USN | | | | | | | | | |

Internal Assessment Test 1 Answer scheme & Solutions – October 2024

Sub:	DEEP LEARNING					Sub Code:	21CS743	Branch:	AInDS
Date:	16/10/2024	Duration:	90 minutes	Max Marks:	50	Sem	VII		OBE

		Answer any FIVE Questions	MARKS	CO	RBT
1		**Explain the different parts of an AI system relate to each other within different AI disciplines.** A computer can reason about statements in these formal languages automatically using logical inference rules. This is known as the knowledge base. **(2 Marks)** AI systems need the ability to acquire their own knowledge, by extracting patterns from raw data. This capability is known as machine learning.**(2 Marks**) A representation learning algorithm can discover a good set of features for a simple task in minutes.**(2 Marks)** Deep learning allows the computer to build complex concepts out of simpler concepts.**(2 Marks)** Diagram **(2 Marks)**	[10]	1	L3
2		**Explain in Detail about Probabilistic Supervised Learning and non-probabilistic supervised learning**. **Probabilistic Supervised Learning (5 Marks)** To find the best parameter vector θ for a parametric family of distributions p(y | x; θ). We have already seen that linear regression corresponds to the family p(y | x; θ) = N (y; θ=xT, I) A distribution over a binary variable is slightly more complicated, because its mean must always be between 0 and 1. One way to solve this problem is to use the logistic sigmoid function to squash the output of the linear function into the interval (0, 1) and interpret that value as a probability: p(y = 1 | x; θ) = σ(θTx) This approach is known as logistic regression **non-probabilistic supervised learning**. **(5 Marks)** k-nearest neighbors - there is not even really a training stage or learning process. Instead, at test time, when we want to produce an output y for a new test input x, we find the k-nearest neighbors to x in the training data X. We then return the average of the corresponding y values in the training set decision tree - breaks the input space into regions and has separate parameters for each region is the decision tree	[10]	2	L3
3	a	**List The Historical Trends in Deep Learning.** 1.Deep learning has had a long and rich history, but has gone by many names reflecting different philosophical viewpoints, and has waxed and waned in popularity. 2.Deep learning has become more useful as the amount of available training data has increased. 3.Deep learning models have grown in size over time as computer infrastructure(both hardware and software) for deep learning has improved. 4.Deep learning has solved increasingly complicated applications with increasingaccuracy over time. **(4 *1 = 4)**	[4]	1	L2
	b	**Explain the Key Trends i) Increasing Dataset Sizes ii) Increasing Model Sizes over time in Deep Learning** i) Deep learning algorithm will generally achieve acceptable performance with around 5,000 labeled examples per category **(2 marks)** Diagram **(1 Mark)** ii) Since the introduction of hidden units, artificial neural networks have doubled in size roughly every 2.4 years.**(2 marks)** Diagram **(1 Mark)**	[6]		

<table>
<tr>
<td>4</td>
<td></td>
<td>

What is Unsupervised Learning Algorithms ? Explain PCA.

iii) Unsupervised learning algorithms perform other roles, like clustering, which consists of dividing the dataset into clusters of similar examples.**(2 marks)**

iv) PCA learns a representation that has lower dimensionality than the original input. It also learns a representation whose elements have no linear correlation with each other.**(2 Marks)**

PCA finds a representation (through linear transformation) $\boldsymbol{z} = \boldsymbol{x}$
$\mathrm{Var}[\boldsymbol{z}]$ is diagonal.

