ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 245-250

# A Kernel-Based Adaline

Thilo-Thomas Frieß[†] and Robert F Harrison[†‡]

[†]Dept. AC&SE, University of Sheffield, Sheffield, S1 3JD, UK.

[‡]Dept. MAE, Princeton University, Princeton, NJ 08544, USA

This new algorithm combines the conceptual simplicity of a least-mean-square algorithm for linear regression, but exhibits the power of a universal non-linear function approximator. The method is based on a generalisation of the Widrow-Hoff LMS rule using Mercer kernels. Simple examples in curve fitting and non-linear systems identification are solved by the method.

## 1. Introduction

By expanding a function in series form it can be represented to an arbitrary degree of accuracy by taking enough terms. It is therefore possible, in principle, to conduct a linear regression on a new set of variables, transformed by a fixed mapping. This leads to a large computational burden and to the need for an infeasible amount of data from which the coefficients must be estimated and is not generally practical for function approximation. The algorithm studied in [1] is a linear Perceptron, which is computed *implicitly* in a space of infinite-dimension (the linearisation space) using potential (kernel) functions. These have been further exploited in the Support Vector Machine (SVM) [2], principal component analysis [3], linear programming machines [4] and clustering [5]. The kernel-Adatron [4,6,7], provides a fast, simple, and robust alternative to SVM classifiers providing arbitrary, large-margin discriminant functions *iteratively*, so avoiding the intensive QP computations of the SVM.
We now use kernels to develop a non-linear version of the Adaline [8], yielding a general, non-linear adaptive mapping device via an algorithm with well-documented properties. Selecting an appropriate kernel and its parameter specifies the mapping to the *linearisation space*, which can be done empirically, via cross validation.

## 2. The Adaline

The Adaline [8] has the form:

$$f = \langle w, x \rangle + b \tag{1}$$

where $x$ is an $n$-vector of input data, $w$ a set of $n$ weights, $b$ a bias and $\langle w, x \rangle = \sum_i w_i x_i$ denotes the scalar-product. The Adaline is a member of the general class known as Perceptrons. Its objective is: given some measured data, $y_i \quad i = 1 \ldots L$ ,

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 245-250

adjust the weights, *w*, so that the mean-square error (MSE) between the output, *f*, and the measured data, *y*, is minimised. Replacing the true gradient by its *instantaneous* estimate in the direct gradient descent solution leads to the LMS adaptation rule [8]:

$$w \leftarrow w + \eta(y - f)x\ ,\ b \leftarrow b + \eta(y - f) \qquad (2)$$

where $\eta$ is the adaptation rate. Observe from (2) that the weights can equally be represented as a weighted sum of the data samples, i.e. $w = \sum_{i=1}^{i=L} \alpha_i x_i$ and we re-write the Adaline in its *data dependent* (DD) form [4,9]:

$$f_p = \left\langle \sum_{i=1}^{i=L} \alpha_i x_i, x_p \right\rangle + b = \sum_{i=1}^{i=L} \alpha_i \left\langle x_i, x_p \right\rangle + b \qquad (3)$$

For any DD Perceptron such as (3), the equivalent update rules for the scalar multipliers, $\alpha_i$, and the bias, *b*, are given by:

$$\alpha_i \leftarrow \alpha_i + \eta e_i\ ,\ b \leftarrow b + \eta e_i \qquad (4)$$

where $e_i = y_i - f_i$ [4]. That this must be so is clear from the fact: $w \leftarrow w + \eta e_i x_i$ $= \sum_{j=1}^{j=L} \alpha_j x_j + \eta e_i x_i$ iff $\alpha_i \leftarrow \alpha_i + \eta e_i$, hence after one pass through the data, $w = \sum_{i=1}^{i=L} \alpha_i x_i$. The formulations of (3) and (4), and of (1) and (2) represent the *same* Perceptron, so the known properties of the LMS rule apply directly. In the DD formulation the update consists only of adding values to the $\alpha_i$, and of computing inner products of training patterns. This is central to the sequel.

