VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

http://fit.vut.cz

# Konvoluční neuronové sítě 2022

# Doporučovací systémy

Jan Lorenc, Marek Hlavačka, Jaromír Wysoglad

**Abstrakt**

Cílem práce je vytvořit model pro doporučovací systém nad datasetem Goodreads. Doporučovací systémy obecně využívají kolaborativní nebo obsahové filtrování. Tato práce prezentuje hybridní řešení využívající oba zmíněné přístupy. Model je trénován na explicitních hodnoceních knih a řeší úlohu predikce uživatelského hodnocení. K validaci jsou provedeny experimenty na podmnožině Goodreads datasetu dostupné na platformě Kaggle, jejichž výsledky jsou porovnány s existujícími řešeními.

**Klíčová slova:** Doporučovací systémy — Hybridní doporučovací systém — Kolaborativní filtrování — Neuronové kolaborativní filtrování — Maticová faktorizace — Obsahové filtrování — Regrese — Strojové učení — Goodreads — Kaggle

## 1. Úvod

Doporučovací systémy v poslední době zesílily na významu, neboť jsou dnes jedním z pilířů perzonalizace. Člověk se tak s nimi setkává denně na e-shopech, sociálních sítích, mediálních platformách apod. Toto vede k neustálé snaze vylepšování doporučovacích systémů za účelem maximalizovat uživatelskou zkušenost.

Existují dvě základní dělení přístupů k řešení doporučování. Prvním je kolaborativní versus obsahové filtrování. Kolaborativní přístup hledá podobnosti uživatelů/produktů na základě toho, kteří uživatelé interagovali se stejnými produkty. Lze to vnímat jako podobnost na základě popularity. Obsahové filtrování naopak pohlíží na podobnost atributů uživatelů či produktů. Výhodou obsahového filtrování je to, že netrpí tzv. cold-start problémem, tedy pokud se objeví nový produkt, stále jsme schopni ho doporučit, neboť je podobný jiným produktům. U kolaborativního přístupu dojde k selhání, neboť pro něj neexistují žádné interakce. Druhým základním dělením jsou explicitní versus implicitní interakce. Explicitní interakce znamená jasné hodnocení např. na stupnici 1–5. Udává tedy, zda-li se produkt uživateli líbil a jak moc. Implicitní interakce znamená jakoukoliv formu interakce s produktem (na e-shopu např. kliknutí na produkt nebo koupení produktu). Nelze vyvodit, jestli se uživateli produkt skutečně líbil, nicméně mnohem snáze a ve větším množství je lze od uživatelů získat.

V této práci se zaměřujeme na kombinaci kolaborativního a obsahového filtrování. Snahou je vylepšit state-of-the-art řešení kolaborativního přístupu článku Neural Collaborative Filtering [1] (dále NCF) o obsahové informace produktů. Dále chceme nejen rozpoznat dobrý a špatný produkt jako ve článku, nicméně i predikovat uživatelovo hodnocení. Namísto binární klasifikace se tedy zabýváme regresním problémem, k čemuž proto využíváme pouze explicitní interakce.

Úlohu jsme se rozhodli řešit na oblíbeném datasetu Goodreads [2, 3]. Vzhledem k velikosti datasetu jsme se zaměřili pouze na komiksy a grafické knihy. Pro jednoznačné vyhodnocení porovnáním s jinými řešeními jsme použili další podmnožinu Goodreads datasetu, kterou je goodbooks-10k na platformě Kaggle [4].

Přínosem práce je nové hybridní řešení kolaborativního a obsahového filtrování, které překonává aktuální přístupy predikce uživatelského hodnocení. Dále práce dokazuje schopnost NCF architektury být úspěšně modifikována a rozšířena.

Článek je rozdělen následovně. V sekci 2 je popsáno získání a zpracování dat. Sekce 3 prezentuje vytvoření embeddingů knih na základě jejich metadat. Hybridní architektura je ukázána v sekci 4 a v sekci 5 se dále vyhodnocují dosažené výsledky.

## 2. Zpracování dat

V této práci bylo využito aktuální datové sady Goodreads z května roku 2019, která již neobsahuje duplicitní záznamy, oproti verzi z konce roku 2017 [2, 3].

Celkem tato datová sada obsahuje 3 skupiny dat, jimiž jsou metadata knih, interakce mezi uživatelem a knihami a podrobné recenze knih od uživatelů. V této práci jsou využívány pouze první dvě zmíněné datové sady. Z důvodu velikosti datových sad existují také podmnožiny podle žánrů. V našem případě bylo tedy využito datové sady s komiksy a grafikou [2, 3].

