ISSN 2224-087X. Електроніка та інформаційні технології. 2021. Випуск 16. С. 3–10
Electronics and information technologies. 2021. Issue 16. P. 3–10

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В НАУКОВИХ ДОСЛІДЖЕННЯХ

УДК 004.8, 004.032.26 DOI: https://doi.org/10.30970/eli.16.1

THE SPEED OF LEARNING CONVOLUTIONAL NEURAL NETWORKS ON THE GPU AND CPU TO DETECT SYNTHESIZED SPEECH USING SPECTROGRAMS

L. Demkiv

Ivan Franko National University of Lviv,
50 Drahomanova St., 79005 Lviv, Ukraine
lidia.demkiv@gmail.com

In this work has been investigated the possibility of using convolutional neural networks to detect synthesized speech. The Python programming language, the TensorFlow library in combination with the high-level Keras API and the ASVspoof 2019 audio database in flac format were used to create the software application. The voice signal of synthesized and natural speech was converted into mel-frequency spectrograms. The structure of a convolutional neural network with high indicators of recognition accuracy is proposed. The learning speed of neural networks on GPU and CPU is compared using the CUDA library. The influence of the batch size parameter on the accuracy of the neural network was investigated. The TensorBoard tool was used to monitor and profile the learning process of neural networks.

Keywords: audio deepfake, mel-frequency sound spectrograms, convolutional neural networks, learning speed of neural networks.

Introduction

The development of machine learning algorithms and the synthesis of human speech has given impetus to the field of so-called "deepfake". Deepfake is an artificially created video, sound recording or photo that copies the voice and appearance some other people in order to mislead the viewer. Deepfake algorithms can create such fake media content that humans cannot distinguish them from authentic ones. The number of articles related to deepfakes has increased 20 times between 2018 and 2020, which indicates that the study of algorithms for creating and recognizing deepfakes is an interesting research trend [1].

Deepfakes are usually created and recognized using machine learning algorithms. Rapid evolution of deep-learning (DL) methods, especially Generative Adversarial Networks (GAN), have made it possible to generate deepfakes to disseminate disinformation [2]. In literature, several techniques based on deep learning have been proposed for deepfake detection: convolutional neural network (CNN); recurrent neural network (RNN); long short-term memory (LSTM) [3]. Also, the traces left by the Generative Adversarial Network (GAN) mechanisms during the creation of Deepfakes can be detected by discrete cosine transform (DCT), analyzing special frequencies that represent a unique imprint of different generative architectures [4]. Classical algorithms for classifying natural, synthesized, and vocalized sound (without the use of DL) [5-6] used pitch change templates.

The constantly improving tools for creating deepfake become more sophisticated and that makes the results of using them more dangerous. Therefore, programs for the recognition of deep fakes, which use and combine several methods of recognition of audio-visual speech, recognition of emotions, speech and visual tasks are being developed [7]. The most complicated for recognition are audio deepfakes because unlike video depefake they are very difficult to be recognized by a human ear. Now we can notice the increase rise of popularity in use of voice-controlled systems that identify a user by the voice, but such systems are particularly vulnerable to audio deepfakes. Right now it is a big necessity to develop algorithms and systems that allow you to detect synthesized sound recordings and use them for voice communication over the Internet.

Conversion of audio signal into spectrograms

The data set ASVspoof 2019 includes voice recordings of 107 people who speak English as well as recordings created by 19 different voice synthesizers. ASVspoof, now in its third edition, is a series of community-led challenges, which promote the development of countermeasures to protect automatic speaker verification (ASV) from the threat of spoofing [8].

