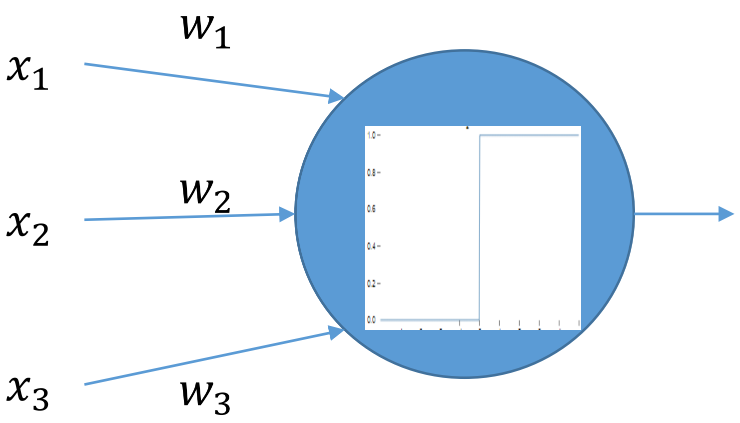
Neural Networks

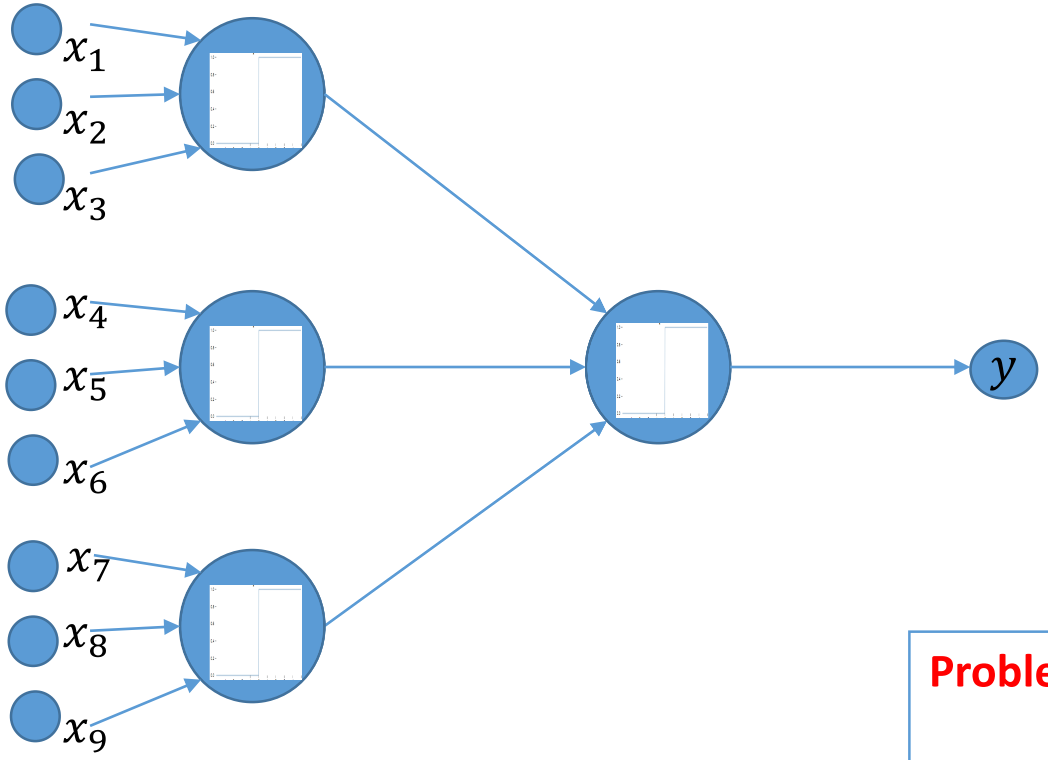
A few fine points

The main idea Behind Neural Networks

Binary Neuron: Perceptron



$$y = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \quad = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$



We train layers of the Neural Network sequentially until we get to the output.

