University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

Spring, 23/24 TRIMESTER EXAMINATIONS

COMP47590

Advanced Machine Learning

Module Coordinator: Assoc Professor Brian Mac Namee

Student Number

Seat Number

Time Allowed: 120 minutes

Materials Permitted in the Exam Venue:
Non-programmable or scientific calculator
Programmable calculator

Materials to be Supplied to Students:
8 Page Answer Booklets
New Cambridge Statistical Tables

Instructions to Students:
Answer any three out of four questions. All questions carry equal marks. Total marks available 90. The value of each part of each question is shown in brackets next to it.

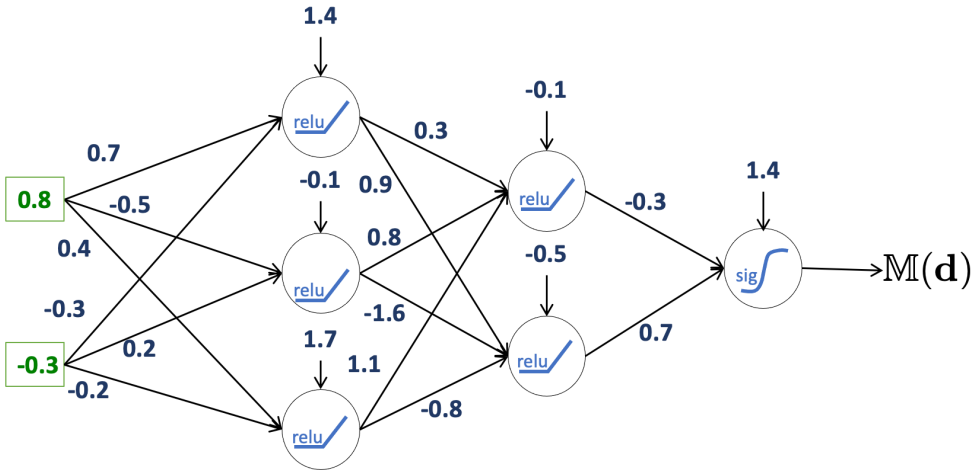
SOLUTIONS

SOLUTIONS

SOLUTIONS

SOLUTIONS

1. **(a)** The image below shows a *feed forward artificial network*. The computational units in the two hidden layers use *rectified linear* (*relu*) activation functions and the output layer unit uses a *sigmoid* activation function. The *weights* and *biases* are shown along the links in the network.



(i) Perform a **forward propagation** through the network using an input feature vector of [0.8, -0.3]. Show your workings.

[12 marks]

2 Setup Input, Weight and Bias Matrices

The network inputs:

$$\mathbf{d} = \begin{bmatrix} 0.8 \\ -0.3 \end{bmatrix}$$

Weights and biases for Layer 1:

$$\mathbf{W}^{[1]} = \begin{bmatrix} 0.7 & -0.3 \\ -0.5 & 0.2 \\ 0.4 & -0.2 \end{bmatrix}$$

$$\mathbf{b}^{[1]} = \begin{bmatrix} 1.4 \\ -0.1 \\ 1.7 \end{bmatrix}$$

Weights and biases for Layer 2:

$$\mathbf{W}^{[2]} = \begin{bmatrix} 0.3 & 0.8 & 1.1 \\ 0.9 & -1.6 & -0.8 \end{bmatrix}$$

$$\mathbf{b}^{[2]} = \begin{bmatrix} -0.1 \\ -0.5 \end{bmatrix}$$

Weights and biases for Layer 3:

$$\mathbf{W}^{[3]} = \begin{bmatrix} -0.3 & 0.7 \end{bmatrix}$$

$$\mathbf{b}^{[3]} = \begin{bmatrix} 1.4 \end{bmatrix}$$

3 Forward Propagate

To perform a forward propagation for the first layer in the network, first calculate $ extbf{z}^{[1]}$:

$$\begin{aligned}\mathbf{z}^{[1]} &= \mathbf{W}^{[1]}\,\mathbf{d} + \mathbf{b}^{[1]} \\ &= \begin{bmatrix} 0.7 & -0.3 \\ -0.5 & 0.2 \\ 0.4 & -0.2 \end{bmatrix} \begin{bmatrix} 0.8 \\ -0.3 \end{bmatrix} + \begin{bmatrix} 1.4 \\ -0.1 \\ 1.7 \end{bmatrix} \\ &= \begin{bmatrix} 2.05 \\ -0.56 \\ 2.08 \end{bmatrix}\end{aligned}$$

then apply the activation function, in this case a relu function, to calculate the activation of the nodes at Layer 1:

$$\begin{aligned}\mathbf{a}^{[1]} &= g(\mathbf{z}^{[1]}) \\ &= g\left(\begin{bmatrix} 2.05 \\ -0.56 \\ 2.08 \end{bmatrix}\right) \\ &= \begin{bmatrix} 2.05 \\ 0.0 \\ 2.08 \end{bmatrix}\end{aligned}$$

