Exploring Genetic Algorithm as an Image Synthesizer for Cases with Limited Training Samples

Sumeyye Sena Kiyma[1]; Rittika Shamsuddin[2]
[1]Bilkent University (Turkey), [2]Oklahoma State University

Introduction

Synthetic data is crucial in various fields. Currently, data synthesis is widely being used in pattern exploration in image data. In other fields, such as healthcare, data synthesis can be leveraged to either increase data volume and/or to improve privacy and data security. In healthcare field specifically, it can intrude privacy, and/or be expensive to collect data from the patients. This often results in lack of data for training state-of-the-art machine learning based programs [1]. Thus, synthetic data can play a huge role in healthcare-based applications, and yet, there is limited data-synthesis work done on healthcare non-image data where only limited data is available.

In this research we want to find a method for data synthesis, such that data can be generated from minimal data, with limited hyperparameter changes between different data types, and such that the user has control over the variation included in the synthetic data (often medical data allows only slight deviations, for example the ECG data that can be seen in Figure 1). As such, we focus on synthesizing data from a small dataset by using the Genetic Algorithm (GA). A genetic algorithm is based on Charles Darwin's theory of natural evolution. In this algorithm the fittest individuals survive and produce offspring of the next generation [4]. For this study, the synthetic data is chosen to be the best fit individual at the end of the process.

NSR Sorted Models		NSR		AFIB		PVC		LBB	
	(%) Test Acc	(%) Spec	(%) Sens	(%) Spec	(%) Sens	(%) Spec	(%) Sens	(%) Spec	(%) Sens
Model (ECG, Synthetic, 822, [No Reg, Reg1, Reg2])*	97.08	***97.78***	***86.67***	97.78	100	96.67	90.00	100	100
Model (ECG, Synthetic, 411, No Reg)	96.67	96.67	86.67	100	96.67	94.44	90.00	100	100
Model (ECG, Synthetic, 2056, No Reg)	97.08	**93.33**	**96.67**	100	93.33	98.89	86.67	100	100
Model (ECG, Real + Syn, 500, [No Reg, Reg1, Reg2])	96.25	100	73.33	100	96.67	91.11	100	98.89	100
Model (ECG, Real + Syn, 100, Reg2)	96.25	100	73.33	100	96.67	91.11	100	98.89	100
PVC Sorted Models		**NSR**		**AFIB**		**PVC**		**LBB**	
	(%) Test Acc	(%) Spec	(%) Sens	(%) Spec	(%) Sens	(%) Spec	(%) Sens	(%) Spec	(%) Sens
Model (ECG, Real,514)	95.00	98.89	70.00	100	90.00	97.78	100	90.00	100
Model (ECG, Synthetic, 2056, No Reg)	97.08	93.33	96.67	100	93.33	**98.89**	**86.67**	100	100
Model (ECG, Synthetic, 822, [No Reg, Reg1, Reg2])*	97.08	97.78	86.67	97.78	100	***96.67***	***90.00***	100	100
Model (ECG, Synthetic, 1234, No Reg)	95.00	95.56	76.67	97.78	96.67	96.67	86.67	96.67	100
Model (ECG, Real + Syn, 500, [No Reg, Reg1, Reg2])	96.25	100	73.33	100	96.67	91.11	100.00	98.89	100

Model with best balance of Metrics for both *NSR* and *PVC

Table 1. Top 5 ECG trained ResNet models sorted to show best balance on specificity and sensitivity for NSR & PVC classes [3].

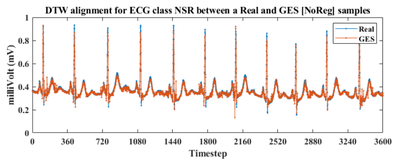

Figure 1. Overlay plot of ECG target class NSR that shows DTW alignment of real and GES synthetic data [3].

