Neural networks

10.1 Draw a neural network

Draw the graph corresponding to a two-layer dense neural network for regression with p input variables $x_1, \dots, x_p$, one hidden layer with U units, activation function $h : \mathbb{R} \mapsto \mathbb{R}$ in the layer, and output $\widehat{y} \in \mathbb{R}$. How many parameters does the model have (including offsets)?

10.2 Vectorization over units

Mathematically, the model above can be described as

$$q_i = h\left(b_i^{(1)} + \sum_{j=1}^{p} W_{ij}^{(1)} x_j\right), \qquad i = 1, \dots, U \tag{10.1a}$$

$$\widehat{y} = b^{(2)} + \sum_{\ell=1}^{U} W_\ell^{(2)} q_\ell \tag{10.1b}$$

where each node contains a linear regression model, and where each node in the hidden layer is squeezed through an activation function h. When we implement this in code, we prefer to use a *vectorized* version of these equations since it runs faster than loops and explicit sums. Vectorize the equations in (10.1) by introducing the variables

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_U \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} b_1^{(1)} \\ \vdots \\ b_U^{(1)} \end{bmatrix}, \quad \mathbf{W}^{(1)} = \begin{bmatrix} W_{11}^{(1)} & \dots & W_{1p}^{(1)} \\ \vdots & & \vdots \\ W_{U1}^{(1)} & \dots & W_{Up}^{(1)} \end{bmatrix}, \quad \mathbf{b}^{(2)} = \begin{bmatrix} b^{(2)} \end{bmatrix}, \quad \mathbf{W}^{(2)} = \begin{bmatrix} W_1^{(2)} & \dots & W_U^{(2)} \end{bmatrix},$$

i.e. write (10.1) on a matrix form

$$\widehat{y} = f(\mathbf{x}), \tag{10.2}$$

which does not include any explicit summation $\sum$ or looping over $\ell = 1, \dots, U$.

10.3 Vectorization over data points

When processing many data points $\{\mathbf{x}_i\}_{i=1}^n$, we have the relation (10.2) for each data point $i = 1, \dots, n$

$$\widehat{\mathbf{y}}_i = f(\mathbf{x}_i), \qquad i = 1, \dots, n. \tag{10.3}$$

Vectorize the equations in (10.3) by introducing the variables

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{bmatrix}, \quad \widehat{\mathbf{Y}} = \begin{bmatrix} \widehat{\mathbf{y}}_1^\mathsf{T} \\ \vdots \\ \widehat{\mathbf{y}}_n^\mathsf{T} \end{bmatrix}, \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^\mathsf{T} \\ \vdots \\ \mathbf{q}_n^\mathsf{T} \end{bmatrix}, \tag{10.4}$$

i.e. write (10.3) on a matrix form

$$\widehat{\mathbf{Y}} = f(\mathbf{X}), \tag{10.5}$$

which does not include any explicit looping $i = 1, \dots, n$.

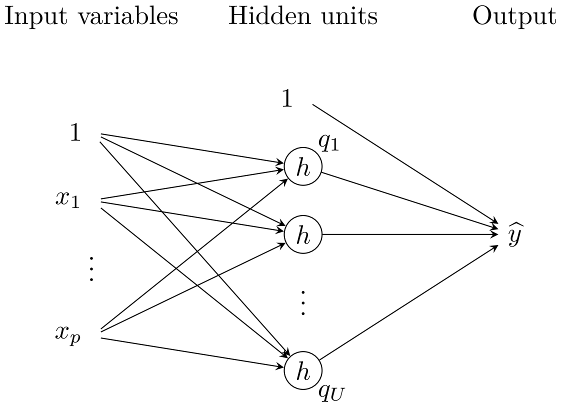
10.4 **Linear activation function**

Show that the model in the (10.1) reduces to a linear regression model if $h(x) = x$. Specifically, show how the parameters of the entire neural network relate to the parameters of the single linear regression model

$$\widehat{y} = \theta_0 + \sum_{j=1}^{p} \theta_j x_j.$$

Solutions

10.1 A dense neural network for regression with one hidden layer can be illustrated as



Each link represents a multiplication of its incoming unit with a parameter. The parameters are different for each link. The number of links in the graph (=number of parameters in the model) is consequently $(p+1) \cdot U + 1 + U$.

