UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE	:	**COMP0171**
ASSESSMENT Pattern:		**COMP0171A7PE**
MODULE NAME	:	**Bayesian Deep Learning**
LEVEL:	:	**Postgraduate**
DATE:	:	**02-May-2023**
TIME	:	**10:00**
DURATION	:	**03:15**

Late submission is permitted for Controlled Conditioned exams but late penalties will apply - any submissions that are up to 40 minutes late will be penalised, after which no submissions will be accepted under any circumstances.
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
2022-23

Duration	<< Exam Duration>>
Additional time for converting handwritten notes to PDF where applicable	**15 mins**
Upload window	**20**
Total time	**3 Hours 35 mins**

Additional material	N/A
Special instructions	N/A

TURN OVER

COMP0171: Bayesian Deep Learning

Final Exam (Main summer examination period)

For all cohorts and 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 5 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. **True/False**

   Please identify if statements are either True or False. Please justify your answer **if false**. Correct "True" questions yield 2 points. Correct "False" questions yield one point for the answer and one point for the justification.

   (a) **(T/F)** Variational Bayes with a Gaussian posterior, is preferred over a Laplace approximation, because the variational Bayes method is able to capture multiple modes of posterior. *[2 marks]*

   (b) **(T/F)** Suppose you are training a machine learning model to identify the specific breed of a dog from a photo. The model mostly works well, but has low-confidence predictions on very rare breeds which only are present in a few training images. This low confidence is an example of epistemic uncertainty, which could be captured and quantified by using a Bayesian neural network. *[2 marks]*

   (c) **(T/F)** Bayesian deep learning is particularly advantageous over non-Bayesian ("regular") deep learning when datasets are very very large, leading to more accurate predictions. *[2 marks]*

   (d) **(T/F)** You run into a long-lost friend who now has two children. You ask if at least one of the children is a boy, and the answer is yes. She shows you pictures of the children one at a time, first showing you a picture of a son. With only this information, and before being told anything about the other child, the probability that this child is also a boy is 50%. *[2 marks]*

   (e) **(T/F)** Suppose you fit a Bayesian neural network for a regression problem, with a Gaussian likelihood (i.e. a squared error loss). If we make a Gaussian approximation to the posterior over model weights, then the posterior predictive distribution for new datapoints is a mixture of Gaussians. *[2 marks]*

   (f) **(T/F)** You train a deep learning model for binary classification, and on a set of 100 challenging test inputs, there was high uncertainty in predictions (predicting one class or another with around 60% probability). However, on further examination, it turns out that it got over 90% of the class labels correct! This high accuracy shows that the model has fit the data well. *[2 marks]*

   (g) **(T/F)** You are interested in training a spam filter, to identify and filter undesirable emails. You have a dataset of emails, but require a human labeller to come in and manually tell you what emails are spam and which are not. This is a setting where active learning might be appropriate. *[2 marks]*

   (h) **(T/F)** Reparameterized gradient estimators for $\nabla_\theta \mathbb{E}_{p(\mathbf{x}|\theta)}[f(\theta)]$ can be applied so long as two conditions hold:

- f is differentiable with respect to θ;
- There exists a function g and a distribution $p(\epsilon)$ such that sampling $\epsilon \sim p(\epsilon)$ and computing $\mathbf{x} = g(\theta, \epsilon)$ yields a sample $\mathbf{x}$ from $p(\mathbf{x}|\theta)$.

[2 marks]

[16 marks total]

2. **Lightning round: short answer.** For each question, write a **brief** response.

   (a) Give **two** reasons to prefer reverse-mode automatic differentiation over forward-mode in training deep learning models, and **one** example of a setting other than in training deep learning models where forward-mode might be preferable instead. *[6 marks]*

   (b) Give one advantage and one disadvantage of using Markov chain Monte Carlo (MCMC) to estimate the posterior over weights in a Bayesian neural network. *[4 marks]*

   (c) A "deep ensemble" is a collection of trained deep learning models, each with weights optimized by gradient descent from a different initialization. In a Bayesian interpretation of the ensemble as approximate posterior inference, describe

      i. how you would evaluate the posterior predictive distribution on a new test point; and

      ii. what the form the predictive distribution might have in a binary classification problem.

