7-1980

Bounds on the Number of Samples Needed for Neural Learning

Kishan G. Mehrotra
Syracuse University, mehrtra@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Sanjay Ranka
Syracuse University

SU-CIS-90-20

Bounds on the Number of Samples Needed for Neural Learning

K. Mehrotra, C.K. Mohan, S. Ranka

July 1990

School of Computer and Information Science
Syracuse University
Suite 4-116
Center for Science and Technology
Syracuse, New York 13244-4100

Bounds on the Number of Samples Needed for Neural Learning

Kishan G. Mehrotra, Chilukuri K. Mohan, Sanjay Ranka
School of Computer and Information Science
4-116, Center for Science and Technology
Syracuse University, New York 13244-4100
315-443-2368
kishan/mohan/ranka@top.cis.syr.edu

July 20, 1990

Abstract

This paper addresses the relationship between the number of hidden layer nodes in a neural network, the complexity of a multi-class discrimination problem, and the number of samples needed for effective learning. Bounds are given for the latter. We show that $\Omega(min(d, n).M)$ boundary samples are required for successful classification of M clusters of samples using a 2 hidden layer neural network with d-dimensional inputs and n nodes in the first hidden layer.

1 Introduction

In recent years, multilayer neural networks have been increasingly popular for applications in pattern recognition, classification, learning, and function approximation. While there is a profusion of empirical results attesting to the usefulness of neural learning techniques, the capabilities, limitations and requirements of neural networks are relatively less well understood. Many important issues (*e.g.*, how many training samples are required for successful learning, how large a neural network is required for a specific task) are solved in practice by trial-and-error. Some results have been achieved recently in an attempt to solve these piquant problems, but the area is largely open for investigation. These questions are hard because there is considerable dependence on the specific problem being attacked using a neural network.

With too few nodes, the network may not be powerful enough for a given learning task. With a large number of nodes (and connections), computation is too expensive. Also, a neural network may have the resources essentially to 'memorize' the input training samples; such a network typically performs poorly on new test samples, and is not considered to have accomplished learning successfully. For neural learning to be considered successful *learning*, it is essential for the system to perform correct classification of test samples on which the system has not been trained. We emphasize capabilities of a network to *generalize* from input training samples, not to memorize them.

In this paper, we address the question of how many samples are needed for adequately successful learning using a 2 hidden layer neural network (Figure 1). As pointed out by several researchers (*e.g.*, [7] [8]), in a 2 hidden layer neural network with d input nodes, first hidden layer nodes often function like hyperplanes that effectively partition d-dimensional space into various regions. Each node in the second hidden layer represents a cluster of points that belong to the same class. We assume that the problem that the neural network is trying to learn is such that these clusters are separable. Other attempts to answer this question are inadequate due to the unrealistic assumptions made, *e.g.* [9], that a single input sample is sufficient to characterize each cluster of inputs.

1.1 Main Results

In the next section, we enumerate the minimum number of 'hyperplane segments' when a given neural network is used successfully for a classification task. This is then related to the number of clusters in a given problem being solved using a neural network. We then

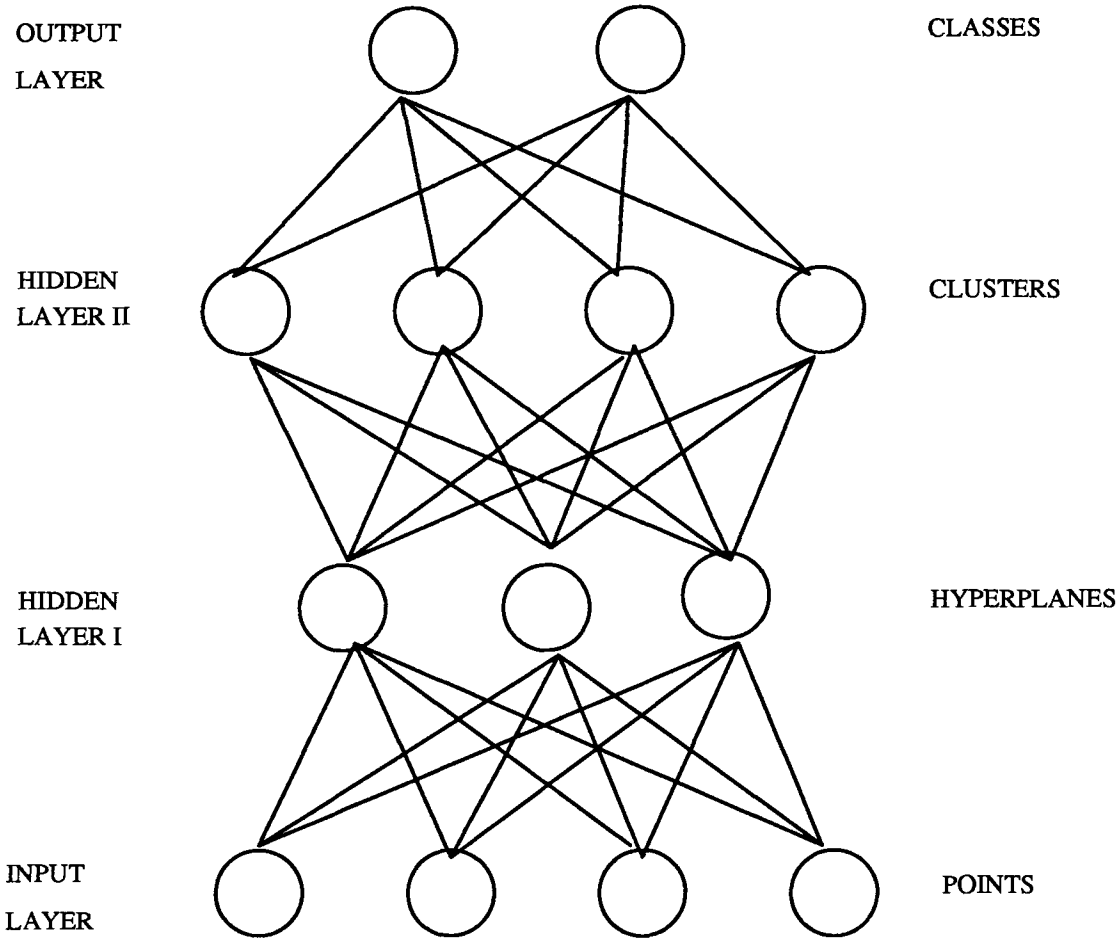

Figure 1: A 2 hidden layer neural network

consider the case where the number of hyperplanes (hidden nodes) exceeds the minimum number required for successful classification of a given number of clusters, and estimate the number of hyperplane segments and regions in such cases. Finally, we establish bounds on the number of samples needed for successful learning. Discussion and conclusions then follow. To improve readability of the main body of the paper, proofs and calculations are relegated to the appendix.

The results obtained in this paper are summarized as follows:

1. Assume that the input sample clusters are in 'general position' in d-dimensional space, *i.e.*, no subset of $(d+1)$ samples lies in a $(d-1)$ dimensional hyperplane [1]. At least $\Omega(n\sum_{i=0}^{d-1} C_i^{n-1})$ boundary samples are required for successful learning of a classification problem using a neural network with least possible number n of hidden nodes[2].