10.2 By stacking the equations in (10.1a) as rows in vectors we get

$$\begin{bmatrix} q_1 \\ \vdots \\ q_U \end{bmatrix} = \begin{bmatrix} h\left(b_1^{(1)} + \sum_{j=1}^{p} W_{1j}^{(1)} x_j\right) \\ \vdots \\ h\left(b_U^{(1)} + \sum_{j=1}^{p} W_{Uj}^{(1)} x_j\right) \end{bmatrix} \tag{10.6a}$$

$$\widehat{y} = b^{(2)} + \sum_{i=1}^{U} W_i^{(2)} q_i \tag{10.6b}$$

Further, by replacing the summations with matrix-vector multiplications we can write this as

$$\underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_U \end{bmatrix}}_{\mathbf{q}} = h\left(\underbrace{\begin{bmatrix} W_{11}^{(1)} & \dots & W_{1p}^{(1)} \\ \vdots & & \vdots \\ W_{U1}^{(1)} & \dots & W_{Up}^{(1)} \end{bmatrix}}_{\mathbf{W}^{(1)}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} b_1^{(1)} \\ \vdots \\ b_U^{(1)} \end{bmatrix}}_{\mathbf{b}^{(1)}}\right), \tag{10.7a}$$

$$\widehat{y} = \underbrace{\begin{bmatrix} W_1^{(2)} & \dots & W_U^{(2)} \end{bmatrix}}_{\mathbf{W}^{(2)}} \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_U \end{bmatrix}}_{\mathbf{q}} + \underbrace{\begin{bmatrix} b^{(2)} \end{bmatrix}}_{\mathbf{b}^{(2)}}. \tag{10.7b}$$

By identifying all the matrices and vectors, we can in a more compact and vectorized manner write these equations as

$$\mathbf{q} = h\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right), \tag{10.8a}$$

$$\widehat{y} = \mathbf{W}^{(2)}\mathbf{q} + \mathbf{b}^{(2)} \tag{10.8b}$$

Note that the activation function h acts element-wise here.

10.3 Following the solutions from the previous exercise, the equations we are about to vecotrize are

$$\mathbf{q}_i = h\left(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}\right), \qquad i = 1, \dots, n, \tag{10.9a}$$

$$\widehat{\mathbf{y}}_i = \mathbf{W}^{(2)}\mathbf{q}_i + \mathbf{b}^{(2)}, \qquad i = 1, \dots, n. \tag{10.9b}$$

We start by taking the transpose of these equations

$$\mathbf{q}_i^\mathsf{T} = h\left(\mathbf{x}_i^\mathsf{T}\mathbf{W}^{(1)\mathsf{T}} + \mathbf{b}^{(1)\mathsf{T}}\right), \qquad i = 1, \dots, n, \tag{10.10a}$$

$$\widehat{\mathbf{y}}_i^\mathsf{T} = \mathbf{q}_i^\mathsf{T}\mathbf{W}^{(2)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}}, \qquad i = 1, \dots, n. \tag{10.10b}$$

We continue by stacking these equations in rows

$$\begin{bmatrix} \mathbf{q}_1^\mathsf{T} \\ \vdots \\ \mathbf{q}_n^\mathsf{T} \end{bmatrix} = \begin{bmatrix} h\left(\mathbf{x}_1^\mathsf{T}\mathbf{W}^{(1)\mathsf{T}} + \mathbf{b}^{(1)\mathsf{T}}\right) \\ \vdots \\ h\left(\mathbf{x}_n^\mathsf{T}\mathbf{W}^{(1)\mathsf{T}} + \mathbf{b}^{(1)\mathsf{T}}\right) \end{bmatrix}, \tag{10.11a}$$

$$\begin{bmatrix} \widehat{\mathbf{y}}_1^\mathsf{T} \\ \vdots \\ \widehat{\mathbf{y}}_n^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^\mathsf{T}\mathbf{W}^{(2)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}} \\ \vdots \\ \mathbf{q}_n^\mathsf{T}\mathbf{W}^{(2)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}} \end{bmatrix}. \tag{10.11b}$$

By bringing the weight matrices and offset vectors outside we get

$$\begin{bmatrix} \mathbf{q}_1^\mathsf{T} \\ \vdots \\ \mathbf{q}_n^\mathsf{T} \end{bmatrix} = h\left(\begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{bmatrix} \mathbf{W}^{(1)\mathsf{T}} + \mathbf{b}^{(1)\mathsf{T}}\right), \tag{10.12a}$$

$$\begin{bmatrix} \widehat{\mathbf{y}}_1^\mathsf{T} \\ \vdots \\ \widehat{\mathbf{y}}_n^\mathsf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^\mathsf{T} \\ \vdots \\ \mathbf{q}_n^\mathsf{T} \end{bmatrix} \mathbf{W}^{(2)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}}. \tag{10.12b}$$

Note that in these equations we add $+\mathbf{b}^{(1)\mathsf{T}}$ and $+\mathbf{b}^{(2)\mathsf{T}}$ to each row. In python-language we call that *broadcasting* and is heavily used in deep learning implementations.

