4 Example selection.

The previous chapter puts forward, that the major obstacles for a reliable itinerary on the error surface (and for the learning reliability, respectively) stem from the peculiar error relief forms such as plateaus or valleys between the solution subspaces, creating conditions in which adequate learning is difficult to achieve even if it is theoretically feasible. The dynamics of the learning process may include situations where further progress is halted on such relief forms. The key to learning success is founded on the order, wherein the examples are presented. The most common way of example ordering is to select them at random from the set of available examples. We argue that this is not sufficient and introduce alternative sampling strategies. Particularly we propose a notion to allow for automation in this problem–driven example selection: the cancelation criterion. This criterion will be formulated and verified to show its potential for enhancing the learning reliability.

The nature of iterative learning on a randomly initialized and (in terms of the selection of training examples) arbitrarily developed network is such that a precise replication of the experiment is practically impossible. Every experiment is dependent on two factors outside the actual problem definition: (a) the choice of the initial parameters, that position the network at a certain point on the error surface, and (b) the order of presented examples that direct the itinerary on this surface.

It is widely known that an unsuitable initialization of the network can be a reason for its bad performance [107]. We claim, that the occasional failure of the learning algorithm is rooted in both above named factors: not only wrong initialization, which makes the learning algorithm to start from difficult to escape areas on the error surface, but also wrong guidance of the example flow in a way that precludes the escape from difficult error surface’s relief forms. In an attempt to generalize, it can be observed that the second case implies the first one: if proper guidance of the learning algorithm can find a way through a difficult landscape during learning, it is certainly possible to surmount a difficult initial position.

This claim can be supported with a very simple experiment scheme. Two groups of experiments are performed with the same network until the cost function for every single run drops below 0.01. Both experiments start in the same initial point with the same training set, ordered according to a different rule. The outcome of the two groups of runs show a difference in training time and in learning trajectory. Typical trajectories from both experiments with average performance (regarding to training duration) are visualized by error function development in figure 4–1.

This experiment shows, that the learning process depends more on the way training examples are presented than on its starting position. This implies, that a successful and reliable convergence of the learning process can be achieved by only controlling the presentation order of the examples during learning. The choice of a starting position can be made according to widely accepted initialization rules.
So far we have put forward that the effort to construct a reliable learning trajectory can
be spent entirely on directing the itinerary of the learning process. In contrast, ensemble
methods (in their simplest form) imply starting from different points in the weight space
and choosing the network (the solution) which takes the best trajectory. Such an ap-
proach does not answer to the question how to proceed if there is not a good solution
found or if the initial metastable state is not escapable for some reasons (see for illustra-
tion the analysis of a symmetrical function in a shallow valley in section 3.3.2.1).

![Figure 4–1: The error function for experiments with examples of the same
training set ordered in different manners. Both experiments are started from
the same initial point in the multi–dimensional weight vector space.](image)

We have characterized the choice of initial parameters and of the example presentation
order as dynamical aspects of neural learning. Within the error surface paradigm the
process dynamics relate to the route, that the learning algorithm takes on this surface
on its way to the optimal solution, while the surface shape is static. Since the route is
determined to a large extent by the example stream, normally constructed according to
some randomization principle, the impact of randomness on neural learning will be dis-
cussed in detail with respect to both its weak and strong points. Though random sam-
pling seems advantageous in helping the network to escape a state of symmetry, it can
not ensure that the same learning trajectory will be repeated twice. In the first instance,
this means, that the learning experiment can not be replicated. Consequently, this is a
cause for bad reliability: the learning duration can not be guaranteed.

Therefore alternatives to the random sampling scheme will be discussed here. In search
for dependencies between the learning quality and the example stream presented, some
experiments are performed with alternative ways of selecting training examples. It il-
ustrates the influence of different pattern presentations on the learning reliability, but
foremost suggests a framework for solving a number of learning problems, i.e. ensuring
the reliability in terms of higher probability for convergence.

Though a closed–form analytical expression to predict network behavior as a function
of example presentation order has not been found, frameworking the interdependence
between the example presentation and the weight changes and considering the peculiar-
ities of the error landscape leads to a practically acceptable criterion: the $\text{KRS–ratio}$. 
It is found out, that if the KRS–ratio converges to zero within one epoch, there is a high
chance for cancelation to appear. The suggested example reorderings are increasing the KRS–ratio development and correspondingly resolve cancelation situations. In accordance with the created cancelation criterion, a sample selection strategy is developed and its feasibility is shown.

4.1 The basics of sampling.

The creation of a training set that ensures optimal functionality from the learning process has many aspects. As the continuum to be modeled is infinite, taking representative examples involves a sampling operation: a finite set of examples is extracted according to a sampling algorithm, such that the set (a) can be learned and (b) allows for generalization. The properly trained neural network has then become an executable model of the infinite continuum.

Characterisic of a sampling algorithm are the sampling rate and the sampled interval. The relation between these two has been worded for information processing by Nyquist in his Sampling Theorem. It states that the information as present in the infinite continuum can be contained in the finite example set. However, the Nyquist Theorem is necessary but not sufficient for neural networks. It derives its significance from the fact, that it gives the minimum amount of examples that must be present in the example set. It is generally possible to reduce the size of the training set, as reported in [69] [146]. But in some applications obtaining the training samples is costly. Important in these cases is to find the smallest number of examples that lead the network to a good result [80].

The next question is therefore, whether all examples in the set must be used? It is clear that according to the Nyquist theorem again a minimum number of the examples must be used. The related question is therefore: if the example set is redundant, which examples must be chosen? This question has far reaching consequences, as it implicitly allows for a discreticized but potentially infinite input space. For the moment, we rather restrict ourselves in the number of ways to optimize learning.

Therefore we assume that an example set is available and will be fully used for learning. So next to the above physical concept of sampling data points from a signal, we will also view sampling as a data selection technique. From the pool of examples, we take all examples in a deliberate order and present them individually to the network for the purpose of learning. In other words, by sampling the data set is transformed into a presentation set, the difference being that in the latter the examples are already queued for presentation.

4.1.1 Sampling techniques.

The presentation order of the examples is of greatest importance for the success of learning. For a long time, a random selection of examples has been prefered as it was noted that by repeatedly presenting the examples in a fixed order forces the network to remember this order too. Often, this order has no real meaning and leads therefore to false learning. By presenting the examples in a random order, the network will never see the same sequence and therefore learns from the individual examples only.

In the following, we will start from this random sampling (RS) technique as a point of reference. Subsequently we introduce some variations to find that next to its obvious
advantages it also has some drawbacks that make for a reliability problem. It is essentially this reliability problem that we aim to solve in this thesis. Therefore a brief review of existing neural sampling schemes is made first.

4.1.1.1 Random sampling.
Most neural algorithms apply a number of random choices during their operation. Hence, even for a fixed input, different runs may give different results in time and accuracy. This nicely fits the definition of a randomized algorithm, according to [92]:

**Def. 4–1:** A randomized algorithm is one that receives in addition to its input data a stream of random bits that it can use for the purpose of making random choices.

There are two major advantages of randomized algorithms. Firstly, often the execution time or space requirement of a randomized algorithm is smaller than that of the best deterministic algorithm known for the same problem. Secondly, all the existing randomized algorithms are extremely easy to understand and implement [92]. Randomization of a known deterministic algorithm with a detrimental worst–case behavior converts it often to an algorithm that performs well with high probability on every possible input.

Randomized algorithms gather information about the distribution of their input data by drawing random samples. In the case where the input data are fixed it is useful to randomize the order at which they are presented. In this sense, the random transformation of the example set from data set to presentation set has been performed by a randomized algorithm.

In general, there are no tangible rules in neural theory relating the signal to be learned and the nature of training patterns. Thus random equidistant sampling gives in most cases a satisfactory result. Moreover, the random factor is crucial for the work of all learning algorithms of a stochastic nature. The founders of the backpropagation algorithm have suggested in [129] that not only the network parameters but also the training examples should be chosen randomly. The motivation for this choice is based on the features of the most commonly used optimization method – the gradient descent. Gradient descent is performed on the local error function:

$$f(w, z^n) = -\nabla_w e(w, z^n) \quad (4–1)$$

The stochastic gradient descent on a local scale approximates the deterministic gradient descent on a global error landscape:

$$E(w) = \lim_{T \to \infty} \frac{1}{T} \sum_{n=0}^{T-1} e(w, z^n) \quad (4–2)$$

Another reason to choose the training samples randomly is to reproduce the density of the underlying distribution. If there is no further information available, this choice is very reasonable, but it usually results in a suboptimal training performance. Beside its easy implementation, random example presentation supports a framework for theoretical investigation of the generalization properties of neural networks. It is generally referred to as “Learning from examples” (Section 1.2.1). Analytically, learning from examples method is examined in many studies, among which are [13], [41] and [46]. A complete description of this approach for analyzing neural networks’ generalization is given by Poggio et al. [123].
Random sampling assumes that training examples are arbitrarily chosen, and that the learning process evolves under its own dynamics. In this case, it can be said that the neural network is a passive learner: the neural network has no control over the construction of the presentation set.

4.1.1.2 Alternative sampling schemes.

The intuitive alternative to randomization is that of selecting the training examples to follow the time order of the extracted signal samples. This happens for instance when the natural order of presentation is of importance (as in time-series prediction tasks) or when this is the only possible way of obtaining the examples (like in on-line learning scenarios).

Both choosing the examples for training at random from the available samples as well as presenting them to the network in their natural order are passive learning schemes: training algorithms do not have control over the example selection and presentation process. The most broadly applied neural algorithms are passive. The Hopfield Network, for example, takes its pattern set as a whole and so does also the Back-Propagation algorithm in its batch version. In the stochastic backpropagation, examples to be learned are randomly selected [46], which is also a passive learning technique.

