UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE : **COMPM090**

ASSESSMENT PATTERN : **COMPM090A**

MODULE NAME : **Applied Machine Learning (Masters Level)**

DATE : **20-May-14**

TIME : **10:00**

TIME ALLOWED : **2 Hours 30 Minutes**

TURN OVER

Applied Machine Learning, GI09, 2014

Answer all THREE questions.

Marks for each part of each question are indicated in square brackets

Calculators are NOT permitted

1. Consider the linear regression problem for training data with vector input $\mathbf{x}^n$ and scalar output y^n, $n = 1, \ldots, N$:

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} E(\mathbf{w})$$

where we define the regularised total square loss

$$E(\mathbf{w}) = \sum_{n=1}^{N} \left(y^n - \mathbf{w}^\top \mathbf{x}^n \right)^2 + \lambda \mathbf{w}^\top \mathbf{w}$$

a. Derive an explicit expression for the optimal weight vector $\mathbf{w}_{opt}$ in terms of the training data and regularisation constant λ. Give also an estimation for the computational complexity required to find $\mathbf{w}_{opt}$ using this expression.

[10 marks]

b. Describe the gradient descent procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach.

[3 marks]

c. Describe the Newton procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach. You should give an explicit update formula for the new weight vector in terms of the old weight vector so that this could be implemented directly by a computer programmer.

[5 marks]

d. Describe the conjugate gradients procedure for finding the minimum of $E(\mathbf{w})$, explaining also any potential practical advantages or disadvantages of this approach. You should give an explicit update formula for the new weight vector in terms of the old weight vector so that this could be implemented directly by a computer programmer.

[10 marks]

e. Consider the case that each training vector $\mathbf{x}$ is sparse, with only $0 \leq s \leq 1$ of the elements of the vector being non-zero. Give estimates for the computational complexity of finding $\mathbf{w}_{opt}$ based on the above procedures, namely (batch) gradient descent, Newton's method and conjugate gradients.

[7 marks]

f. Explain what is meant by an Auto-Regressive (AR) model for a time-series $y_1, \ldots, y_T$ and derive an explicit expression for the optimal AR coefficients in the least squares sense.

[7 marks]

[Total 42 marks]

2. Consider a set of training data with vector input $\mathbf{x}^n$ and vector output $\mathbf{y}^n$. For input vector $\mathbf{x}$, a neural network produces output vector

$$\mathbf{f}(\mathbf{x}|\mathcal{W}) = \sigma^L(\mathbf{W}^L\mathbf{h}^{L-1}),$$

where the hidden layer values are recursively defined by

$$\mathbf{h}^l = \sigma^l\left(\mathbf{W}^l\mathbf{h}^{l-1}\right), \quad l = 2,\ldots,L-1, \qquad \mathbf{h}^1 = \sigma^1\left(\mathbf{W}^1\mathbf{x}\right)$$

The total set of parameters is given by $\mathcal{W} = \left\{\mathbf{W}^1,\ldots,\mathbf{W}^L\right\}$ and the dimension of layer l is defined by d_l where $dim(\mathbf{W}^l) = d_l \times d_{l-1}$. You may assume that the transfer functions $\sigma^1,\ldots,\sigma^L$ are known.

a. For the squared loss objective function

$$E(\mathcal{W}) = \sum_{n=1}^{N}\left(\mathbf{y}^n - \mathbf{f}(\mathbf{x}^n|\mathcal{W})\right)^2$$

derive an efficient recursive procedure to compute the gradient of $E(\mathcal{W})$ with respect to all the parameters $\mathcal{W}$.

[12 marks]

b. Explain what is meant by an autoencoder neural network.

[3 marks]

c. Consider an autoencoder with a single hidden layer ($L = 2$). When the transfer function at the output layer is the identity, $\sigma^L(x) = x$, derive an expression for the optimal weight matrices $\mathbf{W}^2, \mathbf{W}^1$ and relate this to Principal Components Analysis. What would be the optimal weights for an autoencoder with a larger number of layers, $L > 2$ but with the identity transfer function on the output layer?

[10 marks]

[Total 25 marks]

3. This question concerns nearest neighbour methods.

a. i. Explain what is meant by nearest neighbour classification for a dataset of N examples, $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$, for D-dimensional inputs $\mathbf{x}$ and discrete class labels c. Explain how to classify a new input $\mathbf{x}$.

