

Kvak, Daniel

**Generování žánrově specifické hudební transkripce Antonína Dvořáka
prostřednictvím variačního autoenkodéru**

Musicologica Brunensia. 2021, vol. 56, iss. 2, pp. 49-61

ISSN 1212-0391 (print); ISSN 2336-436X (online)

Stable URL (DOI): <https://doi.org/10.5817/MB2021-2-5>

Stable URL (handle): <https://hdl.handle.net/11222.digilib/144818>

License: [CC BY-SA 4.0 International](https://creativecommons.org/licenses/by-sa/4.0/)

Access Date: 16. 02. 2024

Version: 20220831

Terms of use: Digital Library of the Faculty of Arts, Masaryk University provides access to digitized documents strictly for personal use, unless otherwise specified.

Generování žánrově specifické hudební transkripce Antonína Dvořáka prostřednictvím variačního autoenkodéru

Generating Genre-Specific Musical Transcriptions of Antonín Dvořák through a Variational Autoencoder

Daniel Kvak / kvak@mail.muni.cz

Department of Musicology, Faculty of Arts, Masaryk University, Brno, CZ

Abstract

Apart from traditional deep learning tasks such as pattern recognition, stock price prediction, and machine translation, this method also finds practical application within algorithmic composition. This paper explores the use of a generative model based on unsupervised learning of a musical style from a pre-selected corpus and the subsequent prediction of samples from the estimated distribution. The model uses a Long Short-Term Memory neural network whose training data contains genre-specific melodies in symbolic representation.

Key words

algorithmic composition, artificial intelligence, autoencoder, deep learning, generative art, LSTM network, machine learning, recurrent neural network

Úvod

Hudební kompozice je považována za kreativní, intuitivní, a především neodmyslitelně lidskou schopnost. Přesto se k ní váže dlouhá historie interdisciplinárního průsečíku s výpočetními a matematickými modely, a to již od chvíle, kdy Hiller a Isaacson navrhli použití Markovovy řetězce k vytvoření automaticky generované kompozice.¹ V současné době existuje několik výzkumů, jež zkoumají využití hlubokých neuronových sítí ke generování hudebního obsahu. Dosavadní řešení zahrnují buď generování symbolické (tj. generování hudební transkripce) nebo subsymbolické (tj. generování surového zvukového signálu) reprezentace díla.

Současný stav bádání

Většina modelů pro generování hudební transkripce na bázi umělé inteligence používá rekurentní neuronovou síť (RNN) či její varianty. Důvod je nasnadě: tato třída je schopna pracovat s modelováním sekvenčních dat.² V roce 1989 využil Peter M. Todd rekurentní síť ke generování monofonních melodií, což se stalo prvním pokusem aplikovat umělé neuronové sítě v rámci algoritmické kompozice.³ Zatímco RNN je silně limitována závislostmi na delší vzdálenost v rámci časové řady,⁴ síť Long-Short Term Memory (LSTM), varianta RNN navržená Hochreiterem a Schmidhuberem, toto omezení do jisté míry redukuje.⁵ LSTM byla v algoritmické kompozici poprvé využita na přelomu tisíciletí,⁶ od té doby se stala standardem jak pro generativní hudební modely, tak pro počítačové zpracování přirozeného jazyka.

V rámci vícehlasé hudby sledujeme složité vzory podél různých os: existují zde jak sekvenční vzory mezi časovými kroky, tak harmonické intervaly mezi současnými notami. Generování polyfonní hudební transkripce je mnohem složitějším problémem než generace monofonní hudby. Dřívější projekty se rozhodly obejít tuto překážku určitými

1 HILLER, Lejaren Arthur a Leonard M. ISAACSON. *Experimental Music; Composition with an Electronic Computer*. Westport, CT, USA: Greenwood Publishing Group, 1979, 197 s. ISBN 978-0-313-22158-3.

2 Data, jež je možno pozorovat a analyzovat pomocí časové řady. Hudbu lze reprezentovat jako sled událostí a lze ji tedy modelovat jako podmíněnou pravděpodobnost mezi jednotlivými hudebními událostmi.

3 TODD, Peter M. A Connectionist Approach to Algorithmic Composition. *Computer Music Journal* [online]. 1989, 13(4). DOI: 10.2307/3679551. ISSN 01489267. Dostupné z: <https://www.jstor.org/stable/3679551?origin=crossref>.

4 Tzv. *long-term dependency*.

5 HOCHREITER, Sepp a Jürgen SCHMIDHUBER. Long Short-Term Memory. *Neural Computation* [online]. 1997, 9(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735. ISSN 0899-7667. Dostupné z: <http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735>.

