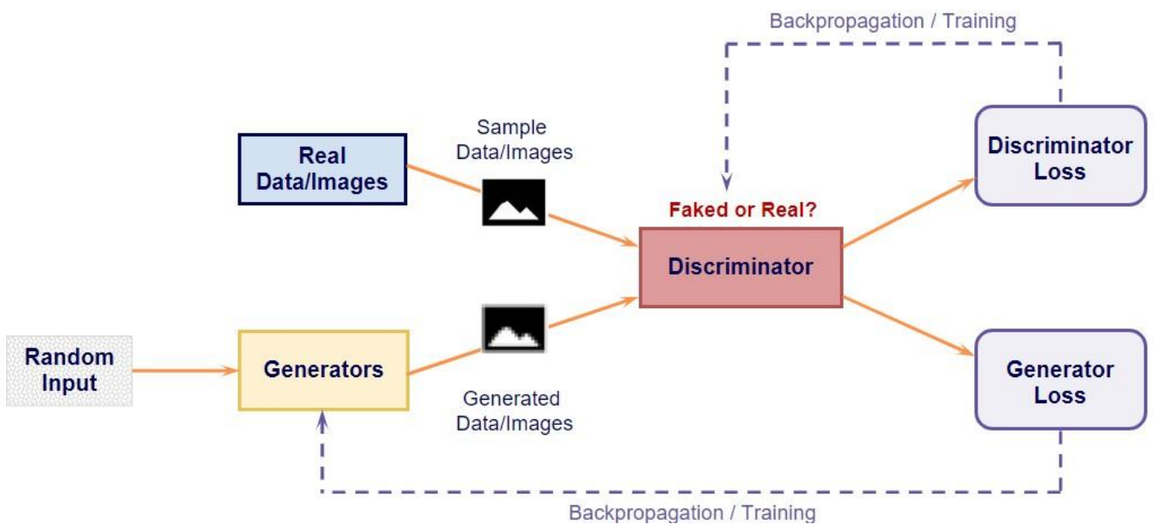
Improving Evolutionary Generative Adversarial Networks

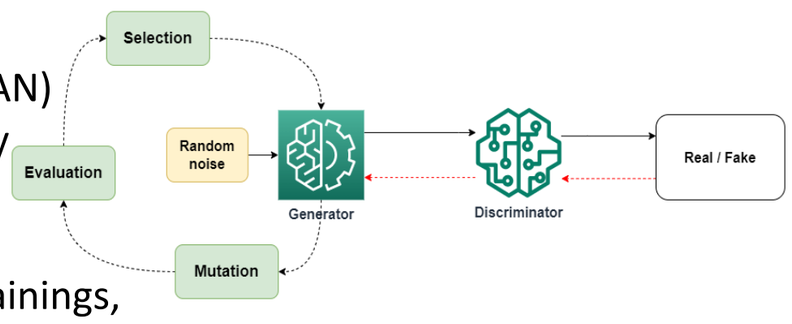
Generative adversarial network (GAN) is a powerful method to reproduce the distribution of a given data set. It is widely used for generating photo-realistic images or data collections that appear real. Evolutionary GAN (E-GAN) is one of state-of-the-art GAN variations. E-GAN combines population-based search and evolutionary operators from evolutionary algorithms with GAN to enhance diversity and search performance. In this study we aim to improve E-GAN by adding transfer learning and crossover which is a key evolutionary operator that is commonly used in evolutionary algorithms, but not in E-GAN.

Generative Adversarial Networks (GAN)

GAN is a prominent learning method that aims to learn the data distribution of given data sets. Currently, GAN is widely used in image generation, image transferring, image de-noising and other nominal data generation. GAN trains two neural networks in an adversarial matter, namely a generator and a discriminator. The generator takes randomized data from a uniform or normal distribution as its input, and outputs data of the required size. The discriminator takes input from both a training data set and the fake samples generated by the generator. It learns to classify each of these input as true of false.



Evolutionary GAN



Evolutionary GAN (E-GAN) Introduces evolutionary approach into GAN to mitigate two problems Encountered in GAN trainings, vanishing gradient and mode collapse. It achieves this goal by Adding mutation, evaluation and selection operators from evolutionary algorithms to GAN training. Conventional GAN updates its generator and discriminator using a single loss function for each. E-GAN provides three different loss functions for its generator, each of them is considered as one mutation.

