Introduction to Neural Networks

Multilayer Neural Networks

Juan Irving Vásquez
jivg.org

Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional

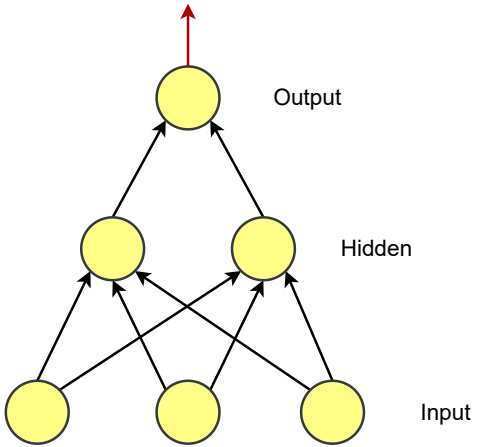
September 25, 2023©

Contents

Roadmap

1. Multilayer neural networks
2. Backpropagation

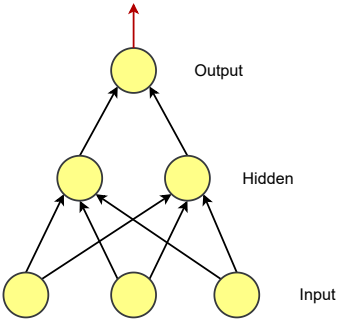
Multilayer Neural Network



Output

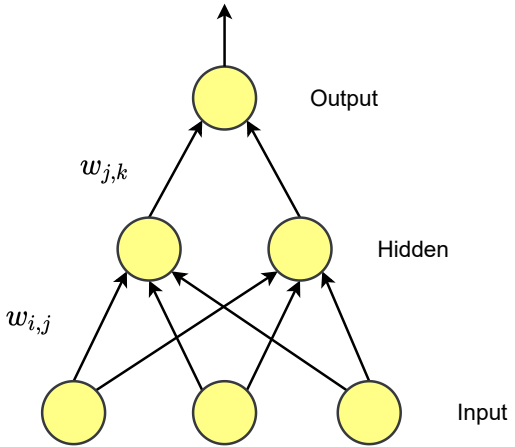
Output layer (k):

$$o_k = f\left(\sum_j w_{jk} a_j + b_k\right) \tag{1}$$



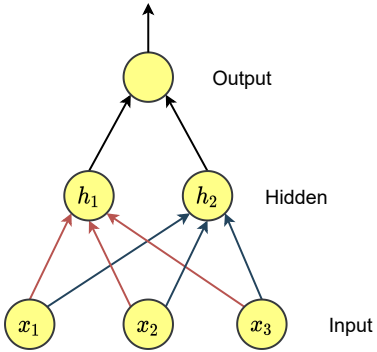
Weights

- Now the weights are labeled $w_{i,j}$, where i denotes input unit and j denotes the hidden node



Weights (2)

- The weights are stored in matrices



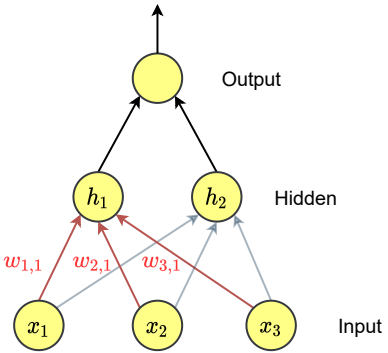
Hidden

Input

$$\begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix}$$

Weights (3)

- The weights are stored in matrices



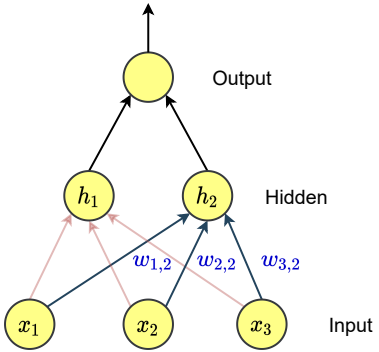
Hidden

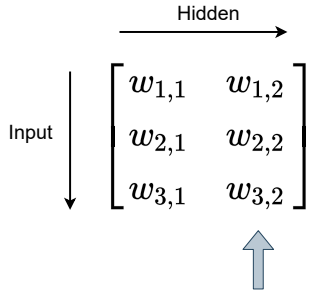
Input

$$\begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix}$$

Weights (4)

- The weights are stored in matrices





Hidden Layers

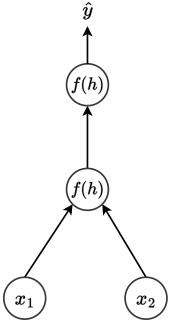
- Input to hidden node j

$$h_j = \sum_i w_{i,j} x_i \tag{2}$$

$$h_j = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix} \tag{3}$$

Exercise

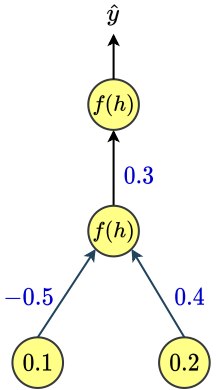
Tomando en cuenta la siguiente red neuronal multicapa:



$$X = [0.1, 0.2], W^1 = \begin{bmatrix} -0.5 \\ 0.4 \end{bmatrix}, W^2 = \begin{bmatrix} 0.3 \end{bmatrix}$$

