



UDC 519.6

PACS 07.05.Tp,

DOI: 10.22363/2658-4670-2023-31-3-273-281

EDN: KPCBBQ

Brain–computer interaction modeling based on the stable diffusion model

Eugeny Yu. Shchetinin

*Financial University under the Government of the Russian Federation
49, Leningradsky Prospect, Moscow, 125993, Russian Federation*

(received: August 7, 2023; revised: August 25, 2023; accepted: September 8, 2023)

Abstract. This paper investigates neurotechnologies for developing brain–computer interaction (BCI) based on the generative deep learning Stable Diffusion model. An algorithm for modeling BCI is proposed and its training and testing on artificial data is described. The results are encouraging researchers and can be used in various areas of BCI, such as distance learning, remote medicine and the creation of robotic humanoids, etc.

Key words and phrases: neural network technology, brain–computer system, stable diffusion

1. Introduction

An increasing number of researches in the field of artificial intelligence are aimed at development of neurotechnological applications using advances in generative deep learning. Many of these studies focus on using machine learning to analyze or decode brain signals, and lead to the creation of various biomedical devices that help people improve their quality of life [1]. One of the applications of machine learning in neurotechnology is modeling brain–computer interaction (BCI). There are many different approaches to BCI modeling, including the use of different machine learning models and neural network architectures. One of the recent developments in this field is Stable Diffusion (SD), which allows generating samples with a predetermined distribution using a stochastic diffusion process [2].

Stable Diffusion (Stable Diffusion) is one of the approaches for implementing stochastic diffusion, which takes into account the peculiarities of the input data distribution. In particular, Stable Diffusion is a generalized Cauchy distribution, which is a mixture of distributions with heavy tails and takes into account the presence of heavy outliers in the data. This makes Stable Diffusion particularly useful for modeling brain–computer interactions because

© Shchetinin E. Y., 2023



This work is licensed under a Creative Commons Attribution 4.0 International License

<https://creativecommons.org/licenses/by-nc/4.0/legalcode>

data from electroencephalography (EEG) or magnetoencephalography (MEG) often contain spikes and noise. Stable diffusion is the process of solving the Fokker–Planck equation, which describes the evolution of the probability density on the time axis. This probability density is usually a set of parameters or latent features that are valuable indicators for processing EEG or MEG data. Stable diffusion allows not only to generate new samples based on these parameters, but also to solve many other tasks related to data analysis, such as classification, regression, clustering, etc. [3, 4]. Moreover, it can be used in improving the quality of BCI, by improving the accuracy of decoding and computer interface control. A number of works have demonstrated the successful application of latent diffusion methods in the tasks of medical rehabilitation [5–7], research of biological structures [7, 8], development of humanoid robots [9–11].

This paper proposes a brain–computer interaction algorithm based on the Stable Diffusion model for BCI modeling in the learning process. The process of model creation is described, starting from data preparation and ending with the implementation of the model in a computer interface control application.

2. Stable Diffusion generative deep learning models

Diffusion models are machine learning models that learn to decompose random Gaussian noise step-by-step to produce a pattern of interest, such as an image [12–14]. The diffusion model has a significant disadvantage, since the denoising process is time and memory consuming. This makes the process slow and requires a lot of memory. The main reason for this is that they work in pixel space, which becomes unreasonably expensive, especially when generating high-resolution images. Stable diffusion was introduced to solve this problem because it depends on latent diffusion. Latent diffusion reduces memory and computational overhead by applying the diffusion process to a lower-dimensional latent space instead of using the actual pixel space.

Understanding the Basics of Denoising Diffusion Probabilistic Models.