To perform a forward propagation for the second layer in the network, first calculate $ extbf{z}^{[2]}$:

$$\begin{aligned}\mathbf{z}^{[2]} &= \mathbf{W}^{[2]}\,\mathbf{a}^{[1]} + \mathbf{b}^{[2]} \\ &= \begin{bmatrix} 0.3 & 0.8 & 1.1 \\ 0.9 & -1.6 & -0.8 \end{bmatrix} \begin{bmatrix} 2.05 \\ 0.0 \\ 2.08 \end{bmatrix} + \begin{bmatrix} -0.1 \\ -0.5 \end{bmatrix} \\ &= \begin{bmatrix} 2.803 \\ -0.319 \end{bmatrix}\end{aligned}$$

then apply the activation function, in this case a extbf{relu}, to calculate the activation of the output nodes of the network:

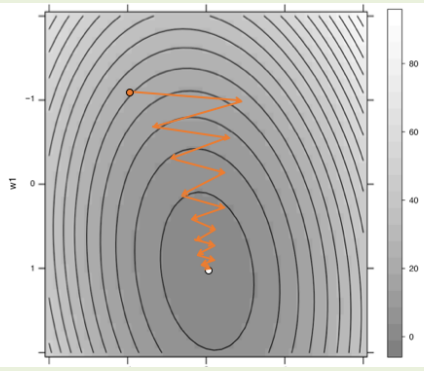
$$\begin{aligned}\mathbf{a}^{[2]} &= g(\mathbf{z}^{[2]}) \\ &= g\left(\begin{bmatrix} 2.803 \\ -0.319 \end{bmatrix}\right) \\ &= \begin{bmatrix} 2.803 \\ 0.0 \end{bmatrix}\end{aligned}$$

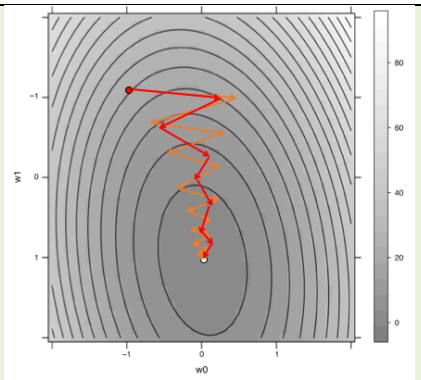
To perform a forward propagation for the third layer in the network, first calculate $ extbf{z}^{[3]}$:

$$\begin{aligned}\mathbf{z}^{[3]} &= \mathbf{W}^{[3]}\,\mathbf{a}^{[2]} + \mathbf{b}^{[3]} \\ &= \begin{bmatrix} -0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 2.803 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.4 \end{bmatrix} \\ &= \begin{bmatrix} 0.559 \end{bmatrix}\end{aligned}$$

then apply the activation function, in this case a extbf{softmax function}, to calculate the activation of the output nodes of the network:

$$\begin{aligned}\mathbf{a}^{[3]} &= g(\mathbf{z}^{[3]}) \\ &= g\left(\begin{bmatrix} 0.559 \end{bmatrix}\right) \\ &= \begin{bmatrix} 0.636 \end{bmatrix}\end{aligned}$$

		(ii)	If the target feature value for the current input vector is 1.0, calculate the **loss** associated with this training instance using **cross entropy loss**.
			[2 marks]
			Calculate cross entropy loss Loss = -(1*log(0.636) + 0*log(0.364)) = 0.45
	(b)	The **adam** approach to optimisation during gradient descent has now become the de-facto standard for training deep learning models. Explain how the adam approach improves upon basic gradient descent.	
			[8 marks]
		Sample Answer (Diagrams such as those included in this answer would be useful.) The gradient descent algorithm optimizes network weight values during a journey across an error (or loss) surface.  Adam improves this process by taking a more direct route across the error surface. This is achieved by adding different types of momentum terms. Adam mixes the gradient descent with momentum and RMSprop approaches in a weighted sum: $$v_{d\mathbf{W}} = \beta_1 v_{d\mathbf{W}} + (1-\beta_1)\, d\mathbf{W}$$ Gradient Descent with Momentum $$v_{d\mathbf{b}} = \beta_1 v_{d\mathbf{b}} + (1-\beta_1)\, d\mathbf{b}$$ $$s_{d\mathbf{W}} = \beta_2\, s_{d\mathbf{W}} + (1-\beta_2)\, d\mathbf{W}^{*2}$$ RMSprop $$s_{d\mathbf{b}} = \beta_2 s_{d\mathbf{b}} + (1-\beta_2)\, d\mathbf{b}^{*2}$$ $$\mathbf{W} = \mathbf{W} - \alpha \frac{v_{d\mathbf{W}}}{\sqrt{s_{d\mathbf{W}}} + \epsilon}$$ $$\mathbf{b} = \mathbf{b} - \alpha \frac{v_{d\mathbf{b}}}{\sqrt{s_{d\mathbf{b}}} + \epsilon}$$ This gives a very efficient route across an error surface.	



