodám.

Tato studie i předchozí práce využívající vyhlazování v rámci svých predikčních modelů byly postavené pouze na predikci jednoduššího rozdělení sekundárních struktur S_3. My se pokusíme aplikovat stejné postupy na náš výchozí model pro predikci sekundární struktury S_8 a pokusíme se tedy prozkoumat možnosti vyhlazování pro toto složitější rozdělení sekundárních struktur. K tomu využijeme zmíněné tři metody dosahující nejlepších výsledků ve zmíněné studii, tedy

SVM, logistickou regresi a neuronové sítě.

5. Experimenty a výsledky

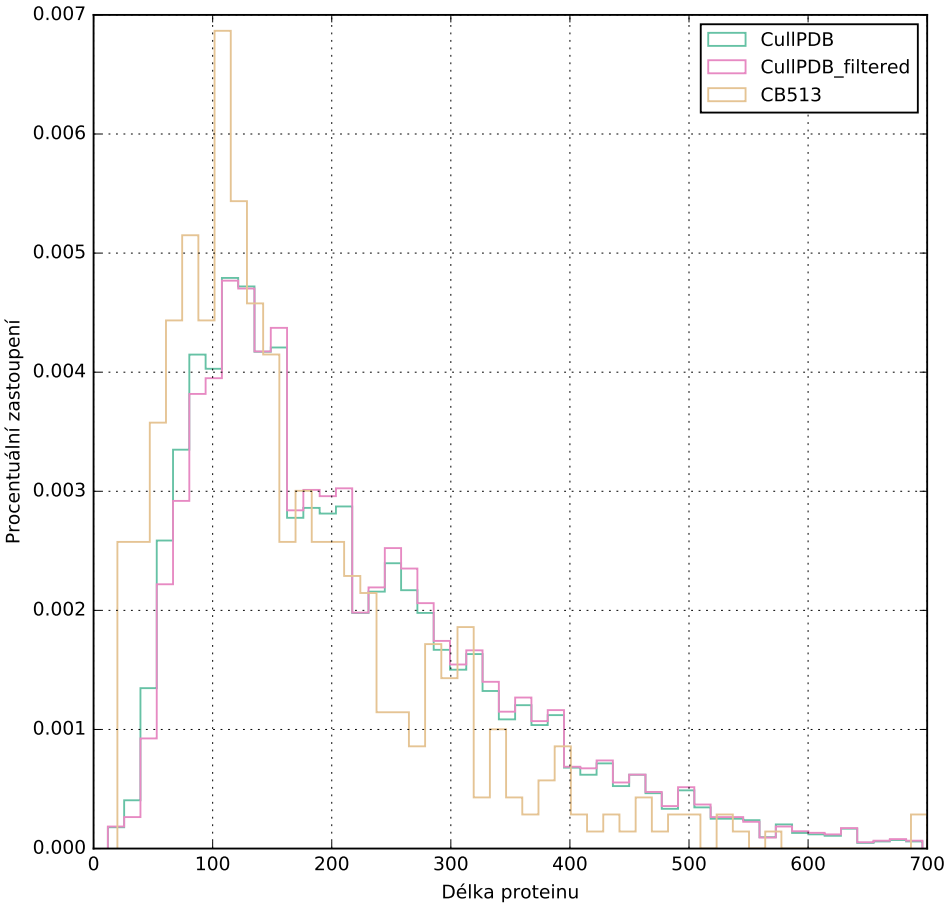
Všechny modely neuronových sítí byly implementovány nad knihovnou Theano (Theano Development Team, 2016) za pomoci knihovny Keras (Chollet a kol., 2015) v jazyce Python 2.7. Všechny parametry, které nejsou přímo specifikovány v této práci, byly nastaveny na výchozí hodnoty definované v knihovně Keras. Modely byly učeny pomocí metody Adam (Kingma a Ba, 2014) s kategorickou křížovou entropie jako chybovou funkcí. Obě tyto volby byly po sérii experimentů určeny jako nejvhodnější. Rychlost učení byla nastavena na 0.001 a po každých 10 epochách učení, během kterých nedošlo k poklesu chybové funkce na validačních datech, byla rychlost učení zmenšena na polovinu. Pro umožnění rychlejšího učení neuronové sítě využitím maticových operací byl každý vstup do neuronové sítě zarovnán nulovými hodnotami na matici o 700 řádcích. Jediný protein, který tuto hranici přesahoval byl rozdělen na dva překrývající se úseky délky 700 reziduí, se kterými bylo dále zacházeno jako s dvěma různými proteiny.

5.1 Experimentální data

Základními datovými množinami, na kterých byly testovány všechny varianty předzpracování, architektur a filtrování, jsou veřejně dostupné množiny `CullPDB` a `CB513` (Zhou a Troyanskaya, 2014). `CullPDB` je velká množina obsahující 6133 proteinů s hranicí maximální identity proteinů 30%. Naopak `CB513` je malý dataset s 513 proteiny určený především pro testování a srovnávání modelů. Za účelem možnosti testovat na `CB513` modely naučené na `CullPDB` byly z `CullPDB` odebrány proteiny s vyšší něž 25% identitou s proteiny z `CB513` a byla tak vytvořena podmnožina `CullPDB_filt` obsahující 5534 proteinů. Oba datasety obsahují také pro každý potein PSSM naškálované sigmoidou a získané třemi iteracemi programu PSI-BLAST nad proteinovou databází UniRef90 s e-hodnotou 0.001. Datasety obsahují také skutečnou sekundární strukturu proteinů a dvě binární hodnoty reprezentující relativní a absolutní hodnotu přístupnosti rozpouštědla k jednotlivým reziduím.

Jednotlivé množiny mají velmi podobné zastoupení různě dlouhých proteinů s průměrnou délkou proteinu okolo 200 reziduí a maximální délkou 700 reziduí. Detailnější pohled na délky proteinů v datasetech nabízí histogram na obrázku 5.1. I zastoupení jednotlivých sekundárních struktur je ve všech datasetech podobné viz. obrázek 5.2. Z procentuálního zastoupení jednotlivých sekundárních struktur také můžeme odvodit úspěšnost triviálního majoritního klasifikátoru,

jehož Q_8^U by se pohybovala okolo 30% až 35% v závislosti na zvoleném datasetu.

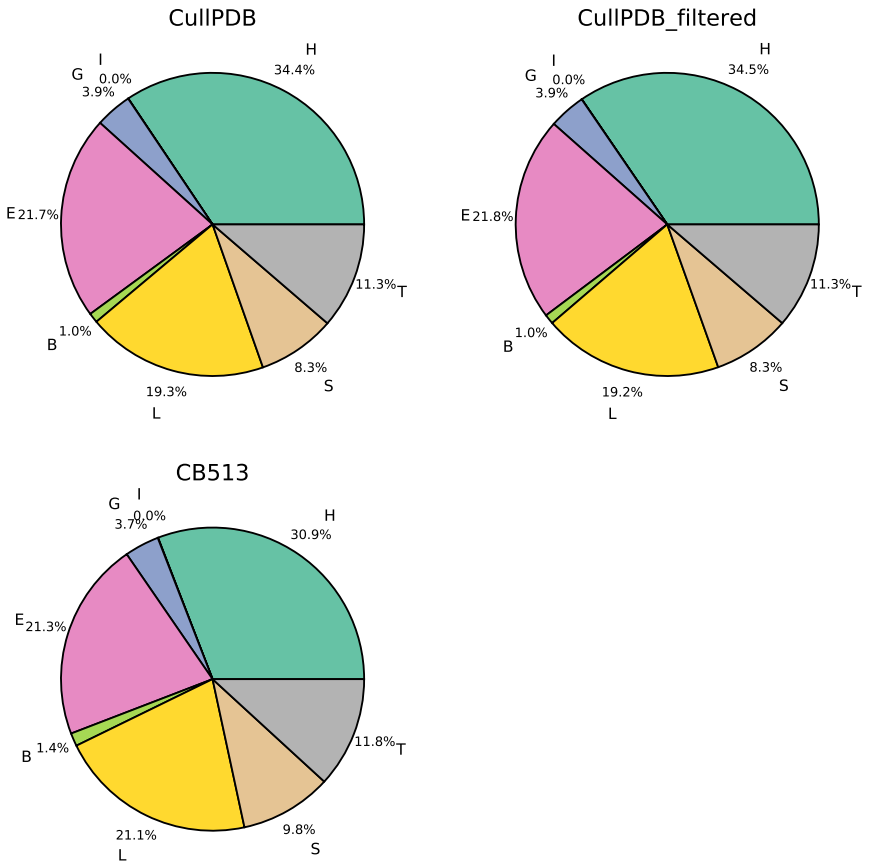


Obrázek 5.1: Normalizovaný histogram délek proteinů v jednotlivých datasetech.

5.2 Implementace výchozího modelu

V publikovaném článku popisujícím architekturu DCRNN byla vynechána důležitá informace o počtu neuronů ve dvou plně propojených vrstvách v klasifikačním bloku. Tento parametr architektury jsme experimentálně určili tak, aby model dosahoval maximálního možného Q_8^U skóre. Horní hranici počtu neuronů jsme stanovili na 392, jelikož právě takový rozměr má vstupní vektor do této vrstvy. Z důvodu velké časové náročnosti učení a testovaní každé konfigurace jsme otestovali pouze 12 konfigurací. Ty jsme stanovili tak, aby v rozumné míře pokrývaly prostor všech možných konfigurací. Počet neuronů v jednotlivých vrstvách včetně dosaženého Q_8^U skóre každé konfigurace je k nahlédnutí v tabulce A.1 v příloze této práce.

Jako výchozí architekturu pro naši práci jsme zvolili variantu s 25 neurony v obou neuronových vrstvách, jelikož tato kombinace podle našich experimentů



Obrázek 5.2: Zastoupení sekundárních struktury proteinů v jednotlivých datasetech. Znázorněné struktury jsou α-helix (H), π-helix (I), 3_{10}-helix (G), β-vlákno (E), β-most (B), neuspořádaná struktura (L), ohyb (S), β-otočka (T)

nabízí nejlepší poměr mezi Q_8^U skóre a počtem parametrů neuronové sítě, které jsou potřeba k reprezentaci této konfigurace.