Sound recordings are stored in flac format, which is a bit inconvenient to use, but allows to reduce the amount of data to download. Audio recordings have a sampling frequency of 16,000 Hz and a bit rate of 16 bits. The data set is divided into 3 parts:

• Data for network development
• Training data (25380 records)
• Data to be checked (71237 records)

The flac audio signal has been converted to .wav, because flac format is not supported by most libraries for audio. To transcode audio to WAV format, use the utility ffmpeg. The ffmpeg utility allows you to transcode audio, video into the most popular formats, and also has high speed due to the use of the C ++ programming language and parallelization of the encoding process. The input method takes the file name and voice authentication. To generate the spectrogram, the program reads the audio signal, its sampling rate and creates a matplotlib canvas of 0.72 inches and 400 dpi without axes. Spectrogram is generated using the melspectrogram method from the librosa library. The process of building a spectrogram takes quite a long time, for example, the generation of spectrograms for a training data set takes 1 hour on a 6-core Intel Core i5 9600k 4.3 GHz processor. As a result, spectrograms of synthesized and natural speech were obtained. Figure (Fig. 1) shows spectrograms for natural and synthesized speech.

Visual analysis and comparison of spectrograms of natural and synthesized speech show the absence of clear visual signs by which it would be possible to clearly say to which type of speech (natural and synthesized) these spectrograms belong.

Modeling, training and testing of convolutional neural networks

The Tensorflow library in combination with the Keras API was chosen for the development and training of convolutional neural networks. Functional API for Keras is used to design the structure of the neural network and implements models as a directed acyclic graph [9]. This representation of the network allows more flexibly describe the structure of the neural network, reuse network layers, save and restore the network model.

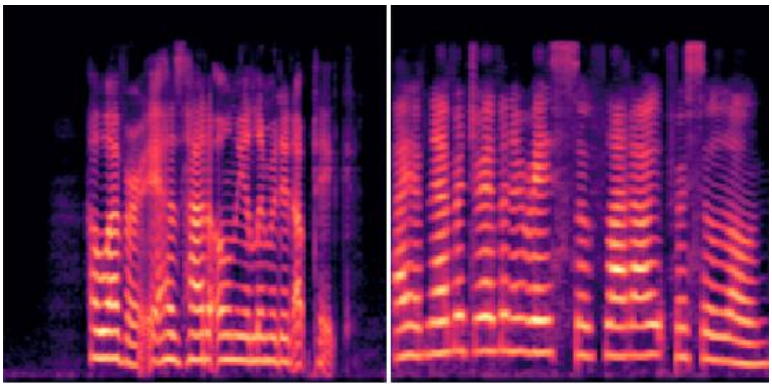

Fig. 1. Spectrograms for natural and synthesized speech.

A convolutional neural network is constructed from several layers of convolution, layers of maximum and average subsample. A single-neuron layer with a sigmoidal activation function stands at the output of the neural network and answers the question of whether the voice on the recording is synthesized. The neural network model that showed the best results is shown in (Fig. 2)

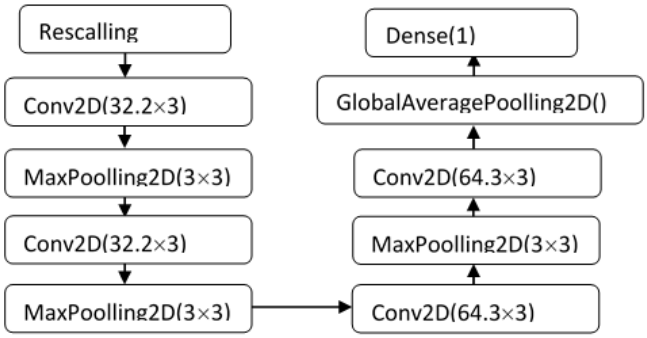

Fig.2. Neural network model.

To improve the quality of training, we will allocate 20% of training data for validation in the training process. The validation process during training will improve accuracy and identify possible network retraining. The *image_dataset_from_directory* method loads images from a directory. All images must be in a subdirectory of their class. Two classes of images with natural and synthesized voice correspond to identifiers 0 and 1. The *label_mode* parameter is responsible for the method of assigning the corresponding identifier to the classes. The *batch_size* parameter determines how many images are processed in one learning step. Also, this parameter affects the accuracy of the network and the speed of the learning process. Network verification data downloaded separately. The accuracy of the network after training is checked for data unknown to the network.

