UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE : **COMP0171**

ASSESSMENT PATTERN: **COMP0171A7PF 001**

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

LEVEL : **Postgraduate**

DATE: : **03/05/2024**

TIME : **10:00**

DURATION : **3 Hours**

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

Year
2023/24

EXAMINATION PAPER CANNOT BE REMOVED FROM THE EXAM HALL. PLACE EXAM PAPER AND ALL COMPLETED SCRIPTS INSIDE THE EXAMINATION ENVELOPE

Hall instructions	N/A
Additional materials	N/A
Standard Calculators	No
Non-Standard Calculators	No

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)** Suppose there are two random variables $\mathbf{x}$ and $\mathbf{y}$ with joint distribution $p(\mathbf{x}, \mathbf{y})$. If they have zero covariance, i.e. $\text{Cov}(\mathbf{x}, \mathbf{y}) = 0$, then they are independent, i.e. $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$. *[2 marks]*

   (b) **(T/F)** Suppose we only know a functional form of a probability distribution up to a normalizing constant, e.g. for some $p(\mathbf{x}|\theta) = \gamma(\mathbf{x}, \theta)/Z$, where θ is a parameter of the distribution, the value of Z is unknown. Then it is **not** possible to perform maximum likelihood estimation for $p(\mathbf{x}|\theta)$, maximizing with respect to θ, without first estimating Z. *[2 marks]*

   (c) **(T/F)** If we are using self-normalized importance sampling to estimate an expectation $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})]$, where the normalizing constant of $p(\mathbf{x})$ is unknown, then for any number of samples and for any proposal $q(\mathbf{x})$ the resulting estimator using weights $w(\mathbf{x}) \propto p(\mathbf{x})/q(\mathbf{x})$ is biased. *[2 marks]*

   (d) **(T/F)** Reverse-mode autodiff can have higher memory requirements than forward-mode autodiff for functions composed of many sequential operations. *[2 marks]*

   (e) **(T/F)** Adaptive learning rate approaches such as Adagrad and Adam avoid the need to manually tune any learning rates, instead setting them automatically during the optimization process. *[2 marks]*

   (f) **(T/F)** Suppose there is a deep learning model for binary classification, with a maximum likelihood estimate $\theta^\star$ and a predictive likelihood $p(y|\mathbf{x}, \theta^\star)$. If we use the Laplace approximation to estimate a posterior covariance matrix $\mathbf{\Sigma}$ and use the resulting predictive distribution that integrates over the posterior distribution $p(\theta|\theta^\star, \mathbf{\Sigma})$, this will change predictive uncertainty but will not change the decision boundary. *[2 marks]*

   (g) **(T/F)** Markov chain Monte Carlo methods cannot easily be used to estimate the marginal likelihood. *[2 marks]*

   (h) **(T/F)** Linear regression, deep learning classification models, and Gaussian process regression all have likelihoods that factorize over datapoints, i.e. with $p(y_{1:N}|\mathbf{x}_{1:N}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i)$, which allows for easy minibatching during training. *[2 marks]*

   (i) **(T/F)** Bi-directional recurrent neural networks can be used to extract features that depend on the entire length of a sequence, but cannot be used as a gener-

ative model. *[2 marks]*

(j) **(T/F)** In order for a Markov chain Monte Carlo method to correctly sample from a target distribution $p(\mathbf{x})$, the underlying Markov chain transition operator $T(\mathbf{x} \to \mathbf{x}')$ must satisfy the detailed balance condition $p(\mathbf{x})T(\mathbf{x} \to \mathbf{x}') = p(\mathbf{x}')T(\mathbf{x}' \to \mathbf{x})$. *[2 marks]*

[20 marks total]

2. **Short answer.** For each question, please write a **brief** response.