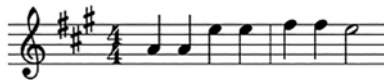
6 ECK, Douglas a Juergen SCHMIDHUBER. *A First Look at Music Composition using LSTM Recurrent Neural Networks* [online]. 2002. Manno, Switzerland: Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale. DOI:10.5555/870511.

způsoby tak, aby byl problém snadněji řešitelný. Zatímco populární model folk-rnn čerpající inspiraci v tradiční irské hudbě dokáže generovat pouze jednostopou monofonní transkripci,⁷ jiné aplikace tyto jednoduché melodie doprovázejí akordy v souladu s pravidly kompozice.⁸

Avšak mimo výše uvedené metody existuje další způsob, pomocí něhož lze modelovat vícehlasou hudební skladbu jako společnou pravděpodobnostní distribuci not a interpunkcí. Jedním z praktických cílů tohoto modelu bude využití tzv. one-hot kódování, což nám dovoluje přiřadit každému stavu unikátní token. Tyto tokeny mohou reprezentovat akordy, sekce, ale taktéž délku not či složitější značky, to vše za použití pouhých ASCII⁹ znaků.¹⁰

ABC Notation

```
[abcjs engraver="{ staffwidth: 300 }"]
X:1
K:A
A2A2e2e2|f2f2e4|
[/abcjs]
```



Obr. 1 Příklad notového zápisu pomocí symbolické reprezentace, formátu ABC.

Vymezení řešené problematiky

Ačkoliv etymologický původ slova „algoritmus“ můžeme sledovat až do první poloviny 9. století, kdy jej Al-Chorezmí (780–846) užil pro provádění aritmetiky pomocí arabských číslic, pro současné použití v oblasti informatiky a informační vědy byl převzat okolo

7 STURM, Bob L. On folk-rnn cheating and music generation [online]. 2017. Dostupné z: <https://high-noongmt.wordpress.com/2017/06/15/on-folk-rnn-cheating-and-music-generation/>.

8 FIEBRINK, Rebecca a Baptiste CARAMIAUX. The Machine Learning Algorithm as Creative Musical Tool. *Handbook on Algorithmic Music* [online]. Oxford: Oxford University Press, 2016. Dostupné z: https://www.researchgate.net/publication/309631691_The_Machine_Learning_Algorithm_as_Creative_Musical_Tool.

9 ASCII se označuje tabulku kódů, jež definuje znaky anglické abecedy, čísla a další znaky používané v informatice.

10 WALSHAW, Chris. A Statistical Analysis of the ABC Music Notation Corpus: Exploring Duplication. *Conference: Folk Music Analysis*. Istanbul, 2014. DOI:10.13140/2.1.4340.0961.

poloviny 20. století.¹¹ Slovo algoritmus definuje Český etymologický slovník jako „*soubor logických pravidel, sestavený s cílem splnit určitý logický úkol*.“¹² Gareth Loy definuje algoritmus jako sadu pravidel nebo sled operací určených k provedení nějakého úkolu nebo k jeho vyřešení s přesnou definicí pro každý krok.¹³

Proces hudební kompozice se naopak často vyznačuje metodou pokusu a omylu. Skladatel improvizuje, poslouchá a určuje, zda jsou nutné revize, přičemž průběžně evaluuje estetickou funkci části skladby ve vztahu k celku. Jestliže náš logický úkol popisujeme jako vytvoření hudební transkripce, pak můžeme algoritmickou kompozici chápat jako automatizovaný proces hudební kompozice či jakékoliv její části. Profesor muzikologie Adam Alpern popsál algoritmickou kompozici jako „*process of using some formal process to make music with minimal human intervention*.“¹⁴

Jednou z výhod použití strojového učení k extrahování vzorů z dat je to, že umožňuje výzkumníkovi obejít potřebu formulovat podrobnosti na nízké úrovni obsažené v datech, jež mohou být jen obtížně vyčísleny či definovány.¹⁵ Pro mnoho aplikací jsou neuronové sítě s hlubokým učením neúčinnější metodou k detekci užitečných informací v souborech dat a jejich použití k reinterpetaci nových vzorků s obdobným obsahem. Tímto způsobem zavádí strojové učení nový přístup k pochopení kreativních počínů, v němž výběr příkladů a definování mapování umožňuje vznik originálních forem tvůrčího vyjádření.¹⁶ Kromě výběru výpočetně neefektivnější architektury či vhodných parametrů se hlavní datový analytik stává tím, kdo definuje prostor interpretace výběrem datové sady, jež tento prostor představuje, výběrem vhodné aktivační funkce pro trénování neuronové sítě; v případě učení s učitelem pak taktéž rozhoduje, jaké výstupy lze vrátit při dotazování sítě.