Now we can identify the matrices in the problem description and write these equations as

$$\mathbf{Q} = h\left(\mathbf{X}\mathbf{W}^{(1)\mathsf{T}} + \mathbf{b}^{(1)\mathsf{T}}\right), \tag{10.13a}$$

$$\widehat{\mathbf{Y}} = \mathbf{Q}\mathbf{W}^{(2)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}}, \tag{10.13b}$$

which now is in a format suitable for efficient vectorized implementation in code. For the implementation you might want to consider using the transposed version of $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ as your weight matrices and offset vectors to avoid taking transpose in each layer.

10.4 **Alternative 1**

The mathematical model corresponding to the neural network above is

$$q_i = h\left(b_i^{(1)} + \sum_{j=1}^{p} W_{ij}^{(1)} x_j\right), \qquad i = 1, \dots, U$$

$$\widehat{y} = b^{(2)} + \sum_{\ell=1}^{U} W_\ell^{(2)} q_\ell$$

If we consider a linear activation function $h(x) = x$ in (10.1) we get

$$\begin{aligned} \widehat{y} &= b^{(2)} + \sum_{\ell=1}^{U} W_\ell^{(2)} \left(b_\ell^{(1)} + \sum_{j=1}^{p} W_{\ell j}^{(1)} x_j\right) \\ &= b^{(2)} + \sum_{\ell=1}^{U} W_\ell^{(2)} b_\ell^{(1)} + \sum_{j=1}^{p}\sum_{\ell=1}^{U} W_\ell^{(2)} W_{\ell j}^{(1)} x_j, \end{aligned}$$

which is a linear regression model

$$\widehat{y} = \theta_0 + \sum_{j=1}^{p} \theta_j x_j$$

where

$$\theta_0 = b^{(2)} + \sum_{\ell=1}^{U} W_\ell^{(2)} b_\ell^{(1)} \qquad \text{and} \qquad \theta_j = \sum_{\ell=1}^{U} W_\ell^{(2)} W_{\ell j}^{(1)}. \tag{10.14}$$

Alternative 2

We can also solve the exercise by starting from the vectorized version of the model in (10.2). With $h(x) = x$ we get

$$\mathbf{q} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}, \tag{10.15a}$$
$$\widehat{y} = \mathbf{W}^{(2)}\mathbf{q} + \mathbf{b}^{(2)} \tag{10.15b}$$

which gives us

$$\widehat{y} = \mathbf{W}^{(2)}\mathbf{W}^{(1)}\mathbf{x} + \mathbf{W}^{(2)}\mathbf{b}^{(1)} + \mathbf{b}^{(2)} \tag{10.16}$$

This can be compared with the linear regression model

$$\begin{aligned}\widehat{y} &= \theta_0 + \sum_{j=1}^{p} \theta_j x_j \\ &= \boldsymbol{\theta}_{-0}^{\mathsf{T}}\mathbf{x} + \theta_0 \end{aligned} \tag{10.17}$$

where $\boldsymbol{\theta}_{-0} = [\theta_1, \ , \dots, \ \theta_p]^{\mathsf{T}}$. By comparing (10.16) and (10.17) we get that

$$\boldsymbol{\theta}_{-0} = \mathbf{W}^{(1)\mathsf{T}}\mathbf{W}^{(2)\mathsf{T}}, \qquad \text{and} \qquad \theta_0 = \mathbf{W}^{(2)}\mathbf{b}^{(1)} + \mathbf{b}^{(2)} \tag{10.18}$$

and by using the definitions in Exercise 10.2 this is

$$\begin{bmatrix}\theta_1 \\ \vdots \\ \theta_p\end{bmatrix} = \begin{bmatrix} W_{11}^{(1)} & \dots & W_{U1}^{(1)} \\ \vdots & & \vdots \\ W_{1p}^{(1)} & \dots & W_{Up}^{(1)} \end{bmatrix} \begin{bmatrix} W_1^{(2)} \\ \vdots \\ W_U^{(2)} \end{bmatrix}, \qquad \text{and} \qquad \theta_0 = \begin{bmatrix} W_1^{(2)} & \dots & W_U^{(2)} \end{bmatrix} \begin{bmatrix} b_1^{(1)} \\ \vdots \\ b_U^{(1)} \end{bmatrix} + b^{(2)} \tag{10.19}$$

which is equivalent with the solution in (10.14).