* Explanations JS Neural
* **Generate the parameters of the model:**

>> nn.generate\_parameters() – automated build the weights and bias

>> layer1.w,layer2.w, .. , layerN.w – check the weights

* **Forward in the layer is possible through the command:**

>> layer1.forward(X\_train) – propagate from input X\_train to the layer 1

>> layer1.h,layer1.z

>> layer2.forward(layer1.h) - propagate from layer 1 to the layer 2

>> layer2.h,layer2.z

* The automatic update for minibatch is x\_loop:

>> nn.L\_model\_forward(x\_loop)

* **After declaring a network with 3 layers where:**

>> layer0 = jsneural(input\_x=X\_train,n\_nodes=2)

>> layer1 = jsneural(n\_nodes=3,activation="sigmoid")

>> layer2 = jsneural(n\_nodes=3,activation="relu")

>> layer3 = jsneural(output\_y=Y\_train,activation="softmax")

>> jsneural.generate\_parameters() – generate the weights and bias

>> layer1.update(n\_nodes=5,activation="relu") – update parameters of layer1