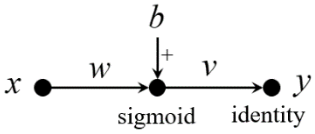
Solution to Homework 2 Problems



1. A multilayer perceptron (MLP) with one hidden layer is shown on the right figure. The activation function of the neuron in the hidden layer is the sigmoid function while the activation function of the output neuron is the identity function.

1.1 Write a mathematical expression of the network output *y* in term of the input *x* and the network parameters *w*, *b*, and *v*. (1 point)

<u>Answer:</u>

$$\begin{aligned} z &= wx + b \\ h &= \sigma(z) \\ y &= vh \end{aligned}$$

1.2 This MLP is used for 1D regression in which the mean square error (MSE) is used as the loss function. Suppose there are *n* data points (x_i, y_i), $i = 1, 2, \ldots, n$. Write a mathematical expression of the loss function $L(w, b, v)$ in this case. (1 point)

<u>Answer:</u>

$$L(w,b,v) = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

where

$$\begin{aligned} z_i(w,b) &= wx_i + b \\ h_i(w,b) &= \sigma\left[z_i(w,b)\right] \\ \hat{y}_i(w,b,v) &= vh_i(w,b) \\ r_i(w,b,v) &= \hat{y}_i(w,b,v) - y_i \end{aligned}$$

and $\sigma(x)$ is the sigmoid function of *x*.

1.3 Derive the formulas for computing the derivatives of the loss function with respect to the network parameters: $\partial L/\partial w, \partial L/\partial b, \partial L/\partial v$. (3 points)

<u>Answer:</u>

$$\frac{\partial L}{\partial w}(w,b,v) = \frac{\partial L}{\partial w}\left(\frac{1}{n}\sum_{i=1}^{n} r_i^2\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial r_i^2}{\partial w} = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial r_i}{\partial w} = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial w}\left[\hat{y}_i(w,b,v) - y_i\right]$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial w}\left[vh_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v\frac{\partial}{\partial w}h_i = \frac{2}{n}\sum_{i=1}^{n} r_i v\frac{\partial}{\partial w}\sigma(z_i) = \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)\frac{\partial z_i}{\partial w}$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)\frac{\partial}{\partial w}(wx_i + b) = \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)x_i$$

$$\frac{\partial L}{\partial b}(w,b,v) = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial b}\left[\hat{y}_i(w,b,v) - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial b}\left[vh_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v\frac{\partial}{\partial b}h_i$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v\frac{\partial}{\partial b}\sigma(z_i) = \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)\frac{\partial z_i}{\partial b} = \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)\frac{\partial}{\partial b}(wx_i + b) = \frac{2}{n}\sum_{i=1}^{n} r_i v\sigma'(z_i)$$

$$\frac{\partial L}{\partial v}(w,b,v) = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial v}\left[\hat{y}_i(w,b,v) - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i\frac{\partial}{\partial v}\left[vh_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i h_i$$

$$\mathbf{g}(w,b,v) = \left[\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial v}\right]^T = \frac{2}{n}\left[\sum_{i=1}^{n} r_i v\sigma'(z_i)x_i, \sum_{i=1}^{n} r_i v\sigma'(z_i), \sum_{i=1}^{n} r_i h_i\right]^T$$

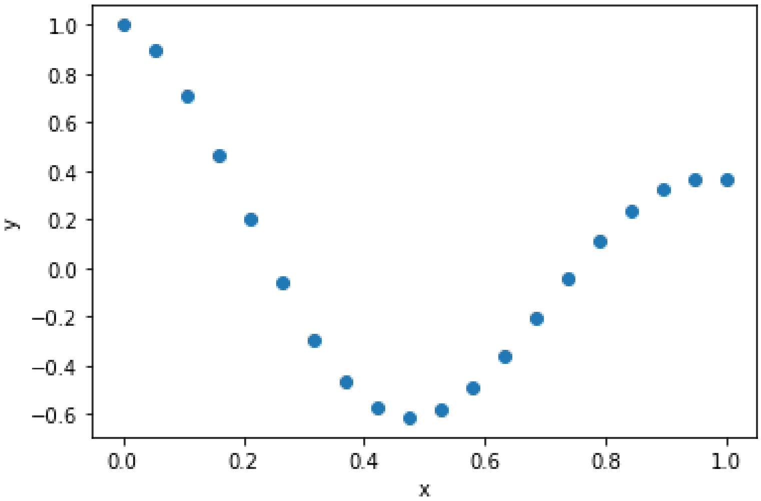
where $\sigma(z_i) = 1\big/\left(1 + e^{-z_i}\right), \sigma'(z_i) = \sigma(z_i)\left[1 - \sigma(z_i)\right]$

