Windowed Active Sampling for Reliable Neural Learning.

E. I. Barakova and L. Spaanenburg
Groningen University, Dept. of Computing Science,
P.O.Box 800, 9700 AV Groningen, The Netherlands.

Abstract

The composition of the example set has a major impact on the quality of neural learning. The popular approach is focused on extensive preprocessing to bridge the representation gap between process measurement and neural presentation. In contrast, windowed active sampling attempts to solve these problems in an on–line interaction between problem selection and learning. This paper provides an unified view on the conflicts that may pop–up within a neural network in the presence of ill–ordered data. It is marked that such conflicts become noticeable from the operational learning characteristics. An adaptive operational strategy is proposed that closes the representation gap and its working is illustrated in the diagnosis of power generators.

Keywords: *neural networks, backpropagation, active sampling, reliability, symmetry.*

1: Introduction.

The reliability of neural learning is not a widely discussed topic. Instead, particular failures to learn have been researched for decades [1], [4] and led to countermeasures in learning algorithm, network topology, network initialization, and training set construction. The problem dependent nature of these work–arounds defies a general–purpose learning strategy and still requires an in–depth analysis into the cause and nature of various bad learning phenomena.

This paper focuses on the low reliability of learning behavior, as caused by restrictions on the back–propagation algorithm in combination with a specific training set structure. As a result of the presentation of training sets, whose elements have the potential to provoke long–term changes in the network state with comparable but opposite impact, the learning does not have a guaranteed convergence. Random equidistant sampling of some signals as well as some selective sampling strategies can make this restriction ostensible, which because of its nature we call cancelation. An extreme case of this phenomenon, when convergence never occurs, is often noticed in situations, where realization, topology and target signal fully satisfy the symmetrical conditions under which error backpropagation breaks down [12].

Our contribution is, that we show how the cancelation phenomena, generally known as an artifact and unlikely to occur in practice [12], affects real–life situations. Most of the time, its effect will be that the learning time becomes not reproducible or that the final approximation is less accurate. The analysis made in the paper shows, that the prolonged learning time indicates potential failure of the number of consequent learning trials. A windowed sampling strategy will be suggested that guarantees high–quality results in the presence of cancelation conditions, as inobtrusively satisfied in actual measurements.

First, in section 2, we review the techniques and notions in example set construction. Then, in section 3, two typical applications visualizing the major occurrences of cancelation phenomena are shown. Later on, after analyzing the reasons which can bring the training algorithm to convergence problems and giving a brief framework of example set construction, we are applying in section 5 an windowed sampling algorithm that eliminates the cancelation effect. The results of applying this algorithm on real–life data, taken from the emergency working mode of a power generator, are shown in section 6. A discussion on the obtained results and their potential application is presented as an open–end for future work.

2: Example set construction: Arbitrary or Selective.

A substantial problem in neural computing is that except for the trivial problems they fail on a small percentage of subsequent tests. There exists a number of systematic exceptions in which neural networks always generalize in a wrong way. The performance of a trained network is dependent a lot on the function, that the network should map and thus on the training set used. Creating the training set which will ensure optimal functionality of the learning algorithm has many aspects. Overall it can be defined as finding the optimal ratio between the number of training samples and their distribution over the signal to learn.

In the absence of any tangible rules relating the signal to be learned and the nature of training patterns, presenting equidistant samples in a random way gives in most cases a satisfactory result. One of the main advantages of random selection is its easy implementation. Moreover, the random factor is crucial for the work of all learning algorithms of a stochastic nature. The creators of the backpropagation algorithm have suggested in [16] that not only the network parameters but also the training examples should be chosen randomly. The reason is that error back–propagation stresses the differences over the various paths through the network. When these differences are not present, clearly nothing can be stressed. In contrary, the more factors able to break the symmetry in the network are present, the higher the chance on success for the learning process. Some authors [9] suggest even adding noise on the inputs for a better learning performance.

These are the reasons why most neural networks are studied by means of random sampling. Random sampling assumes that training examples are arbitrarily chosen, and that the network learning process evolves under its own dynamics. In this case, it can be said that the neural network is a passive learner. This approach is generally referred to as “Learning from examples”. Analytical examination of this problem for neural networks is done by Baum and Haussler [5], the generalization properties are empirically investigated by Cohn and Tesauro [6], while Le Cun [13] attempts further improvements. A complete description of this approach for analyzing neural networks generalization is given by Poggio *et al.* [8].

In the context of the passive learning framework, the ordered presentation of examples presumes that also the relation between subsequent presentations is of importance. This happens for instance when the natural order of presentation must be preserved (as in time–series prediction tasks) or when this is the only possible way of obtaining the examples. Often the ordering is combined with a degree of filtering to remove spurious detail. As reported by Morgan and Boulard [14], one can produce results by using only a small fraction of the available examples, that are close to those obtained when all available data are used. Another reason for using a specific strategy of pattern selection is that network performance can be drastically improved (Cloete and Ludik [7]).

This latter result, together with [2], [15], etc. establishes neural active learning as an alternative for passive learning from random examples. Active learning presumes some control over the way of selecting training examples. Active algorithms applied to neural networks aim to assure success of the neural learning process by optimizing the information coming from the environment. They are either oriented towards the strategy of pattern presentation or to the selection of the best training set. Accordingly, there are two distinct groups of techniques for choosing training examples. The first group assumes that the network is partially trained on a set of previously acquired examples. This group of techniques is known as *active* *sampling* or *progressive learning* and can be defined as the task of adding new examples to the set of available examples. The second group of active learning techniques is known as *active selection* or *informative learning* and implies selection of training exemplars from the set of available examples. Properly selected, these actions can drastically reduce the amount of data and computation time required for learning to be completed. Fig. 2 summarizes the above discussion.

