in: Artificial Neural Networks and Machine Learning - ICANN 2012, Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

Comparison of Long-Term Adaptivity for Neural Networks

Frank-Florian Steege[1,2] and Horst-Michael Groß[1]

[1] Ilmenau Technical University, Neuroinformatics and Cognitive Robotics Lab, P.O.Box 100565, D-98684 Ilmenau, Germany
[2] Powitec Intelligent Technologies GmbH, 45219 Essen-Kettwig, Germany
frank-florian.steege@stud.tu-ilmenau.de

Abstract. Neural Networks can be used for the prognosis of important quality measures in industrial processes to complement or reduce costly laboratory analysis. Problems occur if the system dynamics change over time (concept drift). We survey different approaches to handle concept drift and to ensure good prognosis quality over long time ranges. Two main approaches - data accumulation and ensemble learning - are explained and implemented. We compare the concepts on artificial datasets and on industrial data from three cement production plants and analyse strengths and weaknesses of different approaches.

Keywords: Neural Network, Concept Drift, Incremental Learning, Long Term Learning

1 Introduction

Artificial Neural Networks (ANN) for function approximation can be used for modern process monitoring and process control. One of the most common approaches is to use ANN to predict future values of a measurement [1]. If this prediction is used in combination with a controller, the system is known as Model Predictive Control (MPC).
One of the most important problems with MPC in industrial applications is a slightly changing environment which results in a drift of properties and dynamics of the process. This changing is known as concept drift [2]. Hence, the prognosis of the ANN will worsen the more time has passed since training the ANN.
To counter this worsening, the ANN has to be retrained with new data to adapt to changes of the process. In this paper, we compare different methods to retrain neural networks (in particular Multi-Layer Perceptrons) with new data. We are especially interested in long term effects of different retraining approaches. We compare all methods on artificial data and on data obtained from several industrial cement production plants.
The remainder of this paper is organized as follows: we present a Model Predictive Control scenario and the data we used to benchmark different algorithms in Sect. 2. Accordingly in Sect. 3, we give a brief review of related approaches to adapt Neural Networks for function approximation to new data and compare

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

their applicability to the environment. The algorithms we used are explained in Sect. 4 and a description of the experimental investigations is given in Sect. 5. We conclude with a summary and outlook on possible improvements.

2 Experimental Environment and Prerequisites

Automatic learning and continuous adaptation are required in process control if the dynamics of the controlled system changes over time. Industrial combustion processes as used in coal-fired power plants, waste incineration plants, or cement plants are a good example for such changing dynamics. Carbon black and slag are combustion by-products and coat the furnace walls. Over time, the coating grows and changes properties of the combustion process. An even more challenging issue are the fast changes in raw material qualities.
Cement plants are very challenging but also very promising applications for Model Predictive Control. Today cement usually is produced with rotary kiln plants [3]. The residence time of the raw material in the rotary kiln varies from 35 to 60 minutes and depends on the steepness, length, and rotation speed of the kiln. The quality of the produced cement is determined by laboratory analysis which can take from five minutes (X-ray analysis) up to two hours (chemical analysis). A cement sample is usually taken every two hours. Hence it may take up to four hours until a quality measurement is available for the current process situation. If a neural network can predict and estimate cement quality from continuous measurements, like air temperature, kiln rotation speed, raw meal feed etc. a controller can react much faster to changes of the cement quality. Figure 1 shows the prognosis of one main quality measure, the free lime value, obtained with a Multi-Layer Perceptron.

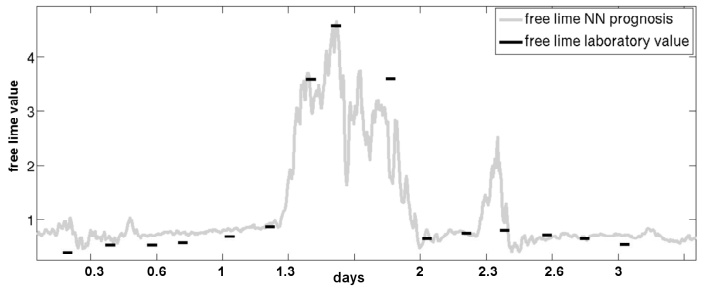

Fig. 1. Free lime value prognosis at a cement plant. The plot shows three days of plant operation with laboratory measurements and neural network prognosis of the free lime value. While a laboratory measurement is only available every four hours, the prognosis is available the whole time.

