Regularized Greedy Algorithms for Neural Network Training with Data Noise

M. Burger*
Department of Mathematics, University of California,
520 Portola Plaza, Los Angeles, CA 90095, USA

A. Hofinger*†
SFB Numerical and Symbolic Scientific Computing,
Johannes Kepler University Linz, Freistädterstraße 313,
A-4040 Linz, Austria

Abstract

The aim of this paper is to construct a modified greedy algorithm applicable for an ill-posed function approximation problem in presence of data noise. This algorithm, coupled with a suitable stopping rule, can be interpreted as an iterative regularization method. We provide a detailed convergence analysis of the algorithm in presence of noise, and discuss optimal choices of parameters. As a consequence of this analysis, we also obtain results on the optimal choice of the network size in presence of noise.

Finally, we discuss the application of the modified greedy algorithm to sigmoidal neural networks and radial basis functions, and supplement the theoretical results by numerical experiments.

Keywords: Greedy Approximation, Data Noise, Neural Networks, Regularization, Training

AMS Subject Classification: 41A46, 47A52, 41A65, 92B20

1 Introduction

Function approximation by neural networks and similar techniques has received growing attention in the last decade, in particular due to its (almost) dimension-independent approximation properties (cf. [2, 11]). These nonlinear approximation techniques are not only able to approximate large classes of functions arbitrarily well as the number of nodes (respectively parameters) tends to infinity, but also yield a sequence of functions f_n approximating the original function f, such that

$$\|f - f_n\| \leq Cn^{-\frac{1}{2}}, \tag{1.1}$$

[*] This work has been supported by the Austrian National Science Foundation FWF under project grant SFB F 13/08

[†] Corresponding author, e-mail: andreas.hofinger@sfb013.uni-linz.ac.at

where n denotes the number of nodes in the network (cf. [2, 7]) and C is a constant depending on the function f to be approximated only.

The pay off for this nice convergence behaviour is the necessity to compute global minimizers of rather high-dimensional nonlinear optimization problems (in particular for large n) in order to obtain the approximating functions f_n. In the context of neural networks, the so-called *backpropagation algorithm* is the most popular approach to solve the arising optimization problems (called training), mainly due to its simple realization. But since the backpropagation algorithm is a version of the gradient method *steepest descent* one cannot expect global convergence.

Moreover, the performance of such iterative algorithms is limited by the inherent ill-posedness of the training problem (cf. [4]). It has been demonstrated recently that mainly two sources of ill-posedness exist in the training of neural networks and similar nonlinear approximation techniques in presence of noise:

- *Asymptotic instability*: this type of instability arises with number of nodes tending to infinity. Under typical conditions, data noise can be modelled as an L^2-perturbation, so that (due to compact embedding) the problem of approximation with respect to Sobolev norms or the supremum norm is ill-posed (cf. [4] for a detailed discussion).

- *Nonlinear instability*: this type of instability arises even if the number of nodes is kept fixed. Due to the nonlinearity it is possible to construct objective functions which are arbitrary small but correspond to arbitrary large parameters. We refer to [3] for an illustration of this instability in the context of fuzzy control and to [10] for sigmoidal neural networks.

In typical applications, the noise is caused by two main reasons, namely by output measurement error and by partially missing measurements in some regions. The error introduced by the latter effect is usually called *generalization error* and analyzed by stochastic methods. In this paper we focus on the first type of error (output noise), leaving a generalization of our approach to the second type for future research.

The ill-posedness of the approximation problem raises several key questions, which are only answered in part at the current stage of research:

- Applicability of standard regularization methods: Tikhonov-type regularization methods have been analysed recently (cf. [6, 3]), but they still require the solution of nonlinear optimization problems similar to the original problem in the training of neural networks.

- Regularizing properties of iterative methods: general results on iterative methods for ill-posed problems (cf. [8] and the references therein) show that it is fundamental to choose the stopping index of any iterative method in dependence of the noise level in order to obtain regularizing effects. Stopping rules and convergence properties for iterative methods have not been obtained so far in the context of neural networks and nonlinear approximation.

- Choice of the number of nodes: the data noise clearly limits the approximation capabilities and therefore it seems reasonable that the number of nodes in the network should be chosen in dependence of noise, a topic hardly treated so far.

In this paper we investigate an interesting form of iterative methods, so-called *greedy algorithms* (also called *projection pursuit* or *convex approximation techniques*, cf. [7, 12]), and investigate there regularizing properties. The main idea of such algorithms is to increase the number of nodes in the network step by step (by using suitable convex combination) and to optimize only over the parameters of the new node, which yields a sequence of low-dimensional optimization problems. The original motivation for such methods is the possibility to maintain the convergence rate $n^{-\frac{1}{2}}$ with low computational effort. As we shall see below, such methods can also be considered as iterative regularization methods if an appropriate stopping rule in dependence of the noise is used. Since the iteration index in greedy algorithms is directly related to the number of nodes in the network, this also provides an answer to the optimal choice of the network size in dependence of the noise level.

The paper is organized as follows: first of all some results about convex approximation are presented in a very general and abstract setting in Section 2, where we also present the original greedy algorithm. In Section 3 we investigate the influence of noise in the data and present a modification of the greedy algorithm together with a stopping rule, which leads to convergence of the regularized approximations. In Section 4, we apply this algorithm to the training of neural networks and to radial basis function networks, and give results about convergence properties in stronger Sobolev norms. Finally, the results of numerical experiments are given in Section 5.

2 Greedy Approximation

In this section we give a short review on greedy algorithms for function approximation and sketch some of the main ideas in their analysis in the noise-free case, in order to provide some insight fundamental for the later convergence analysis in presence of noise.

Greedy approximation denotes an iterative algorithm for training a neural network, which realizes the dimension-independent convergence properties and can be implemented efficiently. The greedy algorithm we shall present increases the size of the network step by step by one neuron . In each step of the iteration we will seek for a neuron that approximates the objective almost optimally (with respect to the accuracy that can be obtained with the current number of nodes), therefore we will call this procedure *greedy algorithm*.

2.1 Preliminaries

In the following let G be a subset of an inner product space H with induced norm $\|\cdot\|$. Furthermore let the elements of G be bounded in the norm by some constant B, which may be abbreviated as $G \subset \mathrm{B}(0; B)$, and let $\overline{\mathrm{co}}(G)$ denote the closure of the convex hull of G.

We assume that the function f to be approximated is an element of $\overline{\mathrm{co}}(G)$. For the further analysis we define the constant γ via

$$\gamma = \inf_{v \in H} \sup_{g \in G} \left(\|g - v\|^2 - \|f - v\|^2 \right). \tag{2.1}$$

This value is in some sense a measure for the number of different elements of G that are needed to represent f. If the norm of f is close to the bound B and

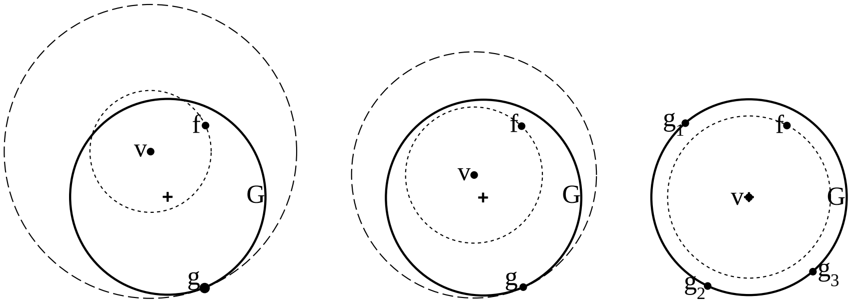


Figure 1: Interpretation of condition (2.1). The objective function is the difference of the radii of the two dashed circles. In this symmetric case the infimum is attained for v lying in the center of G.

therefore f is close to the boundary of $\overline{\text{co}}(G)$ the value of γ will be very small. In this case f can be represented by few different elements of G (cf. Figure 1). Note that the value of γ can be bounded from above by $B^2 - \|f\|^2$ since $0 \in H$.

The constant γ provides an estimate for the rate of convex approximation, as we will see in the following lemma [7, Lemma 2]:

Lemma 2.1. *Let $G \subset \mathrm{B}(0; B)$, $f \in \overline{\text{co}}(G)$, $h \in \text{co}(G)$ and let γ be defined via (2.1). Then the estimate*

$$\inf_{g \in G} \|f - \lambda h - (1-\lambda) g\|^2 \leq \lambda^2 \|f - h\|^2 + (1-\lambda)^2 \gamma$$

holds for $\lambda \in [0, 1]$.

2.2 The Greedy Algorithm

The greedy algorithm in the noise-free case is defined iteratively by the following procedure (cf. [7]):

Algorithm 2.2 (Greedy Algorithm). Greedy approximation in the noise free case

```
Initialization:
    Choose a constant $M$, such that $M > \gamma$ (as defined in (2.1)).
    Choose a positive sequence $\varepsilon_k$, tending to zero that fulfills
        $\varepsilon_k \leq \frac{M - \gamma}{k^2}$    for    $k = 1, 2, \ldots$
    Set $f_0 = 0$.
Iteration:
    for $k :=$ to maxit do
        Find an element $g_k \in G$ such that
            $\left\| f - \frac{k-1}{k} f_{k-1} - \frac{1}{k} g_k \right\|^2 \leq \inf_{g \in G} \left\| f - \frac{k-1}{k} f_{k-1} - \frac{1}{k} g \right\|^2 + \varepsilon_k$
        is fulfilled and define $f_k$ as
            $f_k = \frac{k-1}{k} f_{k-1} + \frac{1}{k} g_k$ .
    end for
```

Note that in each step only *one* element of G is chosen, the other components of f_k are fixed. Nevertheless with the greedy algorithm still the dimension independent convergence rate is obtained, as can be seen in the next theorem (cf. [7]):

Theorem 2.3. *Let the conditions of Lemma 2.1 be satisfied. Then the approximating functions f_k generated by Algorithm 2.2 fulfill the error estimate*

$$\|f - f_k\|^2 \leq \frac{M}{k} \,. \tag{2.2}$$

Proof. The proof is based on an induction argument, using Lemma 2.1 and can be found in [7]. In Section 3.2 we present a modified version of this proof, dealing with noisy data. □

If instead of f, we only know a perturbed function $f^\delta \in L^2(\Omega)$ (with noise of level δ), then Algorithm 2.2 and Theorem 2.3 cannot be applied to the approximation problem since the perturbation f^δ is not necessarily an element of $\overline{\text{co}}(G)$ anymore. In the next section we shall modify the greedy algorithm so that data noise can be handled, and investigate the convergence properties of the modified method.