5.2.1 Srovnání

Autoři původního článku o architektuře DCRNN publikovali výsledky měřené na dvou datasetech, a to `CullPDB` a `CB513`. Oba tyto datasety máme k dispozici, provedeme tedy srovnání námi a jimi naměřených výsledků na rozdělení dat, které zvolili autoři původní práce. V případě testování na `CB513` neprobíhá učení modelu na podmnožině `CB513`, ale na datasetu `CullPDB_filt`. Na této datové množině se náš výchozí model dosáhl mírně horší výsledek než autoři DCRNN, konkrétně o 1.5%. Naopak na datasetu `CullPDB` se nám ale podařilo autory překonat o 0.5%.

Mimo samotného modelu DCRNN testovali také autoři na `CB513` složený model. Jedná se o kompozici 10 modelů, které byly samostatně natrénovány na náhodné podmnožině `CullPDB_filt`. Zbytek `CullPDB_filt` (zhruba 10%) je použit k validaci modelu. Z každého běhu je vybrán jako finální model ta epocha, která dosahuje nejvyššího Q_8^U skóre na validační množině. Predikce všech 10 modelů jsou pak pomocí aritmetického průměru zkombinovány do finální predikce. Původní DCRNN kombinováním 10 samostatných modelů dosáhl pouze malého zlepšení o 0.3%, konkrétně na testovací množině `CB513` dosáhl Q_8^U skóre 69.7%. Námi vytvořená kombinace 10 modelů měla Q_8^U skóre rovno 69.3%, zlepšení tedy o celých 1.4%. Naměřené výsledky samostatných i složených modelů ukazuje tabulka č. 5.1.

Model	CB513	CullPDB
DCRNN	69.4	73.2
složený DCRNN	69.7	-
výchozí model	67.9	73.7
složený výchozí model	69.3	-

Tabulka 5.1: Srovnání Q_8^U skóre nahlášeného autory DCRNN s našimi výsledky naměřenými na stejných datasetech pomocí našeho výchozího modelu odvozeného od DCRNN. Zobrazovány jsou nejlepší dosažené hodnoty.

Autoři DCRNN tedy reportovali pro samostatný model na `CB513` lepší výsledky, než které se podařilo získat nám, v ostatních případech byly výsledky

srovnatelné. Z důvodu náhodné inicializace vah při vytvoření neuronové sítě by jistého zlepšení šlo dosáhnou větším počtem opakování pokusu. Ale tento výchozí model v rámci této práce poslouží pouze jako referenční model pro testování dalších modifikací, proto maximální dosažená hodnota úspěšnosti predikce není příliš důležitá.

5.3 Modifikace architektury sítě

Pro všechny testované modifikace výchozího modelu jsme zvolili identické rozdělení dat. Modely byly učeny a validovány na množině `CullPDB_filt` rozdělené na trénovací (proteiny 1 až 5100) a validační část (proteiny 5101 až 5534). Testování modelů následně probíhalo na datasetu `CB513`.

5.3.1 Úpravy konvolučního bloku

Na výše popsaném rozdělení dat byla otestována možnost extrakce lokálního kontextu reziduí v proteinu pomocí posloupnosti sériově zapojených konvolučních vrstev. Aktivační funkci filtrů jsme ponechali identickou jako v původní architektuře DCRNN, tedy funkci $ReLU$. Počet filtrů v každé z konvolučních vrstev byl také zafixován na 64 filtrů. Volnými parametry tedy zůstali pouze počet vrstev a zorné pole filtrů. Velikost zorných polí byla zvolena 3, 5 a 7 a počet vrstev byl shora omezen na 3 vrstvy. Kombinace těchto parametrů totiž pokrývají již dostatečné velké okolí klasifikovaného rezidua. Naměřené Q_8^U a SOV_8 skóre všech konfigurací zobrazuje tabulka v příloze č. A.2.

Překvapivě, dosažené skóre jednotlivých konfigurací bylo téměř identické nehledě na počet vrstev a velikost zorných polí. Navíc se nelišilo ani od původní architektury výchozího modelu. Na základě těchto výsledků byla otestována důležitost celého konvolučního bloku jeho odebráním z výchozího modelu o otestováním na stejném datovém rozdělení jako ostatní konfigurace. I varianta postrádající konvoluční blok dosahovala stejných výsledků, konkrétně pro 5 nezávislých modelů bylo naměřené Q_8^U resp. SOV_8 skóre 67.530% ($\pm$0.148) resp. 65.526% ($\pm$0.224). Podle těchto výsledků je tedy celý konvoluční blok ve výchozím modelu nadbytečný a stejné úspěšnosti predikce lze dosáhnout i bez s jeho vynecháním. Tím byly ukončeny další experimenty s modifikacemi tohoto bloku.

5.3.2 Přechod na LSTM paměťovou buňku

V této části byla zkoumána možnost nahrazení GRU v rekurentním bloku větší paměťovou buňkou typu LSTM. V první řadě byla provedena série pokusů za cílem nalézt nejvhodnější aktivační funkci. Na zafixované architektuře s dimenzí výstupu z každé LSTM buňky rovné 100 a o třech vrstvách obousměrných LSTM byly otestovány aktivační funkce *ReLU*, *tanh* a *sigmoid*. V případě aktivační funkce *ReLU* nastávala během výpočtu chybové funkce chyba a výsledek byl označen jako `NaN` (Not a Number). To bylo pravděpodobně způsobeno neomezeností *ReLU*, kvůli které docházelo k nárůstu hodnot při průchodu sítí nad hranici, kterou je možné reprezentovat. Dále tedy byly srovnávány pouze funkce *tanh* a *sigmoid*, jejichž dosažené skóre je zobrazeno v tabulce č. 5.2. Z tohoto srovnání vyšla lépe funkce *tanh*, kterou tedy dále využívaly jako aktivační funkci LSTM buněk všechny další varianty neuronových sítí v této sekci.

Aktivační funkce	Q_8^U		SOV_8	
	Průměr	Odchylka	Průměr	Odchylka
tanh	65.755	0.036	63.549	0.152
sigmoid	64.847	0.242	62.124	0.240

Tabulka 5.2: Dosažené Q_8^U a SOV_8 skóre výchozího modelu s LSTM paměťovou buňkou s dvěma různými aktivačními funkcemi. Zobrazené skóre je průměrné skóre dosažené 5 nezávislými běhy pro každou z aktivačních funkcí.

Následně byla laděna dimenze výstupu z LSTM buněk a počet vrstev. Pro velkou časovou náročnost učení a testování těchto modelů, která pro největší modely přesahovala 2 dny, byla dimenze výstupu shora omezena na 140 a počet vrstev na 3. Celkem bylo natrénováno a otestováno 21 různých architektur. Podrobné konfigurace a jejich dosažené skóre na testovací množině `CB513` jsou popsány v tabulkách A.3 a A.4 v příloze této práce.

Získané výsledky ukazují, že spíše než počet vrstev je klíčová dimenze výstupu paměťových buněk. Navíc se zdá, že LSTM buňky vyžadují k dosažení svého maximálního predikčního skóre pouze relativně málo vnitřních neuronů, jelikož všechny varianty s dimenzí výstupu alespoň 40 dosahovaly srovnatelného Q_8^U nehledě na počet vrstev. Oproti tomu výchozí model s GRU rekurentním blokem vícevrstevnatost vyžaduje. Varianta výchozí architektury s jedinou vrstvou GRU jednotek a původním počtem neuronů v každé bráně paměťové buňky dosahoval

signifikantně horších výsledků (t-test při hladině významnosti 0.01).

Všechny varianty využívající LSTM ale dosahovaly znatelně nižšího Q_8^U a SOV_8 skóre než výchozí model s GRU. Na modelech, které dosáhly nejvyššího Q_8^U skóre, můžeme srovnáním s výchozími modely pozorovat původ tohoto zhoršení. To je celkem rovnoměrně rozprostřeno mezi všechny sekundární struktury s výjimkou β-skládaného listu, kde modely založené na LSTM dosahovaly stejné nebo dokonce lepší úspěšnosti. Podrobnější přehled úspěšnosti predikce nabízejí matice záměn nejlepších modelů výchozí architektury a architektur využívající LSTM zobrazené v přílohách B.1 a B.2.

Pro srovnání byla také implementována varianta s jednodušším rekurentním blokem. V tomto případě byla také každá rekurentní vrstva tvořena dvěma jednotkami čtoucími vstupní sekvenci z opačných stran. Tyto jednotky ale byly tvořeny jedinou vrstvou neuronů plně propojenou se vstupy, která také obsahovala rekurentní hrany mezi všemi neurony v dané vrstvě. Aktivační funkcí neuronů zůstala jako v případě GRU a LSTM funkce *tanh*. Nižší výpočetní náročnost učení a testování těchto variant nám umožnila zkoušet i hlubší architekturu. Nicméně i nejsložitější testovaná varianta obsahovala v rekurentním bloku přibližně polovinu parametrů (vstupních vah neuronů a prahů) oproti variantě LSTM se třemi vrstvami a dimenzí výstupu 140. Všechny testované varianty modelu využívající tyto jednodušší rekurentní jednotky dosahovaly srovnatelných výsledků s modely využívajícími LSTM buňky. Všechny testované architektury s klasickou rekurentní vrstvou a jejich dosažené Q_8^U a SOV_8 skóre znázorňuje tabulka A.5 v příloze této práce.

V této sekci se nám tedy nepovedlo laděním rekurentního bloku využívajícího LSTM paměťové buňky dosáhnout zlepšení oproti výchozímu modelu založeného na GRU. To tedy naznačuje, že GRU v kontextu predikce sekundární struktury je vhodnější paměťovou buňkou než LSTM, jejíž výsledky se nám podařilo dorovnat také jednoduššími rekurentními neuronovými vrstvami. Využití LSTM přineslo podobné zhoršení v predikci všech sekundárních struktur pouze s výjimkou β-skládaného listu.

5.4 Zrychlení výpočtu PSSM

Pomocí námi implementovaného programu FastProteinPSSM[1] pro rychlý výpočet PSSM jsme získali alternativní PSSM pro všechny proteiny z datasetů

[1]Program je dostupný na CD přiloženém k práci nebo na adrese `https://github.com/michalfilippi/FastProteinPSSM`

CullPDB a CB513. Zvolenými databázemi byli SwissProt obsahující přibližně 500 000 proteinů a UniProt90 s téměř 60 miliony proteiny. Tato metoda předzpracování umožnila výrazně zrychlit výpočet z řádů hodin do řádů minut pro každou PSSM. Navíc, jak jsme experimentálně ověřili, výchozí model pracující s novou PSSM maticí dosahoval srovnatelných výsledků. Konkrétně 5 běhů učení na množině CullPDB_filt a testování na množině CB513 dosahovalo průměrně 67.423% ($\pm$0.271) Q_8^U skóre a 65.156% ($\pm$0.371) SOV_8 skóre. Tedy oproti modelům pracujícími s původní metodou předzpracování proteinů se jedná o zhoršení průměrného skóre o 0.144% resp. o 0.108%. Podle statistického t-testu při hladině významnosti 0.05 se ale nejedná o signifikantní rozdíl.

