Deep Learning pri tvorbe hudby

Tomáš Juščík[1] a Richard Kačur[2]

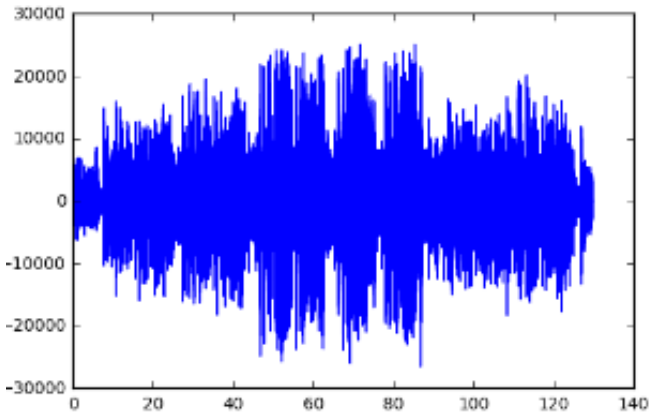
***Abstrakt*— Popri tradičných úlohach hlbokých neurónových sieti, ako je predikcia a klasifikácia, sa začína do popredia dostávať kompozícia hudby. Motiváciou pri používaní hlbkových neurónových sietí je, aby sa automaticky naučili rozpoznávať hudobné štýly a následne aby generovali hudbu pre rozpoznaný hudobný štýl. V tejto práci analyzujeme hlbkové neurónové siete, ktoré sa využívajú pri kompozícii hudby a pri identifikácii hudobného žanru hudby.**

I. ÚVOD

Hudba nás obklopuje v našom každodennom živote. Môžme ju počuť, ked sme v aute, pri nakupovaní, cestovaní a na mnoho iných miestach. Hlavné účely hudby spočívajú v ovplyvnení nálady či už pozitívne alebo negatívne, taktiež sa hudba využíva na zábavu, sústredenie a mnoho iného. Je to taktiež spôsob, ako vyjadriť svoje pocity, nálady, túžby, ktoré často nevieme opísať slovami. Hudobné štýly sa menili s tým, ako ľudstvo napredovalo. Tieto štýly odzrkadľovali dané obdobia, počas ktorých vznikli. Veľký rozkvet hudba zaznamenala v obdobiach baroka a klasicizmu, kedy excelovali skladatelia ako Johann Sebastian Bach a Wolfgang Amadeus Mozart.

V súčasnosti sa v hudbe začínajú aplikovať hlboké umelé neurónové siete. Využívajú sa na kompozíciu, analyzovanie a aj odporúčanie hudby.

Cieľom pri kompozícii hudby je vytvoriť hudbu, ktorá bude príjemná na počúvanie. Jedným zo spôsobov kompozície hudby je pomocou zvukových vĺn. Na Obr. 1 môžme vidieť ukážku vizualizácie zvukovej vlny. Ak sú prechody medzi zvukovými vlnami plynulé zvuk vtedy pôsobí prívetivejšie.



Obr. 1. Vizualizácia zvukovej vlny [1]

[1]T. Juščík, Fakulta Elektrotechniky a Informatiky, Technická Univerzita Košice, Slovensko

[2]R. Kačur, Fakulta Elektrotechniky a Informatiky, Technická Univerzita Košice, Slovensko,

V hudbe sa využíva viacero typov hlbokých neurónových sietí:

- Convolutional Deep Neural Networks (CDNN)
- Long Short-Term Memory (LSTM)
- Continuous recurrent neural networks with generative adversarial networks (C-RNN-GAN)

II. DEEP LEARNING

Hlboká neurónová sieť je trieda strojového učenia, ktorá obsahuje viac ako jednu skrytú vrstvu neurónov medzi vstupnou a výstupnou vrstvou. Každý level hlbokej neurónovej siete sa učí transformovať vstupné dáta do viac abstraktného zobrazenia. [2]

III. CONVOLUTIONAL DEEP NEURAL NETWORKS (CDNN)

CDNN je druh hlbokej neurónovej siete, ktorá sa využíva hlavne na analýzu obrázkov, ale aj na klasifikáciu hudby do hudobných žanrov.