A lot of studies show superior results when training examples are chosen in some systematic way. Generally, the decision on which examples are selected, is made by the training algorithm. This is the reason to refer to this training scheme as active learning. Active algorithms applied to neural networks aim to ensure 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 classes of techniques for choosing training examples. The first group assumes that the network is partially trained on a set of previously acquired examples. This class of techniques is known as active sampling or progressive learning [46] and can be defined as the task of adding new examples to the set of available examples. The second class of active learning techniques is known as active selection or informative learning [80] and implies selection of training examples from the set of available examples.

In the active learning scheme the necessary number of training samples heavily depends on their distribution over the input set. Properly selected, these actions can drastically reduce the amount of fatalities and the computation time required for learning to be completed. As reported by Morgan and Boulard [106], 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 [39]). This latter result, together with [7], [121] etc., establishes neural active learning as an alternative for passive learning from random examples.

In summary, it can be said that the sampling of a signal to be learned as well as the selection of the examples for actual training are important characteristics of the training process. Different manners of sampling or example presentation require changes in the training algorithm, but not in the optimization principles. Therefore, active and passive learning are terms concerning not the training process in general, but the way of example selection and its benefits to the learning success. Different sampling and ordering methods are relevant to one of the sampling schemes, as summarized in figure 4–2.
4.1.2 Neural active learning.

Active learning has been studied independently by researchers in many different areas such as neural networks, robotics, computational learning theory, experiment design, information retrieval, and reinforcement learning. It addresses the effects of pattern selection and presentation on the learning process. The active learning algorithms applied to neural networks aim to ensure success of the neural learning algorithm by selecting the optimal information from the environment. They are either oriented to the strategy of pattern presentation or to the selection of the best training set.

4.1.2.1 Active selection (Informative learning).

A logical way of actively selecting the training examples is: to choose from a large and redundant set of samples those, that will allow one to achieve the ultimate goal of learning the problem at hand. The reduction in the number of training samples can be necessary for several reasons. Firstly, reducing the training set size will eventually speed up learning. Lets assume that the number of selected examples is much smaller than the number of overall available samples. Thereby learning can be expected to have a shorter duration, because it is accepted that the training time is proportional to the number of examples to be learned. Secondly, obtaining the training samples is costly in some applications. Then finding the smallest number of examples that lead the network to a good result, is of importance [80].

The training set reduction can be achieved either by an incrementally growing training set [121], by selecting samples or queries [14] or by skipping examples in the training set [2] [69] [146].

If the training set is not carefully chosen, learning can result in bad generalization. Then it is important to select a concise but efficient (in terms of generalization) training set. Learning schemes, that make it possible to select training examples such that the redundancy contained in them is avoided and generalization abilities are not worsened, are called query learning. It contains two non–separable tasks: data modeling and data selection.

The data modeling process starts with choosing a small data set from the problem to be learned and consists of fitting this data into a neural network model by an optimiza-
tion method [126]. The model should be accurate, but not over–fitted. One way to fulfill this restriction is to base the model selection process on cross–validation techniques [19] [89] [93]. The so–obtained model paves the ground for the data selection process, where an optimization criterion is defined and computed. This criterion can for instance correspond to the model’s estimate of the squared distance between its actual output and the expected correct output for a new data point. The data that minimizes this criterion is then added to the training set.

After the data model has been selected, there are three different approaches to find new data. The first approach is applicable when a neural network model has been established using already seen input/output examples and the exploration of the data space will include an incremental expansion of the training set. If the input value of a new example is unknown, it is necessary to calculate the input value to be queried. Usually, this new value is calculated stochastically. This first approach is called Query Construction [141], but, if there exists a string of random input values that can be queried, it is normally referred to as Query Filtering [121]. The process of adding more examples to the training set is often called data subset selection [30], [163].

In every training set there are “difficult to learn” training examples – the network learns them least and in approximate borders. Some work in neural active selection aims to improve the learning and generalization capabilities of the network by modifying the presentation frequency of patterns [30] [108].

In [46], a focus on particular examples is envisaged, which constitutes active learning, to escape from situations where the network is stuck in a local minimum. For all the studies mentioned above, numerous criteria exist to select the pattern to be learned, but, in all cases, the general principle is to choose the example that will give the maximum of new information about the environment. The way of implementing the method of changing the frequency of pattern presentation is the influence of each example of the training set to be weighted [30] [108].

4.1.2.2 Active sampling (Progressive learning).

Learning can be considered as an intrinsically temporal process: the environment is not learned as a whole and at once, but in several stages. As seen in figure 3–19, the network approximates the median of the samples at the very beginning of the learning process. Later the network learns the global tendency of the target function (Figure 3–19b). In the following steps the network approximates small details from the target. Finally, the complete signal is learned by the network (Figure 3–19d).

This subdivision of the learning process can be considered as an advantage, since it gives the possibility of learning progressively. Progressive learning (or also known as active sampling) assumes, that the training set is not learned at once. The final goal to learn the posed problem is achieved by fulfillment of intermediate tasks: the partial learning. There are not many studies on that aspect of progressive learning in neural networks, although it was evoked some years ago [46].

Active sampling is essentially concerned with determining the distribution of training examples, which may differ from the distribution according to which examples occur naturally in the learning environment. The modified distribution is determined by making use of a priori knowledge about at least one of three objects: (a) the computational model i.e. the neural network architecture, (b) the learning algorithm as largely determined by the error criterion used for training and the learning rule by which this error
is minimized, and (c) the learning task which includes the environmental distribution and any a priori knowledge of the target function. In some cases, there is enough knowledge about the learning task and the network model, and thereby it is possible to determine the optimal sampling distribution in advance, without need for active sampling \[5\] \[43\]. Active sampling determines the sampling distribution according to the result of learning upon previous examples. In this sense the network is an active participant in its training.

The most well-known study on progressive learning is done by Elman \[55\]. He solves a grammatical prediction task. Sentences, generated by an artificial grammar, are presented to a simple recurrent network, which must predict at each time the probability of occurrence of every word in a dictionary. When the corpus of sentences is presented as a whole to the network, training fails. Conversely, when the training set is divided into two subsets (one containing easy sentences and the other containing complex sentences) and when the proportion of complex sentences increases progressively in the same training cycle, then the training succeeds.

The improvements achieved by progressive learning can be intuitively interpreted from the shape of the error surface. The error surface is entirely determined by the training set and the network topology and transfer. As discussed in the previous chapter, the surface is the same during the whole learning period in the classical passive scheme of back-propagation learning. In active learning, the error surface changes continuously, since the training set changes. Consequently, when beginning with an easy subset, the gradient dynamic applied in a first error surface easily and rapidly leads close to a minimum of the surface, which places the network’s weights in a good configuration to learn the second subset, and this is repeated at each stage, until the end of the learning. This explanation is related to studies about initialization of the weights in back-propagation networks \[83\].

Such studies on progressive learning are very interesting, but there are some reasons to suppose that their validity is limited. Firstly, they suppose that the decomposition of the example space in successive training sets is known a priori. Secondly, the training set has to be changed. In \[55\] the next training set is selected every five epochs. In other experiments, there is a threshold on the error below which the system takes the following subset. This threshold is diminished at each subset, either by taking a priori defined values \[39\] or according to some law \[39\] \[45\]. In all cases, a priori information is given, that has a strong influence on the results. Thirdly, most experiments are made on recurrent networks \[54\] , where the design of consecutive training sets is based in the analysis of the temporal sequences; see \[132\] for some rather disappointing experience.

The historically first approach in progressive learning is the combined subset training \[30\] \[45\]: the network is trained on a subset of random patterns selected from the whole training set until a certain criterion is reached. Then a second subset is added to the first one, and so on until the whole training set is learned. Cloete and Ludik \[40\] suggest also an increased subset training, which implies a multiplication of the training set size by a fixed number. Another limitation of this approach arises from the problem to predict the difficulty of a task by the number of examples to be learned. Thus, the first subset can be quite hard to learn, almost as hard as the whole training set.

The natural development of the active sampling method is to build consecutive training sets by increasing not the size of the training set, but its difficulty. This is achieved in
[39], where the network is first trained to count from 0 to 1, then to 3, then to 7 and lastly to 15. The obtained results are significantly better than with a combined subset training. Similar experiments, but with different motivation, are reported in [55]. The author shows in a task of complex sentence prediction, that starting with a training set containing short sentences and increasing the length of the sentences allows the network to learn the whole environment, whereas it is unable to learn it otherwise. In the third approach, partial training as suggested by Jacobs [83], several intermediate tasks are presented to the network (before it learns the final task).

An interesting experiment is described by Fahlman [56], where the recurrent cascade correlation architecture is used to learn the Morse code. Learning is achieved by presenting the patterns from the easiest characters together with the most difficult ones. Implicitly an important part of pre-processing has been performed here, as the Morse code already takes the importance of characters into account. The code provides a sequence of dots and strokes for each character, such that the most frequent characters are assigned the shortest sequence. This makes it clear beforehand which characters are easy and which are difficult.

4.1.2.3 Active learning implementation principles.

A widely accepted and practical way to implement active learning principles is in the framework of so-called optimal experiment design [6], as popularized in statistics. This framework is based on a technique of Maximum Likelihood Estimation. The work in this direction has been carried out in [42] [44] [99]. An integrated prediction error has been used as the objective function for selecting a new data point. This error function assumes that the data points have asymptotically a normal distribution [121], of which the parameter values must still be found. It is possible to use a Markov Chain Monte Carlo simulation [115] to establish this measure.

Another framework to be considered is the sequential optimal recovery [13]. This type of analysis does not make probabilistic assumptions, but concentrates on the worst-case behavior of the algorithm. It has been subjected to a detailed study in the area of function approximation, in which one tends to compute the maximum possible error for the guidance of selecting the next data point. All the methods mentioned so far can be described with the Bayesian inference paradigm [142].