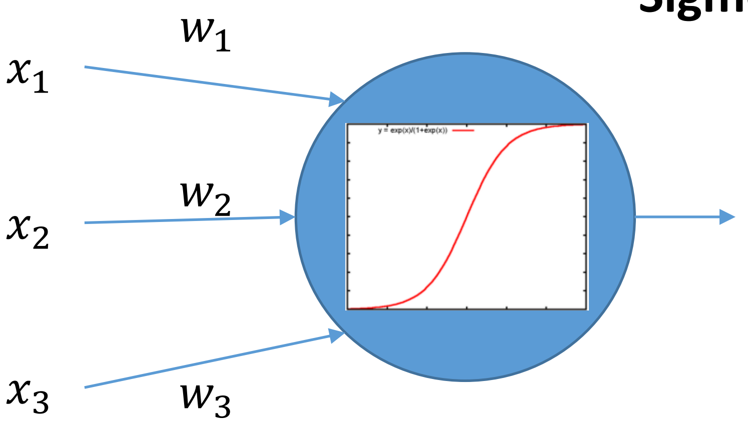
Within each layer, we can parallelize the training process

Problem:

Does not generalize well.

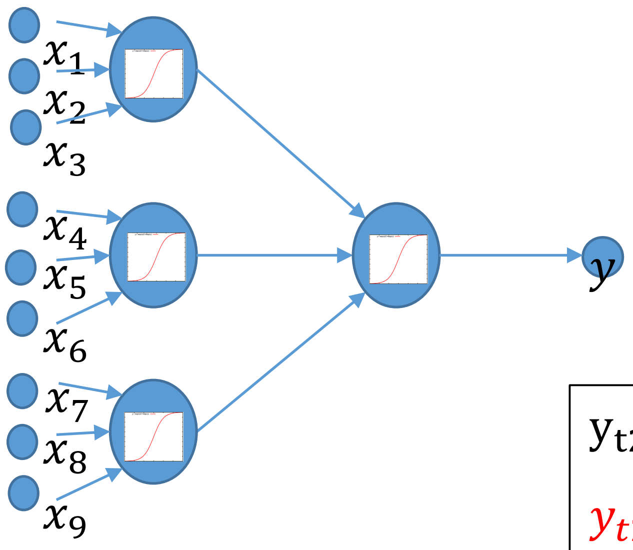
The main idea Behind Neural Networks

Sigmoid Neuron



$$y = \frac{1}{1 + \exp(-\sum_j w_j x_j - b)}$$

1. Straight-line relationship: $y = a_0 + a_1 * x$
2. Quadratic relationship: $y = a_0 + a_1 * x + a_2 * x^2$
3. Exponential relationship: $y = a_0 * e^{a_1 * x}$
4. Logarithmic relationship: $y = a_0 + a_1 * \ln(x)$
5. Power relationship: $y = a_0 * x^{a_1}$
6. Hyperbolic relationship: $y = \frac{a_0}{a_1 + a_2 * x}$
7. Sigmoid relationship: $P\{y\} = a_0 + a_1 * x$



$$y_{t1} = \frac{1}{y} - 1 \Rightarrow$$
$$y_{t1} = e^{-\sum_j (w_j * x_j) - b}$$