Trained Models	(%) Train - Test Acc	(%) Spec	(%) Sens
Model(EEG, Real, 93)	93.68 - 77.42	75.21	75.21
Model(EEG, Real+Syn, 186)	99.17 - 96.77	96.15	96.15

Table 2. Performance metrics: Models trained with real and GES synthetic data [3].

Trained Models	(%) Test Acc	(%) Spec	(%) Sens
EEG			
Model (EEG, Real + Syn, 160, No Reg)	**93.55**	**94.44**	**94.44**
Model (EEG, Real + Resampled, 160)	65.00	65.00	65.00
ECG			
Model (ECG, Real + Syn, 200, No Reg)	**92.50**	**95.00**	**85.00**
Model (ECG, Real + Resampled,200)	85.83	90.56	71.67
Model (ECG, Real + Syn, 100, No Reg)	95.00	96.67	90.00
Model (ECG, Real + Syn, 100, Reg1)	**96.25**	**97.50**	**92.50**
Model (ECG, Real + Syn, 100, Reg2)	**96.25**	**97.50**	**92.50**

Table 3. Classification performance for ResNet models trained with GES synthetic data with [No Reg, Reg1, Reg2] and models trained with data perturbation synthetic data [3].

Related Work

Guided Evolutionary Synthesizer (GES) has been used to synthesize time series healthcare data for Residual Network (ResNet) models and it is shown that data synthesized by GES worked better than perturbated data for training models (see Table 3). Table 1 and Table 2 also shows that how the ResNet results improved with synthetic data. However, this model was not used with images and was not compared to other synthesizers [3]. Virtual Patient Model (VPM) was also built for data synthesis in healthcare with genetic algorithm, yet it was not comparable to other synthesizers [1].

As of now, Generative Adversarial Networks (GANs) and its variations dominate image synthesis in the field of computer science. GANs are synthesizers that constitute two major parts: generator and discriminator. The generator synthesizes new data that is sent to the discriminator. The discriminator tries to discriminate between "real" and "generated" data by a classifier. By using back propagation, the generator tries to fool the discriminator [2]. This adversarial training between discriminator and generator makes it difficult to converge and the volume of the training data is immense. Our research is focused on synthesizing data with smaller datasets. On the other hand, the objective function of GANs must be differentiable for backpropagation which cannot be achieved in every study. The purpose of this study is finding a method for data synthesis with smaller datasets and where the objective function is not limited to be differentiable.

Methodology

Since we want to combine pattern exploration with minimal training and controlled variation, we begin with the simplest dataset e.g., MNIST dataset, to develop our model as an expert will not be needed to evaluate the synthesized data (unlike most healthcare data). Once the synthetic digits are satisfactory the next part will be based on CIFAR-10 dataset, then healthcare images and finally other types of healthcare data such as time series data. Another reason to use the MNIST dataset is to compare the synthesizer with other models that have been built previously.
So far, we have used 2 different methods based on genetic algorithm. You can see the pseudocode for these methods in Figure 2 and Figure 3. *Method 1* uses the genetic algorithm with one individual where the individual is considered as the "to be synthesized image". This individual is being mutated in each iteration and if the mutated individual is more fit than the previous one it is replaced to become the individual. This goes on for each iteration until a certain number of generations is reached.

In *Method 2* there is a population with more than one member, and the offspring are created by both mutation and crossover between the best fit parents. This method is more open to allow variation since *Method 1* is only pushing one member to become more fit whereas in *Method 2* the members of the population can crossover and mutate for variance.

The fitness value that is used for both methods is calculated by the objective function and indicates whether the synthetic image meets the criteria and is a part of the class that is to be generated by using distance metrics. The equations for the objective function and the fitness value can be seen below.

$$g = \sum_{i=1}^{k} d(x_{real_i} - x_{synthesized})$$

$$f = \frac{1}{g}$$

$g: objective\ function$
$d: distance\ function$
$f: fitness\ value$
$k: number\ elements\ in\ seed-data$
$x_{real_i}: i^{th}\ image\ in\ seed-data$
$x_{synthesized}: to\ be\ synthesized\ data$

```
Start with one individual
For each generation:
        Compute fitness of the individual
        Mutate the individual
        Compute fitness of the mutated individual
        Replace individual with mutated individual if mutation increased
        fitness
Stop when (max number of generations is reached)
```

Figure 2. Pseudocode of *Method 1*.

```
Start with n individuals as the population
For each generation:
        Compute fitness of the population
        Do crossover over the population
        Mutate the individuals
        Replace individuals with mutated individuals if mutation increased
        fitness
Stop when (max number of generations is reached)
```

Figure 3. Pseudocode of *Method 2*.