Neuronové sítě a hluboké učení

Umělé neuronové sítě jsou jedním z nejpobulárnějších paradigmat současné umělé inteligence. Jedná se o výpočetní model sloužící k realizaci metod strojového učení, při nichž se počítač učí splnit určitý úkol analýzou tréninkových příkladů.¹⁷ Do jisté míry můžeme hovořit o primitivním matematickém modelu mozku, který se používá k paralelnímu zpracování nelineárních vztahů mezi vstupy a výstupy, jako to dělá lidský mozek každou

11 DAVIS, Martin. *Engines of Logic: Mathematicians and the Origin of the Computer*. USA: W. W. Norton & Co., Inc., 2001. ISBN 978-0-393-32229-3.

12 REJZEK, Jiří. *Český etymologický slovník*. Voznice: Leda, 2001. ISBN 80-85927-85-3. Heslo „algoritmus“.

13 LOY, Gareth. *Musimathics, Volume 1: The Mathematical Foundations of Music*. Reprint edition. vyd. Cambridge, Mass.: The MIT Press, 2011. ISBN 978-0-262-51655-6.

14 Přeloženo: „*Proces používání určitého formálního procesu k tvorbě hudby s minimálním lidským zásahem*.“ ALPERN, Adam. *Techniques for Algorithmic Composition of Music* [online]. Hampshire College, 1995. DOI:10.1.1.23.9364.

15 FERNANDEZ, Jose David a Francisco VICO. AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*. 2013, roč. 48, s. 513–582. ISSN 1076-9757. DOI: 10.1613/jair.3908

16 Ibid.

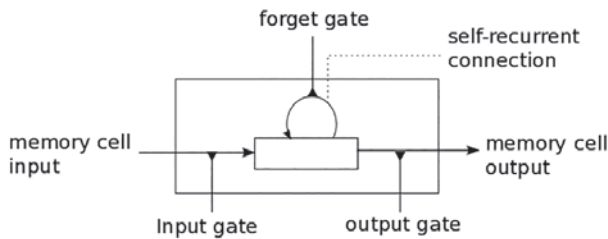
17 RUSSELL, Stuart J. a Peter NORVIG. *Artificial intelligence: a modern approach*. Third edition. Boston: Pearson, 2016. Prentice Hall series in artificial intelligence. ISBN 9781292153964.

sekundu. Umělé neuronové sítě nacházejí své využití v široké škále úkolů, přičemž mezi nejpobulárnější úlohy řadíme klasifikaci, regresi a klastrování.

Marně bychom hledali konsensuální definici hlubokého učení. *Deep learning* je podskupinou strojového učení, jež využívá výpočetní model na bázi hluboké neuronové sítě. Společným znakem této třídy je pojem *deep*, jenž můžeme definovat jako existenci více vrstev neuronové sítě. Hloubka se může u modelů užívající hluboké učení pohybovat i v řádech mnoha desítek vrstev.¹⁸ Hluboké učení je jedním z nejdynamičtější se rozvíjejících paradigmat pro výpočetní úlohy. K rozmachu této disciplíny došlo poté, co výrazně překonalo standardní techniky pro klasifikační a regresní úlohy pomocí algoritmů a filtrů.

Za současný stav vděčíme především:

- Technologickému pokroku, který byl podmíněn vývojem rekurentních modelů¹⁹ a LSTM buněk.²⁰ Tyto nové poznatky řeší původní neefektivnost trénování rekurentních neuronových sítí.
- Dostupnosti objemných dat, tzv. big data.
- Dostupnosti účinného a levného výpočetního výkonu prostřednictvím GPU (grafické karty).



Obr. 2 Zjednodušená struktura LSTM buňky.

Redukce dimenzionality

Jednotlivé vrstvy hluboké neuronové sítě zpracovávají různé úrovně abstrakcí, jež jsou učením extrahovány ze vstupních dat. Tato metoda se nazývá extrakce příznaků.²¹ Motivací je nalezení podmnožiny příznaků, která bude disponovat dostatkem korelací pro vytvoření očekávaného rozhodnutí. Data jsou transformována z vysokorozměrného do nízkorozměrného prostoru tak, aby si nízkodimenzionální reprezentace zachovala některé důležité

18 HE, Kaiming et al. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]* [online]. 2015. Dostupné z: <http://arxiv.org/abs/1512.03385>.