   [4 marks]

[14 marks total]

3. **Deep learning architectures.** In a deep learning model, suppose we have a hidden layer defined as

$$\mathbf{h} = g(\mathbf{W}\mathbf{x} + \mathbf{b})$$

where $\mathbf{W}, \mathbf{b}$ are weights, $\mathbf{x}$ is an input, and $g(\cdot)$ is an elementwise function.

(a) If $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{h} \in \mathbb{R}^M$, how many parameters does this layer have? *[2 marks]*

(b) Consider three possible choices of function $g(\cdot)$: the ReLU $g(z) = \max(z, 0)$, the sigmoid $g(z) = \tanh(z)$, and the identity $g(z)$. For each of these three options, give one advantage and one disadvantage of this choice. *[6 marks]*

(c) Suppose we stack two of these layers, as follows:

$$\mathbf{h}_1 = g(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}_2 = g(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$

The input $\mathbf{x}$ is used to compute the hidden layer $\mathbf{h}_1$, which is then used to compute the next hidden layer $\mathbf{h}_2$.

i. How many parameters are in these two layers combined, assuming both $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^M$? What might be an advantage of using two layers, rather than one single layer with more hidden units? *[4 marks]*

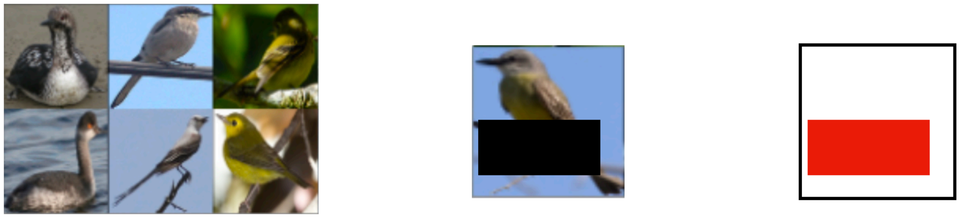
ii. Suppose you want to add a **skip connection** which goes directly from $\mathbf{x}$ to $\mathbf{h}_2$, bypassing $\mathbf{h}_1$. How would you modify the two equations above? You can add a single additional parameter matrix, $\mathbf{V}$. *[3 marks]*

iii. Why would you add this skip connection — what might doing this accomplish? *[2 marks]*

[15 marks total]

4. **Generative models and missing data.** In this question, suppose we have first fit a generative model to image data. We will then try to use that model for "inpainting" missing regions in a test image.

   For example, suppose our training data is like figure (i) in the image below, but at test time we are given figure (ii), plus a binary map in figure (iii) which indicates which pixels are "missing":



(i) Example training images   (ii) Obscured test image   (iii) Binary mask

   Our goal is to estimate the values of these missing pixels in the test image.

   Let $\mathbf{x}_1, \ldots, \mathbf{x}_N$ denote a set of N training images. For a test image $\mathbf{x}$, let $\mathbf{x}^v$ denote the subset of pixels which are visible, and $\mathbf{x}^m$ denote those which are missing. We would like to estimate the value of the missing pixels, given the value of the visible pixels: that is, we are interested in $p(\mathbf{x}^m|\mathbf{x}^v)$.

   (a) Suppose we had access to a trained probabilistic model $p(\mathbf{x})$ which is tractable, in the sense that we can evaluate the probability for any input $\mathbf{x}$.

      i. Give two examples of generative models we discussed where this is possible, including a deep generative model if possible. *[3 marks]*

      ii. Describe a possible Metropolis-Hastings MCMC algorithm to sample from $p(\mathbf{x}^m|\mathbf{x}^v)$. Include the acceptance ratio, and a description of any proposal distributions. *[5 marks]*

      iii. Do you think this is a good approach to estimating the missing pixels in the test image? *[2 marks]*

   (b) Now suppose we instead model the training data with a latent variable model; in particular, we will use a variational autoencoder model with latent variables $\mathbf{z}_i$. As in the coursework and lectures, this model has a prior $p(\mathbf{z})$ over the latent variables, a likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ defined by a deep neural network with parameters θ, and an approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ defined by an inference network with parameters ϕ.