3 Greedy Algorithms in Presence of Noise

Now we modify the greedy algorithm such that it can be applied also for noisy data. Since we cannot guarantee that f^δ also lies in the closed convex hull of G, we have to introduce the projection of f^δ onto $\overline{\text{co}}(G)$.

The modification that we will finally find for the greedy algorithm from the section before is a *stopping rule*, i. e., a rule that tells us after how many iterations k we have to terminate the algorithm. This is a typical result for iterative regularization methods (see [8, Chap. 6]), here the iteration index plays the same role as the regularization parameter α in Tikhonov regularization, the stopping rule corresponds to the parameter selection method.

In general the stopping rule is a function of the noise level δ, but it may also depend on the noisy data f^δ. The rule we present is an *a-posteriori stopping rule* and is a function of the noise level δ and the projection Pf^δ.

3.1 Projection onto the Closed Convex Hull

The projection is defined as the element in $\overline{\text{co}}(G)$ which is closest to f^δ. Since $\overline{\text{co}}(G)$ is by definition convex, this element is *unique.* We denote the (nonlinear) operator that maps a function to its projection with P and have the equation

$$\|f^\delta - Pf^\delta\| = \inf_{h \in \overline{\text{co}}(G)} \|f^\delta - h\|.$$

For function approximation the projection has useful properties which will be needed later. It should be mentioned that we will need all these properties only inside the proofs but not in the algorithm. The projection itself never has to be computed, only the related value γ^{Pf^δ} (see below) must be estimated.

The orthogonality of $f^\delta - Pf^\delta$ (cf. e.g. [15, Chapter 5.3]) to $\overline{\mathrm{co}}(G)$ implies that elements f_n^δ of $\overline{\mathrm{co}}(G)$ that approximate f^δ are even better approximations to Pf^δ, i.e.

$$\left\|Pf^\delta - f_n^\delta\right\| \leq \left\|f^\delta - f_n^\delta\right\|. \tag{3.1}$$

As in the noise free case we now define

$$\gamma^{Pf^\delta} = \inf_{v\in H} \sup_{g\in G} \left(\|g-v\|^2 - \left\|Pf^\delta - v\right\|^2\right), \tag{3.2}$$

which is a measure for how many different elements of G have to be used to represent Pf^δ. The value of γ^{Pf^δ} can be estimated in terms of the constant B, the norm of f, and the noise level δ via

$$\gamma^{Pf^\delta} \leq B^2 - (\|f\| - \delta)^2. \tag{3.3}$$

Lemma 2.1 together with an application of the triangle inequality yields:

Corollary 3.1. *Let G be as above, γ^{Pf^δ} defined by equation* (3.2) *and h be an element of the convex hull of G. Moreover, let f^δ be an arbitrary element of the Hilbert space H and Pf^δ denote its projection onto $\overline{\mathrm{co}}(G)$.*

Then the estimate

$$\begin{aligned}\inf_{g\in G} \left\|f^\delta - \lambda h - (1-\lambda)g\right\|^2 \leq \delta^2 + \lambda^2 \left\|Pf^\delta - h\right\|^2 + (1-\lambda)^2\gamma^{Pf^\delta} \\ + 2\delta\sqrt{\lambda^2 \left\|Pf^\delta - h\right\|^2 + (1-\lambda)^2\gamma^{Pf^\delta}}\end{aligned} \tag{3.4}$$

holds for all $\lambda \in [0,1]$.

Observe that the element f^δ does not necessarily belong to the closed convex hull of G but can be chosen arbitrarily in H.

3.2 The Greedy Algorithm for Noisy Data

Now we investigate the behaviour of the algorithm when applied to a function f^δ that is not necessarily in the closed convex hull of G, but whose distance to the convex hull is limited by the noise level δ. As we shall see below, for data with noise of level δ the convergence rate $\mathcal{O}(\frac{M}{k})$ can be obtained if we modify the choice

$$\varepsilon_k \leq \frac{M-\gamma}{k^2}$$

used in Algorithm 2.2 to

$$\varepsilon_k \leq \frac{M}{k} - \left(\delta + \sqrt{\frac{k-1}{k^2}M + \frac{1}{k^2}\gamma^{Pf^\delta}}\right)^2. \tag{3.5}$$

Note that for $\delta = 0$ these two choices coincide. This yields Algorithm 3.2, where the iteration is terminated when for the first time the right hand side of (3.5) becomes negative.

Algorithm 3.2. Greedy approximation in the case of noisy data

```
Initialization:
    - Choose a positive constant $M$ (see Method 3.4)
    - Determine the index $k_* = k_*(M, \gamma^{Pf^\delta}, \delta)$ at which for the
      first time the right hand side of (3.5) becomes negative and
      choose a positive sequence $\varepsilon_k$ that fulfills condition (3.5) for
      $k = 1, \ldots, k_* - 1$.
    - Set $f_0^\delta := 0$.
Iteration:
  for $k := 1$ to $k_* - 1$ do
      Find an element $g_k^\delta \in G$ such that
          $\left\| f^\delta - \frac{k-1}{k} f_{k-1}^\delta - \frac{1}{k} g_k^\delta \right\| \leq \inf_{g \in G} \left\| f^\delta - \frac{k-1}{k} f_{k-1}^\delta - \frac{1}{k} g \right\| + \varepsilon_k$
      is fulfilled and define $f_k$ as
          $f_k^\delta = \frac{k-1}{k} f_{k-1}^\delta + \frac{1}{k} g_k^\delta$ .
  end for
```

This modified algorithm maintains the convergence rate $\mathcal{O}(\frac{M}{k})$ up to the iteration index where it is terminated, as we shall see in the next theorem.

Theorem 3.3. *Let the conditions of Corollary 3.1 be satisfied. Then the approximating functions f_k^δ generated by Algorithm 3.2 fulfill the error estimate*

$$\left\| f - f_k^\delta \right\|^2 \leq \frac{M}{k} \quad \text{for} \quad k = 1, \ldots, k_* - 1 \,. \tag{3.6}$$

Proof. This proof is a modification of the proof of Theorem 2.3, and is based on an induction argument, using Corollary 3.1. We first look at the initialization step $k = 1$.

- For the step $k = 1$ we obtain using Corollary 3.1

$$\begin{aligned} \left\| f^\delta - f_1^\delta \right\|^2 &\leq \inf_{g \in G} \left\| f^\delta - g \right\|^2 + \varepsilon_1 \\ &\leq \delta^2 + \gamma^{Pf^\delta} + 2\delta \sqrt{\gamma^{Pf^\delta}} + \varepsilon_1 \\ &\leq M \end{aligned}$$

  since according to (3.5) ε_1 is chosen such that

$$\varepsilon_1 \leq M - \left(\delta + \sqrt{\gamma^{Pf^\delta}} \right)^2$$

  holds.

- Now we inspect the case $1 < k < k_*$. We assume that the convergence rate was preserved up to this step of the iteration, this means that the estimate $\left\| f^\delta - f^\delta_{k-1} \right\|^2 < \frac{M}{k-1}$ holds. From (3.1) we know that this estimate remains valid if we replace f^δ with Pf^δ.

$$\begin{aligned}
\left\| f^\delta - f^\delta_k \right\|^2 &\leq \inf_{g \in G} \left\| f^\delta - \frac{k-1}{k} f^\delta_{k-1} - \frac{1}{k} g \right\|^2 + \varepsilon_k \\
&\leq \left(\delta + \sqrt{\left(\frac{k-1}{k}\right)^2 \left\| Pf^\delta - f^\delta_{k-1} \right\|^2 + \frac{1}{k^2} \gamma^{Pf^\delta}} \right)^2 + \varepsilon_k \\
&\leq \left(\delta + \sqrt{\frac{k-1}{k^2} M + \frac{1}{k^2} \gamma^{Pf^\delta}} \right)^2 + \varepsilon_k \\
&\leq \frac{M}{k}
\end{aligned}$$

since ε_k is chosen such that

$$\varepsilon_k \leq \frac{M}{k} - \left(\delta + \sqrt{\frac{k-1}{k^2} M + \frac{1}{k^2} \gamma^{Pf^\delta}} \right)^2 . \qquad \square$$

For $\delta \to 0$, $k_* \to \infty$, which implies that the residual at the end of the iteration tends to 0. On the other hand for $k \to \infty$ the right hand side of (3.5) tends to $-\delta^2$, it becomes negative. Since ε_k has to be chosen larger than zero, the algorithm has to terminate after a number of steps, depending on the magnitude of M, δ and γ^{Pf^δ}. For a given function f^δ also the values of δ and γ^{Pf^δ} are fixed, and consequently the iteration index k_* for which ε_k becomes negative and the iteration terminates, depends only on the magnitude of M. In the next section we show how M can be chosen in an optimal way.

