Improving motor imagery classification using generative models and artificial EEG signals

Krish Kabra, Daniel Truong, Calvin Chang
Department of Electrical and Computer Engineering, UCLA
420 Westwood Plaza, Los Angeles, CA 90095
{krish97, dktruong, calvinchang33}@g.ucla.edu

Abstract

Convolutional neural networks (CNN) are shown to be extremely successful at motor imagery task classification with high accuracy. However, a major pitfall of deep learning models is the necessity to train with large datasets. In this report, we procure artificial EEG data from a limited dataset using generative models such as the generative adversarial network (GAN) and variational autoencoder (VAE). We show approximately 4% improvement in task classification when using a mixed training set containing both real and artificial data generated by a GAN. We fail to train a VAE and discuss possible reasons, as well as future directions.

1. Introduction

Motor imagery classification (MIC) using electroencephalogram (EEG) signals is an essential tool for constructing non-invasive brain-computer interfaces (BCI) [1, 2, 3]. With recent advances in deep learning for image classification, there has been an interest to use similar architectures for MIC [4]. In particular, convolutional neural networks (CNN) have shown promising results for EEG-based MIC [5].

One of the major challenges for EEG-based MIC is the small dataset sizes that are obtained from individual subjects. This is a major drawback for classification algorithms, especially those reliant on deep learning methods. Consequently, research on generating artificial EEG data is of great interest within the community.

Generative adversarial networks (GAN) [6] have proven themselves to be state-of-the-art tools for generating synthetic images, audio and videos. For MIC, recent works have already used GANs to improve classification accuracy through augmenting training dataset sizes by including artificially generated data [7, 8]. Our work is in large part inspired by these efforts.

In this report, we present MIC results for data obtained from the *BCI Competition IV, Dataset 2a* [9, 10]. We first construct a baseline classification model, which is chosen to be the Shallow CNN from [5] with minimal changes, due to its robustness and high accuracy over other classification networks. We then implement various data augmentation techniques to observe how classification is affected. These include subsampling, obtaining random crops, obtaining sequential crops (similar to [5]), and applying a continuous wavelet transform (CWT).

Subsampling and cropping are chosen due to their simplicity and ability to increase dataset size. More specifically, subsampling is possible as the dataset is sampled at 250 Hz, whereas typical EEG brain activity is within the range of 0-15 Hz. Cropping is performed as the dataset includes activity from a 4 second window after a visual cue. Humans typically have a reaction time on the order of 250 ms, and the subject may not perform the activity throughout the entire recorded window. CWT is performed as feature extraction method since it is known that brain activity can be divided into frequency bands. It is chosen over other time-frequency analysis methods such as the short-time-Fourier transform (STFT) due to its strong ability to analyze transient signals [11].

Finally, we construct two generative models for artificial EEG data production: a variational autoencoder (VAE) [12] and a Wasserstein GAN (WGAN) with a gradient penalty (GP) [13]. The VAE is considered due to its strong ability to learn input data distributions. We expect the model to successfully encode useful features from the raw EEG signal. However, we also suspect it to perform worse at generating artificial data. On the other hand, the WGAN-GP is used to due to its excellent ability to recreate input data features. However, if fed raw EEG signals, we fear the GAN will fail to understand what features are important when generating artificial data. To combat this, we feed the GAN input data that has undergone a CWT. The WGAN-GP is chosen over other GAN architectures due to its empirical training stability, which includes avoiding mode-collapse and successful

convergence. Specific details of all the network implementations are included in Appendix B.

2. Results

We used four different aforementioned data augmentation techniques to augment the size of the EEG dataset and evaluated their performance effects on the CNN test accuracy. As shown in Fig. 1, sequential cropping performs the best. Moreoever, compared to the baseline of no data augmentation at all, sequential cropping is the only method that adds a beneficial effect. To evaluate these data augmentation methods on the test set, the extrapolated samples from each trial, whether it be subsampled or cropped, have their scores averaged to get one prediction from each trial. From the model utilizing sequentially cropped data, we analyzed the accuracies for each task by creating a confusion matrix shown in Fig. 2, which is normalized to the predicted labels.