In the evaluation step, a fitness score That evaluates both the quality and diversity of offspring is calculated. The quality factor of fitness function guarantees the quality of generated data and prevents gradients from diminishing. On the other hand, diversity factor ensures the offspring generating diversified data which help mitigate mode collapse. E-GAN follows survival-of-the-fittest rule, picking the offspring With the highest fitness score in the selection step and pass it to become new parents in next iteration.

$$\mathcal{M}_G^{minimax} = \frac{1}{2}\mathbb{E}_{z\sim p_z}[log(1 - D(G(z))]$$

$$\mathcal{M}_G^{heuristic} = -\frac{1}{2}\mathbb{E}_{z\sim p_z}[log(D(G(z))]$$

$$\mathcal{M}_G^{least-square} = \mathbb{E}_{z\sim p_z}[(D(G(z) - 1)^2]$$

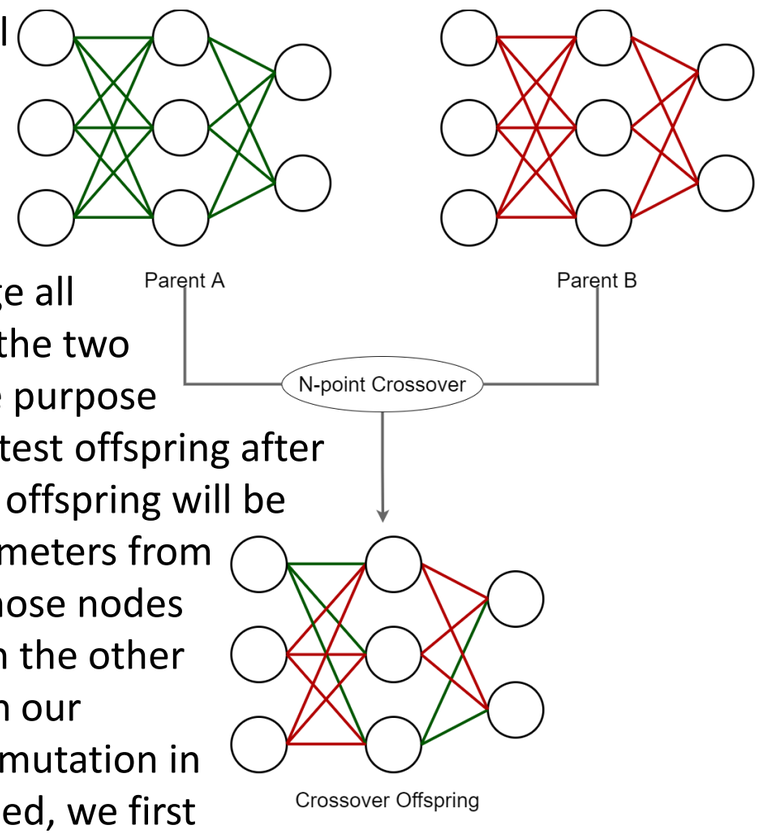
$$\mathcal{F}_q = \mathbb{E}_z[D(G(z))]$$

$$\mathcal{F}_d = -log||\nabla_D - \mathbb{E}_x[logD(x)] - \mathbb{E}_z[log(1 - D(G(z)]||$$

