CS329 Machine Learning

Homework #5

Fan Site
fanst2021@mail.sustech.edu.cn

Question 1

Consider a regression problem involving multiple target variables in which it is assumed that the distribution of the targets, conditioned on the input vector x, is a Gaussian of the form

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{x},\mathbf{w}), \mathbf{\Sigma})$$

where $\mathbf{y}(\mathbf{x},\mathbf{w})$ is the output of a neural network with input vector $\mathbf{x}$ and wight vector $\mathbf{w}$, and $\mathbf{\Sigma}$ is the covariance of the assumed Gaussian noise on the targets.

(a) Given a set of independent observations of $\mathbf{x}$ and $\mathbf{t}$, write down the error function that must be minimized in order to find the maximum likelihood solution for $\mathbf{w}$, if we assume that $\mathbf{\Sigma}$ is fixed and known.

(b) Now assume that $\mathbf{\Sigma}$ is also to be determined from the data, and write down an expression for the maximum likelihood solution for $\mathbf{\Sigma}$. (Note: The optimizations of $\mathbf{w}$ and $\mathbf{\Sigma}$ are now coupled.)

Solution (a)

The likelihood function:

$$p(\mathbf{T}|\mathbf{X},\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{x},\mathbf{w}), \mathbf{\Sigma})$$

The error function to be minimized:

$$E(\mathbf{w}, \mathbf{\Sigma}) = \frac{1}{2}\sum_{n=1}^{N}\left\{[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]^{\mathrm{T}}\mathbf{\Sigma}^{-1}[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]\right\} + \frac{N}{2}\ln|\mathbf{\Sigma}| + \frac{N}{2}\ln(2\pi)$$

If $\mathbf{\Sigma}$ is known and fixed,

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]^{\mathrm{T}}\mathbf{\Sigma}^{-1}[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]\right\} + \mathrm{const}$$

By minimizing this error function, we obtain $\mathbf{w}_{\mathrm{ML}}$.

Solution (b)

The error function:

$$E(\mathbf{w}, \mathbf{\Sigma}) = \frac{1}{2}\sum_{n=1}^{N}\left\{[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]^{\mathrm{T}}\mathbf{\Sigma}^{-1}[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]\right\} + \frac{N}{2}\ln|\mathbf{\Sigma}| + \frac{N}{2}\ln(2\pi)$$

Let the gradient of the error function with respect to $\mathbf{\Sigma}$ be 0:

$$\frac{\partial}{\partial \mathbf{\Sigma}}E(\mathbf{w}, \mathbf{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{\Sigma}^{-1}[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n][\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]^{\mathrm{T}}\mathbf{\Sigma}^{-1}\right\} + \frac{N}{2}\mathbf{\Sigma}^{-1} = 0$$

Solve the equation, we obtain $\mathbf{\Sigma}_{\mathrm{ML}}$:

$$\mathbf{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}[\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n][\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n]^{\mathrm{T}}$$

Question 2

The error function for binary classification problems was derived for a network having a logistic-sigmoid output activation function, so that $0 \le y(\mathbf{x},\mathbf{w}) \le 1$, and data having target values $t \in \{0, 1\}$. Derive the corresponding error function if we consider a network having an output $-1 \le y(\mathbf{x},\mathbf{w}) \le 1$ and target values $t = 1$ for class $\mathcal{C}_1$ and $t = -1$ for class $\mathcal{C}_2$. What would be the appropriate choice of output unit activation function?

Hint. The error function is given by:

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}.$$

Solution

Scaling and shifting the binary outputs from [0,1] to [−1,1]:

$$y = 2\sigma(a) - 1 \in [-1, 1]$$

The conditional distribution:

$$p(t|\mathbf{x},\mathbf{w}) = \left[\frac{1+y(\mathbf{x},\mathbf{w})}{2}\right]^{\frac{1+t}{2}}\left[\frac{1-y(\mathbf{x},\mathbf{w})}{2}\right]^{\frac{1-t}{2}}$$

where the conditional probability is $p(\mathcal{C}_1|x) = \frac{1+y(\mathbf{x},\mathbf{w})}{2}$.

