Introduction to Machine Learning
Fall 2019
University of Science and Technology of China

Lecturer: Jie Wang | Homework 5
Posted: Nov. 20, 2019 | Due: Nov. 27, 2019
Name: San Zhang | ID: PBXXXXXXXX

Notice, to get the full credits, please show your solutions step by step.

Exercise 1: Decision Tree 10pts

Please build a decision tree based on the information gain to classify the following dataset (you need to show the calculation steps in detail).

Sample	A_1	A_2	A_3	Response
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

Table 1: Dataset

The dataset consists of five samples x_1, x_2, x_3, x_4, x_5. For each sample, we can observe the features A_1, A_2, A_3 and the corresponding response.

Solution: Let $D = \{x_1, x_2, x_3, x_4, x_5\}$. The information gain is

$$\begin{aligned}
\mathrm{Gain}(D, A_1) &= \mathrm{Entropy}(D) - \frac{4}{5}\log 2, \\
\mathrm{Gain}(D, A_2) &= \mathrm{Entropy}(D) - \frac{1}{5}\log(\frac{27}{4}), \\
\mathrm{Gain}(D, A_3) &= \mathrm{Entropy}(D) - \frac{1}{5}\log 27.
\end{aligned}$$

As $\mathrm{Gain}(D, A_2) > \mathrm{Gain}(D, A_1) > \mathrm{Gain}(D, A_3)$, the first node is A_2.

For all $A_2 = 0$ the response is 0, then we only need to consider $A_2 = 1$:

$$\begin{aligned}
\mathrm{Gain}(D', A_1) &= \mathrm{Entropy}(D'), \\
\mathrm{Gain}(D', A_3) &= \mathrm{Entropy}(D') - \frac{2}{3}\log 2,
\end{aligned}$$

where $D' = D \backslash \{x_1, x_2\}$.

As $\mathrm{Gain}(D', A_1) > \mathrm{Gain}(D', A_3)$, the second node is A_1 when $A_2 = 1$. The decision tree is shown in Figure 1. ■

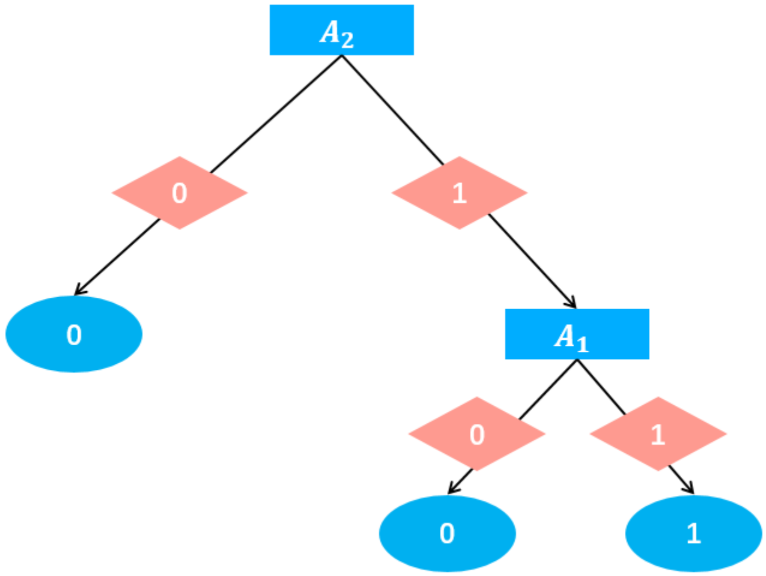

Figure 1: Decision tree

Exercise 2: Softmax and Cross Entropy 30pts

The softmax function $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined by:

$$f_i(x) = \frac{\exp(x_i)}{\sum_{k=1}^{n} \exp(x_k)}, i = 1, \dots, n,$$

where x_i is the i^{th} component of $x \in \mathbb{R}^n$. The function $f(x) = (f_1(x), f_2(x), \dots, f_n(x))^\top$ converts each input x into a probability (stochastic) vector in which all entries are nonnegative and add up to one.

1. Please find the gradient and Jacobian of $f(x)$, i.e., $\nabla f(x)$ and $Df(x)$.
2. Show that $f(x) = f(x - c)$, where $c = \max\{x_1, x_2, ..., x_n\}$. When could we need this transformation?
3. Please find the gradient of cross entropy function:

   $$g(x) = -\sum_{i=1}^{n} H_i \log(f_i(x)),$$

   where $H \in \mathbb{R}^n$ is a one-hot vector.

