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 M units, activation function $\sigma : \mathbb{R} \mapsto \mathbb{R}$ in the layer, and output $z \in \mathbb{R}$. How many parameters does the model have (including offsets)?

10.2 Vectorization over units

Mathematically, the model above can be described as

$$h_m = \sigma\left(\beta_{0m}^{(1)} + \sum_{j=1}^{p} \beta_{jm}^{(1)} x_j\right), \qquad m = 1, \dots, M \tag{10.1a}$$

$$z = \beta_0^{(2)} + \sum_{m=1}^{M} \beta_m^{(2)} h_m \tag{10.1b}$$

where each node contains a linear regression model, and where each node in the hidden layer is squeezed through an activation function σ. 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

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} \beta_{01}^{(1)} & \dots & \beta_{0M}^{(1)} \end{bmatrix}, \quad \mathbf{W}^{(1)} = \begin{bmatrix} \beta_{11}^{(1)} & \dots & \beta_{1M}^{(1)} \\ \vdots & & \vdots \\ \beta_{p1}^{(1)} & \dots & \beta_{pM}^{(1)} \end{bmatrix}, \quad \mathbf{b}^{(2)} = \begin{bmatrix} \beta_0^{(2)} \end{bmatrix}, \quad \mathbf{W}^{(2)} = \begin{bmatrix} \beta_1^{(2)} \\ \vdots \\ \beta_M^{(2)} \end{bmatrix},$$

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

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

which does not include any explicit summation $\sum$ or looping $m = 1, \dots, M$.

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 $i = 1, \dots, n$

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

Vectorize the equations in (10.3) by introducing the variables

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \quad \text{and} \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1^\mathsf{T} \\ \vdots \\ \mathbf{h}_n^\mathsf{T} \end{bmatrix}, \tag{10.4}$$

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

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

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

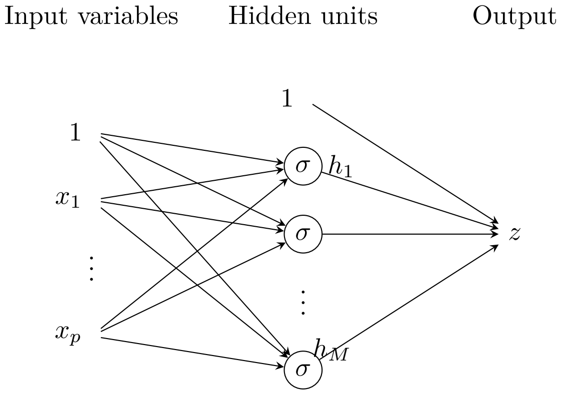
10.4 Linear activation function

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

$$z = \beta_0 + \sum_{j=1}^{p} \beta_j x_j.$$

Solutions

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



Each link represents a multiplication of its incomming 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 $(p+1)\cdot M + 1 + M$.

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

$$\begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix} = \begin{bmatrix} \sigma\left(\beta_{01}^{(1)} + \sum_{j=1}^{p} \beta_{j1}^{(1)} x_j\right) \\ \vdots \\ \sigma\left(\beta_{0M}^{(1)} + \sum_{j=1}^{p} \beta_{jM}^{(1)} x_j\right) \end{bmatrix} \tag{10.6a}$$

$$z = \beta_0^{(2)} + \sum_{m=1}^{M} \beta_m^{(2)} h_m \tag{10.6b}$$

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

$$\underbrace{\begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix}}_{\mathbf{h}} = \sigma\left(\underbrace{\begin{bmatrix} \beta_{11}^{(1)} & \dots & \beta_{p1}^{(1)} \\ \vdots & & \vdots \\ \beta_{1M}^{(1)} & \dots & \beta_{pM}^{(1)} \end{bmatrix}}_{\mathbf{W}^{(1)\mathsf{T}}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \beta_{01}^{(1)} \\ \vdots \\ \beta_{0M}^{(1)} \end{bmatrix}}_{\mathbf{b}^{(1)\mathsf{T}}} \right), \tag{10.7a}$$

$$z = \underbrace{\begin{bmatrix} \beta_1^{(2)} & \dots & \beta_M^{(2)} \end{bmatrix}}_{\mathbf{W}^{(2)\mathsf{T}}} \underbrace{\begin{bmatrix} h_1 \\ \vdots \\ h_M \end{bmatrix}}_{\mathbf{h}} + \underbrace{\begin{bmatrix} \beta_0^{(2)} \end{bmatrix}}_{\mathbf{b}^{(2)\mathsf{T}}}. \tag{10.7b}$$

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

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

$$z = \mathbf{W}^{(2)\mathsf{T}}\mathbf{h} + \mathbf{b}^{(2)\mathsf{T}} \tag{10.8b}$$