3.3 Optimal Parameter Choices

If we denote the index where ε becomes negative by k_*, then we can estimate the residual with

$$\left\| f^\delta - f^\delta_{k_*-1} \right\|^2 \leq \frac{M}{k_* - 1}.$$

For this reason it is desired to find a combination of M and the corresponding k_* such that the right hand side of this equation becomes minimal.

To find this optimal combination (for given values of δ and γ^{Pf^δ}) we can for instance use the software system *Mathematica*. First we look at the zeroes of equation (3.5) i. e., we investigate the equation

$$\frac{M}{k} - \left(\delta + \sqrt{\frac{k-1}{k^2} M + \frac{1}{k^2} \gamma^{Pf^\delta}} \right)^2 = 0 \, .$$

Since this equation is of degree 6 with respect to k and only of degree 2 with respect to M we will first look for solutions for M and search for the minimum with respect to k afterwards. This yields two solutions $M_1 > M_2$, of which only M_1 is of interest, since M_2 is not a solution for $k \geq 1$. The remaining solution is given as

$$M_1 = g + (2k-1)\delta^2 k^2 + 2\delta k^{\frac{3}{2}} \sqrt{g + (k-1)k^2\delta^2}\,. \tag{3.7}$$

For the solution M_1 we try to find the optimal value for k, this means we try to solve the minimization problem

$$\frac{M_1(\gamma^{Pf^\delta}, \delta, k)}{k} \to \min_k \,. \tag{3.8}$$

To find stationary points we differentiate the equation with respect to k and search for zeroes. *Mathematica* finds 6 stationary points, from which one is negative and two others are not real numbers. The remaining 3 stationary points depend only on the ratio of γ^{Pf^δ} and δ^2, for this reason we define the positive constant

$$\nu := \frac{\sqrt{\gamma^{Pf^\delta}}}{\delta} \tag{3.9}$$

to simplify the equations. For small values of ν (exactly for $0 \leq \nu < \frac{2}{3\sqrt{3}}$) two other solutions become complex numbers and so only a single solution remains. In general the solution k_3 seems to be a maximum of (3.8) and not a minimum, but we could not prove this.

To abbreviate the formulas for k_2 and k_3 we define

$$a = -2 + 27\nu^2 + 3\sqrt{3}\nu\sqrt{27\nu^2 - 4}$$

$$b = \frac{\nu}{\sqrt{6}} \sqrt{-4 - \frac{2\,2^{\frac{1}{3}}}{a^{\frac{1}{3}}} - 2^{\frac{2}{3}}\,a^{\frac{1}{3}} + 48\,\nu^2 + \frac{12\,\sqrt{6}\,\nu\,\left(8\,\nu^2 - 1\right)}{\sqrt{-2 + \frac{2\,2^{\frac{1}{3}}}{a^{\frac{1}{3}}} + 2^{\frac{2}{3}}\,a^{\frac{1}{3}} + 24\,\nu^2}}}\,.$$

With these auxiliary values we can express the three stationary points as

$$k_1 = \nu \tag{3.10a}$$

$$k_{2/3} = 2\,\nu^2 + \frac{\nu}{\sqrt{6}} \sqrt{-2 + \frac{2\,2^{\frac{1}{3}}}{a^{\frac{1}{3}}} + 2^{\frac{2}{3}}\,a^{\frac{1}{3}} + 24\,\nu^2} \mp b\,. \tag{3.10b}$$

As long as $\nu \geq \frac{2}{3\sqrt{3}}$ all these are positive and real valued. If ν decreases below this bound then only the solution k_1 remains.

Method 3.4. The results of this section lead to the following rule for determining the constant M in Algorithm 3.2:

- Calculate $\nu(\delta, \gamma^{Pf^\delta})$ as defined in (3.9).
- Compute the 3 different solutions for k and the corresponding values for M_i via (3.10) and (3.7)
- Choose the index i for which $\frac{M(k_i)}{k_i}$ is minimal and set

$$M_{\text{opt}} := M_i, \qquad k_{\text{opt}} := k_i\,.$$

Observe that the choice of k depends on δ and (via γ^{Pf^δ}) also on f^δ. Therefore the parameter $k = k(\delta, f^\delta)$ describes an *a-posteriori stopping rule.* With the choice for M and k proposed in Method 3.4 the greedy algorithm for noisy data, Algorithm 3.2 is a regularization method as we will see in the next theorem.

Theorem 3.5. *Let M and k be chosen according to Method 3.4. Then for decreasing noise level also $\frac{M_{\text{opt}}}{k_{\text{opt}}}$ tends to zero. The convergence rate is given as*

$$\frac{M_{\text{opt}}}{k_{\text{opt}}} = \mathcal{O}\left(\delta^{2/3}\right).$$

Furthermore the approximations $f^\delta_{k_{\text{opt}}}$ obtained according to Algorithm 3.2 fulfill the rates

$$\left\| f^\delta - f^\delta_{k_{\text{opt}}} \right\| = \mathcal{O}\left(\delta^{1/3}\right) \quad \textit{and} \quad \left\| f - f^\delta_{k_{\text{opt}}} \right\| = \mathcal{O}\left(\delta^{1/3}\right).$$

Proof. The first rate can be shown by an asymptotic analysis of the solutions k_1 to k_3 and the corresponding values for M (see also Figure 10 and the corresponding comments in Section 5.3). The rate for $\left\|f^\delta - f^\delta_k\right\|$ now follows immediately from Theorem 3.3 and since $\left\|f - f^\delta\right\| \leq \delta$ the last rate is a consequence of the triangle inequality. □

The stopping rule presented in Method 3.4 cannot be improved if we use the estimate (3.4), but for special architectures it might be possible to obtain a sharper estimate than (3.4) and consequently a better stopping criterion. Deriving such estimates is one focus of future work.

4 Applications

In this section we investigate applications to two important approximation schemes, namely ridge construction type neural networks and radial basis functions. Since many properties are similar for these two approximation schemes we will state our results such, that they are valid for both methods. We do this by introducing the *activation function* Φ. A "feed forward neural network with one hidden layer" can be represented by

$$f_n = \sum_{i=1}^{n} c_i \sigma(a_i^T x + b_i)$$

whereas an approximation scheme using radial basis functions is given as

$$f_n = \sum_{i=1}^{n} c_i \Xi(\|x - t_i\|^2)$$

hence both methods can be written as

$$f_n = \sum_{i=1}^{n} c_i \Phi(x, t_i)$$

with an appropriate function Φ. In Sections 4.2.1 and 4.3.1 we show explicitly how the function Φ corresponds to these approximation schemes and impose natural assumptions on this function.

4.1 Notations

In order to apply the results of the previous sections we concretize the abstract setting using the activation function Φ. The natural Hilbert-space H seems to be the Lebesgue-space $L^2(\Omega)$. As subset G we choose the set

$$G_B = \{c\Phi(\cdot,t) \mid |c| \leq B,\, t \in P\} \subset L^2(\Omega)\,,$$

where Φ is the activation function of the network and P is the compact set of parameters. The set G_B can be interpreted as the set of all possible nodes of the network. If the function Φ is scaled such that its L^2-norm is bounded above by 1 uniformly in t, i. e.,

$$\int_\Omega |\Phi(x,t)|^2 d\,x \leq 1 \qquad \forall t \in P\,, \tag{4.1}$$

then G_B is bounded and $G_B \subset \mathrm{B}(0;B)$. For the sake of simplicity we assume (4.1), otherwise one can use the bound $\tilde{B} = B/\sup_{t\in P} \|\Phi(\cdot,t)\|_{L^2(\Omega)}$ for the factor c to bound the norm of the elements of G_B by B. Observe that $\tilde{B}$ can not be zero because the set P is compact. The value γ is now given as

$$\gamma = \inf_{v\in L^2} \sup_{|c|\leq B,\, t\in P} \|c\Phi(x,t) - v\|^2 - \|f - v\|^2\ . \tag{4.2}$$

The convex hull of the set G_B is defined as

$$\mathrm{co}(G_B) = \{f \in L^2(\Omega) \mid f = \sum_{i=1}^{n} c_i\Phi(\cdot,t_i),\, \sum|c_i| \leq B,\, t \in P, n \in \mathbb{N}\},$$

the sign of the parameters c_i does not matter, because the original set G_B is symmetric. Further the sum need not be equal to B but can also be smaller, because the zero function is an element of G_B. If we compute the closure of this set the sums turn into integrals and we find

$$\overline{\mathrm{co}}(G_B) = \{f \in L^2(\Omega) \mid f = \int_P \Phi(\cdot,t)d\mu(t),\, \mu \in \mathcal{M}_1,\, \|\mu\|_{\mathcal{M}_1} \leq B\}. \tag{4.3}$$

Here $\mathcal{M}_1$ denotes the set of all *Radon measures* (see e. g. [13]). Note, that $\overline{\mathrm{co}}(G_B)$ contains all functions having a representation of the form $\sum^n c_i\Phi(\cdot,t_i)$ and $\int_P c(t)\Phi(\cdot,t)\,dt$.

In order to apply Algorithm 2.2 to neural networks, it is necessary that the function f that shall be approximated is an element of the closed convex hull $\overline{\mathrm{co}}(G_B)$. For a given function f this can often be assured by choosing B sufficiently large. Nevertheless, to obtain good convergence rates (i. e., a small value for M) it is preferable that the norm of f is close to B. This can only be ensured by a proper choice of the activation function Φ depending on the specific structure of f.

