HW 8

Runze Wang

0.1 1. Question 10.10.1 on Page 458 in ISLRv2.

0.2 1. Consider a neural network with two hidden layers: p = 4 input units, 2 units in the first hidden layer, 3 units in the second hidden layer, and a single output.

0.3 (a) Draw a picture of the network, similar to Figures 10.1 or 10.4.

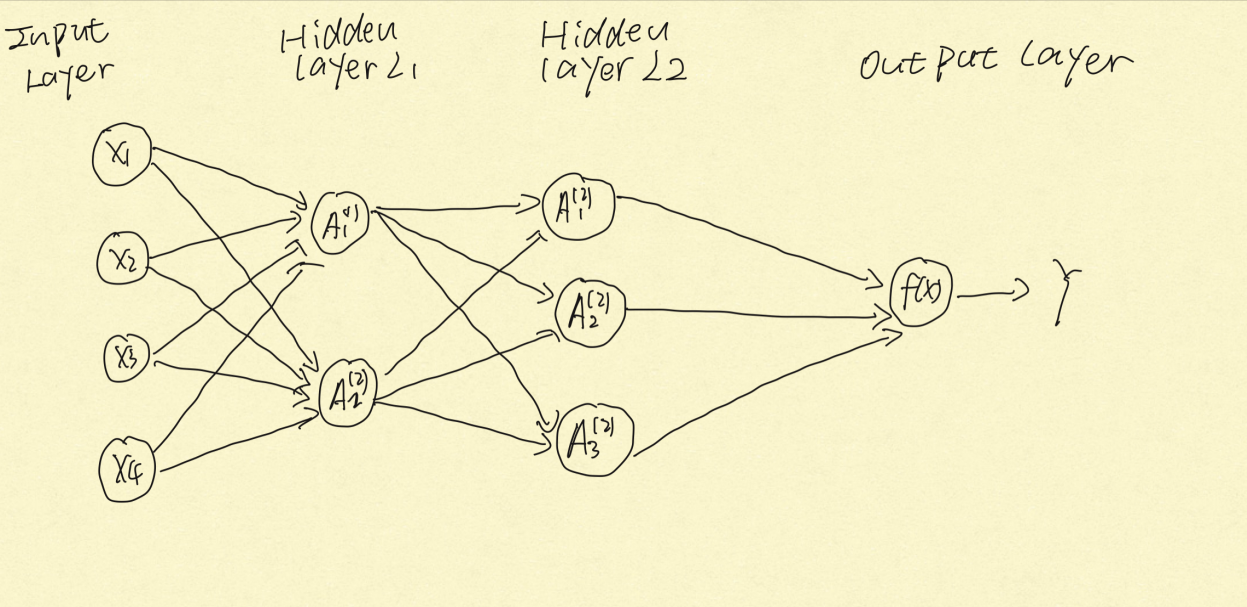

Figure 1: a

0.4 (b) Write out an expression for f(X), assuming ReLU activation functions. Be as explicit as you can!

$$f(X) = \beta_0 + \beta_1 A_1^{(2)} + \beta_2 A_2^{(2)} + \beta_3 A_3^{(2)}$$

$$A_l^{(2)} = g\left(w_{l0}^{(2)} + \sum_{k=1}^{K_1} w_{lk}^{(2)} A_k^{(1)}\right)$$

$$A_k^{(1)} = g\left(w_{k0}^{(1)} + \sum_{j=1}^{p} w_{kj}^{(1)} X_j\right)$$

for $K_2 = 3$, $K_1 = 2$, $p = 4$, $l = 1, 2, 3$, $k = 1, 2, 3, 4$

$$g(z) = (z)_+ = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}$$

0.5 (c) Now plug in some values for the coefficients and write out the value of f(X).

```
set.seed(1000)
K1=2
K2=3
p=4
X = c(1,runif(p))
W1 = matrix(runif((p+1)*K1),K1,p+1)
W2 = matrix(runif((K1+1)*K2),K2,K1+1)
beta = runif(K2+1)
Re <- function(u) max(0,u)
A1 = sapply(W1%*%X,Re)
A1 = sapply(W2%*% c(1,A1),Re)
f= sum(beta* c(1,A1))
f
```

```
## [1] 2.423612
```

0.6 (d) How many parameters are there?

$p * K_1 + K_1 + K_1 * K_2 + K_2 + K_2 * 1 + 1 = 4 * 2 + 2 + 2 * 3 + 3 + 3 * 1 + 1 = 23$ There are 23 parameters.

0.7 Consider the softmax function in (10.13) (see also (4.13) on page 141) for modeling multinomial probabilities.

0.8 (a) In (10.13), show that if we add a constant c to each of the zl, then the probability is unchanged.

