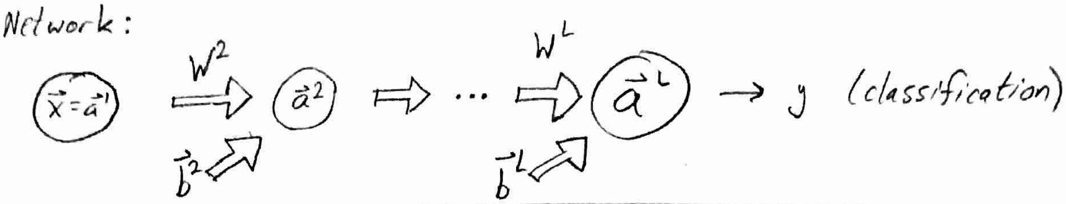
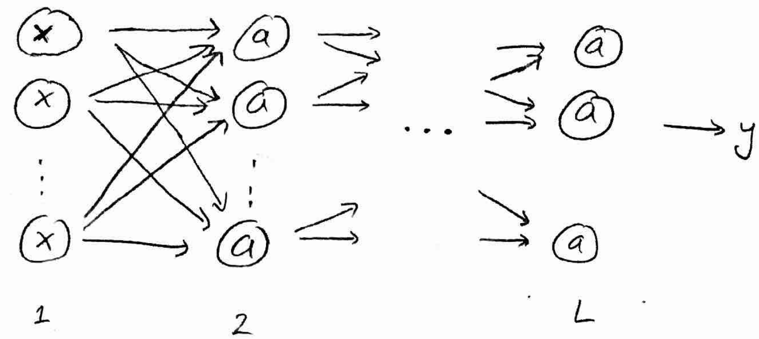
Feedforward algorithm derivation

Network:

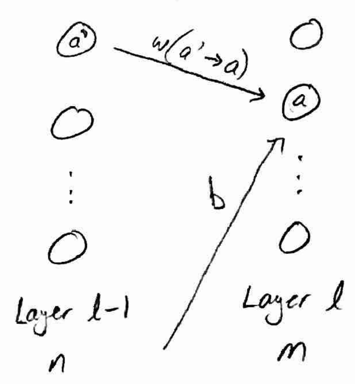




For each neuron in a layer, we weight the incoming activations from the previous layer and add a bias.

Formally, for a layer l, let n be the number of neurons in the previous layer $l-1$, and m be the number in this layer l. Let a denote activation values in this layer l, and a' denote values from the previous layer $l-1$. Finally, b denotes bias values and w denotes weights coming into this layer l. We will express w as $w(a_j' \to a_i)$ to indicate it is the weight applied to a_j' in computing a_i. (Don't worry much about indices; we'll see that they're not really necessary.)

Note: Layers are indexed 1 to L, (L = # layers).



Then for the i^{th} neuron in this layer, we compute the activation as:

$$z_i = \sum_{a'} w(a' \to a_i) \cdot a' + b_i$$

$$a_i = \sigma(z_i)$$

So, putting all of the activations a_i together $1 \le i \le m$, we get:

$$\vec{z} = \sum_{a'} \vec{w}(a' \to \vec{a}) \cdot a' + \vec{b} = W\vec{a}' + \vec{b}$$ where W is the matrix of column vectors in the sum.

$$\vec{a} = \vec{\sigma}(\vec{z})$$

(Here I am defining the matrix W as the list of column vectors shown above, and I use the notation $\vec{\sigma}(\vec{z})$ to mean $(\sigma(z_1), \ldots, \sigma(z_m))$).

To express this in terms of layer numbers (i.e. l), we'll use the superscript $\vec{a}^{\,l}$, $\vec{b}^{\,l}$, W^l, etc. to denote activations in layer l, and biases/weights coming into layer l. Then we translate the a/a' notation to:

$$\vec{z}^{\,l} = W^l \vec{a}^{\,l-1} + \vec{b}$$

$$\vec{a}^{\,l} = \vec{\sigma}(\vec{z}^{\,l})$$

The classification chosen is therefore the index of the largest activation value in the last layer, that is:

$$y = \underset{i}{\operatorname{argmax}}\ \vec{a}^{\,L}$$ where i indexes values of $\vec{a}^{\,L}$

With this info, we present the <u>feedforward algorithm</u>, which produces the activations in each layer $\vec{a}^{\,1}, \ldots, \vec{a}^{\,L}$, and the <u>classification algorithm</u>, which produces a classification.

<u>Feedforward ($\vec{x}$)</u>

- Given weight matrices $W^2, \ldots, W^L$ and bias vectors $\vec{b}^{\,2}, \ldots, \vec{b}^{\,L}$ for each layer,
- Declare vars:
  - A, the list of activations
  - $\vec{a}'$, the activations from the previous layer
  - $\vec{a}$, the activation in current layer
  - $\vec{z}$, intermediary weighted sum in current layer
- Add $\vec{x}$ to A
- $\vec{a}' = \vec{x}$
- For each $\vec{b}, W$ in bias vectors, weight matrices:
  - $\vec{z} = W\vec{a}' + \vec{b}$
  - $\vec{a} = \vec{\sigma}(\vec{z})$
  - Add $\vec{a}$ to A
  - $\vec{a}' = \vec{a}$
- Return A

Classify ($\vec{x}$)

- Declare vars:
  - $\vec{a}$, the activations in the last layer after feeding forward $\vec{x}$
  - y, the classification
- $\vec{a}$ = last item in Feedforward($\vec{x}$)
- $y = \underset{i}{\text{argmax}}\ \vec{a}$
- Return y