Pro vyhodnocení byla použita datová sada nazývající se goodbooks-10k. Jedná se o datovou sadu obsahující 10 000 populárních knih obsahujících okolo 100 hodnocení pro každou knihu. Výhodou této sady je možnost porovnání výsledků s jinými řešeními [4].

Parametry využívaných datových sad nalezneme v tabulce 1.

| Datová sada | Počet knih | Počet interakcí |
|---|---|---|
| Kompletní datová sada | 2 360 65 | 228 648 342 |
| Kaggle – goodbooks | 10 000 | 981 756 |
| Komiksy a grafika | 89 411 | 7 347 630 |

**Tabulka 1.** Přehled využitých datových sad [2, 3, 4]

Pro úpravu dat k vytvoření embeddingů knih bylo třeba správně zvolit atributy, které obsahují podstatné informace o dané knize. V našem případě byly z textových atributů vybrány titul, vydavatel, popis knihy, seznam autorů (bez zahrnutí ilustrátorů, překladatelů atp.), knižní série a žánry. Všechny tyto hodnoty byly spojeny do řetězce a uloženy jako ID knihy a řetězec. Dále byl zvlášť vytvořen i vektor číselných atributů skládající se z ID knihy, počtu hodnocení, průměrného hodnocení, počtu stran a počtu textových recenzí dané knihy. V případě nevalidní hodnoty tj. vstup není v číselném tvaru, došlo k nahrazení číslem -1. Pokud kniha ještě nebyla hodnocena, tedy nemá žádné explicitní interakce, je ignorována.

Při zpracování interakcí bylo potřeba provést několik úprav datové sady, tak aby odpovídala požadavkům pro další práci. Byly tedy využity pouze sloupce pro hodnocení, ID uživatele a ID knihy. Z důvodu práce s explicitními interakcemi byly využity pouze interakce obsahující hodnocení 1–5, zbylé interakce se netýkaly hodnocení. Dále bylo provedeno namapování ID uživatele na číslo. Na závěr došlo k odstranění interakcí uživatelů, kteří jich nemají dostatek. Například pro datovou sadu komiksů byly zachovány pouze interakce uživatelů s alespoň 5 interakcemi.

Pro datovou sadu goodbooks-10k platí to stejné jako výše. Byly staženy metadata knih a interakce, z nichž však musely být vyfiltrovány duplicity. Dále v interakcích došlo k namapování ID knih, neboť se tam nacházelo jen pořadí. Na závěr se z již výše zpracovaných metadat knih (textových i číselných) vyfiltrovaly dle ID knih pouze ty, které se v goodbooks-10k nachází.

## 3. Embeddingy knih na základě atributů

Po zpracování dat přišlo na řadu vytvoření embeddingu pro každou knihu. V našem řešení se každý embedding skládá ze dvou samostatných částí, které jsou nakonec spojeny do jednoho embeddingu.

První část embeddingu je vytvořena z textových atributů knihy. Pro zpracování těchto dat byla použita tf-idf vektorizace[1] pro 10 000 slov s nejčastějším výskytem napříč celým datasetem. Při tf-idf vektorizaci je pro každou knihu vypočítán vektor o velikosti 10 000. Každá položka tohoto vektoru souvisí s jedním ze zkoumaných slov a vyjadřuje množství použití daného slova u dané knihy v porovnání s ostatními knihami. Tedy pokud se nějaké slovo, například jméno autora, vyskytuje u dané knihy mnohokrát, zatímco u ostatních knih se toto slovo vyskytuje pouze minimálně nebo vůbec, tak na toto místo ve vektoru uložíme vysokou hodnotu. Naopak pokud se nějaké slovo u dané knihy vyskytuje pouze minimálně, nebo se vyskytuje u mnoha knih s podobnou četností, tak na toto místo ve vektoru uložíme nízkou hodnotu. Takto získaný vektor nám matematicky popisuje získaná textová data o každé knize takovým způsobem, že podobné knihy mají podobné vektory a odlišné knihy mají odlišné vektory.