To experiment further with the Shallow CNN, we trained and tested per subject to see how the model performed. In Fig. 3 the test accuracies average to 72.3% with a standard deviation of 10.5%. We note each model had different learning rates implemented for optimal performance.

The WGAN-GP is implemented using data only from subject 6 with 5 EEG channels corresponding to probe locations Fz, C3, Cz, C4 and Pz, which we refer to as channels 1-5 respectively. Further explanation for this limited input data is given in Appendix B. Also included in the appendix are examples of generated CWT EEG data for all 4 motor imagery tasks (see Fig. 6). Although comparisons between original and generated data is hand-picked, it clearly shows that the GAN has learned features that are prevalent within the original input data.

We then generate 100 artificial CWT EEG signals for each of the 4 tasks, for a total of 400 additional samples in our training data set for subject 6. We show the results when appending the training dataset with various ratios of the total artificial dataset in Fig. 4. 3 trials of training were taken on each augmented dataset with 0%, 25%, 50%, and 100% of the artificial data appended, for 30 epochs each. For reference, the natural dataset had 1180. The boxplot shows the range, indicated by the length of the vertical line, and the mean of the test accuracies, which is indicated by the horizontal orange line. The box itself represents the cutoff of a quartile from the mean, assuming a normal distribution.

Unfortunately, we failed to train our VAE to generate EEG data. An example of a single artificial trial signal is show in Fig. 5. Although the trial looks like a valid EEG signal form, all the probe channels collapse to the same signal, suggesting mode-collapse failure.

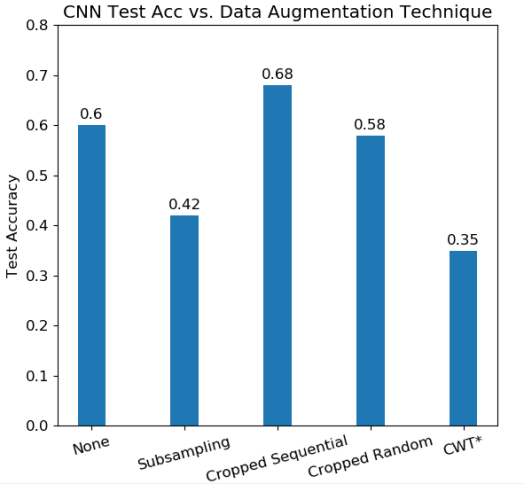

Figure 1. Overall test set accuracy of the Shallow CNN against different data augmentation techniques. Sequential cropping performs the best, while subsampling and random cropping deteriorates the accuracy from the baseline. *CWT data is evaluated on different CNN due to being an invalid size to the Shallow CNN. This data also uses only 5 EEG channels, as opposed to the given 22 channels.

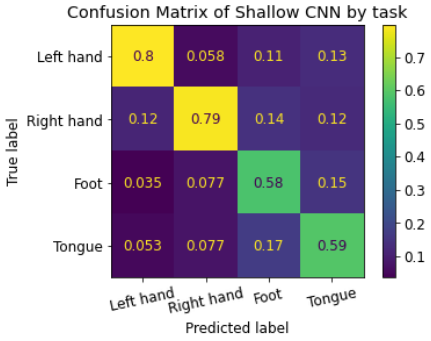

Figure 2. Confusion matrix on predicted labels for the Shallow CNN on cropped data across all subjects. Values are normalized to amount of predictions made for each task.

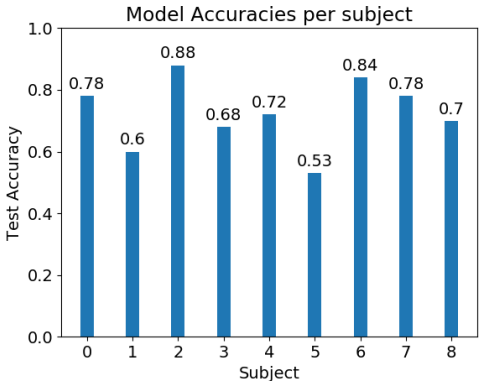

Figure 3. Using the cropped sequential data augmentation, we trained and tested the Shallow CNN only on one subject at a time. We can see how the CNN extracts features that generalize nicely to the test set in subjects 0, 1, 6, and 7.

