Dokumentacja implementacji algorytmu wstecznej propagacji na przykładzie funkcji NXoR

Michał Korniak

Spis treści

1 Wprowadzenie **2**

2 Algorytm wstecznej propagacji **2**

2.1 Sposób działania wielowarstwowych sieci neuronowych 2

2.2 Implementacja wielowarstwowej sieci neuronowej 4

1 Wprowadzenie

Algorytm wstecznej propagacji błędu jest metodą uczenia wielowarstwowych polegającą na korekcie wag połączeń między neuronami na podstawie błędu całej sieci. W tym dokumencie prześledzę działanie tego algorytmu na przykładzie sieci realizującej funkcję NXoR, jednocześnie przedstawiając jego autorską implementację w języku C#.

2 Algorytm wstecznej propagacji

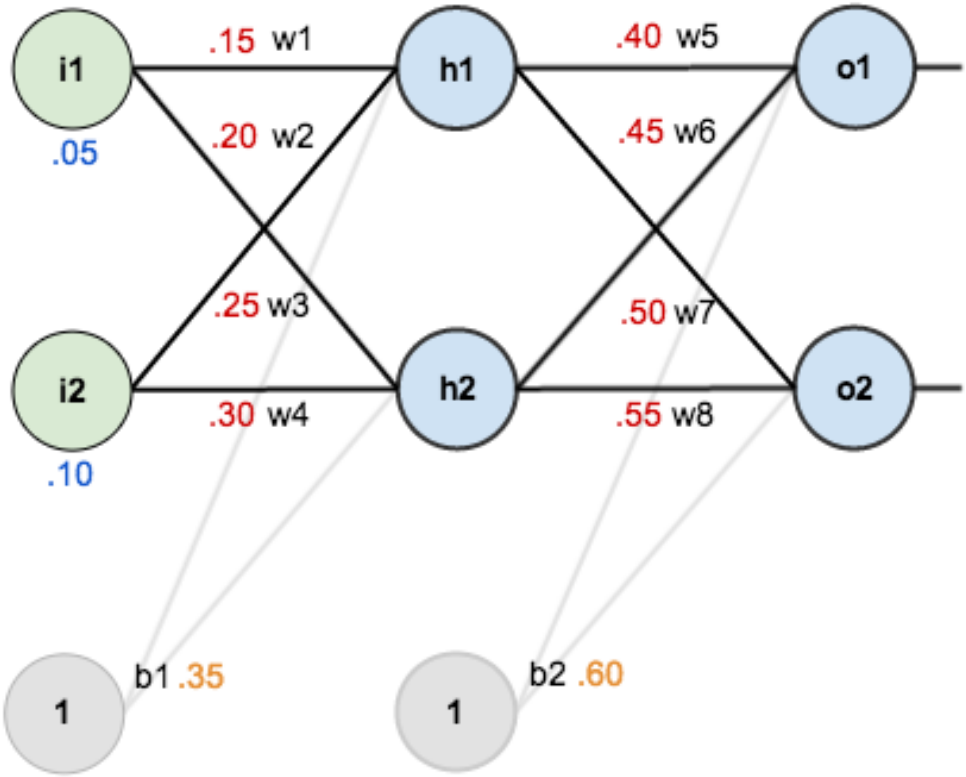
2.1 Sposób działania wielowarstwowych sieci neuronowych

Algorytm wstecznej propagacji jest wykorzystywany do nauczania wielowarstwowych sieci jednokierunkowych. Taka sieć zawiera warstwę wejściową i warstwę wyjściową, może posiadać również warstwy ukryte (w zależności od problemu jedną albo dwie). Każda warstwa zawiera dowolną ilość neuronów. Neurony są połączone ze sobą w taki sposób, że każdy neuron warstwy innej niż wyjściowa jest połączony z każdym neuronem kolejnej warstwy, a każde połączenie ma określoną wagę. Dodatkowo możliwe jest połączenie do tego zwanego biasa, czyli połączenia do neuronu, który zawsze przyjmuje wartość 1. Połączenia wejściowe do neuronu będą wpływać na to jaką będzie miał wartość.

Rysunek 1 przedstawia przykład jednokierunkowej sieci neuronowej opartej o dwa neurony warstwy wejściowej i tyle samo neuronów w warstwach ukrytej i wyjściowej. Oprócz połączeń do innych neuronów, istnieją również połączenia do biasa, który jest wspólny dla neuronów w ramach jednej warstwy.

Rysunek 1 zawiera wartości neuronów wejściowych oraz wag, dzięki czemu będziemy mogli pokazać jak wyliczane są wartości kolejnych neuronów. Robimy to sumując wartości połączeń do tego wektora, a następnie poddając wynik funkcji aktywacji. Przez wartość połączenia będziemy rozumieli iloczyn wagi i neuronu wejściowego dla połączenia. Przykładowo dla neuronu h1 wartość przed użyciem funkcji aktywacji będzie równa:

$$net = i1 * w1 + i2 * w2 + 1 * b1$$



Źródło: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Rysunek 1: Przykład wielowarstwowej sieci jednokierunkowej

Co po podstawieniu wartości da:

$$net_{h1} = 0.05 * 0.15 + 0.10 * 0.20 + 1 * 0.35 = 0.3775$$

Taka wartość jest następnie poddawana działaniu funkcji aktywacji. Takie działanie umożliwia normalizowanie wartości neuronów do oczekiwanych przedziałów wartości. W tym przypadku skorzystamy z funkcji sigmoidalnej, której zakres wartości mieści się w przedziale [0,1]:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Po podstawieniu otrzymanej wcześniej wartości do funkcji dostaniemy wartość neuronu h1:

$$out_{h1} = f(net_{h1}) = \frac{1}{1 + e^{-net_{h1}}} = \frac{1}{1 + e^{-0.3775}} \approx 0.5932$$

W ten sposób będziemy liczyć wartość kolejnych neuronów aż otrzymamy wynik.

2.2 Implementacja wielowarstwowej sieci neuronowej

Teraz przedstawię implementację mechanizmu zaprezentowanego w poprzednim podrozdziale. Jej podstawą jest klasa odwzorowującej sieć neuronową, w naszym przypadku taką funkcję pełni klasa NeuralNetwork, która to implementuje poniższy interfejs.

```
1   public interface INeuralNetwork
2   {
3       IEnumerable<IInputNeuron> InputLayer { get; }
4       IEnumerable<IHiddenNeuron> HiddenLayer { get; }
5       IEnumerable<IOutputNeuron> OutputLayer { get; }
6       IErrorFunction ErrorFunction { get; }
7       void FillInputNeurons(IEnumerable<double> input);
8       IEnumerable<double> CalculateOutput();
9   }
```

W tym momencie skupmy się na linijkach 3, 4 i 5, które to wskazują na to, że sieć neuronowa zawiera warstwy wejściową, ukrytą i wyjściową. Jak widzimy każda warstwa posiada inny typ neuronu, wynika to z tego, że w zależności od warstwy neurony się różnią. Przykładowo:

- Neuron warstwy wejściowej nie ma możliwości dodawanie połączeń wejściowych
- Neuron warstwy wyjściowej nie ma możliwości dodawanie połączeń wyjściowych
- Neuron warstwy ukrytej ma możliwość dodawanie obu typów połączeń