Otestovali jsme také záměnnost obou metod předzpracování, a to v obou možných směrech. Pět predikčních modelů získaných pěti běhy učení na datasetu CullPDB_filt a testování na datasetu CB513 s tradičně získanými PSSM jsme otestovali znovu na datasetu CB513, ale s PSSM vypočítanými naší metodou. V tomto případě došlo oproti původnímu výsledku k průměrnému poklesu Q_8^U i SOV_8 skóre o 0.923% ($\pm$0.075) resp. o 0.905% ($\pm$0.155). To ale ukazuje, že za cenu malého poklesu přesnosti predikce je možné získat predikci sekundární struktury řádově rychleji, a to pomocí stávajících modelů učených na tradičních datech.

Otestován byl i opačný směr záměny. Tedy predikční modely získané učením resp. otestováním výchozího modelu na datasetu CullPDB_filt resp. CB513 s naší rychlejší metodou získávání PSSM byly otestovány na CB513 s původními PSSM. Při tomto nastavení došlo k výraznému vylepšení Q_8^U i SOV_8 skóre, a to průměrně o 0.377% ($\pm$0.101) resp. o 0.346% ($\pm$0.242). Toto zlepšení bylo dokonce tak výrazné, že modely naučené na datech s PSSM získanou novým způsobem dosahovaly signifikantně (t-test při hladině významnosti 0.05) lepšího Q_8^U skóre na CB513 s původními PSSM než modely, které k učení využívaly také původní PSSM.

5.5 HP mřížková struktura

Pomocí programu HPstruct jsme získali předpovězenou FCC mřížkovou strukturu 1727 proteinů z datasetu CullPDB. Pro ostatní proteiny program nebyl schopen strukturu předpovědět. Převod mezi absolutní a relativním popisem struktury proteinu popsaný v části 4.3.2 byl implementován do převodního programu PAS2RS[2] a všechny mřížkové struktury byly pomocí toho programu předzpraco-

[2]Program je dostupný na CD přiloženém k práci nebo na adrese https://github.com/michalfilippi/PAS2RS

vány. Z množiny proteinů s předzpracovanou mřížkovou strukturou byl vytvořen nový dataset `CullPDB_FCC`, který byl rozdělen na trénovací data (proteiny 1 až 1500) a testovací data (proteiny 1501 až 1727). Se vstupem v podobě předzpracované mřížkové struktury neuronová síť pracuje stejně jako s PSSM, tedy tato část vstupu neprocházela blokem pro kódování vstupu. Na zmíněném rozdělení dat jsme na výchozím modelu otestovali 3 různé kombinace vstupů do neuronové sítě.

1. Matice reprezentující jednotlivé aminokyseliny, PSSM, matice s předzpracovanou mřížkovou strukturou.

2. Matice reprezentující jednotlivé aminokyseliny, matice s předzpracovanou mřížkovou strukturou.

3. Matice s předzpracovanou mřížkovou strukturou. V tomto případě byl z výchozího modelu odstraněn nadbytečný blok zakódování vstupu.

Výsledky jednotlivých variant byly srovnávány s alternativními modely, kterým byl ale ze vstupu navíc odebrána mřížková struktura. V případě varianty 3, která na vstupu obsahuje pouze mřížkovou strukturu, jsme srovnání prováděli oproti majoritnímu klasifikátoru. Pro každou z těchto variant bylo natrénováno a otestováno 5 modelů. První dvě varianty vstupů dosahovaly téměř identické predikční úspěšnosti jako jejich alternativy bez mřížkové struktury. V případě třetí varianty již ke zlepšení Q_8^U skóre oproti majoritnímu klasifikátoru došlo. Toto zlepšení ale není příliš výrazné. Využití pouze mřížkové struktury na vstupu neuronové sítě nedosahovalo ani srovnatelného Q_8^U skóre jako model pracující na vstupu pouze s maticí reprezentující aminokyseliny v daném proteinu. FCC mřížková struktura se tedy podle našich experimentů nedá považovat za užitečnou pro predikci sekundární struktury, respektive její přínos je snadno překonán libovolným z ostatních běžných vstupů. Výsledky v podobě Q_8^U skóre jednotlivých variant a jejich alternativ s odebranou mřížkovou strukturou na vstupu jsou vypsány v tabulce č. 5.3.

Abychom ověřili, že chyba nespočívá ve způsobu předzpracování mřížkové struktury nebo případně v nedostatečné síle architektury výchozího modelu, postup jsme otestovali na skutečných terciárních strukturách proteinů. Z proteinové databáze PDB bylo získáno 1292 záznamů proteinů z datasetu `CullPDB`. Z těchto záznamů pak byla pro každý z proteinů extrahována posloupnost souřadnic centrálních uhlíků získaná rentgenovou krystalografií. Posloupnost centrálních uhlíků byla dále zpracována stejným způsobem jako FCC mřížková struktura. Vytvořený

Varianta	Výsledky		Výsledky bez FCC MS	
	Průměr	Odchylka	Průměr	Odchylka
Varianta 1	72.938	0.433	72.833	0.179
Varianta 2	62.902	0.486	63.337	0.424
Varianta 3	44.309	0.126	39.4	0.0

Tabulka 5.3: Q_8^U skóre tří variant modelů s odlišnými vstupy na datasetu `CullPDB_FCC`. Sloupec výsledků bez FCC mřížkové struktury odpovídá výsledkům na modelech s odebranou mřížkovou strukturou na vstupu. V případě varianty 3 se pak jedná o úspěšnost majoritního klasifikátoru (tedy pouze predikce každé struktury na α-helix). Pro každou variantu bylo spuštěno 5 nezávislých běhů pro model s FCC MS i be ní.

dataset `CullPDB_STR` o 1292 proteinech byl dále rozdělen na trénovací množinu (proteiny 1 až 1100) a testovací množinu (proteiny 1101 až 1292).

No tomto rozdělení dat jsme naučili a otestovali model pracující na vstupu pouze s předzpracovanou terciární strukturou. Takový model dosahoval průměrně 87.375% ($\pm$0.191) Q_8^U skóre a o 86.963% ($\pm$0.300) SOV_8. Modely pracující s rozšířeným vstupem o matice aminokyselin, případně i o PSSM, dosahovaly srovnatelného Q_8^U i SOV_8 skóre a nedocházelo k dalšímu zlepšení. Tento výsledek je velkým skokem oproti výchozímu modelu s klasickými vstupy, který na stejném datasetu dosahuje 68.269% ($\pm$0.283) Q_8^U skóre a o 66.404% ($\pm$0.446) SOV_8 skóre. Především se ale s využitím terciární struktury výrazně lépe dařilo predikovat i méně časté struktury, především pak 3_{10}-helix, ohyb a β-otočku, jejichž pravděpodobnost úspěšné predikce stoupla o 58%, 63% a 35%. Tyto a další rozdíly v úspěšnosti predikce je možné vidět v maticích záměn pro obě varianty modelů v přílohách č. B.3 a B.4.

Úspěšnost modelů využívající terciární strukturu proteinů ukazuje, že predikci terciární struktury lze využít pro zlepšení predikce sekundární struktury. Naše experimenty ale ukázaly, že zjednodušený strukturní model postavený na FCC mřížce není dostačující. Bylo by tedy vhodné otestovat tento přístup pomocí přesnějších predikcí. Ty je ale extrémně výpočetně náročné získat, a proto nebylo možné provést tyto experimenty v rámci této práce.

5.6 Vyhlazování

Pro zajištění korektnosti evaluace vyhlazování výstupu pomocí metod strojového učení je ale potřeba učit a testovat tyto metody na datech nezávislých na datech určených pro trénování a testování predikčního modelu. Pro tuto sérii experimentů jsme zvolili následující rozvržení dat `CullPDB`. Výchozí model jsme natrénovali na nové trénovací části dat (proteiny 1 až 4500) a pomocí validační množiny (proteiny 5801 až 6000) jsme vybrali epochu dosahující nejvyššího Q_8^U skóre. Pomocí získaného predikčního modelu jsme pro všechny ostatní proteiny z `CullPDB` získali predikci sekundární struktury ve formě osmiprvkového vektoru pro každé reziduum reprezentující předpovězené pravděpodobnosti pro jednotlivé sekundární struktury. Ze získaných predikcí proteinů jsme vytvořili tři nové množiny, a to trénovací (proteiny 4201 až 5700), validační (proteiny 5701 až 5800) a testovací (proteiny 6001 až 6133). Ty byly využity na otestování vyhlazování pomocí SVM, logistické regrese a neuronové sítě. K implementaci těchto metod byla použita knihovna scikit-learn (Pedregosa a kol., 2011), parametry jednotlivých metod, které nejsou dále specifikovány byly nastaveny na výchozí hodnoty určené knihovnou scikit-learn. Každá z těchto metod při vyhlazování zohledňuje pouze jisté okolí rezidua, které vyhlazuje. Velikosti těchto plovoucích oken byly nastaveny podle studie vyhlazovacích metod pro predikci sekundárních struktur z roku 2012 (Kountouris a kol., 2012), tedy 5 pro neuronové sítě a 19 pro logistickou regresi a SVM. Predikce krajních 2 resp. 9 reziduí tedy není vyhlazováním ovlivněna.

Zmíněná studie pracovala pouze se základními konfiguracemi jednotlivých metod. My jsme v základním nastavení ponechali pouze logistickou regresi. Pro SVM a neuronové sítě, které nabízejí širší možnosti nastavení, jsme využili validační množiny a otestovali více možných konfigurací. Konkrétně jsme nejdříve učili nejvhodnější kernelovou funkci pro SVM. Výběr jsme omezili na lineární funkci, RBF, sigmoidu a polynomiální funkce stupně 2, 3 a 4. Výsledné Q_8^U a SOV_8 skóre všech variant na validační množině zobrazuje tabulka č. 5.4. Varianta s lineární kernelovou funkcí, která na validační množině dosáhla nejvyššího SEL skóre, tedy součtu Q_8^U a SOV_8 skóre, byla zvolena jako nejvhodnější.