Hence we obtain the error function:

$$\begin{aligned} E(\mathbf{w}) &= -\sum_{n=1}^{N}\left\{\frac{1+t_n}{2}\ln\frac{1+y_n}{2} + \frac{1-t_n}{2}\ln\frac{1-y_n}{2}\right\} \\ &= -\frac{1}{2}\sum_{n=1}^{N}\{(1+t_n)\ln(1+y_n) + (1-t_n)\ln(1-y_n)\} + N\ln 2 \end{aligned}$$

So we obtain the activation function:

$$y(a) = 2\sigma(a) - 1 = \frac{1-e^{-a}}{1+e^{-a}} = \frac{e^{\frac{a}{2}} - e^{-\frac{a}{2}}}{e^{\frac{a}{2}} + e^{-\frac{a}{2}}} = \tanh\left(\frac{a}{2}\right)$$

Question 3

Verify the following results for the conditional mean and variance of the mixture density network model.

(a)

$$\mathbb{E}[\mathbf{t}|\mathbf{x}] = \int \mathbf{t}p(\mathbf{t}|\mathbf{x})\ \mathrm{d}\ \mathbf{t} = \sum_{k=1}^{K}\pi_k(\mathbf{x})\mu_k(\mathbf{x}).$$

(b)

$$s^2(\mathbf{x}) = L\sum_{k=1}^{K}\pi_k(\mathbf{x})\sigma_k^2(\mathbf{x}) + \left\|\mu_k(\mathbf{x}) - \sum_{l=1}^{K}\pi_l(\mathbf{x})\mu_l(\mathbf{x})\right\|^2.$$

Solution (a)

$$\begin{aligned} \mathbb{E}[\mathbf{t}|\mathbf{x}] &= \int \mathbf{t}p(\mathbf{t}|\mathbf{x})\ \mathrm{d}\ \mathbf{t} \\ &= \int \mathbf{t}\sum_{k=1}^{K}\pi_k(\mathbf{x})\mathcal{N}(\mathbf{t}|\mu_k, \sigma_k^2)\mathrm{d}\mathbf{t} \\ &= \sum_{k=1}^{K}\pi_k(\mathbf{x})\int \mathbf{t}\mathcal{N}(\mathbf{t}|\mu_k, \sigma_k^2)\mathrm{d}\mathbf{t} \\ &= \sum_{k=1}^{K}\pi_k(\mathbf{x})\mu_k(\mathbf{x}). \end{aligned}$$

Solution (b)

Noticing that $\mathbb{E}[\mathbf{t}^{\mathrm{T}}A\ \mathbf{t}] = \mathrm{Trace}[A\sigma^2] + \boldsymbol{\mu}^{\mathrm{T}}A\boldsymbol{\mu}$, let $A = \mathbf{I}$, we obtain

$$\mathbb{E}[\mathbf{t}^2] = \int \|\mathbf{t}\|^2 \mathcal{N}(\mathbf{t}|\boldsymbol{\mu}, \sigma^2\mathbf{I})\mathrm{d}\mathbf{t} = L\sigma^2 + \|\boldsymbol{\mu}\|^2$$

where L is the dimension of $\mathbf{t}$.

