ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Effects of Noise-Reduction on Neural Function Approximation

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

1- Ilmenau University of Technology
Neuroinformatics and Cognitive Robotics Lab
P.O.Box 100565, D-98684 Ilmenau, Germany

2- Powitec Intelligent Technologies GmbH
45219 Essen-Kettwig, Germany

Abstract. Noise disturbance in training data prevents a good approximation of a function by neural networks. To achieve better approximation results we combine neural networks with noise reduction algorithms. We compare different methods to distinguish between samples with high noise level (outliers) in a dataset and samples with low noise level. Drawbacks of common outlier detection approaches are analysed and a new approach is defined which increases the quality of network function approximation. We demonstrate the effects of noise reduction on artificial datasets and on real data from the process control domain.

1 Introduction

In the context of signal processing, noise describes a disturbance superimposed on a measurement signal. Noise occurs in all scopes of applications were a measuring device is used to determine a value. The measured value does not equal the actual value due to inaccuracies of the measuring device or influences of the environment. This inaccuracy is known as noise [1].
Since the value of noise is random, it can not be approximated. The mean error of any function approximator on a noisy dataset can therefore never be less than the average noise on the dataset [2]. Apart from this error bound, noise on the training data also reduces the ability of data-driven function approximators to learn a function correctly. In Table 1, we show the influence of noise on approximation quality of a neural network. Training algorithm and approximated function are explained in Section 4.

σ_{target} / σ_{noise}	1.0 / **0**	1.0 / **0.05**	1.0 / **0.2**	1.0 / **0.5**	1.0 /**1.0**
$\overline{MSE}_{Q50}$	0.0002	0.0005	0.0039	0.0198	0.0630

Table 1: Effect of increasing training data noise on the approximation of noiseless test data. σ_{target} denotes the standard deviation of the target signal. σ_{noise} denotes the standard deviation of the $\mathcal{N}(0, \sigma^2_{noise})$ distributed noise added to the target. Every test was repeated 100 times. $\overline{MSE}_{Q50}$ is the median of the mean squared error of all 100 network approximations.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

The approximation on **noiseless** test data is getting worse, the more intense the noise on the training data is. So, noise in the data significantly worsens the capability of a neural network to approximate a target function. If a very good approximation of a target is required, it is therefore advantageous if the training data has a low noise level.
In the following, we compare different approaches to reduce the negative effects of noisy training data on neural network function approximation. We present a new approach to detect samples with the largest noise level and to reduce the average noise on a dataset. We show the improvement of the target approximation on artificial data and on real data from an industrial cement production process.

2 State of the Art

Different approaches exist to separate information from noise. Most concentrate on certain types of information or noise, like audio noise or noise in images. A comprehensive overview of common approaches to detect noise for special applications is given in [1]. If the characteristics of the signal are not known more general approaches are needed. In [2] the Gamma-Test was introduced which approximates the average noise level of a dataset and can distinguish noisy input channels. But still the algorithm is not able to estimate the noise level of a single data sample and hence to select samples for a network training.
Data samples with a high noise level are commonly referred to as *outliers*. In [3] several approaches to detect outliers are compared. The authors argue that most common statistical approaches as explained in [4] or [1] are not useful for real datasets because they require the assumption of a specific data distribution. Instead the best approaches tested in [3] use local neighbourhoods in combination with global data attributes. The best approach tested in [3] was the *Local outlier factor* (LOF) from [5].
A Support Vector Data Description (SVDD) was proposed in [6] and improved in [7] to separate outliers and normal data. However the SVDD is a pure classification approach with best results obtained if two classes existed to train the SVDD: one with known outliers and one with normal data. This precondition does not apply to industrial data we use for our experiments.
In [4] and [8] both authors assert that outliers (especially gross outliers) worsen the results of common regression approaches, but do not give examples for training with neural networks. Bishop [9] argues that outliers have a negative effect on network trainings mainly because the *Mean Squared Error* (MSE) used to train networks reinforces the negative effects of outliers. [10] agrees and proposes the *Least Mean Log Squares* (LMLS) to train networks that are robust to outliers. Unfortunately this approach also reduces the training effects of samples which are not outliers but have a large initial network approximation error.
A more general approach to detect samples which support a good approximation is the RANSAC algorithm [12].
In the following, we compare the above mentioned approaches with regard to their capability to achieve better approximation results with neural networks.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