### 2.1 A kernel Adaline for function approximation

If a fixed mapping of the input data $z_p = \varphi(x_p)$, were known a priori, where $\varphi(.)$ is rich enough to capture the underlying form of *f*, the data could be transformed and fitted by a linear combination of the *z*s [1]. Through the use of Mercer kernels it is possible to perform the mapping *implicitly*. Mercer kernels represent inner-products in some Hilbert space [10] and thus the mapping of two samples into a high-dimensional *linearisation* space, *Z*: $(x_u, x_v) \mapsto (\varphi(x_u), \varphi(x_v)) = (z_u, z_v)$ $z_u \in Z\ \forall u$, and the calculation of inner-products there, $k(x_u, x_v) = \langle \varphi(x_u), \varphi(x_v) \rangle = \langle z_u, z_v \rangle$. Any function satisfying Mercer's conditions may be used, such as the Gaussian radial basis function, $k_{rbf}(x_u, x_v) = \exp\left(-\|x_u - x_v\|^2 / 2\sigma^2\right)$, or the polynomial kernel, $k_{pol}(x_u, x_v) = \left(\langle x_u, x_v \rangle + 1\right)^d$. Each kernel has an associated parameter providing a design degree of freedom or "smoothing".

The scalar product of (3) may now be replaced with an inner product in *Z*, avoiding explicit expansion. The vector, *w*, now resides in *Z* and is not, therefore, accessible for update but in the DD form the *multipliers*, $\alpha_i$, are accessible so the scalar product in

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 245-250

(3) is replaced by a kernel and the $\alpha_i$ updated as per (4). Now the DD, *non-linear* Adaline is given by: $f(x_i) = \sum_{p=1}^{p=L} \alpha_p < \phi(x_p), \phi(x_i) > + b$ $= \sum_{p=1}^{p=L} \alpha_p k(x_p, x_i) + b$. The "representer" theorem guarantees that for any function, $f(x_i) = \sum_{p=1}^{p=L} \alpha_p k(x_p, x_i) + b \quad =< w, \varphi(x_i) > + b$, there exists a function in the kernel Hilbert space [11,12]. To allow for an affine solution in *Z*, the explicit bias, *b*, is maintained [7] so that augmentation of data vectors can be performed implicitly in *Z*. i.e. Augment $z$ thus: $\left[z^t \; 1\right]^t$, and noting that $b = \sum_{i=1}^{i=L} \alpha_i$ we have: $f(x_i) = \sum_{p=1}^{p=L} \alpha_p k(x_p, x_i) + b \qquad = \sum_{p=1}^{p=L} \alpha_p \left(k(x_p, x_i) + 1\right)$ $= \sum_{p=1}^{p=L} \alpha_p k_1(x_p, x_i)$ and since $k_1$ is positive semi-definite, the properties of the algorithm with augmented linearisation-space vectors follow from those given above [7,9]. Evidently, the update algorithm follows from the linear Adaline simply by replacing *x* with *z*. A pseudocode algorithm is given below.

**Kernel-Adaline Learning**

1. Choose $\alpha_i = 0, \; i = 1 \ldots L, \; b = 0$ and $\eta$
2. WHILE (early) stopping criterion not met
3. FOR $i = 1 \ldots L$
   choose an input $x_i \quad i \in [1, L]$
   calculate kernel-Adaline error $e_i = y_i - \sum_{p=1}^{p=L} \alpha_p k\left(x_p, x_i\right) + b$
   update corresponding multiplier and bias by the rule:
   $\alpha_i \leftarrow \alpha_i + \eta e_i; \; b \leftarrow b + \eta e_i$
4. END FOR
5. END WHILE

Monitoring training and out-of-sample (OOS) performance allows early stopping at the minimum point of the OOS cost (e.g. MSE, MAE) to regularise the solution. It can be shown that after *q* iterations using a learning rate, $\eta$, the vector of multipliers, $\alpha$, lies in a hyper-rectangle of size $q \times \eta$ in the dual space of the parameters $\alpha$. Since the DD structure is fixed, the weight vector, *w*, will also lie in a hyper-rectangle. Early stopping, therefore, implies weight decay regularisation both in the convex dual space of multipliers, and in the linearisation space implied by the kernel function (where weight vector, *w*, resides). The linearity of the problem in *Z* ensures that the well-known convergence properties of the LMS algorithm continue to apply to the kernel-Adaline. Our approach can also operate with Vapnik's $\varepsilon$-insensitive zone by ensuring that only those multipliers whose current errors exceed $\varepsilon > 0$ are updated every iteration, increasing solution sparsity but possibly biasing the result. To

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 245-250

expedite computation cache matrices can be used. The extension to multi-output problems is obvious.

