Using Ensembles Of Neural Networks With Different Scales Of Input Data For The Analysis Of Telemetry Data

Yauheni Marushko, United Institute of Informatics Problems of National Academy of Sciences of Belarus

Abstract
This article gives a brief description of the main methods of forming parallel ensembles of experts, in particular ensembles of neural networks. Also the learning algorithm of neural network ensembles with elements of the evolution strategy described. The problem of concept drift and methods of its solution using incremental learning ensembles of experts described. Also the method of searching the important features of the time series for forecasting is given. And the model of two-level learning algorithm of neural network ensembles with different time scale is proposed for the prediction of telemetry data. This model is based on an ensemble of neural networks that use mentioned algorithm of learning. Also presents the results of tests of the model on the task of forecasting multivariate time series of telemetry.

1. Introduction
Telemetry covers the measurement of physical quantities characterizing the state of objects or processes, the transfer of the results of these measurements, recording and data processing.

The processing and analysis telemetry data in a continuous process is accompanied by noisy data. This usually creates by non-deterministic sources of noise. This makes it the preferred use of the technology of artificial neural networks.

The following benefits are the basis of the development of systems to analyze telemetry data using a neural network [1]:
- The ability for self-learning, the replacement of complex mathematical apparatus necessary volume of information, the initial function of its transformation and the structure of the neural network;
- The possibility of realizing nonlinear mappings with substantial non-linearities in the hidden neurons of multilayer neural networks;
- High degree of parallelism possible for a neural network software system in real time.

These advantages provide application of neural network algorithms in the diagnosis of complex dynamic objects. Neural networks appear here as the apparatus of formalization of complex algorithms that convert the information.

There are examples of the use of neural networks in on-board intelligent decision support systems for managing complex dynamic objects and diagnosis of its condition [2].

The problem of forecasting requires an individual approach to processing data when using traditional neural networks. Such an approach requires a lot of time and resources, as determined during the design of a large number of parameters of both the neural network model and the processing parameters of time series (prediction window, the time horizon, etc.). It is therefore necessary to develop a tool automatic settings and automatic choice of architecture. Such a mechanism can be implemented using neural network complexes [3, 4, 5]. So an important task is to develop a neural network structure, which allows you to automatically search for the optimal solution. Such structures can be used to determine the preferred structure of a neural network, or to determine the individual characteristics of the structure and process. Most often, these structures are implemented based on ensembles of neural networks.

Also, there is a problem of learning in a nonstationary environment (or learning “concept drift”), where the underlying data distribution changes over time. So, the main goal of machine learning methods is to learn from large volumes of data that come from real applications, then the need for a general framework for learning from – and adapting to – a nonstationary environment can be hardly overstated. Given new data, such a apparatus would allow us to learn any novel content, reinforce existing knowledge that is still relevant, and forget what may no longer be relevant, only to be able to recall, if and when such information becomes relevant again in the future [6].

2. Ensembles of neural networks (ENN)
The parallel organization of neural network models is typically referred to as an ensemble, and the involved models themselves are referred to as experts. Each expert is a particular method of
solving the same given task. With parallel organization of the separate models, each of the experts solves the task with a certain quality, and the combination of the experts (the committee) serves to improve the resulting quality of the solution.

Two criteria must be optimized during the formation of the ensemble of models - the quality of training of particular network and optimal integration of models.

There are several basic methods of combining independent models in ensembles: Bagging, Stacking, specialization of experts, maps of expert mixture of experts, Learn++ [6, 7, 8, 9, 10].

2.2 Combining using bootstrap

The approach is based on the independent learning of individual models on bootstrap samples from the training data set and then combining the obtained models in the committee majority (ensemble). The method is called "beggings" (bagging - bootstrap aggregating) [8]. For the sequential computation committee may expand in stages, with the first results of the classification system begins to produce after the first training models.

2.3 Stacking

The algorithm uses as the base models different classification (prediction) algorithms that are trained, using cross-validation, on the same data. Then, a meta-classifier (supervisor) is trained on the input data, supplemented by basic algorithms. The supervisor can use when teaching the estimates of parameters of the distribution (for example, estimates the probability of each class) instead of the results of basic algorithms [9].

2.4 Specialization of experts

The method assumes that after training all of the examples are stored along with the estimates of errors that were made to them by all the members of the ensemble. Then, after receiving a new example, it matches closest prototype of the training set in the database and in the vote involving only those members of the ensemble, which allow small errors found on prototypes. Thus, there is a dynamic union of some members of the ensemble for solving each new task [7].