1.4 Generate 20 data points using the equation $y = e^{-x}\cos(2\pi x)$, $0 \le x \le 1$. Plot your data points. (1 point)

Answer:

Code to generate the data points are as follows.

```
import numpy as np
from matplotlib.pyplot import *
n = 20
x = np.linspace(0,1,n)
y = np.exp(-x)*np.cos(2*np.pi*x)
plot(x,y,'o') ;xlabel('x');ylabel('y')
```



1.5 Use the dataset generated in Problem 1.4 to train the MLP using the batch gradient descent method. You can choose your own values of learning rate (step length) and the maximum number of iterations. Compare the MLP regression result with the dataset. Also plot the loss function versus iteration number and show your code. (5 points)

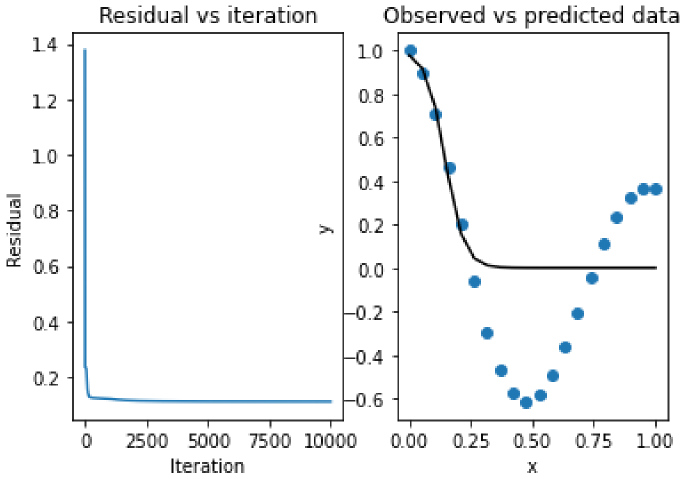
Answer:

The code written for this problem is as follows.

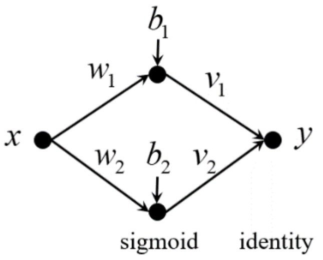
```
def sigmoid(x):
    return 1/(1+np.exp(-x))
# derivative of the sigmoid function
def sigmoid_prime(x):
    s = sigmoid(x)
    return s*(1-s)
# gradient of the loss function with respect to model parameters (w,b,v)
def grad(x,y,n,w,b,v):
    z = w*x+b
    h = sigmoid(z)
    yp = v*h
    r = yp - y
    factor = 2./n
    gw = factor*np.sum(r*v*sigmoid_prime(z)*x)
    gb = factor*np.sum(r*v*sigmoid_prime(z))
    gv = factor*np.sum(r*h)
    return gw,gb,gv
# mean square error loss function
def loss(x,y,n,w,b,v):
    z = w*x+b
    h = sigmoid(z)
    yp = v*h
    r = yp-y
    return np.sum(r*r)/n
```

```
# implementation of the batch gradient descent method
iter_max = 10000
w = np.random.randn()
b = 0.0
v = np.random.randn()
lr = 5.
residual = np.zeros(iter_max+1)
residual[0] = loss(x,y,n,w,b,v)
for iter in range(iter_max):
    gw,gb,gv = grad(x,y,n,w,b,v)
    w = w - lr*gw
    b = b - lr*gb
    v = v - lr*gv
    residual[iter+1] = loss(x,y,n,w,b,v)
subplot(1,2,1);plot(np.arange(iter_max+1),residual)
xlabel('Iteration');ylabel('Residual');title('Residual vs iteration')
z = w*x+b; h = sigmoid(z); yp = v*h
subplot(1,2,2);plot(x,y,'o',x,yp,'k')
xlabel('x');ylabel('y');title('Observed vs predicted data')
print(w,b,v)
```

The batch gradient descent method was carried out for 10,000 iterations with the learning rate of 5. After running the program several times to get good initial values for *w*, *b*, and *v*, the results are shown in the figure below. The final values of *w*, *b*, and *v* are -25.9, 3.8, and 1.0, respectively. The predicted data fit well with the observed data at small values of *x* up to about 0.25. When *x* > 0.25, the predictions are very inaccurate.



