**Growing Neural Gas**

**Project 2**

**Timotius Andrean Patrick Lagaunne**

* 1. Introduction……………………………………………………………………….…....….2
  2. Method…………………………………………………………………………………….2
  3. Experiment and Parameters……………………………………………….………………6
  4. Results……………………………………………………………….…………………….7
  5. Reflection and Conclusion.…………………………………………………...………….12

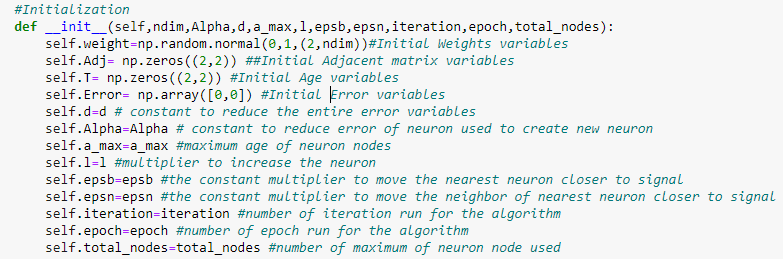
**Introduction**

Growing Neural Gas is an unsupervised machine learning method that is improvisation of Neural Gas that capable in adjusting in the number of neurons automatically by adding or deleting neuron according the state of neural gas. It is named Neural gas because during adaptation process, it behaves and distributed like gas within data space.

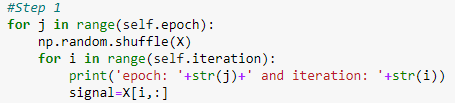
**Method**

In the original paper of Growing Neural Gas, the algorithm is consisting of 11 steps from step 0 to step 10. Below are the all description of the steps from the paper with my code and some clear description adjustment for each step.

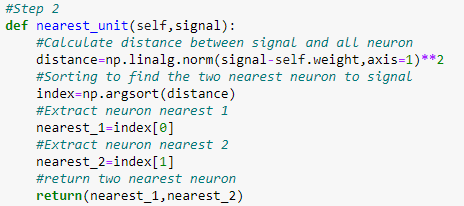
1. Initialized all variables and also initialize two unit of neurons with random weights in Rn



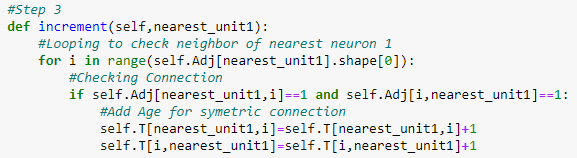
1. Generate an input signal, β according the distribution of P(β) or in this project the input signal generated from randomized data.



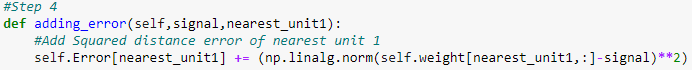
1. Find two nearest unit of neuron from signal which we label s1 (the nearest 1) and s2 (the nearest 2)



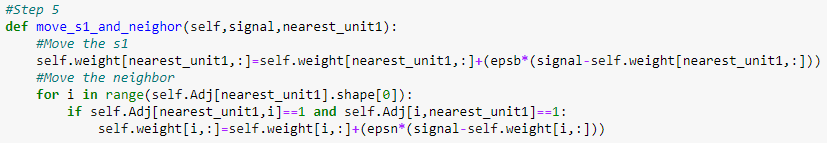
1. Increase the age of all edges connected to unit neuron s1



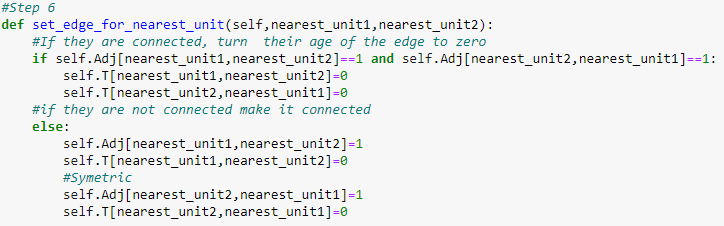
1. Add the square distance between s1 and signal to error variable s1



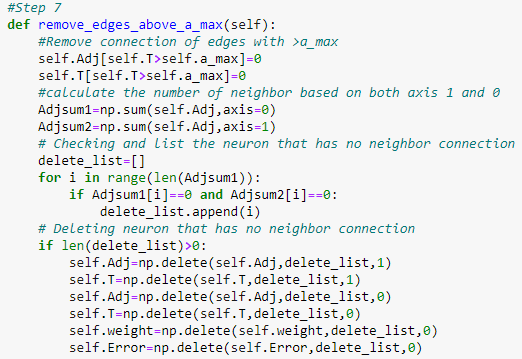
1. Move s1 closer to signal with ϵb multiplier and move the unit n connected with s1 closer to signal with ϵn.