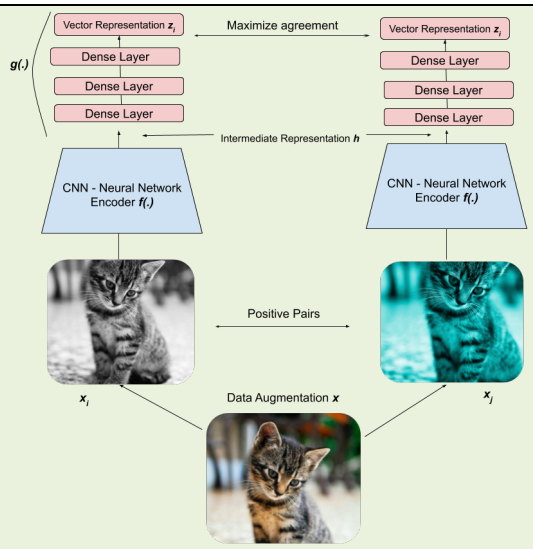
(c) **Transfer learning** takes advantage of **embeddings** learned in one situation in other situations and can allow us bring the power of deep learning to scenarios where only small amounts of data are available. For transfer learning to be effective embeddings must be **disentangled** from the tasks used to learn them. Describe an approach that can be used to learn disentangled embeddings of images.

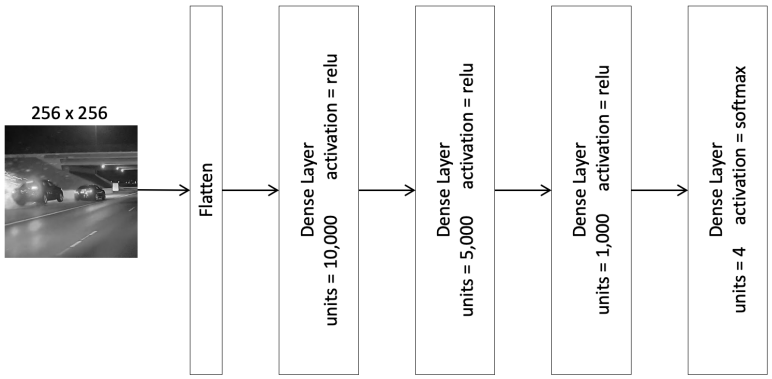
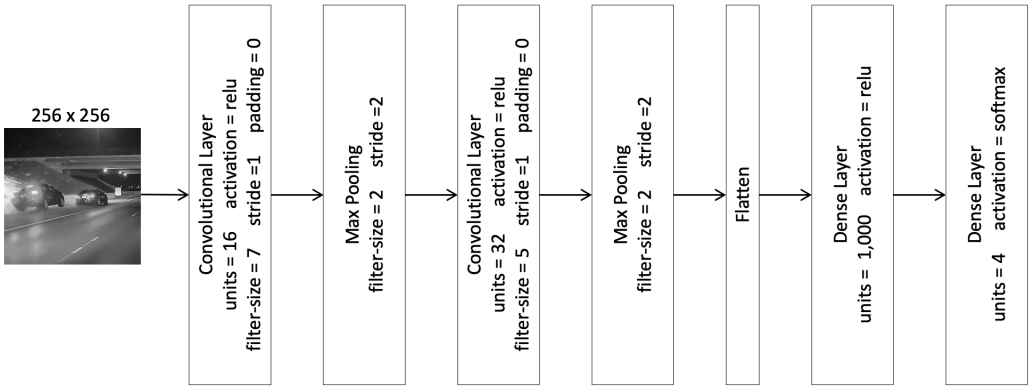
[8 marks]

A good description of any appropriate approach would be accepted, but the most obvious one to describe (based on course material) would be contrastive learning, e.g. SimCLR. To score highly students should give a high level, but informative, description of the SimCLR algorithm (below) possibly accompanied by appropriate images.