2. If there are groups of $p_1, \cdots, p_k$ hyperplanes in $d \geq 2$ dimensional space, such that all and only the hyperplanes from different groups intersect (i.e. hyperplanes within a group are parallel to each other), then the number of regions formed by these is:

$$R_k = \sum_{i=0}^{d} \left[\sum_{\forall j.a_j < a_{j+1}} p_{a_1} \cdots p_{a_i} \right] \text{ in general, and } \prod_{i=1}^{k}(1+p_i) \text{ when } k \leq d.$$

3. In the same situation, the number of hyperplane segments is

$$A_k = \sum_{i=1}^{d} \left[i.[\sum_{\forall j.a_j < a_{j+1}} p_{a_1} \cdots p_{a_i}] \right] \text{ in general, and } \sum_{i=1}^{k} \left[p_i . \prod_{\substack{j=1 \\ i \neq j}}^{k} (1+p_j) \right] \text{ when } k \leq d.$$

4. The ratio A_k/R_k lies between $min(k,d)/2$ and $min(k,d)$.

5. For a given classification problem with M clusters of samples in d-dimensional input space, such that adjacent clusters belong to different classes, if a sufficiently powerful neural network with n hidden nodes is chosen, then the number of boundary samples required is at least $\Omega(M.\min(n,d))$.

6. Under the assumption that $\Omega(d)$ samples are needed in d-dimensional space to identify each hyperplane segment, the number of boundary samples needed to learn (successfully) M clusters of training samples is at least $\Omega(Md^2)$; if each hyperplane segment

[1] For the rest of the paper we assume hyperplanes to be of dimension $(d-1)$, unless otherwise specified

[2] For the rest of the paper hidden nodes will refer to nodes in the first hidden layer, unless otherwise specified

can be learnt by only a constant number of samples, then only $\Omega(Md)$ boundary samples are needed for successful learning.

1.2 Related Work

Cover's work [5] established the number of dichotomies that can be implemented by a single threshold unit. Given n samples in general position in d-dimensional space, Nilsson [10] showed that a network with one hidden layer containing $n-1$ units was capable of learning any dichotomy; Baum [3] showed that $\lceil n/d \rceil$ hidden units are sufficient and necessary.

In [11], the learning time for back-propagation networks is examined in the context of learning boolean logic equations. The learning time is observed to increase with training sample set size, motivating the question: 'how many samples are adequate'?

By a different approach and with different assumptions, Baum and Hassler [4] show that any learning algorithm using lower than $\Omega(\frac{Number\ of\ weights}{\epsilon})$ random training samples will (for some distributions) fail a fixed fraction of the time to correctly classify more than $(1-\epsilon)$ fraction of the future test samples.

Our analysis, which follows a different approach, gives bounds for the number of boundary samples for effective generalization. Boundary samples are samples of each class that are 'near' (in the input space) samples of a different class. The importance of boundary samples ('salient examples', 'near-misses') has been recognized in the research area of machine learning [6]. Experimental results [1] confirm that boundary samples are better than random samples for training neural networks. Ahmad and Tesauro [1] also mention the number of samples required in practice for learning the linearly separable majority function mapping d-dimensional inputs to two classes: at least $3d$ boundary samples were required, but generalization was better with more samples ($7d$). These numbers are in accordance with our results in this paper.

In [9], a lower bound for the number of training samples required for neural learning is given to be the number of regions in d-dimensional space. The analysis in [9] assumes that one point in the input space is sufficient to identify a region in d-dimensional space. When the task is that of learning or classification, particularly under noisy conditions, a region is identified by a cluster of points, not just one point in the region. For good "generalization" capabilities, it is not useful for a neural network merely to memorize each input sample and the class to which it belongs. When only a few points in each region are given as training samples, there is much greater leeway in formulating the hyperplanes which divide these points into different regions, and hence the network is not likely to classify new test

samples correctly.

The dependence of the number of samples needed for successful learning on the number of hyperplane segments is illustrated in figure 2. Figure 2(a) depicts two (linearly separable) classes of points in two-dimensional space, whose partition has to be learnt by a neural network. Encircled points are the samples actually presented to the neural network, in each of three cases. In figure 2(b), a few arbitrary samples are presented to the neural network, but the network's performance on test samples may be poor, since it may learn any among a wide range of lines (each of which can partition the training samples) which result in high misclassification errors. Figure 3(c) illustrates how the situation is improved by presenting boundary samples (close to the intended partition) in the training phase; however, a sufficient number of them has not been presented, hence learning performance still has considerable scope for improvement. Finally, figure 2(d) indicates that performance is significantly improved with a larger number of boundary samples. Now the partitioning line is much more restricted, and has only a small degree of freedom.

2 Pairwise Intersecting Hyperplanes

A "hyperplane segment" is defined as a continuous part of a hyperplane, possibly bounded by its intersection with other hyperplanes. A "region" is a section of d-dimensional space separated from other regions by some hyperplane segments. Possibly (but not necessarily), each region could belong to a separate category for classification purposes. In this section, we evaluate the number of regions and hyperplane segments formed by the intersection of various hyperplanes separating points in general position in d-dimensional space.

Each region can be uniquely identified by the hyperplane segments which bound it together with the information as to which 'side' of the hyperplane segment it lies. There must be sufficient number of samples to identify each hyperplane segment. This is the reason for our interest in the number of hyperplane segments. Figure 3 illustrates certain regions ($a - k$) and hyperplane (line) segments ($1 - 16$) formed in 2-dimensional space by four intersecting lines.

2.1 Number of Regions (R)

The first issue we address is that of enumerating the number of different clusters of samples that may be distinguishable using a neural network with a given number of hidden