As described in section 2.3.1.1, an advanced approach for neural network training to overcome the dependence of the learning process from the choice of initial state is the committee method. This method copes very well with the Bayesian paradigm. When a committee of neural network models is obtained as a result of the data modelling process, there is an expected disagreement on where to sample for new data in the problem domain. This disagreement can be expressed as an ambiguity measure [93], or by the estimated output variances [89] [124].

The choice of committee model differs with each application area. Bootstrapping techniques have been used by [53] for the time-series prediction task. It was found that a better generalization performance and a more accurate estimation of output error bars is achieved [27] [149]. This confirms our expectation that pre-structuring of the presentation data based on a windowing technique is worthwhile. This idea will be further detailed later in this chapter. But first we need some more background information.
4.2 Alternative example selection schemes.

The alternative sample selection scheme, as will be motivated and developed here, aims to increase the reliability of neural learning. For this purpose it should correspond to the concept of reliability, namely achieving robustness and effectiveness through finding an optimal training trajectory. For making the learning itinerary reliable and always reaching a satisfactory solution, it should be ensured that even difficult error landscape areas will be passed. As elaborated on in previous chapters, such are very flat and very curvacious regions. Normally, a randomized training algorithm can escape flat areas and shallow minima or valleys; for a specific group of signals, which will be named cancelation signals here, such areas are usually an unsurpassable obstacle. The experiment in section 3.3.3 analyses one simple example of this signal group; others will follow in the remainder of this thesis.

There is not a learning method that can solve all potential problems. We attempt to get over unreliable convergence by a proper example selection scheme. Here it will be shown how selecting the training examples in a different manner can change the learnability of cancelation signals. Additionally, as was shown by the opening experiment of this chapter, the learning speed of an arbitrary signal may be increased by orders of magnitude through a proper example reordering.

Since the order of presenting the training examples is crucial for improving learning reliability and speed, the impact of example streams on the learning process has to be revealed first. After defining the overall characteristics of a cancelation signal further in this section, the cancelation training set will be introduced for the training itself. It predicts the learnability of a particular training sequence. The main point here is, that even if the signal has a cancelation nature, presenting the extracted training samples in a special order can avoid a number of learning problems. In order to define the way of selecting the training examples a simple criterion, called KRS–ratio, is developed. Its simplicity makes it easy to calculate for the current training sequence and thus to be implemented on–line in the training algorithm.

Calculating the KRS–ratio predicts which signals will be difficult to train. The learning difficulties are investigated by systemizing the results of many experiments with signals with different level of cancelation. Manners for increasing the KRS factor are easily derived from the features of the optimization method and random processes. The results of training large numbers of sets, produced by reordering the extracted samples in a special manner, are confirming the usefulness of the chosen resampling strategies.

4.2.1 Example presentation order and learning success.

The analysis of the data–driven nature of the learning process requires a corresponding framework. Here only general directions will be given to provide a basis for the sampling method which will be proposed in a further section. Moreover, the general analysis and solution for above described learning problems will be made within this framework.

Following the definition, given in 1.2.1, the learning algorithm’s task is encoded in the set of examples \( z^n \). Finding the function \( f(\lambda) \in \mathbb{F} \) which maps the problem best is equivalent to discovering the conditions in which differences between the responses,
given by the network and by the supervisor, to the same input vector are smallest. This difference for the pattern \( z^p = (x^p, y^p) \) is denoted as error \( e(z^p, w) \) in equation (2–1). The algorithm processes the sequence of examples \( z^p \) until the network error (2–3) falls below some given magnitude.

If \( z^p \) is a randomly drawn example from the training set \([z^1, z^2, \ldots, z^N]\), the total error function can be formulated as:

\[
E(z) = \frac{1}{N} \sum_{p=1}^{N} p(z^p).|f(x^p, w) - y^p|^2
\]  

(4–3)

where \( p(z^p) \) is the probability that an example \( z^p \) is presented, \( y^p \) is the target part of this example, and \( f(x^p, w) \) is the network response. The widely used random pattern presentation method requires that in one epoch all training examples are presented. Correspondingly, every example is equally likely to be presented to the network.

### 4.2.1.1 Impact on the learning success.

This implies that the optimization can be performed on the development of the weight values in time. The probability that example \( z^p \) is presented can be expressed by the probability that it is the example with number \( t \), taken for training the neural system. This correspondence indicates that the conclusions on the effect of consequence of pattern presentation can be derived by calculating the probability characteristics of the training set.

Learning reliability is to be judged on the trajectory which the learning process development makes on the error surface. As it was shown by equation (3–8), the changes of the error value with respect to the network parameters (weights) is equivalent by value and opposite by sign to the changes of the weights in time. This derivation is very helpful, since the change of the weights in time is related to the presentation order of the training examples. In order to give more insight to that statement, let us remember that the weight vector values at moment \( t+1 \) depend on those at moment \( t \) and on the weight change, caused by the example with sequence number \( t \):

\[
w_{kj}(t + 1) = w_{kj}(t) + \Delta_{z(t)}w_{kj}
\]

(4–4)

where \( \Delta_{z(t)}w_{kj} \) is the weight change when the \( t^{th} \) randomly selected example \( z(t) \) is supplied to the network. For the on–line mode the weight change after the presentation of the \( t^{th} \) random pattern is:

\[
\Delta_{z(t)}w_{kj} = -\eta \nabla e(z(t), w) = \eta \delta_{o}(t)\sigma_{j}(t)
\]

(4–5)

The generalized delta rule for updating the weights \( w_{kj}(t) \) includes also a momentum term, contributing the influence of the previous weight changes to the current one, i.e. of the change the previous examples caused to the weight values:

\[
\Delta_{z(t)}w_{kj}(t) = \alpha \Delta_{z(t-1)}w_{kj}(t - 1) + \eta \delta_{o}(t)\sigma_{j}(t)
\]

(4–6)

In order to see the effect of the sequence of patterns on the synaptic weights it is necessary to sum the changes which every pattern causes on the weight values:

\[
\Delta_{z(t)}w_{kj}(t) = \Delta_{z(t-1)}w_{kj}(t - 1) + \cdots + \Delta_{z(1)}w_{kj}(1)
\]

(4–7)

From equations (4–6) and (4–7) the current change of the weight value, caused by the current example \( z(t) \), can be represented as a time series.
\[ \Delta_{\omega \theta}w_{ij}(t) = \eta \sum_{n=0}^{t} a^{t-n}\delta_j(n)\sigma_j(n). \quad (4-8) \]

This form of the generalized delta rule has been used for analyzing the behavior of a symmetrical function in the initial shallow valley (equation (3-27)). The notation \( z(t)w_{kj}(t) \) makes here to show explicitly the dependence of the weight values on the training examples. In general such a notation makes the description less distinct, and will be avoided in further sections by substituting with \( w_{kj}(t) \). Besides, \( k(n) \) in equation (4-8) can be substituted, leaving

\[ \Delta w_{ij}(t) = \eta \sum_{n=0}^{t} a^{t-n}(y(n) - f(x(n), w))\varphi_i'((\theta(n)))\sigma_j(n). \quad (4-9) \]

As already explained, \( f(x, w) = \varphi((\theta(t))) \). Thus equation (4-9) will change into:

\[ \Delta w_{ij}(t) = \eta \sum_{n=0}^{t} a^{t-n}(y(n) - \varphi((\theta(t)))\varphi_i'((\theta(t)))\sigma_j(n). \quad (4-10) \]

In summary, the learning algorithm (a) chooses in an arbitrary way the examples \( z^n \), (b) orders them to a pattern stream \( z(t) \), and (c) transforms the pattern stream \( z(t) \), (where \( t \) is the natural numbering \( 1, 2, \ldots \) of the examples in the order of their appearance) to the sequence of weight values \( w_{ij}(t) \), formed by adding a weight change \( \Delta_{\omega \theta}w_{ij} \) to the previous weight value. In this sense the two streams \( w_{ij}(t) \) and \( \Delta_{\omega \theta}w_{ij} \) contain equivalent information:

\[ \{z^1, z^2, \ldots, z^n\} \rightarrow \{z(1), z(2), \ldots, z(t), \ldots\} \rightarrow \{w_{ji}(1), w_{ji}(2), \ldots, w_{ji}(t), \ldots\} \rightarrow \{\Delta_{z(1)}w_{ji}, \Delta_{z(2)}w_{ji}, \ldots, \Delta_{z(t)}w_{ji}, \ldots\} \quad (4-11) \]

This reasoning is valid in all the example presentation manners, which ensures an equal probability \( p(z^n) \) for every example \( z^n \) to be presented. The described equations will help for instance to imagine when the example stream leads to weight degradation and how this can eventually be avoided.

Later in this thesis, we will develop the idea that structuring of the example presentation ordering is necessary to avoid learning problems. From the above equivalence between sample presentation ordering and weight value change, one may therefore rightfully deduce that structuring the example set implies a structuring of the network. In this sense, the technique to be developed can be viewed as a first step in the direction of modular synthesis of neural networks with a guarantee on learning by construction. This makes our technique to a viable alternative to the “knowledge annealing” technique advocated by [144].

### 4.2.1.2 Resampling schemes.

Although an indication about the changes of the trajectory on the global error surface, characterized with the change of the weight vectors, is given by the data stream \( z(t) \), this indication is not straightforward. But certainly, some useful conclusions can be drawn.

The most general conclusion about the dependence between the weight changes and the differences between subsequent examples follows directly from equation (4-9): large
weight changes are proportional either to an abrupt change of input stimuli or to a large error (large differences between the target signal and the network output). In contrast, a stabilization on the weight oscillations is to be expected when the target signal is almost learned or when the input values are changing smoothly. The learning trajectory as well as the weight plots have one global path of changes, but on the short term they are exposed to small fluctuations [160]. In the long run, large changes in the weight values can be expected when consequent examples cause weight changes with the same sign (in the same direction). This obvious conclusion, derived from equation (4–10) will be considered when attempting to direct the learning trajectory in the forthcoming sections.