Solution:

1. We first derive the gradient of $f_i(x)$ with respect to x_i:

   $$\begin{aligned}\frac{\partial f_i(x)}{\partial x_i} &= \frac{\exp(x_i)(\sum_{k=1}^{n} \exp(x_k) - exp(x_i))}{(\sum_{k=1}^{n} \exp(x_k))^2} \\ &= \frac{\exp(x_i)}{\sum_{k=1}^{n} \exp(x_k)}(1 - \frac{\exp(x_i)}{\sum_{k=1}^{n} \exp(x_k)}) \\ &= f_i(x)(1 - f_i(x)).\end{aligned}$$

   Then, if $j \neq i$, we can get:

   $$\begin{aligned}\frac{\partial f_i(x)}{\partial x_j} &= \frac{-\exp(x_i)\exp(x_j)}{(\sum_{k=1}^{n} \exp(x_k))^2} \\ &= -f_i(x) f_j(x).\end{aligned}$$

   Then, we can write the gradient and Jacobian of $f(x)$ directly:

   $$\nabla f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_1} & \cdots & \frac{\partial f_n(x)}{\partial x_1} \\ \frac{\partial f_1(x)}{\partial x_2} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_n(x)}{\partial x_2} \\ \vdots & \vdots & \vdots & \ddots \\ \frac{\partial f_1(x)}{\partial x_n} & \frac{\partial f_2(x)}{\partial x_n} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$$

and

$$Df(x) = (\nabla f(x))^\top = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \cdots & \frac{\partial f_n(x)}{\partial x_n}. \end{bmatrix}$$

2. As

$$\begin{aligned} f_i(x-c) &= \frac{\exp(x_i - c)}{\sum_{k=1}^n \exp(x_k - c)} \\ &= \frac{\exp(-c)}{\exp(-c)} \frac{\exp(x_i)}{\sum_{k=1}^n \exp(x_k)} \\ &= \frac{\exp(x_i)}{\sum_{k=1}^n \exp(x_k)} \\ &= f_i(x), \end{aligned}$$

we can conclude that $f(x) = f(x-c)$.

If c is very large, then the value of $\exp(c)$ will overflow when we compute softmax function in the computer. Therefore, we need this transformation in this situation.

3. Suppose the i^{th} element of H is 1, and 0, otherwise. Then, we can rewrite $g(x)$ as:

$$\begin{aligned} g(x) &= -\sum_{i=1}^n H_i \log(f_i(x)) \\ &= -\log(f_i(x)). \end{aligned}$$

Thus, we have

$$\frac{\partial g(x)}{\partial f_i(x)} = -\frac{1}{f_i(x)}$$

Combining the results of problem 1 and the chain rule, we can conclude that:

$$\frac{\partial g(x)}{\partial x_j} = \frac{\partial g(x)}{\partial f_i(x)} \frac{\partial f_i(x)}{\partial x_j} = \begin{cases} f_i(x) - 1, & i = j \\ f_j(x), & i \neq j. \end{cases}$$

■

Exercise 3: Convolutional Neural Network 40pts

1. The average pooling in convolutional neural network can be formulated as

$$f_1(x) = \frac{\sum_{i=1}^{n} x_i}{n},$$

where x_i is the i^{th} component of $x \in \mathbb{R}^n$. Please derive the gradient of $f_1(x)$.

2. The max pooling in convolutional neural network can be formulated as

$$f_2(x) = \max\{x_1, \ldots, x_n\},$$

where x_i is the i^{th} component of $x \in \mathbb{R}^n$.

   (a) Find the set containing all differentiable points of of f_2.

   (b) We call $d(x)$ is a subgradient at x f_2 if

$$f_2(y) \geq f_2(x) + \langle d(x), y - x \rangle, \forall x, y.$$

   Find a subgradient $d(x)$ of f_2 at x.

3. Suppose that we have a convolutional neural network as shown in Table 2.

   (a) The convolutinal layer parameters are denoted as "conv⟨filter size⟩-⟨number of filters⟩".

   (b) The fully connected layer parameters are denoted as "FC⟨number of neurons⟩".

   (c) The window size of pooling layers is 3.

   (d) The stride of convolutinal layers is 1.

   (e) The stride of pooling layers is 3.

   (f) There is no padding in both convolutional and pooling layers.

   (g) For convenience, we assume that there is no activation function and bias.

   Suppose that the input is a **386** $\times$ **386 RGB** image. Please derive the size of all feature maps and the number of parameters.

conv3-64	max pool	conv3-256	conv1-512	max pool	FC-2048	FC-1000

Table 2: The architecture of convolutional neural network

Solution:

1. The gradient of $f_1(x)$ is

$$\nabla f_1(x) = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^\top.$$

2. (a) The set containing all differentiable points of f_2 is

$$A = \{x : x_i = \max_k\{x_k\} \text{ and } x_j < x_i, \forall\, j \neq i\}.$$

(b) Suppose that $f_2(x) = x_i$. Consider $d(x) = e_i$, then

$$\begin{aligned} f_2(x) + \langle e_i, y - x \rangle &= x_i + y_i - x_i \\ &= y_i \leq f_2(y) \end{aligned}$$

for all $y \in \mathbb{R}^n$. Thus, e_i is a subgradient of f_2 at x.

