Assignment_02

Table of contents

1 Question

(20 marks) We consider the artificial neural network (ANN) with two hidden layers that we studied in Assignment 1 (see the following Figure 1), and let the activation function be the rectified linear unit (ReLU). The ReLU-ANN has been trained with a dataset and the training results are shown in Table 1. Let y = gANN (x0,1, x0,2, x0,3) be the trained ReLU-ANN model.

We are to solve the following optimization problem that involves the ReLU-ANN model:

Minimize:

$$y - 1.08x_{0,2}$$

Subject to:

$$y = g_{ANN}(x_{0,1}, x_{0,2}, x_{0,3})$$

$$-2475x_{0,1} + 4703x_{0,2} - 2475.6 \leq 0$$

$$-1 \leq x_{0,1} \leq 1$$

$$-1 \leq x_{0,2} \leq 1$$

$$-1 \leq x_{0,3} \leq 1$$

Please answer the following questions (either in Chinese or in English):

1.1 Express the optimization problem as a mixed-integer linear programming (MILP) problem, and solve it with MATLAB "intlinprog". Please include your MATLAB codes in the answer. (8 marks) (Note: You need some "big-M" value in the formulation. Please use a value that is big enough but not overly large)

For the first hidden layer's nodes, the calculations using the ReLU activation function are:

- For node j:

$$h_{1,j} = \max(0, w_{0,1j} \times x_{0,1} - w_{0,2j} \times x_{0,2} + w_{0,3j} \times x_{0,3} + b_{1,j})$$

The ReLU activation applies a threshold operation to the weighted sum of inputs and biases at each node. In addition, since the objective function is not linear, the first thing should be done is to linearlize the objective function. In other words, we need to linearlize the constraint $y = g_{ANN}(x_{0,1}, x_{0,2}, x_{0,3})$. Big-M method is used.

Since y = ReLU(f) is equivalent to the $y = max(f, 0)$, and $y = max(x, 0)$ could be denoted as

$$y \geq f$$

$$y \geq 0$$

$$y \leq f + M(1 - z)$$

$$y <= Mz$$

where, - M is a number which is big enough; - z is a binary variable.

Thus, based on this transformation, we can decomposite the problem, for layer 1, the output is $[x_{11}, x_{12}, x_{13}]^T$

$$x_{11} \geq w_{0,11}x_{01} + w_{0,21}x_{02} + w_{31}x_{03} + b_{11}$$

$$x_{11} \geq 0$$

$$x_{11} \leq w_{0,11}x_{01} + w_{0,21}x_{02} + w_{31}x_{03} + b_{11} + M(1 - z_{11})$$

$$x_{11} <= Mz_{11}$$

The same things happen for x_{12} and x_{13} and also the different layers, thus a general universal expression can be written as:

$$x_{kj} \geq w_{k-1,1j}x_{(k-1)1} + w_{k-1,2j}x_{(k-1)2} + w_{k-1,3j}x_{(k-1)3} + b_{kj}$$

$$x_{kj} \geq 0$$

$$x_{kj} \leq w_{k-1,1j}x_{(k-1)1} + w_{k-1,2j}x_{(k-1)2} + w_{k-1,3j}x_{(k-1)3} + b_{kj} + M(1 - z_{kj})$$
$$x_{kj} <= Mz_{kj}$$

And write it into matrix format:

$$X_k \geq W_{k-1}X_{(k-1)} + B_k$$
$$X_k \geq 0$$
$$X_k \leq W_{k-1}X_{k-1} + B_k + M(I - Z_k)$$
$$X_k \leq MZ_k$$

And finally we will get $[x_{21}, x_{22}, x_{23}]^T$, then,

$$y = w_{2,11}x_{21} + w_{2,21}x_{22} + w_{2,31}x_{23} + b_{31}$$
$$y = W_2X_2 + B_3$$

The objective function can be denoted as: $y = W_2X_2 + B_3 - 1.08x_{0,2}$

Since here $x_{0,2}$ is not a matrix element, we need to replace it. Obviously we have:

$$\begin{bmatrix} 0 & -1.08 & 0 \end{bmatrix} \begin{bmatrix} x_{01} \\ x_{02} \\ x_{03} \end{bmatrix} = -1.08x_{0,2}$$

Let $[0, -1.08, 0]$ be A, the objective function will be:

$$y = W_2X_2 + B_3 - AX_0$$

Similarly,

$$\begin{bmatrix} -2475 & 4703 & 0 \end{bmatrix} \begin{bmatrix} x_{01} \\ x_{02} \\ x_{03} \end{bmatrix} - 2475.6 = -2475x_{0,1} + 4703x_{0,2} - 2475.6$$

