HERIOT
WATT
UNIVERSITY

SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES

Department of Computer Science

F21BC

BIOLOGICALLY INSPIRED COMPUTATION

Semester 1 2021/22

December 2021

Duration: 24 hours

ANSWER THREE OF THE FOUR QUESTIONS

Q1

(a) Consider the function *y=abs(x),* where “abs” is the absolute value function, which returns the absolute value of the input *x*. Explain why this function is not a good choice of activation function in neural networks. (4)

(b) Why is Logistic Regression a linear classifier, even though its activation (sigmoid function) is non-linear? Please give your explainations supported by mathematical analysis. (6)

(c) You are training a multilayer perceptron using backpropagation, and during the first few epochs you find that the loss curve (the value of the loss function along with the epochs) is very unstable: it goes up and down dramatically. You then change the learning rate from 0.5 to 0.1, and you find that the loss function is still unstable, but the change become less dramatic. What would be the reason for this? How could you deal with this situation? (4)

(d) The below *Algorithm 1* is a variant of the perceptron learning algorithm. In the algorithm, *train_X* is the training set, and *train_y* is the corresponding set of labels. **x**=(x_1, x_2, …, x_D) is the input vector, where D is the number of dimensions. The label *y* is either +1 or -1.

w=(w_1, w_2, …, w_D) and *b* are the weight vector and bias term, respectively. w_d and x_d are the *dth-dimension* component of ***w*** and ***x***, respectively.

(i) Please explain the function of line 8. (2)

(ii) Please prove the effectiveness of the weight and bias update rules in lines 9 and 10. You should use mathematical analysis to prove this. (4)

Note: *Algorithm 1* is not a gradient decent algorithm.

Algorithm 1: PeceptronLearn(*train_X, train_y*)

```
1: init_parameters()                              // initialise parameters
2:  for epoch = 1 … maximum_iterations do         // for every epoch
3:       $preds \leftarrow []$
4:      for all $(x, y) \in$ train_X, train_y do
```

```
        $a \leftarrow \sum_{d=1}^{D} w_d x_d + b$          // compute activation for this example
        $\hat{y} = sign(a)$               // predict the label, if a>=0, $\hat{y}$=1, otherwise, $\hat{y}$=-1
        $preds \leftarrow preds + \hat{y}$          // add the predicted label to the results
        if $ya < 0$ then               // why do we have ya<0 here?
            $w_d \leftarrow w_d + \mathbf{y}\mathbf{x}_\mathbf{d}$   for all d=1…D   // update weights
            $\boldsymbol{b} \leftarrow \boldsymbol{b} + \mathbf{y}$                              // update bias
        end if
    end for
end for
```

Q2

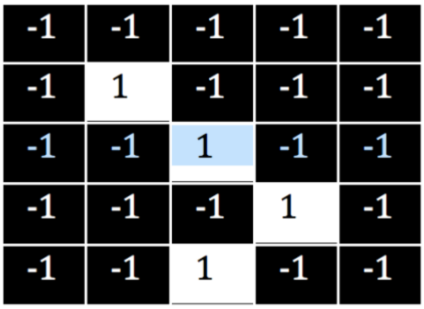
(a) Deep Learning always uses neural networks as the choice of architecture. Please explain if this is true or not. (3)

(b) Explain what bias is in neural networks and use a simple example to illustrate why we need bias in neural networks. You can consider a 2D situation and draw a graph to illustrate your ideas. (4)

(c) Explain why we can still use backpropagation to train convolutional neural networks.

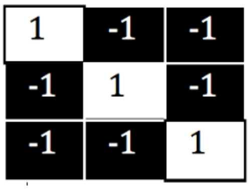
(d) In an evolutionary algorithm, the size of the population is 1 (only one individual in the population), the crossover operator is not used, and the mutation probability is 100%. What will this algorithm become? Explain the behaviour of this algorithm when it is used to solve a problem. (3)

(e) We use a convolutional neural network to process the following binary image of 5*5 pixels. The size of the filters is 3*3, and the size of the pooling window is 2*2. The values of the input raw picture and one of the filters are given as follows: (4)



-1	-1	-1	-1	-1
-1	1	-1	-1	-1
-1	-1	1	-1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1

Input: The Raw Picture



1	-1	-1
-1	1	-1
-1	-1	1

One of the Filters (Kernels)

In the above, assume we first use the average filter to get the convolutional layer, and then we use Rectifier as the activation function to obtain the ReLU (Rectified Linear Unit) layer, and finally use the max pooling operator to generate the pooling layer. We also set the stride for both filter window and pooling window to 2.

Note: The Rectifier activation function is defined as follows: f(x)=max (0, x).

(i) Calculate the size of the convolutional layer and all the values in this layer. Please provide the detailed steps of the calculation. (2)

(ii) Calculate the values of the output of the ReLU layer. (2)

(iii) Calculate the value of the output of the pooling layer. (1)

(iv) What observations can you make and what issues have you identified by calculating the above values? (1)

Q3

In the coursework, you were required to use Particle Swarm Optimisation (PSO) to optimise the weights of a neural network.

(a) Consider if the neural network was required to solve a problem with many more inputs. Do you think this would make it more challenging to use PSO to optimise the neural network? If so, why? (4)

(b) Consider if you were also required to use PSO to optimise the choice of activation function used in each neuron or layer of the neural network.

(i) Why would this be a useful thing to do? (2)

(ii) How would you do it? (2)

(iii) Do you anticipate any challenges in doing so? (2)

(c) Why might you want to apply a multiobjective optimiser to the problem of neural network optimisation? (4)

(d) Classification problems can also be solved by Genetic Programming (GP). Would there be any advantage to using GP rather than a neural network? (3)

(e) Another bio-inspired model of computation we looked at in the course is Cellular Automata (CA). Do you think it would be possible to use a CA to solve a classification problem, such as the one that was addressed in the coursework? If so, how might you go about designing a CA to solve this problem? (3)

Q4

(a) It is important to optimise the hyperparameters when applying a machine learning algorithm to a problem. What do you think are the important hyperparameters within Koza's tree-based Genetic Programming (GP) and why? (4)

(b) GP systems are limited to evolving relatively small programs. Why do you think this is the case? (4)

(c) GP systems are usually based around evolutionary algorithms. Do you think Particle Swarm Optimisation (PSO) could be used instead as the optimiser within a GP system? What do you think would be the challenges in doing so? In answering this, you might consider how solutions are represented and optimised within PSO. (4)

(d) Ant Colony Optimisation (ACO) is another well-known swarm computing algorithm. What are the main differences between PSO and ACO? (4)

(e) A key characteristic of swarm systems is that the overall behaviour of the swarm emerges from bottom-up interactions between the swarm's members. Discuss one advantage and one disadvantage of this from a computational and/or engineering perspective. (4)

END OF PAPER