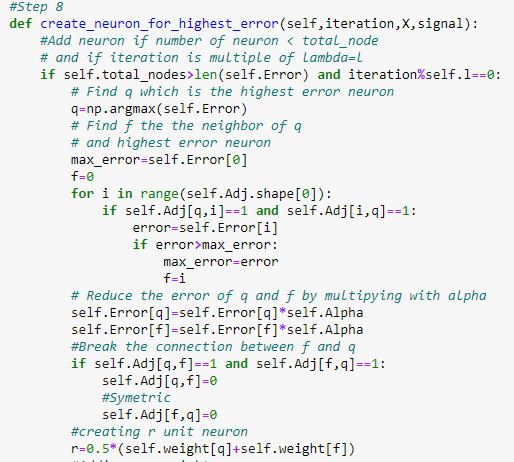
1. If s1 and s2 connected make the age of connection to zero. If they are not connected, make it connected.

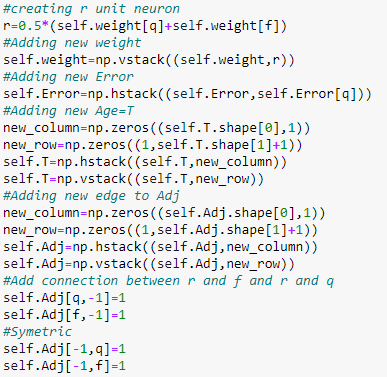


1. Remove the edges with an age more than amax. if it resulting to a point with no connection to other unit neuron, remove the unit neuron.

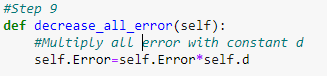


1. If iteration of input signal generated are multiple of lamda, generate new neuron based on neuron q which is neuron with highest error and neuron f which is connected to q and the highest error as well. I also add maximum node in this step to limit the number of neurons
   1. wr= 0.5\*(wq+wf)
   2. Error of q and f multiplied with α
   3. Error of r is the copy of new error of q





1. Decrease all error by multiply of d.



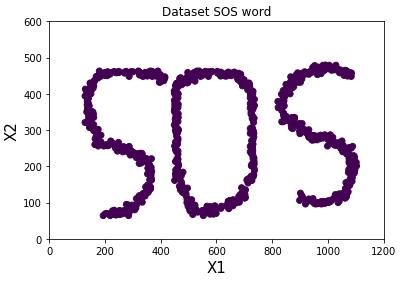
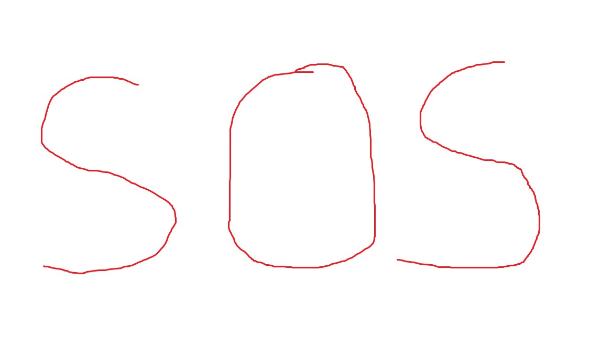
1. Stop criterion based on network size or some performance measure (I decide to use iteration and epoch as stop criterion), if stop criterion is not met come back to step 1.



In simple term, the method works by moving the nearest neuron and its neighbor based on signal input. The neuron is being added based on the multiplier of the iteration and the neuron being deleted based on the age of edges of the neuron. The age regularly increases based on the nearest neuron and the neighbor. When the age size of the edge higher than maximum age, the edge is turn off and when the neuron has no edges in result of deleting the edge, the neuron got deleted. The result of the movement, addition and deletion of neuron and its edges give the pattern of neural gas.

**Experiment and Parameters**

As part of experiment, I use dataset of SOS word created by paint. To analyze my growing neural gas algorithm, there are a lot of parameters to be explore. There is number of node of neurons, ϵb and ϵn as the parameter to move the neuron closer to the signal, alpha and d as the parameter to reduce the error of the neurons, amax as the highest age for the node, l as the multiplier for adding node of neuron and epoch and iteration to iterate the data and repeat iterating the data.