2. A MLP has two neurons in the hidden layer. The activation function of both neurons in the hidden layer is the sigmoid function while the activation function of the output neuron is the identity function. This MLP is also used for the 1D regression given in Problem 1 with MSE as the loss function.



2.1 Write a mathematical expression of the network output *y* in term of the input *x* and the network parameters w_1, b_1, w_2, b_2, v_1 and v_2. (1 point)

Answer:

$$\alpha = w_1 x + b_1$$

$$\beta = w_2 x + b_2$$

$$\gamma = \sigma(\alpha)$$

$$\kappa = \sigma(\beta)$$

$$y = v_1\gamma + v_2\kappa$$

2.2 Write the expression of the loss function L in this case. (1 point)

Answer:

$$L(w,b,v)=\frac{1}{n}\sum_{i=1}^{n}r_i^2$$

where

$$\begin{aligned}\alpha_i&=w_1x_i+b_1\\ \beta_i&=w_2x_i+b_2\\ \gamma_i&=\sigma(\alpha_i)\\ \kappa_i&=\sigma(\beta_i)\\ \hat{y}_i&=v_1\gamma_i+v_2\kappa_i\\ r_i&=\hat{y}_i-y_i\end{aligned}$$

2.3 Derive the formulas of the derivatives of the loss function with respect to the network parameters: $\partial L/\partial w_1, \partial L/\partial b_1, \partial L/\partial w_2, \partial L/\partial b_2, \partial L/\partial v_1, \partial L/\partial v_2$. (3 points)

Answer:

$$\frac{\partial L}{\partial w_1}=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial w_1}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial w_1}\left[v_1\gamma_i\left(w_1,b_1\right)+v_2\kappa_i\left(w_2,b_2\right)\right]$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\frac{\partial\gamma_i}{\partial w_1}=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\frac{\partial}{\partial w_1}\sigma\left(\alpha_i\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)\frac{\partial\alpha_i}{\partial w_1}$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)\frac{\partial}{\partial w_1}\left(w_1x_i+b_1\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)x_i$$

$$\frac{\partial L}{\partial b_1}=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial b_1}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial b_1}\left[v_1\gamma_i\left(w_1,b_1\right)+v_2\kappa_i\left(w_2,b_2\right)\right]$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\frac{\partial\gamma_i}{\partial b_1}=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\frac{\partial}{\partial b_1}\sigma\left(\alpha_i\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)\frac{\partial\alpha_i}{\partial b_1}$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)\frac{\partial}{\partial b_1}\left(w_1x_i+b_1\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_1\sigma'\left(\alpha_i\right)$$

$$\frac{\partial L}{\partial w_2}=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial w_2}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial w_2}\left[v_1\gamma_i\left(w_1,b_1\right)+v_2\kappa_i\left(w_2,b_2\right)\right]$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\frac{\partial\kappa_i}{\partial w_2}=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\frac{\partial}{\partial w_2}\sigma\left(\beta_i\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)\frac{\partial\beta_i}{\partial w_2}$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)\frac{\partial}{\partial w_2}\left(w_2x_i+b_2\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)x_i$$

$$\frac{\partial L}{\partial b_2}=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial b_2}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial b_2}\left[v_1\gamma_i\left(w_1,b_1\right)+v_2\kappa_i\left(w_2,b_2\right)\right]$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\frac{\partial\kappa_i}{\partial b_2}=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\frac{\partial}{\partial b_2}\sigma\left(\beta_i\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)\frac{\partial\beta_i}{\partial b_2}$$
$$=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)\frac{\partial}{\partial b_2}\left(w_2x_i+b_2\right)=\frac{2}{n}\sum_{i=1}^{n}r_iv_2\sigma'\left(\beta_i\right)$$

$$\frac{\partial L}{\partial v_1}=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial v_1}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\frac{\partial}{\partial v_1}\left[v_1\gamma_i+v_2\kappa_i\right]=\frac{2}{n}\sum_{i=1}^{n}r_i\gamma_i$$

$$\frac{\partial L}{\partial v_2}=\frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial v_2}\left[\hat{y}_i\left(w_1,b_1,w_2,b_2,v_1,v_2\right)-y_i\right]=\frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial v_2}\left[v_1\gamma_i+v_2\kappa_i\right]=\frac{2}{n}\sum_{i=1}^{n} r_i\kappa_i$$