Therefore

$$\begin{aligned}
s^2(\mathbf{x}) &= \mathbb{E}\left[\|\mathbf{t} - \mathbb{E}[\mathbf{t}|\mathbf{x}]\|^2 |\mathbf{x}\right] \\
&= \mathbb{E}[\mathbf{t}^2|\mathbf{x}] - \mathbb{E}[\mathbf{t}|\mathbf{x}]^2 \\
&= \int \|\mathbf{t}\|^2 \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathcal{N}(\boldsymbol{\mu}_k, \sigma_k^2)\mathrm{d}\mathbf{t} - \left\|\sum_{l=1}^{K} \pi_l(\mathbf{x})\mu_l(\mathbf{x})\right\|^2 \\
&= \sum_{k=1}^{K} \pi_k(\mathbf{x})\left(L\sigma_k^2(\mathbf{x}) + \|\boldsymbol{\mu}_{\boldsymbol{k}}(\mathbf{x})\|^2\right) - \left\|\sum_{l=1}^{K} \pi_l(\mathbf{x})\mu_l(\mathbf{x})\right\|^2 \\
&= L\sum_{k=1}^{K} \pi_k(\mathbf{x})\sigma_k^2(\mathbf{x}) + \sum_{k=1}^{K} \pi_k(\mathbf{x})\|\boldsymbol{\mu}_{\boldsymbol{k}}(\mathbf{x})\|^2 - \left\|\sum_{l=1}^{K} \pi_l(\mathbf{x})\mu_l(\mathbf{x})\right\|^2 \\
&= L\sum_{k=1}^{K} \pi_k(\mathbf{x})\sigma_k^2(\mathbf{x}) + \left\|\mu_k(\mathbf{x}) - \sum_{l=1}^{K} \pi_l(\mathbf{x})\mu_l(\mathbf{x})\right\|^2
\end{aligned}$$

Question 4

Can you represent the following boolean function with a single logistic threshold unit (i.e., a single unit from a neural network)? If yes, show the weights. If not, explain why not in 1-2 sentences.

A	B	$f(A, B)$
1	1	0
0	0	0
1	0	1
0	1	0

Solution

Yes, this function is linearly separable.

$$y = [A \quad B]\begin{bmatrix} 2 \\ -1 \end{bmatrix} - 1 = 2A - B - 1$$

We use a threshold as the activation function:

If $y > 0$, then $f(A, B) = 1$, otherwise $f(A, B) = 0$.

Question 5

Below is a diagram of a small convolutional neural network that converts a 13x13 image into 4 output values. The network has the following layers/operations from input to output: convolution with 3 filters, max pooling, ReLU, and finally a fully-connected layer. For this network we will not be using any bias/offset parameters (b). Please answer the following questions about this network.

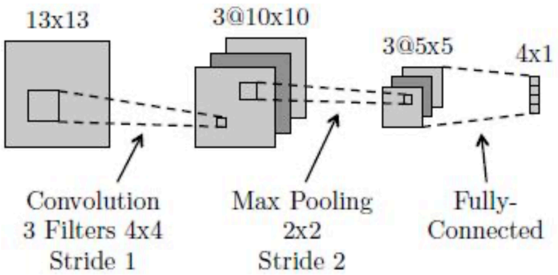

Figure 1: The Convolutional Neural Network for Question 5

(a) How many weights in the convolutional layer do we need to learn?

(b) How many ReLU operations are performed on the forward pass?

(c) How many weights do we need to learn for the entire network?

(d) True or false: A fully-connected neural network with the same size layers as the above network $(13 \times 13 \rightarrow 3 \times 10 \times 10 \rightarrow \times 5 \times 5 \rightarrow 4 \times 1)$ can represent any classifier?

(e) What is the disadvantage of a fully-connected neural network compared to a convolutional neural network with the same size layers?

Solution

(a) $3 * 4 * 4 = 48$

(b) $3 * (10/2) * (10/2) = 75$

(c) $3 * 4 * 4 + 3 * 5 * 5 * 4 = 348$

(d) False. The fully-connected neural network can capture a wide range of functions, but it may not be able to represent highly complex or nonlinear decision boundaries. However, the fully-connected neural network can represent any classifier that the convolutional neural network can represent.

(e)

- Fully-connected neural networks lack translation invariance, meaning that they may not recognize patterns in different spatial locations.
- In fully-connected neural networks, each neuron in a layer is connected to every neuron in the previous and subsequent layers. This leads to a large number of parameters, resulting in increased computational requirements and the risk of overfitting, especially when dealing with high-dimensional data like images.