$$\mathcal{F} = \mathcal{F}_q + \gamma \times \mathcal{F}_d$$

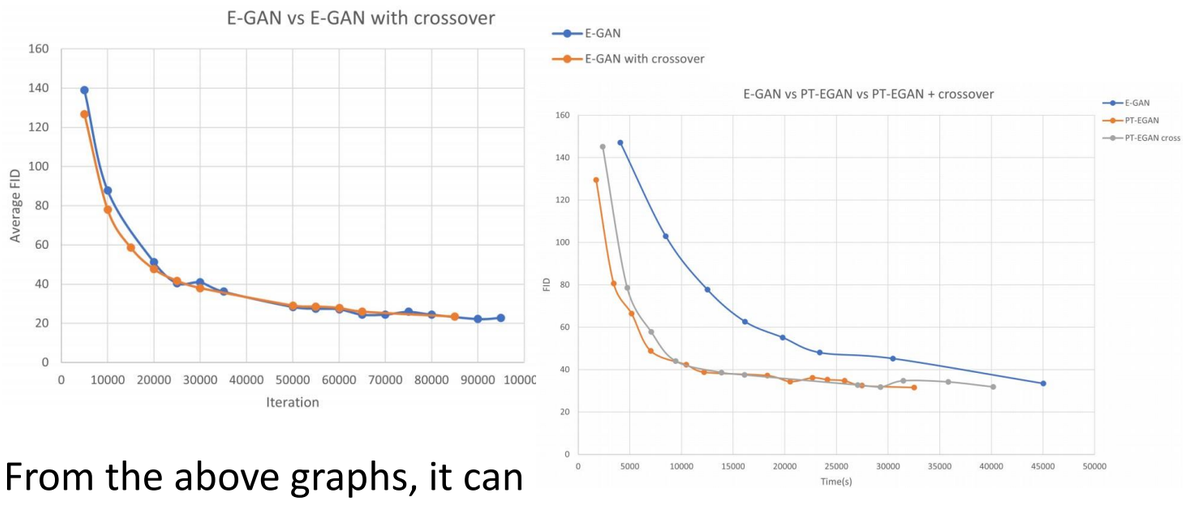
E-GAN with n-point crossover

In this study we propose adding n-point crossover into E-GAN. The idea of n-point crossover is straightforward. Firstly, two fittest offspring from the population will be selected as parents for this crossover operator. The selection method is by computing the fitness scores of all individuals in the population, and rank them accordingly. After two parents are selected, a given number of nodes in the neural network architectures will be randomly selected, i.e. n points will be selected, where n is an even number to enable pairing. For each pair of selected nodes, this method will exchange all nodes between them in the two parent networks. For the purpose of producing only one fittest offspring after crossover operation, the offspring will be initialized using the parameters from the fitter parent. Then those nodes are to be exchanged with the other parent which is less fit. In our proposed method, after mutation in each iteration is performed, we first evaluate all the mutated offspring by computing their fitness values. Next, two fittest offspring will be retained, and they will be parents of the crossover operator. After crossover, one crossover offspring will be produced, and its fitness score is evaluated. If its fitness score exceeds at least one of its parents, we replace the inferior parent with this crossover offspring. At the end of each iteration, we always select the fittest two offspring to proceed into next iteration.



Experiments & Results

We conducted experiments on original E-GAN and PT-EGAN. PT-EGAN is one of our previous work, it variates E-GAN for less training time.



From the above graphs, it can be seen that adding n-point crossover to E-GAN and PT-EGAN did not significantly improving the performance. However, the select rate on crossover offspring is not low at all. From diagrams on the right, the crossover is not often selected at the very early stages. But as the training progresses, the select rate dramatically increases until finally reached around 0.48. This implies crossover offspring have very high probability being superior to other mutation offspring in later stage of the training. This do prove that crossover is having positive influence to the performance of E-GAN, nevertheless, from the evaluation results, this improvement is insignificant.

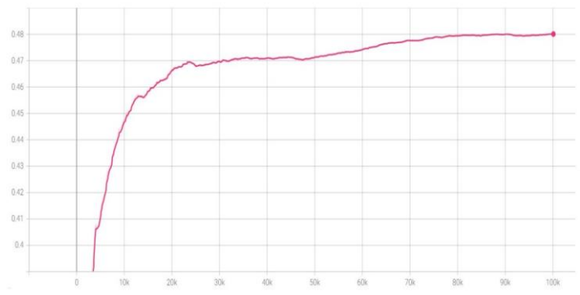


Fig. 5: Crossover Offspring Select Rate during one training

Our research still digs out some valuable findings from these results. Firstly, we found exchanging more nodes would have destructive effects to the performance. The intuition is simple. Exchanging more nodes in the fitter parent from the inferior parent will make the crossover offspring more like the inferior parent. Therefore, the final performance of exchanging too much nodes is not that promising. Secondly, we found exchanging nodes in the high layer or low layer brings better performance than exchanging nodes in the middle layers.

Conclusion

In this study, we aim to improve the performance of E-GAN by adding crossover operator. The crossover operator we experimented with was n-point crossover which simply exchanges an arbitrary number of nodes in parent networks. Although there are some researches supporting that adding n-point crossover can be constructive to building the neural networks, our experiment results did not show much of significance.