By now the importance of the inputs has been discussed in neural literature in terms of saliency or sensitivity analysis. The two basic methods for inputs importance estimation are known as predictive importance and casual importance. **Predictive importance** is concerned with the increase of generalization error when an input is omitted from a network [101]. By **casual importance** estimation the inputs are manipulated in order to find out how much the outputs will change [101]. A common conclusion of the sensitivity analysis methods is that different measures of importance are likely to be used in different applications of neural networks.

### 4.2.1.3 Bootstrap resampling.

Bootstrapping is used in neural research as a method for estimating generalization error based on "resampling". Since resampling is a typical feature of the bootstrap method, it can be used to asses the impact of sampling variation in the data set on the trained weights.

The bootstrap method generally requires no major assumptions other than simple random sampling and finite variance. This allows us to investigate the sample variability in a statistics of interest, by making repeated computations by a randomly drawn samples with the same probability distribution as the initial samples. The distribution of the statistic of interest can then be observed over the resampling experiments. Under appropriate conditions the distribution of the resampled statistics corresponds to the sampling distribution of the statistics of interest.

In [15], the bootstrap method has been used to assess the sampling variation. First a large number N of pseudo samples of size n have been drawn independent of the original sample and the statistical characteristics of every such sample are used to draw the conclusion for the original statistics. This bootstrap methodology allows for the identification of those variables that are truly predictive for the target. Our interest in this work is that it gives a statistical validation of the observed relationships between the input information, processed by the neural network and its target.

### 4.2.2 Example stream features.

Our experience shows, that some signals are more difficult to be learned by random pattern selection than others, see for example the discussion in Section 3.3.3. The perfectly symmetrical signal in figure 3–14 will never converge to the desired value, while the "easier" signal with imperfect symmetry leads almost effortlessly to a satisfactory solution. Since the difference in the two experiments lies only in slight changes of the contents of the target patterns set, one may expect, that there is a relation between target signal features and its learnability, correspondingly with its reliability.
The analysis of the experiment illustrated in figure 3–14 brings us to direct our attempts on the definition of signals that can cause reliability problems. Moreover, it suggests a way to overcome reliability problems, caused by such signals, by supplying the training samples in a way that the correspondent weight updates will not cancel each other. The experiment in figure 4–1 at the beginning of this chapter shows the influence on speeding up the learning process of different manners to order the examples of a training set. This latter experiment concerns a signal, which is learnable by a “standard” random pattern selection scheme.

The described experiments from figures 3–14 and 4–1 have been especially chosen to illustrate the potential impact on learning speed and reliability of the ways to select the training samples. An interesting question is whether the approximation of an arbitrary signal is affected by the example selection manner. If the relation between signal learnability and its structure, the way it is sampled or the order of presentation of the samples can be found, then learning quality and reliability is to be considerably improved. Such a relation, expressed by a (preferably simple) criterion will allow an easy check on whether the current training set can be learned easily, and – if not – how its learnability can be improved. This check can be made before training starts, which will give an indication whether learning problems can be expected with this signal if it is trained by means of random sampling. Reliability problems can appear as well during training.

4.2.2.1 Definition of a cancelation training set.

In Section 3.3.3, the behavior of a symmetrical function in the initial shallow valley has been analyzed. The initial shallow valley is a peculiarity especially for networks, constructed by zero–centered sigmoid neurons. As can be concluded by the analysis in section 3.3.3:

\[
\text{If the training set consists of pairs of samples which cancel their impact on the progress along the learning itinerary, a reliable learning by means of random example selection can not be guaranteed.}
\]

It is to be expected, that if the learning process meets a very flat error region, such a training set will bring prolonged training times as well. In both cases there is a large chance for the learning process to be of a poor quality. For instance in the approximation task from section 3.3.3 optimization can be directed to the optimal point on the bottom of this valley. In a flat region, such a signal will cause an itinerary which rather follows the lows of the random process (Figure 2–12b) than to take a direction, which corresponds to a better mapping of the training problem. The formalization of this learning problem, as well as the made experiments, brings us to generalize that this effect appears by a wider group of signals which have the potential to bring the learning process to an unwanted ending. A basis for determining the range of signals, which have the potential to cause reliability problems with a cancelation nature, will be the observations on the global weight vectors behavior (respectively on the taken learning trajectories) when unreliable learning occurs.

Different weight behaviors, associated with non–learning are (1) degradation of the network parameters; (2) all the parameters go to a few parallel trajectories; and (3) saturation of the network units. The first two are observed when the cancelation situation is present.

An intuitive rule for reliable learning in this case is to keep the weight vectors in the range such that neither saturation of the network units, nor network degradation occurs.
This implies, that the internal representation should be kept distributed enough not to allow a degradation of the weight vector values. Of course, this might not be true for networks that need pruning. For some tasks, networks with a very large amount of weights are designed, and the learning process prunes out the unnecessary parameters.

Interpreted through the error landscape paradigm, degradation of the network takes place, when a high energy valley, or local minima is reached. The optimization goes into the direction of the local lowest point. Degraded to a few parameters, the network is able to learn a detail of the problem, corresponding to this high energy stationarity.

The observed effects of the existence of a cancelation phenomena during training have been analyzed by a typical example in approximating a symmetrical training set. In this case degradation is not an obvious phenomenon since the weight vectors have no time to grow initially and loose potential due to cancelation. The degradation in this case can not be distinguished from the lack of initial symmetry breaking. Here an attempt will be made to generalize the analysis made for a symmetrical signal to cases where a secondary cancelation of the training examples and respectively network degradation will appear.

As discussed already, the degradation can be encountered in two forms. The first one is characterized by the typical behavior of the network parameters to approach a zero point which, when mapped on the global error surface paradigm, corresponds to a multi–dimensional high energy valley, at which most of the network parameters reach its bottom. The bottom of a multi–dimensional valley can have several non–zero directions, which correspond to the non–zeroed network parameters. The second form of degradation is characterized by all network parameters taking a few parallel trajectories.

Let’s go back to the assumption, made for analyzing a symmetrical signal, that $n$ uniformly distributed samples are taken from the input signal. Let the input signal is scaled in x–direction between –1 and 1. Then there are $n/2$ samples $x_i$ in the range $[–1 0]$, and $n/2$ samples in $[0 +1]$, such that for every sample $x_i$ there exist a sample $x_j$, $i \neq j$, for which $x_i = –x_j$. To every input tuple $x_i, x_j (x_i = –x_j)$ corresponds $y_i, y_j$ tuple of the target signal, such that $y_i = y_j$.

The explanation of the cancelation effect was made on the assumption, that two corresponding elements of a tuple are presented in subsequent order. This way the impact, which the second example from the tuple will have on the learning development, is canceling the impact from the first one. In practice the probability that two such examples are presented in subsequent order is not very high. But since two subsequent examples are most often on different sides of the zero point, in the longer run the summary impact of the example presentation is zero.

Equation (4–10) will be used in order to generalize the appearance and influence of this phenomenon for an arbitrary signal. In it the weight adaptation $\Delta w$ was expressed as a sum of the changes, caused from a sequence of training examples. This sum can be divided into two sub–sums:

$$g_i(t) = \eta \sum_{n=0}^{i} \frac{\alpha^{i-n}}{2} y(n)q'(\delta(n))o_j(n)$$ (4–12)
This division of the weight adaptation time series is helpful to reveal the impact of the target sequence when the input sequence does not lead to meaningful changes in the network output, i.e. when a degradation of the weight vectors occurs. Only the first sum depends on the training signal (equation (4–12). The second sum is externally dependent only on the flow of input values (equation (4–13)). A similar description has been used for the analysis of the experiment in section 3.3.3 from equation (3–28).

By analogy with this analysis, it can be expected, that cancelation problems will appear when the first sum (equation (4–12) converges close to zero for one training epoch. With the assumptions made for random sampling this implies that the training signal has little impact on the training process. What the network produces as an output is in this case a combination of the mean of the signal and the impact of the input sequence. When the example sequence is arbitrarily ordered, the network approximates only the signal mean if the training has just begun, or stays at the already learned part feature of the signal in later stages of learning.

Therefore, in order to find out the impact of the target signal on the training process (in peculiar error surface areas), we need to analyze only the first additive component $g_1$.

If equation (4–12) converges close to zero during one epoch:

$$\lim_{t \to N} g_1(t) \to 0$$  \hspace{1cm} (4–14)

then it can be expected that the examples in the signal to be learned will cancel each other’s impact even in shorter intervals due to the additive properties of random, zero–symmetrical sequences.

To determine the signals that will satisfy (4–14) let us look at this equation into more detail. The constants $\alpha$ and $\eta$ can be ignored, since they are not adaptable for the “standard” way of performing backpropagation. Except on the target values, the equation (4–12) (which is implied in (4–14)) depends on the multiplicative term $\psi$, which for the $i$–th example is:

$$\psi^i = \varphi'(\dot{\theta}(x_i))o_j(x_i)$$  \hspace{1cm} (4–15)

Therefore it depends indirectly on the input distribution. As in the symmetrical case it is convenient to scale the input signal between −1 and 1 and the target signal $y$ between −1 and 1 for the zero–centered sigmoid transfer and 0 and 1 for the logistic one. This choice is made because of the active parts of the corresponding transfer functions. Since the scaling factor for the target signal $y$ is not of importance for the condition (4–14), it is chosen to be [0...1].