Následujúca sekcia opisuje danú neurónovú sieť z hľadiska identifikácie hudobného žánru zo zvukovej vzorky.

A. Architektúra

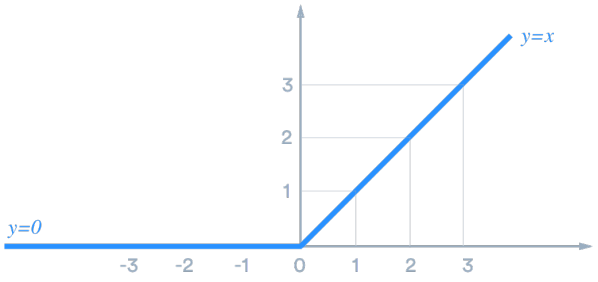
Sieť znázornená na Obr. 4 pozostáva z týchto vrstiev:

- **Convolutional layer + ReLU** - základný stavebný kameň konvolučnej siete. Táto vrstva vykonáva konvolúciu pomocou konvolučnej masky. Výstup z convolutional layer vstupuje do ReLU vrstvy. ReLU (Obr. 3) je aktivačná funkcia, ktorá je definovaná ako:

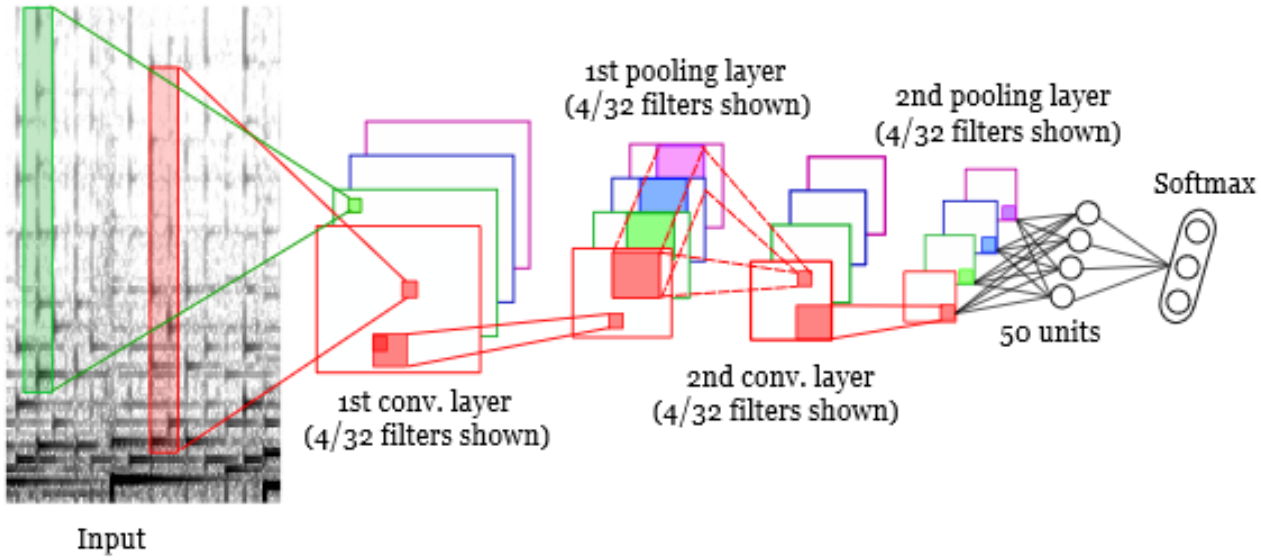
$$f(x) = x^+ = \max(0, x)$$

Obr. 2. Rectified Linear Unit [7]

Aplikovanie ReLU mení vstup podľa funkcie (Obr. 2), tak aby všetky negatívne prvky boli nahradené nulou a všetky pozitívne prvky ostali nezmenené.