[2 marks]

ii. For the Euclidian distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2}$$

and the dataset $\mathcal{D}$ above, describe the computational complexity (both time and storage) of computing the nearest neighbour classifier for a novel input $\mathbf{x}$.

[2 marks]

iii. Explain what is meant by a metric distance and show that the Euclidean distance is a metric.

[4 marks]

b. Orchard's algorithm is a way to speed up the calculation of the nearest neighbour (for a metric distance) of a query $\mathbf{q}$ to a set of D-dimensional training vectors $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$.

i. For a metric distance show that if $d(\mathbf{q}, \mathbf{x}^i) \leq \frac{1}{2} d(\mathbf{x}^i, \mathbf{x}^j)$ then $d(\mathbf{q}, \mathbf{x}^i) \leq d(\mathbf{q}, \mathbf{x}^j)$. Draw a picture to describe this mathematical result.

[4 marks]

ii. Explain in detail how Orchard's algorithm works.

[4 marks]

iii. Give an example dataset and query for which Orchard's algorithm will not be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

[2 marks]

iv. Give an example dataset and query for which Orchard's algorithm will be faster than simply explicitly evaluating the nearest neighbour by computing all distances from the query point $\mathbf{q}$.

[2 marks]

c. Consider a metric distance and a set of datapoints $\mathbf{x}^i$, $i \in I$ for which $d(\mathbf{q}, \mathbf{x}^i)$ has already been computed.

i. Show that we may form the lower bound

$$d(\mathbf{q}, \mathbf{x}^j) \geq \max_{i \in I} \left\{ d(\mathbf{q}, \mathbf{x}^i) - d(\mathbf{x}^i, \mathbf{x}^j) \right\} \equiv L_j$$

[2 marks]

ii. Explain how the Approximating and Elimination Search Algorithm (AESA) uses the above result to attempt to speed up the computation of the nearest neighbour of a query point $\mathbf{q}$.

[2 marks]

d. The KD tree is a hierarchical data-structure that can be used to potentially speed up nearest neighbour search.

i. Explain how to form a KD tree, giving an example of a dataset (of two-dimensional data) and the corresponding KD tree.

[3 marks]

ii. Consider a query vector $\mathbf{q}$. Let's imagine that we have partitioned the datapoints into those with first dimension x_1 less than a defined value v (to its 'left'), and those with a value greater or equal to v (to its 'right'):

$$\mathcal{L} = \{\mathbf{x}^n : x_1^n < v\}, \qquad \mathcal{R} = \{\mathbf{x}^n : x_1^n \geq v\}$$

Let's also say that our current best nearest neighbour candidate is $\mathbf{x}^i$ and that this point has squared Euclidean distance $\delta^2 = (\mathbf{q} - \mathbf{x}^i)^2$ from $\mathbf{q}$. Show that if $q_1 \geq v$ and $(v - q_1)^2 > \delta^2$, then $(\mathbf{x} - \mathbf{q})^2 > \delta^2$.

[3 marks]

iii. Explain how the above result can be used with the KD tree to search for the nearest neighbour of a query. Use your example KD tree above and an example query point to describe how to find the nearest neighbour using the KD tree.

[3 marks]

[Total 33 marks]

END OF PAPER

UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE : **COMPGI09**

ASSESSMENT PATTERN : **COMPGI09D**

MODULE NAME : **Applied Machine Learning**

DATE : **28-May-15**

TIME : **10:00**

TIME ALLOWED : **2 Hours 30 Minutes**

TURN OVER

Applied Machine Learning, GI09, 2015

Answer all THREE questions.

Marks for each part of each question are indicated in square brackets

Calculators are NOT permitted

1. a. Explain what is meant by Forward Automatic Differentiation (AutoDiff) and give two procedures (one exact and the other an approximation) that compute the gradient of a subroutine $\mathbf{x}$ with respect to its arguments $\mathbf{x}$, giving time complexities of the approaches.

[5 marks]

b. Explain what is meant by Reverse AutoDiff. Describe the algorithm and its time complexity. Give an example to illustrate how the algorithm works.