Then if two subsequent training examples $z^{2i} = (x^{2i}, y^{2i})$ and $z^{2i-1} = (x^{2i-1}, y^{2i-1})$ have equal but opposite values for the products of target and $\psi$ values (equation (4–16)), they will cancel to a large extent the impact of each other’s presentation.

$$\psi^{2i}y^{2i} = -\psi^{2i-1}y^{2i-1}$$  \hspace{1cm} (4–16)

If the complete training set consists of such sample tuples, their sum will be equal and describes the condition in which cancelation will appear.

\[ g_z(t) = \eta \sum_{n=0}^{\infty} \frac{\alpha^{t-n}}{2} \varphi(\dot{\theta}(n))\varphi'((\dot{\theta}(n))o_j(n)) \]  \hspace{1cm} (4–13)
As can be seen from the plot in figure 4–3 of overall $\psi_i$–values of a zero–centered sigmoid network, there is a very high probability that the last two multiplicative terms of equation (4–12) provide an equal probability for the network to give positive or negative output. When this altering of the sign of network output is present in a short series of steps, the values are subtracted from each other. Thus only on the third multiplicator from equation (4–12), the target sample set determines whether training will lead to degradation.

\[
\sum_{i=1}^{N/2} (y_i^2 \psi_i^2 - y_i^{2i-1} \psi_i^{2i-1}) = 0 \quad (4–17)
\]

This equation for a cancelation check is in general not suitable for prediction of the learnability (or reliability) of a signal, because the $\psi_i$ values are not known. Our aim is to derive an approximate criterion, which depends entirely on the extracted training examples. For a symmetrical signal it is possible to substitute the $\psi_i$ values with $x_i$ values, as discussed in the following subsection.

### 4.2.2.2 Symmetrical signals and cancelation.

For every $y$–symmetrical signal the following dependence is valid:

\[
\int_{-1}^{0} - y(x)dx + \int_{0}^{1} y(x)dx = 0 \quad (4–19)
\]
If sampled equidistantly and scaled under the assumed conditions, the input values \( x_i \) and \( x_{i+1} \) (corresponding to the tuple values \( y_i \) and \( y_{i+1} \)) have opposite signs and equal magnitude. This implies, that the \( x\)-values of the examples are contributing effectively only with their sign to the adaptation process, if presented sequentially. Then (4–18) can be represented in the following way:

\[
\sum_{i=1}^{2N} y_i \mathcal{E}(x_i) = 0 \tag{4–20}
\]

where the function \( \mathcal{E}(x) \) is as follows:

\[
\mathcal{E}(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
0 & \text{if } x = 0 \\
-1 & \text{if } x < 0 
\end{cases}
\]

By sampling uniformly and at random, the probability that examples with numbers \( i \) and \( i + 1 \) have opposite valued \( x \) parts is very high. Cancelation will take place, if the \( x \)-coordinates of the drawn samples alter often. As was already shown in a number of experiments, cancelation can appear after a part of the approximation process has already been done. At later stages of the learning process, after some details of the signal or its main tendency are already mapped onto the network, equation (4–20) can be valid for the unlearned part.

![Graph](image)

Figure 4–4: Fast detection of cancelation in sinewaves.
Equation (4–19) gives an illustrative rule, which allows to determine the danger for learning reliability in simple signals. For a half sinewave scaled as shown in figure 4–4a, equation (4–21) is valid. If substituted in equation (4–19), the integration will lead to a zero result and thereby signifies the occurrence of cancelation. A similar reasoning can be made for the signal from figure 4–4c, which consists of two positive sinewaves with additional uniformly distributed random noise.

\[ y_1(x) = \sin\left(\frac{1}{2} \pi x + \frac{1}{2} \pi\right) = \sin\left(\frac{1}{2} \pi (x + 1)\right) \quad (4–21) \]

In contrast, the full sinewave in figure 4–4b does not satisfy the continuous cancelation as can be concluded by equation (4–22). Once substituted in eq. (4–19), the integration will provide a non–zero result, which signifies the exclusion of cancelation.

\[ y_2(x) = - \sin(\pi x + \pi) = \sin(\pi (x + 1)) \quad (4–22) \]

The corresponding experiments confirm that the first and the third signal have cancelation nature, which reveals by random sample selection.

The way shown to predict reliability, respectively learnability, of simple signals is convenient because it is illustrative. In real–life applications the signal shape is usually not known. The predictions about the possible reliability problems should be made on the basis of the available (often noisy) examples. In such a case calculating the integral from equation (4–21) is not possible.

4.2.2.3 Cancelation criterion.

Neural systems learn from examples taken from the training signal by means of some sampling rule. The extracted samples determine what the network can learn about the task, represented by this signal. As was elaborated in the previous section the acquisition of the knowledge, contained in the extracted samples, might be obstructed by a restriction on the interaction between network and learning algorithm, which can be revealed by the specific representation of some tasks, i.e. a specific example set. The definition of a cancelation training set relates the signal, or more specifically the extracted example set, to its learnability. The definition assumes that the training examples are presented in a random manner.

Beside the contents of the extracted example set, its ordering (for instance the current randomization) can cause variation of the learning performance. Further in this section, experiments that show how different orderings of the same training set can considerably change the training duration and repeatability (and even avoid some learning problems) will be shown. Furthermore, resampling based on signal variability can bring a different outcome when repeating an experiment. Therefore, instead of checking the learnability of a signal (which we consider equivalent to checking the extracted example set) it is more convenient to examine the current ordering of the training examples and to change it if necessary. Still, checking in advance the cancelation properties of the signal remains a necessary step to determine whether further resampling or example reordering will be necessary in a particular case.

Both examinations (of the learnability of a signal, and of the current sample sequence) can be done in the same way if the cancelation criterion is expressed through the training examples only. Such a representation of the cancelation criterion will determine what type of example reordering will lead to improved learning. This will be the basis for creating an adequate sampling strategy.
The definition of a cancelation training set relates the signal to the potential and the certainty it can be learned. Therefore the formalization of the cancelation danger only through the training examples will help controlling learnability and learning reliability, respectively.

Equation (4–20) determines the range of signals, which in terms of random uniform sampling (which presumes altering of the x-value sign at almost every iteration) has the potential to cause failure of the training process. The conclusion, which can be drawn from the previous section and from the corresponding experiments is, that the existence and subsequent ordering of pairs of training examples, which exactly cancel each other’s impact, is not a strict condition determining the learning quality. If the summary effect of a few sequential target value presentations from the negative region with respect to the x-axis is equal to the summary impact of the target examples from the positive x-range, then equation (4–18) will reach the zero point. As a result the weight values continue to oscillate around the state they have reached already or degrade to zero.

As was shown before (see figure 4–1) reordering of the samples from the same training set can not only improve the quality considerably, but also speed up the learning process and ensure a repeatable training duration. In order to be able to test whether the constructed training set (after a randomization or special manner of sampling or ordering of the examples) has the potential to cause learning problems, the cancelation criterion should be constructed in a way, that will encounter the sequential order of the presented examples. Again the training set is assumed to be scaled within the \([-1, 1]\) range on its x coordinate and within \((0,1)\) range on its y-coordinate. This scaling helps to define the cancelation properties of the training set and may be independent of the scaling of the examples, which will be used for actual training.

The analysis of a symmetrical training set in a shallow valley, made in the previous chapter, was based on the assumption that two nearly equidistant (measured from the beginning of the coordinate system) samples are supplied sequentially to the network. The difference between their y-values \((y^i - y^{i-1})\) is approximately zero. A subtraction of their corresponding x-values \((x^i - x^{i-1})\) will give the doubled x-value. On the small time scale weight vectors have slow dynamics: they are adapted by small amounts and rarely change their sign. Therefore the change of the value of the network function is small, and its sign is determined by the sign of the training examples. In the beginning of the training process the parameter values are small enough to provide noticeable changes in the current values of \(g_2(t)\) terms (see eq.(4–13)).

In summary, if two equidistant patterns of a symmetrical signal are supplied to the network in the beginning of the training process in consecutive order, the following set of equations is valid:

\[
\begin{align*}
    x^i &= -x^{i+1} \\
    y^i &= y^{i+1} \\
    g_1^i(t) &= -g_1^{i+1}(t) \\
    g_2^i(t) &= g_2^{i+1}(t)
\end{align*}
\]  

In the beginning of the training process the values of \(g_2(t)\) are negligible, because of the small parameters values. In this case, the initial symmetry breaking can occur only
due to the influence of the elements of the \( g_i(t) \) sum. The instant values of \( g_i(t) \), corresponding to equidistant examples, are canceling each other impact being approximately equal by value and opposite by sign.

\[
\begin{align*}
\begin{cases}
    x^i &= -x^{i+1} \\
    y^i &= y^{i+1} \\
    g_1^i(t) &= -g_1^{i+1}(t) \\
    g_2^i(t) &= g_2^{i+1}(t)
\end{cases}
\] (4–24)
\]

is valid, then the calculated \( g_1(t) \) values are different and the initial symmetry breaking occurs easily, since the adaptation steps are big enough due to the non-cancelation subsequent \( g_1(t) \) values. Such a case is shown in figure 4–5c, where the weight vector symmetry is easily broken. In tangible terms a broken initial symmetrical phase corresponds to the learned main tendency of the signal, as shown in figure 4–5b. After the mapping of the main signal tendency, the combination network–signal shows cancelation properties. In the not learned part of the signal, \( y \)-values correspond to opposite and comparatively close values of the \( x \)-vector. This contradicting information stops further adaptation (when training is performed on the zero–centered sigmoid transfer network).
In both cases it is useful to define a directional coefficient of the changes in the signal that should be learned. The sequential presentation of two examples which have equal teacher values and different stimuli vectors will cause contradictory (canceling each other) adaptation steps. If the growing (decreasing) direction of a signal is taken, which will correspond to a non–zero direction coefficient value, the consequent adaptation steps will provide meaningful changes. The so–calculated ratio prevents from contradiction when for x–samples that are close in time correspond y–values that are close in magnitude. In many smooth signals, such a correspondence between x– and y–couples of values exists and there is in fact no conflict situation. In conclusion it can be stated, that a small direction coefficient ratio, indicated by a very small value of the fraction (4–25), is an indication for a cancelation situation.

$$\left| \frac{y_i - y_{i+1}}{x_i - x_{i+1}} \right| \ll 1$$  \hspace{1cm} (4–25)

This fraction takes into account the ambiguity, which appears when two stimuli, having the potential to cause similar $g_2(t)$ adaptation components, correspond to equal teacher values, which provide similar but opposite $g_1(t)$ components. A cancelation occurs rarely in two subsequent training steps, but even in a short training interval the normalized sum of the non–absolute values of all such tuples of examples is:

$$KRS(z) = \frac{1}{M} \sum_{i=0}^{M} \frac{y_i - y_{i+1}}{x_i - x_{i+1}}$$  \hspace{1cm} (4–26)

where $M$ is the number of the performed training iterations. $M$ can span from one to few presentations of the training set. If for an arbitrary training set the sum of all such a fractions for the already ordered (or just randomized) training groups of $N$ samples converges to a value near zero,

$$\lim_{M \to \infty} KRS(z) \to 0$$  \hspace{1cm} (4–27)

it can be expected that the so–ordered training set will bring the network to degradation and the learning process will be unreliable.