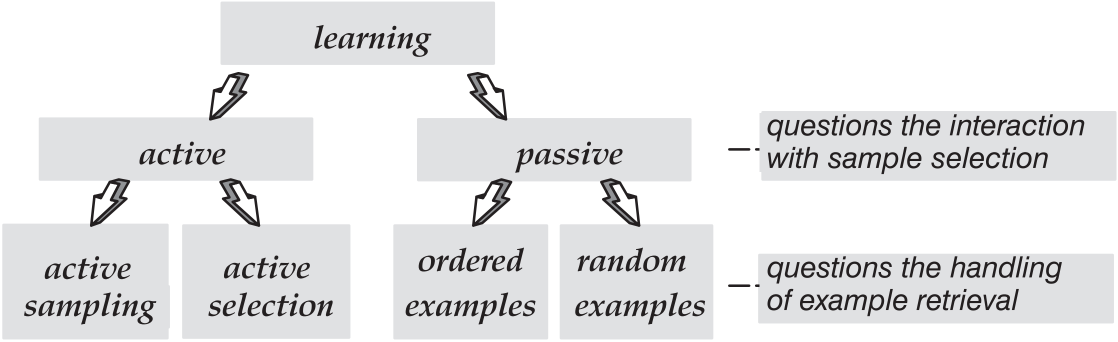

learning
active
passive
questions the interaction with sample selection
active sampling
active selection
ordered examples
random examples
questions the handling of example retrieval

Fig. 1: Classification of the learning process with respect to example presentation.

This paper suggests to base active selection on the detection and elimination of notoriously bad learning conditions. It has its counterpart in the preprocessing of measured data by filtering and ordering as applied successfully in passive ordered learning. Though this has the advantage of using pre–knowledge, its disadvantage of a potentially extensive preprocessing stage precludes the potential usage in real–time applications.

With active selection we rather aim to improve the learning quality by on–line adaptation. By a rigorous windowing of the available data, it extracts a behavioral hierarchy that by rapid application is guaranteed to have the proper representation. Our analysis of the reasons for bad convergence suggests several possibilities for active selection. The choice we made tends to preserve the advantages of random pattern presentation in the local window range, while the size and position of the windows are determined by active selection strategy. This allows for an easy introduction in current practice.

3: Examples of Cancelation Training Sets.

The cancelation training set, that visualizes a fundamental drawback of the back–propagation algorithm, can be constructed in many ways. The following examples will give a more clear indication how and why cancelation can appear in practice. We focus largely on illustrating that the cancelation can be easily introduced in a number of ways, ranging from the problem definition to the applied intermediate representation. In a later section we will provide some analytical considerations, by which one can check whether a problem can suffer from cancelation.

3.1 Sampled Symmetry.

Although it is proven in [10] that feedforward networks with one hidden layer can approximate an arbitrary function, practice shows a number of systematic hindrances to achieve this goal. One example for monitoring potential convergence problems in training backpropagation networks is the approximation of a signal, when the selected training set is either near to or fully symmetrical. As shown in Fig. 2 a symmetrical target set can be extracted from a large class of functions. Thus, sampled symmetry sets can be created by choosing not–equally spaced contradictory patterns. We have created such training sets artificially (Fig. 2a,b), but a number of pattern selection methods or practical sampling recommendations, efficient otherwise, can also end up with creating a cancelation example set (Fig. 2c).

Handwritten Word Recognition is a typical example of when sampled symmetry can creep in through the choice in intermediate representation. Handwritten Word Recognition, also called Isolated Handwritten Word Recognition (HWWR), deals with the problem of machine reading of handwritten words, generally with the assistance of a lexicon of all valid words. A handwritten word is typically scanned in from a paper document and made available in the form of a binary or grayscale image to the recognition algorithm for Off–line HWWR. The problem differs from On–line HWWR where the writing surface is frequently an electronic notepad or a tablet, and where temporal information (the trajectory of the pen as it traces the word) is available to the recognition algorithm, which attempts to recognize the writing as it is being written.

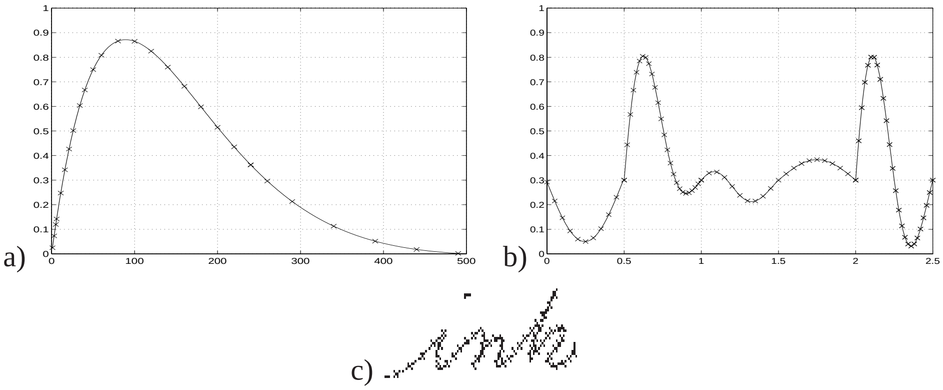


Fig. 2: Functions from which a symmetrical training set can be extracted.

An often used approach takes the handwritten word as an on–line signal and stores it as a sequence of “strokes”: line segments between two sign changes in the writing direction. Each stroke is characterized by a 5–tuple: the starting point, three equally spaced intermediate samples and the end point. Ensembles of strokes can be identified as characters, which in turn can be assembled to words of fuzzy segmentation. Our critique here is on the normalizing storage of strokes by 5–tuples, that blurs away the differences in angular writing. As a consequence, the first character of the word in Fig. 2c will be found from an upgoing and a downgoing line on different angles, but with a symmetrical representation on stroke level.

So far we have focussed on a symmetry that is directly visible in the training set. Even when the symmetry is not ostensible a lack in reliable training performance may easily appear, as it will be discussed further.