2.4 Use the dataset generated in Problem 1.4 to train the MLP using the stochastic gradient descent method. You can choose your own values of learning rate and the maximum number of iterations. Compare the MLP regression result with the dataset. Also plot the loss function versus iteration number and show your code. (5 points)

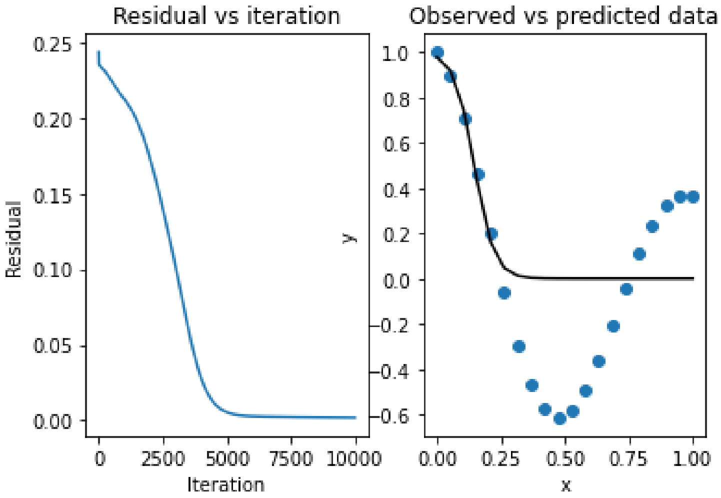
Answer:

The code for this problem is given below.

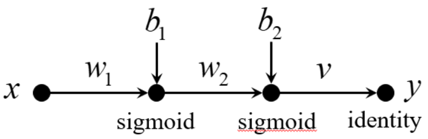
```
def grad_sgd(x,y,n,i,w1,b1,w2,b2,v1,v2):
    alpha = w1*x[i]+b1
    beta = w2*x[i]+b2
    gamma = sigmoid(alpha)
    kappa = sigmoid(beta)
    yp = v1*gamma+v2*kappa
    r = yp - y[i]
    factor = 2./n
    gw1 = factor*(r*v1*sigmoid_prime(alpha)*x[i])
    gb1 = factor*(r*v1*sigmoid_prime(alpha))
    gw2 = factor*(r*v2*sigmoid_prime(beta)*x[i])
    gb2 = factor*(r*v2*sigmoid_prime(beta))
    gv1 = factor*(r*gamma)
    gv2 = factor*(r*kappa)
    return gw1,gb1,gw2,gb2,gv1,gv2
def loss(x,y,n,w1,b1,w2,b2,v1,v2):
    alpha = w1*x+b1
    beta = w2*x+b2
    gamma = sigmoid(alpha)
    kappa = sigmoid(beta)
    yp = v1*gamma+v2*kappa
    r = yp-y
    return np.sum(r*r)/n
# implementation of the stochastic gradient descent method
num_epoch = 10000; lr = 0.1
w1 = np.random.randn();b1 = 0.0
w2 = np.random.randn();b2 = 0.0
v1 = np.random.randn();v2 = np.random.randn()
residual = np.zeros(num_epoch+1)
residual[0] = loss(x,y,n,w1,b1,w2,b2,v1,v2)
for e in range(num_epoch):
    for i in np.random.permutation(n):
        gw1,gb1,gw2,gb2,gv1,gv2 = grad_sgd(x,y,n,i,w1,b1,w2,b2,v1,v2)
        w1 = w1 - lr*gw1
        b1 = b1 - lr*gb1
        w2 = w2 - lr*gw2
        b2 = b2 - lr*gb2
        v1 = v1 - lr*gv1
        v2 = v2 - lr*gv2
    residual[e+1] = loss(x,y,n,w1,b1,w2,b2,v1,v2)
subplot(1,2,1);plot(np.arange(num_epoch+1),residual)
xlabel('Iteration');ylabel('Residual');title('Residual vs iteration')
alpha = w1*x+b1; beta = w2*x+b2;
gamma = sigmoid(alpha); kappa = sigmoid(beta); yp = v*h
```

```
subplot(1,2,2);plot(x,y,'o',x,yp,'k')
xlabel('x');ylabel('y');title('Observed vs predicted data')
print(w1,b1,w2,b2,v1,v2)
```

The final values of the parameters are w_1 = 2.7, b_1 = -0.9, w_2 = 8.3, b_2 = -1.8, v_1 = 5.7, v_2 = -4.4. The results are given in the figure below.