Let $[-2475, 4703, 0]$ be C:

$$CX_0 - 2475.6 \leq 0$$

Conclusively,

Objective fucntion: minimize

$$y - AX_0$$
$$A = [0, -1.08, 0]$$

is subject to constraints:

$$CX_0 - 2475.6 \leq 0$$

$$C = [-2475, 4703, 0]$$

$$-1 \leq X_0 \leq 1$$

$$X_k \geq W_{k-1}X_{(k-1)} + B_k$$

$$X_k \geq 0$$

$$X_k \leq W_{k-1}X_{k-1} + B_k + M(I - Z_k)$$

$$X_k \leq MZ_k$$

$$k \in \{1, 2\}$$

$$X_0 \subseteq \mathbb{Z}$$

$$y = W_2X_2 + B_3$$

Ps: when a metrix is compared to a real number, it means you compare element wisely in Matlab.

Based on this general formulation, we implement it into matlab.

```
% Big-M method
M = 1000;

%Input weights and biases
Weights = {
% Layer 1
[1.842, -3.016, 0.039; 1.120, -0.045, 0.172; 1.122, -0.169, 0.235],
% Layer 2
[0.215, -0.936, -0.412; 0.267, -0.536, -0.649; -0.191, 0.578, -0.571],
% Layer 3
[-0.555, -0.119, 0.948]
};

Biases = {
[-0.392; 1.209; 0.301],  % Layer 1
[1.804; -1.293; -1.339],  % Layer 2
[0.065]  % Layer 3
};


% Parameters setting
Z = [0, -1.08, 0];
```

```
C = [-2475, 4703, 0];
I = eye(3);
O = zeros(3,3);

% Ignore the constant in objective function temporarily
const = Biases{3};

% Optimization
% Variables: X0, X1, X2, Z1, Z2 [3*1]

% Objective function
f = [-Z, [0,0,0], Weights{3},[0,0,0],[0,0,0]] % -AX0 + 0X1 + W2X2 + 0Z1 + 0Z2

intcon = [[1,2,3],[10,11,12],[13,14,15]]; %MILP problem,
%input should be integer, and variables z1 and z2 are binary

A = [C,[0,0,0],[0,0,0],[0,0,0],[0,0,0];
    Weights{1},-I,O,O,O;
    -Weights{1},I,O,1000*I,O;
    O,I,O,-M*I,O;
    O,Weights{2},-I,O,O;
    O,-Weights{2},I,O,1000*I;
    O,O,I,O,-M*I]

b = [2475.6;
    -Biases{1};
    Biases{1}+M;
    [0;0;0];
    -Biases{2};
    Biases{2}+M;
    [0;0;0]]

ub = [[1;1;1]; [inf;inf;inf]; [inf;inf;inf]; [1;1;1]; [1;1;1]] % Upper bound
lb = [[-1;-1;-1]; [0;0;0]; [0;0;0]; [0;0;0]; [0;0;0]] % Lowwer bound

[x_optimal, fval] = intlinprog(f,intcon,A,b,[],[],lb,ub);

disp(x_optimal) %X0,X1,X2,Z1,Z2
disp(fval + const)
```

After running this code, we could get the optimal solution

$$X_0 = (x_{01}, x_{02}, x_{03}) = (-1, -1, -1)$$

$$Optimal = -2.1049$$

Figure 1: Result(a)

1.2 The convex relaxation of the MILP is a linear programming (LP) problem. Solve the LP problem with MATLAB "linprog". What is the relaxation gap (i.e., the difference between the optimal objective values of the LP relaxation and the original MILP)? Please include your MATLAB codes. (6 marks)

Since LP is the convex relaxation of the MILP, what we do is just to get rid of integerization conditions. And in order to obtain the optimal solution, we will have 2 situations:

$$X_{k,j} = \sum W_{k-1,j} X_{k-1,j}$$

or

$$X_{k,j} = 0$$
$$\forall j \in \{1, 2, 3\}$$
$$\forall k \in \{1, 2\}$$

We perform the LP problem optimizations for all cases with the help of for loop, and since the union of the four parts has been completed, we compare the obtained results and discussed its feasibility.

Objective fucntion:

minimize

$$y - AX_0$$
$$A = [0, -1.08, 0]$$

is subject to constraints:

$$CX_0 - 2475.6 \leq 0$$

$$C = [-2475, 4703, 0]$$

$$-1 \leq X_0 \leq 1$$

$$X_k \geq W_{k-1}X_{(k-1)} + B_k$$

$$X_k \geq 0$$

$$X_1 = f(X_{k-1})$$

$$k \in \{1, 2\}$$

$$y = W_2X_2 + B_3$$

Ps: when a metrix is compared to a real number, it means you compare element wisely in Matlab.

