Exam Preparation

Example Exam Model Answers

The model answers provided here are intended as an example only; use relevant equations to support your explanations, be concise and to the point, organise your answer well, show a critical view where appropriate

Q1) Answer True or False (20 marks):

a) A perceptron is a single layer feedforward neural network

b) A perceptron is an auto-associative neural network

c) A recurrent neural network is a network without any feedback

d) An auto-associative network is a network that contains only one feedback

e) An auto-associative network has the same number of input and output neurons

f) Neural networks are interpretable, i.e. can explain their answers by themselves

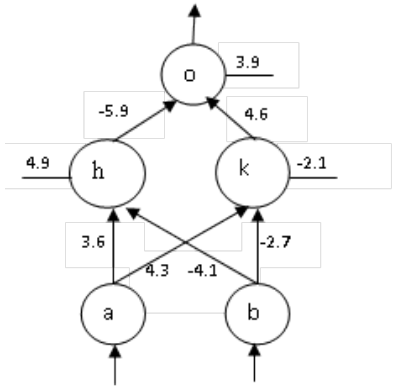
Answer:

a) True
b) False
c) False
d) False
e) True
f) False

Q2)

a) In the network below, calculate output o given (a, b) = (0.7, 0.3) and sigmoid activation function (5 marks)

b) Briefly describe how backpropagation works (10 marks)



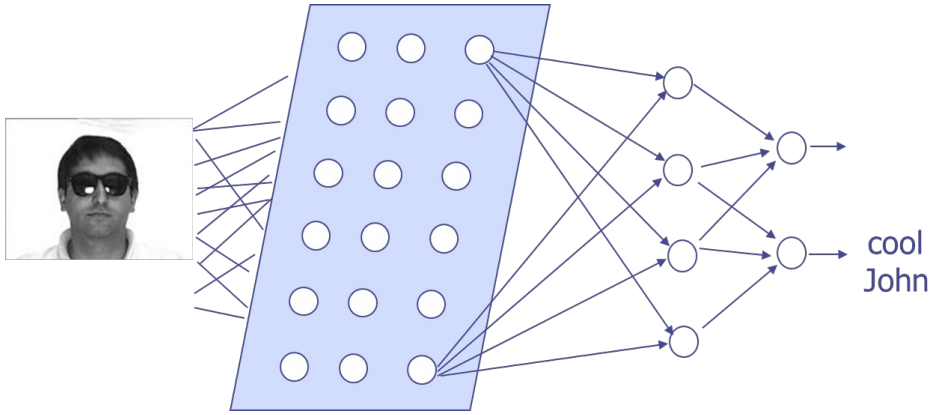
(a) Sigmoid function $\sigma(x)=1/(1+e^{-x})$; h = σ(3.6x0.7-4.1x0.3+4.9), k = σ(4.3x0.7-2.7x0.3-2.1), o = σ(-5.9h+4.6k+3.9)

(b) Parameters (weighs W) are adjusted according to $\Delta W=-\eta \nabla E_{\mathbf{w}}$ so as to try and minimize error $E=1/2 \sum_i (o_i - t_i)^2$ where o are the network's outputs and t are the desired (target) outputs. ∇E is the gradient of the error function, which can be approximated by sending local error $e_i = o_i - t_i$ back through the network calculating local derivatives, e.g. $\partial o = e . \sigma'(-5.9h+4.6k+3.9)$ in the case of the above network, where σ'(x) is the derivative of the sigmoid function. Finally, weights are adjusted according to $\Delta W_{ij}=-\eta\, \partial i\, \sigma j$, e.g. $\Delta W_{ok} = -\eta\, \partial o\, k$ (from (a)). The repeated application of weight changes ΔW can be shown to approximate in a distributed way the gradient computation efficiently.

Q3) Explain why self-organising maps (SOM) can be used for clustering. Give an example of a hybrid system combining SOM and a feedforward neural network (10 marks)

SOM can be seen as a nonlinear version of Principal Component Analysis, whereby weights are updated according to $\Delta \mathbf{W}_j = \eta \cdot h_{ji} \cdot (\mathbf{x} - \mathbf{W}_j)$, where $x - W_j$ denotes the distance between input vector x and the weight vector of neuron j. Although x and W_j may be in a high-dimensional space, the actual neurons are in a 2-dimensional space. Since learning is the process of changing the weights until the distribution of the weight vectors approaches that of the input data, such learning can be simulated in 2-d as weights change for the winning neuron (and its neighbours in the topological space). This can be visualized as neurons that move and produce clusters, e.g. a uniformly distributed data should produce a lattice of equidistant neurons.