3 Noise Reduction Algorithm

We implemented the LOF- [5], LMLS- [10], and RANSAC- [12] approach as explained in the respective paper. The purpose of the algorithms is to determine a value for the noise intensity of every sample of a dataset and weight each sample in the network training depending on its noise value.
Since most approaches have difficulties to distinguish between noisy samples and samples with uncommon input values, we also defined a new approach to determine the noise intensity of a sample through the use of Local Linear Models [11] at the neighbourhood of each sample. In the following we call this approach NELLM (*Noise Estimation with Local Linear Models*).
Precondition for the application of the algorithm is a dataset with n sample points. Each sample is a tuple $s_i = (x_1, \ldots, x_m, y)$, where $x_1, \ldots, x_m$ are the input values for m input dimensions and y is the target value of the respective sample. Our new algorithm works as follows:

1. for all $s_i \in S$ where $s_i = (x_1, \ldots, x_m, y)$, $i = 1 \ldots n$ do:
   1.1. find the k nearest neighbours $\{s_j\}$ to s_i with euclidian distance in the inputspace $d_{ij} = \|s_i - s_j\|_2$ for $s_j \in S, s_i \neq s_j$ and group them in N_i
   1.2. determine a linear regression model $y_j^p = p_1 x_{1j} + p_2 x_{2j} + \ldots + p_m x_{mj} + p_{m+1}$ for the targets $\{y_j\}$ with $s_j \in N_i$
   1.3. calculate regression error $e_i = |y_i - y_i^p|$ for the current sample s_i
2. train a network with S but weight each $s_i \in S$ influence on the training depending on e_i; weighting can be done in two ways:
   2.(a) every sample s_i with $e_i > e_{limit}$ is deleted from S
   2.(b) a new training dataset $\tilde{S}$ is generated; samples s_i with $e_i < e_{limit}$ are added to $\tilde{S}$ more than once; we use batch-training to train the networks hence the number of appearances of a sample in a dataset influences its impact on the training process

The method described in 2.(a) reduces the noise more effective than 2.(b), however samples are deleted from the dataset. In cases were only few samples exist this reduces the available information more than the noise on the data restrains the learning (see also [4] for the dilemma of deleting noisy samples). Therefore we use method 2.(b) in our approach.

4 Experiments

In Section 1 we showed the negative effects of increasing noise at the approximation capability of a neural network. The target function $y(t)$, which should be approximated, was calculated from three input signals $x_1(t)$, $x_2(t)$ and $x_3(t)$ and normal distributed noise $d(t)$ with the formula:

$$y(t) = \alpha \cdot (x_1(t) \cdot x_2(t) + sin(x_3(t)) \cdot x_1(t)) + d(t) \quad (1)$$

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

$x_1(t)$, $x_2(t)$ and $x_3(t)$ have a Gaussian distribution of $\mathcal{N}(0.5, 0.03)$. The target function $y(t)$ (without noise) is normalized with the parameter α to a distribution of $\mathcal{N}(0, 1)$. We created two dataset of 300 samples based on formula (1). One set was used to train the network, the second one to test the accuracy of the trained approximator. We used a Multi-Layer-Perceptron with 5 hidden neurons which was trained using the Levenberg-Marquardt training algorithm.
We applied different Noise-Reduction algorithms as mentioned in Section 2 on the training dataset to test the ability of the algorithms to reduce negative effects of noise on the network training. The results are shown in Table 2. For the *LOF*-approach [5] we got the best results with a local neighbourhood of 15 samples. The *NELLM*-algorithm was applied with a neighbourhood of 60 samples. We set e_{limit} to be higher than 15% of the train data and weighted the samples below e_{limit} 10 times as much as the other samples.
For the RANSAC algorithm [12] we used 50 iterations per test. The results got better the more iterations RANSAC used, but it also required much more computation time. The *LMLS*-training method from [10] needed no extra parameters. To get an impression of the neural network approximation quality we also used a linear regression model to approximate target values and compared it with the neural network.