1. Calcule la salida de la capa oculta (Utilice notación matricial).
2. Calcule la predicción de la red.

Exercise



- To do:
  - Calculate the network output

Exercise

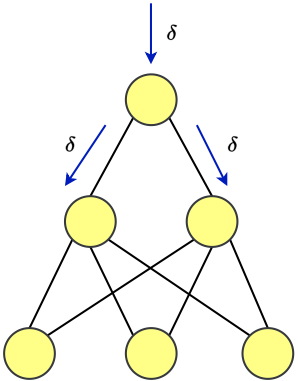
- Calculate:
  - the input to the hidden layer
  - the hidden layer output
  - the input to the output layer
  - the output of the network
- Excercise: `05_red_multicapa`

Roadmap

1. Multilayer neural networks
2. Backpropagation

Backpropagation[1]

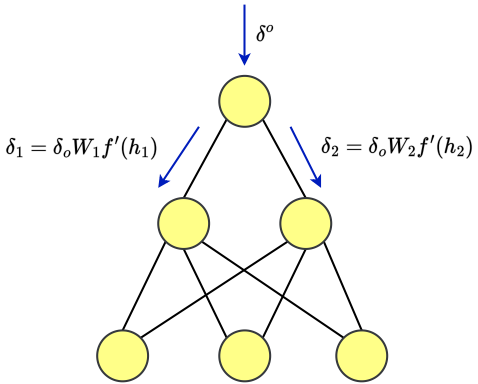
- How do we find the errors for the gradient descent step?



[1] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Backpropagation introduction

- The errors are proportional to the error in the output layer times the weight between the units.



Ready, set...

Backpropagation rules

- It is an extension of the gradient descent
- General contribution of the weight:

$$\frac{\partial E}{\partial w_{ho}} = \frac{\partial E}{\partial h_o}\frac{\partial h_o}{\partial w_{ho}} = \frac{\partial E}{\partial h_o} y_h \tag{4}$$

  where o indicates the output layer and h indicates the hidden layer.
- General update rule:

$$w^{new} = w^{old} + (-1)\eta \frac{\partial E}{w^{old}} \tag{5}$$

Backprogation implementation

- In the output layer you have errors in each unit o,
- We can generalize previous error term as δ_o,
- Term error for hidden layer node h,

$$\delta_h = \sum_o w_{h,o} \delta_o f'(h_h) \tag{6}$$

- The gradient descent step is the same,

$$\Delta w_{i,h} = \eta \delta_h y_i$$

Backprogation Algorithm

Initialization

- Set the weight steps for each layer to zero
  - The input to hidden weights $\Delta w_{h,o} = 0$
  - The hidden to output eights $\Delta W_{i,h} = 0$

Backpropagation algorithm (special case)

While we are doing gradient descent for a given number of epochs.

1. For each $x \in Data$
   1. Make a forward pass, $\hat{y} = f(xW)$
   2. Calculate the error term in the output unit $\delta_o = (y - \hat{y})f'(h)$
   3. Propagate the errors to hidden layer $\delta_h = \delta_o W_h f'(h_h)$
   4. Update the weight steps:
      $\Delta w_{h,o} = \Delta w_{h,o} + \delta_o \hat{y}_h$
      $\Delta w_{i,h} = \Delta w_{i,h} + \delta_h x_i$
2. Update the weights, where η is the learning rate and m is the number of records:
   $w_{h,o} = w_{h,o} + \frac{\eta}{m}\Delta w_{h,o}$
   $w_{i,h} = w_{i,h} + \frac{\eta}{m}\Delta w_{i,h}$

Backpropagation remarks

- Backpropagation is fundamental in neural networks.
- It is already implemented in many frameworks.
- As engineers/scientists we have to understand it very well.

Backpropagation remarks

- Backpropagation is fundamental in neural networks.
- It is already implemented in many frameworks.
- As engineers/scientists we have to understand it very well.

Exercise

- Implement the backpropagation algorithm for a network trained on the graduate school admission data
- Implement the forward pass
- Implement the backpropagation
- Update the weights

References

- **Section 6.5, Back propagation and other diferentiation algorithms.** Goodfellow, Ian, et al. Deep learning. Vol. 1. No. 2. Cambridge: MIT press, 2016.
- **The back propagation algorithm.** Buduma, N., Buduma, N., & Papa, J. (2022). Fundamentals of deep learning. " O'Reilly Media, Inc.".
- **4.6 Backpropagation.** Skansi, S. (2018). Introduction to Deep Learning: from logical calculus to artificial intelligence. Springer.
- Stewart, James, Daniel K. Clegg, and Saleem Watson. Calculus: early transcendentals. Cengage Learning, 2020.