DD2437 Online Exam Answers, 22VT (2022/03/12)

Question 1 (4p)

Fundamental similarity between transfer learning and multi-task learning (MTL) is that they perform domain adaptation, knowledge transfer. In deep neural networks both MTL and transfer learning rely on distributed hierarchical representations.

The key difference lies in how the knowledge transfer is performed. In transfer learning knowledge is sequentially passed on between tasks/domains. We first train a network on one source task and then use it on another target task (often with subsequent fine-tuning/ retraining). MTL is simultaneously trained on multiple tasks. Also, the selection of tasks in MTL is usually done to support the main task (so we find auxiliary tasks for that purpose) whereas in transfer learning it is more common for the tasks (datasets) to be independent. Simultaneous learning of multiple tasks in MTL implies also that the MTL network has multiple output modules (corresponding to different tasks) whereas transfer learning relies on the same configuration of the outputs.

Question 2 (9p)

A) It is a classification problem with 48 classes. Since data samples are images it seems most suitable to use a convolutional neural network (CNN). It has to be equipped with 48 sigmoidal outputs and the input space should correspond to the size of the images. One challenge at the start is that 48 classes requires a large network and a considerable number of training samples, this could be a potential problem. Since the training images were collected in a lab environment and then mobile app is supposed to work with photos collected in real-world context (out in the nature), there is a risk that the training data and data for recall have different statistical properties (come from different distributions). It could also imply that the images are of different sizes. To minimise the risk of poor generalisation, regularisation techniques could be used though may not be a sufficient preventive measure. In that case some form of data augmentation, potentially denoising could be tried. If the images are of different size some form of cropping or resizing (with upsampling or downsampling) needs to be employed to ensure the corresponding size of the input layer. Finally, as mentioned, this is a mobile app so there should not really be any learning performed on the mobile device so the training has to be done earlier off the device and for the recall only weights uploaded to the app. If there was a requirement to train on the device then a different neural network model may have to be chosen.

B) The problem is now less precisely defined with respect to the output categories. If they are not given as labels accompanying photos then a more exploratory approach has to be taken. For that it is recommended to apply a self-organising map (SOM) capable of clustering the outputs. To avoid working with images in the SOM input, it is recommended to use hidden representations of sample images obtained from the CNN network in A. If there are some data samples (based on the neighbourhood in the output grid space). This way we can flexibly combine labelled data with unlabelled data, categorise to a predefined number of categories or form new categories

Question 3 (8p)

This is a temporal prediction problem and thus a regression problem. I will use an LSTM since this architecture can handle data with multiple time scales. The input to the network are the 41 factors and the output is 3 nodes representing the prediction for each of the 3 days. The training is done by using the sales prices as target data, using a training window of 7 days of data and using Backprop as the learning rule. I will use cross-validation and partition the data into 80% training and 10% test and 10% validation (where the test set is used for early stopping). I will enhance generalization by performing data augmentation (adding various forms of transformations) or add noise to data, use L2 regularization and apply dropout. To test generalization capability, the crossvalidation (see above) is used. Potential challenges or difficulties includes there could be a factor imbalance problem as some fish products or "external factors" could be rare. Also, the 7 day time window might be too small for some trends.

Question 4 (9p)

A) Since the original data that the biologists deal with is in the form of images, a convolutional neural network (CNN) appears particularly suitable to extract the desirable features. 90% of manually annotated data is enough to set up a CNN as a classifier and/or regressor that maps input images to five selected qualitative descriptors. For that purpose the proceed architecture has the input layer corresponding to the image size and the output layer should have 5 units with transfer functions corresponding to the nature of the descriptors (e.g. continuous descriptors- > linear units, binary descriptors -> sigmoidal units). The CNN could be trained and validated on the available 90% of data and then used for new data to infer the descriptors. The learning algorithm is backprop with labels corresponding to the aforementioned 5 descriptors.