The Adam optimizer with a learning speed of 0.001 was used to train the model. Since we need to distinguish between two classes of data, the loss function will choose binary cross-entropy (logarithm of losses). The neural network has been studied for 20 epochs. As a result of training, the following values were achieved: accuracy (training) 0.99 and validation accuracy 0.98 (Fig. 3); the loss function reached 0.02 in the training sample and 0.03 in the validation sample (Fig. 4).

After learning the network, network was tested using data that the network did not see during the training. As a result of testing, accuracy of 0.9147 was obtained.

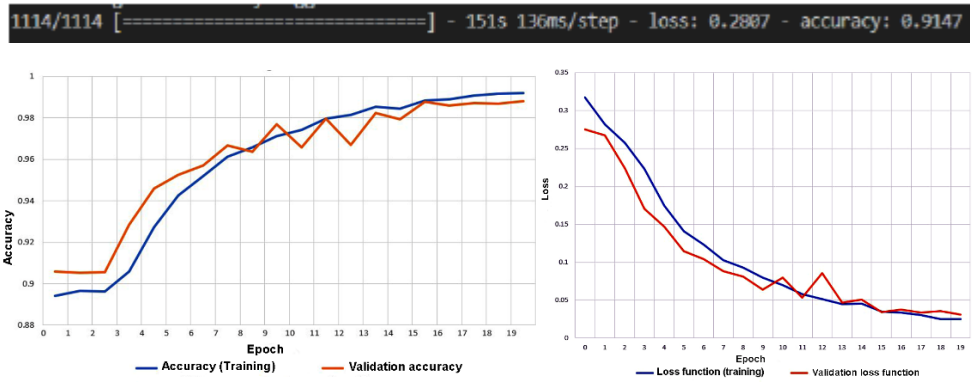

Fig.3. Accuracy and validation accuracy during training

Fig.4. Loss function and validation loss function during training

Using GPU to Speed-up Deep Learning

One of the benefits of working with TensorFlow is the ability to use all of your computer's resources to learn about the neural network. It is known that to increase the speed of the learning process of neural networks to recognize deepfake can be using graphics processors, To get started with the GPU, download the appropriate version of tensorflow-gpu, as well as libraries CUDA and cuDNN [10]. Developed by Nvidia for its GPUs, CUDA technology has made it possible to perform parallel computations using GPU cores. OpenCL technology is not supported by the TensorFlow library, so it could not be used for calculations. To determine the impact of using a graphics processor on the learning time of the neural network, an experiment was conducted in which we will train the neural network for 5 epochs with the parameter batch_size = 32 and 64 on the CPU and GPU. The experiment was performed on an NVIDIA GTX 1060 6GB graphics processor and an Intel Core i5 9600k CPU. The results of the experiment are in the table (Table 1).

Table 1. Comparison of learning speed of neural networks on CPU and GPU

Processor type	Batch_size	Step training time, ms	Epoch training time, s
GPU	64	170	50
GPU	32	90	61
CPU	64	660	210
CPU	32	332	211

Experiment results confirm the well-known fact that the use of a graphic processor for learning neural networks has a significant effect on increasing the speed of the process. Even better results could be achieved if the graphics processor used for the experiment was equipped with specialized tensor cores and a large amount of video memory.

The *batch_size* parameter also affects the learning speed of the neural network, but the amount of video memory on the computer does not allow it to be set too high, so a lot of time is spent by copying data from disk to video memory of the graphics processor. This parameter can significantly speed up the learning process in more complex neural networks with more hidden layers. The *batch_size* parameter does not affect the learning speed of the neural network using the CPU because most of the time is still spent on mathematical calculations and not on working with memory. As was found during the neural network learning process, the batch_size parameter to have a strong effect on network accuracy. The experiment was performed with neural network training for 5 epochs with different values of the batch_size parameter. How batch_size parameter effects the validation accuracy of the network are presented in Table 2.