The free lime value [3] is a major quality criterion of the cement. If a good prediction of this value is possible, the whole production process can be stabi-

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

lized. This prognosis can only be used for control purposes if it is correct. To obtain a correct prognosis over a long time range of two or more years is very demanding as process dynamics of the cement plant change as mentioned afore. Figure 2 shows long term changes of an important process measurement from a cement plant over a period of two years. To test the capability of different long-term network adaptation algorithms, we used data from three cement plants. The target for the approximation was the free lime value of the cement produced by the plant. Network inputs were signals obtained from the process such as kiln rotation speed, kiln temperature, and raw meal feed.

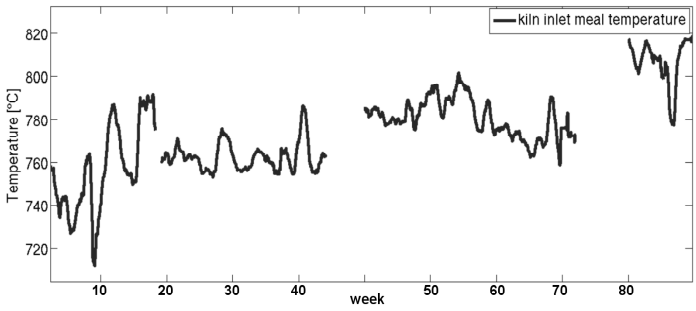

Fig. 2. Time plot of kiln inlet meal temperature over the period of two years. The measurement is low-pass filtered with a sliding time window of one week. Time spans without signal denote stops of the plant. Over the whole time, a change in the level and dynamics of the signal of about 80 degrees is visible.

3 State-of-the-Art

One of the first publications that mentions the difficulties in long term adaptivity of artificial neural networks in changing environments is [4]. The author analysed the problem of catastrophic interference in neural networks. Catastrophic interference describes the phenomenon that learning of new facts disrupts performance on previously learned old facts, however, in [4] the author did not give solutions how the problem could be solved or avoided.
In [5] the author proposes the FLORA framework which accepts only certain samples for training of a classificator. Old samples, that do not suit to the current window are not used for training. A method to weight old and new samples for a two layered network is introduced in [6] by using a forgetting function which reduces influence of old samples in the training process.
Other approaches use not only one approximator but an ensemble. Examples are $Learn^{++}$ [7], dynamic weighted majority [8], incremental adaptive learning [9], or iRGLVQ [10]. All ensemble based approaches are similar in their use of more

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

than one model, where each model is trained with different data. They can be discerned by the basic model used and the way the ensemble members are used to compute an approximation.
Summarizing aforementioned publications, two different classes for handling concept drift can be distinguished:

1. **Data accumulation/Instance selection** [5, 6, 9]: learn a new model with all/selected data acquired; discard the old model
2. **Ensemble learning** [7–10]: learn a new model with new data; select the best model for every situation or merge output of old and new model

While comparing these different approaches two major problems occur: first, every approach uses a different basic approximation model which is adapted to concept drift. In [10] Learning Vector Quantisation (LVQ) is used, in [9, 7] Multi-Layer Perceptrons (MLP), in [6] a combination of two subnetworks, in [5] attribute-value logic, and in [8] an Incremental Tree Inducer (ITI) and Bayes learners. The second problem is that most results are obtained on artificial data or real data with artificially induced concept drift.
In the following, we compare algorithms on industrial data, use the same basic approximator (MLP), and compare results of different adaptation algorithms. For purpose of explanation and to allow for other researchers to reproduce our results, we also use three artificial datasets for benchmarking.