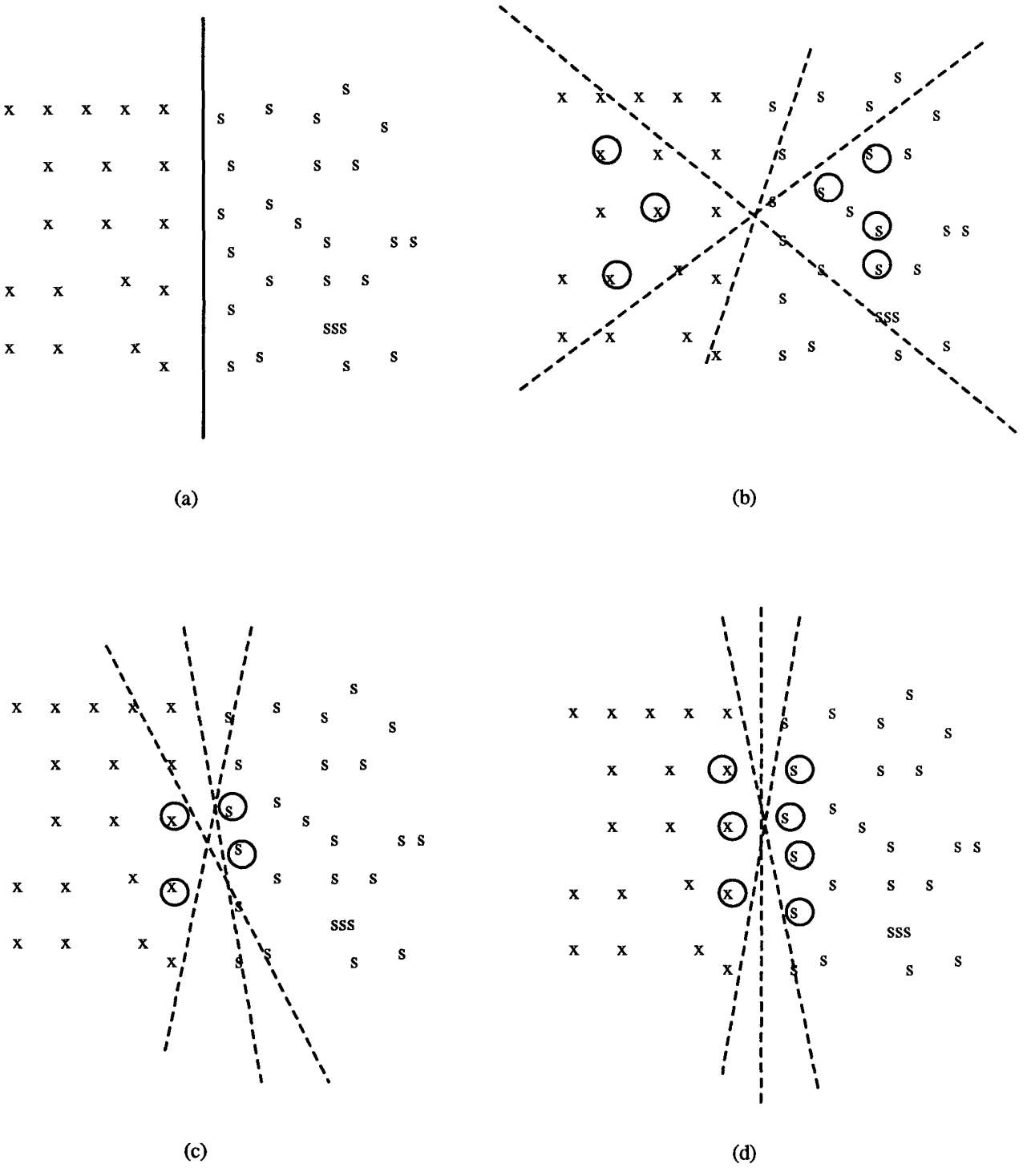

Figure 2: Boundary points vs non-boundary points

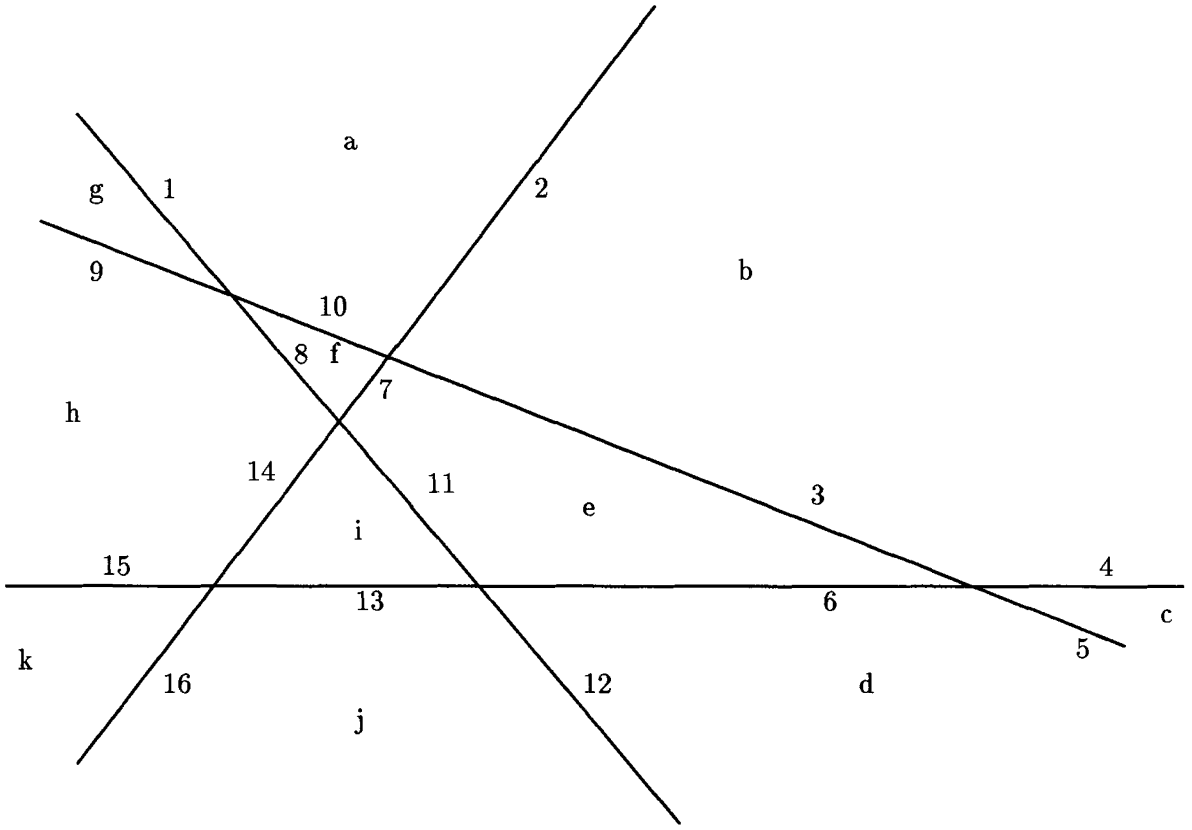

Figure 3: Hyperplane segments (1-16) and regions $(a - k)$, in 2 dimensions

nodes. The problem can be reformulated as that of enumerating the number of regions in hyperspace obtained using a given number of hyperplanes. By "regions", we denote distinct partitions of the hyperspace. Note that each *class* of input samples may consist of a number of *clusters* of points distributed in different regions of the relevant hyperspace.

Let $R(n, d)$ denote the maximum number of regions into which a d-dimensional hyperspace can be partitioned using n (mutually intersecting) hyperplanes. Note that $R(n, 1) = n + 1$, and $R(1, d) = 2$, while $R(n, 0)$ is assumed to be 0. It has been shown [13], [10], that

$$R(n, d) = \sum_{i=0}^{d} C_i^n \text{ where } C_i^j = \frac{j!}{i!(j-i)!} \text{ for } j \geq i, \text{ and 0 otherwise.}$$

If the given classification problem has M clusters of points, and the neural network is chosen to have the least possible number of hidden nodes n, then, in general, we must have $R(n, d) \geq M$, but $R(n-1, d)$ may be less than M. This condition is used below: the analysis is in terms of the number of hyperplanes partitioning the clusters, which is assumed to correspond to the number of hidden nodes. Note, however, that as many as $M - 1$ hyperplanes may be needed in the worst case to separate M clusters, *e.g.*, to separate M collinear points in two-dimensional space, when adjacent points belong to different classes.

2.2 Number of Hyperplane Segments (A)

Given n hyperplanes in d-dimensional space, we ask how many hyperplane segments are obtained by their mutual intersections. We now obtain an expression to evaluate "$A(n, d)$", the maximum number of hyperplane segments formed by mutual intersections among n hyperplanes in d-dimensional space. Obviously, when there are no hyperplanes, $A(0, d) = 0$; also note that $A(n, 1) = n+1$. Clearly, since each hyperplane contains at least one segment, $A(n, d) \geq n$. Since no hyperplane can be divided into more than 2^{n-1} segments by the $n-1$ other hyperplanes, we must also have $A(n, d) \leq n.2^{n-1}$. An exact value is now obtained.

Each hyperplane segment in d-dimensional space corresponds to a region in $(d-1)$-dimensional space. For instance, when $d = 3$, a hyperplane segment corresponds to a 2-dimensional region on the surface of the relevant hyperplane. In the maximal case, all hyperplanes intersect each other, thereby dividing each hyperplane into various segments. The maximum number of segments of a given hyperplane (obtained in this way) corresponds to the number of $(d-1)$-dimensional regions formed by the $(n-1)$ other hyperplanes which intersect the relevant hyperplane. This is just $R(n-1,\ d-1)$, since we can view each hyperplane as itself being a $(d-1)$-dimensional space in which there are $(n-1)$

hyperplanes (in the smaller dimension), formed by the intersections with the original (d-dimensional) hyperplanes. Since each of the n original hyperplanes can thus be divided into $R(n-1,\ d-1)$ segments in this way, the total number of hyperplane segments is

$$A(n,d) = n.R(n-1,\ d-1) = n\sum_{i=0}^{d-1} C_i^{n-1}.$$

2.3 Bounds for the Ratio A/R

The above formula for $A(n,d)$ is an abstract problem-independent property of a neural network with a given number of input nodes and hidden nodes. But if the network can successfully accomplish learning a specific task with M clusters of samples, we know that $R(n,d) \geq M$. So, to bring in consideration of the specific task to be learned, we consider the ratio $A(n,d)/R(n,d)$, which can help us obtain better bounds on the number of samples required for successful learning in a specific case. Naturally, if $R(n,d) < M$ for a given problem, the number of hidden nodes is inadequate for successful learning, and the following analysis is not applicable. Note that $n > n'$ iff $R(n,d) > R(n',d)$ iff $A(n,d) > A(n',d)$.