*Figure 3.1-2 The visualization of Dataset SOS word from paint before processing and after processing*

With that abundant parameter to be explore, I decide for the experiments to do four double variable variation and one single variable variation and visualize the result in 2D. Below are the list of the double variation and one single variable variation to be explore

1. Epoch and Iteration variation

* Epoch (1, 5, 10)
* Iteration (100, 500, 1000)

Maximum Node of neuron individual variation

* Maximum Node of neurons (100, 250, 500)

1. ϵb and ϵn variation

* ϵb (0.2, 0.5, 1)
* ϵn (0.005, 0.1, 0.5)

1. α and d variation

* α (0.1, 0.25, 0.5)
* d (0.3, 0.5, 0.995)

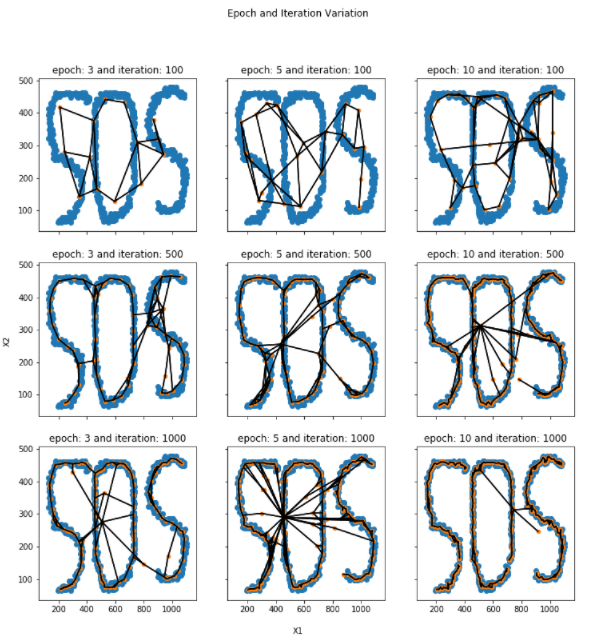
1. amax and l multiplier variation

* amax (10, 25, 50)
* L multiplier (10, 25, 50)

In double variable variation, two variables are being varied while the rest variables I use the middle value variable. Similarly, in single variable variation, one variable is being varied while the rest I use the middle value variable. As mentioned above, the result will be visualized in 2D dataset. It includes the neurons, the edges and the data visualized in 2D.

**Results**

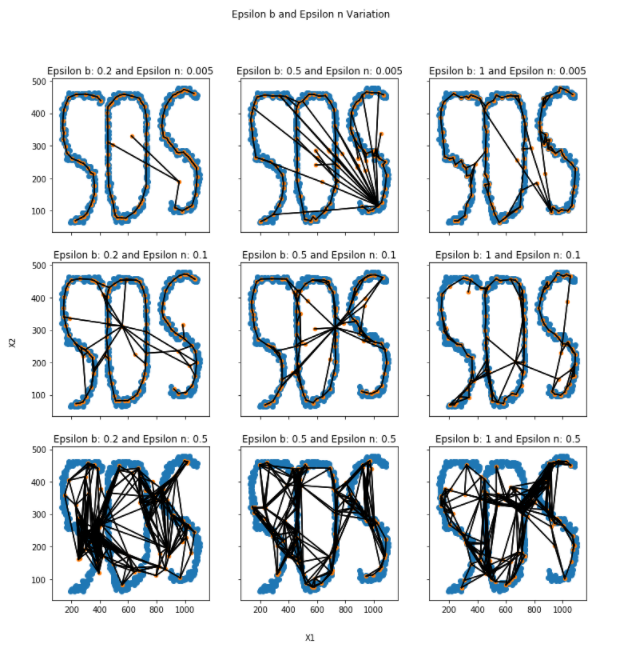
**Epoch and Iteration variation**



*Figure 4.1. Epoch and Iteration Variation Visualization Result for Epoch 3,5, and 10 and Iteration 100, 500, and 1000*

In this variation, we can see that low epoch and iteration number show low performance. The best result, which is epoch number 10 and iteration number 1000 show almost clean cut between each letter of SOS. Other epoch number especially epoch with low iteration does not show clean separation to cluster the letter. The rest, cluster the letter but has edges that clutter the letters. Finally, it makes sense that the result with high epoch and iteration show best result. High epoch and iteration help with the convergence of the result in terms of addition and reduction of neurons. There is possibility that higher epoch than 10 will give better result but there is also possibility that it might not improve because other parameters limit the performance.