3.2 General Cancelation.

After we have reached a basic understanding of the internal mechanisms that cause the cancelation phenomenon, predicting the range of signals, that can cause bad approximation and too long learning time is an easy task. Not only a sampled symmetry set makes the gradient algorithm to oscillate into stationary areas of the error surface, but also training sets as shown in Fig. 3. In Fig. 3a the entire signal is subject to cancelation by symmetry; moreover the phenomenon affects also sub–training sets (Fig. 3b,c), where only the global tendency or other dominant input feature of the signal can be approximated.

A general cancelation signal can be constructed by choosing at random, equally spaced patterns of a periodic signal, that is symmetrical in itself or contains additive symmetrical components. Sometimes the symmetrical component is not obvious, because of additive noise with a normal distribution (Fig. 3a,b). Moreover, signals as the one shown at Fig. 3c also shows cancelation nature. The signal, shown in Fig. 3c is recorded during the emergence working mode of a power generator. It contains a large percentage of cancelation examples and its approximation usually fails when a random equidistant sampling is done on it. It is a typical example of a general cancelation signal, as it will be shown further on.

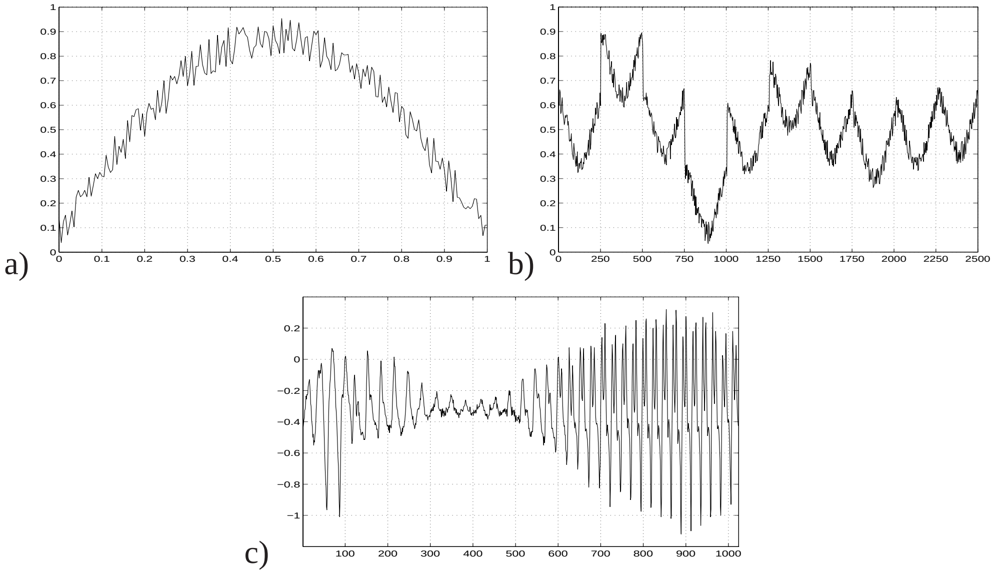


Fig. 3: Examples of general cancelation signals.

4: Analysis.

Analytically, the properties of a generalized cancelation set can be understood by the following reasoning. The single output network with one hidden layer is equivalent to the nested sigmoidal scheme as shown at eq. (1):

$$f_j(x, w) = \varphi(\sum_i w_{ji}\varphi(\sum_k w_{ik}x_k)). \tag{1}$$

After one full presentation of the entire training set D_n, output $f(x, w)^1$ depends on D_n and the development of a learning process, i.e. from the previous weight values. The generalized delta rule for updating the weights $w_{ji}(n)$ is:

$$\varDelta w_{ji}(n) = \alpha\varDelta w_{ji}(n-1) + \eta\delta_j(n)y_i(n) \tag{2}$$

In order to see the effect of the sequence of patterns on the synaptic weights it is useful to represent eq.(3) as a time series with index t (Jacobs [11]).

$$\varDelta w_{ji}(n) = \eta\sum_{t=0}^{n}\alpha^{n-t}\delta_j(t)y_i(t). \tag{3}$$

The equality of the product $\delta_j(t)y_i(t)$ to $-\partial\mathcal{E}(t)/\partial w_{ji}(t)$ can be seen from the derivation of the backpropagation algorithm. Then the equation (3) can be rewritten in the following way:

$$\varDelta w_{ji}(n) = -\eta\sum_{t=0}^{n}\alpha^{n-t}\frac{\partial\mathcal{E}(t)}{\partial w_{ji}(t)}. \tag{4}$$

Here, $\varDelta w_{ji}(n)$ is an exponentially weighted sum. When subsequent partial derivatives $\partial\mathcal{E}(t)/\partial w_{ji}(t)$ have the same sign $\varDelta w_{ji}(n)$ grows in magnitude, thus weights are adjusted by

[1] The index j in equation (1) is a notation for the j – th neuron from the output layer. Indexing the only neuron of the output layer we consider as not necessary and further on it will be used only if necessary.

a large amount. When the partial derivatives $\partial\mathcal{E}(t)/\partial w_{ji}(t)$ have opposite signs on consequent iterations, $\Delta w_{ji}(n)$ shrinks in magnitude, which presumes a small adjustment of the weight values. Random equidistant sampling of some signals presumes altering of the sign of $\partial\mathcal{E}(t)/\partial w_{ji}(t)$ at a very small intervals, and correspondingly almost smooth shrink of the borders of the weight changes. To make more clear this statement let us look in detail at the derivative $\partial\mathcal{E}(t)/\partial w_{ji}(t)$.

$$\frac{\partial\mathcal{E}(t)}{\partial w_{ji}(t)} = \frac{\partial\mathcal{E}(t)\partial e_j(t)\partial y_j(t)\partial v_j(t)}{\partial e_j(t)\partial y_j(t)\partial v_j(t)\partial w_{ji}(t)}. \qquad (5)$$

This equation represents in fact the dependence of the network state change from the current value of the error ($\partial\mathcal{E}(t)/\partial e(t) = e(t) = d - y$) , the derivative of the network output to its input or indirectly from the network input $\partial y(t)/\partial v(t)$ and the output of the neuron i from the hidden layer ($\partial v(t)/\partial w_{ji}(t) = y_i(t)$). The term $\partial e(t)/\partial y(t)$ contributes only with a negative sign.