Simple bounds are derived for the ratio

$$\frac{A(n,d)}{R(n,d)} = \frac{n\ R(n-1,d-1)}{R(n,d)} \tag{1}$$

Summarizing the analysis in appendix 6.1, we have:

Theorem 1

$$\frac{min(n,d)}{2} \leq \frac{A(n,d)}{R(n,d)} \leq \frac{min(n,2d)}{2}.$$

3 Groupwise Intersecting Hyperplanes

The preceding analyses, for a least number of hidden nodes, are now generalized by answering the following question. Given a specific classification task with M clusters, and given a (fixed) neural network which is adequate for a classification task, how many samples are required for successful learning? An answer is obtained by generalizing the idea that all the hyperplanes intersect each other.

Previously, we had assumed that the clusters in the given problem are in general position, and the number of hidden nodes n in the network is chosen, according to the

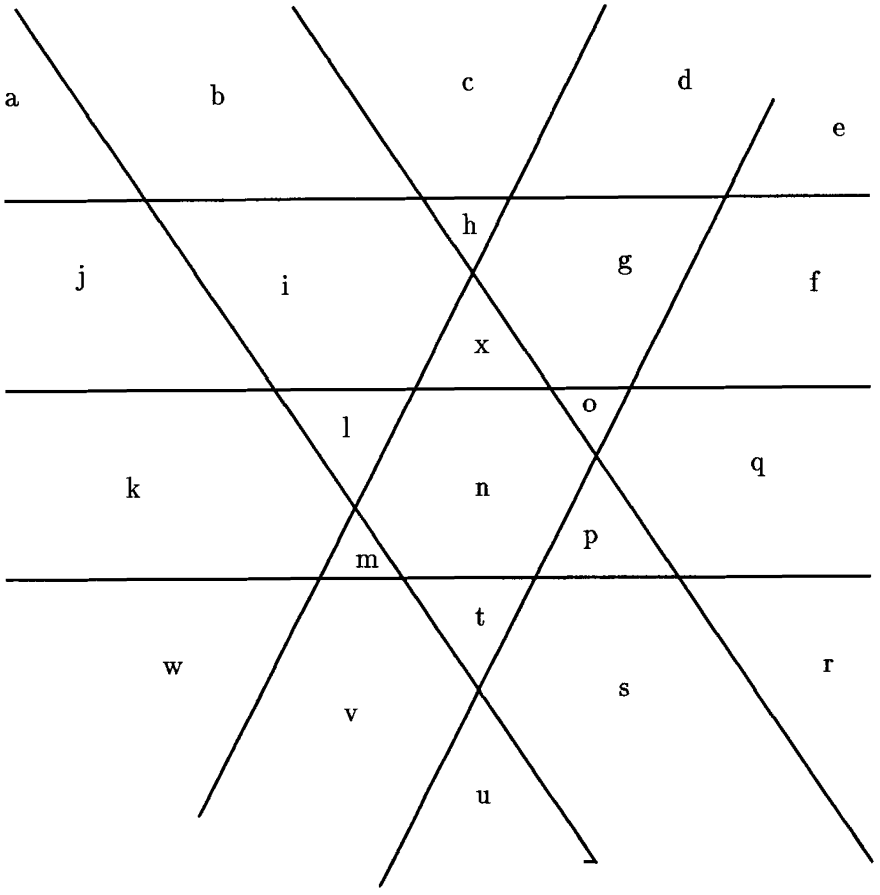

Figure 4: Groups of 2,2, and 3 parallel hyperplanes, with $d = 2$

requirements of the problem, such that $R(n, d) \geq M > R(n-1, d)$. We now assume only that $R(n, d) \geq M$, so that neural learning is possible. For instance, some of the hyperplanes may be mutually 'parallel' (non-intersecting), instead of being in general placement. In one extreme case, none of the hyperplanes intersect, so that we require $M - 1$ hyperplanes, and there are as many hyperplane segments; so if $\Omega(d)$ boundary samples are required to identify each hyperplane, we require altogether $\Omega(d.(M-1))$ boundary samples. In general, we may assume that there are groups of $p_1, \cdots, p_k$ parallel hyperplanes, where $\sum_{i=1}^{k} p_i = n$.

Let '$R_k(p_1, \cdots, p_k, d)$' denote the number of regions formed in d dimensions by n hyperplanes, consisting of k groups of 'parallel' p_i hyperplanes; *i.e.*, no two hyperplanes in the same group intersect. Similarly, let '$A_k(p_1, \cdots, p_k, d)$' denote the number of hyperplane segments formed by k groups of 'parallel' p_i hyperplanes, in d-dimensional space. Note

that the order of arguments p_i is irrelevant, *e.g.*, $R_3(2,2,3,2) = R_3(3,2,2,2)$. Figure 4 above illustrates this situation, with $k = 3, d = 2, p_1 = 3$ and $p_2 = p_3 = 2$; we have $R_3(3,2,2,2) = 24$ and $A_3(3,2,2,2) = 39$. The '$R(n,d)$' used in earlier analyses corresponds to $R_n(1,\cdots,1,d)$ in this notation.

3.1 Number of Regions (R_k)

For the case of regions in one-dimensional space, we define $R_k(p_1,\cdots,p_k,1)$ to be $1+\sum_{i=1}^{k} p_k$. Analysis for the case of higher dimensional spaces uses the following argument: if we previously had parallel groups of $a_1,\cdots,a_k$ hyperplanes in δ-dimensional space, adding one new hyperplane which is not parallel to any of the existing hyperplanes results in increasing the number of regions by $R_k(a_1,\cdots,a_k,\delta-1)$. Hence adding a_0 new parallel hyperplanes which are not parallel to any of the existing hyperplanes results in increasing the number of regions by $a_0.R_k(a_1,\cdots,a_k,\delta-1)$. It hence follows that:

$$R_k(p_1,p_2,\cdots,p_k,d) = R_{k-1}(p_2,\cdots,p_k,d) + p_1.R_{k-1}(p_2,\cdots,p_k,d-1). \tag{2}$$

By expanding this recurrence relation for $(k-1)$ steps, the following result is proved (in the appendix).

Theorem 2 For $d \geq 2$ and $k \leq d$, we have:

$$R_k(p_1,\cdots,p_k,d) = \prod_{i=1}^{k}(1+p_i).$$

When $k > d$, the simplification procedure in the proof of the above theorem cannot proceed in the same way for $k-1$ steps, since the right-hand-side of equation (2) contains a term which invokes a reduced dimension, and this cannot be repeated for more than $d-1$ steps. But an analogous result can still be obtained.

Theorem 3 For $d \geq 2$ and $k \geq d$, we have:

$$R_k(p_1,\cdots,p_k,d) = S_0^k + S_1^k + \cdots + S_d^k,$$

where $S_0^k = 1, S_1^k = \sum_{i=1}^{k} p_i, S_2^k = \sum_{i=1}^{k}\sum_{j=i+1}^{k} p_ip_j, S_3^k = \sum_{i=1}^{k}\sum_{j=i+1}^{k}\sum_{m=j+1}^{k} p_ip_jp_m, \cdots$, *i.e.*, $S_0^k + S_1^k + \cdots + S_k^k = \prod_{i=1}^{k}(1+p_i)$.