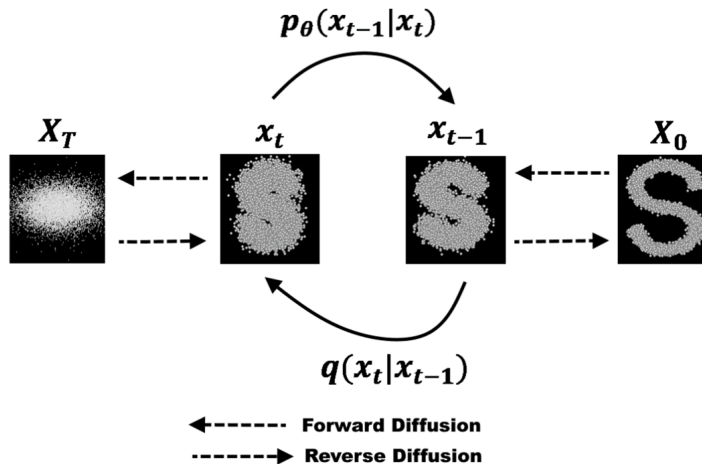


Figure 1. Process of Denoising Diffusion Probabilistic Model (Image by author)

There are three main components in latent diffusion, the most important of which is the variation autoencoder (VAE). The autoencoder (VAE) consists of two main parts: an encoder and a decoder. The encoder converts the images into a low-dimensional latent representation, which will be the input for the next component, the U-Net. The decoder does the reverse work, converting the latent representation back into an image. The U-Net is also made up of encoder and decoder parts, and both are made up of ResNet blocks. The encoder compresses the image into a lower resolution image, and the decoder decodes the lower resolution image back into a higher resolution image.

To ensure that the U-net does not lose important information when downsampling, short connections are usually added between the encoder’s ResNet networks for downsampling and the decoder’s ResNet networks for upsampling. In addition, a stable diffusion U-net is capable of conditioning its output to text embeddings by means of cross-attention layers. Cross-attention layers are added to both the encoding and decoding parts of the U-net, usually between ResNet blocks. The encoder is used to obtain a latent representation (latent) of the input images for the direct diffusion process during latent diffusion training. While during output, the VAE decoder converts the latent representation back into an image.

Text encoder. The text encoder converts an input query, such as “Pikachu will have a nice dinner with a view of the Eph-phil tower”, into a space of embedding that can be understood by the U-net. This would be a simple transformer-based encoder that maps a sequence of tokens into a sequence of hidden embeddings of text. It is important to use a good cue to get the expected result. That’s why cue engineering is trending right now. Cue engineering is the search for specific words that can make a model produce a result with certain properties.

The reason latent diffusion is fast and efficient is because the U-net of latent diffusion works in low-dimensional space. This reduces memory size and computational complexity compared to diffusion in pixel space. For example, the autoencoder used in Stable Diffusion has a reduction factor of 8. This means that the shape image (3, 512, 512) becomes (4, 64, 64) in latent space, which requires 64 times less memory. The stable diffusion model first takes a latent seed and a text cue as input. The latent seed is then used to generate random representations of latent images of size 64×64 , and the text cue is converted into 77×768 text blobs using the CLIP text encoder.

The U-network then iteratively discolors the random representations of the hidden images, being conditioned by the text embeddings. The output of the U-net, which is the residual noise, is used to compute the representation of the hidden image using the scheduler algorithm. The scheduler algorithms compute the predicted representation of the cleaned image based on the previous noise representation and the predicted noise residual. Many different scheduler algorithms can be used for these calculations, each with its own pros and cons. For Stable Diffusion, we recommend using one of the following:

- PNDM scheduler (used by default);
- DDIM scheduler;
- K-LMS scheduler.

The denoising process is repeated about 50 times to get a better representation of the latent image step by step. After the process is completed, the latent image representation is decoded by a part of the variational autocoder, the decoder.

Pre-trained latent diffusion models were used to develop our project. The pre-trained diffusion model includes all the components needed to create a complete diffusion pipeline. They are stored in the following folders:

- `text_encoder`: Stable Diffusion uses CLIP, but other diffusion models may use other encoders, such as BERT.
- `tokenizer`: This must match the one used by the `text_encoder` model.
- `scheduler`: The scheduling algorithm used to postpone adding noise to the image during training.
- `U-Net`: The model used to generate a latent representation of the input data.
- `VAE`: The autoencoder module we will use to decode the latent representations into real images.