3. A MLP with two hidden layers is used for the 1D regression problem with MSE as the loss function. The activation functions of the neurons are shown in the right figure.



3.1 Write a mathematical expression of the network output y in term of the input x and the network parameters w_1, b_1, w_2, b_2, and v. (1 point)

Answer:

$$\alpha = w_1 x + b_1$$
$$\beta = \sigma(\alpha)$$
$$\gamma = w_2 \beta + b_2$$
$$\kappa = \sigma(\gamma)$$
$$y = v\kappa$$

3.2 Write the expression of the loss function L in this case. (1 point)

Answer:

$$L(w,b,v) = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

where

$$\alpha_i = w_1 x_i + b_1$$
$$\beta_i = \sigma(\alpha_i)$$
$$\gamma_i = w_2 \beta_i + b_2$$
$$\kappa_i = \sigma(\gamma_i)$$
$$\hat{y}_i = v\kappa_i$$
$$r_i = \hat{y}_i - y_i$$

3.3 Derive the formulas of the derivatives of the loss function with respect to the network parameters: $\partial L/\partial w_1, \partial L/\partial b_1, \partial L/\partial w_2, \partial L/\partial b_2, \partial L/\partial v$. (2.5 points)

Answer:

$$\frac{\partial L}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial w_1}\left[\hat{y}_i - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial w_1}\left[v\kappa_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial \kappa_i}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial}{\partial w_1}\sigma\left(\gamma_i\right)$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial \gamma_i}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial}{\partial w_1}\left[w_2\beta_i + b_2\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \frac{\partial \beta_i}{\partial w_1}$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \frac{\partial}{\partial w_1}\sigma\left(\alpha_i\right) = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right)\frac{\partial \alpha_i}{\partial w_1}$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right)\frac{\partial}{\partial w_1}\left[w_1 x_i + b_1\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right) x_i$$

$$\frac{\partial L}{\partial b_1} = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial b_1}\left[\hat{y}_i - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial b_1}\left[v\kappa_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial \kappa_i}{\partial b_1} = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial}{\partial b_1}\sigma\left(\gamma_i\right)$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial \gamma_i}{\partial b_1} = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial}{\partial b_1}\left[w_2\beta_i + b_2\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \frac{\partial \beta_i}{\partial b_1}$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \frac{\partial}{\partial b_1}\sigma\left(\alpha_i\right) = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right)\frac{\partial \alpha_i}{\partial b_1}$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right)\frac{\partial}{\partial b_1}\left[w_1 x_i + b_1\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right) w_2 \sigma'\left(\alpha_i\right)$$

$$\frac{\partial L}{\partial w_2} = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial w_2}\left[\hat{y}_i - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial w_2}\left[v\kappa_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial \kappa_i}{\partial w_2} = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial}{\partial w_2}\sigma\left(\gamma_i\right)$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial \gamma_i}{\partial w_2} = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial}{\partial w_2}\left[w_2\beta_i + b_2\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\beta_i$$

$$\frac{\partial L}{\partial b_2} = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial b_2}\left[\hat{y}_i - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial b_2}\left[v\kappa_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial \kappa_i}{\partial b_2} = \frac{2}{n}\sum_{i=1}^{n} r_i v \frac{\partial}{\partial b_2}\sigma\left(\gamma_i\right)$$

$$= \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial \gamma_i}{\partial b_2} = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)\frac{\partial}{\partial b_2}\left[w_2\beta_i + b_2\right] = \frac{2}{n}\sum_{i=1}^{n} r_i v \sigma'\left(\gamma_i\right)$$

$$\frac{\partial L}{\partial v} = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial v}\left[\hat{y}_i - y_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \frac{\partial}{\partial v}\left[v\kappa_i\right] = \frac{2}{n}\sum_{i=1}^{n} r_i \kappa_i$$