The derivative of the network output to its input $\partial y(t)/\partial v(t) = \varphi'(v)$ has always a small positive value, as can be seen at Fig. 4 for both: nonsymmetrical and zero–centered sigmoid.

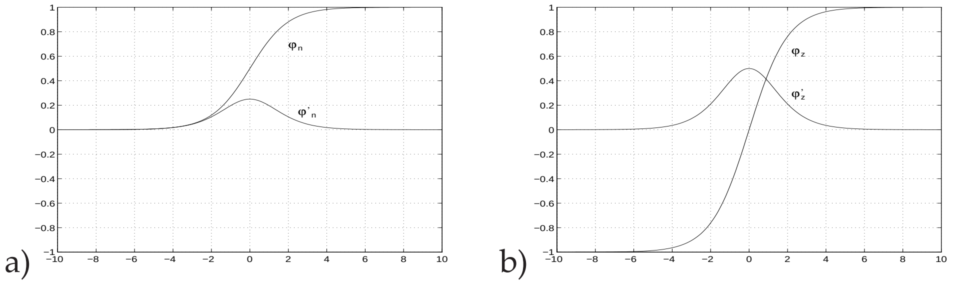


Fig. 4: a) Nonsymmetrical sigmoid $\varphi_n(v) = 1/(1 + e^{-v})$ **and its first derivative** $\varphi_n{}'(v) = \varphi_n(v)[1 - \varphi_n(v)]$**. b) zero–centered sigmoid** $\varphi_z(v) = (1 - e^{-v})/(1 + e^{-v})$ **and its first derivative** $\varphi_z{}'(v) = [1 - \varphi_z^2(v)]/2$**.**

The values of $\partial v(t)/\partial w_{ji}(t) = y_i(t)$ and $\partial\mathcal{E}(t)/\partial e(t) = e(t) = d - y$ are going to be analysed in combination. When consider the 1–N–1 architecture, the hidden neuron outputs are in fact scaled values of the input examples. The hidden neuron output $y_i(t)$ is always positive in case of a nonsymmetrical sigmoid. This way the only component, able to change the sign of the weight value is the calculated error $e(t) = d(t) - y(t)$. When the network is constructed from neurons with zero–centered transfer, $y_i(t)$ changes its sign either because an input example with a different sign was introduced or (quite rarely in fact) because of altering the sign of its weight to the input neuron. Both changes can not happen only to one hidden output, but to all the hidden neurons at once, which will provoke the corresponding change in the networks output, relevant in our case with the calculation of $e(t) = d(t) - y(t)$.

In the both cases have to be looked at the error value $e(t) = d(t) - y(t)$. In the beginning of the training process only a small area in the middle of the activation function $\varphi(v)$ is active, because of the initialization with small random weights and scaling of the examples. Then at the very beginning we can consider $y_j(t)$ as a linear function with a little slope, biased at the average of the target[2]:

$$f_j(\boldsymbol{x}) \;=\; \varphi\Big(\sum_i (w_{ji}\varphi\big(\sum_k (w_{ik}x_k \;+\; Q_k)\big) \;+\; Q_i)\Big) \;+\; Q_{j.} \tag{6}$$

Then the summed difference between the network target and output has approximately zero impact for the period of one complete presentation of a symmetrical pattern set:

$$E(n) \;=\; \sum_{t=1}^{t=n} (d_t \,-\, y_t) \;\approx\; 0. \tag{7}$$

Moreover, random sampling of such a function provokes the zero summed effect of presenting parts of the input–target set. Fig. 5 depicts the sum of all previous output errors for a certain step while learning to approximate respectively nonsymmetrical and symmetrical functions. The error sum graph for the symmetrical function is plotted with a solid line. The dashed line follows the error sum for non–symmetrical function. Random sampling of a symmetrical function helps in this case to obtain zero summed effect also for subsets.

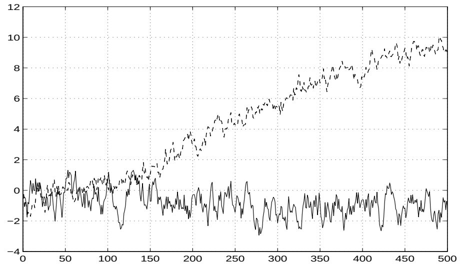


Fig. 5: The sum of the output error for the sampling steps up to one full pattern presentation. The dashed line shows the error sum development for non–symmetrical target. The solid line corresponds to a symmetrical target.

As mentioned before, the learning problems have a distinct similarity to a statistical long run effect of which the run length is by definition finite but varying. In the simple experiment, concerning Fig. 6, cancelation leads to bad approximation – the network gives a straight line output, if a random equidistant sampling is done. In training more complex signals, the cancellation can be expected to produce a wider range of learning times or poor approximation. Poor approximation appears most often as learning only the global signal structure.

Our theoretical conclusions are inspired by and coordinated with the mathematical analysis of Wiegerinck and Heskes [18]. Elaborating on how dependencies between successive examples affect on–line learning they suggest that the reason for bad convergence is the existence of flat areas in a global error surface (or also called plateau). Plateaus cause an extremely long training time and a bad generalization. After the network reaches such a flat area the weights hardly change anymore. Consequently good approximation becomes extremely difficult if uncorrelated input patterns are used.

It is important to point out, that networks with zero–centered sigmoid neurons suffer much more from cancelation phenomena than nonsymmetrical sigmoid networks. The reason is the larger influence of the error parameter when forming the weight correction value, if both networks are initialized in the same way and trained to learn the same function on the same input interval.