4 Algorithms for Automatic Network Adaptation

In Sec. 3 we showed that there are two paradigms how to adapt a model to concept drift: 1. data accumulation and 2. ensemble learning. In this section we propos algorithms to apply these paradigms to training of Multi-Layer Perceptrons as function approximator for the purpose of Model Predictive Control.
Every approach starts with the same preconditions: there is an initial dataset S_{init} with $s_i \in S_{init}, s_i = (x_1, \ldots, x_m, y)$, where $x_1, \ldots, x_m$ are the input values/measurements for m input dimensions and y is the target value of the respective sample. This dataset is used to train a MLP N_{init} with Levenberg-Marquardt training algorithm.

4.1 Data Accumulation

The principle of data accumulation is to have one model which is adapted when a certain amount of new data S_{new} is available. We applied three variations of this concept:

- **data acc.1:** create a dataset $S_{accu} = S_{init} \cup S_{new}$ and retrain N_{init} with dataset S_{accu}
- **data acc.2:** retrain N_{init} only with dataset S_{new}; ignore old data
- **data acc.3:** split S_{new} in training data S_{new}^{train} and validation data S_{new}^{val} create a new MLP N_{new} and train it with S_{new}^{train}
  if the approximation error of N_{new} on S_{new}^{val} is lower than approximation error of N_{init} on S_{new}^{val} delete N_{init} and use N_{new}

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

Each of the three concepts is repeated every time a new dataset S_{new} is available (the exact number of samples sufficient to create a new dataset depends on the data used). The first concept uses all samples/information acquired over time but may suffer catastrophic interference as described in [4]. The second concept is a very basic version of sliding time window used in [5]. The third concept tries to compensate drawbacks of the first two concepts: a new (retrained) model is only accepted if its results are better than old model results.

4.2 Ensemble Learning

The principle of ensemble learning is to train a new model N_{new} if sufficiently new data S_{new} is available. This results in a pool P of models $N_i \in P, i = 1 \dots n$ where n is the number of currently available models. The important question is which model N_i is activated at a certain time step t? We compare two different ways to determine the active model N_i:

- **ensemble1:** use all $N_i \in P$ to simulate target y for the last d time steps: $\overline{y}^s = (y^s_{t-d}, \dots, y^s_{t-1})$ and compare prognosis error $e = \sum |\overline{y}^s - \overline{y}|$ for all $N_i \in P$; the model with lowest error is chosen
- **ensemble2:** compare current input and training data of each model $N_i \in P$ to choose the best model; therefore:
  - cluster the training data (for example with a k-means clusterer) of each network i into k cluster K and calculate the characteristic input values $\overline{x}^{ik} = (\overline{x}^{ik}_1, \dots, \overline{x}^{ik}_m)$ and the mean prognosis error $\overline{e}^{ik}$ for each cluster centre
  - for every new sample $s_{new} = (x_1, \dots, x_m)$ calculate the Euclidean distance d_{ik} to each cluster centre $\overline{x}^{ik} = (\overline{x}^{ik}_1, \dots, \overline{x}^{ik}_m)$
  - calculate the distance weighted error $e^i = \sum_{c \in K} \frac{\overline{e}_c \cdot d_c}{\sum_j d_j}$ of mean prognosis error for each cluster centre $c \in K$ of each model $N_i \in P$
  - the model N_i with the minimal e^i is chosen and set active

Both approaches are simpler versions of the ensemble methods used in [7–10]. The comparison of input/output-relations is not as easy in Multi Layer Perceptrons as in LVQs or rule based systems. Nevertheless both approaches enable to use adaptive ensembles of MLPs for Model Predictive Control.

