UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE : **COMP0171**

ASSESSMENT PATTERN : **COMP0171A7PD**

MODULE NAME : **COMP0171 - Bayesian Deep Learning**

LEVEL: : **Postgraduate**

DATE : **26-May-2022**

TIME : **10:00**

Controlled Condition Exam: 3 Hours exam

You cannot submit your work after the date and time shown on AssessmentUCL – you must ensure to allow sufficient time to upload and hand in your work

This paper is suitable for candidates who attended classes for this module in the following academic year(s):

Year
2021/22

Additional material	N/A
Special instructions	N/A
Exam paper word count	N/A

TURN OVER

COMP0171: Bayesian Deep Learning

Final Exam (Main examination period)

For 2021–2022 Cohort, all levels

Always provide justification and show any intermediate work for your answers. A correct but unsupported answer may not receive any marks.

For questions which ask for a short answer (or a derivation), please support your reasoning, but it is not necessary to write a long essay. A few clear lines will suffice.

The exam includes 4 questions in multiple parts, worth a total of 100 marks. All questions must be answered. For multi-part questions, you should try to answer later parts of the question even if you cannot complete one of the earlier parts. The marks available for each part of a question are indicated in the square brackets.

1. **Bayes' Rule and Classifier Performance.** A recently announced machine-learning based smartphone app claims it can detect whether a person has COVID-19, based on audio recordings of coughs. It is reported to have sensitivity, or true positive rate, of 98.5% and a specificity, or true negative rate, of 94.2%. That is,

$$p(\text{positive test}|\text{infection} = 0.985$$

and

$$p(\text{negative test}|\text{no infection} = 0.942$$

   (a) At the time the app was developed, the base rate of COVID-19 infections in the overall population was estimated to be 0.3%, i.e. 3 out of 1000 people. If someone tests positive, what is the probability they are infected? *[5 marks]*

   (b) If someone tests negative, what is the probability they are **not** infected? *[5 marks]*

   (c) Should this system be deployed? Why or why not? How could such an app be useful as part of a larger public health approach? *[5 marks]*

[15 marks total]

2. **Bayesian last layer.** Suppose we define a deep learning model which takes input and label pairs $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$, and

$$\mathbf{h}_i = g(\mathbf{A}\mathbf{x}_i + \mathbf{b})$$
$$\hat{y}_i = \mathbf{w}^\top \mathbf{h}_i.$$

Here, $\mathbf{A} \in \mathbb{R}^{H \times D}$, $\mathbf{b} \in \mathbb{R}^H$, and $\mathbf{w} \in \mathbb{R}^H$. We define a Gaussian prior distribution and likelihood

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})$$
$$p(y_i|\hat{y}_i) = \mathcal{N}(y_i|\hat{y}_i, \sigma^2).$$

Our goal in this question is to perform Bayesian inference over $\mathbf{w}$ (i.e., compute the posterior $p(\mathbf{w}|\mathcal{D})$) and optimize (i.e., find point estimates of) the parameters $\mathbf{A}$ and $\mathbf{b}$ in the hidden representation. This is the "Bayesian inference over the last layer only" setting which we discussed in the lectures.

To make notation simpler, we will assume that $\sigma^2 = 1$ is a known, fixed parameter which does not need to be learned.

(a) *Person A* has an idea for how to fit this model, in two stages: **First**, find maximum likelihood estimates of $\hat{\mathbf{A}}$ and $\hat{\mathbf{b}}$, and maximum a posteriori (MAP) estimates of $\hat{\mathbf{w}}$. **Second**, with $\hat{\mathbf{A}}, \hat{\mathbf{b}}$ fixed, compute $p(\mathbf{w}|\mathcal{D}, \hat{\mathbf{A}}, \hat{\mathbf{b}})$. Please fully describe this approach:

i. What is the objective function to optimize for the first step? (Write it down!) How would you optimize it? *[5 marks]*

ii. How would you compute the posterior in the second step? Characterize $p(\mathbf{w}|\mathcal{D}, \hat{\mathbf{A}}, \hat{\mathbf{b}})$ in as much detail as you can (include math!). *[5 marks]*

iii. What do you think of this approach — do you think this is a good proposal? Identify at least one positive and one negative, and give your opinion. *[5 marks]*

(b) *Person B* has a different idea: they say it is important to fit this model using a single, unified objective function, which fits the posterior over $\mathbf{w}$ simultaneously with estimating $\hat{\mathbf{A}}, \hat{\mathbf{b}}$.

i. How do you suggest going about this? Propose an objective function that you would optimize in order to find $\hat{\mathbf{A}}, \hat{\mathbf{b}}$, and (very important!) **explain why** this is an appropriate objective. Include any derivations if necessary. *[5 marks]*

ii. How would you go about optimizing this objective? *[5 marks]*

iii. Do you think this is a better idea than the two-stage proposal from *Person A*? Why or why not? Identify at least one positive and one negative.