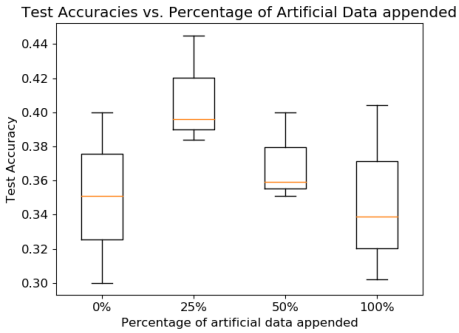

Figure 4. Results of the CNN trained on CWT data for one subject with only 5 EEGs extracted. The orange line within each box represents the mean. 3 trials were ran for each of the scenarios. The natural dataset had 1180 trials while the artificial dataset had 400 trials.

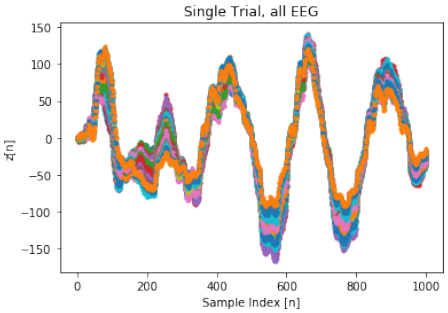

Figure 5. Example of generated EEG signal for all 22 channels. Each color corresponds to a different EEG channel probe. The signal looks like a legitimate EEG signal, however, the fact that all the channels look almost identical is a clear indication of training failure due to mode-collapse.

3. Discussion

When augmenting the data using subsampling, we sampled every 5 time bins, causing the number of data trials to be 5 times larger and the time length to be 5 times smaller. Subsampling causes the training and validation data to be highly correlated to each other, making the CNN overfit on the training data. Once the CNN overfits, it will learn the nuances and noise of the training data, features which cause the CNN to perform badly on new data.

Random cropping also caused the CNN to perform worse on the testing data. This may be because the cropping is not guaranteed to emphasize important time bins. The data augmentation method that improved performance was cropping sequentially, as it gave an 8% increase in test accuracy after training. An empirical reason why cropping sequentially may work is because the middle portion on the EEG signal will be repeated in all the crops taken of the signal, given that the length of the crop > 500 time bins.

The CWT augmentation resulted in a very poor baseline classification accuracy. One explanation for this inferior result is due to poor architecture choice. The CNN for CWT was built upon the approximation that the CWT data can treated like images, which may not necessarily be true. Furthermore, the CNN for CWT was made to be shallow such that it could be compared with the baseline Shallow CNN. We suspect the architecture could be made deeper in order to improve accuracy. Another explanation for the inferior result is the reduced dimensionality of the EEG data when training the CNN for CWT. Due to computational memory and processing time considerations, we chose to reduce the number of EEG channels used in training from 22 channels to 5 channels.

With regards to artificial data synthesis, the WGAN-GP successfully identifies features present in the real data and generates a variety of artificial data. This results in a significant improvement of the CNN classification when implemented on CWT data. The training accuracy is largest when 25% of the artificial data is appended to the training set, and worsens for larger and larger ratios. This worsening is expected as the network begins to learn more features from the artificial data that may not be present in the real data. The architecture itself was robust to mode-collapse and convergence failure, which is a common pitfall for most GAN architectures.

Finally, the VAE suffered from individual examples being mapped to the same random distribution in the latent space. This implies that the decoder ignored the latent variable input and generated an output less arbitrarily. Not identically similar, but in essence, the VAE suffered from mode-collapse. This led to problems including, not being able to create a data set for the CNN to train on. Due to the time constraint, the exact cause of the problem still remains unclear. The leading insight is that the model is too constrictive for individual training examples [14].