Equation (4–26) represents the sum of the direction coefficients for the current sampling of a signal. If condition (4–27) is valid for a training set, this training set can not be learned by means of random example selection. For example, if we apply the cancelation criterion to the signals from the experiment visualized at figure 3–14, then the KRS–ratio, calculated for the first signal, converges to zero. The development of the KRS–ratio corresponding to the second signal predicts that this signal can be learned if the extracted training examples are randomly ordered. In figure 3–14b the KRS–ratio of the sinewave, trained on the interval $[0, 2\pi]$, is shown. It predicts problemless training, which is confirmed by the corresponding experiments.

Actually the three ratios indicate the network learning performance on the same signal, exemplified on different ranges. The three example sets are representative: they contain enough information so that the internal generator of data can be learned by the network. Correspondingly, it is indifferent which of them will be used for training the network. The three KRS–ratios converge to different values. It could be expected, that the highest KRS–ratio ensures faster training, since it predicts that there are less examples, that have the potential to cancel each other’s impact. Indeed, the performed experiments have shown, that the complete sinewave is much easier to be learned than the one,
trained on the interval \([0, 0.95\pi]\) or the noisy half sinewave. This observation suggests, that beside the learnability, the KRS–ratio development is related to the training duration and to the learning reliability, respectively. This dependence will be investigated in the forthcoming section.

The KRS–plots, shown in figure 4–6, predict learnability of very simple signals. The training process is entrapped in a stationary state in the very beginning of the training process. Once escaped from this state, there are no further risks for the learning success. By more complex signals the learnability, respectively the reliability problem, can appear in the beginning as well as in later stages of training, when the network has already learned to map a feature of a signal. Our goal is to investigate both learnability and learning reliability in realistic learning scenarios. The following investigations will be made by signals, which have the potential to cause prolonged learning or stand–still situations in later stages of learning, and not when an initial symmetry breaking still has not occurred.

**Figure 4–6:** The KRS–ratio evolution of the training sets, extracted by random sampling of the function \(\sin(x)\) on intervals \([0, \pi]\), \([0, 95\pi]\) and \([0, 2\pi]\) and with added noise.

### 4.2.3 Reliability of cancelation signals.

The previously defined cancelation criterion aims to predict the learnability of a signal or of the extracted and (most often randomly) ordered training set. It gives a quantitative
estimate whether the created training sequence will cause bad learning. As discussed in chapter 2, the learnability of a signal is the main condition for estimation of its reliability. The other aspects of the learning reliability are training time and mapping accuracy, i.e. the degree to which a signal has been learned.

To estimate the reliability of cancelation signals, the mapping accuracy will be taken as a fixed value. The evaluation will be made on (a) whether a signal has been learned to the desired (fixed) degree, and if so, (b) how long the achievement of this goal took and whether the learning duration and reproducibility of the experiment with duration of the same range can be guaranteed.

In terms of the introduced cancelation criterion, a degree of cancelation will be defined. This degree will be computed in percentage of cancelation examples. The connection between the degree of cancelation and the measured reliability aspects will help to understand whether and how the existing training set can be learned faster by resampling or reordering of the training examples. Moreover, whether the example reordering can guarantee the training time (i.e. the replicability) of an experiment.

As already discussed in chapter 2, the stochastic nature of neural learning does not allow exact evaluation of reliability of the neural learning process. In an attempt to quantify the learning success and duration, a statistical investigation over the signals with unreplicable convergence will be done. The investigated cancelation signal is shown at figure 3–14. This signal has been chosen for higher practical convenience of the experimental results than could be provided by a symmetrical signal. This signal overcomes the initial symmetrical phase and shows cancelation behavior in later stages of learning. As already explained, cancelation concerns the effect of long–term presentation of training examples, which have the potential to provoke changes in the network state with equal but opposite strength. This phenomenon becomes noticeable when the error surface contains shallow valleys and flat areas, which can entrap the learning algorithm.

The level of cancelation corresponds to the degree of internal symmetry and can be measured in percentage of examples with cancelation potential.

4.2.3.1 Periodicity.

As elaborated in other studies [10] [21], periodical signals may contain a risk for learning reliability. We agree, that for such signals cancelation can appear more often than by others. Since a periodical signal will give a more realistic view on the cancelation problem than simple symmetrical signals, our investigation will be continued on such signals. The periodical signal from figure 4–7 is artificially created to allow easy control over the content of cancelation examples in the extracted training set. To generate example sets with an a priori known level of cancelation, two different fragrances of variation are added. The main tendency of the signal is a fading sinusoid, and the additive components with higher frequency are providing the required percentage of cancelation examples. A signal with a high level of cancelation will correspond to

\[ f(x, y) = \sin(\pi x)\exp(-x) + \cos(\pi y). \]

For the sake of convenience, uniformly distributed noise is added. The noise level is less than 10% of the magnitude of the training signal. The relation between \( x, y \) and \( t \) is as follows: \( t = \frac{T}{X}x + y \), where \( T, X, Y \) are correspondingly the total number of samples and the maximum number \( x \) and \( y \) values \((T = XY)\). The corresponding KRS–plots predict that the first signal will be more difficult to learn. We expect that the can-
cancelation additive components will cause bad learning in a single experiment and unreliable performance when subsequent trainings are performed. If during training the difference between the signal and the learned part satisfies the criterion in (4–26), it can be expected, that further optimization will be impossible. Every periodical training signal has almost such a potential. We expect that there are similar reasons, which cause degradation in on–line learning, when the new coming information causes forgetting of the already learned mappings.

![Periodical signal with noise](image)

*Figure 4–7: The cancelation signal investigated.*

4.2.3.2 **Mean and variance of training duration.**

The learning time and the success of approximation of this signal has been extensively experimented with. The percentage of cancelation examples in the extracted training sets varies from 10% to 100%. For every particular number of cancelation examples 200 different training sets are extracted, ordered and applied.

![Network generalization performance](image)

*Figure 4–8: Network generalization performance.*

In figure 4–8a the results of this investigation into the effect of cancelation pattern sets on network reliability are depicted. The performance of the network on subsequent experiments with differently randomized training sets is plotted against the perceptual
presence of cancelation in the examples. It can be seen, that once a certain amount of cancelation patterns is present in a training set, the percentage of successful trials decreases quadratically and the experiment becomes non-reproducible. Simultaneously the training duration increases drastically. This is shown in figure 4–8b. The learning duration is plotted versus the percentual presence of cancelation examples after all the non-learned experiments are discarded.
two of the aspects of reliability, as defined in chapter 2: the increase of the training time and the decrease of the number of successful learning trials. It is obvious that the third reliability factor – the mapping accuracy – is implied in this evaluation.

Investigating the reliability of neural learning has two goals: first to find out when learning success can be guaranteed and, second, to try to answer whether a training experiment can be replicated within the tolerance defined from the random sampling.

As commented before, the result of approximating cancelation signals with networks composed by logistic transfer neurons has not so big convergence problems. The reasons for that were explained in Section 3.2. 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 figure 4-9. In case of 100% cancelation there is a small percentage (about 2%) of experiments, that fail.

The experiments with different contents of cancelation examples are repeated 200 times for every cancelation level. Figure 4-9a exhibits the distribution of convergence time for training sets which contains 40% cancelation examples. While increasing the amount of cancelation (60% in b, 80% in c, and 100% in d), it can be observed that the spread in learning time increases considerably, leading to irreproducible learning behavior long before the effect becomes noticeable as a stand-still.

The degree of learnability of a signal relates to the predictions, made by the suggested cancelation criterion. A few experiments with different randomizations for the already chosen amounts of cancelation examples are depicted in figure 4-9 e,f,g,h in terms of the corresponding KRS-ratios. It is obvious that the higher the number of cancelation examples, the closer the cancelation curves approach zero. These plots confirm the validity of the cancelation criterion predictions for the investigated signal.

The straightforward conclusion from the shown empirical results is that the example selection can be derived from observed changes in runtime results. Therefore a windowed active sampling strategy is suggested to be based on the analysis and a observations made so far. Moreover, we tend to create an easy-to-implement method, preserving the advantages of randomness on the sub-training set level.

### 4.3 Sampling strategy.

The created cancelation criterion relates the signal, represented by the extracted training set, to its learnability. This has been shown by an experimental setting, organized in the following way. Firstly, various training sets have been constructed to have different percentages of cancelation examples. To do so, the original signal, which contains additive symmetrical components, has been sampled unequidistantly. This way of sampling was chosen to ensure easy control over the percentage of cancelation examples in the training set. Secondly, the cancelation criterion has been used to predict the learnability of example sets with various cancelation content. An empirical confirmation on learnability and learning reliability of these example sets has been made by repeatedly training the same network with random variations of these sets.