For (10.13),

$$f_m(X) = \Pr(Y = m|X) = \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}$$

Now add the constant c

$$\begin{aligned}
\Pr(Y = m|X) &= \frac{e^{Z_m + c}}{\sum_{l=0}^{9} e^{Z_l + c}} \\
&= \frac{e^{Z_m} e^{c}}{\sum_{l=0}^{9} e^{Z_l} e^{c}} \quad \text{(Exponent Product Rule)} \\
&= \frac{e^{Z_m} e^{c}}{e^{c} \sum_{l=0}^{9} e^{Z_l}} \\
&= \frac{e^{Z_m}}{\sum_{l=0}^{9} e^{Z_l}}
\end{aligned}$$

0.9 (b) In (4.13), show that if we add constants cj , j = 0, 1,. . . ,p, to each of the corresponding coefficients for each of the classes, then the predictions at any new point x are unchanged.

$$\begin{aligned}
\Pr(Y = m|X = x) &= \frac{e^{\beta_{m0}x_0 + c_0 + \beta_{m1}x_1 + c_1 + \ldots + \beta_{mp}x_p + c_p}}{\sum_{l=0}^{m} e^{\beta_{l0}x_0 + c_0 + \beta_{l1}x_1 + c_1 + \ldots + \beta_{lp}x_p + c_p}} \\
&= \frac{e^{c_0 + c_1 + \ldots + c_p}\left(e^{\beta_{m0}x_0 + \beta_{m1}x_1 + \ldots + \beta_{mp}x_p}\right)}{\sum_{l=0}^{m} e^{c_0 + c_1 + \ldots + c_p}\left(e^{\beta_{l0}x_0 + \beta_{l1}x_1 + \ldots + \beta_{lp}x_p}\right)} \quad \text{(Exponent Product Rule)} \\
&= \frac{e^{\beta_{m0}x_0 + \beta_{m1}x_1 + \ldots + \beta_{mp}x_p}}{\sum_{l=0}^{m} e^{\beta_{l0}x_0 + \beta_{l1}x_1 + \ldots + \beta_{lp}x_p}}
\end{aligned}$$

0.10 3. Show that the negative multinomial log-likelihood (10.14) is equivalent to the negative log of the likelihood expression (4.5) when there are M = 2 classes.

Equation 10.14 is

$$-\sum_{i=1}^{n}\sum_{m=0}^{9} y_{im}\log(f_m(x_i))$$

When M=2, it means there are two classes. y_{i1} and y_{i0} denonted they are belong to class 1 and 0 (or 0 and 1) when ith observation is 1 (or 0). The probability of class 1 is $p(x_i)$, so the probability of class 0 is $1-p(x_i)$.Then we can rewrite 10.14 into

$$-\sum_{i=1}^{n}\left[y_{i1}\log(p(x_i)) + y_{i0}\log(1-p(x_i))\right]$$

Equation 4.5 is:

$$\ell(\beta_0,\beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_i'=0} (1-p(x_i'))$$

The log- likelihood is

$$\begin{aligned}\log(\ell(\beta_0,\beta_1)) &= \log\left(\prod_{i:y_i=1} p(x_i) \prod_{i':y_i'=0} (1-p(x_i'))\right)\\ &= \sum_{i:y_1=1}\log(p(x_i)) + \sum_{i':y_i'=0}\log(1-p(x_i')) \quad \text{(Product Rule Of Logarithms)}\end{aligned}$$

Negative log-likelihood :

$$-\log(\ell(\beta_0,\beta_1)) = -\left[\sum_{i:y_i=1}\log(p(x_i)) + \sum_{i':y_{i'}=0}\log(1-p(x_{i'}))\right]$$

The y_i is indicators variavle, so we rewrite negative log-likelihood into:

$$-\log(\ell) = -\sum_i y_{i1}\log(p(x_i)) + \sum_i y_{i0}\log(1-p(x_i'))$$

so the negative multinomial log-likelihood (10.14) is equivalent to the negative log of the likelihood expression (4.5) when there are M = 2 classes.

0.11 4. Question 10.10.5 on Page 459 in ISLRv2.

0.12 5. In Table 10.2 on page 433, we see that the ordering of the three methods with respect to mean absolute error is different from the ordering with respect to test set R2. How can this be?

MAE is calculated the the mean of the difference between the predicted value and the observed value. R^2 measures the proportion of the variance for the dependent variable that's explained by the independent variables. They have distinct properties.

It may caused by the complexity of the model, Neural Network might be able to get a much tighter fit to the data, so it may have high R^2. However they may not predict the data accuracy, it will lead to a large MAE.