DD2437 Exam Answers, 22 HT (24/10/2022)

Question 1 (8p)

Ad.a) For Hopfield network it is desirable to use the Hebbian rule for learning. Hebbian learning modifies weights based only on the interaction/correlated activity of two connected neurons, so the rule is local. Encoding memory patterns implies that the energy landscape defined for the recurrent Hopfield network as a function of weights and neuron activities gets changed – in particular, as a result of memorising patterns there are corresponding local minima in the network's energy, i.e. the activity of the network matching the pattern learnt is a stable attractor state (with locally minimal energy).

Ad.b) In order to successfully encode multiple patterns (i.e. robustly so that the memory capacity is not low), they should be as uncorrelated as possible (their pair-wise dot products/correlation should be limited). For that purpose, real-world inputs should be first represented as sparse patterns with little cross-talk, i.e. as orthogonal as possible. One could rely on orthogonalisation operations, e.g. Gram-Schmidt process, but a more practical and scalable solution for large networks with a considerable number of patterns is the use of neural network based representation learning techniques – for visual input it could be a convolutional neural network together with (deep) autoencoder for sparsification and for auditory input – a deep belief network or autoencoder.

Ad.c) The key parameter controlling the capacity is the network size but the covariance of patterns also plays a role. To empirically examine the capacity one could incrementally encode new patterns and after learning each new pattern one could check if all patterns learnt previously can be successfully retrieved. This way one can identify the number of patterns where the network is no longer memorise previous patterns. As a consequence, the overloaded (beyond its memory capacity) Hopfield network is likely to suffer from catastrophic forgetting with recall performance dramatically decaying with newly encoded patterns

Ad.d) In principle, Hopfield network should be able to retrieve the original pattern (learnt) with only a partial (stimulus) cue but this depends on the correlation with other patterns and with the proportion of missing pattern components (the ratio of cued units) in relation to the network size.

Question 2 (4p)

The temperature parameter is part of the transfer function of the node of the Boltzmann machine. It regulates the level of stochasticity of the node output. More specifically, it determines the steepness of the transfer function that maps the summed input to the probability of the output to be 1 or -1. During operation of the network, the stochasticity which the temperature controls leads to fluctuation in the total energy of the network and this forms the basis for escaping local minima and transcend towards the global minima. The procedure is called simulated annealing whereby the temperature goes from a large value to a small value over many small steps.

Question 3 (9p)

This is a clustering problem or neighborhood preserving map problem. I use a SOM-network motivated by the need to produce an output that shows which locations are similar to which. The input is the 15-dimensional feature vector for each of the measurements and output is the SOM network output, see more below on how output is calculated.

Architecture details (pooling layer, conv layer), hyperparameters. I will use 1225 number of RBF nodes (approximately 10 times the number of locations) arranged in a 2-D 35x35 rectangular connectivity grid.

The result is presented in the output space. To determine the output, I will iterate over all the patterns once. For each pattern, I print the name of the location at the node which responded most strongly to that location. Since a node might respond to more than one location, each location is printed slightly shifted in some direction so that location names can still be read. RBF-units are initialized using data samples, to avoid dead units.

The different input dimensions/channels are normalized separately for each channel. Training is done using winner-takes all competitive learning and the Delta-like weight update rule. I will iterate over all the patterns 100 times while decreasing the neighbourhood size from NS to 1 and decreasing the learning rate from EPS to 0.001, where NS and EPS are two of the hyperparameters to experiment with.

To improve robustness of interpretation (kind of generalization), I make my selection of hyperparameters based on a combination that gives more compact ("smooth") boundaries between the region (group of nodes dominated by one location) and the region of another location. That is, if a location has its name printed in a pattern that looks like a star-fish it is deemed bad and if it is smoothly elliptic it is deemed good. The rationale for this is that I want as smooth and regular transition boundaries between regions so that if we move along a direction, we only want to have at most one change of region.

Note: you cannot apply cross-validation for SOMs. You also typically do not use dropout, but some amount of noise could be added to the data.

Question 4 (6p)

Ad.a) Given little available data it would be advisable to rely on transfer learning and use the existing labelled data to train particularly fully connected classification layers of the CNN.

Ad.b) To benefit from the unlabelled data it is recommended that some form of semi-supervised learning is used. One way would be to deploy a generative model with the use of a CNN, for example a generative adversarial network (GAN) or deep belief net (DBN), which can learn representations without labels. Then it should be feasible building on the learnt representations to use only a few existing labels for training the classification layer(s) or even tuning weights in the entire network structure to adjust representations.

Ad.c) Data augmentation for training the network is recommended. In principle, the existing training data could be modified by unstructured noise, varying contrast, translation/rotation of salient objects in images etc.

Question 5 (10p)

Ad.a) It is a regression problem where three images are mapped to a continuous output value.

It is suggested that a convolutional neural network (CNN) is used to process image data.

The number of inputs to the CNN should correspond to the number of pixels in all three images and there could be either a 3D filter applied in the convolutional layer across the three images to produce a 2D map (like in the treatment of colour channels in RGB images) or, alternatively, three independent streams of CNN processing could be used until the representations are combined and passed on to the fully connected regression layer(s) with the final layer consisting of a single output unit. The activation functions could be ReLUs, the training should be performed with backprop and the loss function to minimise should be the mean square error.