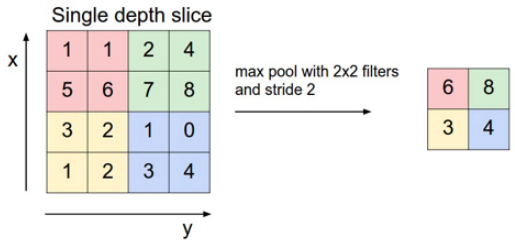


Obr. 3. Graf funkcie ReLU [3]



Obr. 4. Convolutional Deep Neural Network (CDNN)[5]

- **Pooling layer** - funkciou tejto vrstvy (Obr. 5) je postupné znižovanie množstva parametrov, čo sa vykonáva pomocou operácie MAX s filtrami veľkosti 2x2.



Obr. 5. Výsledok pooling vrstvy s jadrom 2x2 [4]

- **Fully connected layer** - táto vrstva obsahuje 50 neurónov. Posledná pooling layer je plne prepojená s toutu vrstvou.
- **Softmax** - tento prístup je jeden z najbežnejších prístupov pre klasifikačnú úlohu do viac ako dvoch tried. Táto funkcia predstavuje rozdelenie pravdepodobnosti na diskrétny výstup s viacerými možnosťami (Obr. 6). Softmax zabezpečuje to, že suma pravdepodobností pre každú hodnotu sa rovná 1.[8]

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{i=1}^{n} e^{z_i}}$$

Obr. 6. Softmax [8]

  - σ - označenie Softmax funkcie
  - **n** - počet hodnôt vo vektore
  - **z** - vstupný vektor
  - **i** - hodnoty od 1, ..., n

Architektúra CDNN (Obr. 4) najprv aplikuje úzke vertikálne filtre (zelený a červený obdĺžnik) na vstupný sonogram. Dlhý obdĺžnikový tvar je preferovaný oproti štvorcom, keďže mnohé zvuky vykazujú silné harmonické štruktúry, ktoré pokrývajú veľkú časť počuteľného spektra.[5]

Výstup z filtrov sa posiela do 1st convolutional layer. Na túto vrstvu navezuje 1st pooling layer, za ktorým navezuje druhý pár convolutional a pooling vrstvy. Výsledok poslednej pooling layer je plne spojeny s finálnou skrytou vrstvou, ktorá obsahuje 50 neurónov. Na túto vrstvu navezuje výstupová jednotka Softmax. Táto jednotka rozdelí podľa pravdepodobnosti do akého žanru daná vzorka patrí.[5]

Vstupný spektogram dostáva 100 časových výsekov každých 23 ms, čo znamená, že posledná vrstva sumarizuje informácie za 2.3 sekundy.[5]

B. Hyperparametre

CDNN využíva viacero hyperparametrov:

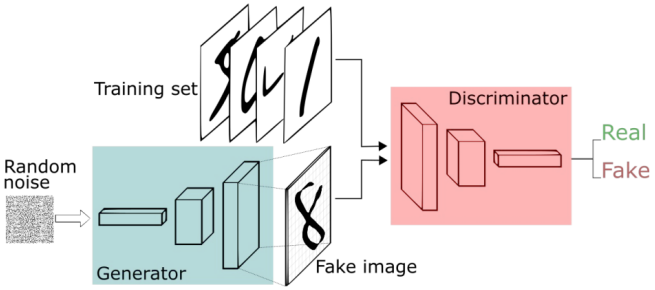
- **Počet vrstiev** - 2 konvolučné a 2 pooling vrstvy, 1 skrytá vrstva
- **Počet filtrov** - konvolučné vrstvy obsahujú 32 filtrov. Pooling vrstvy obsahuju tiež 32 filtrov.
- **Tvar filtrov** - filtre použité v konvolúcii majú rozmery 8 x 8. Pooling filtre majú tvar 4 x 4.
- **Krok** - posun pri pooling filtroch je 2 (Znázornené na Obr. 5).
- **Aktivačné funkcie** - ReLU

