

STRESZCZENIE PRACY INŻYNIERSKIEJ

Temat pracy: Algorytm tworzenia profilu pisma odręcznego oparty o metody sztucznej inteligencji

Cel:

Celem pracy było stworzenie algorytmu opartego o metody sztucznej inteligencji pozwalającego przy pomocy komputera tworzyć pismo naśladowujące odręczne. Synteza ta mogła zostać zrealizowana jako tworzenie pojedynczych liter albo całych wyrazów.

Rozwiązanie:

Rozwiązanie oparto o architekturę generatywnych sieci przeciwstawnych (ang. generative adversarial networks, GAN) [1]. Celem użytej sieci neuronowej było nauczyć się generować obrazy ciągów liter. Do zadania tego sieć potrzebowała wielokrotnie przetworzyć dane reprezentujące wszystkie ciągi liter – po kilka tysięcy egzemplarzy na każdą możliwość.

Tworzenie par liter:

Generowanie całych wyrazów wymagałoby przetworzenia ogromnej ilości danych i byłoby bardzo kosztowne obliczeniowo, być może nawet nie możliwe do zrealizowania bez zastosowania nietrywialnych rozwiązań. Niech świadczy o tym fakt, że pomimo popularności sieci typu GAN, najstarsze znalezione źródło opisujące problem tworzenia całych wyrazów przy jej pomocy powstało w marcu 2019 r.[2] (projekt do streszczonej tu pracy zaczął powstawać w 2018 r.).

Z drugiej strony tworzenie pojedynczych znaków posiadało wadę która odbierała temu rozwiązaniu prawdziwość – brakowało ciągłości pisma, czyli łączenia pomiędzy literami oraz zaburzeń z tym związanych. Dlatego zdecydowano się na tworzenie par liter. W przyszłości mogłyby być one wykorzystane do tworzenia całych wyrazów, albo rozwiązanie to mogłoby być rozszerzone.

Tworzenie i filtrowanie danych:

Samo tworzenie par liter niosło za sobą duży problem – brak potrzebnego zbioru danych. Poradzono sobie z tym używając znalezionej generatora opartego o architekturę sieci neuronowej ze sprzężeniami zwrotnymi (ang. recurrent neural network, RNN) [1]. Przetworzono dane tak, aby były możliwe i optymalne do wykorzystania przez sieć typu GAN. Tworzenie i użycie zbioru niosło za sobą kolejne problemy, takie jak potrzeba odrzucenia błędnie wygenerowanych danych. Nie udało się odfiltrować wszystkich złych egzemplarzy prostymi metodami i konieczne okazało się użycie kolejnego programu również opartego o metody sztucznej inteligencji [4]. Generator oraz program użyty do filtrowania danych nauczyły się swoich zadań na podstawie tego samego zbioru danych [5], co miało pozwolić na lepszą trafność filtrowania.

Ograniczenie ilości par:

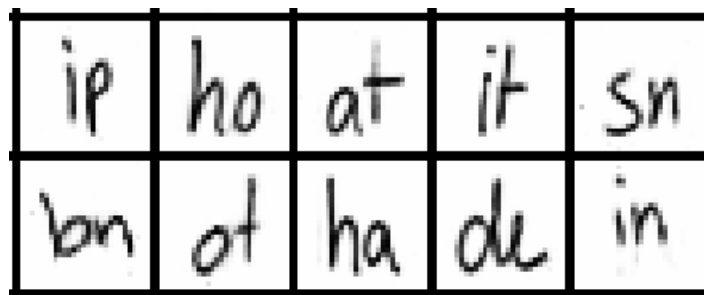
Odrzucona została spora część stworzonych danych. Niektóre z nich były źle wygenerowane od początku, wiele dodatkowo odrzuciły programy filtrujące. Ostatecznie zdecydowano się brać pod uwagę 97 par liter. Pozytywny aspekt tej decyzji był taki, że zmniejszył się zbyt duży czas potrzebny na wygenerowanie całego zbioru danych oraz wytrenowanie sieci.

Uczenie sieci neuronowej:

Stworzono program opierający się o sieć neuronową korzystającą z parametryzacji zaproponowanej tutaj [6]. Służył on generowaniu cyfr i miał za zadanie pełnić rolę pomocniczą i podstawę do wypracowania docelowego modelu sieci neuronowej. Za jego pomocą testowano generowane zbiory danych, a także przeprowadzono jego uczenie z użyciem głównego zbioru. Trenowanie przebiegało zaskakująco pomyślnie – generowane były pary liter coraz bardziej przypominające te pisane przez człowieka.

Wyniki:

Osiągnięte rezultaty były dobre – oceniono, że stworzony program nauczył się syntezować pary liter.



Rysunek 1 Wygenerowane pary liter.

Ocena:

W pracy została umieszczona analiza zawierająca ocenę jakości wygenerowanych egzemplarzy. Dokonano tego wzrokowo, przypisując punkty obrazkom zgodnie z wcześniej zdefiniowanymi zasadami. Nie doszukano się pary nierozpoznawalnej dla człowieka, niemal wszystkie obrazki oceniono jako dopuszczalne w drobnym tekście, a większość wyglądała bardzo dobrze.

Podsumowanie:

Na koniec podsumowano zrealizowane rozwiązanie i przedstawiono możliwe ścieżki rozwoju oraz użycia stworzonego projektu.

Bibliografia:

- [1] Alonso, Eloi & Moyssset, Bastien & Messina, Ronaldo. (2019). *Adversarial Generation of Handwritten Text Images Conditioned on Sequences*, https://www.researchgate.net/publication/331485428_Adversarial_Generation_of_Handwritten_Text_Images_Conditioned_on-Sequences.
- [2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. *Generative Adversarial Networks* [arXiv:1406.2661\[stat.ML\]](https://arxiv.org/abs/1406.2661), <https://arxiv.org/abs/1406.2661>
- [3] Grzegorz Opoka (Grzego), *handwriting-generation*, <https://github.com/Grzego/handwriting-generation>
- [4] Harald Scheidl (githubharald), *SimpleHTR* <https://github.com/githubharald/SimpleHTR>
- [5] U. Marti and H. Bunke. *The IAM-database: An English Sentence Database for Off-line Handwriting Recognition*. Int. Journal on Document Analysis and Recognition, Volume 5, pages 39 - 46, 2002., <http://www.fki.inf.unibe.ch/databases/iam-on-line-handwriting-database/download-the-iam-on-line-handwriting-database>
- [6] Diego Gomez Mosquera, *GANs from Scratch 1*, <https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f>