$\sigma_{target}/\sigma_{noise}$	1.0/**0**	1.0/**0.05**	1.0/**0.2**	1.0/**0.5**	1.0/**1.0**
$\overline{MSE}_{Q50}$ normal	0.0002	0.0005	0.0039	0.0198	0.0630
$\overline{MSE}_{Q50}$ NELLM	0.0003	0.0012	0.0045	0.0156	0.0455
$\overline{MSE}_{Q50}$ LOF	0.0003	0.0016	0.0187	0.0650	0.1480
$\overline{MSE}_{Q50}$ RANSAC	$1 \cdot 10^{-5}$	0.0011	0.0112	0.0527	0.1695
$\overline{MSE}_{Q50}$ LMLS	0.0002	0.0005	0.0040	0.0203	0.0613
$\overline{MSE}_{Q50}$ Regression	0.0392	0.0379	0.0384	0.0422	0.0512

Table 2: Approximation results for data with different noise levels. σ_{target} denotes the standard deviation of the target signal (without noise). σ_{noise} denotes the standard deviation of $\mathcal{N}(0, \sigma^2_{noise})$ distributed noise added to the target. The test data was always noiseless. Every test was repeated 100 times. *normal* denotes experiments without any noise reduction algorithm applied. Displayed are the median approximation errors $\overline{MSE}_{Q50}$. We evaluated the median and not the mean because a small percentage ($< 1\%$) of networks produced very high errors which influence the mean error.

With a noise of $\sigma_{noise} = 0$ and $\sigma_{noise} = 0.05$ all approaches produced very low errors nearly equal to the results achieved without applying any noise reduction algorithm. With noise levels of $\sigma_{noise} = 0.2$, $\sigma_{noise} = 0.5$ and $\sigma_{noise} = 1.0$ the approximation quality decreased. The linear regression approach produced the worst results with low noise levels but the negative influence of high noise is not as high as on neural networks. Only the *NELLM*-algorithm reduced the error compared to training without noise reduction. This result is surprising since all algorithms are designed to reduce noise and hence should improve the

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

approximation result. Thus we analysed the weighted dataset produced by the LOF-, NELLM- and RANSAC-algorithm and compared the noise of the samples with the original dataset in the case of very high noise with a $\sigma_{noise} = 1.0$. The result is shown in Table 3.

Dataset	original	LOF	NELLM	RANSAC
σ_{noise}	1.0	0.65	0.20	0.97
$\overline{MSE}_{Q50}$	0.0630	0.1480	0.0455	0.1695
σ_x	0.523	0.3347	0.518	0.524

Table 3: Noise-Level σ_{noise}, median approximation error $\overline{MSE}_{Q50}$ and input standard deviation σ_x of dataset produced by noise reduction algorithms.

The RANSAC algorithm only slightly reduces the noise. The LOF approach reduces the noise by nearly 35% but nonetheless does not reduce the approximation error. The reason is that the LMLS- and LOF-algorithm prefer data samples with a low standard deviation in the input space and classifie samples as outliers which are far away from the mean value of the data, regardless if they are true outliers or seldom observed inputs. The NELLM-algorithm reduces the noise by nearly 80% and does not reduce the standard deviation of the chosen samples