19 Oproti dopředným sítím se v případě rekurentní neuronové sítě (RNN) signál nešíří pouze od vstupní vrstvy směrem k vrstvě výstupní, ale dochází i ke zpětnovazebnému přenosu informace od vrstev vyšších zpět do vrstev nižších. Rekurentní neuronové sítě do jisté míry čerpají inspiraci z autoregresních modelů.

20 LSTM (Long Short-Term Memory) je model speciálních paměťových bloků, které disponují tzv. *forget* bránou. Díky tomu se může neuron rozhodnout, které informace jsou relevantní a které může vypustit. Výstupní hodnoty neuronů mají vliv na synaptické váhy sítě.

21 Tzv. *feature extraction*.

vlastnosti původních dat, ideálně blízkých jejich vnitřní dimenzi. Zbytek informací pak označujeme jako redundantní šum: tento fenomén se nazývá redukce dimenzionality.

Cílem extrakce příznaků je nalézt takové zobrazení množiny X do množiny X' , jež by maximalizovalo schopnost učicího algoritmu správně klasifikovat zpracovávaná data. V našem případě se jedná o *autoenkodér*: druh neuronové sítě vycházející z nesupervizovaného učení. Ten používá metody redukce dimenzionality k projekci samotného vstupu. Využití více vrstev v autoenkodérech má hned několik významů: hloubka může exponenciálně snížit výpočetní náklady na reprezentaci některých funkcí či redukovat množství tréninkových dat potřebných k extrakci složitějších vzorů.²²

Latentní prostor označuje abstraktní vícerozměrnou plochu obsahující hodnoty funkcí, jež slouží ke kódování smysluplné vnitřní reprezentace pomocí externě pozorovaných událostí, které nemohou být přímo interpretovány.²³ Stejně jako my, lidé, rozumíme širokému spektru témat a událostem, které se k těmto tématům vážou, úkolem latentního prostoru je poskytnout obdobné porozumění počítači prostřednictvím kvantitativní, resp. statistické prostorové reprezentace.²⁴ Motivací naučit se latentní prostor (soubor interních reprezentací) nad pozorovanými daty (soubor událostí) je, že významné rozdíly v pozorovaných vzorech mohou být způsobeny pouze nepatrnými odchylkami v latentním prostoru. Učení latentního prostoru tedy modelu pomáhá lépe pochopit redukovaná pozorovaná data.

Formát vstupních dat

Nejpřímějším vyjádřením digitálních skladeb libovolného formátu je surový zvukový signál, ve kterém osa x představuje čas a osa y amplitudu signálu. Problém nastává v situaci, kdy chceme skladbu pro potřeby strojového učení rozložit na binární kód, tedy sadu jedniček a nul. Přestože subsymbolická reprezentace představuje adekvátní vizualizaci pro rozpoznávání významných událostí a pro odpovídající úpravu zvuku, jinak není příliš bohatá na informace. Proto, stejně jako v případě jazyka, vznikla nutnost zavést vhodnou reprezentaci, jež by se inspirovala lingvisticko-antropologickým vztahem, který můžeme pozorovat mezi řečí a písmem.

Moderní notace

Hudební partituru lze považovat za sadu symbolů, které jsou uspořádány v časové posloupnosti. Ty představují hudební entity reprezentující noty, trvání tónů či dynamické

22 RUSSEL a NORVIG, op. cit., 2016.

23 TIU, Ekin. Understanding Latent Space in Machine Learning. In: *Medium* [online]. 2020. Dostupné z: <https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d>.

24 TIU, op. cit., 2020.

změny.²⁵ Hlavní výhodou této metody je definovaný formalismus reprezentace a implementace: záznam lze pozorovat, ověřovat jeho pravidla, studovat, upravovat či rozšiřovat samotné dílo. Nejedná se o nečitelnou černou skříňku, ale spíše o formální jazyk, kterému můžeme porozumět nejen my, ale i stroje. Tradiční západní hudební notace se pokouší reprezentovat hudbu pomocí symbolického jazyka, přičemž plynulí čtenáři notového zápisu jej mohou použít k opětovné reprodukci. Taková notace se však ve skutečnosti nesnaží poskytnout vizuální analogii hudby. Systémy notového zápisu jsou obvykle určeny k tomu, aby poskytly umělci symbolické znázornění některých aspektů struktury hudby či nabídly instruktážní tabulku, jak pomocí nástroje produkovat požadovanou hudbu. Znázornění skladeb pomocí symbolů taktéž významně usnadňuje jejich zpracování pomocí strojového učení; hledá totiž podstatu písní v pouhém kompozičním procesu, zatímco zanedbává definování specifik nástrojů a přispění hudebníků.