B) Once the descriptors are extracted from individual samples/images, then their sequences can be processed. There are a number of ways we can identify repetitive patterns of fixed length (here: corresponding to 3 hours). We can use a recurrent neural network (RNN) or time-delayed feed forward network (MLP) that maps sequences into scalar outputs. In this case, the output corresponds to a tissue type. For both MLP and RNN, the data should be divided into a training and validation set. To do that, a moving window through multivariate (5 descriptors) time-series

should be applied to generate a pool of sequences. This pool then can be divided into training (90%) and validation (10%) subsets preserving temporal ordering. The learning algorithms are either backprop through time (RNN) or backprop (MLP), and their hyperparameters (mainy the parameters of the network architecture except the size of the input and output layers) should be selected based on the validation performance (estimate of the generalisation error). After successful training and validation, we can pick sequence samples that are correctly classified.

C) Taking sample sequences that are correctly classified in B), we now have to explore how they group (across tissue labels). This is a case of an exploratory analysis with the recommended unsupervised learning approach. To this end, a self-organising map (SOM) could be chosen. The sequences (frames of data, i.e. fixed-length sequences of 5-dimensional vectors of descriptors) can be fed into an input SOM layer and mapped to a two-dimensional output grid. After training with the available data (sequences mentioned above), we can identify grouping based on the activations in the output grid. Ultimately, we can perform clustering of these activations and the resulting groups would correspond to the requested categories. In fact, we should investigate how the output categories match tissue labels. We would expect that sample sequences characteristic of specific tissues are grouped together, potentially with some groups accounting for more than a single tissue (as requested in the question C).

D) The most straightforward way to tackle sequences of varying duration would be to use a long short-term memory (LSTM) network in the place of the RNN in q.B. LSTM setup is more flexible and allows for processing sequences of varying length.

(Alternatively, exploring sequences of longer duration could be done with the extended network based on the design in q.B. In fact, rather than classifying sequences now the network could be re-purposed to perform time series prediction. The input data would be sequences identified in q.B and the output should be configured to allow multi-step-ahead prediction. The quality of the prediction beyond the original 3 hours (the fixed-size sequence length proposed earlier) would testify the feasibility of using longer sequences.)

E) Building on the results in q.C, we just use the sample sequences as inputs and the corresponding categories identified from the SOM output in C to build a classifier.

The most recommended approach to building a diagnostic tool that solves a classification problem is an RNN that maps the input sequences to the aforementioned categories. For training a backprop through time should be used. The same data used in q.C should now be divided into training, validation and test sets. Hyperparameters describing the dimensionality of the hidden layer(s) should be selected with the estimate of the generalization error obtained on the validation set as the criterion. Unseen test set should then be utilised as the evaluation of the quality of the diagnostic tool produced. Even if cross-entropy is exploited for learning purposes, the test evaluation could be made more interpretable for the doctors by redefining the performance

measure as a false positive rate or other ways to capture sensitivity vs specificity of the diagnosis (even the receiving operating characteristic, ROC curve, could be used for that purpose).

(Alternatively, given the nature of the input data it could be interpreted as images (sequence length by 5 descriptors) and then a CNN could be used for classification (here the context is different from that in A, where we explicitly worked with images) or a simple MLP with the vector inputs of length corresponding to the sequence length multiplied by 5 (5 descriptors). In these cases, backprop for learning should be used.)

Question 5 (7p)

This is a regression problem, to map socio-economic and geo-political factors to air particle factors. I will use a multi-layer perceptron to solve the task. The reason is that it is a regression problem where factors are mapped to features. The input consists of the 28 factors and the output will be 15 nodes (the 14 features + 1 number). I will start by using one hidden layer, and the number of nodes will be decided based on experimentation (watching out for overfitting, see generalization below). For training, I will use 10-fold cross validation and partition the data into 80% training, 10% validation and 10% test. The use of the test set is for early stopping. Backprop is used as the learning rule. Generalization is assessed by using cross-validation. To enhance generalization, I could apply some regularization technique like L2 or use dropouts. In terms of challenges or potential difficulties, there could be an imbalance problem in that some input factors could be rare.