      i. Would the MCMC scheme you described in part (a) work for sampling from

$p_\theta(\mathbf{x}^m|\mathbf{x}^v)$, where $p_\theta(\mathbf{x})$ is defined using this latent variable model? Why or why not? *[3 marks]*

ii. Consider an alternative MCMC scheme for latent variable models, which targets $p_\theta(\mathbf{x}^m, \mathbf{z}|\mathbf{x}^v)$. That is, it draws samples of both $\mathbf{z}$ and $\mathbf{x}^m$, conditioned on the visible $\mathbf{x}^v$. Describe such an approach: what would the Metropolis-Hastings acceptance ratio be, and what would be good choices of proposal distributions for $\mathbf{x}^m$ and $\mathbf{z}$? *[5 marks]*

iii. Does this alternative MCMC scheme draw samples from $p_\theta(\mathbf{x}^m|\mathbf{x}^v)$? Why or why not? *[2 marks]*

(c) Instead of sampling values of $\mathbf{x}^m$, suppose instead you want to find the optimal values for the missing pixels by maximizing $\log p_\theta(\mathbf{x}^m|\mathbf{x}^v)$ in the variational autoencoder model from part (b). Assume that $p_\theta(\mathbf{x})$ is differentiable with respect to pixel values $\mathbf{x}$, and that our goal is to

i. Derive a tractable objective function for gradient-based optimization of $\mathbf{x}^m$, by treating the missing data as a latent variable, and derive a lower bound of $\log p_\theta(\mathbf{x}^m|\mathbf{x}^v)$. (Hint: remember how we derived the ELBO!) *[7 marks]*

ii. Describe how you would optimize this objective using gradient-based methods, and write down an estimator for the gradient of this objective. *[3 marks]*

iii. As an alternative approach, suppose we directly tried to estimate the gradient $\nabla_{\mathbf{x}^m} p(\mathbf{x}^m|\mathbf{x}^v)$. Can you suggest a Monte Carlo estimator for this gradient? (Hint: consider a likelihood weighting or importance sampling approach!) *[5 marks]*

[35 marks total]

5. **Linearizing predictions** In class, we talked about "linearizing" networks when making predictions. In particular, suppose we have data $\mathcal{D}$ with inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and real-valued labels $y_1, \ldots, y_N$, and consider the probabilistic model

$$\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$$
$$y_i|\mathbf{x}_i \sim \mathcal{N}(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \beta^{-1})$$

where $f_{\boldsymbol{\theta}}$ is a deep network. Suppose we have fit an approximate posterior

$$q(\boldsymbol{\theta}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

(a) For a new test input $\mathbf{x}'$ with unknown label y', write out the predictive distribution $p(y'|\mathbf{x}', \mathcal{D}, \alpha, \beta)$ as a function of our posterior approximation $q(\boldsymbol{\theta}|\mathcal{D})$. *[Note: this will involve an intractable integral, and does not have a closed form; we are just looking for the general expression.]* *[5 marks]*

(b) In class, we considered the approximation

$$p(y'|\mathbf{x}', \mathcal{D}, \alpha, \beta) = \mathcal{N}(y'|f_{\boldsymbol{\mu}}(\mathbf{x}'), \beta^{-1} + \mathbf{g}(\mathbf{x}')^\top \boldsymbol{\Sigma}\mathbf{g}(\mathbf{x}'))$$

where $\mathbf{g}(\mathbf{x}') = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}')$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\mu}$.

i. Provide a derivation of this expression. *[5 marks]*

ii. Under what circumstances will this be a good approximation to the predictive distribution? Outline the assumptions made for this approximation, and give an example of when these assumptions might be violated. *[3 marks]*

iii. We introduced this approach in class for use with Laplace approximations. Could you use this with any Gaussian approximation to the posterior (e.g. found by variational Bayes)? Why or why not? *[2 marks]*

(c) This linearized predictor has a posterior predictive variance $\beta^{-1} + \mathbf{g}(\mathbf{x}')^\top \boldsymbol{\Sigma}\mathbf{g}(\mathbf{x}')$.

i. Provide an interpretation of the two terms: β^{-1}, and $\mathbf{g}(\mathbf{x}')^\top \boldsymbol{\Sigma}\mathbf{g}(\mathbf{x}')$. *[2 marks]*

ii. Suppose instead of a fixed value of β, the likelihood variance were input-dependent, and specified by another deep learning model $\beta_\psi(\mathbf{x})$ where ψ are trainable parameters. How does this change your interpretation? *[3 marks]*

[20 marks total]

Question	Points Scored	Max Points
True/False		16
Short answer		14
Deep learning architectures		15
Generative models and missing data		35
Linearizing predictions		20
TOTAL		100