5 Experiments

In this section, we apply the adaptation algorithms explained in Sec. 4 to data with concept drift. We use two different types of data. The first three data sets are obtained from rotary kiln cement production plants. The target y for the MLP prognosis is the free lime value which indicates the quality of the cement produced [3]. Five to ten different measurements from each kiln, such as kiln inlet temperature, secondary air temperature, raw meal feed, etc (see [3] for a detailed description of the measurements) are used as input values for a sample $s_i = (x_1, \dots, x_m, y)$. We use two years of data of each plant which results in

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

2,100-4,000 samples, depending on the sample rate of the laboratory (from three up to eight hours) and the revision times of the plants.
For purpose of explanation and to allow for other researchers to reproduce our results, we also use three simple artificial datasets. We generate a target $y(t)$ from five input signals $x_1(t), \ldots, x_5(t)$ as shown in equation 1:

$$y(t) = \alpha \cdot x_1(t) \cdot x_2(t) + x_3(t) + \alpha \cdot x_4(t) + x_1(t) \cdot x_5(t) + d(t) \tag{1}$$

Each input $x_i(t)$ is randomly sampled from a Gaussian distribution of $\mathcal{N}(0, 1)$. We add noise data $d(t)$ sampled from $\mathcal{N}(0, 0.3)$. Linear concept drift is induced by the parameter α which changes over the simulation time. We apply three different variations for the concept drift:

1. α changes linear from 0.1 to 1 and is set back to 0.1 at certain times; this corresponds to slagging in industrial plants which grows over time but is set back to a low level after a plant revision
2. α is linearly changing from 0.1 to 1
3. α does not change, which corresponds to a process without concept drift

Fig. 3 illustrates the progress of parameter α at all three artificial datasets.

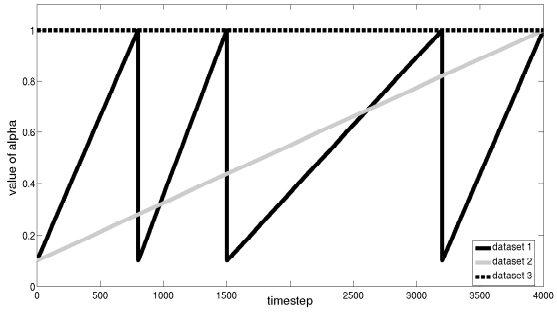

Fig. 3. Progress of parameter α used to model concept drift in three artificial datasets.

For the prognosis of the target, we use a Multi-Layer Perceptron featuring one hidden layer with five neurons to approximate the target. Training algorithm is standard Levenberg-Marquardt training as included in the Neural Network library of Matlab. All networks (except approach *data acc.*1) are trained/retrained with 250 samples of data where the last 50 samples are used for validation. For *data acc.*1, we use a growing training set which includes all samples available since starting retraining. For *ensemble*2, we apply a k-means clusterer with $k = 10$ to cluster the training set of each network. The test performed in *ensemble*1 to determine the best model is carried out with the last 50 samples observed. No additional pruning algorithms are used as we are focused

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

$\overline{e}_{Q50\%}$	plant1	plant2	plant3	$\sum$plant	art.data1	art.data2	art.data3	$\sum$art.
no adapt.	0.635	0.871	0.782	2.288	0.309	0.334	0.135	0.778
data acc.1	0.492	0.766	0.714	1.972	0.220	0.217	0.124	0.561
data acc.2	0.701	0.773	0.768	2.242	0.201	0.156	0.124	0.481
data acc.3	0.520	0.801	0.779	2.100	0.249	0.167	0.134	0.550
ensemble1	0.478	0.795	0.749	2.022	0.193	0.168	0.134	0.495
ensemble2	0.524	0.850	0.793	2.167	0.306	0.275	0.185	0.766
revisions	3	4	0		3	0	0	

Table 1. Median prognosis error $\overline{e}_{Q50\%}$ of 200 trials network training. The two best results of each data set are marked with a grey background. The number of plant revision (resets of the concept drift) and the sum $\sum$ of errors over all datasets are also listed.

on effects the different adaptation algorithms have on long term prognosis error. Table 1 shows results on the different datasets.