It is suggested that 900 patients are divided into a training/validation set of, say, 800 patients and the remaining 100 test cases given that it is feasible to build these sets preserving the same and uniform (as much as possible) class distributions. Data of the 800 patients would be used for cross-validation to optimise hyperparameters (learning rate, hidden layers and any other design assumptions/decisions). Then, for the optimal set of hyperparameters the model should be trained from scratch on the available 800 patients (or less if one wants to use early stopping etc.) and then tested on the 100 test cases.

The testing sample should be matched with the clinicians' scores in terms of a correlation plot – one variable against the other one across all the test samples. The quantification could also be a correlation coefficient. Alternatively, if there is a preference to ensure good fit for more risky medical scenarios, they could be at focus for example by upscaling their importance in the correlation calculations.

Ad.b) Using an arbitrary threshold for the diagnostic score the data could be pulled into two sets with the corresponding ranges. Naturally, some data with scores between 1.5 and 3.5 would be omitted. As for the network training, it is suggested that it is repeated for the new dichotomous sets of labels and in the framework of a classification task (with entropy loss function and sigmoidal output units). An alternative, which is not recommended (worse in terms of performance), would be to reuse the existing network (case a) and postprocess its output.

The newly trained network (following a cross-validation based model selection process) would be then evaluated on the new 100 cases. As the metric communicative to the clinicians one would report specificity and sensitivity (false positives, true negatives etc.) or the area under the curve of receiver operating characteristic (AUC ROC).

Ad.c) Essentially the comparison is between predicting the diagnostic outcomes based on a sequence of images vs a single image. In both cases the number of samples (either an image or a sequence of images) corresponds to the number of patients (600). Essentially, the classification approach with a

CNN producing a dichotomous decision, developed in b), could be reused for a single image classification. It could also be used for a sequence-based classification by concatenating the representations learnt from each image in a sequence and feeding the composite vector to the fully-connected layers for classification. HOWEVER, this path is sub-optimal and hence not recommended here. To fully leverage the sequential information (and partly build on the CNN method developed in b) it is actually suggested that a recurrent neural network (RNN), e.g. long-short term memory (LSTM), operating on CNN representations is developed. Alternatively, an MLP with a time-delayed input configuration could be deployed for processing sequences of CNN image representations.

There could be two major directions in the evaluation – 1) to train and validate on a subset of available 600 patients (e.g. if the cross-validation is used for model selection then 500 cases should suffice for that purpose) and to use the remaining cases for the final evaluation purposes, or 2) to perform cross-validation on all 600 patients (recommended) and statistically compare the outcomes for a group of the cross-validation test folds (e.g. for 10-fold cross-validation we have 10 results/outcomes per method).

Question 6 (9p)

Ad.a) The problem at hand is a classical example of time series prediction. Therefore, candidate neural network approaches include multi-layer perceptrons (MLPs) that can perform nonlinear autoregressive (past data samples are used to predict future) modelling and recurrent networks. Considering the latter family of methods, there are a number of architectures including echo state networks but recently long short-term memory (LSTM) architectures have been shown to perform particularly well in time series forecasting problems. So, for a comparative analysis MLP and an LSTM networks can be chosen.

In both cases time series embedding has to be decided upon (the number of past observations used to predict the future observation(s)). Also, the decision has to be taken as to whether it is a single- or multi-step ahead prediction. Another decision to take is whether all garments should be treated separately or rather incorporate a multivariate approach combining information and predictions for all 4 garment types. The latter option is preferred as there are likely co-dependencies between garment sales that could be exploited by the prediction network.

A fair comparison would require that the same past observations (time series embedding) are considered to predict the same future observations (single- or multi-step for the same prediction horizon). Also, the number of free tuneable parameters between different networks should be a factor to take into consideration. Hyperparameters on the other hand should be selected independently for each network (e.g., number of hidden layers and units in MLP; the number of neurons and batch size in LSTM) and validated on a validation data subset.

Ad.b) The data should be split into three non-overlapping subsets for training (say, 80%), validation (10%) (or collectively 90% training/validation if cross-validation is used for model selection), and testing (10%). It is important that continuous (in time) chunks of data are used.*)* It would be advisable to keep together data from the same company within one of these sets (rather then split them across training, validation and test subsets) in the attempt to develop and evaluate a generalisable solution across factories.

Ad.c) Since there are periodic patterns on a monthly basis, one approach would be to have 4 inputs (assuming there are 4 weeks per month), x(t-3), x(t-2), x(t-1), x(t) and one output x(t+2x4) or multiple outputs sampling the prediction horizon up to x(t+8) for MLP, where x is the univariate time series for one garment type. Analogously, there would be a sample frame of length 4 with one output in LSTM.

In the case of a multivariate approach the number of input streams is quadrupled.

Ad.e) The criterion for comparison could be the normalized mean square error as the prediction performance measure. Since of the key source of uncertainty comes from random initialisation and potential data reordering, it is important that multiple simulations are run and the mean results are compared using statistical null hypothesis testing (where the distribution of the results from multiple runs matter, not only the means). In relation to the discussion of the generalisation (already referred to in b), it is desirable to examine whether the prediction power generalises across networks. This can be answered by a leave-one-out cross-validation at the company level (leave one company out).