Overall, we show that it is not only possible to synthesize artificial EEG data, but it is also possible to use this data to improve MIC. For future work, an interesting architecture we suspect may be ideal for artificial EEG data generation is the VAE-GAN [15]. As the name suggests, this model combines the impressive feature encoding ability of the VAE with the strong feature replication ability of the GAN. The implications of our findings are exciting for the broader community utilizing deep learning models for BCIs and MIC. Nevertheless, we remain wary that the ability to generate artificial EEG data may have negative ramifications in the future, such as exposing BCI users to bad actors who could send fake tasks forcing the user to perform unwanted actions.

References

[1] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, p. 031005, apr 2018. [Online]. Available: https://doi.org/10.1088%2F1741-2552%2Faab2f2

[2] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767 – 791, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1388245702000573

[3] L. M. Alonso-Valerdi, R. A. Salido-Ruiz, and R. A. Ramirez-Mendoza, "Motor imagery based brain–computer interfaces: An emerging technology to rehabilitate motor deficits," *Neuropsychologia*, vol. 79, pp. 354 – 363, 2015, special Issue: Sensory Motor Integration. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002839321530155X

[4] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," *Journal of Neural Engineering*, vol. 16, no. 3, p. 031001, apr 2019. [Online]. Available: https://doi.org/10.1088%2F1741-2552%2Fab0ab5

[5] R. Tibor Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *arXiv e-prints*, p. arXiv:1703.05051, Mar. 2017.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[7] S. M. Abdelfattah, G. M. Abdelrahman, and M. Wang, "Augmenting the size of eeg datasets using generative adversarial networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–6.

[8] Q. Zhang and Y. Liu, "Improving brain computer interface performance by data augmentation with conditional deep convolutional generative adversarial networks," *CoRR*, vol. abs/1806.07108, 2018. [Online]. Available: http://arxiv.org/abs/1806.07108

[9] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller, "Seperability of four-class motor imagery data using independent components analysis," *Journal of Neural Engineering*, vol. 3, no. 3, pp. 208–216, jun 2006. [Online]. Available: https://doi.org/10.1088%2F1741-2560%2F3%2F3%2F003

[10] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. Miller, G. Mueller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the bci competition iv," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2012.00055

[11] M. Kıymık, İnan Güler, A. Dizibüyük, and M. Akın, "Comparison of stft and wavelet transform methods in determining epileptic seizure activity in eeg signals for real-time application," *Computers in Biology and Medicine*, vol. 35, no. 7, pp. 603 – 616, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010482504000691

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *CoRR*, vol. abs/1704.00028, 2017. [Online]. Available: http://arxiv.org/abs/1704.00028

[14] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," 2017.

[15] A. B. L. Larsen, S. K. Sønderby, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *CoRR*, vol. abs/1512.09300, 2015. [Online]. Available: http://arxiv.org/abs/1512.09300

[16] C. Sønderby, T. Raiko, L. Maaløe, S. Sønderby, and O. Winther, "How to train deep variational autoencoders and probabilistic ladder networks," in *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, ser. JMLR: Workshop and Conference Proceedings, 2016.

A. Performance

Model Name	Data Augmentation Method	Test Set Accuracy
Shallow CNN	None	60%
Shallow CNN	Subsampling	42%
Shallow CNN	Random Cropping	58%
Shallow CNN	Sequential Cropping	68%
CNN for CWT	CWT with 5 EEGs	37%

Table 1. Performance of the model on different data augmentation methods. Sequential cropping proved to outperform all other techniques

Number of artificial samples appended	Test set accuracy
0	35.03%
100	40.82%
200	37.01%
400	34.83%

Table 2. Comparison of how many artificial samples appended to natural dataset of 1180 vs. Test set accuracy

Model Name	Performance Summary
DCGAN	Mode-collapsed
WGAN	Convergence failure (500 epochs)
WGAN-GP	Successfully converges
CNN-VAE	Mode-collapsed

Table 3. Details on the performance of various generative models implemented.