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{x_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{x}_{2k-1} = t(x_k)$
    $h_{2k-1} = f(\tilde{x}_{2k-1})$ # representation
    $z_{2k-1} = g(h_{2k-1})$ # projection
    # the second augmentation
    $\tilde{x}_{2k} = t'(x_k)$
    $h_{2k} = f(\tilde{x}_{2k})$ # representation
    $z_{2k} = g(h_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = z_i^\top z_j / (\|z_i\| \|z_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$


Vector Representation z_i
Maximize agreement
Vector Representation z_j
g(.)
Dense Layer
Dense Layer
Dense Layer
Dense Layer
Dense Layer
Dense Layer
Intermediate Representation h
CNN - Neural Network Encoder f(.)
CNN - Neural Network Encoder f(.)
Positive Pairs
x_i
Data Augmentation x
x_j

2.	**(a)**	You have been tasked with training a neural network to control a self-driving racing car from image input. The model should output one of four control signals - *left, right, brake, or accelerate* - from each input image frame. The only input to the model is a 256 pixel by 256 pixel greyscale image from the front of the car. Image (a) shows the architecture of a multi-layer perceptron neural network designed for this problem. Image (b) shows the architecture of a convolutional neural network designed for this problem. Both architectures are composed of four layers.  (a) Multi-layer perceptron network architecture  (b) Convolutional neural network architecture Calculate the number of parameters (weights and biases) that need to be learned for each network architecture.
		[12 marks]
		<u>Sample Answer</u> **Multi-layer perceptron:** This is pretty straight-forward as it involves multiplying through the sizes of the network layers. Input: 256 * 256 = 65,536 Layer 1: 65,536 * 10,000 + 10,000 = **655,370,000** Layer 2: 10,000 * 5,000 + 5,000 = **50,005,000**

Layer 3: 5,000 * 1,000 + 1,000 = **5,001,000**

Layer 4: 1,000 * 4 + 4 = **4,004**

Total parameters: **710,380,004**

Convolutional neural network:

Students need determine the number of weights based on the size of each filter and the size of each layer. To calculate the number of activations at the flattening layer they also need to keep track of the number of activations flowing through the network.

Layer 1 Dim: 256 x 256 x 1

Layer 1: 7 * 7 * 1 * 16 + 16 = **800**

Layer 1 Output Dim: 250 x 250 x 16

Layer 1 Output Dim After Pooling: 125 x 125 x 16

Layer 2: 5 * 5 * 16 * 32 + 32 = **12,832**
Layer 2 Output Dim: 121 x 121 x 32

Layer 2 Output Dim After Pooling: 60 x 60 x 32

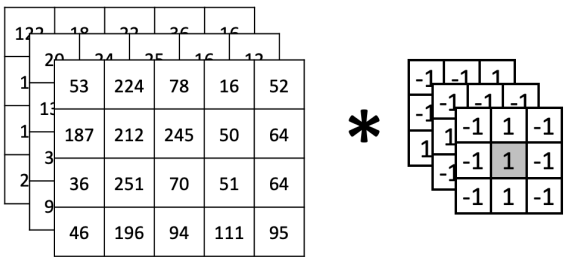
Layer 2 Dimensionality after flattening: 60 x 60 x 32= **115,200**