For evaluation of the results, we repeated every simulation 200 times. The mean prognosis error over the whole time period was calculated. Afterwards, we compared the median $\overline{e}_{Q50\%}$ of all 200 trials for each concept and data set. We chose the median and not the mean because approximately 1% of the networks trained produces a very high error because of disadvantageous initialisation which influences the mean error of all 200 simulations disproportionately.
Prognosis without adaptation of the network produces the worst result. This was expected as it does not counter the concept drift. *ensemble*2 also performs very bad. This is a result of the imprecise representation of the input space we choose with the k-means clustering. If a better method is aquired to map and compare input/output relations in trained MLPs, this approach would surely produce better results. The potential of ensembles is revealed by *ensemble*1, which is the second best method of the six approaches we tested. Only if the concept does not change (*art.data*3) or there is no revision of the plant included in the data (*art.data*2, *plant*3), data accumulation approaches outperform this ensemble approach.
Of the three different data accumulation approaches *data acc.*1 performs best on real world data. This is surprising, since *data acc.*1 uses all data available, which results in ambiguous data due to the changing parameters (boiler slagging in plants and α in artificial data). Nonetheless the prediction acquired with unambiguous data but fewer training samples is worse. We expect the results of *data acc.*2/3 to get better if the sampling rate is increased and more samples are available for the used time window.
On *plant*3 the differences between the approaches are smaller than on other plants. The reason is that in *plant*3 other sensor measurements than in *plant*1 and *plant*2 had to be used because of the plant architecture. Hence the overall prognosis quality decreases and differences between the adaptation concepts disappear.

in: Proc. 22. Int. Conf. on Artificial Neural Networks (ICANN 2012), Lausanne, Switzerland, Part II, LNCS 7553, pp. 50-57, Springer 2012

6 Conclusion

Concept drift does influence the quality of neural network prognosis in industrial combustion processes. Through growing boiler slagging and the use of other fuels, the prognosis of important performance figures is getting worse if the networks used are not adapted to changing data. We applied different approaches to adapt networks to concept drift over long time ranges. The best approach depends on the type of the concept drift. If dynamics and properties of the plant change very slowly and old states do not appear again, it is advantageous to use sliding window technic and data accumulation to constantly retrain a single network with new data.
If changes in the dynamics and properties appear very abrupt and old states reappear (due to revisions of a plant or a small selection of used fuels) ensemble learning with more than one model is superior to other concepts.
Our future work concentrates on improving the use of ensemble methods. Furthermore we want to apply the approaches to other industrial MPC problems and compare them with the results gained on cement plants.

References

1. Agachi, P.S., Nagy, Z.K., Cristea, M.V., Imre-Lucaci, A.: Model based control: Case Studies in Process Engineering. Wiley-VCH (2006)
2. Tsymbal, A.: The problem of concept drift: Definitions and related work. Technical Report, Department of Computer Science, Trinity College: Dublin, Ireland (2004)
3. Alsop, P.A.: The Cement Plant Operations Handbook. Tradeship Publications Ltd., 5th ed. (2007)
4. McCloskey, M., Cohen, N.: Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation, 24, pp. 109–164 (1989)
5. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden context. Machine Learning 23(1), pp. 69–101 (1996)
6. Pérez-Sánchez, B., Fontenla-Romero, O., Guijarro-Berdiñas, B.: An incremental learning method for neural networks in adaptive environments. Int. Joint Conf. on Neural Networks (IJCNN) 2010, pp. 1–8, (2010)
7. Elwell, R., Polikar, R.: Incremental Learning of Concept Drift in Nonstationary Environments. IEEE Transactions on Neural Networks 22(10), pp. 1517–1531 (2011)
8. Kolter, J.Z. ,Maloof, M.A.: Dynamic weighted majority: A new ensemble method for tracking concept drift. In Proc. IEEE Int. Conf. on Data Mining (ICDM2003), pp. 123–130 (2003)
9. He, H.: Self-Adaptive Systems for Machine Intelligence. John Wiley & Sons (2011)
10. Kirstein, S., Wersing, H., Gross, H.-M., Koerner, E.: A life-long learning vector quantization approach for interactive learning of multiple categories. Neural Networks 28, pp. 90–105 (2012)