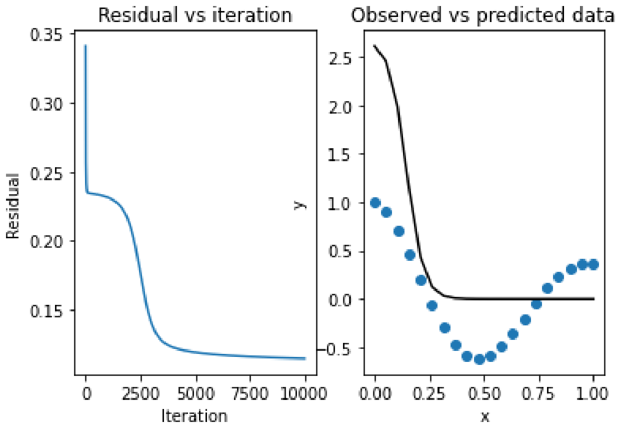
3.4 Use the dataset generated in Problem 1.4 to train the MLP using the minibatch gradient descent method with the batch size of 5. You can choose your own values of learning rate and the maximum number of iterations. Compare the MLP regression result with the dataset. Also plot the loss function versus iteration number and show your code. (5 points)

Answer:

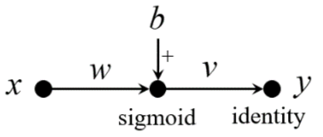
The code for this problem is given below.

```
def grad(x,y,n,w1,b1,w2,b2,v):
    alpha = w1*x+b1
    beta = sigmoid(alpha)
    gamma = w2*beta+b2
    kappa = sigmoid(gamma)
    yp = v*kappa
    r = yp - y
    factor = 2./n
    gw1 = factor*np.sum(r*v*sigmoid_prime(gamma)*w2*
                        sigmoid_prime(alpha)*x)
    gb1 = factor*np.sum(r*v*sigmoid_prime(gamma)*w2*
                        sigmoid_prime(alpha))
    gw2 = factor*np.sum(r*v*sigmoid_prime(gamma)*beta)
    gb2 = factor*np.sum(r*v*sigmoid_prime(gamma))
    gv = factor*np.sum(r*kappa)
    return gw1,gb1,gw2,gb2,gv
def loss(x,y,n,w1,b1,w2,b2,v):
    alpha = w1*x+b1
    beta = sigmoid(alpha)
    gamma = w2*beta+b2
    kappa = sigmoid(gamma)
    yp = v*kappa
    r = yp-y
    return np.sum(r*r)/n
num_epoch = 10000
b = 5
nb = int(n/b)
lr = 0.05
w1 = np.random.randn();b1 = 0.0
w2 = np.random.randn();b2 = 0.0
v = np.random.randn()
residual = np.zeros(num_epoch+1)
residual[0] = loss(x,y,n,w1,b1,w2,b2,v)
for e in range(num_epoch):
    i = np.random.permutation(n).reshape(nb,b)
    for j in range(nb):
        xb, yb = x[i[j]], y[i[j]]   # minibatch data
        gw1,gb1,gw2,gb2,gv = grad(xb,yb,b,w1,b1,w2,b2,v)
        w1 = w1 - lr*gw1
        b1 = b1 - lr*gb1
        w2 = w2 - lr*gw2
        b2 = b2 - lr*gb2
        v = v - lr*gv
    residual[e+1] = loss(x,y,n,w1,b1,w2,b2,v)
subplot(1,2,1);plot(np.arange(num_epoch+1),residual)
xlabel('Iteration');ylabel('Residual');title('Residual vs iteration')
alpha = w1*x+b1; beta = sigmoid(alpha);
gamma = w2*beta+b2; kappa = sigmoid(gamma); yp = v*h
subplot(1,2,2);plot(x,y,'o',x,yp,'k')
xlabel('x');ylabel('y');title('Observed vs predicted data')
print(w1,b1,w2,b2,v)
```

The results are shown in the right figure.

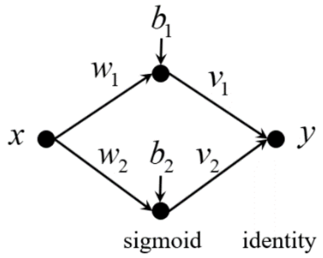


Summary



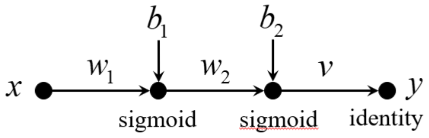
$$z_i = wx_i + b$$
$$h_i = \sigma\left[z_i\right]$$
$$\hat{y}_i = vh_i$$
$$r_i = \hat{y}_i - y_i$$
$$L = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

$$\mathbf{g}(w,b,v) = \begin{bmatrix} \dfrac{\partial L}{\partial w} \\ \dfrac{\partial L}{\partial b} \\ \dfrac{\partial L}{\partial v} \end{bmatrix} = \frac{2}{n}\begin{bmatrix} \sum_{i=1}^{n} r_i v\sigma'(z_i)\, x_i \\ \sum_{i=1}^{n} r_i v\sigma'(z_i) \\ \sum_{i=1}^{n} r_i h_i \end{bmatrix}$$



$$\alpha_i = w_1 x_i + b_1$$
$$\beta_i = w_2 x_i + b_2$$
$$\gamma_i = \sigma(\alpha_i)$$
$$\kappa_i = \sigma(\beta_i)$$
$$\hat{y}_i = v_1\gamma_i + v_2\kappa_i$$
$$r_i = \hat{y}_i - y_i$$
$$L = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