In the following we present two typical choices for the activation function Φ, namely ridge construction type neural networks and radial basis functions.

4.2 Sigmoidal Neural Networks

4.2.1 Assumptions

Standard Assumptions 4.1. In a ridge construction neural network the activation function Φ fulfills the assumptions:

- $\Phi(x,t)$ is a function of form

$$\Phi(x,t) = \sigma(a^T x + b) = \sigma(\zeta_1 + \cdots + \zeta_n), \tag{4.4}$$

  with the new variables $\zeta_1 = a_1 x_1 + b$ and $\zeta_i = a_i x_i$ for $i \neq 1$.

  The values a_i and b_i correspond to the parameters t_i and are only introduced to emphasize the different meanings of the various parameters.

- The factors a_i are chosen such that the vector $(a_1, \ldots, a_n)$ is an element of some compact set that does not contain 0.

- the function σ does not have the form

$$\sigma(\xi) = Ce^{\alpha\xi} \qquad \text{or} \qquad \sigma(\xi) = (\alpha\xi + \beta)^\gamma$$

  for any combination of real numbers α, β and γ.

- the function σ is twice continuously differentiable.

Remark 4.2. The activation function σ is not allowed to be a constant function. This restriction is already contained in the forbidden special form for the choice $\alpha = 0$.

We defined an artificial correspondence between b_1 and ζ_1. It is no matter to which of the ζ_i the variable b_1 is associated, the goal is just to have as many variables ζ_i as variables x_i.

It is possible to choose some of the parameters a_i equal to zero (this happens when the ridge is orientated parallel to an axis), but the choice $a = 0$ is not allowed.

4.2.2 Modified Ridge Constructions

According to the standard Assumptions 4.1 the parameters a and b must be restricted to compact sets. Within the greedy algorithm only elements $g \in G_B$ may be chosen, therefore also the parameter c has to be bounded. We now present a method for implementing these bounds in the greedy algorithm if an optimization method such as Landweber's or Newton's method is used to determine the parameters of the individual nodes.

In the case of ridge constructions the parameter b describes the distance between the ridge and the origin. If during the iteration b becomes very large then the ridge will be situated outside the domain Ω and the function will be (almost) constant along the domain Ω. Its derivative will therefore be almost zero and the updates for b computed by Landweber's or Newton's method vanish. So the parameter b stays within a bounded interval, even if we do not implement any bound for it.

For the vector a we propose to use a decomposition $a = sa_0$ into a unit vector a_0 and a scaling factor s, and to represent a_0 by use of angles in polar coordinates. The angles need not be restricted to a compact set and so the iteration can again be implemented without any additional cost. For the scaling factor s we can use the same procedure as described below for the weight c.

Above we argued that even if b does not stay inside the interval P at least it lies in a sufficiently large compact set $\tilde{P} \supset P$. For the weight c we may not use such arguments, because the corresponding bound B may not be enlarged to a different bound $\tilde{B}$, otherwise we lose the convergence statement of Theorem 2.3. Nevertheless there is a simple possibility to bound the functions in the set G_B, namely to modify the network operator and define it as

$$F(c,t) = \kappa(c)\Phi(x,t), \tag{4.5}$$

where $\kappa(\cdot)$ is an invertible, differentiable mapping of $\mathbb{R}$ to the interval $[-B,B]$ with the properties

$$\lim_{c\to\infty} \kappa(c) = B, \quad \lim_{c\to-\infty} \kappa(c) = -B, \quad \text{and} \quad \kappa(0) = 0\,.$$

Using this setting the value of c does not have to stay within a compact set, but the norm of the elements of the set G_B remains bounded by B.

4.2.3 The Greedy Algorithm for Ridge Constructions

Algorithm 4.3. Greedy approximation using ridge constructions.

```
Initialization:
    - For the case of noise-free data choose a constant $M$, such
      that $M > \gamma$ (as defined in (4.2)), e. g., choose $M$ equal to
      $M = \frac{3}{2}(B^2 - \|f\|^2)$ and set $M_{\text{opt}} = M$, $k_{\text{opt}} = \infty$.
    - For the case of noisy data determine the constants $M_{\text{opt}}$
      and $k_{\text{opt}}$ according to Method 3.4.
    - Set $f_0^\delta := 0$.
Iteration:
    for $k := 1$ to Min($k_{\text{opt}}$, maxit) do
        if $\left\|f^\delta - f_{k-1}^\delta\right\|^2 \le \frac{M}{k}$ then set $f_k^\delta := f_{k-1}^\delta$ and jump
        to the next step of the iteration.
        Find parameters $a$, $b$, and $c$ such that
            $\left\|f^\delta - \frac{k-1}{k} f_{k-1}^\delta - \frac{1}{k} c\sigma(a^T x + b)\right\|^2 \le \frac{M_{\text{opt}}}{k}$    (4.6)
        is fulfilled and define $f_k^\delta$ as
            $f_k^\delta := \frac{k-1}{k} f_{k-1}^\delta + \frac{1}{k} c\sigma(a^T x + b)\,.$
    end for
```

Remark 4.4 (Feasibility). This algorithm looks slightly different than the one presented in Section 2.2, the sequence ε_k and the search for infima are gone. In practice we are usually not able to calculate the infimum formulated in the original algorithm and therefore we are also not able to check if the difference to this infimum is less than ε_k. Nevertheless Algorithm 2.2 in combination with Theorem 2.3 ensures that there *exists* an element, which can give us the rate $\frac{M}{k}$. This means that in each step of the iteration we are able to find an approximation and the algorithm above is feasible. Furthermore, with this modified algorithm we are able to check if the approximation we found is sufficiently good, we only have to look if the estimate (4.6) is fulfilled.

In the language of neural networks each step of Algorithm 4.3 can be interpreted as an approximation of a function (e. g. $f^\delta - \frac{k-1}{k} f_{k-1}^\delta$) with a neural network consisting of only a single node.

Note that we do not have to change f_k in each step. If f_k is already a good approximation to f we increase the index k as much as possible. This can be done without affecting the convergence rate, since for the induction argument in the proof of Theorem 2.3 we only needed that the kth approximation fulfills the rate $\|f - f_k\|^2 \leq M/k$, but not that it consists of k nodes.

Remark 4.5. Altogether we find the following advantages of Algorithm 4.3 compared to Algorithm 2.2:

- In each step there exists an element g_k that fulfills (4.6), namely at least one of those that fulfill the infimum-condition in Algorithm 2.2.
- We are able to check if the approximation we found is sufficiently good, since everything in relation (4.6) can be computed, as opposed to the infimum-condition which can not be checked in practice.
- If the stopping index k is chosen according to Method 3.4 then the algorithm is stable under data-perturbations and according to Theorem 3.5 it yields the convergence rate $\left\| f - f_{k(\delta, f^\delta)}^\delta \right\| = \mathcal{O}(\delta^{1/3})$.

4.3 Radial Basis Functions

4.3.1 Assumptions

Standard Assumptions 4.6. For radial basis functions the activation function Φ fulfills the assumptions:

- $\Phi(x, t)$ is a function of form
$$\begin{aligned} \Phi(x,t) &= \Xi\left((a_1 x_1 + b_1)^2 + \cdots + (a_n x_n + b_n)^2\right) \\ &= \Xi\left(\zeta_1^2 + \cdots + \zeta_n^2\right), \end{aligned} \tag{4.7}$$
with the new variables $\zeta_i = a_i x_i + b_i$
- the values of the parameters a_i are chosen from a compact set that does not contain 0.
- the function Ξ does not have the form
$$\Xi(\xi) = C\xi^\alpha$$
for any combination of real numbers α and C
- the function Ξ is twice continuously differentiable.

Remark 4.7. The activation function Ξ is not allowed to be a constant function. This restriction is already contained in the forbidden special form for the choice $\alpha = 0$.

Our definition of radial basis functions is more general than the usual one, e. g. in [4] the "radius" of the function is fixed, in [9] the radius is a parameter but the shape of the function is always radially symmetric. In this setting also ellipsoidal shapes are allowed. If all the a_i are equal (or especially fixed and equal to one) then the original case is attained.

For radial basis functions it is not allowed to choose any parameter $a_i = 0$.

4.3.2 Modified Radial Basis Function

As before in Section 4.2.2 we will now discuss the implementation of the bounds needed in the standard Assumptions 4.6 and within the greedy algorithm.

In the case of radial basis functions the parameter t describes the center of the network function. The parameter set P is therefore chosen approximately as the domain Ω. If t is chosen far outside from P then, for instance in the case of Gaussian functions, the norm of $\Phi(x,t)$ will be very small, since only the part inside Ω contributes to the norm. The corresponding functions Φ and its derivative $\nabla_t\Phi$ are very close to the zero function and if t is sufficiently far outside the domain they are numerically equal to the zero function. Therefore also the operator F'^* will be zero and Landweber iteration as well as Newton's method will stop. Of course the corresponding approximation will be a bad one, but the parameters t stay inside some compact set, even if we do not implement any bound for them.