To test the capability of the different algorithms at an application with a real world dataset, we used data from three cement plants. The target for the approximation was the free lime value of the cement produced by the plant. The free lime value [13] is a major criterion for the quality of the cement. If a good prediction of this value is possible, the whole production process can be stabilized. Network inputs were signals obtained from the process such as kiln rotation speed, kiln temperature and raw meal feed. The resulting errors for the approximation of the free lime value are shown in Table 4.

	noise level	approximation errors					
		normal	LOF	NELLM	RANSAC	LMLS	Regression
plant A	0.145	0.594	0.587	0.572	0.599	0.579	0.600
plant B	0.583	0.699	0.665	0.640	0.679	0.674	0.696
plant C	0.437	0.751	0.718	0.704	0.691	0.727	0.550

Table 4: Noise levels and median approximation results for the free lime values of three cement plants. *normal* denotes experiments without any noise reduction algorithm applied. Every test was repeated 100 times. The noise of the free lime value measurement was determined through repeated measure of the same free lime sample

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

The NELLM-approach still produces the best approximation results but the improvement of 3.7%, 8.4% and 6.2% compared to trainings without noise reduction is not as high as at the artificial dataset. The main reason is that the test dataset of the artificial data was noiseless while the test dataset for the cement plants has the same noise as the training dataset. Hence an optimal approximation of the test dataset is not possible. Nonetheless all noise reduction algorithms enable better approximation results for neural networks.

5 Conclusion

We compared several noise reduction and outlier detection algorithms with special regard to their effects on neural network training for regression tasks. We applied approaches to detect outliers and reduced the weight of outliers in the training dataset. We observed that most outlier detection approaches marked seldom observed samples as outliers. Since such samples are important for network training we designed a new approach called NELLM (Noise Estimation with Local Linear Models) which improves the differentiation between real outliers and uncommon samples. A network trained with the weighted dataset achieved a much better result than with the original noisy data. We compared the approaches on artificial data and data obtained from real cement plants. The cement results were not as good as results on artificial data but still showed an improvement compared to training without noise reduction.

References

[1] Vaesghi, S. V., *Advanced Digital Signal Processing and Noise Reduction*, John Wiley & Sons, 4th ed., 2008

[2] Jones, A. J., Evans, D., Kemp, S. E., *A note on the Gamma test analysis of noisy input/output data and noisy time series*. In Physica D 229, 1, pp. 1–8, 2007

[3] Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A., *Interpreting and Unifying Outlier Scores*. In Proc. SIAM Int. Conf. on Data Mining, SDM 2011, pp. 13–24, 2011

[4] Barnett, V., Lewis, T., *Outliers in Statistical Data*, John Wiley & Sons, 3rd ed., 1994

[5] Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., *LOF: Identifying Density-Based Local Outliers*. In Proc. SIGMOD Int. Conf. on Management of Data, pp. 93–104, 2000

[6] Tax, D.M.J., Duin, R.P.W., *Support Vector Data Description*, In Mach. Learn., vol. 54, pp. 45–66, 2004.

[7] Lee, K.Y., Kim, D.-W., Lee, K.H., Lee. D., *Density-Induced Support Vector Data Description*. In IEEE Transactions on Neural Networks, vol. 18, Issue 1, pp. 284–289, 2007

[8] Huber, P.J., Ronchetti, E.M., *Robust Statistics*, John Wiley & Sons, 2nd ed., 2009

[9] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford Univ. Press, 1st ed., 1995

[10] Liano, K. *Robust error measure for supervised neural network learning with outliers*. In IEEE Transactions on Neural Networks, vol. 7, Issue 1, pp. 246–250, 1996

[11] Walter, J., Ritter, H., Schulten, K., *Non-linear prediction with self-organizing map*. In Proc. IEEE IJCNN1990, Vol. 1, pp. 587–592, 1990

[12] Fischler, M.A., Bolles, R.C., *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Comm. of the ACM 24, pp. 381–395, 1981

[13] Alsop, P.A., *The Cement Plant Operations Handbook*, Tradeship Pub. Ltd., 5th ed., 2007