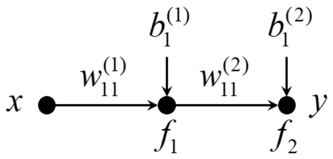
$$\mathbf{g} = \begin{bmatrix} \dfrac{\partial L}{\partial w_1} \\ \dfrac{\partial L}{\partial b_1} \\ \dfrac{\partial L}{\partial w_2} \\ \dfrac{\partial L}{\partial b_2} \\ \dfrac{\partial L}{\partial v_1} \\ \dfrac{\partial L}{\partial v_2} \end{bmatrix} = \frac{2}{n}\begin{bmatrix} \sum_{i=1}^{n} r_i v_1\sigma'(\alpha_i)\, x_i \\ \sum_{i=1}^{n} r_i v_1\sigma'(\alpha_i) \\ \sum_{i=1}^{n} r_i v_2\sigma'(\beta_i)\, x_i \\ \sum_{i=1}^{n} r_i v_2\sigma'(\beta_i) \\ \sum_{i=1}^{n} r_i\gamma_i \\ \sum_{i=1}^{n} r_i\kappa_i \end{bmatrix}$$



$$\alpha_i = w_1 x_i + b_1$$
$$\beta_i = \sigma(\alpha_i)$$
$$\gamma_i = w_2\beta_i + b_2$$
$$\kappa_i = \sigma(\gamma_i)$$
$$\hat{y}_i = v\kappa_i$$
$$r_i = \hat{y}_i - y_i$$
$$L = \frac{1}{n}\sum_{i=1}^{n} r_i^2$$

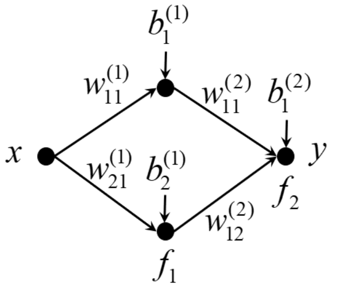
$$\mathbf{g} = \begin{bmatrix} \dfrac{\partial L}{\partial w_1} \\ \dfrac{\partial L}{\partial b_1} \\ \dfrac{\partial L}{\partial w_2} \\ \dfrac{\partial L}{\partial b_2} \\ \dfrac{\partial L}{\partial v} \end{bmatrix} = \frac{2}{n}\begin{bmatrix} \sum_{i=1}^{n} r_i v\sigma'(\gamma_i)\, w_2\sigma'(\alpha_i)\, x_i \\ \sum_{i=1}^{n} r_i v\sigma'(\gamma_i)\, w_2\sigma'(\alpha_i) \\ \sum_{i=1}^{n} r_i v\sigma'(\gamma_i)\,\beta_i \\ \sum_{i=1}^{n} r_i v\sigma'(\gamma_i) \\ \sum_{i=1}^{n} r_i\kappa_i \end{bmatrix}$$

Generalization



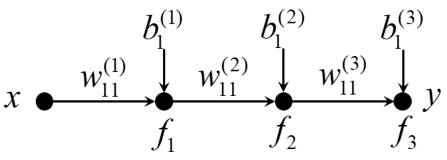
$$
\begin{aligned}
z_{1,i}^{(1)} &= w_{11}^{(1)} x_i + b_1^{(1)} \\
h_{1,i}^{(1)} &= f_1\left[z_{1,i}^{(1)} \right] \\
z_{1,i}^{(2)} &= w_{11}^{(2)} h_{1,i}^{(1)} + b_1^{(2)} \\
\hat{y}_i &= f_2\left[z_{1,i}^{(2)} \right] \\
r_i &= \hat{y}_i - y_i \\
L &= \frac{1}{n} \sum_{i=1}^{n} r_i^2
\end{aligned}
$$