2.5 Maps of expert

Dynamic aggregation can be extended if, instead of storing the prototypes use a cluster structure (e.g., Kohonen maps), which is built on the training set without tags classes and assign to each cluster a list of models that show the best results on the data of this cluster. For the classification is now sufficient to find the nearest cluster and apply its associated model [7].

Each member is assigned to all clusters in the data from which it has a small error, thus the clusters maintain their rating list of models. In particular, based on the map as requests, we can define the "problem areas" of the ensemble. These clusters are filled with data, but the poor are associated experts. This gives a new training strategy and replenishment of the ensemble with new models.

2.6 Mixtures of experts

Combining experts in the processing of a new sample is made with weights that are determined by the control outputs of the neural network. The task of the control neural network - choose the weight of the mixture, the most relevant being processed at the moment the input vector [10]. In various versions of the model as a control network developers use the radial basis function, the system of fuzzy rules, hierarchical structures and other models. Can say that mixtures of experts generalize the above defined models.

2.1 Definitions of concept drift

Informally, concept drift refers to a change in concept definitions over time, and therefore a change in the distributions from which the data for these concepts are drawn. An environment from which such data is obtained is a non-stationary environment.

A shift in likelihood would seem to indicate that the event labels may also be changing. However, it is not until the distribution of one event shifts such that the true event boundaries are altered that we can call this change a real concept drift. Event drift without overlapping of true event boundaries is known as virtual concept drift, and merely shows that the learner is being provided with additional data from the same environment.

Virtual drift is the result of an incomplete representation of the true distribution in the current data. The key difference is that real drift requires replacement learning (where old knowledge becomes irrelevant), whereas virtual drift requires supplemental learning (adding to the current knowledge).

Determining when such a change, i.e., whether concept drift has occurred, is known as drift detection. As mentioned above, concept drift algorithms can be active or passive with respect to the drift detection mechanism. An active drift detection method seeks to pinpoint the time and severity of the drift, and allow the classifier to modify or continue learning accordingly. A significant downside of active learning, however, is the risk of having an imperfect detection mechanism which may—and often does—yield false reports, an all too common occurrence particularly for noisy datasets. In passive drift detection, however, the
learner acknowledges that the environment may change at any time or may be continuously changing. The algorithm then continually learns from the environment by constructing and organizing the knowledge base. If change has occurred, this change is learned. If change has not occurred, existing knowledge is reinforced.

2.7 The Learn++ Family of Algorithms

Provided above methods allow to train only once a set of neural networks, and some of them can also dynamically adapt to changes in the data, using some kind of weighting (in case of specialization, map, mixture of experts). But these methods do not have additional training. And in the case of a significant concept drift their error will increase.

To overcome this, methods can be supplemented by an incremental learning like Learn++ algorithms.

The common denominator in all Learn++ algorithms is an ensemble of experts that are incrementally trained (with no access to previous data) on incoming batches of data, and combined with some form of weighted majority voting. The distribution update rule for choosing data for training subsequent ensemble members, and the mechanism for determining the voting weights are the distinguishing characteristics of different Learn++ algorithms.

For example, Learn++.NSE is an ensemble-based batch learning algorithm that uses weighted majority voting, where the weights are dynamically updated with respect to the experts’ time adjusted errors on current and past environments. It employs a passive drift detection mechanism, and uses only current data for training. It can handle a variety of nonstationary environments, including sudden concept change, or drift that is slow or fast, gradual or abrupt, cyclical, or even variable rate drift [6].

2.8 Using ENN to process concept drift

So, ensemble based algorithms represent a new breed of algorithms to learning in nonstationary environments, and they are particularly effective at providing a good balance between stability (retaining existing and relevant information) and plasticity (learning new knowledge) in the presence of drift. The approach consists of an ensemble of experts, combined to form a final representative decision. In order to prevent irrelevant knowledge from effecting this decision, a combination of voting techniques and forgetting mechanisms are employed. Voting based combination (ensemble weighting) allows experts with varying competences on the current environment to proportionally contribute to the final decision. Weights are often dynamically updated at each training instance, independent of the existence or amount of drift.

2.9 Ensemble pruning

While lower weights allow the knowledge of certain experts to be temporarily forgotten, ensemble pruning allows knowledge believed to be completely irrelevant to be permanently discarded [6].