Corollary 1 For $d \geq 2$ and $k \geq d$, we have:

$$R_k(p, p, \cdots, p, d) = \sum_{i=0}^{d} (p^i . C_i^k).$$

3.2 Number of Hyperplane Segments (A_k)

In d-dimensional space, each segment of a hyperplane (in the first group of p_1 parallel hyperplanes) is a region in $d-1$ dimensions, formed by the intersections of other hyperplanes (in the other $(k-1)$ groups). The number of hyperplane segments in d dimensions is hence the number of regions in $d-1$ dimensions caused by intersections of other hyperplanes with each hyperplane.

Corollary 2 For $d \geq 2$ and $k \geq d$, we have:

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{d} (i . S_i^k) \tag{3}$$

From theorem 2, we also obtain the following result:

Corollary 3 For $d \geq 2$ and $k \leq d$, we have:

$$A_k(p_1, \cdots, p_k, d) = \sum_{i=1}^{k} \left[p_i . \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 + p_j) \right]. \tag{4}$$

We now address the following question: given a network with a fixed number of hidden nodes (n) that have to be partitioned into k groups of non-intersecting hyperplanes to solve a given classification task, how would the number of regions be maximized?

In the special symmetric case when $p_1 = p_2 = \cdots = p_k = n/k = p$, we have

$$A_k(n/k, \cdots, n/k, d) = n . R_{k-1}(n/k, \cdots, n/k, d-1)$$

By the corollary above, if $k \leq d$, we have

$$A_k(p, \cdots, p, d) = pk(p+1)^{k-1}.$$

Such a simplification is not possible when $k > d$, since $R_k(p, p, \cdots, p, 1) = R_1(p, 1) = (p+1)$. The case of minimally differing p_i's provides an upper bound, corresponding to the maximal value of R_k, among all possible distributions of k groups of n parallel hyperplanes.

Theorem 4 $R_k(p_1, \cdots, p_k, d)$ and $A_k(p_1, \cdots, p_k, d)$ are maximized (for a given k and $n = \sum_{i=1}^{k} p_i$) when $|p_i - p_j| \leq 1$ for every p_i, p_j, and minimized when only one of the p_i's is non-zero.

3.3 Bounds on the Ratio A_k/R_k

As a natural generalization of the analysis in section 2.3, we may study the dependence of $A_k(p_1, \cdots, p_k, d)$ on $R_k(p_1, \cdots, p_k, d)$, which is relevant when a specific problem with M clusters is being considered, where $R_k(p_1, \cdots, p_k, d) \geq M$. Starting from the expressions in theorem 3 and equation (3), we now establish bounds for the ratio $\frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)}$:

Theorem 5

$$\frac{min(k,d)}{2} \leq \frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} \leq min(k,d). \tag{5}$$

Note: When more information is available, better bounds can be obtained; e.g., theorem 1 gives a better upper bound when $p_1 = p_2 = \cdots = p_k = 1$ and $k = n$.

4 Number of Samples Required

Among the points that identify a region, the most salient for establishing classification boundaries are (naturally) those that are closest to the boundaries. These "boundary samples" provide the maximum information for classification. For our purposes, it is sufficient to work with a fuzzy definition: we consider boundary samples to be those samples with nearby points that belong to a different class.

We assume that the number of boundary samples needed for successful training is proportional to the number of hyperplane segments needed, since some boundary samples are needed to identify uniquely the two classes corresponding to the sample clusters bordering each hyperplane segment. For the case when all hyperplanes intersect each other ($k = n$), theorem 1 gives the result

$$\frac{min(n,d)}{2} \leq \frac{A(n,d)}{R(n,d)} \leq \frac{min(n,2d)}{2}.$$

The number of boundary samples required for successful classification is hence proportional to $min(n,d).R(n,d)$. When k parallel groups of hyperplanes intersect each other, theorem

5 asserts that

$$\frac{min(k,d)}{2} \leq \frac{A_k(p_1,\cdots,p_k,d)}{R_k(p_1,\cdots,p_k,d)} \leq min(k,d),$$

implying that the number of boundary samples required is proportional to $min(k,d).R(n,d)$. There is no a priori way to determine k; all we know is that $1 \leq k \leq n$; hence we only have an upper bound asserting that proportional to $min(n,d).R(n,d)$ boundary samples are required.

The above characterization is in terms of the parameter n (*i.e.*, number of hidden nodes) describing the network; it is more desirable to obtain an estimate in terms of the given problem space, assuming an optimal network is chosen. Again, if the given classification problem has M clusters of points, and the neural network is chosen to have the least possible number of hidden nodes n, then, in general, we have $R(n,d) \geq M$ but $R(n-1,d) < M$. The number of boundary samples required for successful classification is hence proportional to $min(n,d).M$. This is under the assumption that $M \geq n$; otherwise, if the number of hyperplanes $n > M$, at least d samples are required to identify each hyperplane, hence the number of boundary samples required is instead proportional to nd.

The above analysis assumes that we are trying to use a neural network with as few hidden nodes as possible, putting available resources to the best use; note that the number of connections (weights) in a network is also minimized by minimizing the number of nodes in the first hidden layer. But in some applications, reducing training time may be more important than minimizing the number of hidden nodes. If each of the M clusters is to be separately learnt (possibly in parallel) using $2d$ hidden nodes (hyperplanes describing a bounding d-dimensional hypercube) or $d+1$ hidden nodes (the smallest number of hyperplanes needed to enclose a cluster in d-dimensional space), then the network as a whole still requires $\Omega(Md)$ hidden nodes and hyperplane segments. Assuming $\Omega(d)$ boundary samples are needed to identify the hyperplane segments, the number of boundary samples required is proportional to d^2M.

This provides a pointer towards an estimate of how many input samples are required overall: in general, most input samples are **not** boundary samples. The nature of the distribution of points within clusters determines the proportion of the number of boundary samples, and hence the overall number of input samples required, assuming these are randomly drawn from the distribution. For instance, the ratio of the hypersurface ($\propto r^{d-1}$) to the hypervolume ($\propto r^d$) of a hypersphere is inversely proportional to its 'radius' r, a (1-dimensional) measure of length. Under the assumption that samples are uniformly distributed within a region, the probability of an arbitrary sample being a boundary sample is

inversely proportional to a 1-dimensional distance measure of the region. Hence the number of random input samples required for correct classification is likely to be proportional to $min(d, n)Mr$. Other distributions warrant different assumptions, so that many more input samples may be required for obtaining even a few boundary samples.

We have assumed in the above analyses that a constant number of boundary samples are required to identify each hyperplane segment. In practice, we conjecture that this number is likely to scale up with the dimension d of the input space. This means that the total number of boundary (input) samples is likely to be $\Omega(Md^2)$ rather than just $\Omega(Md)$.

We now consider a specific example, illustrated in figure 5. Let the input data consist of $M = a^d$ clusters in a d-dimensional chessboard-like pattern, where neighboring clusters belong to different classes. The number of classes itself is irrelevant, as long as it is ≥ 2. The minimum number of hyperplanes required to separate these clusters is approximately $da = dM^{1/d}$, assuming $M \gg 0$. But this many boundary samples is not sufficient, since different hyperplane segments may separate different classes. Each cluster is bounded by $2d$ hyperplane segments, but each hyperplane segment is shared by two neighboring clusters. Hence the number of hyperplane segments is approximately Md. Hence the total number of boundary samples required is proportional to Md (or Md^2 by the conjecture that d boundary samples are required to identify each hyperplane segment). A different configuration of hyperplanes may instead be used for the same problem. For instance, assuming the clusters are sufficiently well-separated, $d + 1$ hyperplanes (and hyperplane segments) can enclose each cluster completely. Since $\Omega(d)$ samples are needed to identify each bounding hyperplane, and assuming that each hyperplane is 'shared' by two clusters, the total number of boundary samples needed to enclose M clusters is now proportional to $Md(d + 1)/2$, of the same order of magnitude as Md^2. However, note that the number of hidden nodes has increased to $M(d+1)/2$ in this case, a considerable increase from $dM^{1/d}$.