IV. Continuous recurrent neural networks with generative adversarial networks (C-RNN-GAN)

A. Definícia

Generative adversarial networks (GANs) patria do triedy hlbokých neurónových sietí, ktorých cieľom je generovať realistické data.[9]

Tieto neuronové siete fungujú na princípe natrénovania dvoch neurónových modelov s protichodnými cieľmi. Jeden model zastáva úlohu generátora (G) a druhý model diskriminátora (D). Tieto modely sa snažia vylepšiť jeden druhého. Generátor sa pokúša vytvoriť vzorky, ktoré vyzerajú reálne a diskriminátor sa snaží rozlišovať medzi generovanými vzorkami a reálnymi dátami(Obr. 7).[6]



Obr. 7. GANs [9]

Recurrent neural networks (RNN) sa často využívajú na modelovanie sekvencií údajov. Tieto modely sú zvyčajne natrénované použitím kritéria maximálnej pravdepodobnosti.[6]

Model RNNs sa taktiež využíva na generovanie hudby. Príkladom môžu byť:

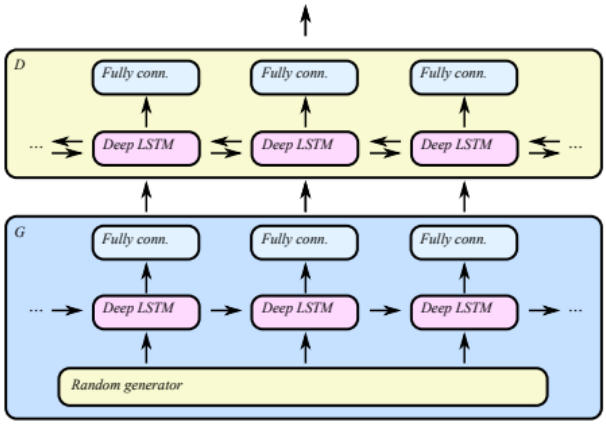
- **Eck and Schmidhuber, r. 2002** - modelovanie Blues(vokálno-inštrumentálna forma hudby), pri ktorom sa využilo 25 hodnôt tónu
- **Nicolas Boulanger-Lewandowski, r.2012** - model v ktorom skombinoval RNN s Boltzmannovím strojom, ktorý reprezentoval 88 rôznych tónov
- **Yu et al, r. 2016** - natrénoval RNN s nepriateľským tréningom, v ktorom sa aplivali policy gradient metódy

Používanie tohto frameworku umožnilo trénovať hlboké generatívne modely, ktoré sú schopné produkovať vysoko realistické vzorky dát.[6]

B. Architektúra

Navrhovaný model je rekurentná neurónová sieť s protichodným tréningom. Protivníci sú dve rôzne hlboké rekurentné neurónové modely, ktorými sú generátor (G) a diskriminátor (D). Generátor je trénovaný na generovanie dát, ktoré sú nerozoznateľné od reálnych dát, pričom diskriminátor je trénovaný na rozpoznanie generovaných dát.[6]

Na Obr. 9 je znázornená schéma navrhnutého modelu pre metódu C-RNN-GAN, v ktorej generátor je reprezentovaný modrým obldžníkom a diskriminátor žltým obldžníkom. Oba modely obsahujú hlboký model rekurentnej siete LSTM.[6]



Obr. 8. C-RNN-GAN : model[6]

Definície pre loss funkciu generátora a loss funkciu diskriminátora sú nasledovné:

$$L_G = \frac{1}{m}\sum_{i=1}^{m} \log(1 - D(G(\boldsymbol{z}^{(i)})))$$

$$L_D = \frac{1}{m}\sum_{i=1}^{m} \left[-\log D(\boldsymbol{x}^{(i)}) - (\log(1 - D(G(\boldsymbol{z}^{(i)}))))\right]$$

Obr. 9. LD, LG loss funkcie[6]

- $\mathbf{z}^{(i)}$ - sekvencia jednotlivých náhodných vektorov v $[0,1]^k$
- $\mathbf{x}^{(i)}$ - sekvencia trénovacích dát
- **k** - rozmer dát v náhodnej sekvencii.

