Pattern Discrimination Using Feed-Forward Networks - a Benchmark Study

Thorsteinn Rögnvaldsson

Dept. of Theor. Physics, Univ. of Lund, Sölvegatan 14 A, S-223 62 Lund, Sweden

The Multilayer perceptron (MLP) network is shown to handle high-dimensional pattern classification problems in a more efficient way than the Learning Vector Quantisation (LVQ) network. This is due to the sigmoidal form of the MLP transfer function, and to the the fact that the MLP uses hyper-planes more efficiently. Both algorithms are also found to be equally robust to limited training sets.

Introduction

The discrimination power of the Multilayer Perceptron (MLP) [1] and the Learning Vector Quantisation (LVQ) [2] is compared on the task of separating two Gaussian distributions (two classes are sufficient since the results carry over to problems with more classes). The problem is designed to resemble "real-life" situations, with many input nodes and overlapping distributions, making discrimination possible only to the *Bayes limit*. It is shown that the MLP is more efficient than the LVQ algorithm on heavily overlapping distributions. Furthermore, the sensitivities of the two algorithms to limited number of training data are examined and the MLP is found to be at least as efficient as the LVQ algorithm.

Analysis

Two different benchmark problems are studied: One with two Gaussian distributions with equal mean values (referred to as "hard" case), and one where the distributions have different means (referred to as "easy" case). Both distributions are normalised to one but with standard deviations differing by a factor of two (for more details and references see ref. [3]). The optimal classification boundary between the two distributions is a d-dimensional hyper-sphere, where d is the number of inputs. The goal of the algorithms is to reproduce this surface. The MLP uses "hyperplanes" (see fig. 1a) whereas the LVQ network divides the space by "tessellation" (fig. 1b). If the MLP uses a step transfer function both algorithms produce a polyhedral-like reproduction of the sphere (fig. 1c) and the generalisation error E can be expressed as

$$E \approx B + \Delta V \Delta P \tag{1}$$

where B is Bayes limit, ΔV the deficit volume of the polyhedron as compared to the hyper-sphere, and ΔP the average absolute difference between the two distributions inside that volume. For the "hard" case this can be shown to be [3];

$$\frac{E-B}{B} \propto dN^{-4/(d-1)} \tag{2}$$

where N is the number of hyperplanes bounding the polyhedral. The generalisation error in the "easy" case is bounded from above by (2) and never scales worse than this.

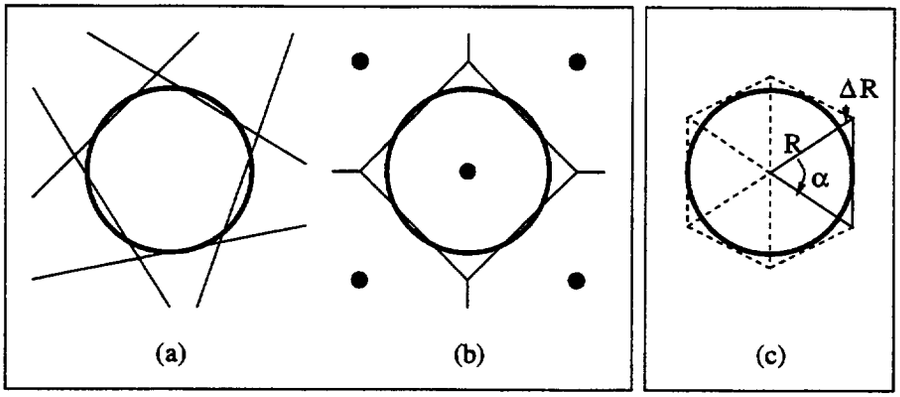

Figure 1: (a) The MLP approximation to a sphere and (b) the LVQ tessellation for the same sphere (dots correspond to reference vectors). (c) The resulting polyhedron.

Each hidden unit in an MLP with a step transfer function corresponds to one hyper-plane. Hence, an MLP with N_{MLP} hidden units has $N = N_{MLP}$. However, a smooth sigmoidal transfer function allows the MLP to "cut corners" in the polyhedron, and the generalisation error scales better than (2). The reference vector method of LVQ is more expensive than using hyperplanes and the LVQ generalisation error will scales like eq. (2) for moderate N_{LVQ} and worse for large N_{LVQ} (N_{LVQ} is the number of reference vectors).

Monte Carlo Simulations

The number of input units d was varied and 12 different architectures of MLP and LVQ networks were set up for each value of d. An ensemble of 100 networks was trained (using `JETNET 2.0` [4]) for each configuration to measure the worst, best, and average performances. Figure 2 shows the $E(N)$-behaviour for $d = 8$. The LVQ errors follow the predicted behaviour well, whereas the MLP errors scale better. The simulation results are thus in good agreement with the analytically predicted behaviours.

The largest architectures were also trained with 7 different training sets of sizes $M \in \{10, 10^{3/2}, 10^2, 10^{5/2}, 10^3, 10^{7/2}, 10^4\}$. The resulting learning curves are well described by $E \propto 1/(M + M_0)$ and the algorithms seem to be (approximately) equally robust to limited training sets [3].