V případě vyhlazování pomocí neuronové sítě bylo potřeba zvolit počet skrytých vrstev, počet neuronů v těchto vrstvách a aktivační funkce neuronů. Počet skrytých vrstev jsme shora omezili dvěma vrstvami. Aktivační funkci neuronů jsme volili mezi funkcemi sigmoida a $ReLU$. Počet neuronů v každé skryté vrstvě jsme zvolili 10, 20, 40, 80 a 160. Pro každou možnou varianty neuronové sítě s těmito parametry jsme spustili 40 běhů učení na trénovacích datech a každý běh

Kernelová funkce	Q_8^U	SOV_8
RBF	72.201	69.784
Sigmoid	71.948	70.167
Lineární	72.074	70.161
Polynomiální st. 2	70.827	67.207
Polynomiální st. 3	67.090	59.286
Polynomiální st. 4	58.630	30.361

Tabulka 5.4: Q_8^U a SOV_8 skóre SVM vyhlazovacích modelů s plovoucím oknem velikosti 19 s různými kernelovými funkcemi na validační množině. Hodnoty jsou udávány v procentech.

otestovali na validační množině. Dosažené Q_8^U a SOV_8 skóre všech možných konfigurací jsou zobrazeny v tabulce č. A.6 v příloze práce. Stejně jako v případě SVM jsme jako finální variantu vybraly tu, která dosahovala na validační množině nejvyššího průměrného SEL skóre. Tedy variantu s 1 skrytou vrstvou a 80 neurony s aktivační funkcí *sigmoid*. Z této varianty jsme pro měření na testovací sadě dat vybrali běh, který dosahoval nejvyššího SEL skóre ze všech běhů.

Z vybraných tří vyhlazovacích metod dosáhla nejlepšího Q_8^U skóre logistická regrese, nejvyššího SOV i SEL skóre pak neuronová síť. Všechny tři metody vyhlazování ale na testovacích datech zhoršily úspěšnost predikce Q_8^U oproti nevyhlazeným predikcím. V případě neuronových sítí to pak bylo vyváženo nárůstem SOV_8 skóre, logistická regrese i SVM výsledné SOV_8 mírně snížili. Přesné naměřené hodnoty všech vyhlazovacích metod na testovací sadě dat jsou zobrazeny v tabulce č. 5.5.

Identickým postupem jsme také otestovali vyhlazování předpovědi pro sekundární struktury S_3. Původní predikce struktur S_8 byly přepočítány na predikce struktury S_3 pro každé reziduum následovně.

$$p_3[H] = p_8[H] + p_8[I] + p_8[G]$$
$$p_3[C] = p_8[S] + p_8[T] + p_8[L]$$
$$p_3[E] = p_8[E] + p_8[B]$$

kde $p_8[m]$ je původní predikovaná pravděpodobnost struktury m typu S_8 na dané pozici v proteinu a $p_3[n]$ je výsledná pravděpodobnost struktury n typu S_3

Metoda	Q_8^U	SOV_8
Bez vyhlazování	73.214	71.237
SVM	72.993	71.076
Logistická regrese	73.189	70.995
Neuronová síť	73.144	71.385

Tabulka 5.5: Q_8^U a SOV_8 skóre vybraných vyhlazování modelů na testovací množině. Hodnoty jsou zobrazeny v procentech.

na dané pozici. Na takto upravených datech byl odladěn SVM model i neuronové sítě. V tomto případě na validačních datech dosáhly nejvyššího SEL skóre SVM s polynomiální funkcí stupně 3 a neuronová síť bez skrytých vrstev a výstupní funkcí *sigmoid*. Vyhlazování pro sekundární struktury typu S_3 na testovacích datech v případě všech tří metod způsobilo zanedbatelný pokles Q_3^U skóre, ale ve všech případech přineslo nárůst SOV skóre. Tento nárůst byl pak nejvýraznější u neuronových sítí, kde SOV skóre narostlo o 0.712% vzhledem k variantě bez vyhlazování. Výsledky na validační množině při ladění SVM a neuronových sítí jsou k nalezení v příloze v tabulkách A.7 a A.8. Celkové výsledky vyhlazování na strukturách S_3 na testovací množině jsou pak zobrazeny v tabulce č. 5.6.

Metoda	Q_8^U	SOV_8
Bez vyhlazování	84.102	81.078
SVM	84.029	81.311
Logistická regrese	84.059	81.532
Neuronová síť	84.076	81.790

Tabulka 5.6: Q_3^U a SOV_3 skóre vybraných vyhlazování modelů na testovací množině. Hodnoty jsou zobrazeny v procentech.

Výsledky získané v této části při vyhlazování struktur typu S_3 lze považovat za konzistentní s výsledky publikovanými ve zmiňované studii vyhlazovacích metod především z pohledu SOV_3 skóre. Nižší nárůst SOV_3 skóre a stagnaci Q_3^U skóre připisujeme rozdílnosti použitých modelů, jejichž predikce byly vyhlazovány. Náš model je výrazně pokročilejší a jeho predikce přesnější, což dělá následné vyhla-

zování komplikovanější. Především ale bylo ukázáno, že vyhlazování výstupu není vhodný nástroj pro dodatečné zvyšování Q_8^U a SOV^U skóre při predikci náročnějšího rozdělení struktur S_8.

Závěr

V rámci této práce byla provedena rešerše problému predikce sekundární struktury proteinů, používaných metod a metrik měřících kvalitu predikcí. Na tuto část bylo navázáno experimentální částí, ve které byl implementován predikční model založený na architektuře DCRNN, která v době před začátkem této práce dosahovala nejvyšších Q_8^U a SOV_8 skóre na veřejně dostupných datasetech.

Implementovaná architektura byla dále modifikována na úrovni dvou komponent, a to konvolučního a rekurentního bloku. Tyto úpravy výchozího modelu ale žádné zlepšení predikce nepřinesly. Naopak bylo ukázáno, že konvoluční blok je ve výchozím modelu nadbytečná komponenta a po jeho odstranění je možné dosáhnout srovnatelných výsledků. Navíc po odstranění se model mírně zjednodušuje, což má pozitivní vliv na dobu učení. V rámci modifikací rekurentního bloku bylo hlavní snahou přejít od GRU k LSTM paměťovým buňkám, které v rámci některých prací dosahovaly lepších výsledků oproti GRU. Naše experimenty ale ukázaly, že LSTM dosahuje mírně horších výsledků při predikci všech sekundárních struktur s výjimkou β-skládaného listu, kde LSTM dosahovala srovnatelných nebo dokonce lepších výsledků. Podle našich experimentů je tedy GRU v kontextu predikce sekundární struktury S_8 vhodnější paměťovou buňkou.

Dále byla navržena úprava postupu klasického předzpracování proteinů, které je využíváno téměř všemi publikovanými modely. Konkrétně námi navržená metoda výpočtu PSSM řádově urychluje předzpracování proteinů z řádů hodin do řádů minut. Navíc modely učené na těchto nových PSSM dosahují srovnatelného výkonu jako modely učené na standardních datech. Jak bylo experimentálně ověřeno, stávající modely, které při učení využívaly PSSM získané tradičním způsobem, lze využít i k predikci struktury proteinů s PSSM vypočítanou naším způsobem. To výrazně urychluje čas predikce, jelikož právě výpočet PSSM je zdaleka nejnáročnějším výpočtem predikce. Cenou za tuto časovou úsporu je mírný pokles Q_8^U a SOV_8 skóre predikce přibližně o 1%. Zajímavého výsledku bylo především dosaženo učením modelů na PSSM získaných naší metodou a následné predikce proteinů s klasicky získanou PSSM. Takto získané predikce dosahovaly signifikantně vyšších hodnot Q_8^U skóre než modely, které využívaly tradiční PSSM i v rámci procesu učení. Domníváme se, že toto zlepšení je dáno nižší přesností PSSM získaných naší metodou, což paradoxně vytváří tlak na generalizaci neuronové sítě, která následně s přesnějšími PSSM dosahuje vyšší přesnosti. Tato hypotéza ale nebyla ověřena.

Mimo samotného zrychlování předzpracování byla zkoumána také možnost

rozšířit sadu vstupů o predikci terciární struktury. Z důvodu zachování korektnosti výsledků nebylo možné využít žádný z běžně používaných prediktorů, jelikož ty při svém výpočtu využívají jiné prediktory sekundárních struktur. Využita tedy byla aproximace struktury získaná pomocí predikce HP mřížkové struktury proteinů. Pomocí nového datasetu `CullPDB_FCC` tvořeného proteiny, pro které se HP mřížkovou strukturu podařilo získat, bylo ukázáno, že mřížková struktura nepředstavuje dostatečně přesnou aproximaci skutečné terciární struktury. Bylo ukázáno, že využití mřížkové predikce struktury je poskytuje méně informací, než kolik je možné získat ze samotné primární struktury proteinu.

Ověření, že chyba skutečně spočívala v nepřesnosti aproximace skutečné struktury mřížkovou strukturou, byl proveden podobný experiment využívající skutečné terciární struktury proteinů získané z PDB. Byl vytvořen nový dataset `CullPDB_FCC` obsahující 1292 proteinů s jejich terciární strukturou předzpracovanou stejným způsobem jako v případě `CullPDB_FCC`. Na tomto datasetu bylo ukázáno, že architektura neuronové sítě byla dostatečná a způsob předzpracování relevantní. Bylo ale tedy také ukázáno, že přesnější predikce terciární struktury lze využít i pro zpřesnění predikce sekundárních struktur. Největší přínos lze pak očekávat pro predikce málo se vyskytujících struktur, především tedy 3_{10}-helix, ohyb a β-otočku.

V rámci zkoumání předzpracování proteinů byly vyvinuty dva programy. Ty slouží k rychlému výpočtu PSSM proteinů a k transformaci sekvence absolutních pozic centrálních uhlíků do sekvence několika úhlů popisující relativní strukturu. Oba tyto programy jsou přiloženy k této práci na CD, případně jsou veřejně dostupné na serveru GitHub[3][4].