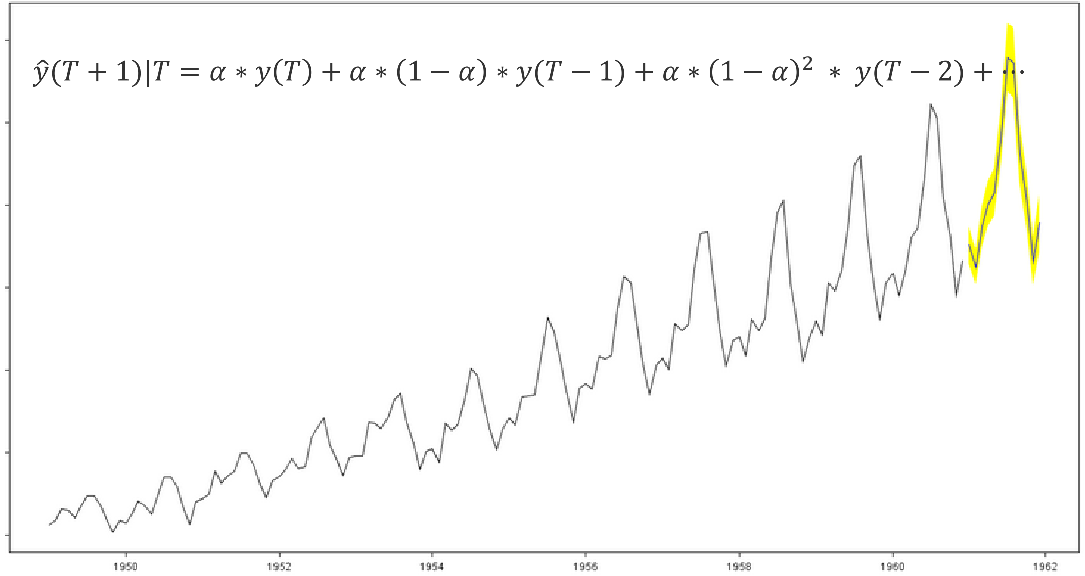
$$y_{t2} = \log(y_{t1}) \Rightarrow$$
$$y_{t2} = -\sum_j (w_j * x_j) - b$$

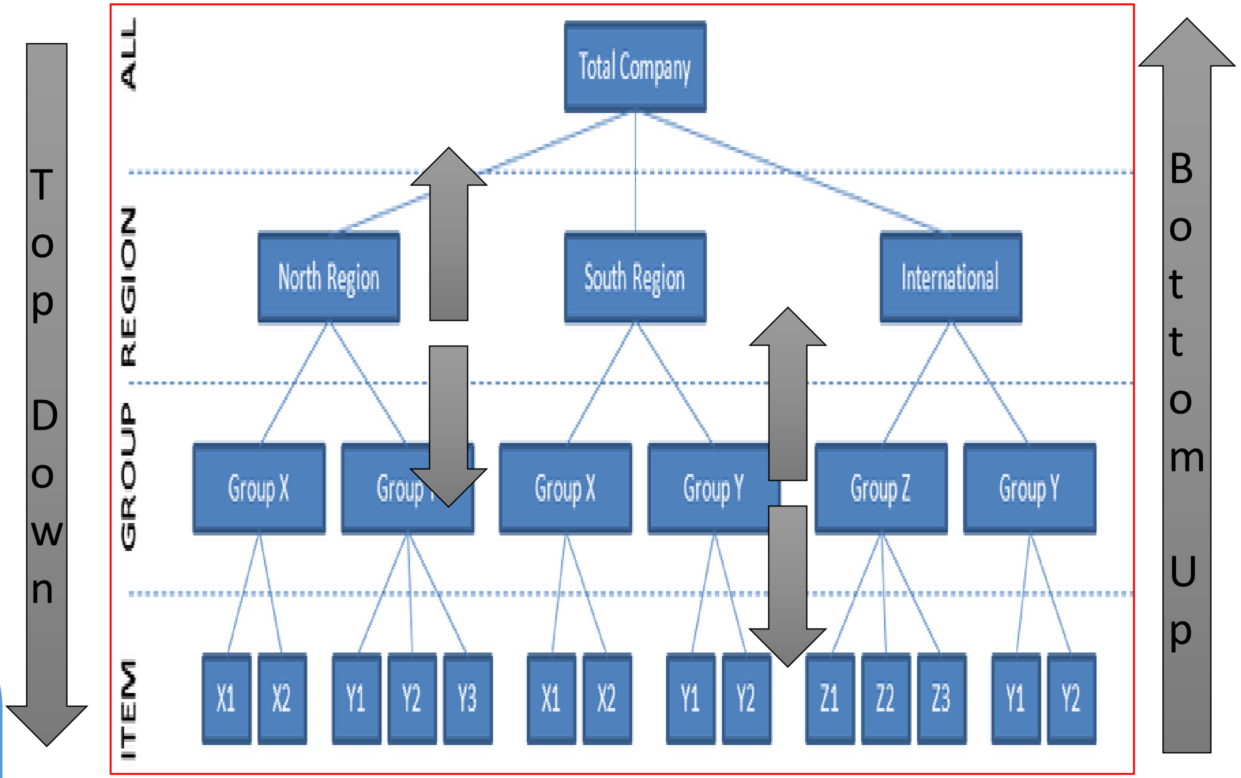
$$C(w_j, b) = \frac{1}{2 * N} * \sum_x ||y(x|w_j, b) - y_{traindata}||^2$$

