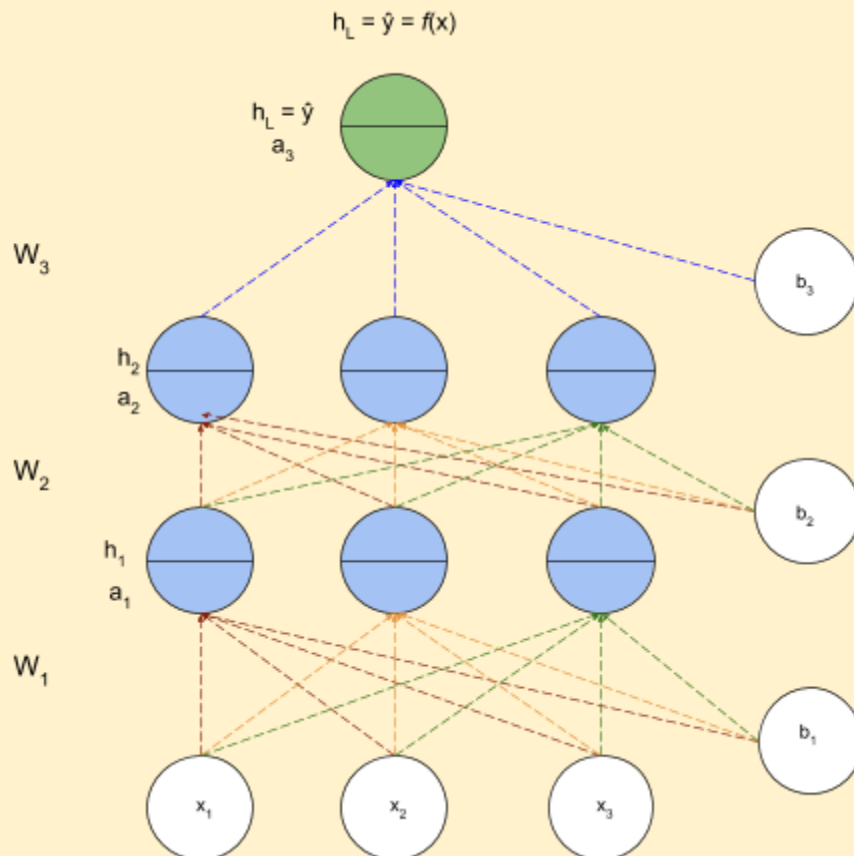


Learning Algorithm (Non-Math version)

Can we use the same Gradient Descent algorithm as before

1. We will be looking at the non-math version of the learning algorithm
2. Consider the following Neural Network



3. The algorithm
 - a. **Initialise:** $w_{111}, w_{112}, \dots, w_{313}, b_1, b_2, b_3$ randomly
 - b. **Iterate over data**
 - i. Compute \hat{y}
 - ii. Compute $L(w, b)$ Cross-entropy loss function
 - iii. $w_{111} = w_{111} - \eta \Delta w_{111}$
 - iv. $w_{112} = w_{112} - \eta \Delta w_{112}$
 - ...
 - v. $w_{313} = w_{111} - \eta \Delta w_{313}$
 - vi. $b_i = b_i + \eta \Delta b_i$
 - vii. Pytorch/Tensorflow have functions to compute $\frac{\partial l}{\partial w}$ and $\frac{\partial l}{\partial b}$
 - c. **Till satisfied**
 - i. Number of epochs is reached (ie 1000 passes/epochs)
 - ii. Continue till Loss < ϵ (some defined value)