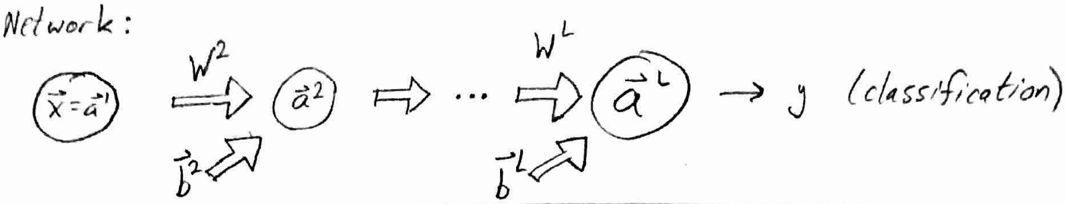
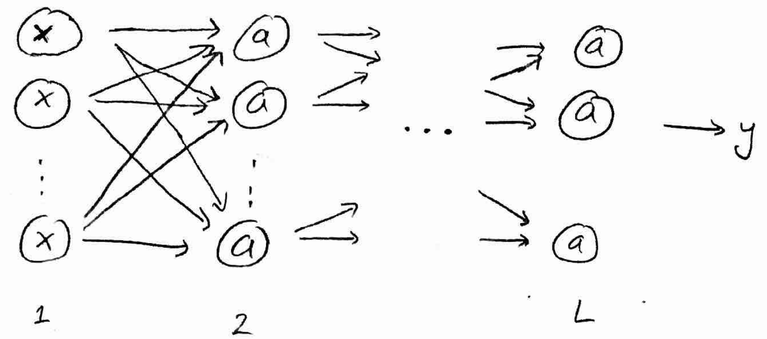
Feedforward algorithm derivation

Network:

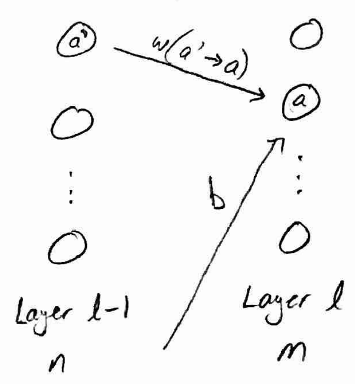




For each neuron in a layer, we weight the incoming activations from the previous layer and add a bias.

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:

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

$$a_i = \sigma(z_i)$$

2

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 $\vec{a}^L$

The formal algorithm is on a separate page.