[2] In presence of learnable bias, equation (1) will be as (6), where *Qk* ,*Qi* ,*Qj* are correspondingly the biases of input, hidden and output neurons. Changes of the bias term will be the same as weight changes, because they are updated with the same adjustment factor. Finally the hidden and input biases will also approach the zero point. This causes the network output to be equal to the output bias value, which is quickly adapted to the middle of the learned function, a fact as observed in all made simulations.

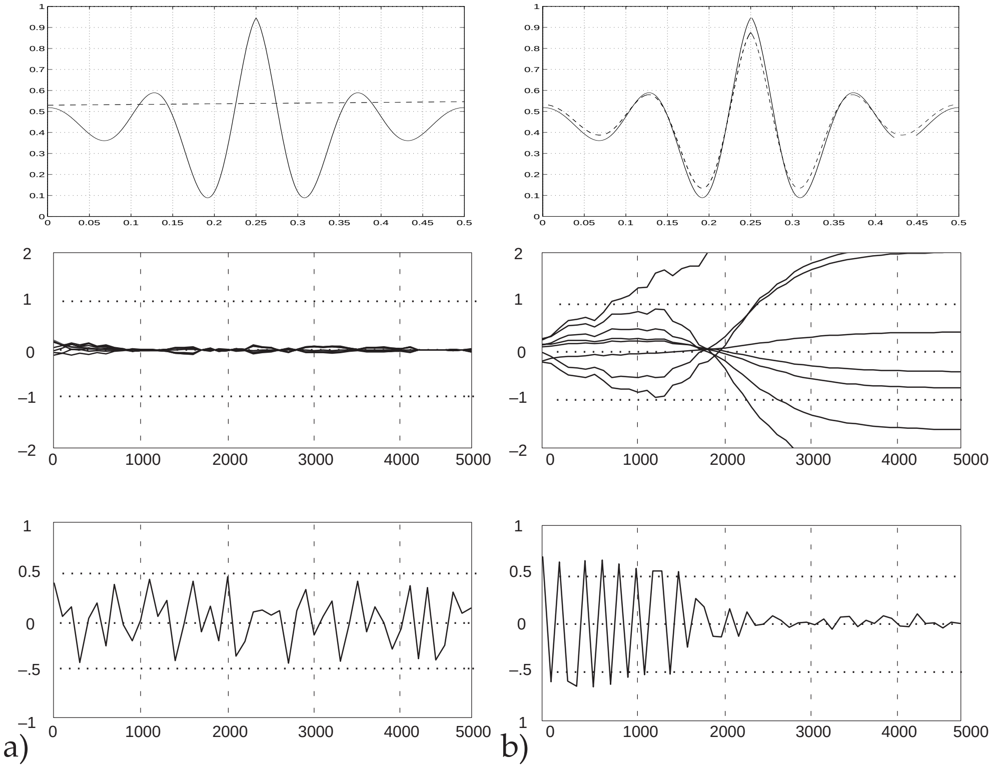


Fig. 6: Weights and error time behavior when a cancelation and noncancelation signals are to be approximated. Choosing a cancelation training set in this case can be done simply by randomizing equidistant samples from the function.

So far we have elaborated on symmetrical training sets, because in this case random sampling orders the training examples in a way, that propagated back updating coefficients are canceling each other and make the network parameters become zero. There are a lot of nonsymmetrical training sets as well which presentation can have zero summed effect, both in the long run as during shorter intervals. Thus, the cancelation can appear during training an arbitrary function if for a long time the cancelation examples, provoking a symmetrical phase in the learning process, are supplied long enough to bring the network parameters to zero values. The way it can be detected will be discussed in the following paragraph.

5: Cancelation detection.

Cancelation training sets either lead the learning process to a dead–end, or (in the case when the percentage of patterns with a cancelation potential is less than 100 from all the training examples) in slowing down the convergence process and in bad reliability. In other words, approximation may fail on a small number of consequent tests. First in this section will be shown how the learning quality can be damaged by the different content of cancelation examples in the training sequence. The possible damages range from not reproducible training duration and inaccurate final approximation until total crash of the learning process. Later on we will propose a method for detecting the cancelation in an arbitrary signal. Decreasing the possibility of training failure or low quality learning can be done in many ways, after the cancelation is detected. The particular way can be adapted to the problem to solve. We are suggesting a windowed active sample selection algorithm, which solves the cancelation problem and in the same time preserves the advantages of the positive impact of randomization inside the selected windows.

5.1 Impact of the cancelation content on the learning quality.

The risk of entering a cancelation situation exists for example when a periodical signal is sampled. Because of the stochastic nature of neural learning the exact borders of appearance of cancelation effects can probably never be determined, but a long statistical investigation over the signal shown at Fig. 3b gives quite informative results. This signal is artificially created to allow easy control over the content of cancelation examples in the training set. The percentage of cancelation examples in the extracted training sets varies within wide borders. For every particular number of cancelation examples 200 different training sets are extracted. With so created training sets two groups of experiments are made.

In the first group of experiments the percentage of cancelation examples is varied and at every step the average learning duration and the number of successful trials are recorded. Fig. 7 summarizes the learning performance of a network, trained with example sets extracted from the signal at Fig. 3b with different percentages of cancelation examples. Averaging is made over 200 training sets. In Fig. 7a the results of a statistical investigation over the effect of cancelation pattern sets on network reliability are depicted. The performance of the network on the subsequent experiments with differently randomized training set is plotted against the percentage content of the cancelation examples. It can be seen, that once a certain amount of cancelation patterns is present in a training set, the experiment becomes non–reproducible. Correspondingly, the necessary training time increases drastically. This is shown in Fig. 7b after all the non–learnable examples are discarded. These results concern networks with zero–centered sigmoid transfer functions.