Question 6 (9p)

A) It is a classification problem so having unbalanced data poses a serious challenge. To minimise its effect, one should perform some form of data augmentation, in this case upsampling the minority classes and downsampling the majority class could be a solution. However, to tackle this problem with a neural network implies the need to design a generative model that learns the data distribution and then can serve as a sample generator. This way new samples representing the minority class, in particular, could be obtained to minimise the class imbalance. For this task a deep belief network (DBN) or generative adversarial network (GAN) or variational autoencoder (VAE) could be used. The most straightforward approach would be to employ DBN in a similar way as in lab 4 since it facilitates sampling from the chosen class (for GANs and VAEs, a class-conditional version would have to be utilised). The dimensionality of the input should correspond to the size of the input images and the output layer would include label units (to encode classes with one-hot-encoding scheme). The training would be first performed with contrastive divergence and then the weights could be fine tuned with contrastive wake-sleep algorithm. The downside of

applying a DBN is its moderate scaling capabilities wrt. the input dimensionality and the fact that this architecture is not particularly suitable for image data. Also, the generative model for the underrepresented (minority) class could be impaired due to a low number of samples, especially for the minority classes. An alternative would be GAN or VAE implemented with a convolution neural network (CNN). The risk then however is also associated with a rather low total number of data samples for training.

B) To deal with noisy data it is recommended that we work with autoencoders (AE). This could have been incorporated into VAE approach used for generation in p.A. However, to trat this set independently, one can recommend the use of denoising autoencoders (AE) with CNNs (still, we are working with spectral images). For that a rather shallow AE is recommended since the number of available data may still be low. If we manage to generate a lot of samples in p.A, then the number of AE layers could be treated as a hyperparameter. The learning algorithm is backprop with mean squared error (MSE) as the loss function. Importantly, after training, the objective could be either to work with AE representations or de-noised images obtained in AE outputs. Let's assume the latter option for further processing.

C) Since it is a classification problem that involves image data (time-frequency, 2D, spectral transform of time series data), it is recommended that a CNN network is employed. The network dimensionality should be reduced as much as possible since the overall amount of data is low (this can be prevented partly by the generative sampling in p.A, even though the focus there was on the minority class). The size of the CNN input layer should correspond to the size of the spectral images and the number of outputs corresponds to the number of conditions - 5. The transfer function of the output nodes should be sigmoidal and the backprop training algorithm should optimise the cross entropy loss, The hyperparameters (model) selection should be performed with a validation scheme using the generalisation loss estimate on the validation data (it would be desirable to run a cross-validation scheme but the number of data samples could be too low for that). Classes should be rather evenly represented in each of the data subsamples, i.e. training, validation and test subsets (folds). Comparing different models follows the same procedure as choosing hyperparameters (model selection) described above except that more attention is given to statistical validation. When comparing different network models it becomes important to provide statistical evidence for their difference (null hypothesis testing with a statistical test). For this however, a population/set of independent results to compare has to be obtained (rather than comparing single-run results). We can aim for quasi-independent results using the outcomes of the test folds in the 10-fold cross-validation scheme (this way we can obtain 10 quasi-independent results that could be subject to a nonparametric statistical test).

D) To make a data driven approach to the problem of absorber’s lifetime prediction, new data should be collected. It is recommended that at regular intervals absorbers are examined/diagnosed and their spectral response is collected. This should ideally be done on a relatively large number

of absorbers until the end of their lifetime. This would allow for annotating collected samples with the time-to-failure value. Then one could build a simple CNN that would map the input spectral image to the time-to-failure output. CNN is motivated by its suitability for image-like data. Alternatively, to account for the temporal aspect (trajectory) of the absorber's changing condition towards failure, a recurrent neural network (RNN/LSTM) or time-delayed feedforward network (MLP) could be employed with CNN representations in this regression task with temporal/sequential input (mapping a time series of CNN representations of spectral images to a time-to-failure value). This approach could provide more reliable results as it allows for exploiting relevant information about temporal dependencies. One of key challenges is the actual data collection since a relatively large number of absorbers would have to be exploited until their failure given the variability of the target value.

(An alternative approach would be to perform a time series prediction on the same data, in that case we do not need a time-to-failure annotation as we only predict the state of the absorber. This way, we follow in time the deterioration of the absorber's condition until failure. To perform such a few-step-ahead prediction, we could use the same RNN (or LSTM) or MLP).