Multi Layer Perceptron
MLP

Junior R. Ribeiro
jrodrib@usp.br

24 de setembro de 2020

Conteúdo

1 Multicamadas

Sejam dados n padrões para treinamento da rede neural, com as entradas e saídas desejadas $\left\{\bar{x}(n) \in \mathbb{R}^{x}, d(n) \in \mathbb{R}^{\gamma_L}\right\}$.

Considere $L \geq 2$ e as camadas $\ell = 0, 1, ..., L$, em que $\ell = 0$ é a camada de entrada, onde os padrões $\bar{x}(n)$ são apresentados, e em cada camada $\ell = 1, ..., L$ temos γ_ℓ neurônios artificiais. A camada de saída $\ell = L$ precisa ter o mesmo número de neurônios que as saídas desejadas $d(n)$, ou seja γ_L neurônios na camada L. Todas as demais camadas $\ell = 1, ..., L-1$ são as camadas ocultas.

Os vetores de fluxo serão indicados por $v^\ell(n) \in \mathbb{R}^{\gamma_\ell}$ para cada camada $\ell = 1, ..., L$. Vamos precisar aplicar uma função de ativação nesse vetor de fluxo a cada camada *forward*, obtendo $\varphi(v^\ell(n)) \in \mathbb{R}^{\gamma_\ell}$, vamos chamá-lo de vetor de fluxo ativado. Para este texto, a função de ativação será a sigmoide $\varphi(z) = 1/(1 + \exp(-z))$.

Vamos chamar os vetores $y^\ell(n)$ de *entrada* da camada $\ell + 1$. Eles são a concatenação do

número 1 com o vetor de fluxo ativado $\varphi(v^\ell(n))$, da seguinte forma

$$y^\ell(n) = \begin{bmatrix} 1 \\ \varphi(v^\ell(n)) \end{bmatrix} \in \mathbb{R}^{\gamma_\ell+1},$$

para todas as camadas $\ell = 1, ..., L$.

Na camada de entrada, temos

$$y^{(0)}(n) = \begin{bmatrix} 1 \\ \bar{x}(n) \end{bmatrix} \in \mathbb{R}^{x+1}$$

As matrizes de pesos e os vetores de *biases* são b^ℓ são W^ℓ para $\ell = 1, ..., L$. Suas dimensões são $b^\ell \in \mathbb{R}^{\gamma_\ell}$ e $W^\ell \in \mathbb{R}^{(\gamma_\ell \times \gamma_{\ell-1})}$. Na primeira camada, temos, $W^1 \in \mathbb{R}^{(\gamma_1 \times x)}$.. Por praticidade, vamos definir a matriz $w^\ell = [b^\ell \;\; W^\ell]$

$$w^\ell = \begin{bmatrix} b_1^\ell & W_{::}^\ell & ... & W_{::}^\ell \\ \vdots & & & \\ b_{\gamma_\ell}^\ell & W_{::}^\ell & ... & W_{::}^\ell \end{bmatrix}.$$

1.1 Forward propagation

Dado um par $\left\{\bar{x}(n), d(n)\right\}$, definimos $y^{(0)}(n)$ e mais adiante definimos o erro obtido. Para calcular os vetores de fluxo, fazemos a multiplicação matricial

$$v^\ell(n) = w^\ell y^{\ell-1}(n)$$

para $\ell = 1, ..., L$, considerando a definição de $y^\ell(n)$ e w^ℓ acima.

O erro obtido ao efetuar o fluxo da entrada $\bar{x}(n)$ pela rede é dado pela diferença entre a saída desejada $d(n)$ e o vetor de fluxo ativado da última camada,

$$e(n) = d(n) - \varphi(v^L(n)).$$

O erro quadrático é então

$$E(n) = 0.5e(n)^T e(n).$$

1.2 Back propagation

Vamos definir a multiplicação "ponto-a-ponto" entre matrizes de mesmas dimensões como sendo $[A \bullet B]_{rs} = A_{rs}B_{rs}$.

Para cada camada $\ell = L, ..., 1$, vamos definir o vetor $\delta^\ell(n)$ de mesma dimensão de $v^\ell(n)$ e a matriz $\Delta w^\ell(n)$ de mesma dimensão de w^ℓ.

Para $\ell = L$, defina

$$\delta^\ell(n) = e(n) \bullet \varphi(v^\ell(n)) \bullet (\mathbf{1} - \varphi(v^\ell(n)))$$

em que $\mathbf{1}$ representa o vetor de uns $[1, 1, ..., 1]^T$ de dimensões apropriadas.

Definimos $\delta^\ell(n)$ para as camadas $\ell = L-1, ..., 1$ da seguinte forma:

$$\delta^\ell(n) = \varphi(v^\ell(n)) \bullet (\mathbf{1} - \varphi(v^\ell(n))) \bullet \left[(W^{\ell+1})^T \delta^{\ell+1}(n)\right].$$

Repare na equação acima, que usamos apenas a parte dos pesos $W^{\ell+1}$ sem os *biases*.

Uma vez calculados os $\delta^\ell(n)$, e dado um tamanho de passo $0 < \eta \leq 1$, calculamos

$$\Delta w^\ell(n) = \eta \delta^\ell(n)(y^{\ell-1}(n))^T$$

para todas as camadas $\ell = 1, ..., L$.

1.3 Atualizando os pesos/*biases*

Modo *batch*: calculamos, para cada padrão $n = 1, ..., N$ os incrementos $\Delta w^\ell(n)$ para todas as camadas $\ell = 1, ..., L$. A atualização dos pesos no ciclo seguinte é a soma desses incrementos:

$$w^\ell \leftarrow w^\ell + \sum_{n=1}^{N} \Delta w^\ell(n).$$

Perceba que os pesos/biases só são modificados depois de serem considerados todos os padrões.

Modo padrão, ou modo cíclico: a cada padrão n apresentado, atualizamos os pesos:

$$w^\ell \leftarrow w^\ell + \Delta w^\ell(n).$$

Perceba que os pesos/biases são modificados a cada novo padrão apresendado.

Referências

[1] Riedmiller, Martin. *Machine learning: multi layer perceptrons*. Disponível aqui.

[2] Haykin, Simon. *Neural networks: a comprehensive fondation*. 2a.ed. Singapore: Prentice Hall, 1999. Disponível aqui.

[3] Haykin, Simon. *Neural networks and learning machines*. 3a.ed. New Jersey: Prentice Hall, 2008. Disponível aqui.