Také jsme v rámci této práce navázali na studii vyhlazovacích metod pro predikční modely. Publikované výsledky na téma vyhlazování sekundární struktury S_3 jsme se pokusili aplikovat na komplikovanější problém predikce sekundární struktury proteinů S_8. Otestovali jsme tři metody dosahující nejlepších výsledků na S_3, a to logistickou regresi, SVM a neuronové sítě. V kontextu predikce S_3 struktur se nám podařilo, stejně jako autorům publikace, dosáhnou zlepšení SOV_3 skóre pomocí všech tří zmíněných vyhlazovacích metod. Naše experimenty ale ukázaly, že vyhlazování v souvislosti s naším výchozím modelem a predikcí sekundární struktury S_8 již není vhodný nástroj ke zvýšení Q_8^U skóre. Vyhlazování pomocí neuronových sítí přineslo mírné navýšení SOV_8 skóre, zbylé dvě zmíněné metody výslednou SOV_8 skóre snižovaly.

[3]FastProteinPSSM na adrese `https://github.com/michalfilippi/FastProteinPSSM`

[4]PAS2RS na adrese `https://github.com/michalfilippi/PAS2RS`

Navazující práce

Na základě výsledků této práce je možné navrhnout několik možných vylepšení. GRU je zdá být vhodnější paměťovou buňkou pro predikci sekundární struktury, bylo by tedy zajímavé pokusit se uplatnit poznatky publikované v souvislosti s LSTM a sekundární strukturou na nové modely postavené na GRU. Další možností by bylo aplikovat zatím nevyzkoušené rekurentní modely. Takovým příkladem by mohly být například neuronové turingovy stroje (Graves a kol., 2014).

Druhým možným vylepšením by mohlo být využití postupné učení bez učitele pro neuronové sítě (Bengio a kol., 2006). To snižuje náročnost učení sítě a umožňuje efektivněji učit i hluboké neuronové sítě. Podle všech námi dostupných informací nebyla tato možnost v souvislosti s predikcí sekundární struktury vyzkoušena.

Velmi zajímavé by pak bylo pokračovat ve zkoumání možností zahrnutí terciární predikce mezi vstupy modelů při predikci sekundární struktury. Například by bylo možné zahrnout predikce, při jejichž výpočtu nebyl použit žádný prediktor sekundární struktury. Tyto predikce jsou ale nejčastěji založené na minimalizaci potenciální energie proteinu a jejich výpočet je velmi výpočetně náročný. Právě proto tato možnost nebyla součástí této práce.

Seznam použité literatury

Altman, R. B. a Dugan, J. M. (2003). Defining bioinformatics and structural bioinformatics. *Methods Biochem Anal*, **44**, 3–14.

Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science*, **181**(4096), 223–230. ISSN 00368075.

Arnold, G. E., Dunker, A. K., Johns, S. J. a Douthart, R. J. (1992). Use of conditional probabilities for determining relationships between amino acid sequence and protein secondary structure. *Proteins: Structure, Function, and Bioinformatics*, **12**(4), 382–399. ISSN 1097-0134. doi: 10.1002/prot.340120410. URL `http://dx.doi.org/10.1002/prot.340120410`.

Aydin, Z., Altunbasak, Y. a Borodovsky, M. (2006). Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. *BMC Bioinformatics*, **7**, 178.

Baldi, P., Brunak, S., Frasconi, P., Soda, G. a Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, **15**(11), 937. doi: 10.1093/bioinformatics/15.11.937. URL `+http://dx.doi.org/10.1093/bioinformatics/15.11.937`.

Bengio, Y., Simard, P. a Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions On Neural Networks*, **5**(2), 157–66. ISSN 10459227.

Bengio, Y., Lamblin, P., Popovici, D. a Larochelle, H. (2006). Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 153–160, Cambridge, MA, USA, 2006. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2976456.2976476`.

Biasini, M., Bienert, S., Waterhouse, A., Arnold, K., Studer, G., Schmidt, T., Kiefer, F., Gallo, C. T., Bertoni, M., Bordoli, L. a Schwede, T. (2014). SWISS-model: modelling protein tertiary and quaternary structure using evolutionary information. *Nucleic Acids Research*, **42** (Web Server issue), W252–8. ISSN 13624962. doi: 10.1093/nar/gku340.

Boyle, J. (2005). Lehninger principles of biochemistry (4th ed.): Nelson, d., and cox, m. *Biochemistry and Molecular Biology Education*, **33**(1), 74–75. ISSN

1539-3429. doi: 10.1002/bmb.2005.494033010419. URL http://dx.doi.org/10.1002/bmb.2005.494033010419.

Busia, A., Collins, J. a Jaitly, N. (2016). Protein secondary structure prediction using deep multi-scale convolutional neural networks and next-step conditioning.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. a Madden, T. L. (2009). BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Capra, J. A. a Singh, M. (2007). Predicting functionally important residues from sequence conservation. *Bioinformatics*, **23**(15), 1875–1875. ISSN 13674803.

Chen, J. a Chaudhari, N. (2007). Cascaded bidirectional recurrent neural networks for protein secondary structure prediction. *IEEE/ACM Transactions On Computational Biology And Bioinformatics*, **4**(4), 572–82. ISSN 15455963.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H. a Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, **abs/1406.1078**. URL http://arxiv.org/abs/1406.1078.

Chollet, F. a kol. (2015). Keras. https://github.com/fchollet/keras.

Chou, P. Y. a Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry*, **13**(2), 222–245.

Chung, J., Gülçehre, Ç., Cho, K. a Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, **abs/1412.3555**. URL http://arxiv.org/abs/1412.3555.

Ciresan, D. C., Meier, U. a Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *CoRR*, **abs/1202.2745**. URL http://arxiv.org/abs/1202.2745.

Collins, J., Sohl-Dickstein, J. a Sussillo, D. (2016). Capacity and Trainability in Recurrent Neural Networks. *ArXiv e-prints*.

Cravens, A. a Probert, C. (2016). Annotating protein secondary structure from sequence.

Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A. a Yannakakis, M. (1998). On the complexity of protein folding. *J. Comput. Biol.*, **5**(3), 423–465.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals*, **2**(4), 303–314. ISSN 09324194. doi: 10.1007/BF02551274.

Deng, X. a Cheng, J. (2011). MSACompro: protein multiple sequence alignment using predicted secondary structure, solvent accessibility, and residue-residue contacts. *BMC Bioinformatics*, **12**, 472.

Dorn, M., E Silva, M. B., Buriol, L. S. a Lamb, L. C. (2014a). Three-dimensional protein structure prediction: Methods and computational strategies. *Comput Biol Chem*, **53PB**, 251–276.

Dorn, M., e Silva, M. B., Buriol, L. S. a Lamb, L. C. (2014b). Review article: Three-dimensional protein structure prediction. *Computational Biology and Chemistry*, **53**(Part B), 251–276. ISSN 14769271. doi: 10.1016/j.compbiolchem.2014.10.001.

Feng, Y., Lin, H. a Luo, L. (2014). Prediction of protein secondary structure using feature selection and analysis approach. *Acta Biotheoretica*, **62**(1), 1–14. ISSN 15728358. doi: 10.1007/s10441-013-9203-7.

Finkelstein, A. V. a Ptitsyn, O. B. (1971). Statistical analysis of the correlation among amino acid residues in helical, beta-structural and non-regular regions of globular proteins. *J. Mol. Biol.*, **62**(3), 613–624.

Fu, L., Niu, B., Zhu, Z., Wu, S. a Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, **36**(4), 193–202. ISSN 1432-0770. doi: 10.1007/BF00344251. URL `http://dx.doi.org/10.1007/BF00344251`.

Garnier, J., Gibrat, J. F. a Robson, B. (1996). GOR method for predicting protein secondary structure from amino acid sequence. *Meth. Enzymol.*, **266**, 540–553.

Gers, F. A., Schmidhuber, J. a Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, **12**(10), 2451–2471. ISSN 08997667. doi: 10.1162/089976600300015015.

Goodfellow, I., Bengio, Y. a Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Graves, A., Beringer, N. a Schmidhuber, J. (2005a). Rapid retraining on speech data with lstm recurrent networks.

Graves, A., Fernández, S. a Schmidhuber, J. (2005b). *Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition*, pages 799–804. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-28756-8. doi: 10.1007/11550907˙126. URL http://dx.doi.org/10.1007/11550907_126.

Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J. a Fernández, S. (2008). Unconstrained on-line handwriting recognition with recurrent neural networks. In Platt, J. C., Koller, D., Singer, Y. a Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 577–584. Curran Associates, Inc.

Graves, A., Wayne, G. a Danihelka, I. (2014). Neural turing machines. *CoRR*, **abs/1410.5401**. URL http://arxiv.org/abs/1410.5401.

Guo, J., Chen, H., Sun, Z. a Lin, Y. (2004). A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *Proteins*, **54** (4), 738–743.

Hart, W. E. a Istrail, S. (1997). Robust proofs of NP-hardness for protein folding: general lattices and energy potentials. *J. Comput. Biol.*, **4**(1), 1–22.

Heffernan, R., Yang, Y., Paliwal, K. a Zhou, Y. (2017). Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. *Bioinformatics (Oxford, England)*. ISSN 13674811. doi: 10.1093/bioinformatics/btx218.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München.

Hochreiter, S. a Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*, **9**(8), 1735–1780.

Holley, L. H. a Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences of the United States of America*, **86**(1), 152–156. ISSN 00278424.

Hua, S. a Sun, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.*, **308**(2), 397–407.

Jacob, E., Horovitz, A. a Unger, R. (2007). Different mechanistic requirements for prokaryotic and eukaryotic chaperonins: a lattice study. *Bioinformatics*, **23**(13), i240–248.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**(2), 195–202.

Judd, J. S. (1990). *Neural Network Design and the Complexity of Learning*. MIT Press, Cambridge, MA, USA. ISBN 0-262-10045-2.

Kabat, E. A. a Wu, T. T. (1973). The influence of nearest-neighbor amino acids on the conformation of the middle amino acid in proteins: comparison of predicted and experimental determination of -sheets in concanavalin A. *Proc. Natl. Acad. Sci. U.S.A.*, **70**(5), 1473–1477.

Kabsch, W. a Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **vol. 22**(issue 12), 2577–2637. ISSN 00063525. doi: 10.1002/bip.360221211. URL `http://doi.wiley.com/10.1002/bip.360221211`.

Kim, H. a Park, H. (2003). Protein secondary structure prediction based on an improved support vector machines approach. *Protein Eng.*, **16**(8), 553–560.

