Back-Propagation Algorithm

November 6, 2018

1 Concept Description

- Matrices $\mathbf{V}$ and $\mathbf{W}$, which store the connection weights between the input and hidden layers and between the hidden and output layers, respectively.

2 Learning Element

- Note that

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

- Assume $\lambda = 1.0$.
- Given are M training pairs:

$$\{\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2, \ldots, \mathbf{x}_M, \mathbf{y}_M\},$$

  where $\mathbf{x}_i$ is $(I \times 1)$, $\mathbf{y}_i$ is $(K \times 1)$, and $i = 1, 2, \ldots, M$.
- Note that the Ith component of each $\mathbf{x}_i$ is of value -1 since input vectors have been augmented.
- Select the size $J-1$ of the hidden layer having outputs $\mathbf{h}$.
- Note that the Jth component of $\mathbf{h}$ is of value -1, since hidden layer outputs have also been augmented.
- $\mathbf{h}$ is $(J \times 1)$ and $\mathbf{o}$ is $(K \times 1)$.
- Step 1: Select $\eta > 0$, where η is the learning rate, and $E_{\min}$, where $E_{\min}$ is the minimum acceptable error.
- Initialize weight matrices $\mathbf{V}$ and $\mathbf{W}$ to small random values.
- $\mathbf{V}$ is $(J \times I)$, $\mathbf{W}$ is $(K \times J)$.
- $q = 1, m = 1, E = 0$, where q counts the total number of presentations made, m counts the number of presentations for a given epoch, and E is the error term for an epoch.
- Step 2: Training step starts here. Present input and compute layers' output:

$$\mathbf{x} = \mathbf{x}_m, \mathbf{y} = \mathbf{y}_m$$

$$h_j = f(\mathbf{v}_j \mathbf{x}), \text{ for } j = 1, 2, \ldots, J,$$

where $\mathbf{v}_j$ is the jth row of $\mathbf{V}$.

$$o_k = f(\mathbf{w}_k \mathbf{h}), \text{ for } k = 1, 2, \ldots, K,$$

where $\mathbf{w}_k$ is the kth row of $\mathbf{W}$.

- Step 3: Compute error value:

$$E = \frac{1}{2}(y_k - o_k)^2 + E, \text{ for } k = 1, 2, \ldots, K$$

- Step 4: Compute error signal vectors $\boldsymbol{\delta_o}$ and $\boldsymbol{\delta_h}$ for both layers.
- Vector $\boldsymbol{\delta_o}$ is $(K \times 1)$, $\boldsymbol{\delta_h}$ is $(J \times 1)$.
- The error signal terms of the output layer in this step are:

$$\delta_{ok} = (y_k - o_k)(1 - o_k)o_k, \text{ for } k = 1, 2, \ldots, K.$$

- The error signal terms of the hidden layer in this step are:

$$\delta_{hj} = h_j(1 - h_j) \sum_{k=1}^{K} \delta_{ok} w_{kj}, \text{ for } j = 1, 2, \ldots, J.$$

- Step 5: Adjust output layer weights:

$$w_{kj} = w_{kj} + \eta \delta_{ok} h_j, \text{ for } k = 1, 2, \ldots, K \text{ and } j = 1, 2, \ldots, J.$$

- Step 6: Hidden layer weights are adjusted:

$$v_{ji} = v_{ji} + \eta \delta_{hj} x_i, \text{ for } j = 1, 2, \ldots, J \text{ and } i = 1, 2, \ldots, I.$$

- Step 7: If $m < M$ then $m = m + 1, q = q + 1$, and go to Step 2; otherwise, go to Step 8.
- Step 8: The training cycle is completed. For $E < E_{\min}$ terminate the training session.
  Output weights $\mathbf{V}$, $\mathbf{W}$, q, and E. If $E > E_{\min}$, then $E = 0, p = 1$, and initiate the new training cycle by going to Step 2.

3 Performance Element

- Given an augmented observation $\mathbf{x}$, where $\mathbf{x}$ is $(I \times 1)$ and the Ith component of $\mathbf{x}$ is the value -1, compute

$$h_j = f(\mathbf{v}_j \mathbf{x}), \text{ for } j = 1, 2, \ldots, J,$$

where $\mathbf{v}_j$ is the jth row of $\mathbf{V}$.

$$o_k = f(\mathbf{w}_k \mathbf{h}), \text{ for } k = 1, 2, \ldots, K,$$

where $\mathbf{w}_k$ is the kth row of $\mathbf{W}$.

- Return $\mathbf{o}$ as the prediction for $\mathbf{x}$.

References

[1] C. M. Bishop. *Pattern recognition and machine learning.* Springer, Berlin-Heidelberg, 2006.

[2] J. M. Zurada. *Introduction to Artificial Neural Systems.* West Publishing, St. Paul, MN, 1992.