B. Architectures

The Shallow CNN is from prior work [5] and is proven to perform well on the BCI dataset. The CNN architecture, shown in Fig. 7, had a matrix input where each row corresponded to an EEG channel and each column corresponded to a time bin. The model uses separate temporal and spatial convolution layers to split the dimensions into two jobs. We added another dense layer with 100 hidden units before the last dense layer. Dropout had a rate of 0.5.

We tried to limit the CNN architecture to only two convolutional layers to be comparable to the Shallow CNN. Kernel sizes were set to be $(7, 7)$ in order to capture a large receptive field. Furthermore, batch norm, dropout, and max pooling were applied as if the input were a regular image. The input data was an image-like tensor where each 'color-channel' corresponded to an EEG channel, each row corresponded to a frequency bin between 1-20 Hz, and each column corresponded to a time bin. Due to computational memory and processing time, we limited the number of EEG channels included in the input data to 5 channels. We

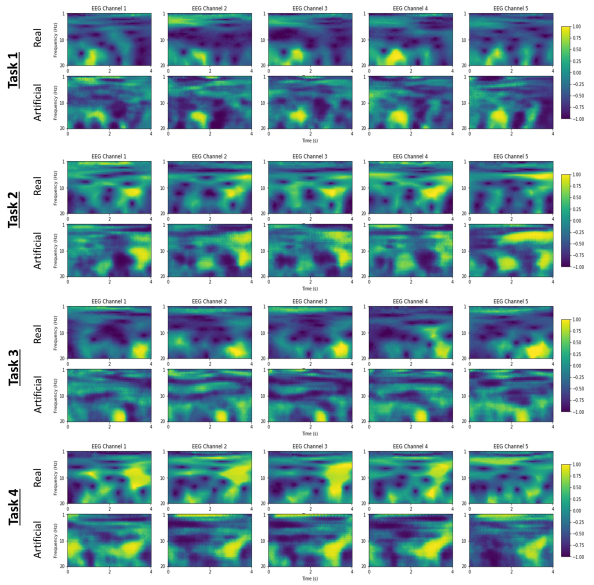

Figure 6. Examples of generated CWT EEG data for all 4 motor imagery tasks. Each column corresponds to a different EEG channel. A comparison between real input data and generated data shows that the WGAN-GP has learnt features. We note these examples were hand-picked, and that there is a wide variety of signal data.

decided on probe locations Fz, C3, Cz, C4 and Pz for maximal scalp coverage.

We implemented 3 GAN architectures: a deep convolutional GAN (DCGAN), a traditional WGAN and the reported WGAN-GP. All 3 GANs are based on the same generator and discriminator networks, shown in Fig. 9, and utilized the same input data as the CNN for CWT. The generator utilizes transposed convolutions for upsampling. The only difference between the models was the output activation of the discriminator and implemented loss function. The DCGAN uses a sigmoid output activation and a binary cross-entropy loss function. Both the WGANs use a linear output activation and calculate a Wasserstein distance loss function. The WGAN-GP augments the Wasserstein distance by adding a gradient penalty term.

The CNN VAE architecture utilises a convolutional encoder and a transpose convolution based in Fig. 10. In the convolutional layers, a temporal and spatial convolution is performed to drastically further reduce the parameter size and training time. In the architecture, regularizers that affect the stochasticity of the model such as batchnorm were left out. Empirically, they seemed to have no noticeable affect on the model's diagnosis for mode-collapse. However, in models with more layers it wo uld be crucial for these regularizers to be implemented in an abnormal and strategic way [16]. This final model was used after iterations of different models including multi-layer perceptron (MLP) VAE and a traditional CNN VAE.