[5 marks]

(c) Suppose that this, instead, were a binary classification problem, i.e. with $y_i \in \{0, 1\}$, with a sigmoid function on the output layer and a Bernoulli likelihood. What, if anything, might change regarding your answers to the previous two parts? Give an overview of how you would recommend fitting a deep learning model with Bayesian inference over the last layer, for this binary classification setting. (You can just describe this in words — no need for more math.)

[10 marks]

[40 marks total]

3. **Gradients and automatic differentiation.**

(a) We talked about two different approaches to automatic differentiation: *forward mode* and *reverse mode*. Explain the difference between forward and reverse mode — in particular, in what contexts would one prefer one over the other? Which is typically applied to deep neural networks, and why? *[6 marks]*

(b) Consider the following function of two variables:

```
def my_func(a, b):
    b = b.abs()
    x = dist.Normal(a, 1).rsample()
    if x > 1:
        y = dist.Exponential(x).rsample()
    else:
        y = my_func(a + b + 1)
    return (x + y)**2
```

Both inputs a, b are scalar. Which of the following could be an appropriate strategies for computing the gradient of the function output of `my_func` with respect to a? Explain your reasoning for each:

- Symbolic differentiation
- Finite differences
- Forward-mode automatic differentiation
- Reverse-mode automatic differentiation

[12 marks]

(c) Suppose you would like to minimize the following loss function with respect to $\theta \in \mathbb{R}$, expressed as an expectation over a random variable z:

$$\mathcal{L}(\theta) = \mathbb{E}_{p(z|\theta)}[f(\theta, z)].$$

i. Suppose f is a differentiable function of both θ and z, where $z \in \mathbb{R}$. Define a Monte Carlo estimator for the derivative $\frac{\partial}{\partial\theta}\mathcal{L}(\theta)$, which can be evaluated using samples $z_1, \ldots, z_K \sim p(z|\theta)$. (Include your derivation!) *[5 marks]*

ii. Now additionally suppose that the distribution $p(z|\theta)$ can be re-parameterized, so that samples $z \sim p(z|\theta)$ can be drawn using the procedure

$$\epsilon \sim p(\epsilon)$$
$$z = g(\epsilon, \theta)$$

for a continuous function g, which is differentiable with respect to θ. Define and derive an alternative Monte Carlo estimator, which can be evaluated using samples $\epsilon_1, \ldots, \epsilon_K \sim p(\epsilon)$. *[5 marks]*

iii. Which of the two estimators would you likely prefer to use? Why? *[2 marks]*

[30 marks total]

4. **Uncertainty quantification.**

(a) Suppose you would like to use deep learning to look at satellite imagery, for localizing and counting sheep in the British countryside, as viewed from space. Specifically, you will use a U-net architecture which takes in an image and produces a segmentation, assigning each pixel either a value of either 1 or 0, indicating whether that particular pixel contains a sheep (1) or does not contain a sheep (0). The satellite in question has a 1 meter spatial resolution, i.e. each pixel corresponds to a 1×1 meter square area on the surface; it captures both the visual color spectrum as well as infrared and other non-visible bands. For training data, we have pairs of satellite images with sheep geolocation data, as supplied by collars with GPS tags.

i. Give an informal definition, in your own words, of both *epistemic uncertainty* and *aleatoric uncertainty*. *[2 marks]*

ii. For each of the following potential sources of errors in predictions, argue whether the source of uncertainty is epistemic or aleatoric in this context:

A. Three sheep are standing very close together, and thus all fall within the same one-square-meter pixel cell in the satellite imagery.

B. Most of the satellite photos used as training data show sheep in grassy fields; predictions are less accurate for sheep that are on rocky terrain.

C. The fields that contain these sheep which we are tracking, also contain a large number of cows and goats; the differences between sheep and other similar-sized mammals are fairly subtle when viewed from a satellite.

D. Sheep which are sleeping under a tree are not visible, due to occlusion from leaves, and thus are missed by the predictive model. *[8 marks]*

iii. For each input image $\mathbf{X}$, the corresponding labels form a matrix $\mathbf{Y}$. At each pixel location (i, j), we have an input $\mathbf{x}_{i,j} \in \mathbb{R}^D$ and a label $y_{i,j} \in \{0, 1\}$. Suppose the U-net f_θ has parameters θ, where the model is defined as

$$\begin{aligned} \mathbf{Z} &= f_\theta(\mathbf{X}) \\ y_{i,j} &\sim \text{Bernoulli}(z_{i,j}). \end{aligned}$$

Each $z_{i,j}$ corresponds to the probability that the image location $\mathbf{x}_{i,j}$ contains sheep. Suppose we have estimated an approximate posterior distribution over the parameters, $q(\theta)$. Make (and justify) a proposal for how to numerically quantify the **epistemic uncertainty** in the predictions made for a new input $\mathbf{X}$. *[5 marks]*

[15 marks total]

Question	Points Scored	Max Points
Bayes' rule and classifier performance		15
Bayesian last layer		40
Gradients and automatic differentiation		30
Uncertainty quantification		15
TOTAL		100