# DD2437 Exam Answers, 23HT (2023/10/20)

**Question 1** (11p)

**A)**

*I. Problem type*: problem of failure detection (not prediction) can be conceptualised as a classification problem for sequences with the classes corresponding to normal and abnormal states (various failure states) with heavy overrepresentation of "normal" state (anomaly detection)

*II) Neural network type*: multi-layer perceptron (MLP) with inputs encoding time-delayed representations of sensor readouts (sequences) OR a recurrent neural network (RNN)

*III) Network architecture*: The number of network inputs would correspond to the number of sensors (for RNN or the number of sensors times time lags for MLP with time delayed representations) and the output layer configuration would depend on the system state encoding (a classical configuration for classification would imply the number of outputs with sigmoidal activation function equal to the number of classes but there are also other encoding strategies). The hidden layers should be configured as part of model selection.

*IV) Training algorithm (including loss function)*: backprop for MLP (or backprop through time for RNN) with early stopping, cross-entropy loss function (potentially with a regularisation term)

*V) Model selection*: Grid-search through the Cartesian product of hyperparameter spaces/domains and cross-validation based estimation of the generalisation error.

Key hyperparameters: the number of time lags, size of the hidden layer, learning rate

*VI) Performance evaluation*: false and true positive rate, sensitivity vs specificity (or AUC ROC, F-score) on unseen data (plus cross-validation estimate)

*VII) Challenges*: Depending on the amount of available data and given the potential high dimensionality of the network input to exploit the time-delayed representations, there is a risk of overfitting. Class imbalance is a serious challenge (towards anomaly detection).

**B)**

*I. Problem type*: anomaly detection, binary classification with heavy class imbalance

*II) Neural network type*: multi-layer perceptron (MLP)

*III) Network architecture*: feed-forward architecture with hidden layers, the number of inputs is determined by the number of attributes and the number of outputs corresponds to the number of states, so it is the number of error states plus one. Hidden-layer size should be determined as part of model selection.

*IV) Training algorithm (including loss function)*: backprop with early stopping and cross-entropy loss function

*V) Model selection*: As in A)

Key hyperparameters: the size of the hidden layer (number of hidden layers and their dimensionality), learning rate

*VI) Performance evaluation*: As in A)

*VII) Challenges*: Class imbalance is a serious challenge (towards anomaly detection).

**C)**

*I. Problem type*: Data augmentation, generation

*II) Neural network type*: Variational autoencoder (VAE) utilising CNN representations

*III) Network architecture*: The number of inputs and outputs the same and corresponding to the size of the available images. The hidden layers are subject to model selection.

*IV) Training algorithm (including loss function)*: backprop with reparameterization trick and potentially with regularisation terms. Mean square error as the loss function.

*V) Model selection*: As in A)

Key hyperparameters: the size of the hidden layer (number of hidden layers and their dimensionality), learning rate, the underlying probabilistic model

*VI) Performance evaluation*: Quality, realism and diversity of images (rther qualitative than quantitative criteria)

*VII) Challenges*: Creating hybrid scenarios depends on the capability of identifying desirable latent variables and the “quality” of the latent space.

**Question 2** (4p)

Assume a Hopfield network with bipolar {1, -1} nodes and the following weight matrix, **W**:

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & -2 & 3 & 4 \\ -1 & 0 & -1 & -2 & 3 \\ -2 & -1 & 0 & -1 & -2 \\ 3 & -2 & -1 & 0 & -1 \\ 4 & 3 & -2 & -1 & 0 \end{bmatrix}$$

Start with any random pattern, like [1 1 1 1 1]

$$\mathbf{xW} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & -2 & 3 & 4 \\ -1 & 0 & -1 & -2 & 3 \\ -2 & -1 & 0 & -1 & -2 \\ 3 & -2 & -1 & 0 & -1 \\ 4 & 3 & -2 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -1 & -6 & -1 & 4 \end{bmatrix}$$

After thresholding, the first iteration gives [1 -1 -1 -1 1]

The second iteration gives [1 1 -1 1 1], the third iteration gives [1 1 -1 1 1]. Since the result of the third iteration is the same as that of the second, we have evidence that the network has converged to a stable output. So, **x** = [1 1 -1 1 1] is a fixed-point attractor. An alternative candidate for a stored pattern is **x** = [-1 -1 1 -1 -1].

