University of Tartu
Algorithmics MTAT.03.238
Mykyta Luzan
Oleksandr Syzonov

Snowflake generation with help of genetic algorithms

Introduction

Genetic algorithms is a class of algorithms, for which we have vectors representing genomes that optimize some function or heuristic. There are different implementations and versions of genetic algorithms, but they have in common population evalution using some fitness function, some sort of selection, reprduction and mutations.

The goal of this project was to apply genetic algorithms to control the process of generating snowflakes. Due to the fact we can vary a range of the snowflake's features then we can use them as a genome in our evolutionary approach to find the best characteristics that represents the most appealing snowflake.

Snowflake generation

The typical snowflake consists of 12 repetitive pieces. If we fold paper to small triangle with the inner angel 30 degrees and perform some cutouts on its sides then we receive a symmetric image of snowflake that has 6 rotated copies and 6 mirrored ones.

We used two types of cutouts: the top cutout which shapes the exterior of the snowflake, and sides cutouts that creates inner patterns of the snowflake. The generation of each snowflake happens randomly, however, we can control and limit this randomness.

To do that we introduces 5 features that represents min and max length, depth, area of cutouts, margins between cutouts and randomness factor of top cutout. We wrapped all these features to a single vector (genome). The values of this vector vary from 0 to 7, and usually smaller values correspond to simpler forms and higher values to more sophisticated patterns. Also we fixed the random seed to have reproducible results, so only genome influence the appearance of snowflakes.

Also one of the main problems was to validate cutouts. We do not want to have intersecting cutouts to have more clear image. For this reason, during the generation phase we perform cuts only if they do not intersect with others.

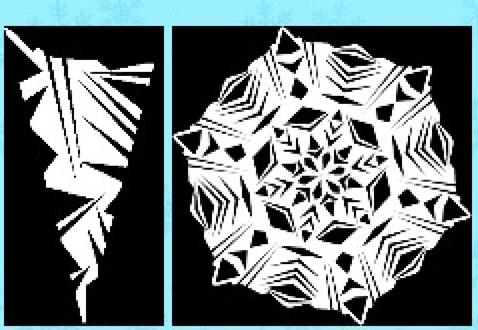

Figure 1. An example of the generated piece and the full snowflake

The generation pipeline consists of the following steps:
1) Create the base empty triangle
2) Set randomness limits according to genome
3) Perform the top cutout
4) Perform inner cutouts on each side of the snowflake
5) Draw a single piece of the snowflake
6) Apply rotations and mirroring to get the full snowflake

Genetic algorithm

Genetic algorithms is a class of algorithms for optimization of some metric called fitness function for a generic task that is inspired by biologically-inspired imitation of evolution, in which instead of organisms we have solutions to the task. Genetic algorithms can be used for a wide array of tasks: from a general function optimization for finding optimal parameters in cases when we cannot find gradient and do gradient descent (or it gets stuck in local maximums or minimums) to various planning, financial and physical tasks. We chose genetic algorithms as our method of optimization, because it's good enough at finding, at least approximately global maximum or minimum instead of being stuck in local extrema, while being more directed in it's search than random search.

The genetic algorithms works as follows:
Some random initial population is generated
According to some rule, some fraction of population is selected
It reproduces and some genomes are randomly mutated
The population is evaluated again and we go to the step 2.

In our case, selection is based on a fitness function, which will be described in detail below. We select the top 40% percent of the population by fitness and then they reproduce using crossover.

Crossover is a process of random merging of 2 genomes to produce an offspring. In our case, genomes are merged as follows: some random cutoff point is chosen and then all genome segments of a new organism before that point are taken from the first parent and all genome after it are taken from the second parent. This way, some optimal solution has a higher probability of being achieved. This way, we have a non-deterministic way to produce an offspring with a wide array of possible offsprings from the same 2 parents.

After reproducing, our new population is subject to mutation. This way, we have not only convergence to some more optimal solution, which is provided by selection based on fitness function and crossover of best individuals, but also an element of random search. In our case, mutations are done with probability of 40%. The mutation is just adding some vector sampled from a normal distribution to our genome. The scale of mutation is 10% of the scale of population to make mutations more significant in terms of aesthetic changes.

Fitness function

Our goal in terms of fitness is to evaluate the aesthetic properties of any given snowflake. It cannot be by some simple algorithm, because aesthetics is highly subjective, provided that our snowflakes all have similar format (there are no differences in image quality). So, we decided to use an AI model to do this, because manually evaluating them would be highly tedious and ineffective, while CNN model could approximate this reasonably well, if trained on a dataset of good quality.

The model of choice is the model from the "NIMA: Neural Image Assessment" [2] paper of Google. It is a Convolutional Neural Network that is trained on a large-scale dataset for aesthetic analysis and predicts the aesthetic of the picture on a scale of 1 to 10. Based on its predictions, we evaluated our snowflake images and did selection and epochs.