ABC

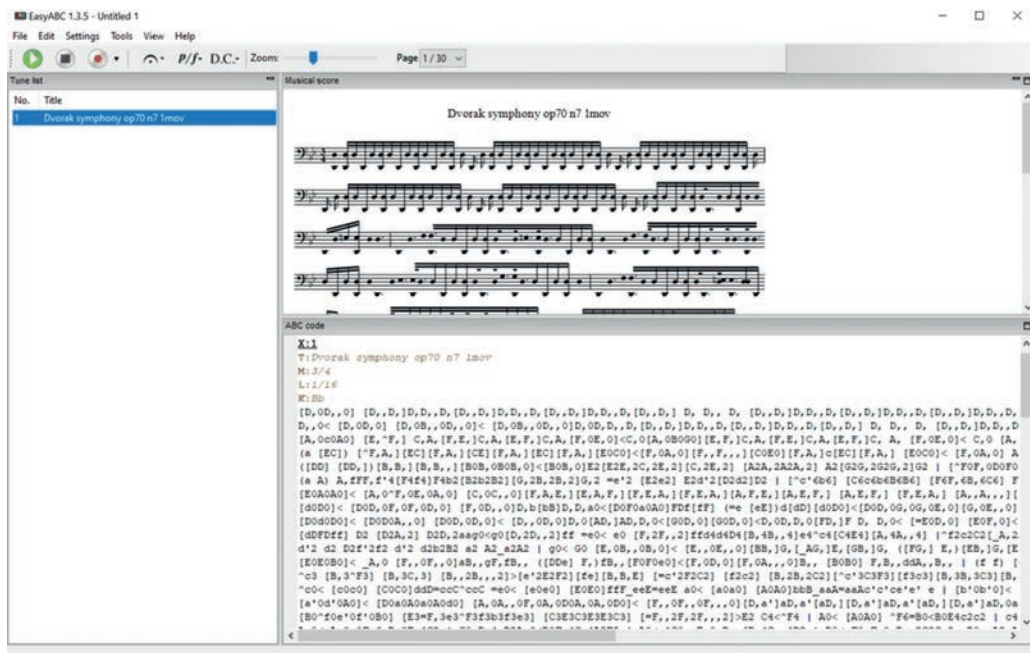
Aby ovšem byla data dále zpracovatelná pomocí strojového učení, je nutné přiřadit jednotlivým entitám jednoduché symboly, které nazýváme tokeny. Tyto tokeny, stejně jako v případě klasické notace, mohou značit noty, trvání či výšky tónů, čímž dochází k extrémní kompresi znalostí: namísto spojitého signálu, který sdružuje soubor velikosti desítek MB získáváme symbolickou extrakci o velikosti několika stovek bitů.²⁶ Vhodným příkladem redukováného datového záznamu je ABC notace, jež našla využití nejen v rámci archivace tradiční irské hudby, kde v 80. letech zaznamenala své hlavní využití. Formát ABC je založen na ASCII znacích, přičemž ke zpracování kódu lze použít libovolný textový editor.²⁷ Právě tato vlastnost je důležitá pro usnadnění aplikace metod strojového učení, kdy se každý jednotlivý token stává nositelem významné hudební informace. I zde lze pozorovat blízkou podobnost se zpracováním jazykových korpusů, snad jen s tím rozdílem, že v rámci lingvistické extrakce pracujeme se shlukem symbolů, jež pouze po splnutí do vyšších entit vykazují vlastní smysl, ať již v podobě slov či vět.

Řádky v první části ABC zápisu začínají písmenem následovaným dvojtečkou. Tyto informace označují různé aspekty informací: index (X:), název (T:), takt (M:), výchozí délku noty (L:), druh melodie (R:) a tóninu (K:). Řádky následující po označení klíče slouží k zápisu samotné melodické složky. Na rozdíl od souborů MusicXML, které jsou určeny pro výměnu hudby mezi aplikacemi pro editaci skóre, formát ABC je navržen tak, aby mohl být jednoduše editován koncovými uživateli.

25 KELLY, Thomas Forrest. *Capturing Music: The Story of Notation*. W. W. Norton & Company; Har/Com edition, 2014, 256 s. ISBN 978-0393064964.

26 CASEY, M. A. et al. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*. 2008, roč. 96, č. 4, s. 668–696. ISSN 1558-2256. DOI:10.1109/JPROC.2008.916370

27 WALSHAW, op. cit., 2014.



Obr. 3 Uživatelské prostředí aplikace EasyABC pro převod MIDI dat do symbolické reprezentace formátu ABC.