Pruning can be useful because the ensemble, growing otherwise uncontrollably over time with new experts trained on new data, may accumulate too many irrelevant experts that can outvote the competent experts in the ensemble, despite weighting strategies. Pruning also helps with another concern with ensemble approaches, namely, the computational complexity that increases with each new expert generated for incoming data. However, that permanent removal of classifiers through pruning runs the risk of discarding information that may later become relevant, should the environment happen to be a cyclical one [12].

Several criteria can be used in pruning. The first criterion is mean square error. The goal is to find a set of n experts that has the minimum mean square error. There also are instance based ensemble pruning [12] and age-weighted pruning [6].

3. Training ensembles of neural networks

Parallel training of ensembles can be optimized by using the elements of evolutionary approaches. The basic operation here is a mutation where in which the connection weight is perturbed by random value with a certain probability. Also one of variants crossover can be applied if for every single model can identify the "genotype".

The algorithm of formation of population consists of the following steps [13]:

1. Initializing weights of each neural network from ensemble.
2. Mutation of weighting links, applied crossover if possible, training.
3. Survival of the fittest. Survival implies the elimination of a set of less adapted experts, and duplication of the fittest.
4. Go to (2) or stop if stopping criteria met.

After the formation of the most successful experts gating element must be formed. It offers two strategies:

If it is assumed that the input time series does not contain the concept drift, or adverse effects are eliminated by preprocessing (removal of the well-known seasonal variations, for example), it can be used a wide range of techniques: neural network, a simple weighted summation, adaptable weighted summation and so on. Gating module is trained on the outputs of single neural networks, when applying the same training, examination and test data sets.

The resulting performance can be improved by increasing the accuracy of each individual model and simultaneously ensuring statistical independence of
the errors of different members of the ensemble. In practice, the statistical independence of the error distribution is checked by examining the linear correlation of all pairs of the models from the ensemble. The closer a certain member of the ensemble resembles the other members, the less benefit it confers, and the less should its weight be when making the final decision. For the errors to be not correlated, it is desirable that the experts be diverse. The particular method of calculating the weight of the expert depends on the type of problem [14].

Otherwise, you must use an algorithm similar to the weighing Learn++.NSE.

4. Two-level learning model

Using only a single ensemble at designing of a forecasting model often does not give the desired result, and leads to development that is repeated over and over again with varying architecture, varying the types and learning algorithms of neural networks [5]. It is proposed to use a two-level learning model [13]. This model allows realizing heterogeneity of the complex neural network. The first level of structure can be: different types of neural networks [3, 15, 16]; similar networks with different time-series parameters [4]; similar networks with different training parameters.

On the basis of the methods described above a two-level learning algorithm of neural network ensembles with different time scale is proposed for the prediction of telemetry data (Fig. 1).

On the first stage of the training problem of finding precursors is solved, i.e. determination of a set of the most significant input features in coordinates “initial time series – lag”. The combination of the most significant input features thus extracted makes up the precursor, whose dimensionality is much lower than that of the initial problem.

This algorithm supposes that some (unknown) combination of values of incoming features, which is called a phenomenon, precedes the occurrence of the interested event such that the delay between the phenomenon appearance and the event occurrence is fixed (but also unknown). Suppose also that an interval of search for the delay between a phenomenon and an event occurrence is given (∆T from T_{min} to T_{max}) and that the duration of the phenomenon is significantly less than the interval of the search. A precursor is a combination of only those features forming the phenomenon, which are essentially related to the event occurrence and can be used for its prediction. To find the most probable phenomenon, the analyzed interval of search is divided into overlapping segments of length equal to the interval of initiation. For each segment, an individual neural network is constructed that can learn to predict the event on the basis of features in the given segment. During the training process, the interval of search is shifted along the time axes. The sequence of responses of the networks for the given time instant can be considered as an estimate for the event probability by a ensemble of independent experts. Shifting the search interval along the time axes and applying the set of neural networks to the corresponding segments of the analyzed series, we can predict the event occurrence. After the end of learning, we can conclude that the desired phenomenon (and, hence, the precursor of the event) belongs to a part of the net that provides the most exact prediction of the event from the results of learning. Thus, the delay between the phenomenon and the event is determined not worse than the distance between neighboring overlapping segments [17].

After analysis of feature significance, a new model (ENN) is built that is like the base one except that it does not take into account the low-significant features. Thus, the weighting algorithm based on the accuracy of the model and the importance of precursor used for generalize to individual neural networks. This allows using incremental retraining algorithms for the detection concept drifts. At the same time, the use of individual parts of the time series (precursors) imposes restrictions in this, as concept drift can affect precursor.