Vektor o velikosti 10 000 je však pro popis knih zbytečně velký. Většina ze zkoumaných 10 000 slov se vyskytuje pouze u malé části knih a proto jsou u zbylých knih na těchto místech ve vektoru hodnoty 0. Pro snížení dimenzionality embeddingů jsme proto použili autoenkodér. Autoenkodér je neuronová síť skládající se ze dvou částí - z enkodéru a dekodéru. Při trénování této sítě se na vstup enkodéru postupně vkládají embeddingy o velikosti 10 000. Enkodér se embeddingy pokouší zmenšit s minimální ztrátou informace. Následně je takto zmenšený embedding předán dekodéru, který se embedding pokusí opět zvětšit do původní velikosti. Nakonec se vstup enkodéru a výstup dekodéru porovnají a celá síť se trénuje, dokud si vstup a výstup nejsou dostatečně podobné. Po natrénování se dekodér zahodí a enkodér se použije ke zmenšení výstupu tf-idf vektorizéru, čímž získáme první část embeddingu. Příklad podobnosti knih na základě vygenerovaných embeddingů si lze prohlédnout v tabulce 2.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

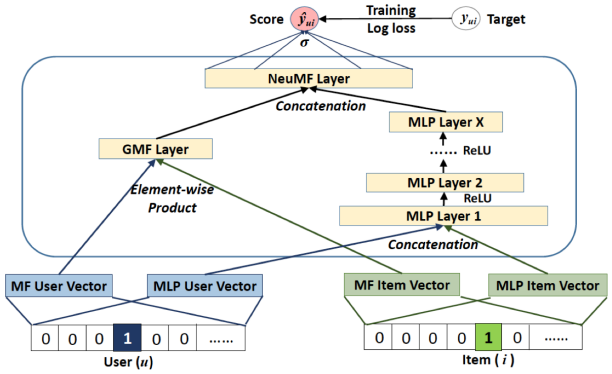
| Kniha | Podobnost |
|---|---|
| Harry Potter and the Sorcerer's Stone | 1.00 |
| Harry Potter and the Deathly Hallows | 0.93 |
| Harry Potter Boxset (Harry Potter, 1-7) | 0.93 |
| Sharpe's Rifles | 0.92 |
| Deal Breaker | 0.92 |

**Tabulka 2.** Podobnost knih ke knize "Harry Potter and the Sorcerer's Stone" na základě vygenerovaných embeddingů z textových dat na datasetu Kaggle - goodbooks [4]

Druhou část embeddingu tvoří číselná data o knize a to konkrétně: počet hodnocení, počet stran, průměrné hodnocení, počet textových uživatelských recenzí a id knihy. Tato data jsou normalizována tak, aby byly v rozmezí mezi nejmenší a nejvyšší hodnotou z první části embeddingu. Nakonec jsou tato data konkatenována k embeddingům z první části, čímž získáme výsledný embedding.

## 4. Architektura hybridního modelu

Jak již bylo zmíněno v úvodu, práce staví na NCF architektuře, kterou dále rozvíjí. NCF kombinuje maticovou faktorizaci a plně propojenou neuronovou síť dle obrázku 1.

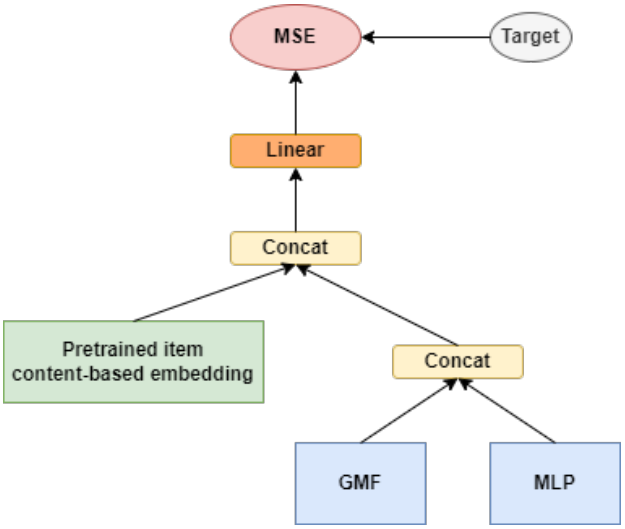



**Obrázek 1.** Originální NCF architektura (převzato z [1])

Lze vidět, že embeddingy uživatelů a knih jsou trénovány dvakrát a to zvlášť větví provádějící maticovou faktorizaci (dále GMF) a zvlášť větví s neuronovou sítí (dále MLP). Výsledky obou větví jsou zřetězeny a dány na vstup konečné lineární prediktivní vrstvě.

První provedená úprava této architektury spočívá v nahrazení chybové funkce cross-entropy metodou nejmenších čtverců (MSE). Neřešíme totiž binární klasifikaci dobrého či špatného produktu, nýbrž predikci uživatelského hodnocení.

Druhou modifikací je zřetězení již předučeného embeddingu produktu (v našem případě knihy) k výstupu GMF a MLP větví. Do prediktivní vrstvy tedy vstupuje vektor skládající půl na půl z výsledků kolaborativního a obsahového přístupu. Tuto upravenou architekturu ukazuje obrázek 2.