The bound for the parameter c can be implemented as before in Section 4.2.2

4.3.3 The Greedy Algorithm for Radial Basis Functions

Algorithm 4.8. Greedy approximation using radial basis functions.

```
Initialization:
    - For the case of noise-free data choose a constant $M$, such
      that $M > \gamma$ (as defined in (4.2)), for example choose $M$
      equal to $M = \frac{3}{2}(B^2 - \|f\|^2)$ and set $M_{\text{opt}} = M$, $k_{\text{opt}} = \infty$.
    - For the case of noisy data determine the constants $M_{\text{opt}}$
      and $k_{\text{opt}}$ according to Method 3.4.
    - Set $f_0^\delta := 0$.
Iteration:
  for $k := 1$ to Min($k_{\text{opt}}$, maxit) do
      if $\left\|f^\delta - f_{k-1}^\delta\right\|^2 \le \frac{M}{k}$ then set $f_k^\delta := f_{k-1}^\delta$ and jump
      to the next step of the iteration.
      Find parameters $c$ and $t$ such that
          $\left\|f^\delta - \frac{k-1}{k} f_{k-1}^\delta - \frac{1}{k} c\Xi(\|x-t\|)\right\|^2 \le \frac{M_{\text{opt}}}{k}$   (4.8)
      is fulfilled and define $f_k^\delta$ as
          $f_k^\delta := \frac{k-1}{k} f_{k-1}^\delta + \frac{1}{k} c\Xi(\|x-t\|)\,.$
  end for
```

The remarks which are given in Section 4.2.3 below Algorithm 4.3 are also valid for this Algorithm.

4.4 Convergence in Stronger Norms

An interesting property of the greedy algorithm is that it leads to convergence in stronger Sobolev norms. This seems surprising at a first glance, since during the algorithm only the L^2-norm of the elements is observed.

However, the set G_B is *compact* in stronger topologies, which allows us to show convergence and even convergence rates in stronger norms. In the following we assume that

$$\Phi(\cdot,t) \in H^s(\Omega) \quad \forall t \in P$$

(for the definition of the Sobolev-space $H^s(\Omega)$ we refer to [1, 14]). Since the set G_B is compact we can find an upper bound for the H^s-norm of the elements of G_B, which is given via

$$B_s = \sup_{|c|\leq B,\, t\in P} \|c\Phi(\cdot,t)\|_{H^s(\Omega)} < \infty\,. \tag{4.9}$$

Hence, G_B is *bounded* in the H^s-topology.

In the first theorem we show that convergence in the L^2-norm implies weak convergence in the H^s-norm if the activation function $\Phi(\cdot,t)$ belongs to $H^s(\Omega)$. From this we can easily deduce strong convergence in the H^r-norm for $r < s$. Finally we give rates for the convergence in spaces $H^r(\Omega)$.

Theorem 4.9. *Let $\Phi(\cdot,t) \in H^s(\Omega)$ for all values $t \in P$, and let (f_k) be the sequence generated according to Algorithm 4.3. Then the sequence is bounded in $H^s(\Omega)$ and converges weakly in $H^s(\Omega)$ with limit f, i. e., $f_k \rightharpoonup f$.*

Proof. Using equation (4.9) we can conclude that the H^s-norm of any f_k is bounded by

$$\begin{aligned}\|f_k\|_{H^s(\Omega)} &= \left\|\frac{1}{k}g_1 + \cdots + \frac{1}{k}g_k\right\|_{H^s(\Omega)}\\ &\leq \frac{1}{k}\left(\|g_1\|_{H^s(\Omega)} + \cdots + \|g_k\|_{H^s(\Omega)}\right)\\ &\leq \frac{1}{k}k \sup_{1\leq i\leq k}\|g_i\|_{H^s(\Omega)} \leq B_s\,. \end{aligned}\tag{4.10}$$

Hence, the H^s-norm of the residual is also bounded and can be estimated via

$$\|f - f_k\|_{H^s(\Omega)} \leq \|f\|_{H^s(\Omega)} + \|f_k\|_{H^s(\Omega)} \leq \|f\|_{H^s(\Omega)} + B_s\,.$$

Since the Sobolev space $H^s(\Omega)$ is reflexive and the sequence f_k is bounded we can find a subsequence f_{k_l} which converges weakly to some function f^*. Since there exists a compact embedding operator E from $H^s(\Omega)$ to $L^2(\Omega)$ and the sequence f_k converges in $L^2(\Omega)$ the relation $Ef^* = f$ must be fulfilled. Furthermore $f \in H^s(\Omega)$, because $\overline{\text{co}}(G) \subset H^s(\Omega)$, and thus, $f^* = f$.

Hence, we found that the sequence (f_k) has a weakly converging subsequence f_{k_l} and that the limit of f_{k_l} is equal to f. Since analogous reasoning applies if we start with a subsequence of (f_k), we obtain that every subsequence of (f_k) contains a weakly converging subsequence whose limit is f. Consequently we can conclude that the original sequence (f_k) itself converges weakly to f and the proof is completed. □

We can use this result to prove strong convergence in spaces $H^r(\Omega)$ with $r < s$.

Corollary 4.10. *Let $\Phi(\cdot, t) \in H^s(\Omega)$ for all values $t \in P$, and let (f_k) be the sequence generated according to Algorithm 4.3. Then for $r < s$ the sequence also converges in $H^r(\Omega)$, i. e.,*

$$\|f - f_k\|^2_{H^r(\Omega)} \to 0 \quad \textit{for} \quad k \to \infty$$

holds.

Proof. From Theorem 4.9 we know that (f_k) converges weakly to f in $H^s(\Omega)$. For $r < s$ there exists a compact embedding K from the Sobolev spaces $H^s(\Omega)$ to $H^r(\Omega)$. Compact operators transfer weakly converging sequences to norm converging sequences, and therefore f_k converges to f in $H^r(\Omega)$. □

Not only the functions f_k remain bounded in stronger Sobolev-norms, but also the approximations f_k^δ which are obtained when a noisy version of f is approximated. Since $f_k^\delta \to f$ in $L^2(\Omega)$ for $\delta \to 0$ we can use similar arguments as above to show convergence of the regularized solutions f_k^δ in stronger norms.

Corollary 4.11. *Let $\Phi(\cdot, t) \in H^s(\Omega)$ for all values $t \in P$, and let (f_k^δ) be the sequence generated according to Algorithm 4.3. Moreover, let for $\delta > 0$ the stopping index $k_* = k_*(\delta, f^\delta)$ and the constant M be chosen according to Method 3.4. Then for $r < s$ the sequence converges to f also in $H^r(\Omega)$, i. e.,*

$$\left\|f - f_{k_*}^\delta\right\|^2_{H^r(\Omega)} \to 0 \quad \textit{for} \quad \delta \to 0$$

holds.

Using the interpolation inequality we can even show convergence rates in spaces $H^r(\Omega)$ with $r < s$.

Corollary 4.12. *Let $\Phi \in H^s(\Omega)$ and $f \in \overline{\mathrm{co}}(G)$. Then for $r < s$ the convergence rate*

$$\|f - f_k\|^2_{H^r(\Omega)} = \mathcal{O}\left(k^{\frac{r-s}{s}}\right)$$

holds.

Proof. In equation (4.10) in the proof of Theorem 4.9 we have seen that the H^s-norm of the approximating functions is bounded. Further we know that the convergence rate in the L^2-norm is given as $\mathcal{O}(k^{-1})$. If we combine these two results the proof follows immediately using the interpolation inequality (see [8, (2.49)] or [14, (2.43)]). □

Using the convergence result $\left\|f - f_{k_*-1}^\delta\right\|_{L^2} = \mathcal{O}\left(\delta^{1/3}\right)$ (see Remark 3.5) and the interpolation inequality we may also conclude a result on the convergence of $f_{k_*-1}^\delta$ in stronger Sobolev-norms if the parameters M and k are chosen in an optimal way:

Corollary 4.13. *Let $\Phi \in H^s(\Omega)$ and $f \in \overline{\mathrm{co}}(G)$. Moreover, let for $\delta > 0$ the stopping index $k_* = k_*(\delta, f^\delta)$ and the constant M be chosen according to Method 3.4. Then for $r < s$ the convergence rate*

$$\left\|f - f_{k_*-1}^\delta\right\|_{H^r(\Omega)} = \mathcal{O}\left(\delta^{\frac{s-r}{3s}}\right)$$

holds.

A special situation arises if the activation function is of class C^∞. In this case we find the same rate of convergence in the H^r-norm as in the L^2-norm, namely $\mathcal{O}(k^{-1})$. Nevertheless, in practice this behaviour will not be visible, since the constants in the convergence rates can be large, and therefore all these results only hold for k sufficiently large. If we interpret Φ as an element of $H^2(\Omega)$ we obtain the weaker convergence rate $\mathcal{O}(k^{-\frac{1}{2}})$, but the constants will be less. So if we observe the H^1-norm of the residual we will find that it decreases, but not from the start with a high convergence rate, but remaining almost constant at the beginning and then converging with gradually increasing speed (see also Figure 6). A similar behaviour was observed also with Tikhonov regularization in [6].

5 Numerical Examples

In this section we verify the theoretical results from Chapter 2 by numerical examples. First of all, we investigate ridge constructions where we inspect the qualitative and quantitative behaviour of the approximations in the L_2- and the H^1-norm during the iteration. Next we examine the influence of noise in the data and compare the numerical results with the prediction, provided by the stopping rule in Section 3.3. Finally we investigate the qualitative behavior of the algorithm when applied to radial basis function networks with Gaussian activation functions.

All examples were computed using the software system *Mathematica* on an SGI Origin 3800.