Kingma, D. P. a Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, **abs/1412.6980**. URL `http://arxiv.org/abs/1412.6980`.

Kononenko, I. a Kukar, M. (2007). *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited. ISBN 1904275214, 9781904275213.

Kountouris, P., Agathocleous, M., Promponas, V., Christodoulou, G., Hadjicostas, S., Vassiliades, V. a C, C. (2012). A comparative study

on filtering protein secondary structure prediction. *IEEE/ACM Transactions On Computational Biology And Bioinformatics*, **9**(3), 731–9. ISSN 15579964. doi: 10.1109/TCBB.2012.22.

Kozlowski, L. P. (2017). Proteome-pI: proteome isoelectric point database. *Nucleic Acids Research*, **45**(D1), D1112. doi: 10.1093/nar/gkw978. URL +http://dx.doi.org/10.1093/nar/gkw978.

Krizhevsky, A., Sutskever, I. a Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, **60**(6), 84–90. ISSN 00010782. doi: 10.1145/3065386.

Källberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H. a Xu, J. (2012). Template-based protein structure modeling using the raptorx web server. *Nature Protocols*, **7**(8), 1511–22. ISSN 17502799. doi: 10.1038/nprot.2012.085.

Li, Z. a Yu, Y. (2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks.

lin, D., Vasilakos, A. V., Tang, Y. a Yao, Y. (2016). Neural networks for computer-aided diagnosis in medicine. *Neurocomputing*, **216**, 700–708. ISSN 09252312. doi: 10.1016/j.neucom.2016.08.039.

Lin, H. N., Chang, J. M., Wu, K. P., Sung, T. Y. a Hsu, W. L. (2005). HYPROSP II–a knowledge-based hybrid method for protein secondary structure prediction based on local prediction confidence. *Bioinformatics*, **21**(15), 3227–3233.

Linderstrøm-Lang, K. U. (1952). Proteins and enzymes. *Lane Medical Lectures*, **6**.

Liwicki, M., Graves, A., Bunke, H. a Schmidhuber, J. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proc. 9th Int. Conf. on Document Analysis and Recognition*, pages 367–371.

Malekpour, S. A., Naghizadeh, S., Pezeshk, H., Sadeghi, M. a Eslahchi, C. (2009). A segmental semi markov model for protein secondary structure prediction. *Mathematical Biosciences*, **221**(2), 130–5. ISSN 18793134. doi: 10.1016/j.mbs.2009.07.004.

Mann, M., Will, S. a Backofen, R. (2008). CPSP-tools–exact and complete algorithms for high-throughput 3D lattice protein studies. *BMC Bioinformatics*, **9**, 230.

Martin, J., Gibrat, J. F. a Rodolphe, F. (2006). Analysis of an optimal hidden markov model for secondary structure prediction. *BMC Structural Biology*, **6**, 25. ISSN 14726807.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D. a Zweig, G. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE-ACM Transactions on Audio, Speech, and Language Processing*, **23**(3), 530–539. ISSN 23299290.

Milo, R. (2013). What is the total number of protein molecules per cell volume? a call to rethink some published values. *BioEssays*, **35**(12), 1050–1055. ISSN 02659247. doi: 10.1002/bies.201300066.

Moghaddam, A. H., Moghaddam, M. H. a Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance*, **21**(41), 89–93. ISSN 20771886. doi: 10.1016/j.jefas.2016.07.002.

Montgomerie, S., Sundararaj, S., Gallin, W. J. a Wishart, D. S. (2006). Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinformatics*, **7**, 301.

Opitz, C. A., Kulke, M., Leake, M. C., Neagoe, C., Hinssen, H., Hajjar, R. J. a Linke, W. A. (2003). Damped elastic recoil of the titin spring in myofibrils of human myocardium. *Proceedings of the National Academy of Sciences of the United States of America*, **100**(22), 12688–12693. ISSN 00278424.

Ozkan, S. B., Wu, G. A., Chodera, J. D. a Dill, K. A. (2007). Protein folding by zipping and assembly. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(29), 11987. ISSN 00278424.

Park, B. H. a Levitt, M. (1995). The complexity and accuracy of discrete state models of protein structure. *J. Mol. Biol.*, **249**(2), 493–507.

Patel, M. S. a Mazumdar, H. S. (2014). Knowledge base and neural network approach for protein secondary structure prediction. *Journal of Theoretical Biology*, **361**, 182–189. ISSN 00225193. doi: 10.1016/j.jtbi.2014.08.005.

Pauling, L., Corey, R. B. a Branson, H. R. (1951). The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, **37**(4), 205–211. ISSN 00278424.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. a Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Peng, J., Bo, L. a Xu, J. (2009). Conditional neural fields. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I. a Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1419–1427. Curran Associates, Inc. URL `http://papers.nips.cc/paper/3869-conditional-neural-fields.pdf`.

Petsko, G. A. (2004). Protein structure and function. In *Protein structure and function / Gregory A. Petsko, Dagmar Ringe*. ISBN 1405119225.

Plaxco, K. W., Simons, K. T. a Baker, D. (1998). Contact order, transition state placement and the refolding rates of single domain proteins. *Journal Of Molecular Biology*, **277**(4), 985–94. ISSN 00222836.

Pollastri, G., Przybylski, D., Rost, B. a Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, **47**(2), 228–235. ISSN 1097-0134. doi: 10.1002/prot.10082. URL `http://dx.doi.org/10.1002/prot.10082`.

Qi, Y., Oja, M., Weston, J. a Noble, W. S. (2012). A unified multitask architecture for predicting local protein properties. *PLoS ONE*, **7**(3), e32235.

Qian, N. a Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, **202**(4), 865–884.

Rashid, S., Saraswathi, S., Kloczkowski, A., Sundaram, S. a Kolinski, A. (2016). Protein secondary structure prediction using a small training set (compact model) combined with a complex-valued neural network approach. *BMC Bioinformatics*, **17**(1), 362. ISSN 14712105. doi: 10.1186/s12859-016-1209-0.

Rawat, W. a Wang, Z. (2017). Deep convolutional neural networks for image classification. *Neural Computation*, pages 1–98. ISSN 1530888X. doi: 10.1162/NECO˙a˙00990.

Rost, B. (2001). Review: protein secondary structure prediction continues to rise. *Journal Of Structural Biology*, **134**(2-3), 204–18. ISSN 10478477.

Rost, B., Sander, C. a Schneider, R. (1994). Redefining the goals of protein secondary structure prediction. *J. Mol. Biol.*, **235**(1), 13–26.

Rumelhart, D. E., Hinton, G. E. a Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA. ISBN 0-262-01097-6. URL `http://dl.acm.org/citation.cfm?id=65669.104451`.

Sakti, S., Ilham, F., Neubig, G., Toda, T., Purwarianti, A. a Nakamura, S. (2015). Incremental sentence compression using lstm recurrent networks. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 252–258. doi: 10.1109/ASRU.2015.7404802.

Salamov, A. a Solovyev, V. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of molecular biology*, **247**(1). ISSN 00222836.

Schaffhausen, J. (2012). Advances in structure-based drug design. *Trends In Pharmacological Sciences*, **33**(5), 223. ISSN 18733735. doi: 10.1016/j.tips.2012.03.011.

Scheraga, H. A. (1960). Structural studies of ribonuclease. iii. a model for the secondary and tertiary structure1,2. *Journal of the American Chemical Society*, **82**(15), 3847–3852. doi: 10.1021/ja01500a015.

Siniscalchi, S. M., Svendsen, T. a Lee, C.-H. (2014). An artificial neural network approach to automatic speech processing. *Neurocomputing*, **140**, 326–338. ISSN 09252312. doi: 10.1016/j.neucom.2014.03.005.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. a Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**(6), 1929–1958. ISSN 15324435.

Sun, X.-D. a Huang, R.-B. (2006). Prediction of protein structural classes using support vector machines. *Amino Acids*, **30**(4), 469–475. ISSN 1438-2199. doi: 10.1007/s00726-005-0239-0. URL `http://dx.doi.org/10.1007/s00726-005-0239-0`.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B. a Wu, C. H. (2015). Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**(6), 926. doi: 10.1093/bioinformatics/btu739. URL `+http://dx.doi.org/10.1093/bioinformatics/btu739`.

Sønderby, S. K. a Winther, O. (2014). Protein secondary structure prediction with long short term memory networks.

Taherzadeh, G., Zhou, Y., Liew, A. W. a Yang, Y. (2016). Sequence-based prediction of protein-carbohydrate binding sites using support vector machines. *Journal Of Chemical Information And Modeling*, **56**(10), 2115–2122. ISSN 1549960X.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, **abs/1605.02688**. URL `http://arxiv.org/abs/1605.02688`.

Wang, S., Peng, J., Ma, J. a Xu, J. (2016). Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, **6**, 18962. ISSN 20452322. doi: 10.1038/srep18962.

Wang, Z., Zhao, F., Peng, J. a Xu, J. (2011). Protein 8-class secondary structure prediction using conditional neural fields. *Proteomics*, **11**(19), 3786–3792.

Wang, Z. X. (1998). A re-estimation for the total numbers of protein folds and superfamilies. *Protein Engineering*, **11**(8), 621–6. ISSN 02692139.

Ward, J. J., McGuffin, L. J., Buxton, B. F. a Jones, D. T. (2003). Secondary structure prediction with support vector machines. *Bioinformatics*, **19**(13), 1650–1655.

Whittle, P. J. a Blundell, T. L. (1994). Protein structure–based drug design. *Annual Review Of Biophysics And Biomolecular Structure*, **23**, 349–75. ISSN 10568700.

Wood, M. J. a Hirst, J. D. (2005). Protein secondary structure prediction with dihedral angles. *Proteins*, **59**(3), 476–81. ISSN 10970134.

Zemla, A., Venclovas, C., Fidelis, K. a Rost, B. (1999). A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, **34**(2), 220–3. ISSN 08873585.

Zhang, H., Zhang, T., Chen, K., Kedarisetti, K. D., Mizianty, M. J., Bao, Q., Stach, W. a Kurgan, L. (2011). Critical assessment of high-throughput standalone methods for secondary structure prediction. *Briefings in Bioinformatics*, **12**(6), 672–688. ISSN 14675463.

Zhou, H. a Zhou, Y. (2005). SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. *Bioinformatics*, **21** (18), 3615–3621.