   (a) Suppose we have data $\mathcal{D}$ and parameters θ, which for a given model class $\mathcal{M}$ have a joint distribution $p(\mathcal{D}, \theta|\mathcal{M})$. Briefly describe two approaches one could use to estimate the model evidence $p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}, \theta|\mathcal{M})\mathrm{d}\theta$. *[4 marks]*

   (b) Suppose you have a deep learning model deployed on the camera for a self-driving car, which takes video of street scenes and segments it into labels, identifying the street, pavement, other cars, pedestrians, buildings, and any additional obstacles. This computer vision system has been trained on data from large amounts of data collected from city centres in the UK, both at daytime and nighttime.

      i. What failure modes do you predict might arise for this system when deployed in the real world? Give one example where you would expect the model to have poor performance due to high **aleatoric uncertainty**. *[3 marks]*

      ii. Now give one example where you would expect the model to have poor performance due to high **epistemic uncertainty**. *[3 marks]*

   (c) What is a "deep ensemble"? Why could one consider this approach a form of approximate Bayesian inference in neural networks? *[3 marks]*

   (d) Suppose we are sampling from a probability distribution $p(\mathbf{x})$ using a Langevin MCMC algorithm.

      i. Given the gradient $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, write out the Langevin dynamics proposal distribution $q(\mathbf{x}'|\mathbf{x})$ that would be used as part of a Metropolis-Hastings algorithm, and write out the MH acceptance ratio. *[4 marks]*

      ii. In class, we discussed a stochastic gradient Langevin dynamics algorithm which used a noisy estimate of the gradient, with no Metropolis-Hastings step. Why? Is this a valid MCMC sampler, and under what conditions? *[4 marks]*

   (e) Convolutional layers have the "equivariance" property, that shifting an input will correspondingly shift the output. **First,** write out the equation for a one-dimensional convolutional layer, for a single length-3 filter $\mathbf{w} \in \mathbb{R}^3$, as applied to an input $\mathbf{x} \in \mathbb{R}^D$. (That is, write the expression for the d^{th} element of the one-dimensional convolution $\mathbf{h} = \sigma(\mathbf{x} * \mathbf{w})$, for a nonlinearity σ).

      Second, show that the equivariant property holds by defining $\mathbf{x}'$ which is shifted to the right, i.e. with $x'_{d+1} = x_d$, would yield an output layer $\mathbf{h}'$ with $h'_{d+1} = h_d$. *[6 marks]*

      [N.B. For purposes of this question you can ignore edge cases at x_1 and x_D.]

(f) Active learning and Bayesian optimization both involve iteratively collecting data and then updating a model. What is the difference, and when would prefer one approach over the other? *[3 marks]*

[30 marks total]

3. **Bayesian linear classifiers.** Suppose we are considering a binary classification problem, using a Bayesian linear classifier. Assume we have a dataset of N observations $\{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, N$, with inputs $\mathbf{x}_i \in \mathbb{R}^D$ and labels $y_i \in \{0, 1\}$. We would like to fit a linear classifier with parameters $\theta \in \mathbb{R}^K$, and a set of features $\phi\, \mathbb{R}^D \to \mathbb{R}^K$. The functional form of the classifier will be

$$\hat{y}_i = \sigma(\theta^\top \phi(\mathbf{x}_i))$$

where σ is the logistic sigmoid function, and $\hat{y}_i$ is an estimate of $\Pr(y_i = 1|\mathbf{x}_i)$.

(a) Suppose we would like to fit this using maximum likelihood estimation for θ, assuming the feature map ϕ is a fixed, known function. Write down an objective function you would optimize to find the maximum likelihood estimate $\hat{\theta}_{\text{ML}}$. *[3 marks]*

(b) Now suppose we would like to regularize our estimate of θ, by performing MAP estimation (maximum a posteriori) instead of maximum likelihood. To do this we include a Gaussian prior on θ, i.e. $\theta \sim \mathcal{N}(0, \beta\mathbf{I})$ where β is a prior variance. Write down a new objective function you would optimize to find the MAP estimate of $\hat{\theta}_{\text{MAP}}$. *[3 marks]*

(c) Assuming we are performing gradient descent to optimize θ, and suppose N is very large. In terms of the likelihood $p(y_i|\mathbf{x}_i, \theta)$, define an estimator for the gradient of your objective in the part (b) which uses only $M \ll N$ data points at each step, where the subset of M points is selected at random from the full dataset. *[3 marks]*

(d) Is your stochastic gradient estimator in part (c) an unbiased estimate of the true gradient? Show why or why not. *[3 marks]*

(e) Suppose you are now optimizing this objective via stochastic gradient descent on θ, and after 10 iterations where the loss reduces, on the 11th iteration the loss goes up. Should you now stop the optimization algorithm, or continue running gradient descent further? Explain your decision. *[3 marks]*

[15 marks total]

4. **Gradient estimators for expectations.** For all of the following, you will be asked to calculate the derivative of an expectation. If there are multiple possible approaches or estimators, provide the one that you think would be the **best** choice, and briefly justify why.