**Question 3** (10p)

**Task 1)** The first task can be addressed as a classification problem. There are 70% data samples with labels (10-dimensional vector of labels corresponding to 10 morphological features) and we need to classify the remaining 30%. If morphological features are numerical rather than categorical, then one could resort to the regression problem. In any case, since we deal with image data it seems suitable to rely on a convolutional neural network (CNN), especially if the number of available data samples is large. The number of inputs should correspond to the unified size of the photos and the number of sigmoidal (for classification) outputs should be 10. The number and size of the hidden layers (convolution and pooling layers) should be decided in the model selection process through cross-validation over a different combination of hyperparameters (e.g. hidden layer size). For this data could be split into 80% for cross-validation and the remaining 20% for testing (unseen data subset).

The recommended learning algorithm is backprop with Adam optimiser (adaptive learning rate and momentum), it is also recommended that regularisation techniques are exploited, e.g. early stopping. There is no need for any particular preprocessing (though whitening could be useful) beyond standard image processing techniques.

Key problems/challenges/risks are concerned with generalisation (e.g. photos may have been taken in varying conditions with cameras of varying quality etc.), potential class imbalance, relatively large class dimensionality and multimodality (the number of outputs of mixed categorical and numerical characteristics). To large extent the potentially challenging natura of the data can be handled with CNN (with crucial invariances) and image preprocessing methods. Potential class imbalance can be overcome with resampling techniques and multimodality of the output can be handled with a hybrid loss function (cross-entropy for categorical and mean square error for numerical outputs).

**Task 2)** The nature of this task is very different from Task 1, as the leaves should be grouped rather than labelled (classified). In the context of remarks made in the assignment about unknown number of categories, their rather open-ended definition (with overlapping characteristics) and a need for preserving similarity relationship between samples including new inputs, it is recommended that a self-organising map (SOM) is employed. The input could be handled by the CNN developed in Task 1 – either intermediate CNN representations could be used (*please see also a comment about the robustness of image processing made at the very end*) or morphological descriptors (recommended to try in the first place due to its semantic content and lower dimensionality that is easier to handle.

SOM grid could be two-dimensional and the learning algorithm consists in a kind of soft competitive learning with adaptively shrinking neighbourhood in the output grid space. Key hyperparameters are concerned with the grid size, learning rate and the schedule for neighbourhood adaptation.

The process of identifying a category for a new input would start with feeding a corresponding photo to the CNN network, trained in Task 1, that feeds (either CNN's intermediate representations or its output morphological descriptors) into a SOM. Due to its topology preserving capacity the SOM maps then the input to one of the nodes in the output grid (causing the activation of the corresponding best matching unit).The location of the best matching unit in relation to other nodes in the output SOM grid offers insights into the similarity of the new input sample to the other samples representing different groups/categories. Since there are no clear boundaries between these categories any given sample can be considered to share similarities with representatives of different groups (overlapping).

As for the requested robustness, the challenging nature of varying colour, position or photo background could be addressed by using representations extracted with CNN in Task 1 as they offer some level of invariance and carry salient information about the objects (in Task 1:

morphological features). To further regularise the process of learning representation with a CNN one could use an additional 5% of data labelled with trees in the spirit of multi-task learning.

**Question 4** (6p)

Reservoir is a recurrent neural network.

To obtain better separation between states reservoir should be a large network with sparse (1-20%) connectivity (very often it is designed as a network with the small world connectivity pattern) and small spectral radius (less than 3) promoting stability.

Training echo state networks (ESNs) is fast since it is only concerned with training output weights, which is a linear problem so we end up with a least mean square problem.

The fundamental differences in solving the problem of sequence learning between ESNs and typical (vanilla) recurrent neural networks (RNN) or long short-term memory (LSTM) networks lies in learning/handling time dependencies. In RNNs we adapt "nonlinear" network parameters through training (backpropagation through time) while in ESNs these nonlinear parameters involved in recurrent processing are fixed and only the linear read-out weights are adjusted with linear methods.

