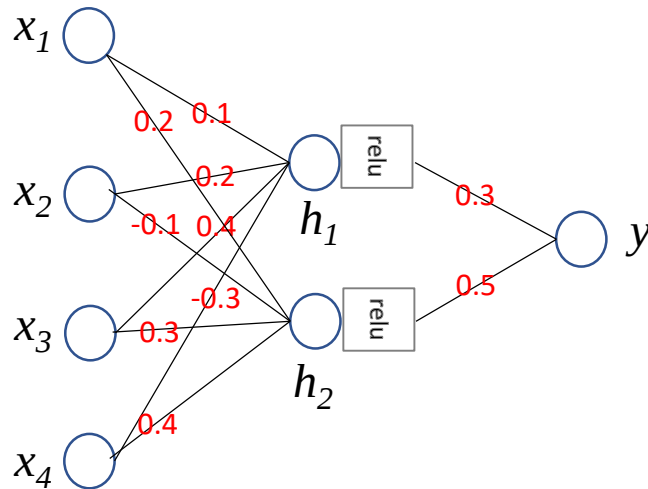


Algorithm 1 (NNexplainer) Example

BASE neural network



$$\mathbf{W}_o = \begin{bmatrix} 0.3 \\ 0.5 \end{bmatrix}$$

$$\mathbf{W}_I = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & -0.1 \\ 0.4 & 0.3 \\ -0.3 & 0.4 \end{bmatrix}$$

Task

Show the contribution of x_1 , x_2 , x_3 , and x_4 to the output y when the input values x are as follows:

$$\text{Input } \mathbf{x} = [0.6 \quad 0.5 \quad 0.5 \quad 0.1]$$

Algorithm

Algorithm 1. Calculation of contribution of input values

```
1  Inputs:
2    test_input    // input data for prediction
3    weights       // list of weight matrixes of predictive model
4    N             // number of layers
5  Output:
6    cont_list     // list of contribution values for each feature
7
8  BEGIN
9
10 // build a list of transformation rate for each layer
11 TR = []          // transformation rate
12 FOR ln FROM 0 To (N-1)
13   calculate weighted sum WS of layer ln
14   calculate layer_outputs of layer ln    // layer_outputs =  $\phi$ (WS)
15   TR_tmp := layer outputs / WS
16   append TR_tmp to TR
17 END FOR
18
19 // calculate contribution of input values
20 cont_matrix_before := diag(test_input) // diagonal of test_input
21
22 FOR ln FROM 0 to (N-1)
23   // Calculate Cx for given layer
24   cont_matrix_this := mat_mul(WEIGHTS[ln].T, cont_matrix_before)
25   cont_matrix_before := cont_matrix_this  $\otimes$  TR[ln]
26 END FOR
27
28 cont_list = cont_matrix_before
29 RETURN(cont_list)
30
31 END
```

```

1 Inputs:
2   test_input    // input data for prediction
3   weights       // list of weight matrixes of predictive model
4   N             // number of layers
5 Output:
6   cont_list     // list of contribution values for each feature

```

```

test_input = [0.6, 0.5, 0.5, 0.1]
weights    = [
    [
        [0.1, 0.2]
        [0.2, -0.1]
        [0.4, 0.3]
        [-0.3, 0.4]
    ],
    [
        [0.3]
        [0.5]
    ]
]

```

N = 2

```

10 // build a list of transformation rate for each layer
11 TR = [] // transformation rate
12 FOR In FROM 0 To (N-1)
13     calculate weighted sum WS of layer In
14     calculate layer_outputs of layer In // layer_outputs =  $\phi$ (WS)
15     TR_tmp := layer outputs / WS
16     append TR_tmp to TR
17 END FOR

```

In = 0

$$WS = [0.6, 0.5, 0.5, 0.1] \bullet \begin{pmatrix} 0.1, & 0.2 \\ 0.2, & -0.1 \\ 0.4, & 0.3 \\ -0.3, & 0.4 \end{pmatrix} = [0.33, 0.26]$$

$$\text{layer_outputs} = \text{relu}([0.33, 0.26]) = [0.33, 0.26]$$

$$TR_tmp = [0.33, 0.26] / [0.33, 0.26] = [1, 1]$$

$$TR = [[1, 1]]$$

In = 1

$$WS = [0.33, 0.26] \bullet \begin{pmatrix} 0.3 \\ 0.5 \end{pmatrix} = [0.229]$$

$$\text{layer_outputs} = \text{relu}([0.229]) = [0.229]$$

$$TR_tmp = [0.229] / [0.229] = [1]$$

$$TR = [[1, 1], [1]]$$

```
19 // calculate contribution of input values
20 cont_matrix_before := diag(test_input) // diagonal of test_input
```

```
test_input = [0.6, 0.5, 0.5, 0.1]
```

```
cont_matrix_before =  $\begin{pmatrix} 0.6, & 0, & 0, & 0 \\ 0, & 0.5, & 0, & 0 \\ 0, & 0, & 0.5, & 0 \\ 0, & 0, & 0, & 0.1 \end{pmatrix}$ 
```

```

22  FOR ln FROM 0 to (N-1)
23    // Calculate Cx for given layer
24    cont_matrix_this := mat_mul(WEIGHTS[ln].T, cont_matrix_before)
25    cont_matrix_before := cont_matrix_this  $\otimes$  TR[ln]
26  END FOR

```

ln = 0

$$\text{cont_matrix_this} = \begin{bmatrix} 0.1 & 0.2 & 0.4 & -0.3 \\ 0.2 & -0.1 & 0.3 & 0.4 \end{bmatrix} \bullet \begin{bmatrix} 0.6 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.06 & 0.1 & 0.2 & -0.03 \\ 0.12 & -0.05 & 0.15 & 0.04 \end{bmatrix}$$

$$\text{cont_matrix_before} = \begin{bmatrix} 0.06 & 0.1 & 0.2 & -0.03 \\ 0.12 & -0.05 & 0.15 & 0.04 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.06 & 0.1 & 0.2 & -0.03 \\ 0.12 & -0.05 & 0.15 & 0.04 \end{bmatrix}$$

ln = 1

$$\text{cont_matrix_this} = [0.3, 0.5] \bullet \begin{bmatrix} 0.06 & 0.1 & 0.2 & -0.03 \\ 0.12 & -0.05 & 0.15 & 0.04 \end{bmatrix} = [0.078, 0.005, 0.315, 0.011]$$

$$\text{cont_matrix_before} = [0.078, 0.005, 0.315, 0.011] \otimes [1] = [0.078, 0.005, 0.315, 0.011]$$

```
28 cont_list = cont_matrix_before  
29 RETURN(cont_list)
```

```
cont_list = [0.078, 0.005, 0.315, 0.011]
```

```
** sum([0.078, 0.005, 0.315, 0.011]) = 0.229
```

Final output $y = 0.229$

Contribution of x_1 , x_2 , x_3 , and x_4 is [0.078, 0.005, 0.315, 0.011]

