CS405 Homework 5

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 1

(a) In a regression problem with multiple target variables where the target's conditional distribution is Gaussian, we have:

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

For a single data point $(\mathbf{x}_n, \mathbf{t}_n)$, the likelihood is expressed as:

$$p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^\top \mathbf{\Sigma}^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))\right\}$$

where D is the dimensionality of $\mathbf{t}$, and $|\mathbf{\Sigma}|$ is the determinant of $\mathbf{\Sigma}$.

The maximum likelihood solution for $\mathbf{w}$ involves minimizing the error function, which is the negative log of the product of individual likelihoods:

$$E(\mathbf{w}) = -\log \prod_{n=1}^{N} p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w})$$

This simplifies to:

$$E(\mathbf{w}) = \frac{N}{2}\log|\mathbf{\Sigma}| + \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^\top \mathbf{\Sigma}^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w})) + \text{const}$$

For fixed and known $\mathbf{\Sigma}$, the error function to minimize is:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))^\top \mathbf{\Sigma}^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n, \mathbf{w}))$$

(b) For determining $\mathbf{\Sigma}$ from data, we differentiate the log-likelihood with respect to $\mathbf{\Sigma}$ and set it to zero:

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\left(-\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))^\top \boldsymbol{\Sigma}^{-1}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))\right) = 0$$

This leads to the equation:

$$-\frac{N}{2}\boldsymbol{\Sigma}^{-1} + \frac{1}{2}\boldsymbol{\Sigma}^{-1}\left(\sum_{n=1}^{N}(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))(\mathbf{t}_n - \mathbf{y}(\mathbf{x}_n,\mathbf{w}))^\top\right)\boldsymbol{\Sigma}^{-1} = 0$$

Solving this equation for $\boldsymbol{\Sigma}$ gives the maximum likelihood solution:

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

Question 2

The error function for binary classification problems was derived for a network having a logistic-sigmoid output activation function, so that $0 \leq y(\mathbf{x},\mathbf{w}) \leq 1$, and data having target values $t \in \{0,1\}$. Derive the corresponding error function if we consider a network having an output $-1 \leq y(\mathbf{x},\mathbf{w}) \leq 1$ 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:

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

Solution 2

In this scenario, we need to modify the error function for a network with an output range of $[-1,1]$ and target values of $t=1$ for class $\mathcal{C}_1$ and $t=-1$ for class $\mathcal{C}_2$.

The provided error function is based on the cross-entropy error function suited for binary classification with outputs in the range $[0,1]$ and targets of 0 or 1. We need to adapt this function to align with our network's output range and target values.

To achieve this, we can transform the network output y from the range $[-1,1]$ to $[0,1]$ using the transformation:

$$y' = \frac{y+1}{2}$$

Here, y' is the transformed output, now in the range $[0,1]$. Similarly, we transform the target values from $t \in \{-1,1\}$ to $t' \in \{0,1\}$ with:

$$t' = \frac{t+1}{2}$$

By applying these transformations, we can use the original logistic-sigmoid error function. Substituting y' and t' into the given error function yields:

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\left\{t'_n \ln y'_n + (1-t'_n)\ln(1-y'_n)\right\}$$

For the output unit activation function, the hyperbolic tangent function ($\tanh$) is an appropriate choice, as it maps inputs to the range [-1, 1]:

$$y(\mathbf{x},\mathbf{w}) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

where z is the input to the output unit before activation. The $\tanh$ function is a rescaled version of the logistic-sigmoid function, making it suitable for scenarios where the output range is [-1, 1].

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

Solution 3

(a) Verification of the Conditional Mean:

For a Gaussian mixture model, the probability density function given $\mathbf{x}$ is expressed as:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathcal{N}(\mathbf{t}; \mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x}))$$

The conditional expectation $\mathbb{E}[\mathbf{t}|\mathbf{x}]$ is obtained by integrating the product of $\mathbf{t}$ and $p(\mathbf{t}|\mathbf{x})$ over $\mathbf{t}$:

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

(b) Verification of the Conditional Variance:

The conditional variance $s^2(\mathbf{x})$ can be expressed as the expected value of the squared deviation of $\mathbf{t}$ from its conditional mean:

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

Expanding the squared term and solving the integral, we find:

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

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

$$\sigma(w_1 + w_2) <= threshold$$

$$\sigma(w_2) <= threshold$$

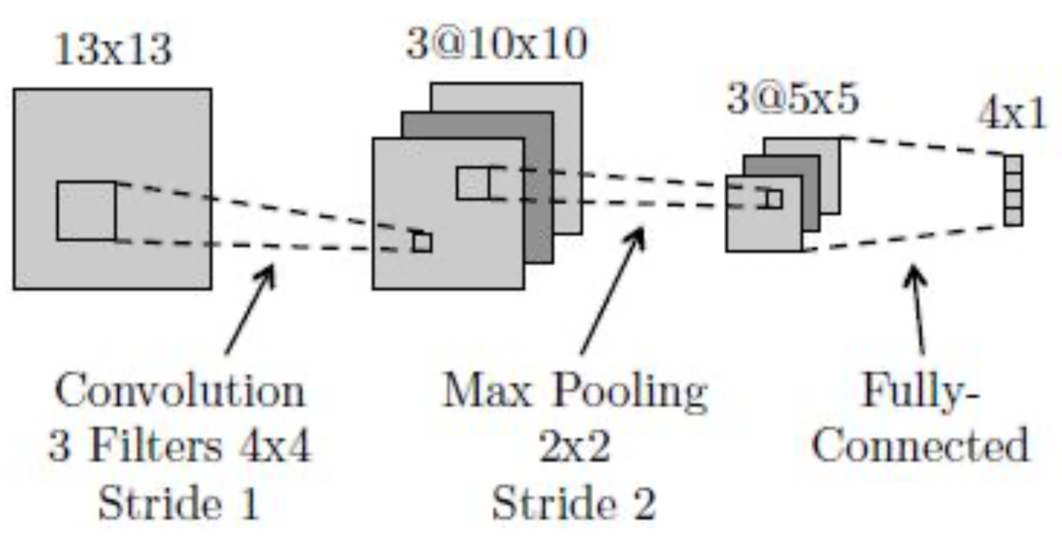
$$\sigma(w_1) > threshold$$

The function can be represented by a sigmoid+threshold unit with threshold 0.5, and the weights are:

$$[w_1, w_2] = [1.77, -12.18]$$

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.



(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 $(13 \times 13 \rightarrow 3 \times 10 \times 10 \rightarrow 3 \times 5 \times 5 \rightarrow 4 \times 1)$ 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 5

(a) Convolutional Layer Weights:
Each of the 3 filters in the convolutional layer is of size 4x4, leading to a total of:

$$3 \times 4 \times 4 = 48 \text{ weights}$$

(b) ReLU Operations:
ReLU is applied element-wise to the output of the max pooling layer, which is of size 3x5x5. Thus, the total number of ReLU operations is:

$$3 \times 5 \times 5 = 75$$

(c) Total Weights in the Network:
The convolutional layer has 48 weights. The fully connected layer connects 75 neurons (from the previous layer) to 4 output neurons, requiring:

$$75 \times 4 = 300 \text{ weights}$$

Adding these, the total weights in the network are:

$$48 + 300 = 348 \text{ weights}$$

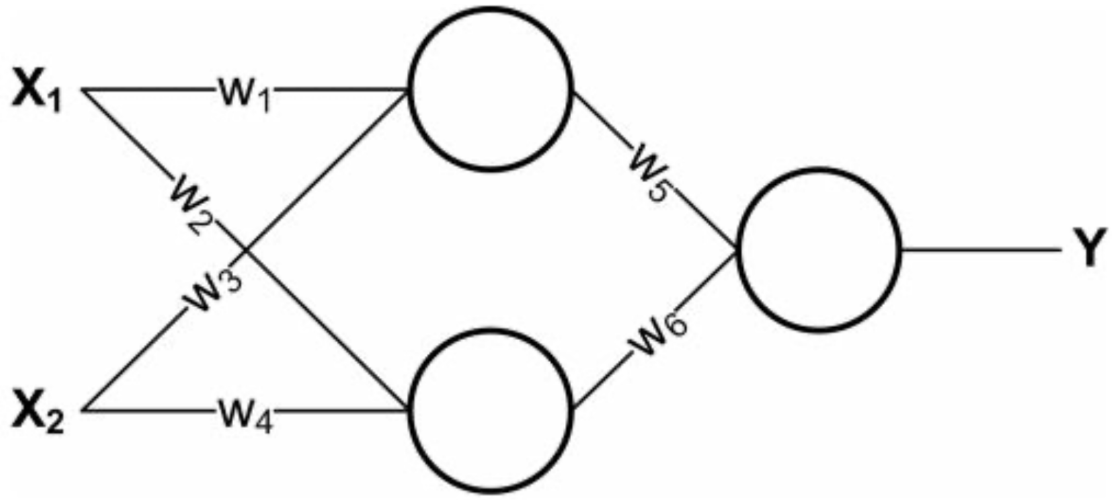
(d) Fully-Connected Network as Universal Approximator:
False. A fully-connected network with the same layer sizes as specified does not guarantee the ability to represent any classifier. The network's representational power depends on its depth, width, and non-linear activation functions.

(e) Disadvantages of Fully-Connected Network Compared to CNN:

1. **Parameter Efficiency:** FCNNs have more parameters due to the lack of weight sharing.
2. **Spatial Hierarchy:** FCNNs do not preserve spatial relationships in the data.
3. **Computational Efficiency:** CNNs are more computationally efficient due to fewer parameters and operations like pooling.
4. **Generalization:** CNNs generalize better to new images as they learn translation-invariant features.

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 $(1 + \exp(W^T A))^{-1}$. 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:



(a) Say that the network is using linear units: that is, defining W and and A as above, the output of a unit is $C * W^T A$ 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 theshold, where the activation is $t(W_T A)$ 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 6

(a) In a linear neural network, the output is a linear function of inputs. For a unit with output $C \times W^T A$, where C is a constant, W is the weight vector, and A is the input vector, the network can be simplified to eliminate hidden layers. The new weights for direct connections from inputs X_1 and X_2 to the output are $W' = C \times W$.

(b) It is not always possible to express a neural network with only linear units as a network without hidden layers. This is because linear networks can only model linear relationships, and many functions require non-linear modeling.

(c) For an XOR operation using a network with sigmoid activation in the hidden layer and a threshold activation in the output layer, we can set the weights as follows:

- Let the first hidden unit activate for $X_1 = 1$ and $X_2 = 0$.
- Let the second hidden unit activate for $X_1 = 0$ and $X_2 = 1$.
- The output unit should activate if either (but not both) hidden units are active.

A possible weight configuration is:

$$w_1, w_2, w_3, w_4, w_5, w_6 = (3.47, 13.84, 0.66, 13.65, -12.94, 12.64)$$

This ensures that the output is 1 for either $X_1 = 1, X_2 = 0$ or $X_1 = 0, X_2 = 1$, matching XOR behavior.