5.1 Ridge Constructions

In the first example we consider a neural network based on a ridge construction for an approximation problem on the 2-dimensional domain $\Omega = [-1,1]\times[-2,2]$. As activation function we choose

$$\Phi(x,t) = \Phi(x_1,x_2,t_1,t_2) = \sigma\big((\sin(t_1),\cos(t_1))\,(x_1,x_2)^{\mathrm{T}} + t_2\big)\,,$$

where σ is given as

$$\sigma(\xi) = \frac{1}{1+\exp(-50\xi)}\,.$$

As proposed in Section 4.2.2 we implement the bound for the parameter c using an auxiliary function κ. The neural network-operator is therefore defined via

$$F(c,t_1,t_2) = \kappa(c)\Phi(\cdot,t_1,t_2)\,,$$

where the function κ is given as

$$\kappa(c) = \frac{2}{1+e^{-c}} - 1\,.$$

To ensure that the function to be approximated is an element of $\overline{\mathrm{co}}(G_B)$ we define it explicitly as a convex combination of three elements of G_B namely as

$$f = \frac{F(5,1,0.6)+F(-2,3,0.3)+F(5,5,0.4)}{3}\,. \tag{5.1}$$

A plot of this function can be seen in the upper right corner of Figure 2. Using this choice, the function f is an element of $\overline{\text{co}}(G_B)$ for $B = 2.45$. Since $\|f\| = 1.24$ it suffices to choose $M = 5$ according to Algorithm 4.3.

To find parameters satisfying equation (4.6) we set $c^0 = 0$ and take (t_1^0, t_2^0) randomly. Then we perform several iterations of Landweber's and Newton's method until a convergence criterion is fulfilled or the maximal number of iterations is exceeded. To ensure that the approximation we compute is an element of $\overline{\text{co}}(G_B)$ we implement the algorithm such that, as soon as the norm of $\Phi(\cdot, t_1^j, t_2^j)$ is greater than[1] the value $B = 2.45$, the search is terminated and restarted with a different initial value (t_1^0, t_2^0).

5.2 Behaviour during the Algorithm

Figure 2 illustrates the qualitative behaviour of Algorithm 4.3 in dependence of the iteration index k. Since the number of nodes need not be increased in order to satisfy the estimate (4.6) in each step, the network size (denoted as k_{eff}) has to be less than or equal to k. In our example k is always much larger than k_{eff}. For instance in the third column the network consists of 17 nodes but the error estimate (4.6) is fulfilled for $k = 40$. This means that the *if*-clause in Algorithm 4.3 was true 23 times. This behaviour can be seen in more detail in Figure 3. The ratio $\frac{k_{\text{eff}}}{k}$ remains approximately constant during the iteration.

In Figure 4 the evolution of the residual, i. e., $\|f - f_k\|_{L_2(\Omega)}$, is shown. The green (smooth) line represents the error estimate

$$\|f - f_k\| \leq \sqrt{\frac{M}{k}},$$

the red one represents the residual. Observe that every time the approximation is improved and the red line moves downwards, the iteration index k is increased (the red line is horizontal) as much as possible, such that the green line is not hit.

Theorem 4.10 ensures convergence in stronger Sobolev-norms, if the activation function Φ is smooth. In our case the activation function is of class C^∞, hence we might expect the same convergence rate in the H^1-norm as in the L_2-norm. Nevertheless, the norm of the derivatives of Φ grows fast, and so the observed rate (i. e., the slope of the curve) for finite k will be less. Figure 5 shows the behaviour of the norm of the derivatives of $f - f_k$. Both derivatives are decreasing almost monotonically.

In Figure 6 the behaviour of the full H^1-norm (blue line) and the L_2-norm (red line) of the error is plotted in a logarithmic scale. As we expected the speed of convergence in the H_1-norm is gradually increasing, and for $k \to \infty$ the slope of the blue line approaches $-\frac{1}{2}$. For instance if only the values $k \geq 50$ are taken into account, then the slope of the blue line is approximately -0.25, i. e., we find numerically the rate $\|f - f_k\|_{H^1(\Omega)} = \mathcal{O}(k^{-1/4})$. According to Theorem 4.12 such a rate can be gained if $\Phi(\cdot, t) \in H^2(\Omega)$ for all values $t \in P$.

[1] This is done only because we want to verify our theoretical results, otherwise it does not matter if the iterates are elements of the set G_B, or of a set $G_{\tilde{B}} \supset G_B$.

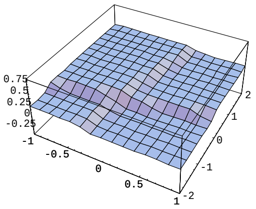
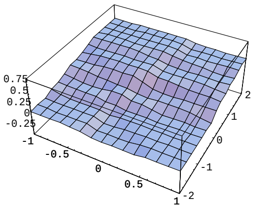
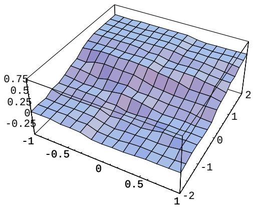
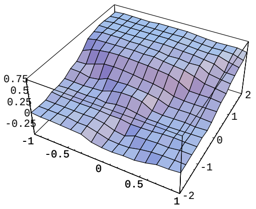
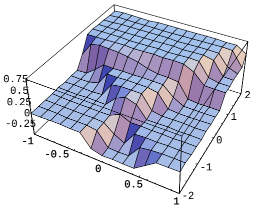
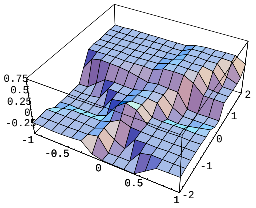
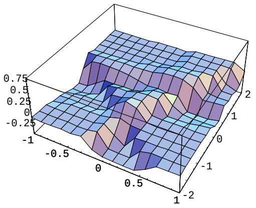
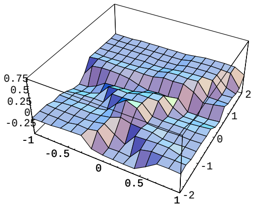
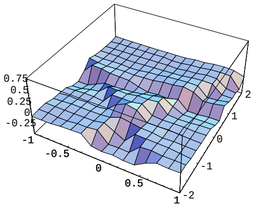
k	10	20	40	100	original
k_{eff}	5	10	17	55	
f_k					
$f - f_k$					

Figure 2: Evolution of the approximation f_k and the error $f_k - f$ during the greedy algorithm.

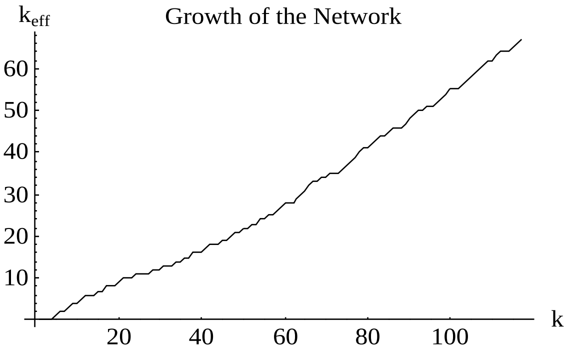


Figure 3: Evolution of the network size during the greedy algorithm.

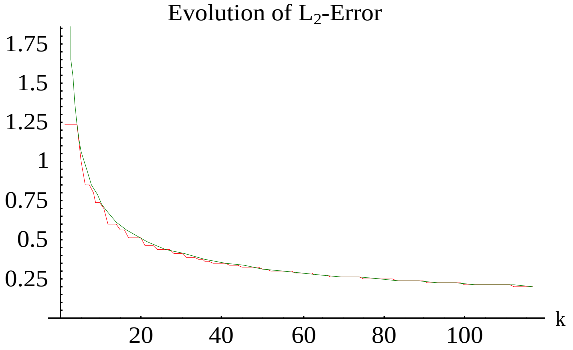


Figure 4: Evolution of the L_2-norm of the error $f - f_k$.

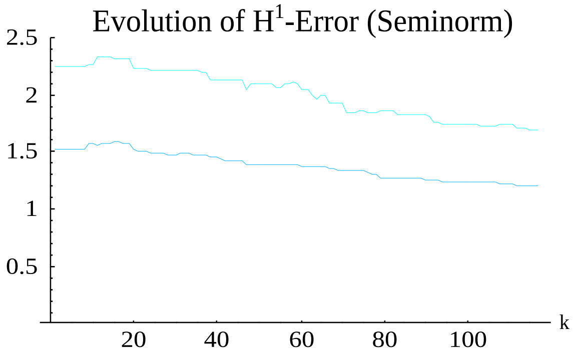


Figure 5: Evolution of the H^1-seminorm of the difference $f - f_k$. The upper line corresponds to $\|\partial_{x_1}(f - f_k)\|_{L_2(\Omega)}$, the lower one to $\|\partial_{x_2}(f - f_k)\|_{L_2(\Omega)}$.

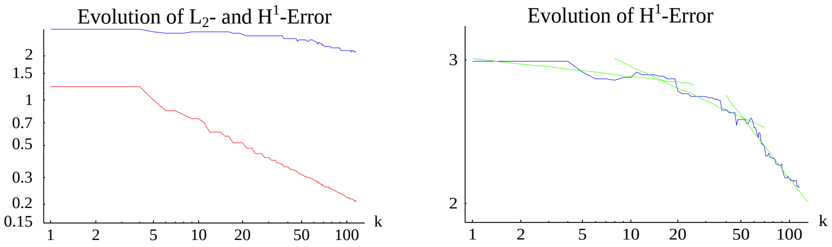


Figure 6: Logarithmic plot of the evolution of the L_2-norm (red) and the H^1-norm (blue) of the difference $f - f_k$. The right graph is a magnification of the left one and shows interpolations of the H^1-error (green). The slope is gradually increasing.

5.3 Influence of Noise

Now we investigate the influence of noise on the algorithm. Therefore we add high frequency deterministic noise with variable amplitude to the data. From Section 3.2 we know that the algorithm will fail to find new updates g_k if noise is present and k is too large.

