

COMP305 First Semester Examination 2020/21  
Biocomputation

ID: 201277571

01/02/2020

# 1 History and Concepts

## 1(a)

Genetic Algorithms (GA) and Artificial Neural Networks (ANN) are used to solve a variety of computer science problems. Initially based on biological phenomena, such as evolution and neurons, the techniques are now applied to problems spanning computer science, engineering, economics and more. From prediction [1], to optimisation [2], these biology-inspired theories are represented algorithmically to solve ground breaking computer science problems. These problems can be biology related (e.g predicting cancer survival rates [1]), or completely removed from biology (e.g spacecraft antenna development [2]). The prolific use of these theories within Computer Science and in situations completely removed from their original biological context, warrants the inclusion of these theories as part of Computer Science.

## 1(b)

ANN can be used to solve problems in a range of disciplines including optimisation, prediction, classification, and pattern recognition. The McCulloch-Pitts neuron was the first proposed computational model of a neuron and first piece of work to be recognised as AI [3]. Many important developments in AI were achieved using ANN as a foundation.

An example of the use of ANN to classify individual handwritten digits can be shown on the MNIST dataset [4]. A sample of the 28\*28 pixel, greyscale dataset can be seen in figure 1

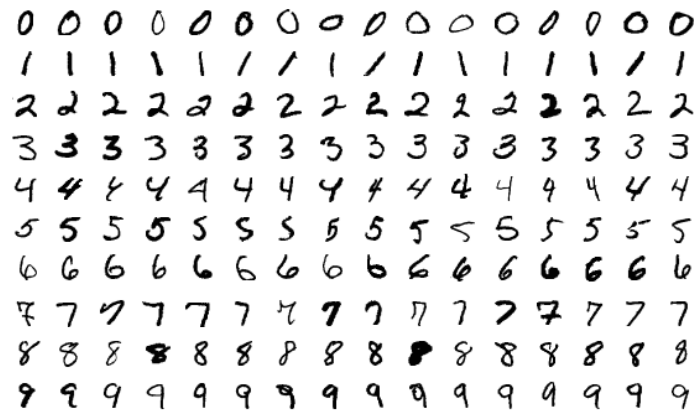


Figure 1: A small sample of the MNIST dataset

As the images are in greyscale, each pixel in the image can be represented as a numeric greyscale value in an array. These arrays can then form the input to a multi-layered perceptron. Depending on the specific architecture used, the model can “learn” what each digit looks like and accurately classify digits it has not seen before. <sup>1</sup>

<sup>1</sup>This is an extreme oversimplification of the complexity of multilayer perceptrons, even in a seemingly simple application such as this.

Character recognition such as this has a wide variety of applications, from automatically reading numbers on forms to verifying signature authenticity.

### 1(c)

Genetic algorithms are used to solve optimisation problems spanning a range of disciplines. Inspired by the process of natural selection, they form a subclass of the wider class of evolutionary algorithms. A population of “chromosomes” represents a population of candidate solutions to a given optimisation problem. A selection of genetic operators (including mutation, crossover and inversion) are applied to the chromomes and mimic the process that occurs during natural selection.

An example that shows the connection between Genetic Algorithms and their origin in natural selection, can be seen in the following paper [5] and figure 2 below.

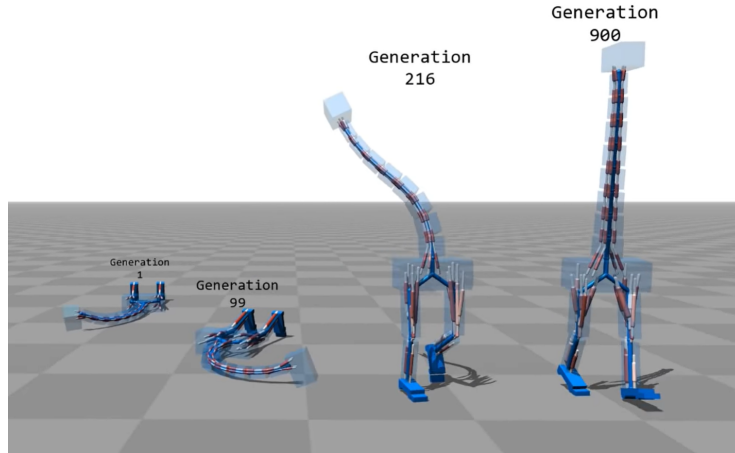


Figure 2: Video still, taken from the video at [6]

In this paper, various optimisation algorithms are used. Most notably in this instance, is the use of the Covariance Matrix Adaptation [7], an evolutionary algorithm. Figure 2 shows how, over the course of many generations, the chromosomes <sup>2</sup> in a population are optimised towards an objective.