Conclusion

The results demonstrate that the MLP architecture, with sigmoidal transfer functions, is superior to LVQ for discriminating between heavily overlapping distributions with convex borders. This is in contrast to previous results comparing LVQ and MLP architectures [5]. Furthermore, both algorithms are equally robust to limited training data and their learning curves follow a $1/M$ behaviour. The results are valid also for binary problems with more than two classes of input patterns. An MLP with n output units can be used for a n-class problem. The output values correspond to the Bayesian *a posteriori* likelihoods that the input pattern belongs to the specific class [6] and a discrimination decision can be made from these output values. The number of

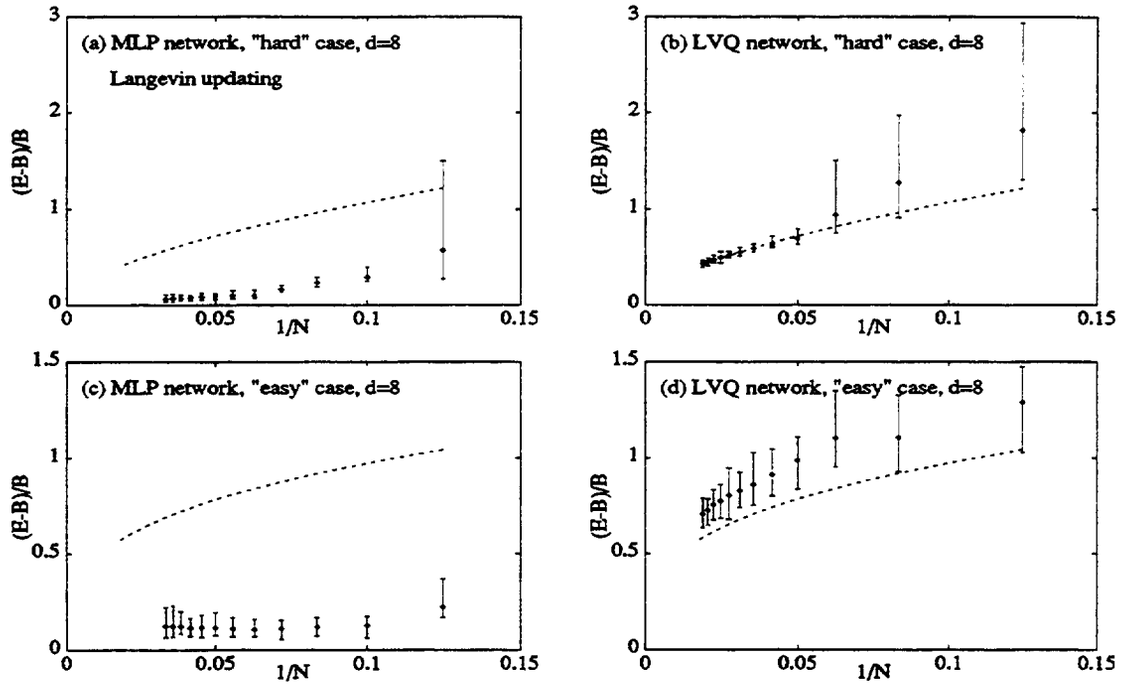

Figure 2: Minimum, maximum and average values of the scaled generalisation error for $d = 8$ using (a) MLP and (b) LVQ on the "hard" task, and (c) MLP and (d) LVQ on the "easy" task. The dashed line in (a) and (b) is the predicted $(E - B)/B \propto dN^{-4/(d-1)}$, whereas in (c) and (d) it is a fitted $(E - B)/B \propto dN^{-5/2d}$

hidden units will in the worst case just be multiplied by n, as for the LVQ network. The relative performances of MLP and LVQ will thus be approximately unchanged. LVQ may however be more efficient than MLP for problems with large $n \gg d$ where each class may need only one or two reference vectors.

References

[1] D. Rumelhart and J. McLelland (eds.) *Parallel Distributed Processing* (Vol. 1), MIT Press (1986)

[2] T. Kohonen, *Self-organization and Associative Memory*, 3^{rd} ed., Springer-Verlag, Heidelberg (1990)

[3] T. Rögnvaldsson, "Pattern Discrimination Using Feed-Forward Networks - a Benchmark Study of Scaling Behaviour", Lund Preprint LU TP 92-18 (1992)

[4] L. Lönnblad, C. Peterson and T. Rögnvaldsson, "Pattern Recognition in High Energy Physics with Artificial Neural Networks - JETNET 2.0", *Computer Physics Communications* **70** (1992)

[5] T. Kohonen, G. Barna and R. Chrisley, "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies", *IEEE* 2^{nd} *Int. Conf. on Neur. Netw.* I, S.Diego, CA: SOS Printing (1988); G. Barna and K. Kaski, "Stochastic vs. Deterministic Neural Networks for Pattern Recognition", *Physica Scripta* **T33** (1990)

[6] M. Richard and R. Lippmann, "Neural Networks Estimate Bayesian *a posteriori* Probabilities", *Neural Computation* **3** (1991)