We can load the components by accessing the folder in which they were saved, using the subfolder argument for `from_pretrained` [8].

3. Developing a brain–computer interaction algorithm based on the SD model

Step 1. Data preparation

The first step in creating the Stable Diffusion model for BCI modeling is to prepare the data. As data, we will use a set of EEG data obtained from subjects who performed the task of memorizing numbers. Each experiment consisted of several trials, each of which required the subject to memorize a specific digit displayed on a screen and then reproduce it using reasoning.

EEG electrodes placed on the subjects' heads were used to acquire the data, and the signals were digitized and recorded on a computer. This data consisted of several channels and included information about the temporal distribution of the signals received from each electrode.

Before processing the data, we performed preprocessing, including noise filtering and outlier elimination. In addition, we created a function to convert the data into a format suitable for model processing and training. At this stage we have prepared the data set which contains the information about the time distribution of the EEG signals for the purpose of training and testing the Stable Diffusion model.

Step 2: Model development

After receiving the data we began to develop the model itself. In our case we chose the Stochastic Diffusion model using Stable Diffusion as the diffusion process. It is worth noting that we used the TensorFlow Probability library to implement this model [15–18]. Stable Diffusion was implemented in the form of a parametrized distribution, which is specified by two parameters – the exponent and the scale. These parameters were trained on EEG data and used to generate new samples. In addition, we used the autoencoder model to train latent features from the EEG data, which were then used as input parameters for the Stable Diffusion model. In general, the model was trained on the EEG data, which were divided into a training set, a control set, and a test set. During training, we used the maximum likelihood method

to optimize the model parameters. In addition, we used L2 regularization to prevent model overtraining.

Step 3: Testing the Model

After training the model, we proceeded to test it on the control and test datasets. We used metrics such as accuracy, AUC ROC, f1-score, and error matrix to evaluate the quality of the model. We also tested how well our model performed on new data by displaying samples created with the model and comparing them to real data. We found that our model fairly accurately reflected the distribution of the EEG data and allowed us to generate new samples that seemed similar to the real data.

Step 4: Implement the model in the application

Finally, we set about incorporating the model into a computer interface management application. We used the TensorFlow Serving library to run our model on a remote server that handled requests from the application and returned real-time predicates. In our application, the user could control the computer with his mind. He could select commands such as “up”, “down”, “left”, and “right” just by thinking about those commands. These mental commands were passed to the TensorFlow server through our Python library, which in turn used the Stable Diffusion model to classify the user’s EEG signals and determine his intentions.

Step 5: Implementing the model in the training process

After we successfully trained and tested the Stable Diffusion model for BCI modeling, we proceeded to implement it into the learning process. We created an interactive application that allows users to control the computer interface through thinking. The application is a scenario in which the user is asked to perform a task, such as moving the mouse cursor around a target and pressing buttons. To control the thinking, the user looks at a symbol that corresponds to a given command and directs his or her attention to that symbol. Then, the application’s interfaces use EEG signals to recognize that symbol and execute the appropriate command.

4. Discussion of results

In this paper, we have reviewed the details of creating a Stable Diffusion model to simulate brain–computer interaction in the learning process. We described in detail how to prepare the data, develop the model, and test it to determine the quality of its performance. We also demonstrated how our Stable Diffusion model can be applied to create an interactive application that allows users to manipulate the computer interface with their thinking. The Stable Diffusion model we developed has been shown to work well in simulating brain–computer interaction in the learning process. We created a model that could process EEG data and generate new samples that matched the distribution of the real data. We also successfully incorporated this model into a computer interface control application where the user could control the

computer by thinking. Moreover, our Stable Diffusion model can solve not only the classification problem, but also some other data processing problems, such as clustering and regression. It can be used not only for modeling the interaction between the brain and the computer, but also for other tasks related to time series analysis and modeling of temporal processes. As a result, the Stable Diffusion model is a powerful tool for modeling and constructing ROC/PR curves to evaluate the quality of model performance.