$$C(w_j, b) = \frac{1}{2 * N} * \sum_x ||y_{t2}(x|w_j, b) - y_{t2\,traindata}||^2$$

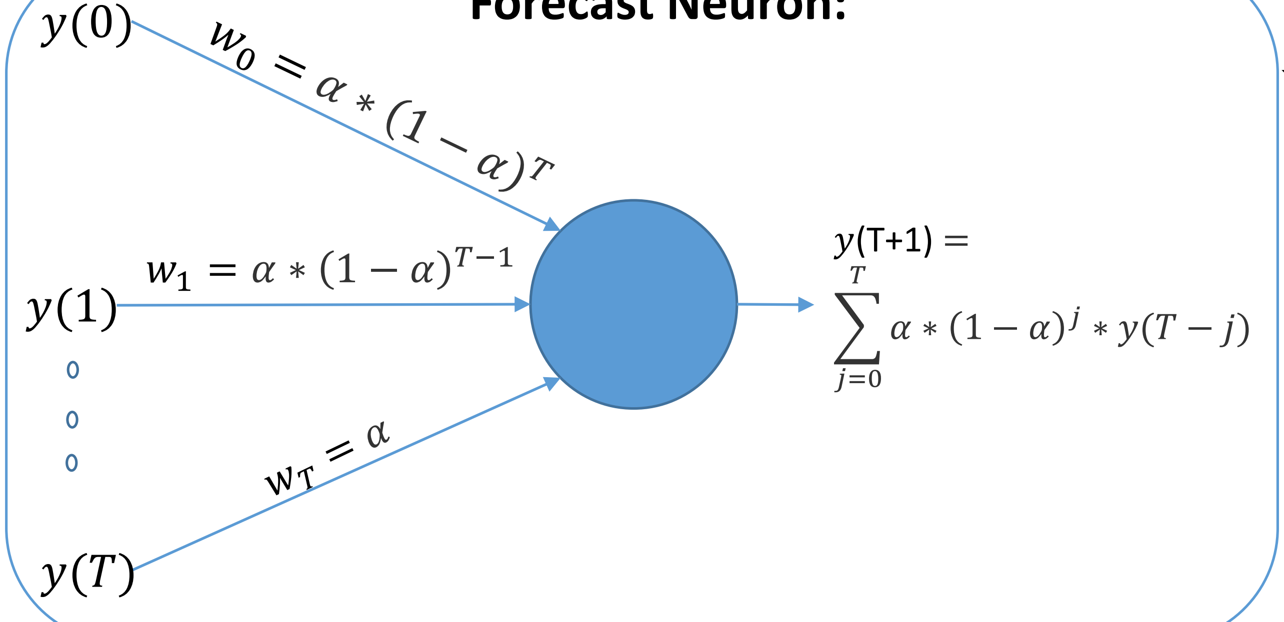
$$w_j, b = argmin\{C(w_j, b)\}$$

Hierarchical Forecasting: An unusual use case for Neural Networks





Forecast Neuron:



Forecast starting layer (top, bottom, or middle) is composed of Forecast Neurons:

$$y(T+1) = \sum_{j=0}^{T} \alpha * (1-\alpha)^j * y(T-j)$$

All other layers are composed of Linear Neurons:

$$y_l(T+1) = \sum_k \left[w_k * y_{k,l}(T+1)\right] + b$$