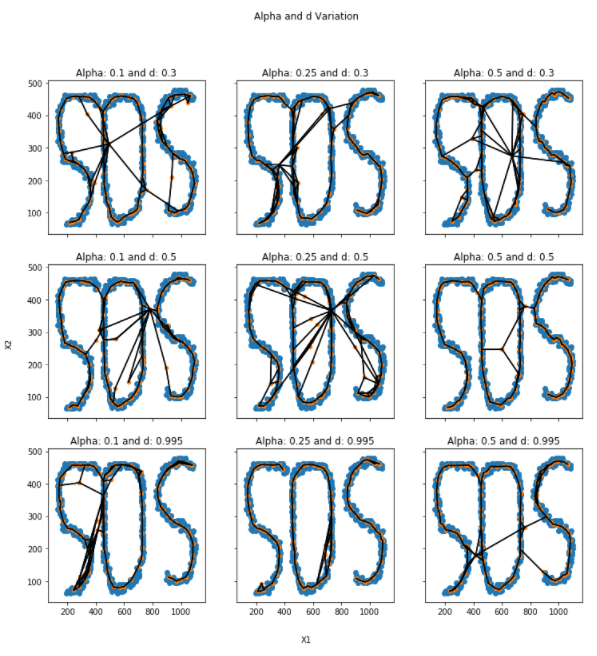
**ϵb and ϵn variation**



*Figure 4.2. ϵb and ϵn Variation Visualization Result for ϵb 0.2, 0.5, and 1 and ϵn 0.005, 0.1 and 0.5*

ϵb is a constant that is used to multiply the movement of the nearest neuron to the signal that is inputted to neural gas and ϵn is a constant that is use to multiply the movement of the neighbor of the nearest neuron to the signal. Figure 4.3, show the result of varying the ϵb and ϵn. Low ϵb and ϵn give the best result with all neuron nodes closely cluster the SOS letter. There are only 3 neuron nodes outside of SOS letter and has edges outside of SOS. On the other hand, both low and Medium ϵn with variation of ϵb show moderate result with neuron nodes mostly cluster the SOS letters with some extra neuron nodes outside of SOS letters and edges that is outside of SOS letters. High ϵn on the other hand show high clutter of neuron and edges for all ϵb. This problem probably shows because the ϵn is bigger than ϵb. Having ϵn bigger than ϵb might make the neighbor of nearest neuron move faster closer to the signal than the nearest neuron itself to the signal. Too much movement from the neighbor might move them outside of the letter and increase the edges.

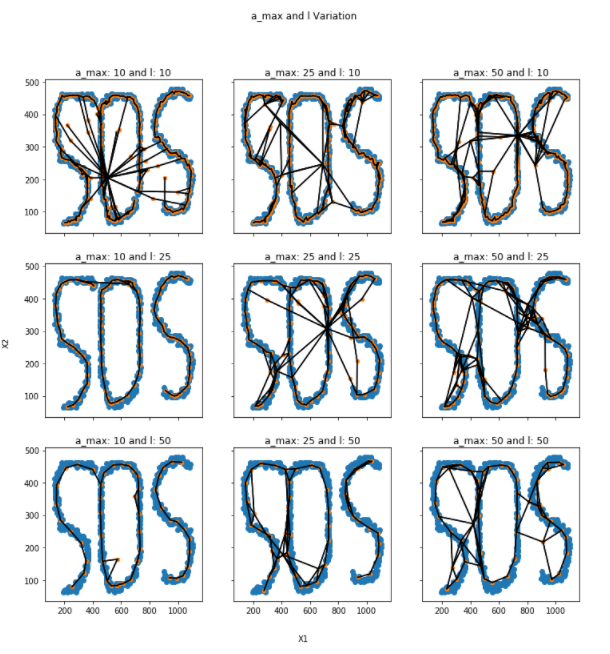
**α and d variation**



*Figure 4.3. Alpha and d Variation Visualization Result for Alpha 0.1, 0.25, and 0.5 and d 0.3, 0.5 and 0.995*

Alpha and d are the variables used to reduce the error for every iteration. Alpha, specifically reduce the error during the addition of new neuron and d, on the other hand, is reducing the error for all neurons at the end of every iteration. As we vary these two variables, we able to obtained best result when alpha = 0.25 and d = 0.995. The best result able to cluster the SOS letter with just little bit of clutter of edges on the O letter. Oppositely, with other alpha and d, we observe cluster on SOS as well, however, it has more clutter of edges that connect a letter to other letter. Medium alpha (0.25) and High d (0.995) perform the best might be because the parameters allowed the next iteration to pick different neuron node to be chosen for adding new neuron between the highest error neuron. Addition of neuron are based on the highest error which is why low alpha and high d reduce the chance of the neuron in previous iteration being picked again