Layer 3: 115,200 * 1,000 + 1,000 = **115,201,000**

Layer 4: 1,000 * 4 + 4 = **4,004**

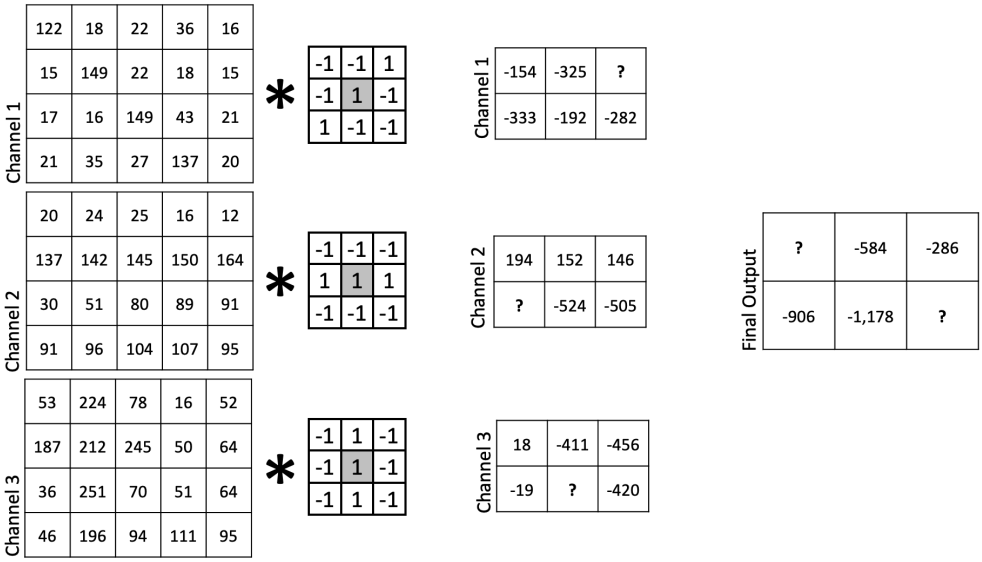
Total parameters: **15,218,636**

(b) The image below shows a 3-channel input that is being convolved (cross correlated) with a 3-channel 3 x 3 kernel.



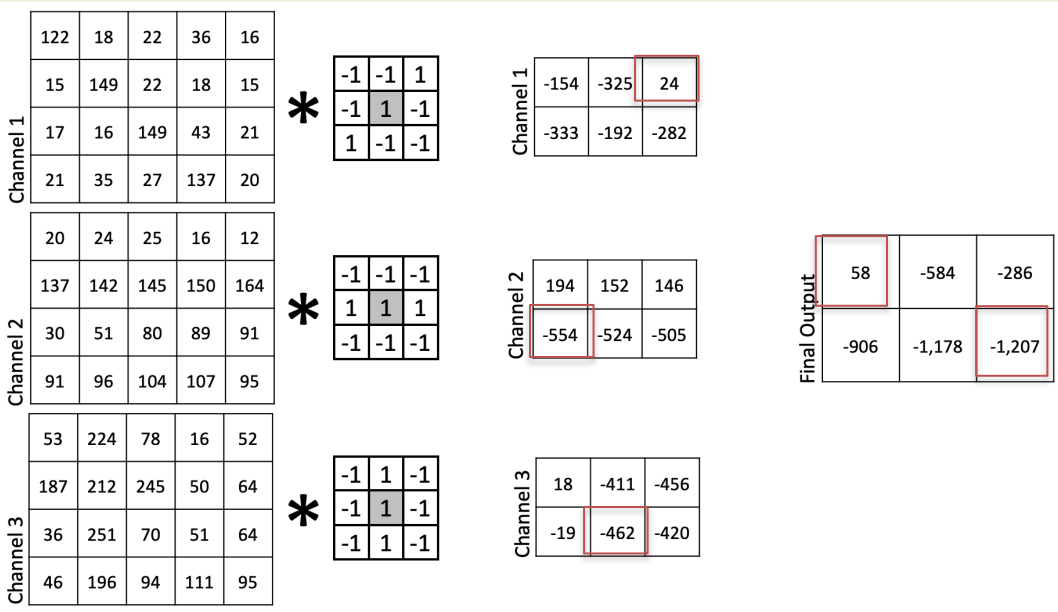
The image below expands the three-channel input and three-channel kernel so that all values can be seen and shows the intermediate convolution result for each channel as well as the final output. Calculate

the values marked with a **?** in the intermediate convolution results and the final output.



[8 marks]

Simple convolutions are calculated for each channel and then summed for the final output.



(c) The 2017 paper "*Attention is all you need*" by Vaswani et al is now one of the most cited papers in machine learning research. Explain what **attention** is and how the **transformer** architecture utilises it.

[10 marks]

Sample Answer

		Attention is a mechanism used in deep learning that allows a model to selectively focus on certain parts of the input during processing. It helps the model to identify the most important features and context of the input data while ignoring irrelevant information. The Transformer architecture is a type of neural network architecture that is specifically designed to handle sequence-to-sequence tasks such as language translation, text summarization, and speech recognition. It replaces the recurrent and convolutional layers commonly used in these tasks with a self-attention mechanism. The self-attention mechanism in the Transformer architecture allows the model to weigh the importance of different parts of the input sequence when making predictions. The self-attention layer computes a weighted sum of the input sequence, where the weights are based on the similarity between each input element and every other element in the sequence. The Transformer architecture utilizes multi-head attention, where the self-attention mechanism is applied multiple times in parallel, each with its own set of weights. This allows the model to capture different relationships between different parts of the input sequence and is particularly effective for capturing long-range dependencies.