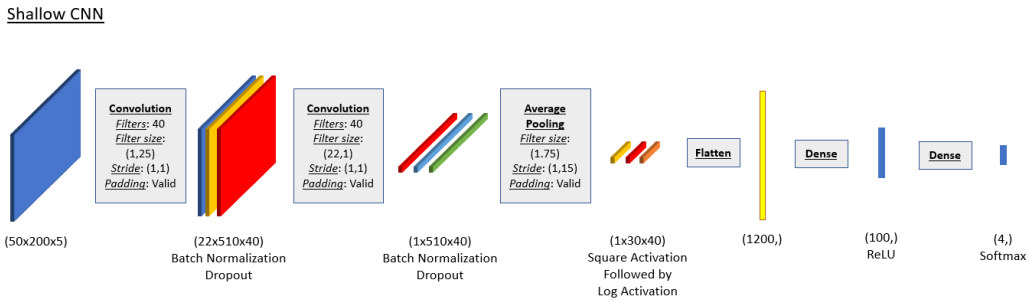

Figure 7. Shallow CNN referenced from [5]

.

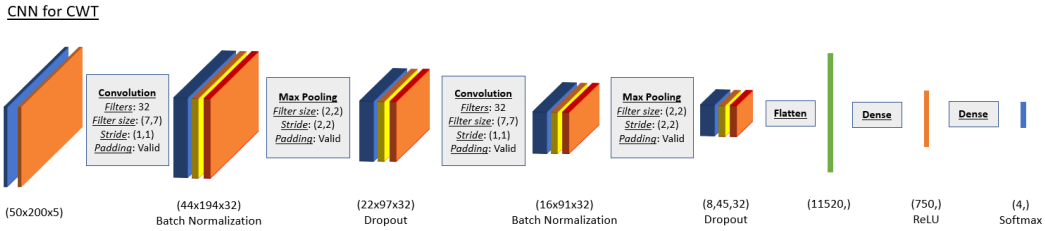

Figure 8. CNN architecture used for evaluating CWT data

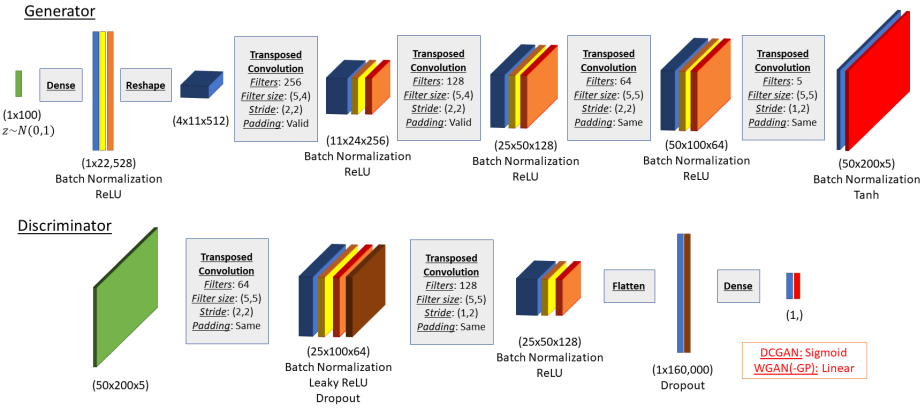

Figure 9. GAN architecture. All implemented GANs utilized the same generator and discriminator architectures. Only the discriminator output activation and loss function differ between GAN models.

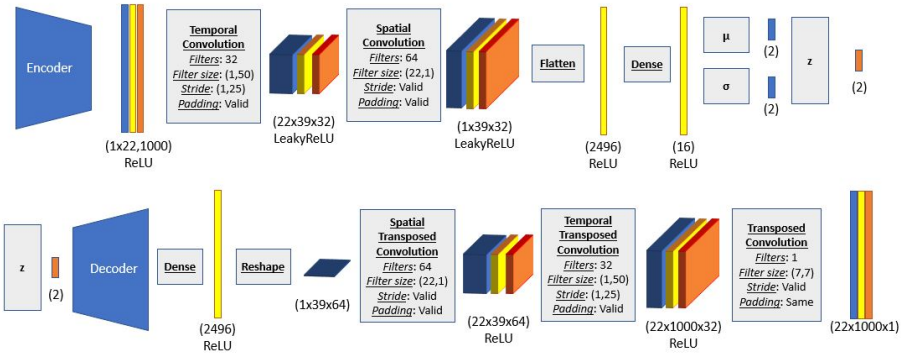

Figure 10. VAE encoder and decoder architecture.