If prediction quality of the new model turns out to be not worse than the quality of the base one, a conclusion can be drawn that the discarded variables were really insignificant for the prediction problem being solved [17].

This algorithm is also applied to the input data of larger scale. A larger scale can be derived from the input time series by averaging over a certain step, either directly from the data source. These data can be used for going deeper into of the past time-series with less precision.

Thus, an ensemble of experts is available for each of the selected scale of the series. It should be borne in mind that the experts for large scale are added to the final model only if they affect the final decision.

A neural network is used as a gating module of the second level.

5. The experimental part

First, pre-processing is carried out: data scaling, data normalization, factor analysis. Telemetry data, in the form of multivariate time series, often contain a large number of items that are correlated, and therefore redundant for training. In this case it is useful to reduce the size of input vectors. To do this, Principal component analysis or Exploratory factor analysis is used.

For forecasting, we used set of «The Santa Fe Time Series Competition Data» and synthetic
generated multivariate time series of states of objects.

Telemetric information about the object was considered in two ways (see Fig. 2). In the first case, the data from the sensors are analyzed directly, and then the resulting forecast determines predicted state. In this case, training and processing is done on the initial data.

In the second case the input data is classified into a number of states of the object, which is analyzed, so it significantly reduces the volume of data being processed. Comparison of predicted and actual data is shown in Fig. 4.

The proposed two-level prediction model with different time scale investigated. As the neural network used feed forward neural network with a hidden layer with RProp learning algorithm. In the first phase of training 4 scale (the input element of the time series contained a mean of 1, 2, 4, 8 values) were investigated. In most cases, 4 and 8 did not bring more precision, so were discarded. Also, only a small portion of single neural networks discarded for the initial time series, this suggests that the predicted value depends on all the data in the prediction window.

Further gating module was trained on the outputs of all the ensembles of the first level (or rather the outputs weighing modules). The result of this module is a short-term prediction for time series. Incremental additional training used minimally, because a significant drop in forecast accuracy in the processing does not occur. Ensemble pruning is not used, because the small size of single neural networks do not lead to a significant increase of the complexity of during the experiments, also in weighting all the elements of the ensembles showed positive weight or have had a zero weight insufficient period of to remove.

Accuracy of the prediction of single neural network ensembles of neural networks, and the proposed two-level model was compared. We used the already defined precursors of the series. The ensemble of neural networks showed accuracy similar to the best single neural network, with the training and selection of architecture does not require time-consuming, Whereas to determine the best of the neural network 600 single network trained with a variation of the size of the hidden layer and some learning parameters. A two-level model showed result similar with the result of ENN, showing improvement only at some points due to the use a second scale.

The results are shown in Table. 2: the average error on time series (Er. 1), and on classified data (Er. 2).

<table>
<thead>
<tr>
<th>Model</th>
<th>Er. 1, %</th>
<th>Er. 2, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single ANN</td>
<td>5,5-16</td>
<td>3,5-13</td>
</tr>
<tr>
<td>ENN</td>
<td>4,6-7,2</td>
<td>2,5-6,5</td>
</tr>
<tr>
<td>The proposed model</td>
<td>4,3-5,3</td>
<td>2,5-3,7</td>
</tr>
</tbody>
</table>

6. Conclusion

In this article, the information was given on the basic methods of forming ensembles of experts. The problem of concept drift and methods of its solution using incremental learning ensembles of experts described. Also the method of searching the important features of the time series for forecasting described.

On this basis, proposed a two-level prediction method using a variety time scales. This method was compared with single models and ENN. The comparison showed that the necessity to use a two-tier structure is not always required. Depending on the requirements can be quite ENN. Also in the solution of new problems using incremental learning requires a data analysis, as it may not be necessary.
Bibliography


[6] Incremental Learning of Concept Drift in Nonstationary Environments


[12] Mining Concept-Drifting Data Streams using Ensemble Classifiers


[14] Hierarchical Approach to Forecasting Recurrent


[17] Multi-stage Algorithm Based on Neural Network Committee for Prediction and Search for Precursors in Multi-dimensional Time Series

Authors:

Yauheni Marushko
United Institute of Informatics Problems of National Academy of Sciences of Belarus
st. Surganova 6,
220013 Minsk

email: marushkoee@gmail.com