Question 6

The neural networks shown in class used logistic units: that is, for a given unit U, if A is the vector of activations of units that send their output to U, and W is the weight vector corresponding to these outputs, then the activation of U will be $(1 + \exp(W^{\mathrm{T}} A))^{-1}$. However, activation functions could be anything. In this exercise we will explore some others. Consider the following neural network, consisting of two input units, a single hidden layer containing two units, and one output unit:

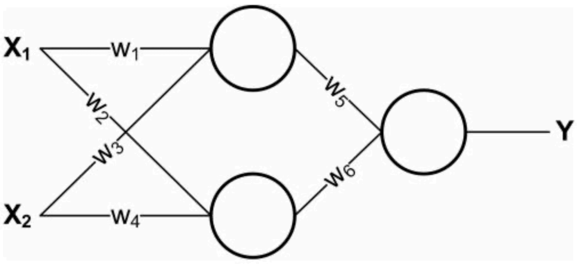

Figure 2: The Neural Network for Question 6

(a) Say that the network is using linear units: that is, defining W and and A as above, the output of a unit is $C * W^{\mathrm{T}} A$ for some fixed constant C. Let the weight values w_i be fixed. Re-design the neural network to compute the same function without using any hidden units. Express the new weights in terms of the old weights and the constant C.

(b) Is it always possible to express a neural network made up of only linear units without a hidden layer? Give a one-sentence justification.

(c) Another common activation function is a threshold, where the activation is $t(W^{\mathrm{T}} A)$ where $t(x)$ is 1 if $x > 0$ and 0 otherwise. Let the hidden units use sigmoid activation functions and let the output unit use a threshold activation function. Find weights which cause this network to compute the XOR of X_1 and X_2 for binary-valued X_1 and X_2. Keep in mind that there is no bias term for these units.

Solution (a)

$$y = [X_1 \;\; X_2] \begin{bmatrix} C(w_1 * w_5 + w_2 * w_6) \\ C(w_3 * w_5 + w_4 * w_6) \end{bmatrix}$$

Therefore the new weight is

$$w_1' = C(w_1 * w_5 + w_2 * w_6)$$
$$w_2' = C(w_3 * w_5 + w_4 * w_6)$$

Solution (b)

No, it is not always possible to express a neural network made up of only linear units without a hidden layer as it would be equivalent to a single-layer perceptron, which cannot capture non-linear relationships in the data.

Solution (c)

Inequations:

$$\begin{aligned} w_5 * \sigma(w_1) + w_6 * \sigma(w_2) &> 0 \\ w_5 * \sigma(w_3) + w_6 * \sigma(w_4) &> 0 \\ w_5 * \sigma(w_1 + w_3) + w_6 * \sigma(w_2 + w_4) &\leq 0 \\ w_5 + w_6 &\leq 0 \end{aligned}$$

A solution to this system is

$$W = \begin{bmatrix} -1.180893 \\ -0.859961 \\ -1.121304 \\ -0.829760 \\ 0.884250 \\ -0.954182 \end{bmatrix}$$

The code for randomly seeking the weight vector is shown below:

```c
#include <math.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>

#define MYRAND(x) (((double)rand() / RAND_MAX) * (x) * 2 - (x))
#define SIGMOID(x) (1 / (1 + exp(x)))

bool satisfy(double w[])
{
    return
    (
      w[4] * SIGMOID(w[0]) + w[5] * SIGMOID(w[1]) > 0 &&
      w[4] * SIGMOID(w[2]) + w[5] * SIGMOID(w[3]) > 0 &&
      w[4] * SIGMOID(w[0] + w[2]) + w[5] * SIGMOID(w[1] + w[3]) <= 0 &&
      w[4] + w[5] <= 0
    );
}

double w[6] = {0};

int main()
{
    srand(42);
    while (!satisfy(w))
    {
        for (int i = 0; i < 6; i++)
        {
            w[i] = MYRAND(1.5);
        }
    }
    printf("[%lf, %lf, %lf, %lf, %lf, %lf]", w[0], w[1], w[2], w[3], w[4], w[5]);
}
```