Module 4F12: Computer Vision and Robotics

Examples Paper 4

Straightforward questions are marked †
Tripos standard (but not necessarily Tripos length) questions are marked *

1. † Entropy and information

Consider the weighing problem introduced in the lecture. When some people first encounter this problem, they think that weighing six balls against six balls is a good first weighing, others disagree claiming that weighing six against six conveys no information at all. Explain to the second group why they are both right and wrong. Compute the information gained about which is the odd ball, and the information gained about which is the odd ball and whether it is heavy or light.

2. * Decision Trees

Consider constructing a decision tree for classifying images into one of two classes. A binary valued variable $x$ indicates the image class. At the current node in the decision tree, three queries are being compared. Each query, $i$, returns a binary value denoted by $y_i$. Probability tables showing the joint distributions over image class value and query values, $p_i(x, y_i)$, for the three queries are shown below.

(a) Compute the entropy of the class values, $H(X)$, in each case. Explain your results.

(b) Compute the entropy of the query values, $H(Y)$, in each case. Comment on whether this entropy is a sensible criterion for selecting the best query.

(c) Compute the conditional entropy, $H(X|Y)$, and argue which query is the most informative.

\[
\begin{array}{c|cc}
 p_1(x, y_1) & x = 0 & x = 1 \\
 y_1 = 0 & 1/4 & 1/4 \\
 y_1 = 1 & 1/4 & 1/4 \\
\end{array}
\]

\[
\begin{array}{c|cc}
 p_2(x, y_2) & x = 0 & x = 1 \\
 y_2 = 0 & 3/8 & 1/2 \\
 y_2 = 1 & 1/8 & 0 \\
\end{array}
\]

\[
\begin{array}{c|cc}
 p_3(x, y_3) & x = 0 & x = 1 \\
 y_3 = 0 & 3/8 & 1/8 \\
 y_3 = 1 & 1/8 & 3/8 \\
\end{array}
\]
3. *Neural networks*

(a) Consider a single neuron that takes a multi-dimensional input $z$ forms the scalar product with a set of weights $w$ and passes this through a sigmoid activation function,

$$x(z; w) = \frac{1}{1 + \exp(-w^T z)}.$$  

The neuron can be trained on a set of example inputs $\{z^{(n)}\}_{n=1}^{N}$ and outputs $\{t^{(n)}\}_{n=1}^{N}$ by minimising the relative-entropy,

$$G(w) = -\sum_n \left[ t^{(n)} \log x(z^{(n)}; w) + (1 - t^{(n)}) \log (1 - x(z^{(n)}; w)) \right].$$

Show that the derivatives of the relative-entropy are given by,

$$\frac{d}{dw} G(w) = -\sum_n (t^{(n)} - x^{(n)}) z^{(n)}.$$  

The single neuron classifier is also called logistic regression.

(b) Consider applying a neural network to scene classification in which images are assigned to one of a number of possible categories. The targets are now represented by a vector $t$ which has a single element set to 1, indicating the correct scene, and all other values are set to 0. Consider a soft-max network in which the output of the network is a vector, $x$, with elements given by a soft-max function,

$$x_i(z; w) = \frac{\exp(w_i^T z)}{\sum_j \exp(w_j^T z)}.$$  

i. Interpreting the output of the network as $x_i = p(t_i = 1|w, z)$ write down a cost-function for training this network based on the log-probability of the training data given the weights $w$ and inputs $\{z^{(n)}\}_{n=1}^{N}$.

ii. What is the relationship between this network and the one described in the first part of this question?

4. *Convolutional neural networks*

Consider applying a simple convolutional neural network to a 1D image. Let $z_{i}^{(n)}$ denote the $n$th image in the training set. The network has a single convolutional stage containing a single convolutional weight, $w_k$,

$$a_{i}^{(n)} = \sum_k w_k z_{i-k}^{(n)}.$$
The non-linear stage uses a point-wise non-linearity, \( y_i = f(a_i) \). There is no pooling stage. The readout occurs via a logistic function that pools across the convolutional layer,

\[
x^{(n)} = \frac{1}{1 + \exp(-\sum_i v_i y_i)}.
\]

The network can be trained on a set of example inputs \( \{z^{(n)}\}_{n=1}^N \) and outputs \( \{t^{(n)}\}_{n=1}^N \) by minimising the relative-entropy,

\[
G(w, v) = -\sum_n \left[ t^{(n)} \log x(z^{(n)}; w, v) + (1 - t^{(n)}) \log \left(1 - x(z^{(n)}; w, v)\right) \right].
\]

Show that the derivatives of the objective function with respect to the convolutional weights, \( w_k \), can themselves be computed efficiently using convolutions.