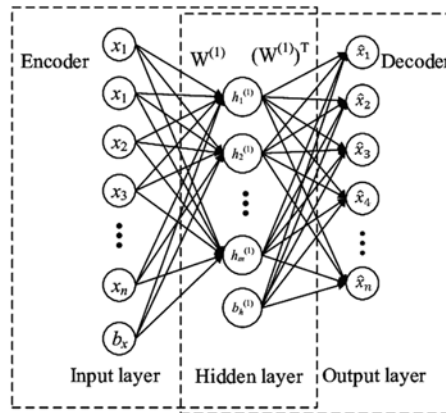
Užitý model neuronové sítě

Vzhledem k povaze sesbíraných dat se přikláníme k metodám nesupervizovaného učení,²⁸ v průběhu něhož dochází k extrakci příznaků ze vstupního datasetu. Autoenkodér je specifickým typem neuronových sítí, jehož hlavní motivací je naučit se aproximaci funkce identity tak, aby se výstup co nejvíce podobal vstupní funkci. Zatímco napodobení funkce identity se zdá být veskrze triviálním problémem, umístěním dodatečného omezení v síti (např. sníženým počtem neuronů ve skryté vrstvě)²⁹ můžeme nalézat originální struktury. Komprimací vstupních dat do nižší dimenze a následnou rekonstrukci je síť schopna smysluplně replikovat data podobna těm, na nichž byla natrénována. Díky schopnosti generovat obsah pomocí vzorkování náhodně spouštěcí veličiny můžeme hovořit o tzv. variačním autoenkodéru.

V našem konkrétním příkladu hovoříme o 340 786 vstupních tokenech reprezentujících volně dostupný dataset skladeb Antonína Dvořáka, přičemž skrytá vrstva s LSTM buňkami obsahuje pouhých 512 neuronů s ReLU aktivační funkcí. Protože existuje pouze omezené množství buněk v bottleneck vrstvě, je síť nucena naučit se komprimovanou

28 Taktéž známo jako učení bez učitele.

29 Tzv. *bottleneck layer* neboli vrstva s úzkým hrdlem. SAINATH, T. N., B. KINGSBURY a B. RAMABHADRAN. *Auto-encoder bottleneck features using deep belief networks*. 2012. DOI: 10.1109/ICASSP.2012.6288833.



Obr. 4 Vizualizace nekompletního autoenkodéru pro zpracování sekvenčních dat.

reprezentaci vstupu. Taková struktura autoenkodéru se nazývá nekompletní.³⁰ Vstupní data nejdříve prochází enkodérem, který vytváří reprezentační kód. Dekodér, který nese obdobnou strukturu, poté produkuje výstup pouze pomocí tohoto redukováného kódu.³¹ Cílem je získat výstup, jenž je pouhou imitací vstupních dat; v případě příliš široké bottleneck vrstvy by naše síť jen slepě kopírovala vstupní sekvence.

K samotné konstrukci sítě bylo využito především rozhraní Keras, knihovna programovacího jazyka Python, jež se nejčastěji využívá pro implementaci umělých neuronových sítí.³² Keras, funkční nadstavba softwarové knihovny TensorFlow, obsahuje celou řadu často používaných stavebních bloků neuronových sítí: vrstvy, callback funkce, aktivační dynamiky, optimalizátory a jiné nástroje, které usnadňují práci s obrazovými a textovými daty, aby došlo ke zjednodušení samotného psaní kódu hluboké neuronové sítě.³³

Optimalizace chybové funkce

Autoenkodér se učí napodobovat svůj výstup pomocí latentního prostoru vstupní trénovací množiny. Imitační schopnost je úzce spojena s dalším důležitým prvkem, chybovou funkcí. Chybovou funkci můžeme snadno definovat jako rozdíl mezi skutečnou a požado-

30 Tzv. *undercomplete autoencoder*. WALLACE a HARIHARAN, op. cit., 2020.

31 DERTAT, Arden. Applied Deep Learning - Part 3: Autoencoders. In: *Medium* [online]. 8. 10. 2017. Dostupné z: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>.

