Neural Networks :

The linear models we covered so far are based on linear combinations of some fixed basis functions / features :

$$y = \sigma\Big(\sum_{i=1}^{d} \underbrace{w_i}_{\text{model params}} \overbrace{\varphi_i(x)}^{\text{fixed}}\Big) + \varepsilon$$

↳ features / basis functions

- σ : linear, φ : identity → L… r…
- σ : linear, φ : nonlinear → Line… regr… with b… fun…
- σ : logistic sigmoid, φ : identity → Logi… regr…

Goal : Allow the features / basis functions $\varphi_i(x)$ to depend on parameters, which can be "trained" / adjusted (along with the w_i) from the observed data.

This idea leads to the basic Multi-Layer Perceptron (MLP) model which can be described as a series of functional transformations :

$$y = \underline{w}^T \varphi(x; \underline{\vartheta}) + \varepsilon$$

↳ Non-linear regression model

, with model params : $\Theta := \{w, \vartheta\}$

Forward pass for an MLP model :

$$h_q^{(1)} = f^{(1)}\Big(\underbrace{\sum_{i=1}^{d} w_{iq}^{(1)} x_i + b_q^{(1)}}_{\text{linear transform.}}\Big) \quad , \quad x_i \in \mathbb{R}^d \;, \; q = 1, \dots, Q^{(1)}$$

$h_q^{(1)}$ ↓ activation in the 1st layer

$f^{(1)}$ ↓ non-linear function

$h \in \mathbb{R}^{Q^{(1)}}$

⋮

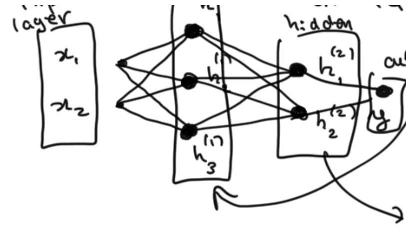
$$h_q^{(2)} = f^{(2)}\Big(\sum_{i=1}^{Q^{(1)}} w_{iq}^{(2)} h_i^{(1)} + b_q^{(2)}\Big)$$

for e.g. assume : $x \in \mathbb{R}^2$

$x = (x_1, x_2)$

hidden $\dim(Q^{(1)}) = 3$

input $h^{(1)}$ $\dim(Q^{(2)}) = 2$



$$H^{(1)} = \begin{bmatrix} h_1^{(1)} \\ \vdots \\ h_{Q^{(1)}}^{(1)} \end{bmatrix} \qquad H^{(2)} = \begin{bmatrix} h_1^{(2)} \\ \vdots \\ h_{Q^{(2)}}^{(2)} \end{bmatrix}$$

In matrix-vector notation:

1st hidden layer:

$$\underset{n\times Q^{(1)}}{H^{(1)}} = f^{(1)}(\underset{n\times d}{X} W^{(1)} + b^{(1)}), \quad \underset{n\times d}{X} = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & & \\ x_{n1} & \cdots & x_{nd} \end{bmatrix}$$

$$\underset{n\times Q^{(1)}}{H^{(1)}} = \begin{bmatrix} h_{11}^{(1)} & \cdots & h_{1Q^{(1)}}^{(1)} \\ \vdots & & \\ h_{n1}^{(1)} & \cdots & h_{nQ^{(1)}}^{(1)} \end{bmatrix}, \quad \underset{d\times Q^{(1)}}{W} = \begin{bmatrix} W_{11} & \cdots & W_{1Q^{(1)}} \\ \vdots & & \\ W_{d1} & \cdots & W_{dQ^{(1)}} \end{bmatrix}, \quad \underset{1\times Q^{(1)}}{b^{(1)}} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_Q \end{bmatrix}$$

2nd hidden layer:

$$\underset{n\times Q^{(2)}}{H^{(2)}} = f^{(2)}(\overbrace{H^{(1)} W^{(2)}}^{n\times Q^{(2)}} + b^{(2)})$$

$$\underset{Q^{(1)}\times Q^{(2)}}{W^{(2)}} \hookrightarrow \text{weights}, \quad \underset{1\times Q^{(2)}}{b^{(2)}} \hookrightarrow \text{biases}$$

$\vdots$

Output layer:

$$\underset{n\times s}{Y} = f^{(o)}(\underbrace{\underset{n\times Q^{(L)}}{H^{(L)}} \underset{Q^{(L)}\times s}{W^{(o)}}}_{n\times s} + \underset{1\times s}{b^{(o)}})$$

In practice the bias is added via broadcasting

$$\hookrightarrow \underset{n\times s}{b^{(o)}} = \begin{bmatrix} b_1^{(o)} & \cdots & b_s^{(o)} \\ b_1^{(o)} & \cdots & b_s^{(o)} \\ \vdots & & \\ b_1^{(o)} & \cdots & b_s^{(o)} \end{bmatrix}$$

Setup: Given data:

$$\mathcal{D} := \{(x_1, y_1), \ldots, (x_n, y_n)\},$$

$$x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}^s$$

We have to make the following choices:

1.) # of hidden layers (L) (i.e. how "deep" is our model)

2.) Dimensionality / # of "neurons": $Q^{(1)}, \ldots, Q^{(L)}$ (i.e. how wide is every hidden layer)

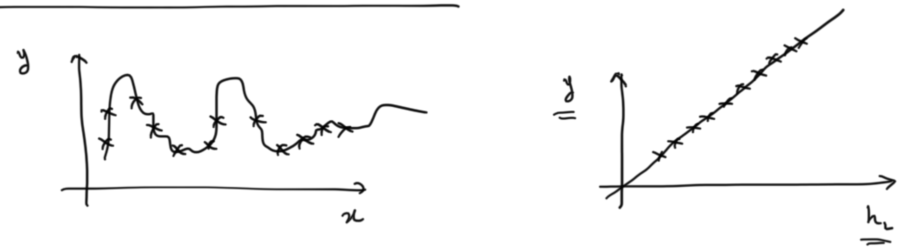
3.) Activation functions: $\underbrace{f^{(1)}, f^{(2)}, \ldots, f^{(L)}}, \underbrace{f^{(o)}}$

typically the same   depends on the task

Choices

#1, 2, 3 : define the MLP architecture.

Intuition on regression :



$$x \longrightarrow h_1 \longrightarrow h_2 \longrightarrow \dots \longrightarrow h_L \longrightarrow y$$

$x \in \mathbb{R}^1$ $\in \mathbb{R}^1$ $\in \mathbb{R}^1$ $\in \mathbb{R}^1$

- Most common activation functions for the output layer:

  - Linear layer : $\underset{n\times s}{Y} = f(\underbrace{\underset{n\times Q^{(L)}}{H^{(L)}}\ \underset{Q^{(L)}\times s}{W^{(o)}} + \underset{1\times s}{b^{(o)}}})$, where $f(x) = x$

    typical choise for regression

  - Logistic sigmoid : $\underset{n\times 2}{Y} = f(H^{(L)} W^{(o)} + b^{(o)})$, $f(x) = \dfrac{1}{1+e^{-x}}$

    typical choice for binary classification.

  - Soft-max $\underset{n\times q}{Y} = f(H^{(L)} W^{(o)} + b^{(o)})$

    typical choice for multi-class classification. , $f(x_i) = \dfrac{\exp(x_i)}{\sum_{j=1}^{q} \exp(x_j)}$