To test the model we used different tasks related to EEG signal decoding. For example, we used different types of classification with a number of classes from 2 to 10, including one-vs-all and multi-class classification. We also performed clustering analysis to see which activity patterns could be extracted from the data. During model testing, we obtained high accuracy and AUC values, as well as high clustering quality. Overall, the model gave good results on all tasks, indicating its validity and applicability to BCI modeling.

One of the main advantages of modeling brain–computer interaction with Stable Diffusion is that it takes into account the peculiarities of input data distribution, which allows to work more efficiently with data containing noise and spikes. Furthermore, using Stable Diffusion maximizes the likelihood of the data, which improves the accuracy of decoding and controlling computer interfaces. Another important advantage of simulating brain–computer interaction with Stable Diffusion is that it can be run in real-time online. This is especially important when applying such a model to real-world computer interface control tasks, where fast response and accuracy are critical.

5. Conclusion

The Stable Diffusion model is a powerful tool for modeling brain–computer interaction, and can be used to create biomedical devices that help people improve their quality of life. For example, such devices can be used to control prostheses, control mouse cursors, or play computer games without using hands or voice. However, at this point, our knowledge of the actual capabilities and limits of this model and the specific solutions it can provide is far from complete, which opens up many possibilities for future research and development. The use of brain–computer interaction modeling to control computational interfaces is an experimental and promising area that has great potential for future development. However, it is also a challenging task that requires high skill and expertise in neurotechnology, machine learning, and biomedical sciences.

Successful implementation of such a model requires the use of modern data processing and analysis methods, as well as a strong technical base, including powerful computing resources and highly specialized devices for EEG and MEG data acquisition and processing.

Despite limitations and disadvantages, the use of brain–computer interaction modeling is one of the most promising and attractive directions in neurotechnology, and has the potential to significantly change our lives in the future. And in conclusion it is worth mentioning that the future of this direction is determined by how fast, scientists can develop models which can be of real interest for people and be widely used in their lives. They can change existing work processes and teach us new ways to manage the world around us through our thoughts, and help people with disabilities in their daily lives.

That is why this area is now actively researched and developed all over the world. To date, there are several successful prototypes of computer-interface control devices that use brain–computer interaction, and undoubtedly, we will see more potential applications of this technology in the future.

However, it is important to note that the use of brain–computer interaction technologies also raises ethical and safety issues. Thus, in order to achieve the most positive results, it is necessary to pay due attention to security and ethical issues, including data privacy and information security rights of individual users.

In general, the application of brain–computer interaction modeling is an important direction in neurotechnology, which opens up possibilities for a wide range of new applications. However, in order to successfully achieve the results and to use this technology in everyday life, careful work and continuous search for new solutions and improvement of technologies are needed.

References

- [1] W. Li, Y. Chen, X. Huang, G. Wang, and X. Zhang, “Combining multiple statistical methods to improve EEG-based decoding for BCI applications,” *Applications. IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 8896–8906, 2019.
- [2] C. Yen and C. Lin, “A real-time brain–computer interface system for the upper limb using feedback training based on motor imagery,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 10, pp. 2087–2096, 2019.
- [3] H. Zhang, W. Zheng, K. Zhang, Y. Li, Y. Wang, and L. Yao, “Stochastic channel effects modeling and training deep spiking neural networks for brain–computer interface,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 350–364, 2019.
- [4] D. Zhu, J. Bieger, and A. Datta, “Brain–computer interfaces in neurorehabilitation: a review of recent progress,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 6, pp. 1319–1339, 2019.
- [5] D. Wu, B. Wang, Y. Li, J. Shen, and G. Wang, “A review of EEG-based brain–computer interface for medical robotic system control,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 6, pp. 1233–1244, 2020.
- [6] S. Bhattacharya, “A brief review of brain–computer interface for neuropsychological rehabilitation,” *IEEE Reviews in Biomedical Engineering*, no. 12, pp. 95–107, 2019.
- [7] X. Li, D. Zhu, H. Chen, Y. Zhang, and X. Wu, “A BMI system for rehabilitation of hemiplegic patients based on transcranial direct current stimulation and affective feedback,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 535–544, 2019.
- [8] Z. Liu, Y. Li, L. Cheng, Q. Zhang, M. Wang, L. Kong, and Y. Wang, “An EEG-based brain–computer interface system for independent living of people with cerebral palsy,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 7, pp. 1927–1935, 2020.