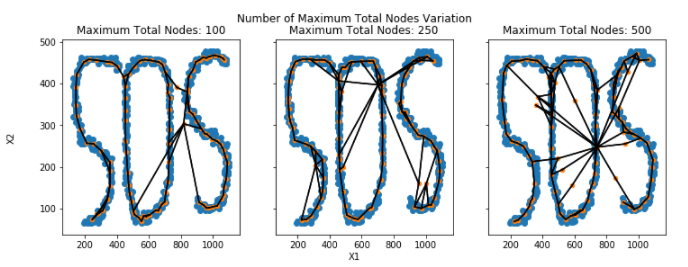
**amax and l multiplier variation**



*Figure 4.4. amax and l Variation Visualization Result for amax 10, 25, and 50 and l 10, 25 and 50*

Variable amax is a constant to limit the highest age we can get for edges of neurons. Oppositely, l is a constant multiplier that determine when we should add a new neuron. Varying both variables produced result seen in figure 4.5. There are two result that can be considered best result. The first one is *amax =* 10 and l = 50 and second one is *amax =* 10 and l = 25. Both result almost cluster the SOS perfectly. The rest of result, cluster the SOS fine but it has a lot of clutter and edges connecting different letter. Both best result pick *amax =* 10 to produce the best result, Though, *amax =* 10 and l=10 produce result that has a lot of edges that clutter and connect different letter. This probably happen because l=10 produce extra neuron faster than deletion coming from *amax =* 10. l = 25 and l=50 tend to be slower in producing new neuron which end up giving good result.

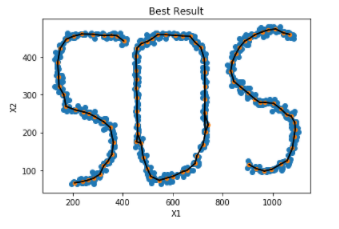
**Maximum Node of neuron individual variation**



*Figure 4.5. Maximum node of neurons Variation Visualization Result for 100, 250, 500*

Maximum total nodes are number of maximum neurons use for the neural gas. It limits the number of neurons that can be use. By varying it with 100,250 and 500, we can see that the result for low maximum total node of neuron 100 is the best result among the rest. In all cases the nodes cluster the SOS letter pretty well but as the maximum total node increase there are more clutter of edges not following the SOS letter. Also, there are some neuron nodes in both total nodes = 250 and total nodes = 500 that are not in the SOS letters which is why the clutter of edges exist more in both variations. Other than that, with high number of nodes, increase of iteration and epoch might be needed to reduce the clutter.

**Possible Best Result**



*Figure 4.6. Best result based on observation from the variation result. The parameter are Iteration=1000, epoch=10, amax =10, l=50, α=0.25, d=0.995, ϵb =0.2 ,ϵn =0.005, and max total nodes=100*

Figure 4.6 show the best result based on the best parameter from each variation. It completely clusters the SOS letter without any clutter of edges and the edges that is connecting one letter to another letter.

**Conclusion and Reflections**

By varying the parameter with single parameter variation or double parameter variation, I able to observe the result of growing neural gas algorithm on 2d dataset, SOS dataset, I created in paint app. Based on observing the result from the parameter variation, I can conclude that each parameter has part in making the result better. In the first experiment, both high Epoch and Iteration seems to be good for the neural gas. In the second experiment ϵb and ϵn also has interesting effect on the result. If ϵn higher than ϵb, the result of neural gas seems to be worse in the dataset. Therefore, using lower value ϵn and medium value ϵb, is best. In the third experiment, medium alpha,0.25, and high d, 0.995, seems to give better result compare to others parameter. The reason might be because medium value or low value alpha and high value of d will allow the new addition of neuron to be from different part of the structure. For the fourth experiment, amax and l, low value of amax and medium or high seems to perform the best. It is possible that low amax help deleting the not necessary edges and good value of l can help balance the deletion by adding the neuron. For the last experiment, the single parameter variation of maximum node of neuron, in original paper it is not included but I decide to include it to see the effect of limiting the neuron. The result show that low maximum of node performs the best. Finally, double parameter variation and single parameter variation probably not enough to understand the performance of growing neural gas. All parameters has part in making it better and worse though there is some general trend but other variable might affect it by limiting the performance or making it worse. So as part of reflection, I find growing neural gas to be nice because it does not require label but it has his own problem with the parameter optimization. Future work might involve analyzing more variation and comparing growing neural gas with other unsupervised machine learning algorithm or combining the algorithm with supervised machine learning algorithm .