

Single Hidden Layer Neural Net with Sigmoid Output Layer: The neural network (function approximator) is just a chain of geometric transformations (functions) each parametrized by $W \in R^{n_x \times n_h}$ on $X \in R^{m \times n_x}$. Note that m is the number of examples in the dataset, n_x is the input size (i.e., output of the previous layer), and n_h is number of hidden units in the current layer. Also θ refers to the weights W and biases \vec{b} in the network.

$$\begin{aligned}
NeuralNet_{\theta}(X) &= \sigma((XW^{[1]} + \vec{b}^{[1]})W^{[2]} + \vec{b}^{[2]}) \\
&= f(g(w)) \\
g_{\theta^{[2]}}(A) &= AW^{[2]} + \vec{b}^{[2]} \\
w_{\theta^{[1]}}(A) &= AW^{[1]} + \vec{b}^{[1]} \\
f(t) = \sigma(t) &= \frac{1}{1 + e^{-t}}
\end{aligned} \tag{1}$$

Neural Network Prediction

$$\hat{y} \leftarrow \tag{2}$$

Gradient Update

$$\theta_i \leftarrow \theta_i - \eta(\nabla_{\theta} \mathcal{L}_{\theta}(\hat{y}, y)) \tag{3}$$