In section 2.12, we saw that the principal components of a desig
are given by the eigenvectors of $\boldsymbol{X}^\top \boldsymbol{X}$. From this view,

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{W} \boldsymbol{\Lambda} \boldsymbol{W}^\top.$$

In this section, we exploit an alternative derivation of the principal comp
principal components may also be obtained via the singular value de
Specifically, they are the right singular vectors of $\boldsymbol{X}$. To see this, le
right singular vectors in the decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top$. We then
original eigenvector equation with $\boldsymbol{W}$ as the eigenvector basis:

$$\boldsymbol{X}^\top \boldsymbol{X} = \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top\right)^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{\Sigma}^2\boldsymbol{W}^\top.$$

The SVD is helpful to show that PCA results in a diagonal $\mathrm{Var}[\boldsymbol{z}$
SVD of $\boldsymbol{X}$, we can express the variance of $\boldsymbol{X}$ as:

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{x}] &= \frac{1}{m-1}\boldsymbol{X}^\top \boldsymbol{X} \\
&= \frac{1}{m-1}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top)^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top \\
&= \frac{1}{m-1}\boldsymbol{W}\boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top \\
&= \frac{1}{m-1}\boldsymbol{W}\boldsymbol{\Sigma}^2\boldsymbol{W}^\top,
\end{aligned}$$

where we use the fact that $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$ because the $\boldsymbol{U}$ matrix of the si
decomposition is defined to be orthogonal. This shows that if we tak
we can ensure that the covariance of $\boldsymbol{z}$ is diagonal as required:

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{z}] &= \frac{1}{m-1}\boldsymbol{Z}^\top \boldsymbol{Z} \\
&= \frac{1}{m-1}\boldsymbol{W}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{W} \\
&= \frac{1}{m-1}\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{\Sigma}^2\boldsymbol{W}^\top \boldsymbol{W} \\
&= \frac{1}{m-1}\boldsymbol{\Sigma}^2,
\end{aligned}$$

(6 Marks)

</td>
<td>[10]</td>
<td>2</td>
<td>L2</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>

What is Task ? List most common machine learning tasks.

Learning is our means of attaining the ability to perform the task **(1 Mark)**

Examples : Classification ,Regression,Transcription,Machine Translation etc **(4 Marks)**

</td>
<td>[5]</td>
<td>1</td>
<td>L2</td>
</tr>
</table>

	b	**Explain Deep Feedforward Networks in detail.** Deep feedforward networks, also often called feedforward neural networks, or multilayer perceptrons (MLPs), are the quintessential deep learning models. **(2 Marks**) Linear models also have the obvious defect that the model capacity is limited to linear functions, so the model cannot understand the interaction between any two input variables.To extend linear models to represent nonlinear functions of x, we can apply the linear model not to x itself but to a transformed input φ(x), then how to choose the mapping φ.**(3 Marks)**	[5]	2	L2
6	a	**What is Gradient-Based Learning ?Explain about Cost Function.** For feedforward neural networks, it is important to initialize all weights to small random values. The biases may be initialized to zero or to small positive values. The iterative gradient-based optimization algorithms used to train feedforward networks and almost all other deep models.train models such as linear regression and support vector machines with gradient descent too, and in fact this is common when the training set is extremely large.**(3 Marks)** In most cases, our parametric model defines a distribution p(y | x;θ) and we simply use the principle of maximum likelihood. This means we use the cross-entropy between the training data and the model's predictions as the cost function, where rather than predicting a complete probability distribution over y, we merely predict some statistic of y conditioned on x. Specialized loss functions allow us to train a predictor of these estimates. The total cost function used to train a neural network will often combine one of the primary cost functions described here with a regularization term. **(2 Marks)**	[5]	2	L3
	b	**Explain Softmax Units for Multinoulli Output Distributions.** linear layer predicts unnormalized log probabilities: z = WT h + b where zi = log P˜(y = i | x). The softmax function can then exponentiate and normalize z to obtain the desired yˆ. Formally, the softmax function is given by softmax(z)i= exp(zi) / ∑jexp(zj). to maximize log P (y = i; z) = log softmax(z)i. Defining the softmax in terms of exp is natural because the log in the log-likelihood can undo the exp of the softmax: logsoftmax(z)i=zi−log(j∑exp(zj)) To see that the softmax function responds to the difference between its inputs, observe that the softmax output is invariant to adding the same scalar to all of its inputs: softmax(z) = softmax(z + c). Using this property, we can derive a numerically stable variant of the softmax: softmax(z) = softmax(z − imaxzi) (5 Marks)	[5]	2	L3