We implement the algorithm as above, which means that in the kth step of the iteration we choose a random value for the parameters (t_1, t_2), set $c = 0$ and perform several Landweber and Newton steps. If the computed approximation is not sufficiently good (i. e., equation (4.6) is not fulfilled), we repeat the same procedure for a different starting value for (t_1, t_2). In the noise-free case this procedure works well and we find a new update after around 2–4 steps. If noise is present the algorithm encounters problems and fails to find a new update if k is too large. For this reason, we terminate the iteration if no valid update is found for 20 different initial values. So we did not implement the stopping rule from Section 3.3, but looked for the point where the algorithm naturally terminates.

Figure 7 illustrates the qualitative behaviour of this procedure. For instance in column 3 the amplitude of the perturbation was set to 0.57 which results in a noise level of 46%. The algorithm stopped after 12 iterations, at this time the network consisted of 8 nodes. The first line shows the noisy function f^δ, the second one the approximation f_k^δ found by the algorithm, line three shows the difference $f^\delta - f_k^\delta$ and the last one the difference between the approximation and the original, undisturbed function f. One observes that the iterates f_k^δ are not sensitive to the noise, they are always smooth functions and better approximations to f than to f^δ. The reason for this is that the search for elements f_k^δ is restricted to the set $\overline{\text{co}}(G)$, which is a set of smooth functions (see also Section 4.4).

In Figure 8 this behaviour is analyzed quantitatively. The blue line indicates the norm of $f^\delta - f_k^\delta$, the red line corresponds to $\left\|f - f_k^\delta\right\|$. Although the blue line is slightly steeper than the red one, the values for the blue line are always above the red ones. The green line corresponds to the theoretical prediction from the stopping rule from Section 3.3 for the error in dependence of δ. The predicted values are always far above the measured ones, but the slope and therefore the rate of convergence is approximately equal to the experimental one. This indicates that the rate expected according to the stopping rule above is obtained also numerically, but that the constants are possibly too large and might be improved, in particular by using sharper estimates instead of (3.4).

Figure 9 shows the behaviour of the H^1-norm of the error. Clearly, if the noise level is high, it tends to zero much faster for $f^\delta - f_k^\delta$ than for $f - f_k^\delta$. This is due to the fact that the H^1-norm of f^δ is much larger than the H^1-norm of f. Since f_k^δ is a smooth function, also the corresponding difference $f^\delta - f_k^\delta$ is larger than $f - f_k^\delta$ in the H^1-norm. As the noise level decreases also this effect vanishes and the slope of the blue line decreases. Note that the blue line always lies above the red one, i. e., f_k^δ always fits better to f than to f^δ.

Figure 10 illustrates the behaviour of the three different solutions given in Section 3.3. For k_1 the ratio $\frac{M(k_1)}{k_1}$ becomes constant for δ tending to zero. For k_3 the ratio increases, as mentioned before this solution seems to correspond to a local maximum. Only for k_2 the ratio $\frac{M(k_3)}{k_3}$ tends to zero, the slope of the corresponding line is $\frac{2}{3}$, this means the convergence rate for noise level tending

Noise	0.8 (65%)	0.57 (46%)	0.4 (32%)	0.2 (16%)	0.1 (8%)
k	24(7)	12(8)	18(5)	48(23)	72(41)

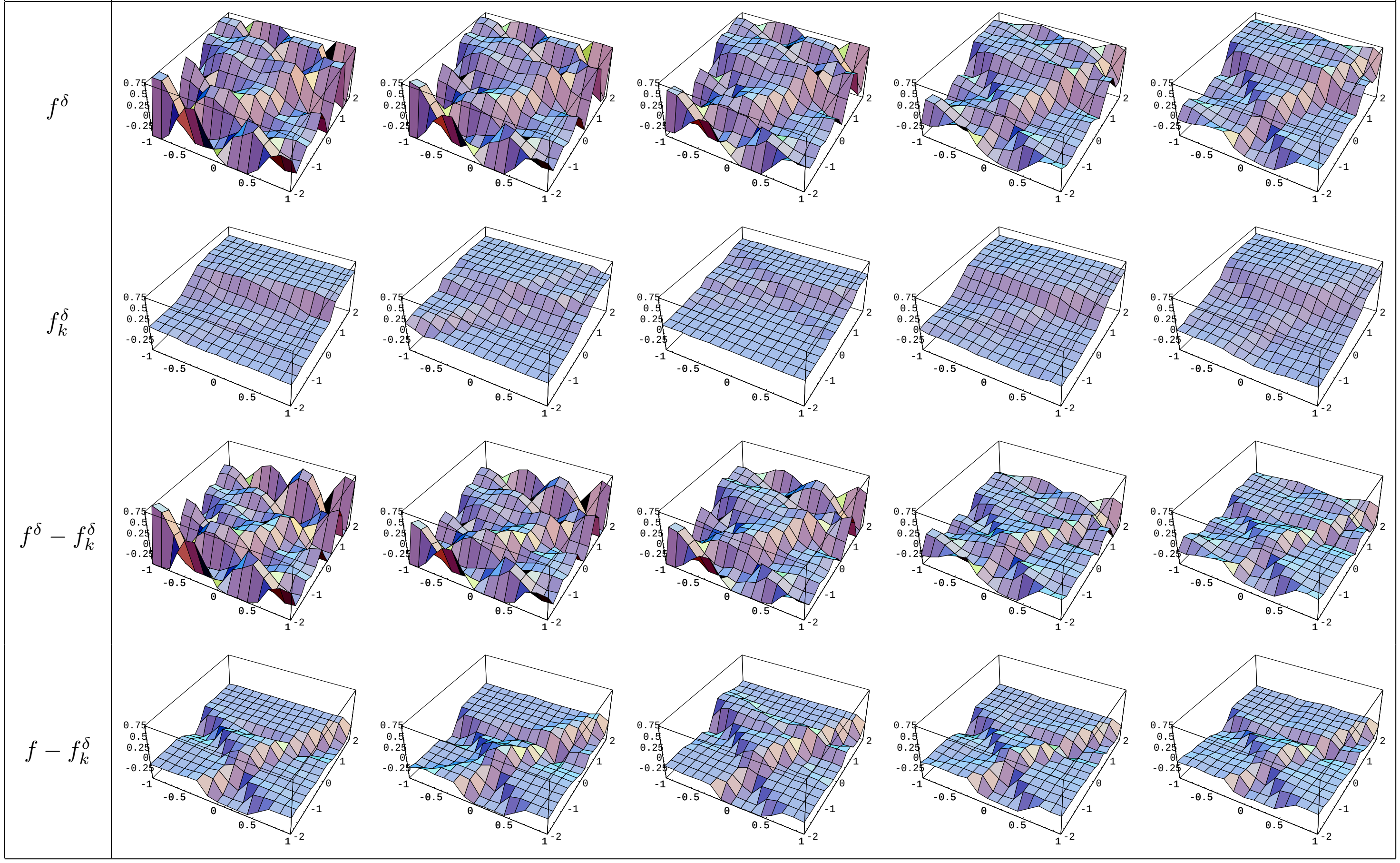


Figure 7: Qualitative behaviour of the algorithm for noise level tending to zero.

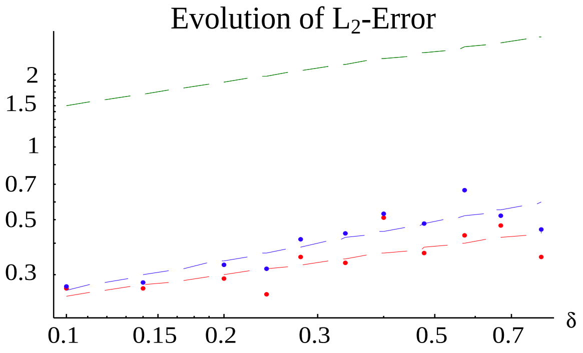


Figure 8: Evolution of the L_2-norm of the difference $f^\delta - f_k^\delta$ (blue) and $f - f_k^\delta$ (red) for noise level tending to zero. The green line indicates the rate which is obtained for the stopping rule.

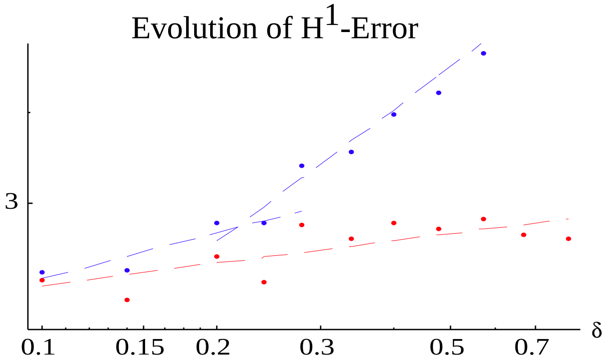


Figure 9: Evolution of the H^1-norm of the difference $f^\delta - f_k^\delta$ (blue) and $f - f_k^\delta$ (red) for noise level tending to zero.

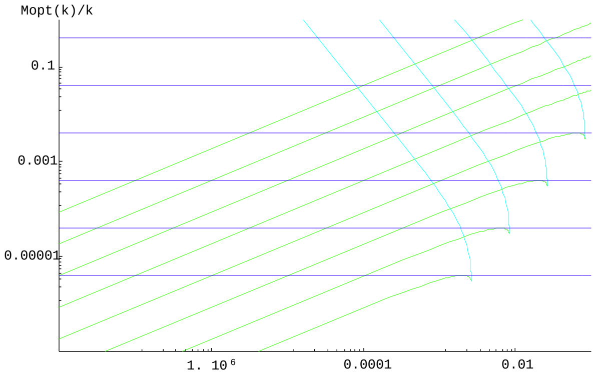


Figure 10: Behaviour of $\frac{M(k_i)}{k_i}$ corresponding to the three different choices for k in Section 3.3. The lower curves correspond to the choice $\gamma = 10^{-6}$, the top ones to $\gamma = 10^{-1}$.

to zero is $\delta^{2/3}$. Since $\frac{M}{k}$ measures the squared norm we obtain

$$\left\| f^\delta - f_k^\delta \right\| = \mathcal{O}(\delta^{\frac{1}{3}})\,.$$

As we have seen above, we find approximately the same rate in Figure 8.