## 3. Computer experiments

**Static curve fitting:** Here we fit the sinc function, uniformly sampled on the interval (-10, 10) with added noise $\sim N(0,0.04)$. Identifying a minimum in the OOS MSE and reverting to that point performs early stopping regularisation. A Gaussian kernel is used with three values of $\sigma$ and $\eta = .01$. A near zero value of training MSE for $\sigma = 0.2$ indicates that the data are over-fitted. This is seen in figure 1 (dotted line),

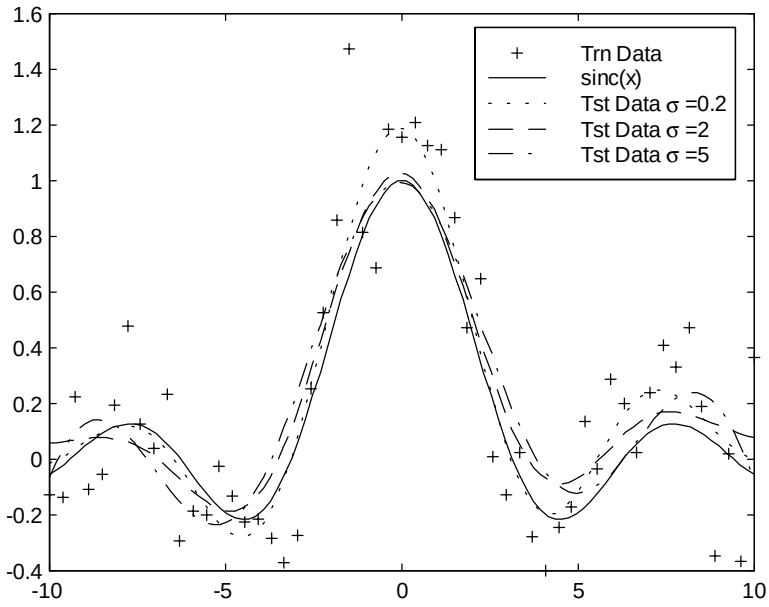


Figure 1: Demonstration of over-, correctly- and under-fitting the sinc function.

where the OOS behaviour is relatively poor (175 iterations). Widening the kernel $(\sigma = 2)$ improves this (dashed line) giving a good fit (35 iterations). Further widening the kernel $(\sigma = 5)$ might be expected to show severe under-fitting (chained line). Indeed, the learning task requires much iteration but early stopping at 5600 yields a tolerable fit. For $\sigma < 0.2$ gross over-fitting occurs while for $\sigma > 5$ excessive training is needed. There is, nonetheless, substantial latitude in the selection of the smoothing parameter provided that the solution is properly regularised.

**Non-linear filtering**: We can now use the kernel-Adaline to generalise the optimal linear filtering theory for non-linear systems identification, time series analysis and prediction, active noise cancellation, channel equalisation, signal deconvolution, delay estimation and inverse-based control. We use our new optimal filtering theory to

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 245-250

identify a non-linear, moving average system: $y(n) = 0.5x(n) + x(n-1)^3 + \varepsilon(n)$, with $\varepsilon \sim N(0,4)$ and $x \sim N(0,1)$ with independent increments. Early stopping is used as appropriate. Figure 2 compares the true and predicted OOS values for two kernel