Zhou, J. a Troyanskaya, O. G. (2014). Deep supervised and convolutional generative stochastic network for protein secondary structure prediction.

Zvelebil, M. J., Barton, G. J., Taylor, W. R. a Sternberg, M. J. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.*, **195**(4), 957–961.

Přílohy

Neuronů	Minimum	Maximum	Průměr	Odchylka	Parametrů
392, 392	67.4735	67.7132	67.5722	0.090891	312828
300, 300	67.3317	67.8194	67.6300	0.180747	211812
200, 200	67.3663	67.8183	67.5485	0.147950	121212
100, 100	67.2080	67.8110	67.4338	0.211702	50612
50, 50	67.2878	67.6992	67.4447	0.147298	22812
25, 25	67.3593	67.8541	67.5673	0.175460	10787
10, 10	66.6979	67.6163	67.1997	0.376581	4172
300, 200	67.4547	68.1152	67.7043	0.268898	180512
200, 100	67.4412	67.7979	67.6408	0.119651	99912
100, 50	67.2249	67.8484	67.6209	0.228730	44962
50, 25	67.2323	67.6186	67.4365	0.156137	21237
25, 10	66.6924	67.5192	67.2319	0.281625	10217

Tabulka A.1: Srovnání různých konfigurací dvou plně propojených neuronových vrstev v modelu. Tabulka zobrazuje hodnoty Q_8^U skóre v procentech na datesetu `CB513` a počet parametrů neuronové sítě nutných k reprezentaci dané konfigurace. Modely byly učeny na datech `CullPDB_filt`. Pro každou konfiguraci bylo natrénováno a otestováno 5 nezávislých modelů.

Zorné pole	Filtrů	Q_8^U		SOV_8	
		Průměr	Odchylka	Průměr	Odchylka
3	1	67.366	0.146	65.050	0.106
3	2	67.510	0.278	65.299	0.510
3	3	67.312	0.299	65.079	0.401
5	1	67.279	0.369	64.818	0.364
5	2	67.427	0.225	65.218	0.296
5	3	67.244	0.498	64.966	0.473
7	1	67.330	0.275	64.973	0.391
7	2	67.415	0.185	64.993	0.272
7	3	67.468	0.229	65.232	0.402

Tabulka A.2: Výsledky různých konfigurací konvolučního bloku. Tabulka zobrazuje průměrné Q_8^U a SOV_8 skóre v procentech na testovací množině dat `CB513`. Pro každou konfiguraci bylo natrénováno a otestováno 5 nezávislých modelů.

Dimenze	Vrstev	Q_8^U		SOV_8	
		Průměr	Odchylka	Průměr	Odchylka
140	1	65.628	0.141	63.143	0.122
140	2	65.934	0.078	63.816	0.172
140	3	65.873	0.127	63.768	0.032
100	1	65.682	0.378	63.059	0.354
100	2	65.434	0.210	62.965	0.252
100	3	65.755	0.036	63.549	0.152
80	1	65.967	0.208	63.220	0.181
80	2	65.715	0.309	63.275	0.399
80	3	65.633	0.216	63.381	0.251
60	1	65.740	0.216	62.880	0.304
60	2	65.689	0.154	63.184	0.204
60	3	65.483	0.141	63.121	0.133

Tabulka A.3: Výsledky různých konfigurací rekurentního bloku výchozího modelu s využitím LSTM, první část. Tabulka zobrazuje průměrné Q_8^U a SOV_8 skóre v procentech na testovací množině dat `CB513`. Pro každou konfiguraci bylo natrénováno a otestováno 5 nezávislých modelů.

Dimenze	Vrstev	Q_8^U		SOV_8	
		Průměr	Odchylka	Průměr	Odchylka
40	1	65.746	0.332	63.004	0.465
40	2	65.513	0.368	62.963	0.428
40	3	65.418	0.323	62.858	0.338
20	1	65.628	0.223	62.851	0.322
20	2	65.337	0.263	62.610	0.304
20	3	65.113	0.606	62.426	0.520
5	1	64.476	0.735	61.708	0.732
5	2	64.679	0.381	62.095	0.387
5	3	64.797	1.027	62.114	1.254

Tabulka A.4: Výsledky různých konfigurací rekurentního bloku výchozího modelu s využitím LSTM, druhá část. Tabulka zobrazuje průměrné Q_8^U a SOV_8 skóre v procentech na testovací množině dat `CB513`. Pro každou konfiguraci bylo natrénováno a otestováno 5 nezávislých modelů.

Dimenze	Vrstev	Q_8^U		SOV_8	
		Průměr	Odchylka	Průměr	Odchylka
100	1	66.042	0.201	63.376	0.233
100	2	65.372	0.446	62.752	0.476
100	3	65.373	0.225	62.904	0.412
100	4	65.665	0.182	63.551	0.196
140	1	66.008	0.276	63.345	0.388
140	2	65.353	0.147	63.180	0.234
140	3	65.804	0.105	63.610	0.174
140	4	65.935	0.171	63.985	0.138
180	1	66.038	0.227	63.358	0.354
180	2	65.504	0.251	62.805	0.216
180	3	65.946	0.333	63.884	0.429
180	4	66.230	0.156	64.334	0.241

Tabulka A.5: Výsledky různých konfigurací rekurentního bloku výchozího modelu s využitím jednoduché rekurentní vrstvy. Tabulka zobrazuje průměrné Q_8^U a SOV_8 skóre v procentech na testovací množině dat `CB513`. Pro každou konfiguraci bylo natrénováno a otestováno 5 nezávislých modelů.

Funkce	Vrstev	Neuronů	Q_8^U		SOV_8	
			Průměr	Odchylka	Průměr	Odchylka
logistic	0	-	72.253	0.066	71.140	0.189
logistic	1	10	72.398	0.106	71.006	0.162
logistic	1	20	72.366	0.135	71.035	0.200
logistic	1	40	72.301	0.071	71.105	0.171
logistic	1	80	72.302	0.099	71.108	0.263
logistic	1	160	72.333	0.086	71.073	0.242
logistic	2	10	72.293	0.136	70.451	0.261
logistic	2	20	72.431	0.127	70.797	0.163
logistic	2	40	72.452	0.118	70.904	0.274
logistic	2	80	72.438	0.111	70.808	0.381
logistic	2	160	72.494	0.090	70.752	0.312
ReLU	0	-	72.288	0.063	71.104	0.201
ReLU	1	10	72.329	0.069	71.033	0.220
ReLU	1	20	72.377	0.100	70.989	0.230
ReLU	1	40	72.441	0.089	70.607	0.319
ReLU	1	80	72.489	0.084	70.504	0.310
ReLU	1	160	72.501	0.089	70.454	0.250
ReLU	2	10	72.331	0.085	70.969	0.225
ReLU	2	20	72.393	0.104	70.780	0.373
ReLU	2	40	72.427	0.103	70.438	0.304
ReLU	2	80	72.465	0.114	70.397	0.373
ReLU	2	160	72.522	0.086	70.525	0.326

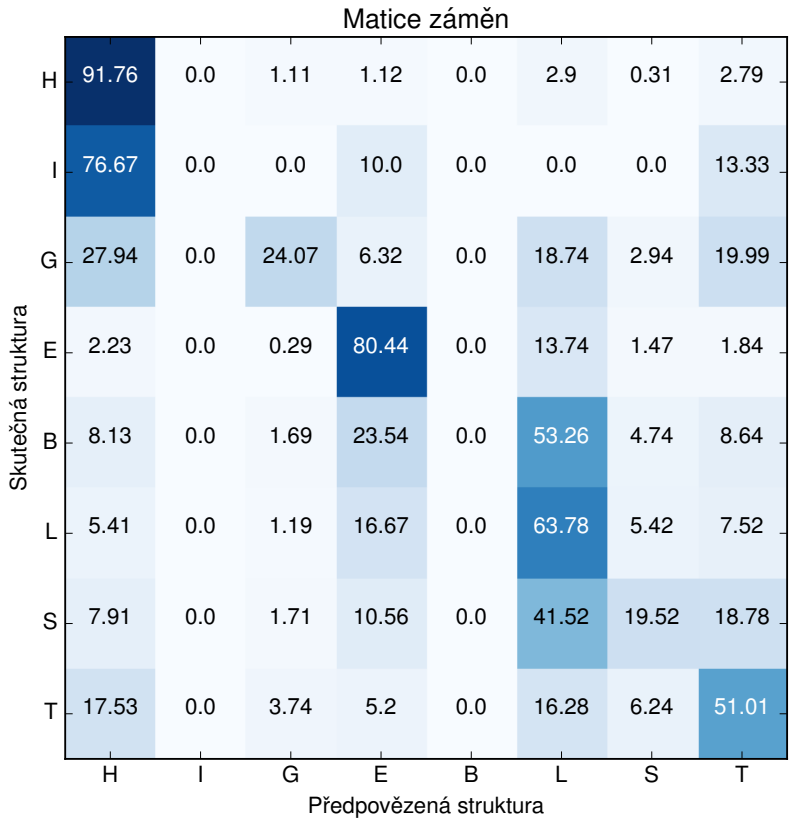
Tabulka A.6: Q_8^U a SOV_8 skóre na validačních datech pro všechny testované konfigurace vyhlazovací neuronové sítě. Pro každou konfiguraci bylo spuštěno 40 nezávislých běhů.

