Hyperplane Training of a Hypersphere Classifier

Yaakov Stein
Efrat Future Technology Ltd.
110 Yigal Alon St.
Tel Aviv 67891, Israel

Abstract

A novel classifier architecture is introduced which belongs to both hyperplane and hypersphere families. The basic computational unit in the architecture is a perceptron whose input is augmented by its squared length. Traditional methods of training hyperplane classifiers (perceptron training algorithm, backpropagation, etc.) function in the augmented input space, and induce hyperspherical decision regions in the original input space. The multilayer architecture based on these units includes, as specific cases, the multilayer perceptron and the radial basis function networks.

1 Hypersphere and Hyperplane Classifier Architectures

One conventionally distinguishes between three types of statistical pattern classifiers, namely example based classifiers (eg. *k nearest neighbors*), hypersphere classifiers (such as the *radial basis functions* or RBF network) (Broomhead and Lowe 1988), and hyperplane classifiers (such as the *multilayer perceptron* or MLP network) (Rumelhart and McClelland 1986). Example based classifiers may not require training, but suffer from large memory requirements, long classification times, and do not, in general, attain the minimal possible (Bayes) classification error. They will not be considered further here. Hypersphere classifiers have modest training and classification times, excellent false alarm rejection rates and can attain Bayes error with large training sets. Hyperplane classifiers may require longer training times but classify faster. They also attain minimal error, but have virtually no inherent false alarm rejection capabilities (Stein et al 1993). While hypersphere classifiers endeavor to capture the class probability distributions, hyperplane classifiers only try to find inter-region boundaries.

Hyperplane classifiers have proven to be more popular than hypersphere ones in practice, for several reasons. The simplest hyperplane classifier, the perceptron, can be trained in a finite number of steps, at least when a separating hyperplane exists. The multilayer perceptron can create arbitrary decision regions, and tends to have somewhat lower misclassification rate than hypersphere classifiers for small training sets, due to the more efficient use of examples during training. The most popular MLP training algorithms are variants of backpropagation, which do not usually converge to a global optimum, but are straightforward to implement. RBF training methods either call for a clustering stage, or arbitrarily chose a small number of input examples as bases.

It would thus be beneficial to combine the best features of hypersphere and hyperplane classifiers. Such a combined classifier would have simple training procedures as well as low misclassification and false alarm rates. In the sequel we propose such an architecture, which can be implemented by making only minimal changes to existing MLP systems.

The connection between neurons which compute the norm of a differences and those which compute compute inner products has been studied previously (Seligson et al 1992) with the objective of replacing the latter with the former. That work proved formal equivalence for neurons with binary input and output, and demonstrated empirically the inferiority of the difference neuron for other cases. The present work is complementary in the sense that difference like decision regions are obtained by exploiting product neurons.

2 The Augmented Perceptron

A tactic often employed by pattern recognition practitioners is to add auxiliary variables to the input. Such auxiliary variables are produced by combining the original input variables in ways that the classifier itself can not. For the simple hard-limiting perceptron, which classifies an input pattern as positive or negative based on *linear* combinations of the input variables x_i

$$\sum_{i=1}^{N} a_i x_i \gtrless 0 \tag{1}$$

one might propose to append powers of inputs. This augmentation of the input space is normally performed unsystematically, perhaps based on intuition or a priori knowledge.

We propose augmenting a perceptron's input with a single variable, the squared length

$$x_{N+1} \equiv \sum_{i=1}^{N} x_i^2 \ .$$

We shall now show that a perceptron operating in the augmented $N+1$ dimensional space induces, in general, hyperspherical decision regions in the original space. Hyperplane boundaries, which are obtained when the coefficient of x_{N+1} is zero, can be considered as limiting cases of hyperspheres with infinite radii. We will need the following lemma.
Lemma. When the $N+1$ dimensional paraboloid surface

$$x_{N+1} = \sum_{i=1}^{N} x_i^2 \tag{2}$$

has a nonempty intersection with the N dimensional hyperplane

$$\sum_{i=1}^{N+1} a_i x_i = \theta \tag{3}$$

then its projection onto the N dimensional space spanned by $x_1 \ldots x_N$ is a hyperspherical surface. Conversely, every hyperspherical surface in N dimensional space can be mapped onto the intersection of the paraboloid surface with a hyperplane.

Proof. The intersection is determined by

$$\sum_{i=1}^{N} a_i x_i \; + \; a_{N+1} \sum_{i=1}^{N} x_i^2 = \theta$$

which is of the form of a hyperspherical surface of radius r and center c

$$\sum_{i=1}^{N} (x_i - c_i)^2 = r^2$$

when we make the identification

$$\begin{aligned} c_i &= -\frac{1}{2} a_i \\ r^2 &= \theta + \sum_{i=1}^{N} c_i^2 \; . \end{aligned} \tag{4}$$

It is easy to see that when the intersection is nonempty, the rhs of the second identity is non-negative. The converse is proven similarly.

It is instructive to consider the two dimensional case. The lemma states that the projection onto the x-y plane of the intersection of a plane with the paraboloid of revolution $z = x^2 + y^2$ is always a circle. Were we to replace the paraboloid surface with the cone $z' = \sqrt{x^2 + y^2}$, the projection would then be a conic section. Note that even for our paraboloid, the intersection itself (not its projection) *is* an ellipse.

In order to return to the problem at hand, we must change the equalities in (2) and (3) into inequalities, thus turning the hyperplanes into half spaces, and the hyperspherical surfaces into hyperspheres. In addition, we must allow for the case where the paraboloid surface and hyperplane do not intersect.