Table 2. Influence of the *batch_size* parameter on the validation accuracy

Batch_size	Validation accuracy
32	0,9799
64	0,9476
84	0,9281

As can be noted from Table 2, the accuracy of the network is the smaller, the larger the *Batch_size* parameter. This effect can be explained by the fact that in large groups of data, contradictions occur more often during the gradient descent and it is not as rapid as in case of smaller groups of input data. As can be seen from Fig.5, the loss function during neural network training for parameter *batch_size*=32 is always less than for parameter *batch_size*=84. Reducing the loss function for large data groups can be achieved by increasing the learning speed of the network, which affects the weight of each gradient descent. The curves for the loss function during training with a learning speed of 0.003 and 0.001 are presented in Fig. 6. As can be seen from Fig.5 for a larger value of the learning speed we obtain smaller values for the loss function.

Neural network learning is a process that requires a large number of attempts to select the correct model, hyperparameters, optimizer and network loss functions. TensorBoard tool created by TensorFlow developers to analyze the processes that occur with data and parameters during neural network learning. TensorBoard provides visualization, tools needed to experiment with machine learning and profile the learning process of the network.

During the learning process with logging, TensorFlow writes the relevant data to the specified folder in real time. TensorBoard uses a web browser to display model parameters (neural network weight and offset) and metrics (loss or accuracy values) in Fig.(3,4) and others. TensorBoard quickly notices problems such as retraining or a slow gradient descent.

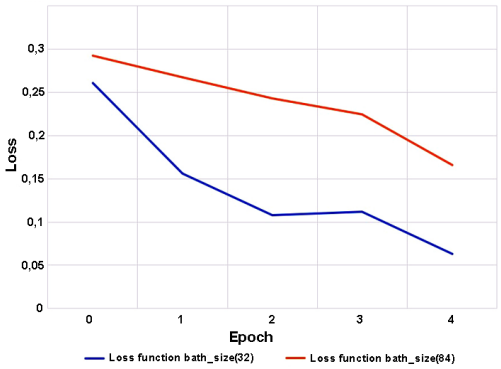

Fig. 5. Loss function with different batch_size parameter

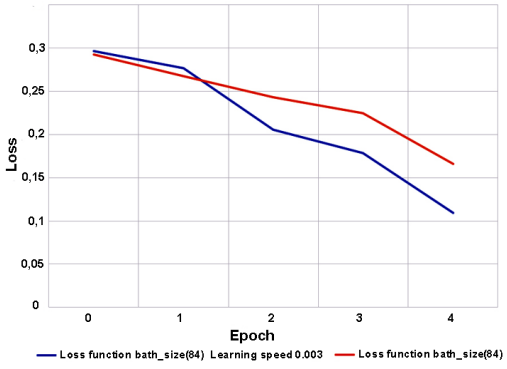

Fig.6. Loss function with different learning speed

The TensorBoard web interface for the profiler tool is shown in Fig.7. The Profiler tool displays statistics on the use of computer resources during network training. It records the time it takes to complete operations, the time it takes to complete the complete steps, gathers information about resource usage in terms of time and memory, and provides visualizations to understand this information. Such information is very important for finding possible optimizations of the learning process because the time spent learning the network strongly affects the speed of finding the optimal network structure. The profile_batch parameter should be added to tensorboard_callback in order to enable profiling. Neural network profiling presented in the article found that 60% of network learning time is spent on memory operations (Fig. 7).