[7 marks]

c. Consider a set of training data with vector input $\mathbf{x}^n$ and vector output $\mathbf{y}^n$. For input vector $\mathbf{x}$, a neural network produces output vector

$$\mathbf{f}(\mathbf{x}|\mathcal{W}) = \sigma^L(\mathbf{W}^L\mathbf{h}^{L-1}),$$

where the hidden layer values are recursively defined by

$$\mathbf{h}^l = \sigma^l\left(\mathbf{W}^l\mathbf{h}^{l-1}\right), \quad l = 2,\ldots,L-1, \qquad \mathbf{h}^1 = \sigma^1\left(\mathbf{W}^1\mathbf{x}\right)$$

The total set of parameters is given by $\mathcal{W} = \left\{\mathbf{W}^1,\ldots,\mathbf{W}^L\right\}$ and the dimension of layer l is defined by d_l where $dim(\mathbf{W}^l) = d_l \times d_{l-1}$. You may assume that the transfer functions $\sigma^1,\ldots,\sigma^L$ are known.

i. For the squared loss objective function

$$E(\mathcal{W}) = \sum_{n=1}^{N}\left(\mathbf{y}^n - \mathbf{f}(\mathbf{x}^n|\mathcal{W})\right)^2$$

derive an efficient recursive procedure to compute the gradient of $E(\mathcal{W})$ with respect to all the parameters $\mathcal{W}$ and relate this to reverse mode AutoDiff.

[5 marks]

ii. Explain how to compute the gradient of $E(\mathcal{W})$ when parameters of the network are tied.

[3 marks]

[Total 20 marks]

2. a. Explain why the gradient of a function points along the direction of maximal change.

[2 marks]

b. Prove that for a convex function $f(\mathbf{x})$ which has a Hessian with eigenvalues less than L, then under the gradient descent procedure

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{1}{2\varepsilon T}(\mathbf{x}_1 - \mathbf{x}^*)^2$$

where $\mathbf{x}_1$ is the starting point, $\mathbf{x}_T$ is the value after $T-1$ gradient descent steps and the learning rate $\varepsilon \leq 1/L$.

[20 marks]

c. Explain what is meant by Nesterov's Accelerated Gradient procedure and compare its theoretical convergence rate with that of standard gradient descent for a convex function.

[4 marks]

d. Explain what is meant by Newton's method for optimisation of a function $f(\mathbf{x})$ and show that the position $\mathbf{x}_T$ obtained after T updates of the algorithm is invariant with respect to a linear coordinate transformation $\mathbf{x} = \mathbf{M}\mathbf{y}$.

[10 marks]

e. Explain what is meant by the Gauss-Newton optimisation method and what advantages this has over the standard Newton method.

[5 marks]

[Total 41 marks]

3. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1, \ldots, N$, define the matrix

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

The covariance matrix of the data, $\mathbf{S}$, has elements

$$S_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

a. Explain why PCA is often used as a pre-processing step in machine learning and explain what the geometric meaning of PCA is.

[4 marks]

b. Explain how to write $\mathbf{S}$ in terms of matrix multiplication of $\mathbf{X}$.

[3 marks]

c. PCA is typically described in terms of the eigen-decomposition of $\mathbf{S}$. Explain how this procedure works and also discuss the computational complexity of performing PCA based on directly computing the eigen-decomposition of $\mathbf{S}$.

[4 marks]

d. Consider the situation in which the datapoints $\mathbf{x}^n$ are very sparse – that is, only a few elements of each vector $\mathbf{x}^n$ are non-zero, resulting also in a sparse matrix $\mathbf{S}$. Describe a computationally efficient procedure to estimate the principal direction (the largest eigenvector of $\mathbf{S}$) and explain why this is efficient.

[4 marks]

e. Continuing with the sparse datapoint scenario, can you conceive a technique that would enable one to perform full PCA (not just the principal eigenvector) efficiently?

[4 marks]

f. An alternative way to perform PCA is based on the singular value decomposition (SVD) of the matrix $\mathbf{X}$. Explain why this is related to the eigen-decomposition of $\mathbf{S}$ and explain the computational complexity of this approach to performing PCA compared to directly computing the eigen-decomposition of $\mathbf{S}$.