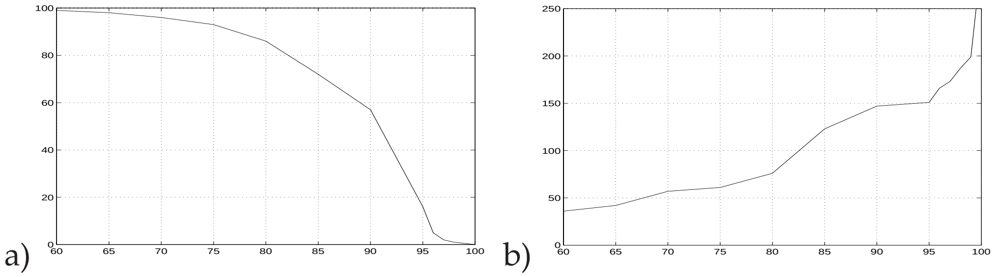
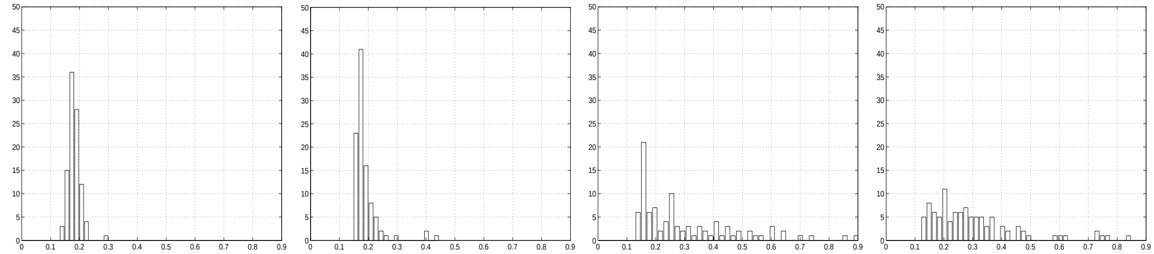


Fig. 7: a) Network generalization performance: Percentage of successful trials from subsequent tests decreases quadratically once a critical number of cancelation examples is present in the training set. b) Number of iterations during training increases once the number of cancelation examples exceeds certain limits.

The second group of experiments shows the replicability of the training duration for 4 groups of training sets, correspondingly with 40%, 60%, 80% and 100% of cancelation examples. The results of training with a network, build with zero–centered sigmoid neurons ($\varphi(x) = (1 - exp(-x))/(1 + exp(-x))$) are not shown, because they are implied more or less in Fig. 7. Instead, the unstable learning duration with nonsymmetrical transfer network is exhibited.

As commented before, the result of approximating cancelation signals with networks composed by nonsymmetrical transfer neurons ($\varphi(x) = 1/(1 + exp(-x))$) has not so big convergence problems. The reasons for that were explained in section 4. In this case the effect remains as not reproducible learning duration, if the training set contains a high percentage of cancelation examples, as shown in Fig. 8. In case of 100% cancelation there is a small percentage (about 2%) of experiments, that fail.



a) 40% b) 60% c) 80% d) 100%

Fig. 8: Convergence behavior of network with nonsymmetrical transfer learning the signal from fig. 3b with differently constructed training sets. a) exhibits the distribution of convergence time normalized for 100 experiments, done with training set which contains 40% cancelation examples. While increasing the amount of cancelation (60% in b, 80% in c, and 100% in d) we observe that the spread in learning time increases considerably, leading to not reproducible learning behavior long before the effect becomes noticeable as a stand–still.

From the shown empirical results we can conclude that the example selection can be derived from observed changes in runtime results. We are therefore suggesting the windowed active sampling strategy based on the analysis and the observations made so far. Moreover, we are tending to create an easy to implement method, preserving the advantages of randomness on the sub–training set level.

5.2 Windowed active sample selection algorithm.

To support the description of the algorithm itself we will illustrate the cancelation detection quality criteria. The illustration is made for training sets, that are not symmetrical themselves but can provoke cancelation. If the training set contains examples which in the order of their presentation have the sum of the direction coefficients equal to zero, ΔW_{ij} will be zero after a full presentation of this training set. At the following pictures the two different training sets (as shown at Fig. 9a,b) from the same signal and the graph of the mean of the training set direction coefficients as shown at Fig. 9c,d.

Fig. 9c clearly shows that the training set from plot 9a will not be learned if presented at random. On the contrary, the plot of the direction coefficients mean, corresponding to randomized training set 9b, approaches the value quite different from zero. This shows that the training set has no cancelation nature and can be easily learned. The here proposed algorithm implies this calculation for adaptation of the window size.

In the beginning we use a large portion of the signal and check by prototype learning on the evolution of the mean values of the direction coefficients. If cancelation is present, it will sharply move to zero and the experiment can be stopped. In sequence we try smaller portions till finally we find a window–size that shows no cancelation behavior. Then, in assembly, we can train from the small windowed segments and build in an hierarchical fashion upwards to finally obtain a full signal coverage. The detection algorithm takes the following steps:

1. The data set $\boldsymbol{D_n} \equiv \left\{(\boldsymbol{x_i},\boldsymbol{y_i})\right\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$ is extracted from the signal $\boldsymbol{S(x,y)}$ by random equidistant sampling.
2. Divide the data set $\boldsymbol{D_n} \equiv \left\{(\boldsymbol{x_i},\boldsymbol{y_i})\right\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$ on equidistant windows $\boldsymbol{D_{n_m}} \equiv \left\{(\boldsymbol{x_l},\boldsymbol{y_l})\right\}_{\boldsymbol{l=1}}^{\boldsymbol{m}}$.
3. After randomization, the training subsets for the first few epochs $\boldsymbol{E_{m_p}} \equiv \left\{(\boldsymbol{x_l},\boldsymbol{y_l})\right\}_{\boldsymbol{i=1}}^{\boldsymbol{pm}}$ are obtained from the data subsets $\boldsymbol{D_{n_m}} \equiv \left\{(\boldsymbol{x_l},\boldsymbol{y_l})\right\}_{\boldsymbol{l=1}}^{\boldsymbol{m}}$.