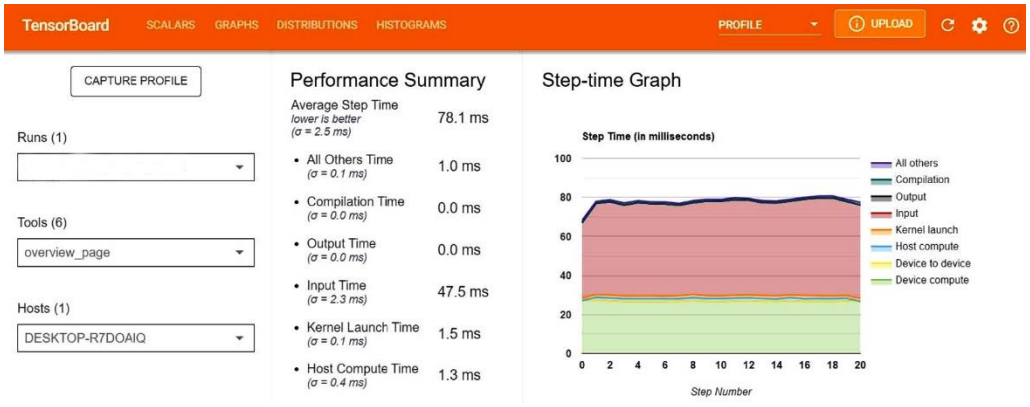

Fig.7. TensorBoard interface for Profiler tool

Conclusion

New tools for manipulation based on artificial intelligence give very convincing results in the field of generation audio deepfake. The article explores the possibility of using deep learning to distinguish deepfake from reality. The structure of a convolutional neural network

with high indicators of recognition accuracy is proposed. A convolutional neural network is constructed from several layers of convolution, layers of maximum and average subsample. The accuracy of the network is the smaller, the larger the *Batch_size* parameter. This effect can be explained by the fact that in large groups of data, contradictions occur more often during the gradient descent and it is not as rapid as in case of smaller groups of input data. The TensorBoard tool was used to monitor and profile the learning process of neural networks. Neural network profiling presented in the article found that 60% of network learning time is spent on memory operations. The created neural network can be used in voice verification systems of users and will allow to check communication channels on attempt of attack by means of the synthesized language.

References

[1] *Nguyen T.* Deep Learning for Deepfakes Creation and Detection: A Survey/ T.Nguyen, Q.Nguyen// Cornell Univer. arXivLabs: Computer Vision and Pattern Recognition arXiv:1909.11573

[2] *Masood M.* Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward/ M.Masood, M.Nawaz, K.Malik, A.Javed, A.Irtaza // Cornell Univer. Cryptography and Security arXivLabs: arXiv:2103.00484

[3] *Almars A.* Deepfakes Detection Techniques Using Deep Learning: A Survey / A.Almars //Journal of Computer Science and Engineering 2021, Vol.9 N.5 DOI: 10.4236/jcc.2021.95003

[4] *Giudice O.* Fighting deepfakes by detecting GAN DCT anomalies/ O.Guidece, L.Guarnera, S.Battiato// Cornell Univer. arXivLabs: Journal Imaging 2021, 7(8), 128 DOI: 10.3390/jimaging7080128/

[5] *Ogihara A.* Discrimination Method of Synthetic Speech Using Pitch Frequency against Synthetic Speech Falsification / A. Ogihara, U. Hitoshi, A.Shiozaki// Iejce Trans. Fundamentals, Vol. E88–A, N.1 2005. P.280-286 DOI:10.1093/ietfec/E88-A.1.280

[6] *Sarasola X.* Application of Pitch Derived Parameters to Speech and Monophonic Singing Classification/ X.Sarasola, E.Navas, D.Tavarez, L.Serrano, I.Saratxaga// Applied Science; Basel Vol.9, Iss.15. 2019. DOI: 10.3390/app9153140

[7] *Mittal T.* Emotions Don't Lie: An Audio-Visual Deepfake Detection Method Using Affective Cues / T.Mittal, U.Bhattacharya, R.Chandra, A.Bera, D.Manocha // Cornell Univer. arXivLabs: Computer Vision and Pattern Recognition arXiv:2003.06711