Theorem. The augmented hard-limiting perceptron classifier, defined as

$$\sum_{i=1}^{N} a_i x_i \; + \; a_{N+1} \sum_{i=1}^{N} x_i^2 \stackrel{>}{<} \theta$$

induces decision regions in the original space, of one of the following five forms :

1. the empty set,
2. the entire space,
3. a half space,
4. a hypersphere (including a single point),
5. the entire space except for a hypersphere.

Proof. The perceptron defines a positive half space delimited by its hyperplane. We are interested in the projection onto the original space of that portion of the surface of the paraboloid which is in the positive half space. There are five possibilities, to which correspond the five cases:

1. the perceptron hyperplane does not intersect the paraboloid surface and the entire paraboloid is in its negative half space, resulting in an empty positive decision region,

2. the perceptron hyperplane does not intersect the paraboloid surface and the entire paraboloid is in its positive half space, resulting in the entire space being in the positive decision region,

3. the perceptron hyperplane is orthogonal to the original space (ie. the coefficient of x_{N+1} is zero), and thus the positive decision region is the half space delimited by the perceptron,

4. the perceptron hyperplane intersects the paraboloid surface and large x_{N+1} is in the negative half space, and thus, according to the lemma, the positive decision region is a hypersphere,

5. the perceptron hyperplane intersects the paraboloid surface and large x_{N+1} is in the positive half space, thus, according to the lemma, the negative decision region is a hypersphere and the positive region includes all points exterior to that hypersphere.

We have thus seen that the augmented perceptron produces one of five types of decision regions. The half space is less probable in general, and in any case the first three cases can be thought of as hyperspheres with zero or infinite radii, thus we are justified in saying that the regions induced are always hyperspherical. From the converse part of the lemma, we know that all hyperspherical regions in the original space correspond to half spaces in the augmented space. Thus when a hyperspherical decision region exists it always corresponds to an augmented perceptron. Exploiting the perceptron learning theorem (Minsky and Papert 1969) we conclude

Theorem. Assuming that a hyperspherical decision region exists, one can be found in a finite number of steps.

3 Augmented multilayer perceptrons

We now turn to the application of the augmented perceptron as a building block in more complex architectures. In order to enable gradient descent learning, multilayer systems usually employ smooth sigmoidal perceptron rather than the hard-limiting ones we dealt with in the last section. This does not significantly change any of the results obtained so far.

The simplest method of utilizing the basic unit is to build an 'input augmented MLP', wherein only the first hidden layer contains augmented perceptrons, the other layers consisting of conventional perceptrons. Pre-existing MLP systems can be converted at minimal expense to this architecture. Input augmented MLP systems have at least the capabilities of the comparable nonaugmented systems, but in addition to being able to form decision regions in input space bounded by hyperplanes, they can form decision regions bounded by hyperspherical surfaces.

The single hidden layer input augmented MLP, is equivalent to the RBF network (Broomhead and Lowe 1988) with a particular basis function. This network is known

to be capable of forming arbitrary decision boundaries. Similarly, using hard limiting perceptrons and hard wired perceptron outputs, we obtain the RCE network (Reilly et al 1982). We are currently in the process of testing the input augmented MLP in several real world applications.

Although the single hidden layer input augmented MLP can already form arbitrarily shaped decision regions, one need not stop there. One can consider multilayer systems with augmentation of several or all layers.

The last question to be addressed is that of the expected generalization. The squared distance between two points in augmented space is equal to their squared distance in input space plus the square of the difference in their squared lengths. Thus two points which are close in input space, may be far removed in augmented space. The significance of this effect varies from place to place and is also dependent on the orientation of the points. Thus test set points, which *seem* to be close to training set points, may actually give quite different results.

4 Remarks

There are cases of interest when the augmentation is ineffective. The most prominent is that of binary patterns, for which single hyperplanes and hyperspheres have the same capabilities (Seligson et al. 1992). To see this more clearly, consider the input patterns to be corners of the N dimensional hypercube $\{S_i\}_{i=1,\ldots,N}$, $S_i = \pm 1$. Since all patterns have the same squared length N, the augmentation has no effect.

It should *not* be surprising that a linear classifier can produce hyperspherical decision regions, since the nonlinearity is specifically introduced. Similarly, one can easily map the inside of a circle to a half plane by a suitable rectangular to polar transformation, without increasing the dimension. However, actual use of such a transformation in training a classifier would require a priori knowledge regarding the circle's center. The augmented perceptron discovers the appropriate hypersphere center as part of the standard learning process.

One can obtain more general elliptical regions in a similar manner, by augmenting the perceptron with $N(N+1)/2$ new dimensions. This significantly increases the number of augmented perceptron weights which must be learned. For example, in the two dimensional case we must augment x and y with x^2, y^2 and xy.

Acknowledgements Fruitful discussions with Y. Metzger and R. Aloni-Lavi are gratefully acknowledged.

References

[1] Broomhead D.S. and Lowe D. 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems* **2**, 321-323

[2] Minsky M. and Papert S. 1969. *Perceptrons.* MIT Press, Cambridge, Mass.

[3] Reilly D.L., Cooper L.N. and Elbaum C. 1982. A neural model for category learning. Biol. Cyber. **45** 35-41.

[4] Rumelhart D.E., Hinton G.E. and Williams R.J. 1986. Nature, **323**, 533-536.

[5] Seligson D., Griniasty M., Hansel D. and Shoresh N. 1992. Computing with a difference neuron. Network **3** 187-204.

[6] Stein Y., Aloni-Lavi R. and Metzger Y. 1993. Reducing the false alarm rate of multilayer perceptrons. to appear.