Other applications of genetic algorithms include spacecraft antenna construction [2], travelling salesman problem [8] and boundless others. Despite the wide array of problem disciplines, there are some constraints as to which problems are suitable. For example, problems must have non-differentiable fitness functions and be able to be coded into a string of characters so the genetic operators can be applied.

### 1(d)

- i. The OCR system is solving a pattern recognition problem. If the target outputs are listed as classes, it can be seen as a classification problem.

<sup>2</sup>In this case, the choromosomes in question are the values input into the simulated muscles

At it's simplest form, this can be seen in the many models trained on the MNIST dataset, mentioned in section 1(b), where inputs are classed as a digit between 0 and 9.

- ii. There are two main types of OCR algorithm which both take different inputs.

Matrix matching aims to classify the shapes it identifies in the document as individual characters. It does this using an array value of the shapes and classifying this as a character, based on it's training data - much like MNIST. Any digit from figure 1 could be an example input, or any individual character on this page. Individual characters such as these are the ideal input however the system could also pick up any other shapes in the document.<sup>3</sup>

Feature Extraction is another type of OCR algorithm that simplifies the document to a series of features; mainly lines, curves, and intersections. This has many benefits including: the dimensionality of the input is reduced, making the process more efficient; the software can more accurately a variety of font and handwriting styles. Various AI methods can be used to compare the features to those the model is trained on.

- iii. There is an intended character/ string for each character/ string being read by the OCR into the system. The training and test data is all labeled. Therefore, this is supervised learning.

However, there have been recent developments in the area of unsupervised text recognition [9]. This paper from the Visual Geometry Group at the University of Oxford proposes a method of recognising text in images without labelled training data. This combats the time and energy intensive process of manually labelling training data.

```

GT  : takes place but complaints of a very anomalous
PRED : takes place but complaints of a very anomalous
takes place, but complaints of a very anomalous

GT  : outer surface of the pia mater was smeared
PRED : the outer surface of the pia mater wass mheard
The outer surface of the pia mater was smeared

GT  : pihealis quite destroyed j nothing feittaining bu
PRED : pimealis quite destroyed u nothing remaining but e
pinealis quite destroyed; nothing remaining, but

GT  : of the extremities become inflamed painful and
PRED : of the extremities become inflamed painfal and
of the extremities become inflamed, painful, and

GT  : tions accompany the other complaints which dis
PRED : tions accompany the other complaints which dis
tions, accompany the other complaints, which dis

```

Figure 3: Example results of the model proposed in [9]

<sup>3</sup>In order to limit the possibility of this occurring, the document will likely undergo pre-processing. Noise can be removed, pages can be aligned and the contrast can be increased to make classification easier.

As shown in figure 3, there were varying levels of success. Despite this, this is an incredible advancement in text recognition and shows that unsupervised learning may be the future of text recognition. Or a worthy competitor at the very least.

- iv. Ignoring the possibility of unsupervised learning, mentioned previously, the network would need to be trained using labelled data. This would likely be in the form of images of characters (handwritten or typed) with their correct label. The NIST dataset (a handwritten character dataset from which MNIST is found) is a good example. Data could also include previously translated and validated documents, any image of text which has an associated translation/ label could be used to train the network. The ability for the network to classify characters depends heavily on the training data used. For example, a network trained on typed fonts would likely struggle when presented with handwritten characters.

## 2 The McCulloch-Pitts neuron

### 2(a)

See figure 4, below.