[8] *Todisco M.* ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection / M. Todisco, X. Wang, V. Vestman, Md Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, K. Lee // Cornell Univer. arXivLabs: experimental projects with community collaborators. – 2019. arXiv:1904.05441

[9] *Abadi M.* TensorFlow: Large_Scale Machine Learning on Heterogeneous Distributed Systems/ M.Abadi, A.Agarwal, Barham P., Brevo E. // Cornell Univer. arXivLabs: Distibuted, Parallel, and Cluster Computing arXiv:1603.04467

[10] *Chetlur S.* cuDNN: Efficient Priitives for Deep Learning /S.Chetlur, C.Woolley, P.Vandermersch, J.Cohen, J.Tran, Catanzaro, Shelhamer E.// Cornell Univer. arXivLabs: Neural and Evolutionary Computing arXiv:1410.0759

ШВИДКІСТЬ НАВЧАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ НА GPU ТА CPU ДЛЯ ВИЯВЛЕННЯ СИНТЕЗОВАНОЇ МОВИ ЗА ДОПОМОГОЮ СПЕКТРОГРАМ

Л.Демків

Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50 79005 Львів, Україна
lidia.demkiv@gmail.com

Розвиток алгоритмів машинного навчання та синтезу людського мовлення дав поштовх сфері створення так званих “deepfake”. Діпфейк – це штучно створене відео, звукозапис, фото, які копіюють голос і зовнішній вигляд інших людей для того щоб ввести глядача в оману. Зазвичай діпфейки створюються за допомогою алгоритмів машинного навчання. Інструменти для створення діпфейків постійно вдосконалюються і стають дедалі небезпечнішими. Серед них дуже небезпечними є аудіо діпфейки тому, що їх дуже важко розпізнати людині на відміну від відео. Також дедалі популярнішими стають системи з голосовим керуванням, які дозволяють ідентифікувати користувача за допомогою голосу. Саме такі системи є особливо вразливими до аудіо діпфейків. Алгоритми і системи, які дозволяють виявити синтезовані звукозаписи становлять велику цінність для подальшого існування голосової комунікації через мережу інтернет.

В роботі розроблено і натреновано згорткову нейронну мережу для детектування і класифікації синтезованого мовлення. Нейронну мережу побудовано з декількох шарів згортки, шарів максимальної і усередненої підвибірки. На виході нейронної мережі розміщено шар з одним нейроном із сигмоїдальною функцією активації для визначення типу мовлення. Розроблено скрипт для генерації мел-спектрограм з вхідного аудіосигналу. Візуальний аналіз та співставлення спектрограм природнього та синтезованого мовлення показує відсутність чітких візуальних ознак за якими можна було б чітко сказати до якого типу мовлення (природнього чи синтезованого) відносяться ці спектрограми. Отримана нейронна мережа володіє високими показниками точності розпізнавання мовлення, яке є представлене за допомогою спектрограм. Проведено порівняння швидкості навчання нейронних мереж на GPU і CPU. Використано інструмент для моніторингу і профілювання процесу навчання нейронних мереж TensorBoard. Встановлено що 60% часу навчання мережі витрачається на операції роботи з пам’яттю. Досліджено вплив параметра нейронної мережі batch_size на точність мережі і на швидкість процесу навчання. Для реалізації проекту використано мову програмування Python, бібліотеку TensorFlow у поєднанні з високорівневим API Keras, CUDA та базу звукозаписів ASVspoof 2019 у форматі flac.

Ключові слова: аудіо діпфейк, мел-частотні спектрограми звуку, згорткові нейронні мережі, швидкість навчання нейромереж.

Стаття: надійшла до редакції 09.12.2021,
доопрацьована 11.12.2021,
прийнята до друку 13.12.2021