5 Concluding Discussion

In this paper, we have analyzed the maximum number of regions and hyperplane segments obtained using a given number of hyperplanes. We have considered first the maximal-intersections case, assuming that all hyperplanes intersect each other. We have then generalized the results to the case when there are groups of 'parallel' hyperplanes, which reduces the number of regions and hyperplane segments obtained by intersections of hyperplanes. We have established that $\Omega(min(n, d).M)$ boundary samples are required for successful

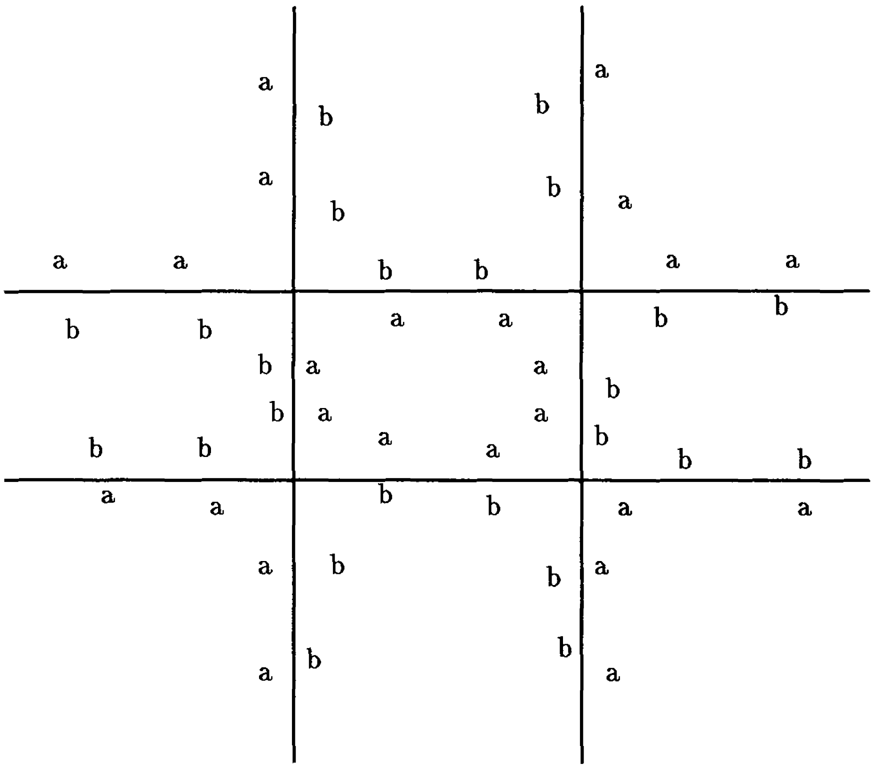

Figure 5: Two classes with chessboard-patterned clusters, in 2-dimensional space

classification, where d is the dimensionality (number of input nodes in a neural network), n is the number of hyperplanes (first layer hidden nodes in a network), and M is the number of clusters of input samples (generally $\geq$ the number of classes or output nodes). Our analysis uses the idea that some number of boundary samples are needed to identify each hyperplane segment.

In order to make effective use of the above analysis in designing neural networks for a given classification task, certain clustering procedures may first be required as a preprocessing step. Using a first estimate of the number of clusters of samples, we can obtain an estimate of the number of regions to be separated by hyperplane segments. Thus clustering is useful even in the supervised learning paradigm. Clustering also helps estimate the "radius" of each region, which is needed to estimate the number of samples needed for correct learning. A preliminary analysis of the clusters can indicate whether a sufficient number of boundary samples have been obtained, as required by our analyses.

The utility of knowing the number of required samples is obvious when we want to place high reliance on the learning accomplished by a neural network. In the interest of reducing training time, we would like to train neural nets with a limited and available number of training samples. But if too few samples have been used, the generalization capability of the neural net will be poor, and we cannot expect the net to perform well on test cases on which it has not been trained. Our results help in this regard, by indicating how many samples are required for correct classification.

References

[1] S. Ahmad and G. Tesauro, *Scaling and Generalization in Neural Networks: A case study,* Proc. Neural Info. Proc. Systems Conf. (NIPS), Vol 1, pp. 160-176, 1988.

[2] S. Amari, *On the Capacity of 3-Layer Neural Networks,* Proc. IJCNN-90, vol. 1, San Diego, June 1990.

[3] E.B. Baum, *On the Capabilities of Multilayer Perceptrons,* Journal of Complexity, vol.4, pp193-215, 1988.

[4] E.B. Baum and D. Haussler, *What Size Net Gives Valid Generalization,* Neural Computation. vol.1, pp151-160, 1988.

[5] T.M. Cover, *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition,* IEEE Trans. Electron. Comput. EC-14, pp326-334, 1965.

[6] D.B. Lenat, *On Automated Scientific Theory Formation: A Case Study using the AM Program,* Machine Intelligence 9 (ed. J.E.Hayes, D.Michie, and L.I. Mikulich), Halsted Press, pp251-286, 1977.

[7] Richard P. Lippmann, *An Introduction to Computing with Neural Nets,* IEEE ASSP Magazine, pp4-22, 1987.

[8] J. Makhoul, A. El-Jaroudi, and R. Schwartz, *Formation of Disconnected Decision Regions with a Single Hidden Layer,* IJCNN-89, 1989.

[9] G. Mirchandani and W. Cao, *On Hidden Nodes for Neural Nets,* IEEE Trans. Circuits and Systems, Vol. 36, No. 5, pp661-664, May 1989.

[10] N.J. Nilsson, Learning Machines, McGraw-Hill, 1965.

[11] N.K. Perugini and W.E. Engeler, *Neural Network Learning Time: Effects of Network and Training Set Size,* Proc. IJCNN'89, vol.II, pp395-401, 1989.

[12] D.E. Rumelhart and J.L. McClelland (Eds.) Parallel Distributed Processing, Vol. 1-2, MIT Press, 1986.

[13] P.K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations,* Pergamon Press, 1990.

6 Appendix

We now give the proofs and detailed calculations omitted from the preceding sections.

6.1 Bounds for $A(n,d)/R(n,d)$

Theorem 1:

$$\frac{min(n,d)}{2} \leq \frac{A(n,d)}{R(n,d)} \leq \frac{min(n,2d)}{2}.$$

Proof: We start from equation (1) in section 2.3:

$$\frac{A(n,d)}{R(n,d)} = \frac{n\ R(n-1,d-1)}{R(n,d)}$$

We first consider the case when $n < d$, when $R(n-1, d-1) = 2^{n-1}$, and $R(n,d) = 2^n$, since each hyperplane divides the relevant space into two regions. Hence $A(n,d)/R(n,d) = n.2^{n-1}/2^n = n/2$.