3.	**(a)**	Describe the concept of **discounted return** that is frequently used in reinforcement learning.
		[4 marks]
		Sample Answer Calculating the expected return from a sequence of actions is key in developing reinforcement learning systems. In a basic formulation of expected return that simply sums rewards, e.g. $$G = r_t + r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_e$$ expected future rewards are considered to be as valuable as the immediate reward that the agent will receive from taking the next immediate action. Just like we might be more excited about receiving a gift of $100 today than a promise to receive a gift of $100 in a year's time, it is reasonable when calculating expected return to pay more attention to the immediate reward we expect to receive from taking the next action, than to the rewards that we expect to receive in 10 or even 100 action's time. This is known as discounted return. We can define discounted return as: $$G_\gamma = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \ldots + \gamma^{e-t} r_e$$ where γ is a discount factor that can take a value between 0 and 1.
	(b)	An intelligent agent trained to play a video game completes an episode and receives the following sequence of rewards over six timesteps: $\{r_0 = 77, r_1 = 105, r_2 = -57, r_3 = -86, r_4 = -112\}$ Compare the **discounted returns** calculated at time $t = 0$ based on this reward sequence when discounting factors of 0.9 and 0.1 are used.
		[6 marks]
		Sample Answer Students should begin by calculating the discounted returns using: $$G_\gamma = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \ldots + \gamma^{e-t} r_e$$ The following tables shows the calculations for discounting factor of 0.9 and 0.1:

Enter e		0	1	2	3	4	
Reward		77	105	-57	-86	-112	
		1	0.9	0.81	0.729	0.6561	
Discounted Return	0.9	77	94.5	-46.17	-62.694	-73.4832	**-10.8472**
		1	0.1	0.01	0.001	0.0001	
Discounted Return	0.1	77	10.5	-0.57	-0.086	-0.0112	**86.8328**

Students should then discuss that with the lower discount factor much less attention is paid to later rewards and so the overall return is much lower.

(c) To try to better understand the behaviour of their chef, a restaurant manager monitored the chef's activities over a period of time, recording the chef's activity at 1 minute intervals. The activity stream looked like this (with time flowing down through the columns):

0	PREPPING	12	PREPPING	24	PREPPING	36	COOKING
1	COOKING	13	PREPPING	25	PLATING	37	COOKING
2	COOKING	14	PREPPING	26	COOKING	38	PLATING
3	PREPPING	15	COOKING	27	PREPPING	39	PLATING
4	PREPPING	16	COOKING	28	PREPPING	40	PLATING
5	COOKING	17	PREPPING	29	PREPPING	41	PLATING
6	PLATING	18	PREPPING	30	PLATING	42	PLATING
7	PLATING	19	PLATING	31	PLATING	43	PREPPING
8	PREPPING	20	PREPPING	32	PLATING	44	PREPPING
9	PREPPING	21	PLATING	33	PLATING	45	PREPPING
10	PREPPING	22	PLATING	34	COOKING	46	PREPPING
11	PREPPING	23	COOKING	35	COOKING	47	PLATING

The restaurant noticed that the chef could occupy one of three states - PREPPING, COOKING, or PLATING - and moved quite freely between them.

(i) Based on the sequence of states given above calculate a transition matrix that gives the probability of moving between each of the three states.

[10 marks]

Sample Answer

The first step to building the transition matrix is to count the frequency of each possible state transition. Working down through the list of states we can count the number of times we move from one state to the next. This gives a transition frequency table:

	PREPPING	COOKING	PLATING
PREPPING	13	3	5
COOKING	4	5	2
PLATING	3	3	9

By normalising each row in the table (dividing each value by the sum of values in the row) we can calculate the final transition matrix:

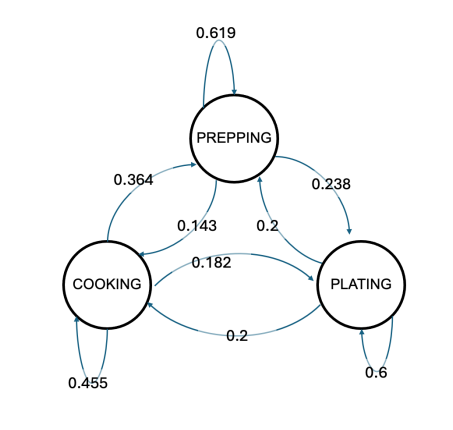
	PREPPING	COOKING	PLATING
PREPPING	0.619	0.143	0.238
COOKING	0.364	0.455	0.182
PLATING	0.2	0.2	0.6

(ii) Draw a Markov process diagram to capture the behaviour of a chef as described above.