- [9] Y. Hu, Y. Hou, M. Wang, T. Yu, and J. Zhang, “EEG-based motor imagery BCI system via supervised joint blind source separation and convolutional neural network,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, no. 29, pp. 1586–1597, 2021.
- [10] G. Choi, W. Ko, Y. Jung, S. Jo, K. Kim, and S. Lee, “A review on recent progress in EEG-based brain–computer interface for assistive robotic control,” *IEEE Reviews in Biomedical Engineering*, no. 12, pp. 141–157, 2019.
- [11] M. Rashid, J. Höhne, G. Schmitz, and G. Müller-Putz, “A review of humanoid robots controlled by brain–computer interfaces,” *Frontiers in Neurorobotics*, no. 14, pp. 1–28, 2020.
- [12] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, 2020. DOI: 10.48550/arXiv.2006.11239.
- [13] P. Dhariwal and A. Nichol, *Diffusion models beat GANs on image synthesis*, 2021. DOI: 10.48550/arXiv.2105.05233.
- [14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and O. B., *High-resolution image synthesis with latent diffusion models*, 2021. DOI: 10.48550/arXiv.2112.10752.
- [15] A. Blattmann, R. Rombach, K. Oktay, and B. Ommer. “Latent diffusion models.” (2022), [Online]. Available: <https://github.com/CompVis/latent-diffusion>.
- [16] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep unsupervised learning using nonequilibrium thermodynamics*, 2015. DOI: 10.48550/arXiv.1503.03585.
- [17] M. Welling and Y. W. Teh, “Bayesian learning via stochastic gradient Langevin dynamics,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML’11, Madison, WI, USA: Omnipress, 2011, pp. 681–688. DOI: 10.5555/3104482.3104568.
- [18] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, 2020. DOI: 10.48550/arXiv.2006.11239.

For citation:

E. Y. Shchetinin, Brain–computer interaction modeling based on the stable diffusion model, *Discrete and Continuous Models and Applied Computational Science* 31 (3) (2023) 273–281. DOI: 10.22363/2658-4670-2023-31-3-273-281.

Information about the authors:

Shchetinin, Eugeny Yu. — Doctor of Physical and Mathematical Sciences, Lecturer of Department of Mathematics, Financial University under the Government of the Russian Federation (e-mail: riviera-molto@mail.ru, ORCID: <https://orcid.org/0000-0003-3651-7629>)

УДК 519.6

PACS 07.05.Tr,

DOI: 10.22363/2658-4670-2023-31-3-273-281

EDN: KPCBVQ

Моделирование взаимодействия «мозг – компьютер» на основе модели стабильной диффузии

Е. Ю. Щетинин

*Финансовый университет при Правительстве Российской Федерации
Ленинградский проспект, д. 49, Москва, 125993, Россия*

Аннотация. В этой статье исследуются нейротехнологии для развития взаимодействия «мозг – компьютер» (BCI) на основе генеративной модели стабильной диффузии с глубоким обучением. Предложен алгоритм моделирования BCI и описано его обучение и тестирование на искусственных данных. Полученные результаты обнадеживают исследователей и могут быть использованы в различных областях BCI, таких как дистанционное обучение, удалённая медицина, создание роботов-гуманоидов и т. д.

Ключевые слова: технология нейронных сетей, система мозг-компьютер, стабильная диффузия