A general conclusion from this experiment is, that every signal contains a certain level of cancelation, and its learnability and the reproducibility of the training results is pro-
portional to this level. This gives reason to believe that the learnability and respectively learning reliability of an arbitrary signal can be improved, if its cancelation potential is decreased.

The unequidistant example selection scheme used in the previous section is not the best choice for enhancing the learnability of a signal, since it does not always provide a representative training set. An unequidistant sampling method, which will catch the idiosyncrasies of the training signal, will be inevitably problem-dependent. To avoid this dependence, an equidistant sampling will be a basic assumption for creating a working sampling strategy.

The equidistant sampling scheme does not allow the content of cancelation examples in the training set to be changed. Instead, we suggest that reordering of the training examples can help avoiding the cancelation effects and thus improve reliability of the learning process.

Previously it has been suggested to overcome the reliability problems in neural networks by optimizing the itinerary on the error surface. The reliable learning is related to the degradation of the network effective capacity. Training with cancelation signals has been found to cause this degradation. Cancelation signals reveal the limitations of performing a gradient optimization with randomized data sets on stationary error relief areas. They cause additional difficulties by escaping shallow valleys, minima, and flat plateaus. In such areas the learning algorithm progresses with very small steps in general. Since the impact of the teacher signal, the major optimization factor, is minor in those signals, optimization goes to the lowest local point in shallow valleys and local minima (see figure 3-10b). In a plateau, the learning process progresses as directed by the input samples since the training is effectively performed without a teacher signal. If these input samples are given to the network in an arbitrary order, the itinerary will follow the lows of a random walk. Being in a flateness (valleys as well as plateaus are flatenesses with different dimensionality) means, that most of the network parameters are almost zero – the network has degraded to a smaller one.

Reordering of the example stream can ensure that stationarities are crossed in an observable time limit by forcing the learning trajectory to go in the right direction. Since there is actually no knowledge which is the direction that leads faster to an optimal solution, an alternative approach is taken. To remedy the first situation (i.e. escaping from curvaceous shallow valley region) the training steps should direct the itinerary to escape the valley. A logical way to do this is by increasing the effective length of the random passage by forcing several adaptation steps to be in the same direction. The same approach is shown to be helpful for escaping flatnesses. Since the learning itinerary of a cancelation signal in such an area follows the rules of a random walk, traversing the plateau faster is possible when the step length of a random walk is larger (see appendix B).

The cancelation criterion gives an indication about the directedness of the learning trajectory. Calculated over the entire training set it gives an approximately constant end value. The reordering of the training examples will rather change the KRS-ratio development and this way help traversing the stationarities. On basis of this criterion a sampling strategy aiming to overcome the reliability problems in the suggested manner will be constructed. In accordance with the reasoning made so far, there is an unified way
of going through the main difficult relief forms such as shallow valleys, local minima, and flat plateaus, which is the guideline for defining our resampling strategy.

The two basic principles, which will form our example selection strategy are: (a) to overcome the limitations which cancelation puts on the learning history progress by giving it a certain direction; and (b) to preserve the advantages of the randomized gradient optimization, in accordance with basic neural trade-off principles, described in the KRS paradigm.

4.3.1 Windowed sampling strategy.

The ways to increase the KRS-ratio are searched in correspondence with the cancelation criterion and the reasoning made so far. Since the KRS-ratio has been developed on basis of the presence of cancelation tuples in a signal, the first reliability improvement condition is to avoid their appearance. Moreover, cancelation examples can be avoided to be present at all in small periods of time. Taking for illustration purposes the frequently recurring example with a symmetrical function, the intuitive way of avoiding cancelation is by selecting the training examples from an interval of a signal, which has only a growing or decaying tendency. In order to fulfill the second condition (i.e. preserving the advantages of random pattern selection) the examples within this interval have to be selected in an arbitrary order.

For higher randomness of the training process the beginning of the training interval can also be chosen at random. Let the training set be scaled in the interval \( s_0 = [-1, 1] \). If the examples are drawn randomly from the training set, the selected example \( z^0 \) can be any member, belonging to the interval \( s_0 \). The equal probability of the example \( z^0 \) to be selected can be denoted by \( p(z^0) \). Dividing the interval \( s_0 \) in subintervals by selecting a random value \( p(z^0) \) and the following \( n_i \) examples – belonging to the subinterval \( s_i = [z^i, z^i + C] \) corresponds to the suggested idea that the training examples should be chosen to destroy the cancelation in the example set, as presented in figure 4–10. By analogy, it can be written:

\[
\begin{align*}
z^0 &= p(z^0), z^0 \in s_0 \\
z^i &= p(z^i), z^i \in s_i
\end{align*}
\] (4–28)

where \( s_0 = s_1 \cup \ldots \cup s_i \cup \ldots \).

In the case of on-line learning, which is the subject of this work, the weight corrections, corresponding to every example, do not exactly follow the negative gradient direction. If the training patterns are selected randomly, the search direction steps oscillate around the exact gradient direction, and, on average, the algorithm implements a form of descent in the error function. The fundamental reason for using on-line training is that adding some noise to the gradient direction can help to escape from some shallow local minima.
The subinterval pattern selection method is going to increase the positive effect of on-line training, because of the already explained considerations. Presenting the training examples in this way will certainly prevent degradation. It is interesting to try alternative ordering schemes. It can be expected that the effect of escaping stationary areas can be achieved also when the samples within a subinterval are trained in their subsequent order. It is interesting to find the optimal ratio between randomness and ordering for optimal learning. Here only a series of experiments will demonstrate some dependencies. Some results of applying the described sampling methods are shown later in this chapter.

Supplying the training examples in the described way gives very good results, but its performance can not be compared easily when other methods for sampling has been applied. When choosing the beginning of a new interval at random and training with the permuted examples within this interval, it can not be said when a training epoch has passed. Some training areas has been overlapped by the sample selection method, and it can not be said when all the samples of the entire training set will be taught. Thus the error measure can not be compared with random or other methods of sampling.

Aiming to find a way of comparing the results of subinterval training with random sample selection, a sampling strategy will be implemented, which covers the complete sample space for the number of iterations, contained in an epoch.

4.3.1.1 Formalization of sample selection techniques.

The new example selection methods should not contradict the used optimization paradigm. Accordingly, the RMS error is defined after the assumption that every training example is presented exactly once during one training epoch. To be able to compare the results obtained by random example presentation, this rule is not violated by the new ways of selecting the examples. The new training sequences are constructed mainly by two levels of randomization. Firstly, the training set has been divided into non-overlapping intervals. The examples for training have been selected by choosing at random either the elements in an interval, or the number of the current training interval or both of them. Correspondingly, there are four example selection possibilities that ensure that every example has be selected exactly once: the examples has been presented in the natural order of sampling of the training signal, which will be referred to as Ordered Exam-
ple Selection (OS); the training examples has been selected at random from the complete training set - Random Example Selection (RS); the training examples follow their natural order of occurrence in the training signal, but the number of the interval from which they are selected is arbitrarily chosen - Ordered Example Selection from Randomly Selected Intervals (OSRI); both, the selection intervals as well as the examples within them are selected at random - Random Example Selection from Randomly Selected Intervals (RSRI).

- **Ordered Example Selection (OS)**
  
  Let the \( n \) available training examples be contained in the set \( Z = \{ z^1, z^2, \ldots, z^n \} \), then the examples will be presented in the order in which they appear in this set. The presentation set will therefore be \( P = Z \).

  Example:
  
  For the set \( \{0, 3, 5, 8\} \) the OS presentation will be \( \{0, 3, 5, 8\} \).

- **Random Example Selection (RS)**
  
  Let us define \( \text{Perm}(Z) \) as the permutation of the elements of \( Z \), then the presentation set will contain a fully permuted set of elements of \( Z \). The presentation set will therefore be \( P = \text{Perm}(Z) \).

  Example:
  
  For the set \( \{0, 3, 5, 8\} \) a RS presentation can be \( \{5, 3, 8, 0\} \).

- **Ordered Example Selection from Randomly Selected Intervals (OSRI)**
  
  For the purpose of windowed selection, the set \( Z \) will be subdivided into a set of intervals \( K = \{k_0, \ldots, k_{n-1}\} \). Let for simplicity assume that all intervals contain the same amount of examples, then \( k_j = \{z^{j}, z^{j+1}, \ldots, z^{j+m-1}\} \), with \( j = 0, \ldots, m - 1 \). Let \( I_{n} = \{0, \ldots, m-1\} \) and \( O = \text{Perm}(I_{n}) \) be the permuted intervals. Further, all intervals are mutually non-overlapping. The presentation set will therefore be \( P = k_{0_{0}} \bigoplus k_{0_{1}} \bigoplus \cdots \bigoplus k_{0_{m-1}} = \bigoplus_{i=0}^{m-1} k_{0_{i}} \) where \( a \bigoplus b \) denotes the concatenation of the sets \( a \) and \( b \).

  Example:
  
  For the set \( \{0, 3, 5, 8\} \) an OSRI presentation can be \( \{5, 8, 0, 3\} \).

- **Random Example Selection from Randomly Selected Intervals (RSRI)**
  
  An (RSRI) is a (OSRI) where also the elements of \( k_j \) are permuted. The presentation set will therefore be: \( P = \bigoplus_{i=0}^{m-1} \text{Perm}(k_{0_{i}}) \).

  Example:
  
  On the set \( \{0, 3, 5, 8\} \) RSRI makes the following changes. First, the intervals \( k_i \) are permuted and concatenated \( \bigoplus_{i=0}^{m-1} \text{Perm}(k_{0_{i}}) \), for instance in the follow-
ing way \([5, 8, 0, 3]\). Second, the elements within every interval are permuted \(\text{Perm}(k_o)\). For instance as: \([8, 5, 0, 3]\), in this case \(\text{Perm}(5, 8) = \{8, 3\}\) and \(\text{Perm}(0, 3) = \{0, 3\}\).