Vstupom do každej bunky v generátore je náhodný vektor, spojený so výstupom z predchádzajúcej bunky. Nabalovanie výstupu z predchádzajúcej bunky je bežnou praxou pri výcviku RNN ako aj pri hudobnej kompozícii. Diskriminátor sa skladá z obojsmernej rekurentnej siete, ktorá umožňuje zbierať súvislosti v oboch smeroch. V bežnej praxi pri kompozícii hudby je rekurentná siet prezentovaná modelom Long short-term memory (LSTM)[6].

C. Reprezentácia hudby

Symbolická reprezentácia skladby zahŕňa akýkoľvek druh reprezentácie s explicitným kódovaním noty v skladbe.[10]

MIDI správy kódujú informácie pre každú sekvenciu tónu, ako je napríklad počiatok noty, posun noty a intenzita(vyjradrená ako ”rýchlosť” v terminológii Musical Instrument Digital Interface (MIDI)).[10]

Číslo tónu, ktoré je reprezentuje MIDI je celé číslo v rozsahu 0 až 127, ktoré kóduje výsku tónu noty. A čo

je najdôležitejšie, tak C4(stredné C) má MIDI poznámku číslo 60 a R4 má MIDI poznámku číslo 69. Rýchlosť tónu je taktiež celé číslo medzi 0 až 127, ktoré riadi intenzitu zvuku.[10]

MIDI kanál je celé číslo medzi 0 a 15, ktorý vyzve syntetizátor, aby použil špecifický inštrument.[10]

MIDI rozdeľuje štvrťovú notu na hodinové impulzy a tiky. Napríklad, ak je počet impulzov na štvrťovú notu definovaný ako 120, potom by 60 tikov predstavovalo dĺžku ôsmej noty.[10]

MIDI môže tiež kódotvať tempo z hľadiska "tepov za minutu" (BPM), čo umožňuje získať absolutné informácie o časovaní.[10]

D. Modelovanie hudby

Táto metóda je inšpirovaná MIDI formátom, ktorý sa využíva pre kuminakčné signály medzi digitálnymi hudobnými nástrojmi[6].

Signál je reprezentovaný štyrmi hodnotenými skalármi v každom dátovom bode: dĺžka tónu, frekvencia, intenzita a čas.

- **Dĺžka tónu** - ako dlho trvál daný tón
- **frekvencia** - stúpanie
- **intenzita** - hlasitosť alebo amplitúda
- **čas** - čas, ktorý prešiel od predošlého tónu

Modelovanie dát týmto spôsobom umožňuje sieti reprezentovať polyfónne akordy s nulovou hodnotou medzi dvoma tónmi[6].

E. Experiment (Olof Mogren)

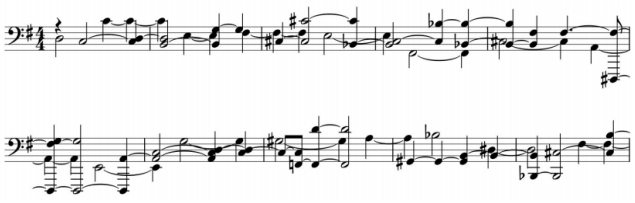
Generátor a diskriminátor obsahuje LSTM sieť, kde každá bunka obsahuje 350 skrytých vrstiev. Dataset tvorí klasická hudba zozbieranú vo formáte MIDI prevedenú do žiadaného formátu[6].