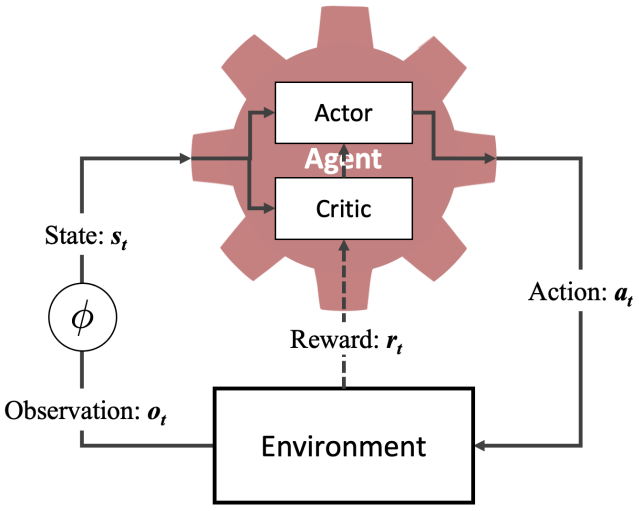
[5 marks]

Sample Answer

A diagram like this one is appropriate for this answer.



(d) The image below shows an illustration of reinforcement learning using **actor-critic** method.

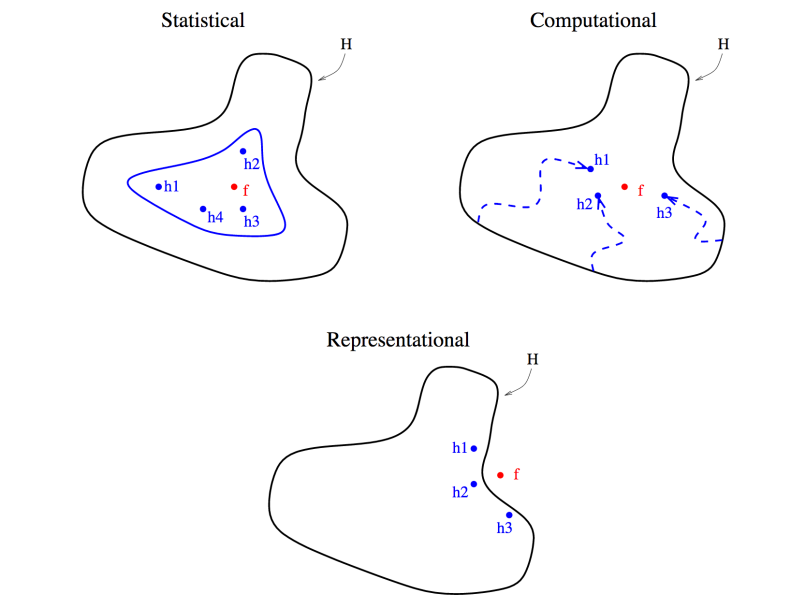


Describe the role of the actor and critic models in this approach.

[5 marks]

Attempt to merge best of policy gradient methods with value function methods.Composed of two models

		– **Critic** is an action value function model $Q_{\mathbb{M}_{\mathbf{w}_C}}(s_t, a_t)$ – **Actor** is a policy model $\pi_{\mathbb{M}_{\mathbf{w}_A}}(s_t)$ During learning when the actor model is being updated it uses outputs from the critic action value function to estimate expected return

4.	**(a)**	Describe *three* different motivations for using **ensemble methods** in machine learning.
		[10 marks]
		Sample Answer Ideally students will refer to the three motivations described by Dietrich: statistical, computational, representational. Reproduction of the diagram below from (Deittrich, 2000) would be useful. The statistical motivation is arises from the fact that we always have a sample of the full data space associated with a machine learning problem and so the likelihood of arriving at a hypothesis matching the true function we are trying to model is low. In fact in all likelihood we will arrive at multiple hypotheses that are all equally accurate in relation to the training dataset sample that we have available. By averaging the outputs of this set of models we are likely to arrive at an overall model that is more close to the true underlying function.  The computational motivation arises from the fact that most machine learning algorithms perform some form of local search through a hypothesis space and can stop at a local minimum rather than the global minimum. By averaging across many