An example of a hybrid SOM and feedforward network is given below where the SOM is used for dimensionality reduction of the images and the result of this is fed as input to a multilayer perceptron which can be trained using backpropagation to perform image classification. This set-up is very common nowadays with different types of networks, one performing unsupervised feature selection and the other (supervised) classification.



Q4) Discuss how the number of hidden neurons can influence network learning and network generalisation (10 marks)

Too many hidden neurons and the network will overfit the training data; too few and it won't learn due to lack of memory space, i.e. the training set error will be too high. When it comes to verifying generalisation, the important measure is the system's performance in the presence of previously unseen data. To estimate such a generalisation error, the network has to be evaluated on a test set, never seen during training, because in general the space of possible generalizations is infinite.

Q5) Consider the following data set where A and B are numerical values and Z is a Boolean (yes/no) classification.

i. Let P be the perceptron with weights wA = 2, wB = 1 and bias -4.5. What is the value of the standard error function for this perceptron on this data set? (10 marks)

ii. Find a set of weights and bias that classifies all the data correctly (10 marks)

A	B	Z
1	2.5	1
2	1.5	0
3	2	1
1	1	0

The equation for the perceptron is p = 2A + B - 4.5
For input (1,2.5), p = 2+2.5-4.5=0. Therefore o = 0
For input (2,1.5), p = 4+1.5-4.5=1, o = 1
For input (3,2), p = 6+2-4.5=3.5, o = 1
For input (1,1), p = 2+1-4.5=-1.5, o = 0
$E = \frac{1}{2} \Sigma_i (z_i - o_i)^2 = \frac{1}{2} ((1-0)^2+(0-1)^2+(1-1)^2+(0-0)^2) = 1$

If we plot the values of A,B in a diagram, it is easy to see that 1.5<B<2 solves the classification problem; taking e.g. B=1.8 this is to say that 0.A + 1.B - 1.8 = 0. Hence, a perceptron with wA=0, wB=1 and bias -1.8 should classify all data correctly. If you want you can check this to make sure.

Q6)

i) What is a deep network? Give an example of deep learning (5 marks)

ii) List the pseudo-code of a Boltzmann machine training algorithm. What is a Boltzmann machine useful for? (10 marks)

iii) Describe how the Contrastive Divergence learning algorithm works (10marks)

Any network with more than a single hidden layer is a deep network, including recurrent networks, but in its origin the term referred to RBMs stacked one on top of another, possibly with a classifier, e.g. an SVM or softmax layer at the top. An example of deep learning includes RBMs trained in an unsupervised, greedy way, using contrastive divergence all the way from image pixels to an image classification task at the top, with each layer performing a feature extraction task at a different level of abstraction.

Input = Boltzmann machine B, visible units v, hidden units h

% positive phase

For each data item in the training data

Repeat

Assign values to v from the training data

Calculate values of h using $p_i = 1/(1+\exp(-\Delta E_i/T))$ for each i in h, where

$$\Delta E_i = E_{i=on} - E_{i=off} = -\sum_j W_{ij} \cdot A_j + \theta_i, \; i \neq j, A_j \in \{0,1\}$$

Until B reaches a stable state

Calculate p+ = probability of neurons i and j being both activated

% negative phase

Repeat

Calculate values of v and h starting from random activation values

Until B reaches a stable state

Calculate p- = probability of neurons i and j being both activated

Change weight W between neurons i and j using

$$\Delta W_{ji} = \frac{1}{T}(p^+_{ji} - p^-_{ji})$$

Contrastive Divergence applies to RBMs with visible layer v and hidden layer h and seeks to make the network's reconstruction of v given h approach the original data distribution. Thus, activation values for h are calculated given v from the training data (step 0), then activation values for v are calculated given h (these will be different because as in Boltzmann machines the network is probabilistic), and finally new values for h are calculated (step 1). Weights are then changed according to the equation below which uses the difference between the multiplication of the activation values obtained in steps 0 and 1. By repeating the process over many steps, performing CD(n) instead of CD1 as above, one expects to minimize such difference in the long run, although no proof exists.

$$\Delta w_{ij} = \eta \left(\langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^1 \right)$$