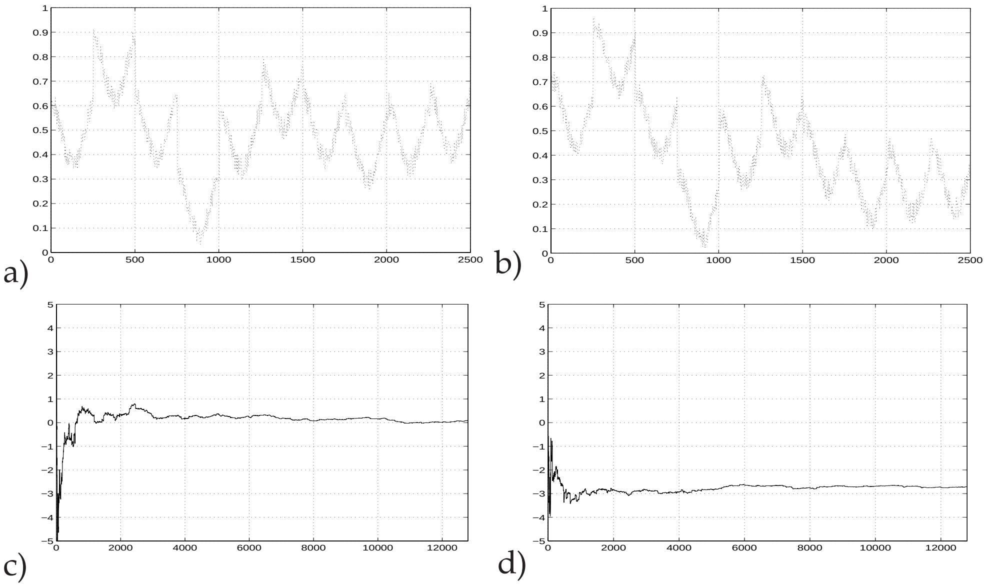


Fig. 9: Two training sets, extracted from the signal, shown at Fig. 2b correspondingly a) possessing and b) missing the cancelation property. c),d) – evolution of the mean value of the direction coeficient for the constructed training sequences.

4. Calculate the direction coefficient mean evolution, for the current training set $E_{m_p} \equiv \left\{(x_l,y_l)\right\}_{i=1}^{pm}$.
5. If detected existence of the cancelation, decrease the size of the window. Go to 3.
6. If evolution curves as calculated in 4 stabilize to show absence of cancelation the learning can be left on its internal dynamics. End.

6: Some practical experience.

We have presented active selective sampling as the on–line alternative for passive ordered sampling. The causes for network–internal conflicts are related to the presentation of the examples and an analytical justification is provided. Then a procedure is outlined that is based on the interactive reconstruction of the hierarchical structure as implicitly present in the example set.

The importance of this contribution is illustrated by the observation that even learning networks can be unreliable in its performance. Reliability is hereby taken as the ability to perform learning in a stable, reproducible way. For applications in an industrial setting, such a reliability should ensure real–time, hazard–free behavior. A typical example can be found in the diagnosis of power generators, as discussed next.

Destabilization of a turbogenerator by shaft torsion can be estimated during its operational lifetime by vibroacoustic measurements. The practical significance of a classical vibration–based fault monitor and report AI–system is limited as:

1. the amount of on–line information is too large for comfortable handling,
2. the count of all probable failures is too high for simultaneous monitoring, and
3. only known faults can be classified.

The first of these arguments stresses the need for on–line data processing, while the latter two state the case for a Connectionist Expert System as reported in [3].

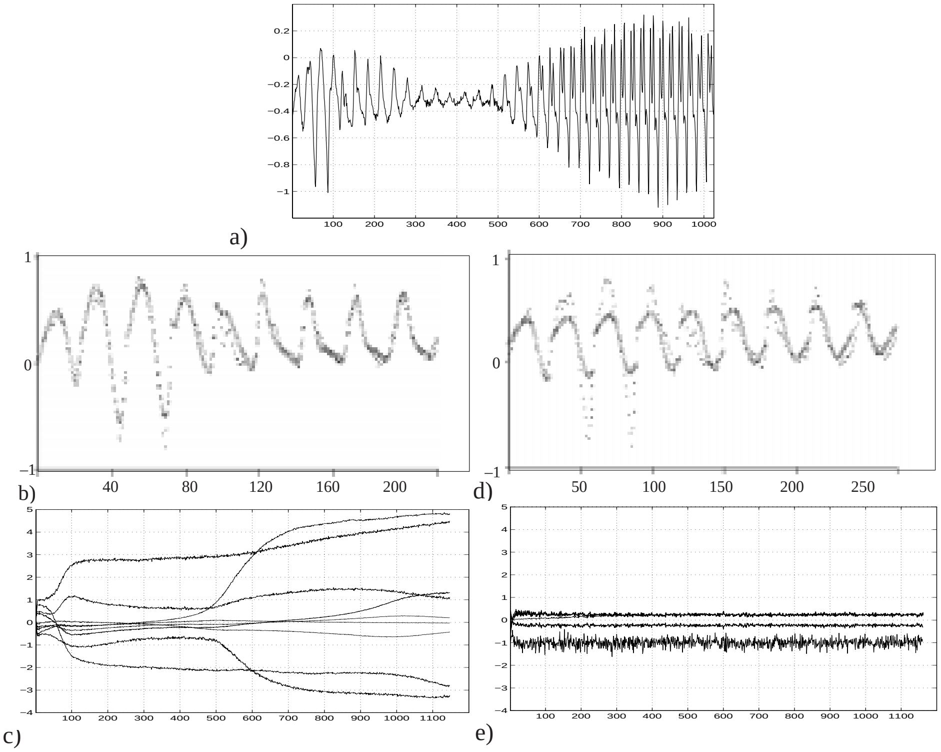


Fig. 10: a) Signal, recorded during the emergence working mode of a power generator. c) weight changes during cancelation–free training, leading to the signal approximation shown in b). e) weight changes during cancelation training leading to the signal approximation shown in d).