[5 marks]

g. PCA can be considered as representing the i^{th} component of the n^{th} datapoint using

$$x_i^n \approx \sum_j y_j^n b_{ji}$$

where b_{ji} are the elements of the basis vectors for the j^{th} basis vector, and y_j^n is the corresponding coefficient. In the case that some of the components of x_i^n are missing, we cannot find the optimal PCA solution by the standard eigen-approach. In this case, define the least squares objective

$$E(\mathbf{B}, \mathbf{Y}) = \sum_{n=1}^{N} \sum_{i=1}^{D} \gamma_i^n \left[x_i^n - \sum_j y_j^n b_{ji} \right]^2$$

where

$$\gamma_i^n = \begin{cases} 1 & \text{if } x_i^n \text{ exists} \\ 0 & \text{if } x_i^n \text{ is missing} \end{cases}$$

Derive a procedure for minimising $E(\mathbf{B}, \mathbf{Y})$ that is guaranteed to decrease the objective function at each stage of the iteration, and for which each iteration corresponds to the solution of a set of linear equations.

[9 marks]

h. Explain the method of Fisher's Linear Discriminants and derive an explicit formula for the projection vector.

[6 marks]

[Total 39 marks]

END OF PAPER

UNIVERSITY COLLEGE LONDON

EXAMINATION FOR INTERNAL STUDENTS

MODULE CODE	:	**COMPM090**
ASSESSMENT PATTERN	:	**COMPM090B**
MODULE NAME	:	**Applied Machine Learning (Masters Level)**
DATE	:	**19 May 2016**
TIME	:	**2:30 pm**
TIME ALLOWED	:	**2 hours 30 mins**

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

2015/16

TURN OVER

Applied Machine Learning, GI09, 2015-2016

Answer all THREE questions.

Marks for each part of each question are indicated in square brackets

Calculators are NOT permitted

1. a. Describe Forward Automatic Differentiation (AutoDiff) and give two procedures (one exact and the other an approximation) that compute the gradient of a subroutine $f(\mathbf{x})$ with respect to its arguments $\mathbf{x}$, giving time complexities of the approaches.

[5 marks]

b. Describe Reverse AutoDiff and explain its time complexity.

[7 marks]

c. Explain how to use Reverse AutoDiff to efficiently calculate the gradient with respect to θ_1, θ_2 of

$$\sum_{n=1}^{N} (y^n - sin(\theta_1 + \theta_2 x^n))^2$$

where (x^n, y^n) are the input-output values for the n^{th} datapoint. Your computation graph should have nodes representing elementary functions. Annotate your graph suitably and define the forward and backward passes explicitly.

[8 marks]

d. Consider a time series prediction problem in which, given a sequence of inputs $x_1, x_2, \ldots, x_t$, we make a prediction $\tilde{y}_t$ for the output at time t. To do this we define:

$$h_1 = x_1$$
$$h_t = f(x_t, h_{t-1}, A) \qquad t > 1$$
$$\tilde{y}_t = g(h_t, B)$$

where A and B are parameters and f and g are some (unspecified) functions. The objective is to find parameters A and B that minimise the loss

$$\sum_{t=1}^{T} (y_t - \tilde{y}_t)^2$$

Explain how to use Reverse AutoDiff to efficiently calculate the gradient of this loss function with respect to A and B.

[7 marks]

e. An input-output time-series (x_t, y_t), $t = 1, \ldots, T$ can be modelled by a recurrent LSTM (Long Short Term Memory) network. Explain the essential components of an LSTM network and what difficulties it tries to overcome (compared to standard recurrent networks).

[10 marks]

[Total 37 marks]

2. Principal Components Analysis (PCA) is a method to form a lower dimensional representation of data. For datapoints $\mathbf{x}^n, n = 1,\ldots,N$, define the matrix

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right]$$

That is, for datapoints $\mathbf{x}$ with dimension D, then $\mathbf{X}$ is $D \times N$ dimensional. The data is such that the mean is zero, that is

$$\sum_{n=1}^{N} \mathbf{x}^n = \mathbf{0}$$

K-dimensional PCA aims to find a representation

$$\mathbf{x}^n \approx \sum_{k=1}^{K} y_k^n \mathbf{b}^k$$

where $\mathbf{b}^1, \ldots, \mathbf{b}^K$ are 'basis' vectors and y_k^n are the coefficients.

a. Explain how to efficiently compute the basis vectors and coefficients in order to minimise the squared loss between the approximation and each $\mathbf{x}^n$, namely

$$\sum_{n=1}^{N} \left(\mathbf{x}^n - \sum_{k=1}^{K} y_k^n \mathbf{b}^k \right)^2$$

[8 marks]

b. Explain how Autoencoders can also be used to find low dimensional representations of data and explain how PCA relates to an Autoencoder.