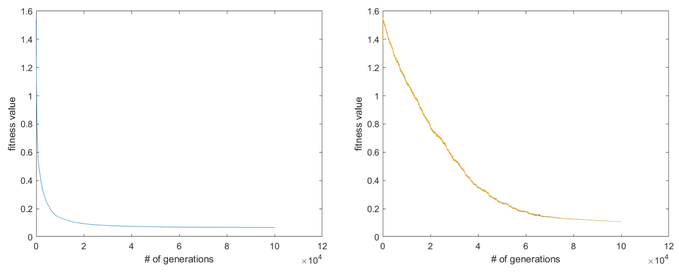

Figure 4. Convergence: Fitness value vs generation number for a) *Method 1*, *b*) *Method 2*.

Figure 5. Synthesized images with *Method 1*.

Figure 6. Synthesized images with *Method 2*.

Results

The synthetic images for each class for each method can be seen in Figure 5 and Figure 6. When these images are classified by a machine learning model it is seen that there is **80% true positive rate with *Method 1***, and **78.13% with *Method 2***. You can see that not all images have the same quality of matching the class, however, almost all of them represents their class from a human view. Also, it must not be forgotten that all these images for each class are generated from the same seven images from the seed-data. The main purpose of this research is to explore and determine whether GA is capable of pattern exploration while allowing researchers to control the amount of variations seen in final synthesized output. At this moment of the research, we do not have a comparison method with GAN, however, we do see some variations in our experimental output. We plan to continue our experiments.

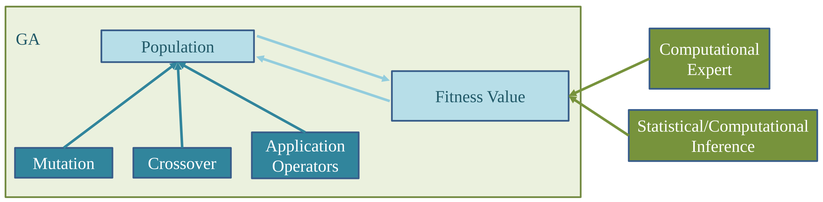

Figure 7. Future work diagram.

Conclusions & Future Work

The purpose of this research is to synthesize healthcare data from a small dataset for different machine learning models that would be used for training. This part of the research is based on digit images that are more common and can be analyzed easily without a healthcare expert. The synthetic images so far have been successful to an extend. However, we will keep experimenting for better results and when satisfied results are observed the research will proceed in healthcare data.

The experiments so far were based on *Method 1* and *Method 2*, and the results were as expected. The research will continue more dominantly with *Method 2;* however, a computational expert and statistical inferences will be combined with the objective function to get class specific results. With these extensions, which can be seen in Figure 7, it is believed that the quality of the images will be increased together with the variation.

References

[1] R. Shamsuddin, B. M. Maweu, M. Li and a. B. Prabhakaran, "Virtual Patient Model: An Approach for Generating," *IEEE International Conference on Healthcare Informatics*, pp. 208-218, 2018.

[2] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322-36333, 2019.

[3] B. M. Maweu, R. Shamsuddin, S. Dakshit and B. Prabhakaran, "Generating Healthcare Time Series Data for Improving Diagnostic Accuracy of Deep Neural Networks," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, vol. 70, pp. 1-15, 2021.

[4] S. Katoch, S. S. Chauhan and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, p. 8091–8126, 2021.