32 GULLI, Antonio a Sujit PAL. *Deep Learning with Keras*. Packt Publishing Ltd, 2017. ISBN 978-1-78712-903-0.

33 Building Autoencoders in Keras. In: *The Keras Blog*. Dostupné z: <https://blog.keras.io/building-autoencoders-in-keras.html>.

vanou odezvou našeho modelu.³⁴ To, co běžně popisujeme jako učení neuronové sítě, tak ve své podstatě není nic jiného než matematická úloha optimalizace, která klade za cíl nalézt vhodné váhy sítě, jež poskytují přesné nebo alespoň dostatečně odpovídající předpovědi. Derivací vícedimenzionální vstupní funkce³⁵ získáváme gradient ztrátové funkce ∇ , který budeme v rámci trénování sítě optimalizovat: hledat globální minimum. Gradient této funkce je složen z hodnot pravděpodobnostního rozdělení, k jehož výpočtu a následné normalizaci dochází během aktivace softmax funkce ve výstupní vrstvě neuronové sítě. Cílem optimalizačního algoritmu je měnit váhy mezi jednotlivými neurony tak, aby byla chyba při následující odezvě menší. To znamená, že optimalizační algoritmus konverguje směrem po gradientu chyby a pomocí metody zpětné propagace chyby upravuje hodnoty synaptických spojení mezi neurony.³⁶

Tab. 1 Detailní popis konstruované neuronové sítě.

Použité knihovny	Keras, TensorFlow, NumPy, sys
Délka vstupní sekvence	340 786
Délka výstupní sekvence	400 (fixed-size vector)
Počet neuronů ve skryté vrstvě	512 LSTM
Hodnota dropout	0,2
Délka zpracované sekvence	200
Velikost dávky	128
Aktivační funkce pro neurony ve skryté vrstvě	ReLU
Aktivační funkce pro neurony ve výstupní vrstvě	Softmax
Chybová funkce	Kategorická křížová entropie
Optimizér	Adam
Hodnota chybové funkce po ukončení trénování	0,3822

Dvorak Render 1



34 VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 2. vyd. Ostrava: Vysoká škola báňská – Technická univerzita, 2000. ISBN 80-7078-949-2.

35 Tato vstupní funkci můžeme definovat jako tenzor.

36 VOLNÁ, Eva. *Umělá inteligence: rozpoznávání vzorů v dynamických datech*. Praha: BEN – technická literatura, 2014. ISBN 9788073005146.

Dvorak Render 2



Dvorak Render 3



Obr. 5 Příklady generovaných partitur

Autoenkodér představuje specifickou architekturu neuronových sítí, která nachází své využití především v generativním modelování. Ačkoliv původní myšlenka využití autoenkodéru odkazovala na samotnou redukci dimenzionality, v průběhu posledních několika let došlo k mnohočetné augmentaci na odlišné formy úloh. Motivací našeho modelu bylo naučit se funkci identity v latentním prostoru pomocí zmenšené skryté vrstvy, jež posloužila jako hrdlo. Díky tomu disponuje nekompletní autoenkodér schopností generalizovat vstupní data a převádět je do prostoru s nižší dimenzionalitou. Hodnota chyby pak představuje, zjednodušeně řečeno, rozdíl mezi vstupním a výstupním latentním prostorem. Kompresi dat se vždy váže ke konkrétnímu datasetu, v našem případě ke sbírce skladeb Antonína Dvořáka v symbolickém formátu ABC. Po dokončení učení dochází k odpojení enkodéru, pro generování obsahu se využívají pouze ty příznaky, jež byly v průběhu trénování sítě extrahovány.

Abychom však mohli ve zhodnocení pokračovat, musíme nejdříve akceptovat fakt, že současné modely na bázi neuronových sítí nerozumějí tomu, co pomocí matematických operací vytváří. Tyto modely vidí vstupní transkripci nikoli jako hudební dílo, nýbrž jako set symbolů, tokenů či proměnných. Pokud bychom chtěli uvažovat nad modelem, který plně pochopí podstatu hudební kompozice, museli bychom se ohlédnout směrem k vývoji kognitivních architektur. I proto je nutné zmínit, že všechny aktuální aplikace neuronových sítí spadají pod tzv. *úzkou umělou inteligenci* čili jsou schopny řešit pouze ty úlohy, pro něž byly stvořeny. Určitou slabou stránkou většiny projektů využívajících umělou inteligenci pro generování hudební transkripce je absence dříve očekávané autonomie. Ačkoliv jsou tyto modely schopny vytvářet nevšední melodie či rytmičky, jedná se stále o pouhé generování hudebního přepisu, jenž počítá s širokým lidským vstupem. Tato skutečnost samozřejmě nečiní tyto modely méně zajímavými, je nutné ovšem brát v potaz zapojení umělců nejen v rámci editačních prací, ale i samotného zhudebnění vzniklé transkripce.

