Problem 1 (Impact of non-linear activation functions on Learning):

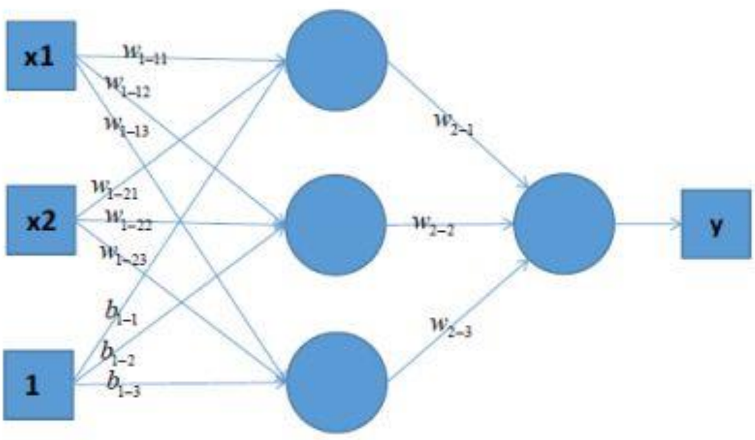
Show that a feed-forward neural network with linear activation function and any number of hidden layers is equivalent to just a linear neural network with no hidden layer.

Intuition and explanation:

The forward propagation of the neural network is a process of projecting into the feature space of the hidden layer; if the hidden layer has infinite nodes, the inputs can project to an infinitely high dimension. However, if the activation function is linear instead of non-linear, this is exactly equivalent to multiplying the original input data with the weight matrix plus the bias term. The linear combinations of linear functions is still a linear function, which can derive and prove mathematically. Thus, no matter how high dimensional the space is projected; the data distribution will only be applied to three linear changes: rotation, translation, and scaling. Linearly indivisible data cannot be classified simply based on linear functions.

If we do not use an activation function (actually the activation function is f(x) = w*x+b), in this case, the input of each layer of the node is a linear function of the output of the upper layer. No matter how many layers the neural network has, the output is a linear combination of inputs, which is equivalent to the effect without a hidden layer. In this case, it is the most primitive Perceptron; the network's approximation ability is quite limited. Because of the above reasons, introducing a non-linear function as the activation function so that the deep neural network approximation ability is more powerful, which is no longer a linear combination of inputs, but can approximate almost any complex function to the original inputs.

Perceptron with one hidden layer



$$y = w_{2\text{-}1}(w_{1\text{-}11}x_1 + w_{1-21}x_2 + b_{1\text{-}1})$$
$$+ w_{2\text{-}2}(w_{1\text{-}12}x_1 + w_{1-22}x_2 + b_{1\text{-}2})$$
$$+ w_{2\text{-}3}(w_{1\text{-}13}x_1 + w_{1-23}x_2 + b_{1\text{-}3})$$

Reference:

1. https://blog.csdn.net/tyhj_sf/article/details/79932893?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task
2. https://www.zhihu.com/question/22334626
3. https://www.zhihu.com/question/29021768

See the next page for the detailed derivation:

Problem 1

A feed-forward neural network with linear activation and any number of hidden layers is equivalent to just a linear neural network with no hidden layer.

For example, let's consider the neural network below, which has two hidden layers and linear activation $f(x)$, $g(x)$, $P(x)$

$$\textcircled{x} \xrightarrow[w_1]{} \boxed{h_1 = f(xw_1+b_1)} \xrightarrow[W_2]{} \boxed{h_2 = g(h_1 w_2 + b_2)} \xrightarrow[W_3]{} \boxed{y = P(h_2 w_3 + b_3)}$$

for all the linear functions, we can represent them as the following form $f(x) = w'x + b'$

Then:

$$h_1 = w_1' \cdot (w_1 x + b_1) + b_1' = w_1' \cdot w_1 x + w_1' b_1 + b_1'$$

$$= \bar{w}_1 x + \bar{b}_1 \quad (w_1' \cdot w_1 = \bar{w}_1,\ w_1' b_1 + b_1' = \bar{b}_1)$$

$$h_2 = w_2' \cdot (w_2 \cdot h_1 + b_2) + b_2' = w_2' \cdot w_2 \cdot h_1 + w_2' b_2 + b_2'$$

$$= \bar{w}_2 h_1 + \bar{b}_2$$

$$= \bar{w}_2 \cdot \bar{w}_1 \cdot x + \bar{w}_2 \bar{b}_1 + \bar{b}_2$$

$$y = w_3' (w_3 \cdot h_2 + b_3) + b_3' = w_3' w_3 \cdot h_2 + w_3' b_3 + b_3'$$

$$= \bar{w}_3 h_2 + w_3' b_3 + b_3'$$

$$= \bar{w}_3 \cdot \bar{w}_2 \cdot \bar{w}_1 x + \bar{w}_3 \cdot \bar{w}_2 \bar{b}_1 + \bar{w}_3 \bar{b}_2 + \bar{b}_3$$

$$= W_3 x + B_3$$

Generally, when we have n hidden layers.

$$y = \bar{w}_{n+1} \cdot \bar{w}_n \cdot \ldots \cdot \bar{w}_1 x + \bar{w}_{n+1} \cdot \bar{w}_n \cdot \ldots \cdot \bar{w}_2 \bar{b}_1 + \bar{w}_{n+1} \cdot \bar{w}_n \cdot \ldots \cdot \bar{w}_3 \bar{b}_2$$

$$+ \ldots + \bar{w}_{n+1} \cdot \bar{b}_n + \bar{b}_{n+1}$$

$$= W_{n+1} x + B_{n+1}$$

As a result, if we only use linear activation function, the problem will turn into linear regression no matter how many hidden layers we add.