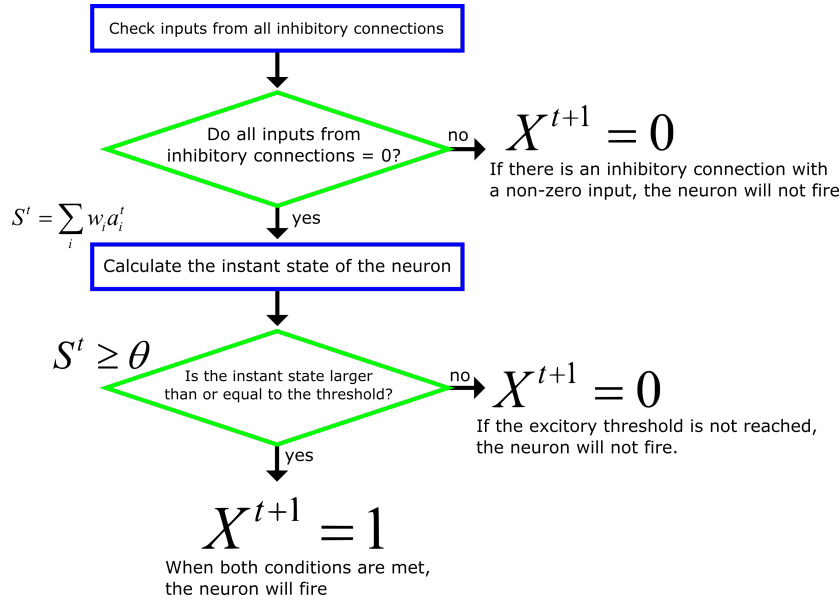


Figure 4: A flow chart diagram of an MP neuron where all weights of connections and the neuron threshold are set up in advance.

### 2(b)

Represented in boolean algebra as  $a_1 \vee a_2$ , and table form below, the logical OR gate can be represented as an MP-neuron as in figure 5.

$a_1$	$a_2$	"OR"
1	1	1
1	0	1
0	1	1
0	0	0

This table shows all possible outputs of the OR gate depending on all possible input values of  $a_1$  and  $a_2$ . If either of  $a_1$  and  $a_2$  is "true" (has an input of 1), then the neuron will fire ( $X = 1$ ).

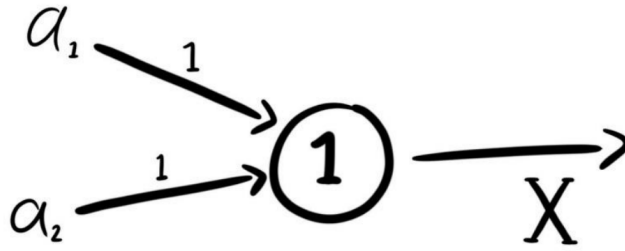


Figure 5: A diagram of an MP-neuron realisation of an "OR" logical gate

For this OR gate, the excitation threshold ( $\theta$ ) must be  $0 < \theta \leq 1$ . The value of 1 is chosen in figure 5, but any value within the given range would be sufficient. This inequality can be proven to be correct by analysing the output for each possible value of  $a_i$  (assuming the weights remain at 1).

The following values of  $a_i$  should fire the neuron ( $X = 1$ ) when the excitation threshold is 1 ( $\theta = 1$ ) as at least one of  $a_1$  and  $a_2$  is always true (equal to 1).

$$a_1 = 1, a_2 = 1 \Rightarrow a_1 + a_2 = 1 + 1 = 2, 2 \geq \theta \Rightarrow X = 1$$

$$a_1 = 1, a_2 = 0 \Rightarrow a_1 + a_2 = 1 + 0 = 1, 1 \geq \theta \Rightarrow X = 1$$

$$a_1 = 0, a_2 = 1 \Rightarrow a_1 + a_2 = 0 + 1 = 1, 1 \geq \theta \Rightarrow X = 1$$

Conversely, if neither of  $a_1$  and  $a_2$  are true then excitation threshold should not be reached and the neuron will not fire.

$$a_1 = 0, a_2 = 0 \Rightarrow a_1 + a_2 = 0 + 0 = 0, 0 < \theta \Rightarrow X = 0$$

The above functions show that the inequality of the excitation threshold must be  $0 < \theta \leq 1$ .

## 2(c)

Represented in boolean algebra as  $\neg a_1$ , and table form below, the logical OR gate can be represented as an MP-neuron as in figure 6.

$a_1$	"NOT"
1	0
0	1

This table shows all possible outputs of the NOT gate depending on all possible input values of  $a_1$ .

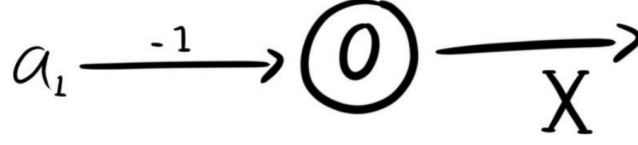


Figure 6: A diagram of an MP-neuron realisation of a “NOT” logical gate

For this NOT gate, the excitation threshold ( $\theta$ ), must be  $\theta = 0$ . This threshold value can be proven to be correct by analysing the output for each possible value of  $a_1$ .

If  $a_1$  is true ( $a_1 = 1$ ), it causes an input from an inhibitory connection, therefore the neuron will not fire. If  $a_1$  is false ( $a_1 = 0$ ), then the threshold is met as  $0 \geq \theta$ . The neuron will fire.