3. **Feature map size:**

When there is no padding, and noth the window size and the stride are 3, we know that the size of feature maps after convolution is

$$N \times (H - n + 1) \times (W - n + 1)\,,$$

where $H \times W$ is the size of the input image, N is the number of filters and n is the size of filters.

As the stride of pooling layers is 1 and the window size of pooling layers is 3, if the input is a $n \times H \times W$ image, then the size of output is

$$n \times \frac{H}{3} \times \frac{W}{3},$$

Thus, we know that the size of feature maps in the network is shown in the Table 3.

Parameter number:

We know that only convolutional layers and fully connected layers have parameters.

As there is no activation function and bias, the number of parameters of three convolutional layers is

$$\begin{aligned} 1728 &= 64 \times 3 \times 3 \times 3, \\ 147456 &= 256 \times 3 \times 3 \times 64, \\ 131072 &= 512 \times 1 \times 1 \times 256. \end{aligned}$$

Thus, the total number of parameters of the convolutional layers is 280256.

As the output of the final max pooling layer has the size $512 \times 42 \times 42$, we know that the first fully connected layer has $512 \times 42 \times 42 \times 2048 = 1849688064$ parameters.

The last fully connected layer has $2048 \times 1000 = 2048000$ parameters.

In conclusion, the number of parameters of the network is

$$1852016320 = 280256 + 1849688064 + 2048000.$$

■

Layers	conv3-64	max pool	conv3-256	conv1-512	max pool
Size	$64 \times 384 \times 384$	$64 \times 128 \times 128$	$256 \times 126 \times 126$	$512 \times 126 \times 126$	$512 \times 42 \times 42$

Table 3: The size of feature maps in each layer

Exercise 4: Matrix Calculus 20pts

Let $L = f(h(Ax+b))$, where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $f : \mathbb{R}^m \to \mathbb{R}$. Define $z = Ax + b \in \mathbb{R}^m$ and $w = h(z) = (\sigma(z_1), \ldots, \sigma(z_n))^\top$, where z_i is the i^{th} component of z and

$$\sigma(z_i) = \frac{1}{1+\exp(-z_i)}.$$

Assume $\nabla_w f$ is known.

1. Please derive $\nabla_x L$.
2. Please derive

$$\nabla_A L = \begin{bmatrix} \frac{\partial L}{\partial A_{11}} & \cdots & \frac{\partial L}{\partial A_{1j}} & \cdots & \frac{\partial L}{\partial A_{1n}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial L}{\partial A_{i1}} & \cdots & \frac{\partial L}{\partial A_{ij}} & \cdots & \frac{\partial L}{\partial A_{in}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial L}{\partial A_{m1}} & \cdots & \frac{\partial L}{\partial A_{mj}} & \cdots & \frac{\partial L}{\partial A_{mn}} \end{bmatrix},$$

where $A_{i,j}$ is the entry in the i^{th} row, j^{th} column of the matrix A.

Solution:

1. We know that

$$\nabla_x L = \nabla_x z \nabla_z w \nabla_w f.$$

Next we compute $\nabla_z w$ and $\nabla_x z$.

$$\nabla_z w = \begin{bmatrix} \sigma(z_1)(1-\sigma(z_1)) & \cdots & 0 \\ 0 & \sigma(z_2)(1-\sigma(z_2)) & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \sigma(z_m)(1-\sigma(z_m)) \end{bmatrix}$$

$$\nabla_x z = A^\top$$

Let $\mathrm{diag}(\sigma(1-\sigma)) = \nabla_z(w)$. Thus,

$$\nabla_x L = A^\top \mathrm{diag}(\sigma(1-\sigma))\nabla_w f.$$

2. From the chain rule for derivation, we have

$$\begin{aligned} \frac{\partial L}{\partial A_{i,j}} &= \frac{\partial f}{\partial w_i}\frac{dw_i}{dz_i}\frac{\partial z_i}{\partial A_{i,j}} \\ &= \frac{\partial f}{\partial w_i}\sigma(z_i)(1-\sigma(z_i))x_j \\ &= [\mathrm{diag}(\sigma(1-\sigma))(\nabla_w f)]_i\, x_j, \end{aligned}$$

Thus

$$\nabla_A L = \mathrm{diag}(\sigma(1-\sigma))(\nabla_w f)x^\top$$

■