		runs of this local search process (even if the individual runs result in local minima) we are likely to arrive a much better overall model. The representational motivation arises from the fact that in many cases it is not possible to actually represent the true underlying function that we are trying to model using a particular modelling algorithm. In the diagram above we show that the true function, f, lies outside the hypothesis space. However, it is possible that the aggregate of an ensemble of models that can be represented will be closer to this true underlying model than any single model that can be represented - an ensemble allows us to jump outside of what can be represented. Any other reasonable answer is also acceptable.
	(b)	When developing a machine learning model that will be deployed to perform a task for a user, we can describe three different goals of evaluation: 1. to determine which model is the most suitable for a task 2. to estimate how the model will perform after deployment 3. to convince users that the model will meet their needs Describe the differences between these goals, and how the evaluation methods used to achieve each of them can be different.
		[10 marks]
		Sample Answer *Students should outline that when performing an evaluation to prepare a model for deployment the first goal is to determine which approach might work best. Decision s to be made based on this evaluation include which modelling approach will be used, which data pre-processing techniques might be used, and the optimal algorithm hyper-parameters to be used. This evaluation is very much driven by the machine learning practitioner. Typically, k-fold cross validation can be used as the main evaluation method for this kind of evaluation.* *Once all of the decisions about what modelling approach to take have been made the next goal attempts to evaluate the likely performance of a model after deployment. This is largely concerned with estimating generalisation error of the model. The only sensible way to evaluate this is to use a hold out test set that has been involved at all in training the model. This is the only reasonable way to evaluate generalisation error.* *Once practitioners are convinced that a modelling approach will achieve acceptable performance after deployment the last step is to convince the people for whom the model is being built that it will meet their needs. This can be done with the same kinds of experiments used to evaluate likely generalisation error, but often different performance measures need to be used so as to speak to a different audience (not machine learning*

		practitioners). It is also important to consider a deployment experiment at this stage too.
	(c)	The **EU AI Act** will probably come into effect in 2025. The act is expected to enforce different obligations for providers of artificial intelligence solutions with different level of risk. Four levels of risk, as illustrated below, are expected.  Discuss these levels of risk, giving examples of solutions likely to be considered at each, and the obligations that are likely to be enforced.
		[10 marks]
		Sample Answer This is an open, discursive question, but something similar to the following will score highly. The proposed rules will: • address risks specifically created by AI applications; • prohibit AI practices that pose unacceptable risks; • determine a list of high-risk applications; • set clear requirements for AI systems for high-risk applications; • define specific obligations deployers and providers of high-risk AI applications; • require a conformity assessment before a given AI system is put into service or placed on the market; • put enforcement in place after a given AI system is placed into the market; • establish a governance structure at European and national level. A risk-based approach The Regulatory Framework defines 4 levels of risk for AI systems:

All AI systems considered a clear threat to the safety, livelihoods and rights of people will be banned, from social scoring by governments to toys using voice assistance that encourages dangerous behaviour.

High risk

AI systems identified as high-risk include AI technology used in:

- critical infrastructures (e.g. transport), that could put the life and health of citizens at risk;
- educational or vocational training, that may determine the access to education and professional course of someone's life (e.g. scoring of exams);
- safety components of products (e.g. AI application in robot-assisted surgery);
- employment, management of workers and access to self-employment (e.g. CV-sorting software for recruitment procedures);
- essential private and public services (e.g. credit scoring denying citizens opportunity to obtain a loan);
- law enforcement that may interfere with people's fundamental rights (e.g. evaluation of the reliability of evidence);
- migration, asylum and border control management (e.g. automated examination of visa applications);
- administration of justice and democratic processes (e.g. AI solutions to search for court rulings).

High-risk AI systems will be subject to strict obligations before they can be put on the market:

- adequate risk assessment and mitigation systems;
- high quality of the datasets feeding the system to minimise risks and discriminatory outcomes;
- logging of activity to ensure traceability of results;
- detailed documentation providing all information necessary on the system and its purpose for authorities to assess its compliance;
- clear and adequate information to the deployer;
- appropriate human oversight measures to minimise risk;
- high level of robustness, security and accuracy.

All remote biometric identification systems are considered high-risk and subject to strict requirements. The use of remote biometric identification in publicly accessible spaces for law enforcement purposes is, in principle, prohibited.

Narrow exceptions are strictly defined and regulated, such as when necessary to search for a missing child, to prevent a specific and imminent

		terrorist threat or to detect, locate, identify or prosecute a perpetrator or suspect of a serious criminal offence. Those usages is subject to authorisation by a judicial or other independent body and to appropriate limits in time, geographic reach and the data bases searched. **Limited risk** Limited risk refers to the risks associated with lack of transparency in AI usage. The AI Act introduces specific transparency obligations to ensure that humans are informed when necessary, fostering trust. For instance, when using AI systems such as chatbots, humans should be made aware that they are interacting with a machine so they can take an informed decision to continue or step back. Providers will also have to ensure that AI-generated content is identifiable. Besides, AI-generated text published with the purpose to inform the public on matters of public interest must be labelled as artificially generated. This also applies to audio and video content constituting deep fakes. **Minimal or no risk** The AI Act allows the free use of minimal-risk AI. This includes applications such as AI-enabled video games or spam filters. The vast majority of AI systems currently used in the EU fall into this category.

oOo