Relatório 2 - Regra de Hebb

1 Introdução

O neurônio de McCullock-Pitts, proposto em 1943, foi o primeiro modelo de neurônio artificial. Ele é uma simplificação do que sabia à respeito do neurônio biológico na época e consiste de diversas entradas (x_1 a x_n) ligadas a um núcleo Y, cada uma das ligações das entradas ao núcleo possuem um peso (w_1 a w_n), que podem ser excitatórias, caso $w_i > 0$, ou inibitórias caso $w_i < 0$.

No núcleo, tem-se uma função de saída $f(y_1)$ que, no caso de uma função degrau, compara um limiar Ø com a soma de um elemento bias (b) com a somatória das entradas vezes o peso de suas ligações ao núcleo, caso a soma seja maior ou igual ao limiar a função retorna 1, caso seja menor ela retorna 0.

Neste relatório deseja-se treinar um neurônio de McCulock-Pitts utilizando a Regra de Hebb que é uma regra proposta pelo neuropsicólogo Donald Hebb e baseada nos experimentos de Pavlov. A regra propõe que o condicionamento de neurônios promove alterações nas suas estrutura sináptica. Essas alterações, por vez, aumentam a eficiência sináptica dos neurônios. A alteração na eficiência sináptica é chamada de plasticidade sináptica e a plasticidade sináptica resulta em um modelo de aprendizado.

O modelo de aprendizado se baseia na afirmação de que a variação do peso de uma ligação é igual a sua entrada (x) vezes a resposta desejada (t). Usa-se disso e do conhecimento de que a variação de b é igual a t para ajustar os valores de w e b de modo que se consiga a saída desejada.

2 Objetivo

- Treinar um neurônio de McCulock-Pitts utilizando a regra de Hebb para que esse realize 14 das 16 operações lógicas possíveis com duas entradas usando a representação bipolar.

3 Desenvolvimento

Para realizar o treinamento do neurônio artificial foi escolhida a linguagem Common Lisp. A escolha da linguagem forçou uma abordagem um pouco diferente da apresentada

no exemplo dado, visto que, foi utilizada uma abordagem essencialmente funcional e o exemplo foi feito em uma linguagem procedural.

Primariamente foi observado que a formula do cálculo de b é muito semelhante à do calculo dos pesos, tento como diferença apenas que no caso do b seu valor valor anterior não é multiplicado pela entrada. Tendo isso em mente foi criada uma função que calcula tanto os pesos quanto o b. Para que essa função funcione é necessário adicionar mais uma entrada no valor de 1, que seria a entrada do calculo de b. Efetivamente, fazendo com que a entrada multiplicando b seja 1, e portanto com que o novo valor de b seja o b antigo vezes 1 mais a saída, ou seja b mais a saída, assim como desejado.

```
(defun ajuste-peso (entradas pesos saida)
  (map 'list #'+ pesos
       (map 'list #'(lambda (x) (* x saida)) entradas)))
```

Após a criação dessa função foi feita uma função recursiva que permite a entrada de uma lista de entradas e saídas, para que assim se consiga obter os pesos e o b ajustados finais

```
(defun treinamento (entradas pesos saidas)
  (if (and (null entradas) (null  saidas)) pesos
      (treinamento
       (rest entradas)
       (ajuste-peso (first entradas) pesos (first saidas))
       (rest saidas))))
```

Como citado anteriormente o resultado da função anterior é uma lista com os dois pesos e o b. Para que fosse possível utilizá-los separadamente foram criadas duas funções:

```
(defun w-separated (pesos-ajustados)
  (butlast pesos-ajustados))
(defun b-separated (pesos-ajustados)
 (last pesos-ajustados))
```

Tendo isso pronto a parte do treinamento do neurônio estava finalizada. Foi necessário, então, criar as funções que compariam a saída desejada com a saída obtida com os novos pesos e o novo b . Primeiro foi criada uma função para realizar o calculo da soma de b com a somatória das entradas vezes os pesos

```
(defun soma (pesos-ajustados entrada)
  (+ (first (b-separated pesos-ajustados))
     (apply #'+ (map 'list  #'* (w-separated pesos-ajustados) (w-separated entrada)))))
```

Com a função de soma criada o próximo passo foi a criação da função degrau.

```
(defun degrau (entrada pesos-ajustados limiar)
  (if (>= (soma pesos-ajustados entrada) limiar) 1 -1))
```

Por fim, foi criada uma função que testava se os pesos e b ajustados produziam as saídas esperadas.

```
(defun compara-saidas (entradas pesos saidas limiar)
  (compara-saidas-aux entradas (treinamento entradas pesos saidas) saidas limiar '()))
(defun compara-saidas-aux (entradas pesos saidas limiar resultado)
  (cond
    ((null entradas) (format t "Pesos: [~{~a~^ ~}] || b: ~a ~%"
                             (w-separated pesos)
                             (first (b-separated pesos))))
    (t
     (format t "Saída esperada:  ~a || Saída obtida: ~a ~%"
             (first saidas)
             (degrau (first entradas) pesos limiar))
     (compara-saidas-aux (rest entradas) pesos (rest saidas) limiar
                         (append resultado
                                 (list (degrau (first entradas) pesos limiar)))))))
```

4 Conclusão

A regra de Hebb e o neurônio de McCulock-Pitts são modelos simples que preparam para conceitos complexos a vir, entretanto mesmo com essa simplicidade foi possível treinar de maneira correta o neurônio artificial e obter as 14 saídas corretas de 16. Tendo apresentado erro somente no treinamento das funções lógicas XOR E XNOR. Segue na próxima sessão os resultados apresentados pelo código.

5 Resultados

```
                                    Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [-2 -2] || b: 2
Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  -1 || Saída obtida: -1
Pesos: [0 0] || b: -4                   Saída esperada:  -1 || Saída obtida: -1
                                        Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [2 2] || b: -2
Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  -1 || Saída obtida: 1
Pesos: [-2 -2] || b: -2                 Saída esperada:  -1 || Saída obtida: 1
                                        Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [0 0] || b: 0
Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  -1 || Saída obtida: -1
Pesos: [2 -2] || b: -2                  Saída esperada:  1 || Saída obtida: 1
                                        Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [4 0] || b: 0
Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  -1 || Saída obtida: -1
Pesos: [0 -4] || b: 0                   Saída esperada:  1 || Saída obtida: 1
                                        Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [2 -2] || b: 2
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  1 || Saída obtida: 1
Pesos: [-2 2] || b: -2                  Saída esperada:  -1 || Saída obtida: -1
                                        Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [0 4] || b: 0
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: -1 Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  1 || Saída obtida: 1
Pesos: [-4 0] || b: 0                   Saída esperada:  -1 || Saída obtida: -1
                                        Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: 1  Pesos: [-2 2] || b: 2
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  1 || Saída obtida: 1
Saída esperada:  -1 || Saída obtida: 1  Saída esperada:  1 || Saída obtida: 1
Pesos: [0 0] || b: 0                    Saída esperada:  1 || Saída obtida: 1
                                        Saída esperada:  -1 || Saída obtida: -1
Saída esperada:  -1 || Saída obtida: -1 Pesos: [2 2] || b: 2
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1   Saída esperada:  1 || Saída obtida: 1
```

```
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1
Saída esperada:  1 || Saída obtida: 1
Pesos: [0 0] || b: 4
```