   (a) Suppose you wanted to estimate the derivative with respect to θ of a function $f(x, \theta) = x\theta^2$, in expectation under a normal distribution $p(x|\theta) = \mathcal{N}(x|\theta, 1)$, i.e. to estimate

   $$\frac{\mathrm{d}}{\mathrm{d}\theta}\mathbb{E}_{p(x)}[f(x, \theta)].$$

   Write down an unbiased Monte Carlo estimator for this derivative, using S samples. Be sure to specify all necessary details, including the function to evaluate as well as any sampling distributions. *[5 marks]*

   (b) Suppose instead the distribution $p(x|\theta)$ was a geometric distribution, with

   $$p(x = k|\theta) = \theta(1 - \theta)^{k-1}.$$

   Repeat part (a), writing out an unbiased Monte Carlo estimator for the gradient of the expectation, for this new distribution. *[5 marks]*

   (c) Now suppose instead the distribution $p(x|\theta)$ is a discrete distribution that takes values $x \in \{1, 2, 3\}$ with probabilities $\theta_1, \theta_2, \theta_3$, respectively. Would you use the estimator above? If so, explain why. If not, write down an alternative computation for the gradient of the expectation. *[5 marks]*

   [15 marks total]

5. **Variational inference over weight subsets.** In class, for deep learning models with multiple hidden we considered a "Bayesian last layer" approach, where we learned a point estimate (maximum likelihood or MAP estimate) for most of the parameters of the model, but estimated a full posterior distribution for the last output layer of the network.

   This approach could also be generalized to "being Bayesian" about any subset of the weights, rather than specifically about the last layer. For a regression problem with a dataset of input and output pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$, consider a generic supervised deep learning model of the form

$$y_i \sim \mathcal{N}(f(\mathbf{x}_i, \theta), \sigma^2).$$

   Here f is some deep learning architecture, and θ are its parameters. The network outputs the mean of a Gaussian distribution over observed $y_i \in \mathbb{R}$ given an input $\mathbf{x}_i$.

   (a) Suppose we introduce a prior $p(\theta)$ over the parameters of the network, and wish to approximate the posterior using variational inference. If the approximating family has the form $q(\theta|\lambda)$, where λ are the variational parameters, write down the evidence lower bound (ELBO), a lower bound for the log marginal likelihood $\sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i)$, that you would use as an objective function for learning λ. *[2 marks]*

   (b) Now suppose we were to split the weights θ into two disjoint sets, θ_b and θ_p, where we aim to perform variational inference over θ_b and maximum a posteriori estimation for θ_p. Assuming that the prior $p(\theta)$ factorizes as $p(\theta) = p(\theta_b)p(\theta_p)$, write down an appropriate objective function for learning an approximate posterior $q(\theta_b|\lambda)$. *[4 marks]*

   (c) Is the objective you defined above also appropriate for estimating θ_p? Is it appropriate for estimating σ^2? Why or why not? Describe (high-level) the procedure you would use for estimating $q(\theta_b|\lambda)$, $\hat{\theta}_p$, and $\hat{\sigma}^2$. *[5 marks]*

   (d) Given a new test input $\mathbf{x}^\star$, what is the form of the predictive distribution $p(y^\star|\mathbf{x}^\star)$ using the model with an approximate posterior over θ_b and a point estimate for θ_p? Write this as a (possibly intractable) integral or expectation, and describe how you would approximate it to make predictions. *[5 marks]*

   (e) Give one reason you might want prefer the "Bayesian last layer" approach instead of the arbitrary splitting of weights described here, and one reason why you might prefer this approach instead. *[4 marks]*

   [20 marks total]

Question	Points Scored	Max Points
True/False		20
Short answer		30
Bayesian linear classifiers		15
Gradient estimators for expectations		15
Variational inference over weight subsets		20
TOTAL		100