The matlab codes are:

```
Assum = [0,0,0];
%Input weights and biases
Weights_layer1 = {
[1.842, -3.016, 0.039; 1.120, -0.045, 0.172; 1.122, -0.169, 0.235],  % Layer 1
[Assum; 1.120, -0.045, 0.172; 1.122, -0.169, 0.235],
[1.842, -3.016, 0.039; Assum; 1.122, -0.169, 0.235],
[1.842, -3.016, 0.039; 1.120, -0.045, 0.172; Assum],
[Assum;  Assum; 1.122, -0.169, 0.235],
[Assum; 1.120, -0.045, 0.172; Assum],
[1.842, -3.016, 0.039; Assum; Assum],
[Assum; Assum; Assum]};

Weights_layer2 = {
[0.215, -0.936, -0.412; 0.267, -0.536, -0.649; -0.191, 0.578, -0.571],  % Layer 2
[Assum; 0.267, -0.536, -0.649; -0.191, 0.578, -0.571],
[0.215, -0.936, -0.412; Assum; -0.191, 0.578, -0.571],
[0.215, -0.936, -0.412; 0.267, -0.536, -0.649; Assum],
[Assum; Assum; -0.191, 0.578, -0.571],
[Assum; 0.267, -0.536, -0.649; Assum],
[0.215, -0.936, -0.412; Assum; Assum],
[Assum; Assum; Assum]
};

Weights_layer3 = [-0.555, -0.119, 0.948]  % Layer 3
```

```
Biases_layer1 = {
[-0.392; 1.209; 0.301],  % Layer 1
[0; 1.209; 0.301],
[-0.392; 0; 0.301],
[-0.392; 1.209; 0],
[0; 0; 0.301],
[0; 1.209; 0],
[-0.392; 0; 0],
[0; 0; 0]
}

Biases_layer2 = {
[1.804; -1.293; -1.339],  % Layer 2
[0; -1.293; -1.339],
[1.804; 0; -1.339],
[1.804; -1.293; 0],
[0; 0; -1.339],
[0; -1.293; 0],
[1.804; 0; 0],
[0; 0; 0]}


Biases_layer3 = [0.065]  % Layer 3


% Ignore the constant in objective function temporarily
const = Biases_layer3;

for i = 1:8
    for j = 1:8
        % Parameters setting
        Z = [0, -1.08, 0];
        C = [-2475, 4703, 0];
        I = eye(3);
        O = zeros(3,3);

        % Optimization
        % Variables: X0, X1, X2, [3*1]

        % Objective function
        f = [-Z, [0,0,0], Weights_layer3]; % -AX0 + 0X1 + W2X2
```

```
        A = [C,[0,0,0],[0,0,0]];

        b = [2475.6];

        Aeq = [Weights_layer1{i},-I,O;
               O,Weights_layer2{j},-I];

        beq = [-Biases_layer1{i};
               -Biases_layer2{j}];

        ub = [[1;1;1]; [inf;inf;inf]; [inf;inf;inf]]; % Upper bound
        lb = [[-1;-1;-1]; [0;0;0]; [0;0;0]]; % Lowwer bound

        [x_optimal, fval, exitflag] = linprog(f,A,b,Aeq,beq,lb,ub);
        if fval < -2.1049
            disp(x_optimal);%X0,X1,X2,Z1,Z2
            disp(fval + const);
        else
            disp(' ');
        end
    end
end
```

We also use the following code to check the feasibility:

```
% Define the weights and biases
weights = {
[1.842, -3.016, 0.039; 1.120, -0.045, 0.172; 1.122, -0.169, 0.235],  % Layer 1
[0.215, -0.936, -0.412; 0.267, -0.536, -0.649; -0.191, 0.578, -0.571],  % Layer 2
[-0.555, -0.119, 0.948]  % Layer 3
};
biases = {
[-0.392; 1.209; 0.301],  % Layer 1
[1.804; -1.293; -1.339],  % Layer 2
[0.065]  % Layer 3
};

x = [-0.9661; -1; -1];
O = [0;0;0];
% Calculate the activations for each layer
z1 = weights{1} * x + biases{1};
a1 = max(O, z1); % ReLU activation for Layer 1
```

```
z2 = weights{2} * a1 + biases{2};
a2 = max(0, z2); % ReLU activation for Layer 2

g = weights{3} * a2 + biases{3} - [0, -1.08, 0] * x; % Output of the network

disp(g)
```

After the comparison, when

$$X_0 = [x_{01}, x_{02}, x_{03}] = [-0.9661; -1; -1]$$

$$optimal = -2.1123$$

The relaxation gap is equal to $-2.1049 - (-2.1123) = 7.4 * 10^{-3}$ ## Write a Lagrangian dual problem of the MILP problem in part (a) such that any dual function can be calculated by solving two independent optimization problems, where one optimization problem includes output variables of the second hidden layer and the output layer and the other optimization problem includes output variables of the input layer and the first hidden layer. (6 marks)