**Obrázek 2.** Hybridní architektura je založená na NCF a využívá již natrénované embeddingy na základě obsahu produktu. Výstupem je predikce hodnocení uživatele pro daný produkt.

Za základní délku embeddingu pro NCF jsme zvolili 32. Důvodů bylo několik. Autoři originálního článku [1] prováděli experimenty zejména s délkami 8, 16 a 32. Dále existuje řešení [5], jenž jejich výsledky dokázalo zreplikovat s velikostí embeddingů právě 32. Tímto řešením jsme se i při implementaci inspirovali a taktéž jsme s touto délkou dosahovali přesvědčivých výsledků. Navíc jsme potřebovali delší embedding (v porovnání s 8 nebo 16) pro efektivní reprezentaci knihy dle obsahu. Embeddingem knihy dle obsahu tak byl vektor délky 64, aby byl stejně dlouhý jako kolaborativní část, neboť zřetězení výstupů GMF a MLP dá ve výsledku vektor délky 32 + 32 = 64.

Data byla pro trénování rozdělena na trénovací, validační a testovací v poměru 70:20:10. Dále byla použita velikost batch 256 opět jako v originálním článku [1]. Nižší velikosti batch navíc nepřinesly žádné zlepšení, pouze delší trénování. Optimalizačním algoritmem byl zvolen Adam s učícím koeficientem 0.001, nicméně pouze pro první epochu, neboť již od druhé epochy docházelo k extrémnímu přetrénování. Fenomén velmi rychlého přetrénování jsme pozorovali i u mnoha jiných řešení. Po první epoše byl proto nahrazen SGD s koeficientem učení 0.0005. Tento nízký koeficient sice silně zpomalil další průběh učení, nicméně zabránil přetrénování a stále dokázal vylepšit chybovou funkci i v řádu setin. K trénování stačilo celkem 6 epoch.

## 5. Výsledky a vyhodnocení

Původním cílem práce bylo otestovat model na žánru komiksů a grafických knih (dále jen komiksů). Hlavní metrikou byla MSE chyba, která je běžná pro regresní úlohy. Za vedlejší metriku jsme zvolili přesnost klasifikace správného hodnocení (Hits). Jedná se o procento shodujících se správných hodnocení se zaokrouhlenými predikcemi. Tato metrika slouží spíše pro orientaci a nelze ji brát zcela směrodatně, neboť např. predikce 3.6 i 4.1 budou zaokrouhleny na 4 a obě si případně připíší hit, nicméně druhá je zřejmě přesnější.

Naším modelem jsme testovali celkem 3 architektury. První bylo naše výchozí řešení, tedy maticová faktorizace s prediktivní vrstvou - prakticky GMF. Druhou byla naše implementace NCF a třetí pak výsledný hybridní model. Výsledky na žánru komiksů ukazuje tabulka 3.

| Model | MSE | Hits [%] |
|---|---|---|
| GMF | 0.58789 | 52.078 |
| NCF | 0.54061 | 53.545 |
| Hybrid | 0.52594 | 55.853 |

**Tabulka 3.** Výsledky na testovacích datech komiksů

Výsledky prokazují, že NCF je opravdu lepší než běžná maticová faktorizace. Dále lze vidět, že naše hybridní řešení skutečně přineslo ovoce, neboť NCF překonává o cca 1.5 setin na MSE a o více než 2 % v přesnosti predikce hodnocení.

Nicméně, nelze říci, zda jsou tyto výsledné hodnoty v obecnosti dobré či špatné. Z toho důvodu byl model otestován i na datasetu goodbooks-10k [4] dostupným na platformě Kaggle, který je taktéž podmnožinou Goodreads datasetu. Porovnání našich 3 architektur ukazuje tabulka 4.

| Model | MSE | Hits [%] |
|---|---|---|
| GMF | 1.166043 | 38.422 |
| NCF | 0.732287 | 43.63 |
| Hybrid | 0.713658 | 46.03 |