Kernelová funkce	Q_8^U	SOV_8
RBF	82.583	79.168
Sigmoid	77.663	73.146
Lineární	82.526	78.751
Polynomiální st. 2	82.559	79.286
Polynomiální st. 3	82.478	79.563
Polynomiální st. 4	82.210	79.142

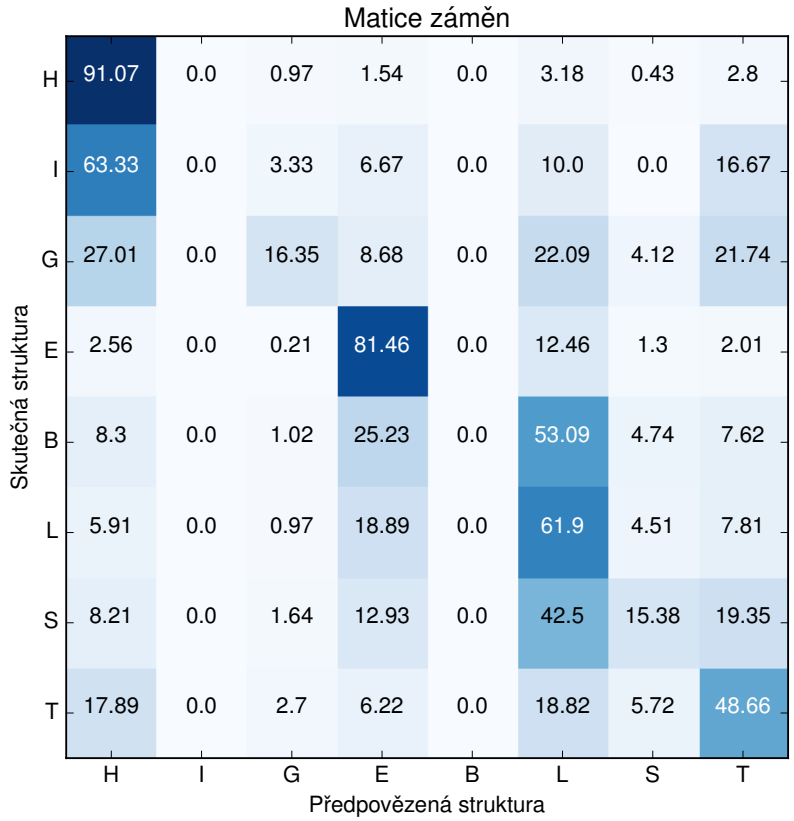
Tabulka A.7: Q_3^U a SOV_3 skóre SVM vyhlazovacích modelů sekundární struktury S_3 s plovoucím oknem velikosti 19 s různými kernelovými funkcemi na validační množině. Hodnoty jsou udávány v procentech.

Funkce	Vrstev	Neuronů	Q_8^U		SOV_8	
			Průměr	Odchylka	Průměr	Odchylka
logistic	0	-	82.669	0.021	79.684	0.114
logistic	1	10	82.667	0.031	79.596	0.152
logistic	1	20	82.665	0.033	79.558	0.185
logistic	1	40	82.672	0.032	79.567	0.166
logistic	1	80	82.670	0.030	79.543	0.135
logistic	1	160	82.657	0.031	79.619	0.135
logistic	2	10	82.647	0.051	79.455	0.272
logistic	2	20	82.666	0.061	79.429	0.281
logistic	2	40	82.656	0.055	79.551	0.232
logistic	2	80	82.648	0.039	79.601	0.156
logistic	2	160	82.611	0.049	79.525	0.223
ReLU	0	-	82.671	0.030	79.673	0.117
ReLU	1	10	82.655	0.038	79.505	0.286
ReLU	1	20	82.635	0.061	79.207	0.314
ReLU	1	40	82.612	0.063	79.016	0.324
ReLU	1	80	82.641	0.052	79.001	0.291
ReLU	2	160	82.624	0.059	78.953	0.191
ReLU	2	10	82.641	0.050	79.250	0.308
ReLU	2	20	82.626	0.062	79.143	0.384
ReLU	2	40	82.618	0.046	78.991	0.225
ReLU	2	80	82.635	0.054	79.165	0.204
ReLU	2	160	82.620	0.055	79.274	0.199

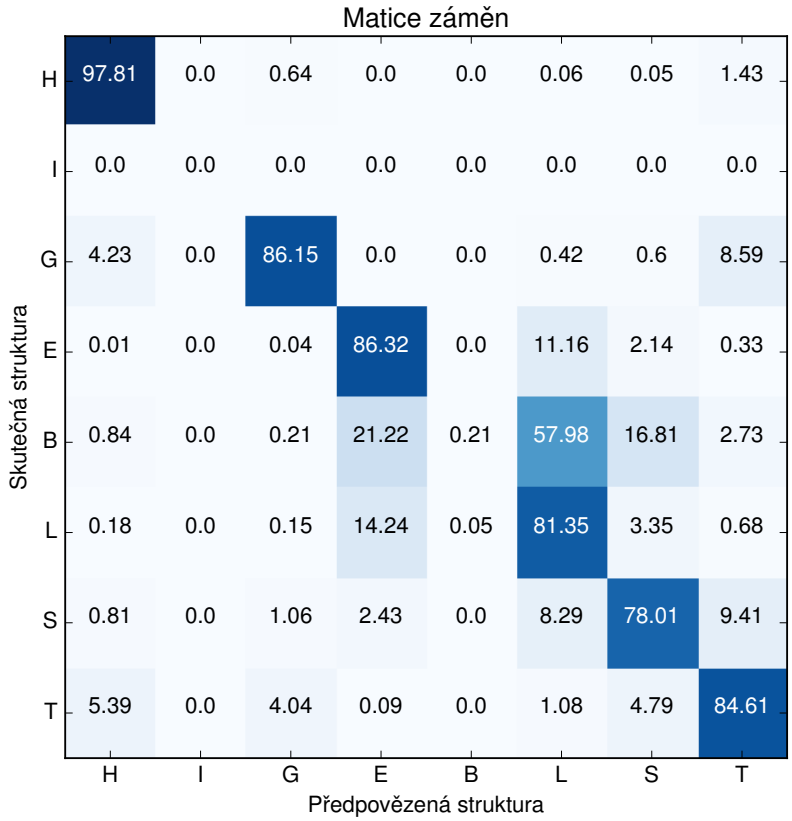
Tabulka A.8: Q_3^U a SOV_3 skóre na validačních datech pro všechny testované konfigurace vyhlazovací neuronové sítě pro sekundární struktury S_3 . Pro každou konfiguraci bylo spuštěno 40 nezávislých běhů.



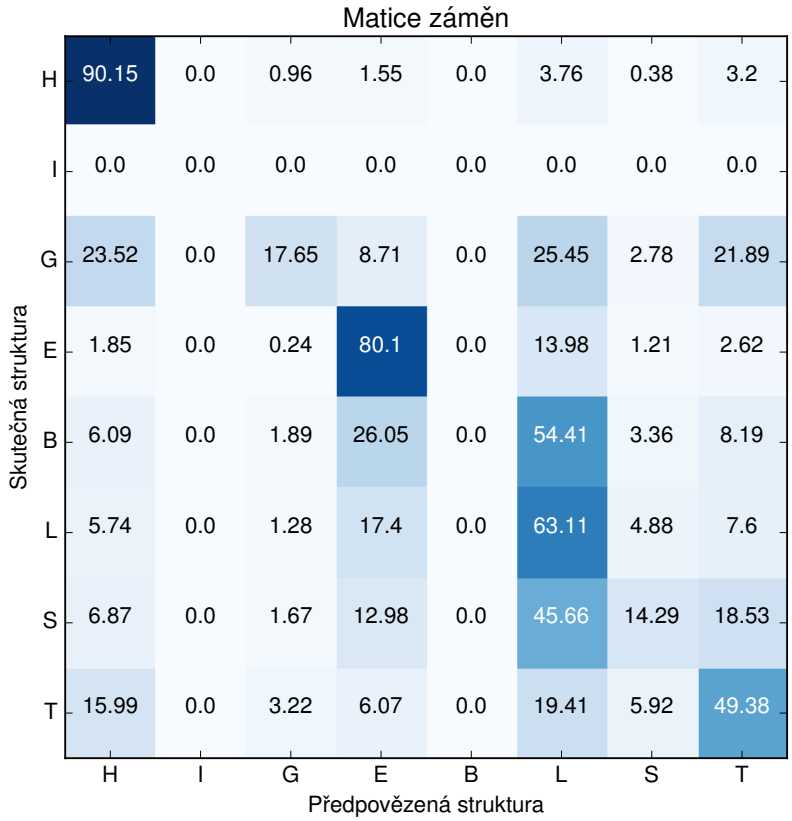
Obrázek B.1: Matice záměn pro nejlepší model s výchozí architekturou na testovacích datech `CB513`. Hodnoty jsou zobrazeny v procentech.



Obrázek B.2: Matice záměn pro nejlepší model využívající LSTM paměťovou buňku na testovacích datech `CB513`. Rekurentní blok v tomto případě obsahoval jednu vrstvy obousměrných LSTM buněk s dimenzí výstupu 100. Hodnoty jsou zobrazeny v procentech.



Obrázek B.3: Matice záměn pro nejlepší model z 5 běhů učení výchozího modelu bez bloku pro kódování vstupu. Hodnoty jsou zobrazeny v procentech. Vstupem modelu byla pouze předzpracovaná terciární struktura. Trénovací i testovací data byla vybrána z `CullPDB_STR`.



Obrázek B.4: Matice záměn pro nejlepší model z 5 běhů učení výchozího modelu s klasickými vstupy. Hodnoty jsou zobrazeny v procentech. Trénovací i testovací data byla vybrána z `CullPDB_STR`.

Obsah přiloženého CD

adresář `FastProteinPSSM`

Adresář obsahující zdrojové kódy programu pro zrychlený výpočet PSSM proteinů využívaný v sekci 4.3.1.

soubor `FastProteinPSSM.sh`

Spouštěcí soubor programu.

soubory `README.md` **a** `README.pdf`

Uživatelská dokumentace programu (anglicky).

adresář `PAS2RS`

Adresář obsahující zdrojové kódy programu pro převod sekvence absolutních pozic centrálních uhlíků na sekvenci relativních pozic využívaný v sekci 4.3.2.

soubor `PAS2ER.sh`

Spouštěcí soubor programu.

soubory `README.md` **a** `README.pdf`

Uživatelská dokumentace programu (anglicky).

adresář `neural_networks`

Adresář obsahující zdrojové kódy všech neuronových sítí využitých v této práci a skripty zajišťující učení a testování modelů.

adresář `models`

Adresář s definicemi modelů.

soubor `model_train.sh`

Spouštěcí soubor učení modelů.

soubor `model_test.sh`

Spouštěcí soubor testování modelů.

soubory `README.md` **a** `README.pdf`

Uživatelská dokumentace programů a modelů (česky).

adresář `data`

Adresář obsahující veřejné datasety `CB513`, `CullPDB` a `CullPDB_filt` rozšířené o PSSM matice proteinů získané v rámci této práce pomocí programu `FastProteinPSSM`. Adresář dále obsahuje datasety `CullPDB_FCC` a `CullPDB_STR` popsané v sekci 5.5.

soubory `README.md` **a** `README.pdf`

Detailnější popis dat (česky).

soubor `prace.pdf`

Elektronická verze této práce.