[5 marks]

c. Consider an Autoencoder with structure $\mathbf{x} \to \mathbf{h} \to \tilde{\mathbf{x}}$, trained to minimise the squared loss

$$\sum_{n=1}^{N} (\tilde{\mathbf{x}}^n - \mathbf{x}^n)^2$$

with $\mathbf{h}^n = f(\mathbf{A}\mathbf{x}^n)$ and $\tilde{\mathbf{x}}^n = \mathbf{B}\mathbf{h}^n$ for matrices $\mathbf{A}$, $\mathbf{B}$ and a non-linear function f.

For K-dimensional $\mathbf{h}$, is this non-linear procedure in principle more powerful than K-dimensional PCA, in the sense that it has a lower squared loss? Explain fully your answer.

[6 marks]

d. For N datapoints $\mathbf{x}^1, \ldots, \mathbf{x}^N$, explain how it is possible to obtain essentially perfect reconstructions of these datapoints using an Autoencoder with N units in the bottleneck layer.

[3 marks]

e. When training a deep Autoencoder (say more than 8 layers) explain why it is important to initialise the parameters of Autoencoders carefully. Suggest a criterion to initialise the parameters and explain the motivation behind this approach.

[5 marks]

f. PCA can be considered a form of matrix factorisation. An alternative matrix factorisation method is probabilistic latent semantic analysis (PLSA) (also called non-negative matrix factorisation). This takes a positive matrix $\mathbf{X}$ whose entries all sum to 1:

$$\sum_{ij} X_{ij} = 1, \qquad 0 \leq X_{ij} \leq 1$$

and forms an approximation based on

$$X_{ij} \approx \sum_{k=1}^{H} U_{ik} V_{kj}$$

for matrices U and V non-negative entries and $\sum_i U_{ik} = 1$ and $\sum_k V_{kj} = 1$.

i. In the lectures, we compared the application of PCA and PLSA on a set of face images. Explain what are the typical characteristics of the 'eigenfaces' compared with the 'plsa' faces.

[4 marks]

ii. Derive an algorithm to find U and V based on an interpretation of X, U and V in terms of probability distributions.

[8 marks]

[Total 39 marks]

3. a. For input-output training points $(\mathbf{x}^n, y^n)$, $n = 1, \ldots, N$, where each input $\mathbf{x}^n$ is a vector and each output y^n is a scalar, the squared loss of a linear regression model is

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left(y^n - \mathbf{w}^\top \mathbf{x}^n \right)^2$$

i. Compute the gradient and Hessian of this objective function and show that $E(\mathbf{w})$ is convex.

[4 marks]

ii. Explain what Stochastic Gradient Descent is and how it could be used to find the $\mathbf{w}$ that minimises $E(\mathbf{w})$.

[4 marks]

iii. In the case that the input vectors are sparse (only a fraction f of the elements of each $\mathbf{x}^n$ are non-zero), explain what computational savings this has when implementing gradient descent.

[4 marks]

iv. Explain how Conjugate Gradients could be used to find the $\mathbf{w}$ that minimises $E(\mathbf{w})$ and what computational savings can be made when the input vectors $\mathbf{x}^n$ are sparse.

[4 marks]

b. Consider a multi-class classification problem with input vector $\mathbf{x}^n$ and corresponding class label $c^n \in \{1, \ldots, C\}$. The softmax log likelihood objective is to maximise

$$L(\mathbf{w}_1, \ldots, \mathbf{w}_C) \equiv \sum_{n=1,\ldots,N} \log p(c^n | \mathbf{x}^n)$$

where

$$p(c^n | \mathbf{x}^n) = \frac{e^{\mathbf{w}_{c^n}^\top \mathbf{x}^n}}{\sum_{c=1}^{C} e^{\mathbf{w}_c^\top \mathbf{x}^n}}$$

i. Calculate the gradient vectors

$$\frac{\partial}{\partial \mathbf{w}_c} L$$

[4 marks]

ii. Show that $L(\mathbf{w}_1, \ldots, \mathbf{w}_C)$ is jointly concave.

[4 marks]

[Total 24 marks]

END OF PAPER