Bibliography

- ALPERN, Adam. *Techniques for Algorithmic Composition of Music* [online]. Hampshire College, 1995. DOI:10.1.1.23.9364.
- Building Autoencoders in Keras. In: *The Keras Blog*. Dostupné z: <https://blog.keras.io/building-autoencoders-in-keras.html>.
- CASEY, M. A. et al. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*. 2008, roč. 96, č. 4, s. 668–696. ISSN 1558-2256. DOI:10.1109/JPROC.2008.916370.
- DAVIS, Martin. *Engines of Logic: Mathematicians and the Origin of the Computer*. USA: W. W. Norton & Co., Inc., 2001. ISBN 978-0-393-32229-3.
- DETTAT, Arden. Applied Deep Learning - Part 3: Autoencoders. In: *Medium* [online]. 8. 10. 2017. Dostupné z: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>.
- ECK, Douglas a Juergen SCHMIDHUBER. *A First Look at Music Composition using LSTM Recurrent Neural Networks* [online]. 2002. Manno, Switzerland: Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale. DOI:10.5555/870511.
- FERNANDEZ, Jose David a Francisco VICO. AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*. 2013, roč. 48, s. 513–582. ISSN 1076-9757. DOI: 10.1613/jair.3908.
- FIEBRINK, Rebecca a Baptiste CARAMIAUX. The Machine Learning Algorithm as Creative Musical Tool. *Handbook on Algorithmic Music* [online]. Oxford: Oxford University Press, 2016. Dostupné z: https://www.researchgate.net/publication/309631691_The_Machine_Learning_Algorithm_as_Creative_Musical_Tool
- GULLI, Antonio a Sujit PAL. *Deep Learning with Keras*. Packt Publishing Ltd, 2017. ISBN 978-1-78712-903-0.
- HE, Kaiming et al. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]* [online]. 2015. Dostupné z: <http://arxiv.org/abs/1512.03385>.
- HILLER, Lejaren Arthur a Leonard M. ISAACSON. *Experimental Music; Composition with an Electronic Computer*. Westport, CT, USA: Greenwood Publishing Group, 1979, 197 s. ISBN 978-0-313-22158-3.
- HOCHREITER, Sepp a Jürgen SCHMIDHUBER. Long Short-Term Memory. *Neural Computation* [online]. 1997, 9(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735. ISSN 0899-7667. Dostupné z: <http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735>.
- KELLY, Thomas Forrest. *Capturing Music: The Story of Notation*. W. W. Norton & Company; Har/Com edition, 2014, 256 s. ISBN 978-0393064964.
- LOY, Gareth. *Musimathics, Volume 1: The Mathematical Foundations of Music*. Reprint edition. vyd. Cambridge, Mass.: The MIT Press, 2011. ISBN 978-0-262-51655-6.
- REJZEK, Jiří. *Český etymologický slovník*. Voznice: Leda, 2001. ISBN 80-85927-85-3. Heslo „algorithmus“.
- RUSSELL, Stuart J. a Peter NORVIG. *Artificial intelligence: a modern approach*. Third edition. Boston: Pearson, 2016. Prentice Hall series in artificial intelligence. ISBN 9781292153964.
- SAINATH, T. N., B. KINGSBURY a B. RAMABHADRAN. *Auto-encoder bottleneck features using deep*

- belief networks*. 2012. DOI: 10.1109/ICASSP.2012.6288833.
- STURM, Bob L. On folk-rnn cheating and music generation [online]. 2017. Dostupné z: <https://highnoongmt.wordpress.com/2017/06/15/on-folk-rnn-cheating-and-music-generation/>.
- TIU, Ekin. Understanding Latent Space in Machine Learning. In: *Medium* [online]. 2020. Dostupné z: <https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d>.
- TODD, Peter M. A Connectionist Approach to Algorithmic Composition. *Computer Music Journal* [online]. 1989, 13(4). DOI: 10.2307/3679551. ISSN 01489267. Dostupné z: <https://www.jstor.org/stable/3679551?origin=crossref>.
- VOLNÁ, Eva. *Umělá inteligence: rozpoznávání vzorů v dynamických datech*. Praha: BEN – technická literatura, 2014. ISBN 9788073005146.
- VONDRÁK, Ivo. *Umělá inteligence a neuronové sítě*. 2. vyd. Ostrava: Vysoká škola báňská – Technická univerzita, 2000. ISBN 80-7078-949-2.
- WALSHAW, Chris. A Statistical Analysis of the ABC Music Notation Corpus: Exploring Duplication. *Conference: Folk Music Analysis*. Istanbul, 2014. DOI:10.13140/2.1.4340.0961.



Toto dílo lze užit v souladu s licenčními podmínkami Creative Commons BY-SA 4.0 International (<https://creativecommons.org/licenses/by-sa/4.0/legalcode>). Uvedené se nevztahuje na díla či prvky (např. obrazovou či fotografickou dokumentaci), které jsou v díle užity na základě smluvní licence nebo výjimky či omezení příslušných práv.