5.4 Radial Basis Functions

In the second example we investigate the qualitative behaviour of the algorithm when applied to *radial basis function* networks. Again we choose the 2-dimensional domain $\Omega = [-1, 1] \times [-2, 2]$. As activation function we use

$$\Phi(x, t) = \Phi(x_1, x_2, t_1, t_2) = \Xi\left(\sqrt{(x_1 - t_1)^2 + (x_2 - t_2)^2}\right),$$

where Ξ is given as a Gaussian function, namely

$$\Xi(\xi) = 5\exp(-10\xi^2).$$

Using this choice the norm of Φ is bounded by the value 2 uniformly in t. The bound for c is implemented as above via the function $\kappa(c)$.

To ensure that the function to be approximated is an element of $\overline{\text{co}}(G_B)$ we again define it explicitly as a convex combination of elements of G_B, but now via an integral, namely

$$f = \int_P \mathbf{1}_{[-\frac{1}{2},\frac{1}{2}]\times[-1,1]}(t_1, t_2)\Phi(\cdot, t_1, t_2)\, dt_1\, dt_2\,, \tag{5.2}$$

where $\mathbf{1}_{[\cdot]\times[\cdot]}$ is the characteristic function. A plot of this function can be seen in the right upper corner of Figure 11. Using this choice, the function f is an element of $\overline{\text{co}}(G_B)$ for $B = 2$. We choose $M = 3.83$ which is equal to $1.2(B^2 - \|f\|^2)$.

In Figure 11 the qualitative behaviour of Algorithm 4.3 is shown in dependence of the iteration index k. Again the network size k_{eff} is far below the iteration index k. Note that the ratio $\frac{k_{\text{eff}}}{k}$ is even smaller than in the example of Section 5.1. The network we computed for $k = 150$ effectively uses only $k_{\text{eff}} = 55$ nodes, but yields a very good approximation. A possible reason for the (optically) better performance in the second example is, that the function being approximated fulfills an integral representation of the form

$$f = \int_P \Phi(\cdot, t)h(t)\, dt$$

(see also equation (4.3)) for a much smoother function h than in the first example. For the function defined via equation (5.1) h is a distribution, whereas for the one defined via equation (5.2) $h \in L^\infty(P)$.

Figure 12 shows the influence of noise on the algorithm for two different noise-levels. Again the algorithm was terminated if no valid approximation was found for 20 different initial values for (t_1, t_2). As in the case of ridge-construction the approximations are smooth functions. They are better approximations to f than to f^δ.

Acknowledgements

The authors thank Prof. Heinz W. Engl (University Linz) for useful and stimulating discussions. Financial support is acknowledged to the Austrian Science Foundation FWF through project SFB F 013 / 08.

References

[1] Robert A. Adams. *Sobolev Spaces.* Academic Press, New-York San Francisco London, 1975.

[2] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[3] U. Bodenhofer, M. Burger, H. W. Engl, and J. Haslinger. Regularized data-driven construction of fuzzy controller. *Journal of Inverse and Ill-posed Problems*, 10:319–344, 2002.

[4] M. Burger and H. W. Engl. Training neural networks with noisy data as an ill-posed problem. *Advances in Computational Mathematics*, 13:335–354, 2000.

[5] M. Burger and A. Hofinger. ?? *A note on iterative methods for neural network training*, in preparation.

[6] M. Burger and A. Neubauer. Analysis of Tikhonov regularization for function approximation by neural networks. *Neural Networks*, 16:79–90(2003).

[7] A. T. Dingankar and I. W. Sandberg. A note on error bounds for approximation in inner product spaces. *Circuits, Systems and Signal Processing*, 15(4):519–522, 1996.

k	10	20	40	150	original
k_{eff}	3	6	15	55	

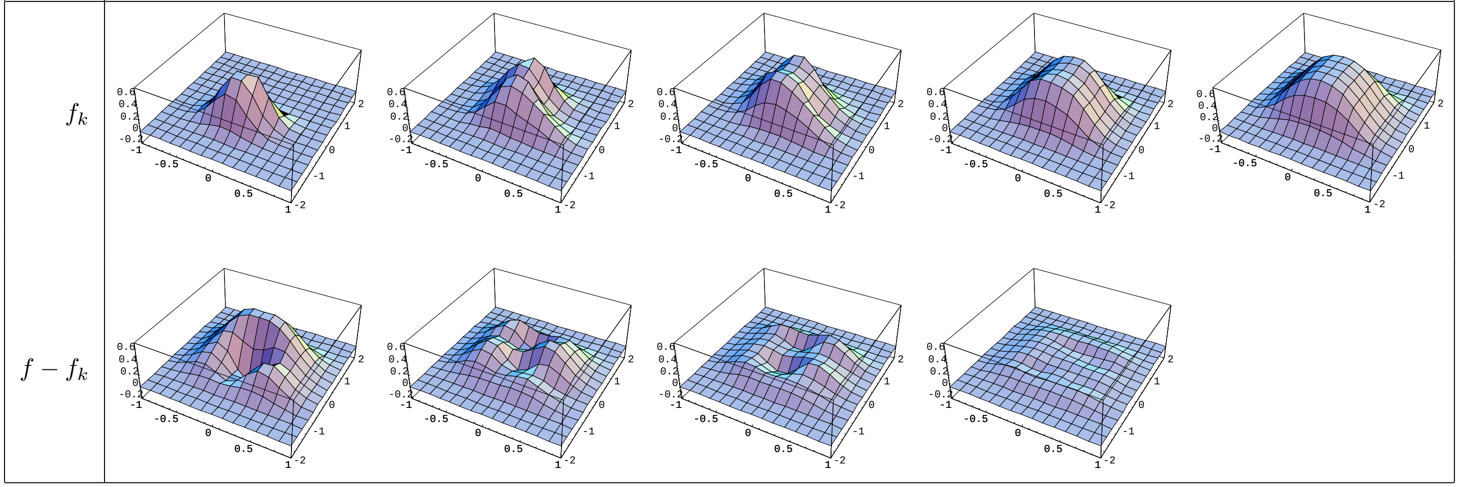


Figure 11: Evolution of the approximation f_k and the difference $f_k - f$ during the greedy algorithm.

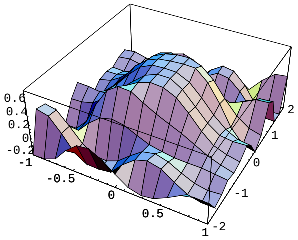
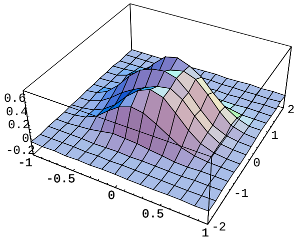
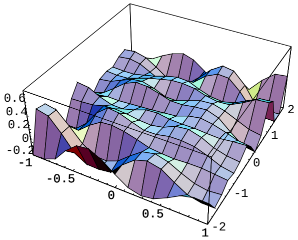
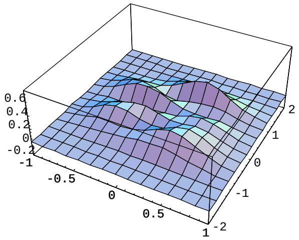
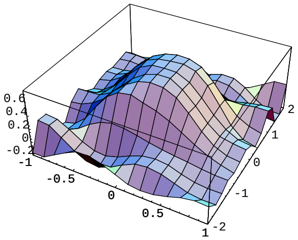
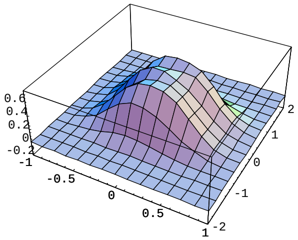
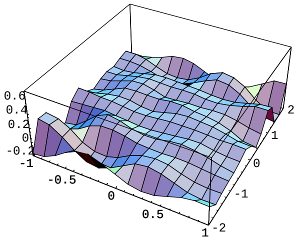
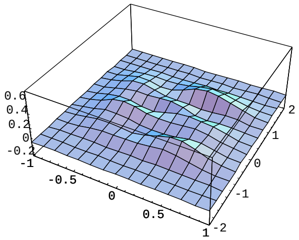
δ	$k(k_{\text{eff}})$	f^δ	f_k^δ	$f^\delta - f_k^\delta$	$f - f_k^\delta$
38%	26(12)				
22%	72(35)				

Figure 12: Behaviour of the algorithm for noisy data and two different noise levels.

[8] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems.* Kluwer, Dordrecht, 1996.

[9] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[10] A. Hofinger. Iterative regularization and training of neural networks. Master's thesis, University of Linz, 2003.

[11] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[12] L. K. Jones. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Stat.*, 20:608–613, 1992.

[13] W. Linde. *Infinitely Divisible and Stable Measures on Banach Spaces.* Teubner, Leipzig, 1983.

[14] J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications*, volume 1. Springer, Berlin, Heidelberg, 1972.

[15] D. Werner. *Funktionalanalysis.* Springer, Berlin, Heidelberg, 1995.