## 2(d)

The first neuron is an OR gate. Since both  $a_1$  and  $a_2$  are 0, it will not as the threshold of 1 is not met. This can also be seen in table format in question 2(b). This is followed by a NOT gate. Since the first neuron has not fired (providing an input of 0), this neuron will fire as the threshold of 0 is met. This logic can be seen in the NOT gate in question 2(c). As this is the final gate, and it has fired, the output is  $X = 1$  (The neuron fires).

These answers can also be found by following the flow chart shown in part 2(a), as seen in figure 7

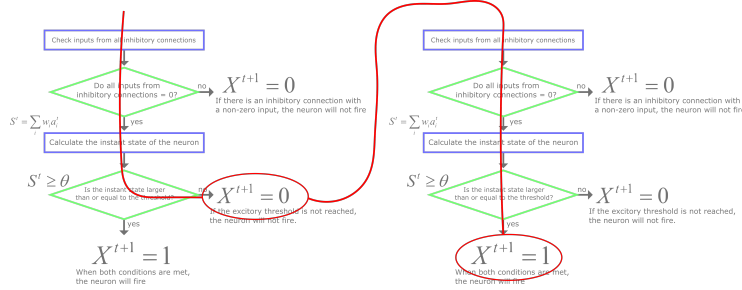


Figure 7: The route taken through the flow chart for the network in question 2(d)

## 2(e)

Figure 8 shows the MP-neuron representation of a memory cell. A memory cell, such as this, can be formed by closing a neuron in a feedback loop. To initialise this cell, an excitatory input ( $a_1$  in this case) is fired. This causes the value of 1 to cycle the loop (repeatedly fire the neuron), essentially storing it in the cell's memory. Even if  $a_1$  were to fire again, the loop would still be constant at 1. That is, until the inhibitory connection fires.  $a_2$  firing would cause the neuron to not fire, therefore breaking the cycle.

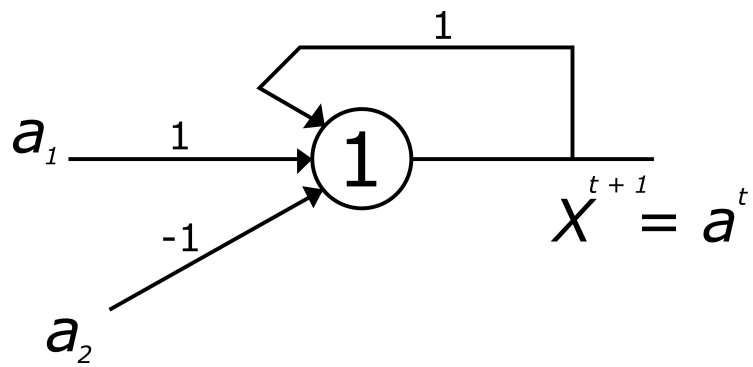


Figure 8: An MP-neuron representation of a memory cell

### 3 Learning rules of the Artificial Neural Networks. Hebb's Rule.

3(a)



# References

- [1] R. Xu, X. Cai, and D. C. Wunsch, “Gene expression data for dlbc cancer survival prediction with a combination of machine learning technologies,” in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2006, pp. 894–897.
- [2] J. D. Lohn, G. S. Hornby, and D. S. Linden, *An Evolved Antenna for Deployment on Nasa’s Space Technology 5 Mission*. Boston, MA: Springer US, 2005, pp. 301–315. [Online]. Available: [https://doi.org/10.1007/0-387-23254-0\\_18](https://doi.org/10.1007/0-387-23254-0_18)
- [3] *Artificial Intelligence: A Modern Approach. Third Edition*. England: Pearson, 2016.
- [4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [5] T. Geijtenbeek, M. van de Panne, and A. F. van der Stappen, “Flexible muscle-based locomotion for bipedal creatures,” *ACM Transactions on Graphics*, vol. 32, no. 6, 2013.
- [6] T. Geijtenbeek, *Flexible Muscle-Based Locomotion for Bipedal Creatures*, 2013. [Online]. Available: [https://www.youtube.com/watch?v=pgaEE27nsQw&feature=emb\\_title&ab\\_channel=ThomasGeijtenbeek](https://www.youtube.com/watch?v=pgaEE27nsQw&feature=emb_title&ab_channel=ThomasGeijtenbeek)
- [7] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–102. [Online]. Available: [https://doi.org/10.1007/3-540-32494-1\\_4](https://doi.org/10.1007/3-540-32494-1_4)
- [8] H. Braun, “On solving travelling salesman problems by genetic algorithms,” in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 129–133.
- [9] A. Gupta, A. Vedaldi, and A. Zisserman, “Learning to read by spelling: Towards unsupervised text recognition,” in *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, 2018, pp. 1–10.