The signal, shown in Fig. 10a is recorded during the emergence working mode of a power generator. In order to discover in one training whether this signal has the cancelation potential it is divided into parts, defined by its odd and even local extrema. So created training set leads to poor approximation (Fig. 10d). The approximation quality will not improve with time, because the weights are oscillating in a small areas, as shown in Fig. 10e, and this can not be changed by repetitive presentation of the same training set. In this case the network output is governed by only 3 hidden neurons, which can successfully learn the periodicity of the signal. Fig. 10c illustrates the weight changes when a non–cancelation training set is extracted. In this case the learning quality is quite satisfactory, as shown at 8b.

The number of weights are the same in both cases, but in Fig. 10e they are graphically overlapping. Also the interval, from which the initial weights have been chosen, is the same. This is not clear from the subplots, because after the first dataset presentation the weights from Fig. 10c are changing noticeably, but this change is represented in very small plotting space (in the scale of 1147 dataset presentations).

A typical facet of this application of a neural network in an industrial environment is the occurrence of new frequency contributions from new failures as well as the shift in existing frequencies because of wear and ageing. For the diagnosis it is required to perform the above learning at regular intervals. These on–line requirements make extensive pre–processing impossible, while on the other hand reliability is of utmost importance to guarantee hazard–free operation

In our experience, we frequently encounter the cancelation phenomenon. Often there is an easy work–around by introducing pre–knowledge on the problem to be solved [17]. Here, hierarchical design to isolate the signal symmetry in pre–designed subnetworks has distinct advantages. In general, nonsymmetrical initialization is the keyword. Despite all this, comparing a number of learning attempts to verify the reliability of the product remains required to provide for a final quality guarantee. Under circumstances, and especially when no quantifiable understanding of the natural process is available, windowed active sampling provides for an attractive alternative.

Acknowledgements.

The intensive effort of the BRAIN section at the Rijksuniversiteit Groningen and the IMPACTs center for Research and Consultancy to create, provide and maintain the InterAct environment for the development of neural products is acknowledged. Furthermore our gratitude is due to the many people that commented on preliminary results of our research.

References.

[1] L. B. Almeida, A learning rule for asynchronous perceptrons with feedback in a combinatorial environment, in: Proc. IJCNN*'87*, (SOS Printing, San Diego, 1987) 609–618.

[2] L. Atlas, D. Cohn, R. Ladner, M. A. El–Sharkawi, R. J. Marks II, M. E. Aggoune and D. C. Park, Training Connectionist Networks with Queries and Selective Sampling, in: D.Touretzky, ed., Advances in Neural Information Processing Systems, Vol. 4 (Morgan Kaufmann, San Francisco, CA, 1992).

[3] E. I. Barakova, L. Spaanenburg, and J. Zaprjanov, Neural fault diagnosis of a turbogenerator by vibroacoustic data , in: Proc. ICSPAT'95 (Boston, MA, USA, 1995) 1454–1458.

[4] S. A. Barton, A matrix method for optimizing a neural network, Neural Computation, 3 (1991) 450–459.

[5] E. Baum and D. Haussler, What size net gives valid generalization, in: D.Touretzky, ed., *A*dvances in Neural Information Processing Systems, Vol. 1 (Morgan Kaufmann, San Francisco, CA, 1989).

[6] D. Chon and G. Tesauro, How tight are the vapnik chervonenkis bounds, Neural Computation, Vol. 4 (1992), 249–269.

[7] I. Cloete and J. Ludik, Increased complexity training , in: J. Mira, J. Cabestany and A. Prieto, ed., Proc. IWANN'93, (Lecture Notes in Computer Science, 1993) Vol. 686.

[8] F. Girosi, M. Jones and T. Poggio, Regularization Theory and Neural Networks Architectures, Neural Computation, 7 (1995), 219—269.

[9] L. Holmstrom and P. Koistinen, Using adaptive noise in backpropagation training, IEEE Trans. on Neural Networks, 3 (1992), 24–28.

[10] K. Hornik, M.Stinchcombe and H. White, Multilayer Feedforward Networks are Universal Approximators, Neural Networks, 2(1989), 359–367.

[11] R. A. Jacobs, Increased Rates of Convergence Through Learning Rate Adaptation, Neural Networks, 1(1988), 295–307.

[12] J. F. Kolen and J. B. Pollack, Back Propagation is Sensitive to Initial Conditions, in: D.Touretzky, ed., Advances in Neural Information Processing Systems, Vol. 4, (Morgan Kaufmann, San Francisco, CA, 1992).

[13] Y. le Cun, Generalization and network design strategies, in: Proceedings of Connectionism in Perspective (North Holland, Amsterdam, 1989).

[14] N. Morgan and H. Boulard, Generalization and paramether estimation in feedforward neural nets: Some experiments., in D.Touretzky, ed., Advances in Neural Information Processing Systems, Vol. 2, (Morgan Kaufmann, San Francisco, CA, 1990).

[15] M. Plutowski and H. White, Selecting concise Training Sets from Clean Data, IEEE Trans. on Neural Networks, 4 (1993).

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning Internal Representations by Error Propagation, in: D. E. Rumelhart and R. J. Williams eds., Parallel Distributed Processing, Vol. 1, Ch. 8 (MIT Press, Cambridge, 1986).

[17] M. H. terBrugge, J. A. G. Nijhuis, W. J. Jansen, H. Drenth, and L. Spaanenburg, On the representation of data for optimal learning, Proc. of EPIA'95, (1995).

[18] W. Wiegerinck and T. Heskes, How dependencies between successive examples affect on–line learning, Neural Computation, (1996).