As we said before, the process of evaluation the appearance of the snowflakes is pretty subjective, therefore the assessment of our final model can be questionable. So for future experiments we can try different approaches for the fitness function, maybe create comparison with real-world snowflakes images. The example of epochs you can see on the figure 3.

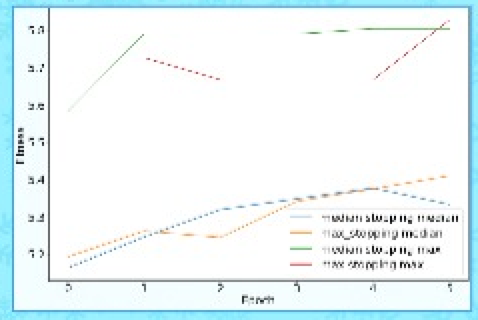

Figure 2. Fitness function values on different epochs.

References

1. Процедурная генерация бумажных снежинок. Habr. Enfriz. Published 01.12.2021. Link: https://habr.com/ru/post/592659/
2. Talebi, Hossein, and Peyman Milanfar. "NIMA: Neural image assessment." IEEE Transactions on Image Processing 27.8 (2018): 3998-4011.
3. Github repository link: https://github.com/Arattel/Genetic_algorithms_snowflake_generation

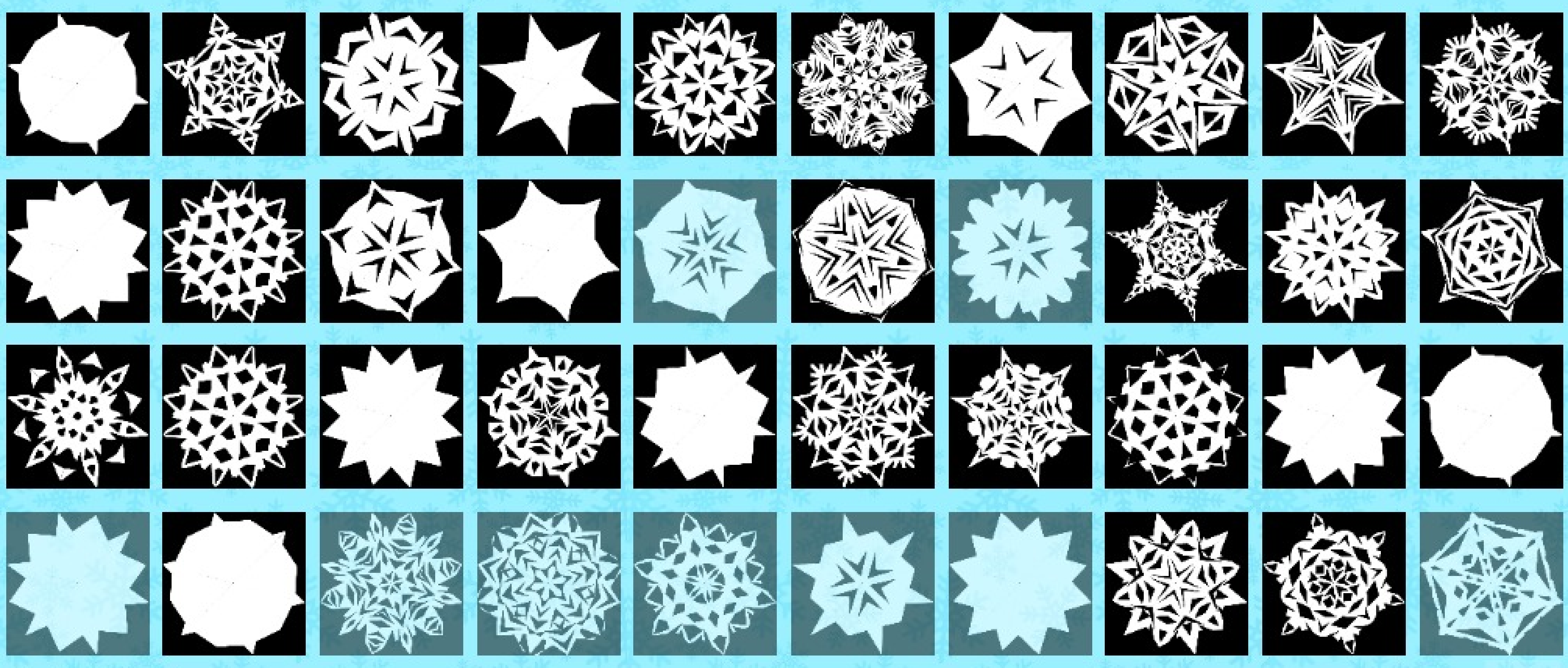

Figure 3. Samples of some of the epochs