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

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

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

$$z_i = \mathbf{W}^{(2)\mathsf{T}}\mathbf{h}_i + \mathbf{b}^{(2)\mathsf{T}}, \qquad i = 1, \dots, n. \tag{10.9b}$$

We start by taking the transpose of these equations

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

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

We continue by stacking these equations in rows

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

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^\mathsf{T}\mathbf{W}^{(2)} + \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{h}_n^\mathsf{T}\mathbf{W}^{(2)} + \mathbf{b}^{(2)} \end{bmatrix}. \tag{10.11b}$$

By bringing the weight matrices and offset vectors outside we get

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

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^\mathsf{T} \\ \vdots \\ \mathbf{h}_n^\mathsf{T} \end{bmatrix}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}. \tag{10.12b}$$

Note that in these equations we add $+\mathbf{b}^{(1)}$ and $+\mathbf{b}^{(2)}$ 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

$$\mathbf{H} = \sigma\left(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}\right), \tag{10.13a}$$

$$\mathbf{Z} = \mathbf{H}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}, \tag{10.13b}$$

which now in a format suitable for efficient vectorized implementation in code.

10.4 Alternative 1

The mathematical model corresponding to the neural network above is

$$h_m = \sigma\left(\beta_{0m}^{(1)} + \sum_{j=1}^{p}\beta_{jm}^{(1)}x_j\right), \qquad m = 1, \dots, M$$

$$z = \beta_0^{(2)} + \sum_{m=1}^{M}\beta_m^{(2)}h_m$$

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

$$\begin{aligned} z &= \beta_0^{(2)} + \sum_{m=1}^{M}\beta_m^{(2)}\left(\beta_{0m}^{(1)} + \sum_{j=1}^{p}\beta_{jm}^{(1)}x_j\right) \\ &= \beta_0^{(2)} + \sum_{m=1}^{M}\beta_m^{(2)}\beta_{0m}^{(1)} + \sum_{j=1}^{p}\sum_{m=1}^{M}\beta_m^{(2)}\beta_{jm}^{(1)}x_j, \end{aligned}$$

which is a linear regression model

$$z = \beta_0 + \sum_{j=1}^{p}\beta_j x_j$$

where

$$\beta_0 = \beta_0^{(2)} + \sum_{m=1}^{M} \beta_m^{(2)} \beta_{0m}^{(1)} \qquad \text{and} \qquad \beta_j = \sum_{m=1}^{M} \beta_m^{(2)} \beta_{jm}^{(1)}. \tag{10.14}$$

Alternative 2

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

$$\mathbf{h} = \mathbf{W}^{(1)\mathsf{T}}\mathbf{x} + \mathbf{b}^{(1)\mathsf{T}}, \tag{10.15a}$$

$$z = \mathbf{W}^{(2)\mathsf{T}}\mathbf{h} + \mathbf{b}^{(2)\mathsf{T}} \tag{10.15b}$$

which gives us

$$z = \mathbf{W}^{(2)\mathsf{T}}\mathbf{W}^{(1)\mathsf{T}}\mathbf{x} + \mathbf{W}^{(2)\mathsf{T}}\mathbf{b}^{(1)\mathsf{T}} + \mathbf{b}^{(2)\mathsf{T}} \tag{10.16}$$

This can be compared with the linear regression model

$$\begin{aligned} z &= \beta_0 + \sum_{j=1}^{p} \beta_j x_j \\ &= \boldsymbol{\beta}_{-0}^{\mathsf{T}}\mathbf{x} + \beta_0 \end{aligned} \tag{10.17}$$

where $\boldsymbol{\beta}_{-0} = [\beta_1, \; , \ldots, \; \beta_p]^{\mathsf{T}}$. By comparing (10.16) and (10.17) we get that

$$\boldsymbol{\beta}_{-0} = \mathbf{W}^{(1)}\mathbf{W}^{(2)}, \qquad \text{and} \qquad \beta_0 = \mathbf{b}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(2)} \tag{10.18}$$

and by using the definitions in Exercise 10.2 this is

$$\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} = \begin{bmatrix} \beta_{11}^{(1)} & \ldots & \beta_{1M}^{(1)} \\ \vdots & & \vdots \\ \beta_{p1}^{(1)} & \ldots & \beta_{pM}^{(1)} \end{bmatrix} \begin{bmatrix} \beta_1^{(2)} \\ \vdots \\ \beta_M^{(2)} \end{bmatrix}, \qquad \text{and} \qquad \beta_0 = \begin{bmatrix} \beta_{01}^{(1)} & \ldots & \beta_{0M}^{(1)} \end{bmatrix} \begin{bmatrix} \beta_1^{(2)} \\ \vdots \\ \beta_M^{(2)} \end{bmatrix} + \begin{bmatrix} \beta_0^{(2)} \end{bmatrix}, \tag{10.19}$$

which is equivalent with the solution in (10.14).