Dadaset obsahuje 3697 MIDI súborov od 160 rôznych skladateľov klasickej hudby[6].

Pri trénovaní bol použitý stochastický gradient Mini-batch a taktiež sa použila L-regulácia na váhach generátora a diskriminátora. Generátor bol predtrénovaný na 6 epochách so štvorcovou stratovou chzbou na predpovedanie ďalšej udalosti v trenovacej sekvencii[6].

Autor experimentu uvádza niektoré metriky na meranie kvality výsledkov[6]:

- **Polyfónia** - meria, ako často sa súčasne prehrávajú dva tóny
- **Konzistencia** - bola vypočítaná spočítaním zlomkov tónov, ktoré boli súčasťou štandardnej škály konzistentnosti
- **Opakovania** - spočítanie sa opakovaných krátkych subsekvencií, ktoré udávali skóre opakovania vo vzorke
- **Tónové rozpätie** - počet krokov polovičného tónu medzi najnižším a najvyšším tónom vo vzorke



Obr. 10. Výsledok experimentu

Na Obr. 10 je znázornený úsek tónov skladby, ktorá bola výsledkom tohto experimentu, ktorý navrhol Olof Mogren[6].

V. ZÁVER

V tejto práci sme sa venovali analýze jednotlivých metód hlbokých neurónových sietí, ktoré sa využívajú pri identifikácii hudobného žánru a kompozícií hudby.

Jedné z najčastejšie využívaných metódy hlbokých neuronóvých sietí pri tejto problematike sú Convolutional Deep Neural Networks (CDNN) a Conntinuous recurrent neural networks with generative adversarial networks (C-RNN-GAN), ktoré sú v práci popísané ako po všeobecnej tak aj po štrukturálnej stránke.

Na záver môžme zhodnotiť, že oblasť identifikácie a generovania hudby pomocou hlbokých neurónových sieti prešla dlhú cestu a v súčasnosti máme k dispozícií množstvo kvalitných metód na klasifikovanie a generovanie hudby, pri ktorej častokrát nazainteresovaný človek nevie rozlíšiť, či danú skladbu zkomponoval človek alebo umelá inteligencia.

REFERENCES

[1] Kalingeri V., Grandhe S. (2016), *Music Generation with Deep Learning*, University of Massachusetts Amherst.

[2] LeCun Y., Bengio Y. & Hinton G. (2015), *Deep Learning*, Dostupné online: [https://www.nature.com/articles/nature14539]

[3] Liu D (2017), *A Practical Guide to ReLU*, Dostupné online: [https://medium.com/tinymind/a-practical-guide-to-relu-b83ca804f1f7]

[4] *CS231n Convolutional Neural Networks for Visual Recognition*, Dostupné online: [http://cs231n.github.io/convolutional-networks/]

[5] Kereliuk C., Sturm B. L., Larsen J. (2015), *Deep Learning and Music Adversaries*, Dostupné online: [https://ieeexplore.ieee.org/abstract/document/7254179]

[6] Olof Mogren (2016), *C-RNN-GAN: Continuous recurrent neural networks with adversarial training*, Dostupné online: [http://mogren.one/publications/2016/c-rnn-gan/mogren2016crnngan.pdf]

[7] *Rectifier (neural networks)*, Dostupné online: [https://en.wikipedia.org/wiki/Rectifier_(neural_networks)]

[8] Briot J. P., Hadjeres G., Pachet F-D. (2019), *Deep Learning Techniques for Music Generation– A Survey*, Dostupné online: [https://arxiv.org/abs/1709.01620]

[9] *A Beginner's Guide to Generative Adversarial Networks (GANs)*, Dostupné online: [https://skymind.ai/wiki/generative-adversarial-network-gan]

[10] *Symbolic Representations* Dostupné online: [https://musicinformationretrieval.com/symbolic$_r epresentations.html$]