$$
\mathbf{g} = \begin{bmatrix} \dfrac{\partial L}{\partial w_{11}^{(1)}} \\ \dfrac{\partial L}{\partial b_1^{(1)}} \\ \dfrac{\partial L}{\partial w_{11}^{(2)}} \\ \dfrac{\partial L}{\partial b_1^{(2)}} \end{bmatrix} = \frac{2}{n} \begin{bmatrix} \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) x_i \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) h_{1,i}^{(1)} \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) \end{bmatrix}
$$



$$
\begin{aligned}
z_{1,i}^{(1)} &= w_{11}^{(1)} x_i + b_1^{(1)} \\
z_{2,i}^{(1)} &= w_{21}^{(1)} x_i + b_2^{(1)} \\
h_{1,i}^{(1)} &= f_1\left(z_{1,i}^{(1)} \right) \\
h_{2,i}^{(1)} &= f_1\left(z_{2,i}^{(1)} \right) \\
z_{1,i}^{(2)} &= w_{11}^{(2)} h_{1,i}^{(1)} + w_{12}^{(2)} h_{2,i}^{(1)} + b_1^{(2)} \\
\hat{y}_i &= f_2\left(z_{1,i}^{(2)} \right) \\
r_i &= \hat{y}_i - y_i \\
L &= \frac{1}{n} \sum_{i=1}^{n} r_i^2
\end{aligned}
$$

$$
\mathbf{g} = \begin{bmatrix} \dfrac{\partial L}{\partial w_{11}^{(1)}} \\ \dfrac{\partial L}{\partial b_1^{(1)}} \\ \dfrac{\partial L}{\partial w_{21}^{(1)}} \\ \dfrac{\partial L}{\partial b_2^{(1)}} \\ \dfrac{\partial L}{\partial w_{11}^{(2)}} \\ \dfrac{\partial L}{\partial w_{12}^{(2)}} \\ \dfrac{\partial L}{\partial b_1^{(2)}} \end{bmatrix} = \frac{2}{n} \begin{bmatrix} \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) x_i \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{12}^{(2)} f_1'\left(z_{2,i}^{(1)}\right) x_i \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) w_{12}^{(2)} f_1'\left(z_{2,i}^{(1)}\right) \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) h_{1,i}^{(1)} \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) h_{2,i}^{(1)} \\ \sum_{i=1}^{n} r_i f_2'\left(z_{1,i}^{(2)}\right) \end{bmatrix}
$$



$$
\begin{aligned}
z_{1,i}^{(1)} &= w_{11}^{(1)} x_i + b_1^{(1)} \\
h_{1,i}^{(1)} &= f_1\left[z_{1,i}^{(1)} \right] \\
z_{1,i}^{(2)} &= w_{11}^{(2)} h_{1,i}^{(1)} + b_1^{(2)} \\
h_{1,i}^{(2)} &= f_2\left[z_{1,i}^{(2)} \right] \\
z_{1,i}^{(3)} &= w_{11}^{(3)} h_{1,i}^{(2)} + b_1^{(3)} \\
\hat{y}_i &= f_3\left[z_{1,i}^{(3)} \right] \\
r_i &= \hat{y}_i - y_i \\
L &= \frac{1}{n} \sum_{i=1}^{n} r_i^2
\end{aligned}
$$

$$
\mathbf{g} = \begin{bmatrix} \dfrac{\partial L}{\partial w_{11}^{(1)}} \\ \dfrac{\partial L}{\partial b_1^{(1)}} \\ \dfrac{\partial L}{\partial w_{11}^{(2)}} \\ \dfrac{\partial L}{\partial b_1^{(2)}} \\ \dfrac{\partial L}{\partial w_{11}^{(3)}} \\ \dfrac{\partial L}{\partial b_1^{(3)}} \end{bmatrix} = \frac{2}{n} \begin{bmatrix} \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) w_{11}^{(3)} f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) x_i \\ \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) w_{11}^{(3)} f_2'\left(z_{1,i}^{(2)}\right) w_{11}^{(2)} f_1'\left(z_{1,i}^{(1)}\right) \\ \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) w_{11}^{(3)} f_2'\left(z_{1,i}^{(2)}\right) h_{1,i}^{(1)} \\ \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) w_{11}^{(3)} f_2'\left(z_{1,i}^{(2)}\right) \\ \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) h_{1,i}^{(2)} \\ \sum_{i=1}^{n} r_i f_3'\left(z_{1,i}^{(3)}\right) \end{bmatrix}
$$