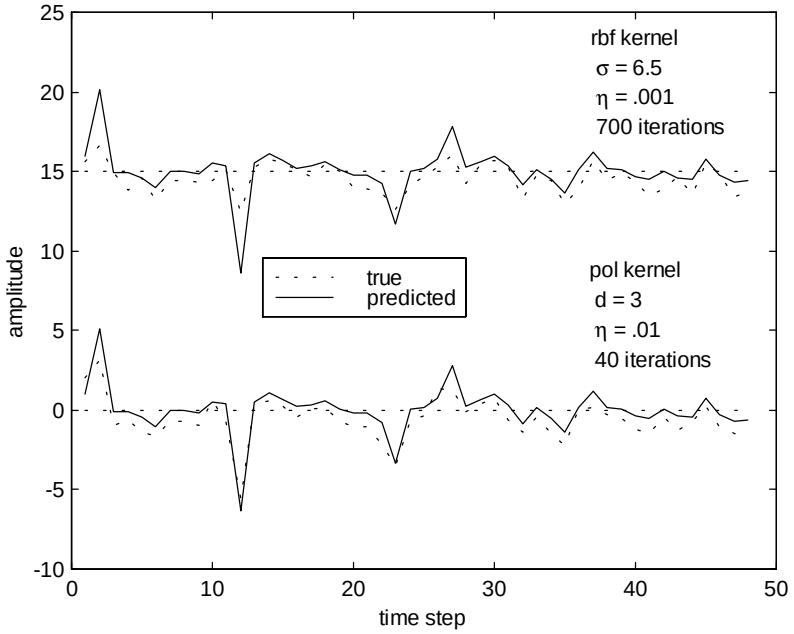


Figure 2: OOS behaviour for the non-linear moving average system for two kernel types. NB the false origin on the vertical axis for clarity.

choices, with unit time delay. Note that the predictive accuracy is high in both cases, despite the poor SNR of the training sample (containing 50 points). In [13] further experiments are described, including identification of an auto-regressive system.

## 4. Conclusions

The kernel-Adaline is a sequential and numerically robust algorithm of the potential function Perceptron type that uses kernels for high (possibly infinite) order expansions and applies the LMS rule in the high-dimensional linearisation space. A non-linear version of the linear Adaline has been developed whose convergence properties follow directly from analysis of the LMS algorithm as a consequence of operating in the linearisation space. The use of an $\varepsilon$-insensitive zone is a trivial extension of the algorithm.

Initial experiments show that, while the kernel-Adaline is subject to the usual (as yet unsolved) problems of model selection, this can be reduced to the choice of a kernel, its parameter and, for filtering, the filter length [13]. The required computations are trivial and are not troubled by local phenomena owing to the convexity of the

problem. They can be implemented efficiently by means of a cache matrix. It appears, therefore, that this procedure has much potential in the field of non-linear signal processing and function approximation.

The financial support of the EPSRC (TTF) and the hospitality of Princeton University (RFH) is gratefully acknowledged.

## References

[1] Aizerman, M., Braverman, E., et al, Theoretical foundations of the potential function method in pattern recognition learning. *Automation And Remote Control*, **25**, 821-837, 1964.

[2] Cortes, C. and Vapnik, V., Support-vector networks. *Machine Learning*, **20**, 273-297, 1995.

[3] Scholkopf, B., Smola, A., et al, Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**, 1299-1319, 1998.

[4] Friess, T.T. and Harrison, R.F., Perceptrons in kernel feature spaces. Research Report No. 720, 1998. The University of Sheffield: Sheffield.

[5] Grappel, T. and Obermayer, K., Fuzzy topographic kernel clustering. Proceedings of the Fifth GI Workshop on Fuzzy Neuro Systems,1998.

[6] Friess, T.T., Cristianini, N., et al. The kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines. In: *Machine Learning: Proceedings of the 15th International Conference*, ed. Shavlik, J.W. San Francisco: Morgan Kauffman, 1998. 188-196.

[7] Friess, T.T. and Harrison, R.F., The kernel Adatron with bias unit: analysis of the algorithm. Research Report No. 728, 1998. The University of Sheffield: Sheffield.

[8] Widrow, B. and Hoff, M., Adaptive switching circuits. Proceedings of the 1960 IRE WESCON Convention Part 4, 96-104, 1960.

[9] Friess, T.T. and Harrison, R.F., Support vector neural networks: the kernel Adatron with bias and soft margin. Research Report No. 725, 1998. The University of Sheffield: Sheffield.

[10] Wahba, G., Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV. Research Report No. 984, 1998. Department of Statistics, University of Wisconsin: Madison.

[11] Kimeldorf, G. and Wahba, G., Some results on Tchebycheffian spline functions. *Journal Of Mathematical Analysis And Applications*, **33**, 82-95, 1971.

[12] Cox, D. and O'Sullivan, F., Asymptotic analysis of penalised likelihood and related estimators. *Annals Of Statistics*, **18**, 1676-1695, 1990.

[13] Friess, T.T. and Harrison, R.F., The kernel Adaline: a new algorithm for non-linear signal processing and regression. Research Report No. 731, 1998. The University of Sheffield: Sheffield.