4.3.1.2 Experiments with sample selection orderings

The (dis--) advantages of the discussed sample selection techniques can be illustrated in the training of a very simple signal. A cosine puls is sampled equidistantly to train a network with symmetrical nodal transfer (Figure 3–13). The set of training examples is ordered in different ways: random, with partial orderings, in the natural order. In figure 4–11 these ordered training set presentations are given.

![Graphs showing different sample selection orderings](image)

Figure 4–11: Part of example sets, constructed by different ways of selection. Connected lines correspond to the target values.

The experiments are made with two types of semi-ordering of the training set. In the first one the beginning of any interval is chosen in an arbitrary order. The examples within this interval are supplied to the network randomly. In the second manner of semi-ordered sampling only the beginning of the intervals are chosen at random. The values in the interval are presented in their subsequent order. The experiments are made with example intervals of different lengths. The training is continued until the summary approximation error falls below 0.01. With every different randomization (except by the completely ordered presentation) 100 experiments are performed.
The different example selection strategies lead to variation in the approximation quality. The training duration varies widely as well. As discussed in section 2.3.3.1, a good measure for the learning reliability in terms of the probability to converge to a proper solution is just to count the successful trials. The percentage of successfully converging experiments facilitates a global comparison between the different possible randomization strategies. The outcome of the empirical investigation is that all the experiments, performed with RS reordering scheme has failed, while the experiments with OS, OSRI and RSRI reorderings has reached the stated approximation level.

For further comparison of the obtained results, different aspects of learning reliability has been considered. Since the comparison should be made between successive experiments, the time, i.e. the period over which a certain degree of performance can be achieved has been evaluated. For this purpose two measures are used: the median convergence time and its variation. The combination of these two values gives a fair idea of how long training should approximately last. In figure 4–12, the mean convergence time and the variance of the pattern presentations with partial orderings are presented.

Ordered selection within random intervals  Random selection within random intervals

![Graphs](image)

Figure 4–12: Mean and variance graphs of runs of 100 interval experiments.

A low variance of the training duration for different runs of the experiment is considered to indicate a better solution with respect to the reliability of the network performance. From the point of view of computational costs, better is the fastest solution. For both
manner of pattern presentation, there are zones with optimal combination of mean-
variance. For the random interval selection and ordered patterns within the intervals,
the best interval sizes are empirically obtained as 50–150. For the random–random
selection of examples, the best intervals measure 20–150 in length. In percentages, this
interval length take from 5 up till 15 % of the overall input range.

It is interesting to show the correlation between the outcome of the performed 
experiments and the KRS predictions. Figure 4–13 shows the KRS—ratios for the selection
of ordered samples within randomly selected intervals (OSRI) when training a cosine
puls. In general, KRS values higher than 0.1 correspond to the solution with a small
number of training iterations (see the corresponding values of figure 4–12a) and low
variance of the obtained results (compare to figure 4–12c). An empirical confirmation
about the applicability of the cancelation criterion for example selection are the low
KRS values for training interval sizes of about 200 examples, as well as by very small
interval sizes. In both cases the outcomes of training are also worse, as illustrated in
Figure 4–12a and c. Training with very small interval sizes shows bad results because
it is similar to training with random examples. For the minor performance by interval
sizes of 200 examples we do not have a very good explanation. It can be due to either
numerical effects or the low number of experiments for a reliable statistical conclusion.

![Figure 4-13: Mean KRS values for OSRI training with different interval sizes.](image)

The improvements to the learning of a fully symmetric function are impressive, but it
remains to be verified whether such improvements will also appear in practice. There-
fore it is interesting to evaluate the effect of the shown manners of example presentation
on the non-cancelation signal. In figure 4–14, the outcome of training a non-cancel-
ation signal with the four selection schemes is shown. Although all the experiments are
successful, the training duration of the OSRI experiments is much shorter than the RS
ones. This observation suggests that the used alternative sampling strategy can increase
reliability in different learning scenarios.
4.3.2 Summary and further suggestions.

This chapter discusses different sampling and example reordering techniques which can provide for enhanced learnability and reliability of various signals. As an indicator for training problems the cancelation criterion has been established. Enhancing the learnability and reliability when training a particular signal is shown to be in accordance with increasing its cancelation ratio.

So far several methods of handling a signal are shown to increase the KRS-ratio and to improve the reliability of the learning process, respectively. The simplest among them is to move the window, from which training samples are drawn, over a signal. Obviously, the content of the cancelation examples differs a lot in simple signals. In general, this method can not be applied by an arbitrary signal for two major reasons. First, when the signal complexity grows, the level of cancelation might not decrease if different historical data is used for training. Second, moving the sampling window over an unknown signal is a measure with unpredictable effect, and one can not count on reliable performance.

The developed sampling strategy suggests alternative ways for handling cancelation signals by dividing of the training set on intervals, within which the extracted samples do not have a high cancelation potential. This idea has been suggested by the analysis of the training process development when the cancelation potential sets in. Problematic for neural training in general, and especially when cancelation in the training examples is high, are the equalities in the interaction between neural system and training task. Analytically, they correspond to (approximate) singularities of the Hessian matrices, and can be represented graphically as very flat and very curvaceous areas. Since the number of cancelation examples can not be changed in general, the basic idea underlying the suggested example reordering method is to select cancelation free examples for training periods shorter than an epoch.

This second method for coping with cancelation gives positive results in many learning situations. It remedies reliability problems in simple shaped cancelation signals, and helps improving the network performance in more complex cases.

There are a number of specific learning scenarios which turn out to have cancelation nature. The most typical are: second-order problems [104], approximation of poly-
mials [135], and periodical signals [10]. For some of them (notably periodical signals), there are additional risks for network degradation to occur. Therefore the approach of handling simple cancelation signals has to be extended to cope with periodicity as well. Simple example reordering methods such as moving the selection interval of the training samples will not help because periodicity keeps the cancelation content of the examples at a high level. The method for reliability enhancement, which implies division of the training set on subintervals might be insufficient as well, because the cancelation content either within an interval or between intervals can stay high due to periodicity. The exact reasons for the learning problems introduced by periodicity will be elaborated on in the following chapter. Here it will be explained why and how the example stream can be changed further on, if its division on subintervals is insufficient for reliable training.

The starting point for deriving the cancelation criterion and developing the subinterval selection strategy has been equation (4-9). It describes the effect of historical data on weight changes in the network state, since the weight changes are formed mostly by the stream of examples, used for training up until that moment. Reliability problems are characterized by degradation of the effective capacity of the network. There are two variations of this phenomenon, which in terms of the parameter space result either in nullation of most of the network parameters, or in their equalization – all the parameters take few (the same) parallel trajectories and do not change any further. There are several possibilities for degradation to occur. All of them are due to the features of the target stream of examples. The symmetry of the network and the network transfer as well as the contribution of the input stimuli has been explained as well. Furthermore, there are restrictions posed by the arithmetic precision which burden additionally network indetermination and cancelation situations, respectively.

To alleviate the training process, the following reasoning will be made. Equation (4-9) gives the time development of an arbitrary hidden-to-output weight. Let us consider the case of cancelation signals, when either all or the majority of hidden-to-output weights degrade to a zero value. This causes that the calculated gradient value, which will be used for further adaption is very small. Since there are two phases of computation, the internal state of the network depends on what has been given as an input during the forward passage as well as the difference between the target and output value, which has been transmitted back during the backward phase. Consequently, if we intend to control the learning process, it is of equal importance to follow the target as well as the input stream of values. This statement is inferred (self-understood) in classification tasks, since various features are forming the input streams of examples. On the other hand, in approximation and prediction tasks the input stream has a lower information content than the target stream. Here, often the time information (the x-coordinate) of a signal is used as input. Beside from the low information content, such an input stems from decreased differentiability (increased symmetry) of the learning system. The following example explains that.

Let a signal from which 2000 samples have been extracted be scaled for neural training. The sigmoid transfer requires scaling range between [0, ..., 1]. This way the difference between two consequent input examples is 0.0005. Furthermore, the training strategy which implies division of the training set on subintervals forces training examples with very close input values to be supplied in a sequential order. This increases the necessity to ensure that the input stream is structured in a proper way.
The current training example causes the change of any hidden-to-output parameter value as follows:

\[ A_{\omega_\text{oj}} w_{ij} = \delta_z(t) o_j(t) = \delta_z(t) \theta_i(\sum w_j x_i) \]  
(4–29)

Without loss of generality, an one input network can be considered. For such a network equation (4–29) will take the form:

\[ A_{\omega_\text{oj}} w_{ij} = \delta_z(t) \theta_i(x_i) \]  
(4–30)

The corresponding change of the input-to-hidden parameters can be expressed by the equation (4–31):

\[ A_{\omega_\text{oi}} w_{i} = \delta_z(t) o_i(t) = \sum \delta_j w_{ij} \theta_j'(o_j)x_i(t) \]  
(4–31)

In both cases \( o_i(t) \) can directly be substituted by \( x_i(t) \), because the input neurons are linear. The dependence of both adaptations (4–30) and (4–31) on the target part of the example is implied in the \( \delta_z \) term. The input part of the current example is \( x_i(t) \).

Attempting to control the network state by the input examples, we can increase the difference between the two consequent \( x \)-examples. Obviously, this will contribute increasingly to the summation of the consequent adaptation values as well as to the input-to-hidden as to hidden-to-output parameters. Increasing the difference between two consequent inputs enhances the differentiability of the input-output space in general. The exact way to do that depends on the training problem: the reconstructions in the input stream should not introduce contradictions with what has to be learned. The exact manners for differentiability enhancement of the input stream will be discussed in chapter 5.