**Tabulka 4.** Výsledky na testovacích datech z Kaggle

Opět lze pozorovat lepší výsledky hybridního modelu před NCF. Ve srovnání s komiksovým datasetem se však goodbooks-10k ukazuje být méně přívětivý, neboť na něm model vykazuje větší chybu. Porovnání s řešeními dostupných na platformě Kaggle ukazuje tabulka 5. Jedno z řešení [6] na datasetu testuje modely KNNBasic, FunkSVD, SlopeOne a CoClustering knihovny Surprise zaměřené na doporučovací systémy. Tato knihovna rozděluje dataset na několik dílů (tzv. folds), na kterých trénuje zvlášť. Ke srovnání byl použit vždy fold s nejlepším výsledkem. Další řešení [7] počítá maticovou faktorizaci algoritmem ALS frameworku Spark. Řešení ze Surprise i Spark používaly chybovou funkci RMSE, nicméně výsledky jsme pro srovnání převedli vztahem $RMSE = \sqrt{MSE}$. Další řešení [8] aplikuje klasickou maticovou faktorizaci (MF), maticovou faktorizaci zakomponovanou do jednoduché neuronové sítě (NN_MF) a plně propojenou neuronovou síť s ReLU aktivacemi (NN_1). Poslední řešení [9] taktéž používá neuronovou síť s ReLU (NN_2). Autor tohoto řešení zkouší v jiných noteboocích i kombinace s tanh aktivacemi, nicméně dosahuje téměř stejných výsledků, proto je zde již nezmiňujeme.

| Model | MSE |
|---|---|
| MF | 2.944 |
| **GMF** | **1.166043** |
| NN_MF | 0.9804 |
| Surprise SlopeOne | 0.848977 |
| Spark MF ALS | 0.80393 |
| Surprise KNNBasic | 0.799593 |
| Surprise CoClustering | 0.778982 |
| NN_1 | 0.766163 |
| NN_2 | 0.750598 |
| **NCF** | **0.732287** |
| Surprise FunkSVD | 0.718087 |
| **Hybrid** | **0.713658** |

**Tabulka 5.** Srovnání s řešeními na platformě Kaggle

Jak lze vidět, náš hybridní model si nad tímto datasetem vede nejlépe ze všech nalezených řešení. Nutno poznamenat, že NCF samo o sobě také dosahuje velmi dobrých výsledků. Z tohoto prvenství je možné vyvodit závěr, že námi navržená hybridní architektura je kvalitní a více než konkurenceschopná. Zároveň lze usoudit, že i výsledky na komiksovém datasetu jsou velmi dobré vzhledem k výrazně menší MSE chybě.

## 6. Závěr

Byla navržena a implementována hybridní architektura kolaborativního a obsahového filtrování založená na NCF. Výsledný model byl testován na dvou podmnožinách Goodreads datasetu a na obou dosahoval lepších výsledků než základní NCF. Dále byl porovnán s jinými řešeními nad datasetem goodbooks-10k a ze všech dosáhl nejlepších výsledků. Pro budoucí práci je možné se více zaměřit na předtrénování obsahových embeddingů knih nebo se pokusit zakomponovat i implicitní interakce.

## Literatura

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. et al. Neural Collaborative Filtering. *CoRR*. 2017, abs/1708.05031. Dostupné z: http://arxiv.org/abs/1708.05031.

[2] WAN, M. a MCAULEY, J. J. Item recommendation on monotonic behavior chains. In: PERA, S., EKSTRAND, M. D., AMATRIAIN, X. a O'DONOVAN, J., ed. *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. ACM, 2018, s. 86–94. Dostupné z: https://doi.org/10.1145/3240323.3240369.

[3] WAN, M., MISRA, R., NAKASHOLE, N. a MCAULEY, J. J. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In: KORHONEN, A., TRAUM, D. R. a MÀRQUEZ, L., ed. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019, s. 2605–2610. Dostupné z: https://doi.org/10.18653/v1/p19-1248.

[4] ZYGMUNTZ. *Goodbooks-10k* [online]. Dostupné z: https://www.kaggle.com/datasets/zygmunt/goodbooks-10k.

[5] GUOYANG9. *NCF* [online]. GitHub, 2019. Dostupné z: https://github.com/guoyang9/NCF.

[6] RACKAITIS, T. *Hybrid Recommender Systems with Surprise* [online]. Dostupné z: https://www.kaggle.com/code/robottums/hybrid-recommender-systems-with-surprise.

[7] PRASAD, M. L. *Eda+ALS pyspark* [online]. Dostupné z: https://www.kaggle.com/code/leelaprasadmoturi/eda-als-pyspark.

[8] SINGH, D. *Books recommendation Engine - dalwin002* [online]. Dostupné z: https://www.kaggle.com/code/dalwindr/books-recommendation-engine-dalwin002.

[9] MANEKMOTI, M. *Book Recommender using ReLU,ReLU* [online]. Dostupné z: https://www.kaggle.com/code/meetcynosure/book-recommender-using-relu-relu.