LSTMs help in addressing the problem of vanishing gradients that vanilla type of RNNs suffer from, which allows LSTM to handle/learn longer sequences (longer time dependencies). In this regard, gating units in LSTM cells play a critical role.

**Question 5** (7p)

i) A representative example of artificial neural networks that allows for retrieving memory by content, a content-addressable memory, is the **Hopfield network**. A typical learning approach for Hopfield networks is **Hebbian learning**.

ii) To encode a memory by means of the Hebbian learning rule the input should correspond to the memory pattern to be encoded (with the dimensionality matching the network size) – a combination of -1 and 1 (bipolar encoding). Once the input cue is provided (the activity of each node is forced to the -1 or 1 state corresponding to the given input coordinate) the Hebbian rule adjusts the weight for each connection (if the pre- and post-synaptic unit states are the same the weight is increased, otherwise decreased).

iii) The number of memorised patterns depends on the memory patterns themselves (how much they are correlated), the size of the network and on the variations of the learning rule.

Generically, the network's memory capacity can be tested iteratively by gradually adding new random memory patterns to encode and testing the recall correctness over all the patterns encoded so far.

There are two main directions to enhance the memory capacity for specific data – either by **refining the learning rate** or by extracting and then using input data representations that lead to lower correlations (cross-talk between patterns).

iv) With the standard Hebbian learning rule it is expected to experience a so-called catastrophic forgetting where the recall of all memories is negatively affected by learning new patterns beyond the upper limit of the network's capacity.

*v)* Some key challenges/problems are

- Spurious patterns may occur in synchronous activity update mode (stable network states that do not correspond to any of the patterns learnt).
- Limited capacity heavily affected by the input data correlations (cross-talk).
- Limited data encoding approach (bipolar or binary patterns).

**Question 6** (8p)

| | |
|---|---|
| ***A.*** *Problem type that is addressed and a neural network architecture (name, the number of nodes in the layers)* | Classification problem<br>MLP or RBF (simple and small-size solutions due to hardware constraints) *(0.5p)*<br>(#inputs = 30x10 for a single 20-sec period; # sigmoidal outputs = 3 classes; size of the hidden layer has to be optimised). |
| ***B.*** *Training algorithm (name) and any data pre-processing + the recommended data representations/encoding* | Backprop for training MLP.<br>Data is already normalised so there is no acute need for preprocessing. Outputs could be made with one-hot encoding (here: 1 out of 3, e.g 001, 010, 100) |
| ***C.*** *Data usage – how data is used for training, validation and/or model selection etc.* | 352000 samples could be split into 60% training, 20% validation (model selection) and 20% for testing. For model selection purposes it is even recommended that the validation+training part is much larger to perform cross-validation (e.g. jointly 70-80%). The split should be stratified to preserved the representative statistics for the classes. |
| ***D.*** *Key hyperparameters, how is model selection conducted* | Model selection through cross-validation (evaluation) and grid search across the Cartesian product of hyperparameter subspaces (possible combinations of hyperparameters).<br>Key hyperparameters: #hidden layers and nodes, learning rate, hidden activation function |
| ***E.*** *Loss function and/or systems level performance metric* | Multi-class cross-entropy as a loss function.<br>Performance measured with classification accuracy and false/true positive rates or area under curve (AUC) for multi-class ROC (in one-vs-all or one-vs-one setup) |
| ***F.*** *How the neural network is used in the production cycle, i.e. how the predictions are made and/or the network's output is interpreted/evaluated/visualised/utilised* | For each input the output with the highest value provides the decision about the output class. Output sigmoidal values can also be interpreted quasi-probabilistically. The evaluation should be performed using performance measures over the test/unseen data partition. |
| ***G.*** *Potential challenges and risks plus extra short comments (e.g. on generalisation)* | Challenges/risks: potential class imbalance, data may have been collected using different microphones etc. so there may be problem with domain shifts (in consequence – generalisation), the distributions of data within each class can have very different characteristics (more vs less consistent classes), challenge to ensure generalisation with a relatively small MLP size (the network has to be simple due to hardware constraints). |