When $n \geq d$, we use the combinatory relation $C_r^{n-1} = C_r^n - C_{r-1}^{n-1}$ to expand the right-hand-side of the equation $R(n-1, d-1) = C_0^{n-1} + \cdots + C_{d-1}^{n-1}$. After regrouping terms, we have:

$$R(n-1,d-1) = [C_0^n + C_1^n + \cdots + C_{d-1}^n] - [C_0^{n-1} + C_1^{n-1} + \cdots + C_{d-2}^{n-1}].$$

Adding (and subtracting) terminal elements to the bracketed sequences, we have:

$$R(n-1,d-1) = \sum_{i=0}^{d} C_i^n - \sum_{i=0}^{d-1} C_i^{n-1} - [C_d^n - C_{d-1}^{n-1}].$$

In other words, we have $R(n-1,d-1) = R(n,d) - R(n-1,d-1) - \frac{(n-d)}{n} C_d^n$ which implies that $R(n-1,d-1) = \frac{1}{2}[R(n,d) - \frac{(n-d)}{n} C_d^n]$.

The expression for the ratio being evaluated is hence $\frac{A(n,d)}{R(n,d)} = \frac{n}{2}[1 - \frac{(n-d)C_d^n}{nR(n,d)}] \geq \frac{n}{2}[1 - \frac{(n-d)}{n}]$, since $R(n,d) = \sum_{k=0}^{d} C_k^n \geq C_d^n$. Hence we obtain the lower limit: $\frac{A(n,d)}{R(n,d)} \geq \frac{d}{2}$. An obvious upper limit of the ratio is $\frac{n}{2}$. Thus, when $d \leq n < 2d$, we have $\frac{A(n,d)}{R(n,d)} < d$.

When $n \geq 2d$, we observe that $\frac{C_d^n}{C_{d-1}^n} = \frac{n-d+1}{d}$, and further that for $d \geq i > 0$, we have $\frac{C_{i-1}^n}{C_i^n} \leq \frac{d}{n-d+1} \leq 1$. Hence $R(n,d) = C_d^n + C_{d-1}^n + \cdots + C_0^n \leq C_d^n[1 + \frac{d}{n-d+1} + (\frac{d}{n-d+1})^2 + \cdots] \leq \frac{C_d^n}{1 - \frac{d}{n-d+1}}$. Hence, substituting into the earlier expression, we have

$$\frac{A(n,d)}{R(n,d)} \leq \frac{n}{2}\left(1 - \frac{(n-d)C_d^n(1 - \frac{d}{n-d+1})}{C_d^n}\right)$$

$$= \frac{1}{2}\left[2d - 1 + \frac{n-2d+1}{n-d+1}\right].$$

Upon simplification, we finally obtain the upper limit:

$$\frac{A(n,d)}{R(n,d)} \leq d.$$

Note that when $d \leq n < 2d$, this analysis is not applicable because $\frac{C_{i-1}^n}{C_i^n}$ is not ≤ 1.

Summarizing the above discussion for the three cases of $n < d$, $d \leq n < 2d$, and $n \geq 2d$, we have:

$$\frac{min(n,d)}{2} \leq \frac{A(n,d)}{R(n,d)} \leq \frac{min(n,2d)}{2}.$$

□

6.2 Estimating R_k, A_k

Theorem 2: For $d \geq 2$ and $k \leq d$, we have:

$$R_k(p_1, \cdots, p_k, d) = \prod_{i=1}^{k}(1 + p_i).$$

Proof (by induction on k, d):
Induction Base: For $d = 2$, there is only one case to consider where $k < d$; clearly, $R_1(p_1, 2) = p_1 + 1$. In fact, we also note that $R_1(p_h, d) = p_h + 1$ for any $p_h, d \geq 1$. Also, $R_2(p_1, p_2, 2) = R_1(p_2, 2) + p_1 R_1(p_2, 1) = (p_2 + 1) + p_1(p_2 + 1) = (p_1 + 1)(p_2 + 1)$.
Induction Hypothesis: $R_h(p_1, \cdots, p_h, d-1) = \prod_{i=1}^{h}(1 + p_i)$ for every $h \leq (d-1)$; and $R_j(a_1, \cdots, a_j, d) = \prod_{i=1}^{j}(1 + a_i)$ for every $j < k$.
Induction Step:

$$R_k(p_1, p_2, \cdots, p_k, d) = \prod_{i=2}^{k}(1 + p_i) + p_1 . \prod_{i=2}^{k}(1 + p_i) = \prod_{i=1}^{k}(1 + p_i)$$

by equation (2) and the induction hypothesis.

□

When $k > d$, the simplification procedure in the above proof cannot proceed in the same way for $k - 1$ steps, since the right-hand-side of equation (2) contains a term which invokes a reduced dimension, and this cannot be repeated for more than $d - 1$ steps. But an analogous result can still be obtained.
Theorem 3: For $d \geq 2$ and $k \geq d$, we have:

$$R_k(p_1, \cdots, p_k, d) = S_0^k + S_1^k + \cdots + S_d^k,$$

where $S_0^k = 1, S_1^k = \sum_{i=1}^{k} p_i, S_2^k = \sum_{i=1}^{k} \sum_{j=i+1}^{k} p_i p_j, S_3^k = \sum_{i=1}^{k} \sum_{j=i+1}^{k} \sum_{m=j+1}^{k} p_i p_j p_m, \cdots$, *i.e.*, $S_0^k + S_1^k + \cdots + S_k^k = \prod_{i=1}^{k}(1 + p_i)$.
Proof: (by induction on d): [For readability, we sometimes omit the superscript k in S_i^k].
Induction Base: $R_k(p_1, \cdots, p_k, 2) = 1 + S_1 + S_2 = 1 + \sum_{i=1}^{k} p_i + \sum_{i=1}^{k} \sum_{j=i+1}^{k} p_i p_j$.

This is proved by induction on k. By theorem 2, we have $R_2(p_1, p_2, 2) = (1 + p_1)(1 + p_2) = 1 + (p_1 + p_2) + p_1 p_2$, conforming with the statement of the lemma. By applying equation (2) twice, we find that $R_3(p_1, p_2, p_3, 2) = 1 + \sum_{i=1}^{3} p_i + \sum_{i=1}^{3} \sum_{j=i+1}^{3} p_i p_j$. If we assume that $R_h(p_1, \cdots, p_h, 2) = 1 + \sum_{i=1}^{h} p_i + \sum_{i=1}^{h} \sum_{j=i+1}^{h} p_i p_j$ for every $h < k$, we have, by equation (2): $R_k(p_1, p_2, \cdots, p_k, 2) = R_{k-1}(p_2, \cdots, p_k, 2) + p_1 . R_{k-1}(p_2, \cdots, p_k, 1)$

$$= 1 + \sum_{i=2}^{k} p_i + \sum_{i=2}^{k} \sum_{j=i+1}^{k} p_i p_j + p_1(1 + \sum_{j=2}^{k} p_j)$$

$$= 1 + \sum_{i=1}^{k} p_i + \sum_{i=1}^{k} \sum_{j=i+1}^{k} p_i p_j.$$

□

Induction Hypothesis: For $k > h \geq d \geq 2$, assume $R_h(p_1, \cdots, p_h, d) = S_0^h + S_1^h + \cdots + S_d^h$.
Induction Step: By equation (2) and the induction hypothesis,

$$R_k(p_1, \cdots, p_k, d) = R_k(p_k, p_1, \cdots, p_{k-1}, d) = S_0^{k-1} + S_1^{k-1} + \cdots + S_d^{k-1} + p_k(S_0^{k-1} + S_1^{k-1} + \cdots + S_{d-1}^{k-1})$$

$$= S_0 + (S_1^{k-1} + p_k S_0^{k-1}) + \cdots + (S_d^{k-1} + p_k S_{d-1}^{k-1}).$$

But we observe that $S_l^{k-1} + p_k S_{l-1}^{k-1} = S_l^k$, for each $l \leq d$. Hence

$$R_k(p_1, \cdots, p_k, d) = S_0^k + S_1^k + \cdots + S_d^k,$$

□

Corollary 2: For $d \geq 2$ and $k \geq d$, we have:

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{d} (i.S_i^k) \tag{6}$$

Proof: Since the number of hyperplane segments in d dimensions corresponds to the number of regions in $d-1$ dimensions caused by intersections of other hyperplanes with each hyperplane, we obtain:

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{k} p_i . R_{k-1}(p_1, \cdots, p_{i-1}, p_{i+1}, \cdots, p_k, d-1). \tag{7}$$

Using theorem 3, we have:

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{k} p_i . [S_0^{k\backslash i} + S_1^{k\backslash i} + \cdots + S_{d-1}^{k\backslash i}], \tag{8}$$

where $S_l^{k\backslash i}$ denotes S_l^k with p_i changed to 0, *i.e.*, the l^{th} term in the expansion of $R_k(p_1, \cdots, p_{i-1}, 0, p_{i+1}, \cdots, p_k, d)$ from theorem 1. We find by expansion that for $0 \leq l \leq d$,

$$\sum_{i=1}^{k} p_i . S_l^{k\backslash i} = (l+1) S_{l+1}^k.$$

From this, it follows that:

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{d} (i.S_i^k) \tag{9}$$

□

6.3 Maximizing A_k, R_k

Theorem 4: $R_k(p_1, \cdots, p_k, d)$ and $A_k(p_1, \cdots, p_k, d)$ are maximized (for a given k and $n = \sum_{i=1}^{k} p_i$) when $|p_i - p_j| \leq 1$ for every p_i, p_j, and minimized when only one of the p_i's is non-zero.

Proof: Let α be $R_k(p_1, p_2, \cdots, p_k, d)$, and β be $R_k(p_1 + 1, p_2 - 1, \cdots, p_k, d)$, where the latter corresponds to switching one hyperplane from the second group to the first. Upon expansion and simplification using equation (2) twice, we find that:

$$(\alpha - \beta) = [R_{k-1}(p_2, \cdots, p_k, d) + p_1.R_{k-1}(p_2, \cdots, p_k, d-1)]$$

$$-[R_{k-1}(p_2 - 1, \cdots, p_k, d) + (p_1 + 1).R_{k-1}(p_2 - 1, \cdots, p_k, d-1)]$$

$$= [R_{k-2}(p_3, \cdots, p_k, d) + p_2.R_{k-2}(p_3, \cdots, p_k, d-1) + p_1.R_{k-2}(p_3, \cdots, p_k, d-1)$$

$$+p_1p_2.R_{k-2}(p_3, \cdots, p_k, d-2)] - [R_{k-2}(p_3, \cdots, p_k, d) + (p_2 - 1).R_{k-2}(p_3, \cdots, p_k, d-1)$$
$$+(p_1 + 1).R_{k-2}(p_3, \cdots, p_k, d-1) + (p_1 + 1)(p_2 - 1).R_{k-2}(p_3, \cdots, p_k, d-2)]$$

This simplifies to:

$$(\alpha - \beta) = (p_1 + 1 - p_2)R_{k-2}(p_3, \cdots, p_k, d-2). \tag{10}$$

If $p_1 > (p_2 - 1)$, the change from α to β results in a lower value; if $p_1 = (p_2 - 1)$ then the change is irrelevant, and simply corresponds to shuffling around the order of the groups that are arguments of R_k. But if $p_1 < (p_2 - 1)$, then the change results in a higher value for R_k. Carrying the argument to its logical conclusion, the maximal R_k-value is obtained by repeatedly making the arguments of R_k 'as equal as possible'. A similar argument shows that A_k is also maximized in the same case; since

$$A_k(p_1, p_2, \cdots, p_k, d) = \sum_{i=1}^{k} p_i.R_{k-1}(p_1, \cdots, p_{i-1}, p_{i+1}, \cdots, p_k, d-1).$$

The expression $A_k(p_1, \cdots, p_k, d) - A_k(p_1 + 1, p_2 - 1, p_3, \cdots, p_k, d)$ simplifies to:

$$\sum_{i=3}^{k} p_i.[R_{k-1}(p_1, \cdots, p_{i-1}, p_{i+1}, \cdots, p_k, d-1) - R_{k-1}(p_1+1, p_2-1, p_3, \cdots, p_{i-1}, p_{i+1}, \cdots, p_k, d-1)]$$

$$+R_{k-2}(p_3, \cdots, p_k, d-1).[p_1 + p_2 - (p_1 + 1) - (p_2 - 1)]$$
$$+R_{k-2}(p_3, \cdots, p_k, d-2).[p_1p_2 + p_1p_2 - (p_1 + 1)(p_2 - 1) - (p_1 + 1)(p_2 - 1)].$$

The first summation is positive iff $p_1 > p_2 - 1$ (by the above analysis maximizing R_k), the second term vanishes (zero coefficient), and the last term simplifies to $2R_{k-2}(p_3, \cdots, p_k, d-$

2).$[p_1 + 1 - p_2]$, which is also positive iff $p_1 > p_2 - 1$. In other words, A_k is also maximized when its arguments are 'as equal as possible'.

Conversely, the above argument also shows that R_k (and A_k) is minimized in the case when all hyperplanes are mutually parallel (non-intersecting). We then have $A_k(n, 0, \cdots, 0, d) = n$ and $R_k(n, 0, \cdots, 0, d) = n + 1$.

□

6.4 Bounds on A_k/R_k

Theorem 5:

$$min(k, d)/2 \leq \frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} \leq min(k, d). \tag{11}$$

Proof: We show that the ratio lies between $\frac{k}{2}$ and k when $k \leq d$, and lies between $\frac{d}{2}$ and d when $k \geq d$.

For $d \geq 2$ and $k \geq 2$,

$$\frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} = d\left[\frac{S_1^k + 2S_2^k + \cdots + dS_d^k}{d + dS_1^k + \cdots + dS_d^k}\right] \leq d. \tag{12}$$

Interestingly, this upper bound is the same as that obtained earlier for the simpler case when all the hyperplanes intersect each other.

Lower Bound: [Note: Superscript k is omitted from S_i^k terms below.] Assuming d is even and $k \geq d \geq 2$, we have;

$$\frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} = \frac{d}{2}\left[\frac{1 + S_1 + S_2 + \cdots + S_d}{1 + S_1 + S_2 + \cdots + S_d}\right]$$

$$- \left[\frac{\frac{d}{2} + (\frac{d}{2} - 1)S_1 + \cdots + 1.S_{\frac{d}{2}-1} - \left(1.S_{\frac{d}{2}+1} + \cdots + (\frac{d}{2} - 1)S_{d-1} + \frac{d}{2}S_d\right)}{1 + S_1 + S_2 + \cdots + S_d}\right].$$

$$\text{Hence}: \frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} \geq \frac{d}{2} + \frac{\theta}{1 + S_1 + S_2 + \cdots + S_d} \tag{13}$$

where $\theta = \frac{d}{2}(S_d - 1) + (\frac{d}{2} - 1)(S_{d-1} - S_1) + \cdots + 1.(S_{\frac{d}{2}+1} - S_{\frac{d}{2}-1})$

We show that θ is positive, by showing that each term $S_{d-i} - S_i$ is positive, when $d - i \geq i$, *i.e.*, $\frac{d}{2} > i$.

For this analysis, we treat each of the summed terms ($\prod_i p_i$) in each S_j as a **string** (an ordered sequence) of p_i's. Let M_j be the set of strings $= \{\tau | \tau$ is a permutation of some string in $S_j\}$. Similarly, let M_{j-1} be the set of permutations of strings in S_{j-1}. For each

string $q_1 \cdots q_{j-1}$ in M_{j-1}, we observe that M_j contains precisely $k-(j-1)$ strings of the form $q_1 \cdots q_{j-1} q_j$, where each q_i is a distinct member of $\{p_1, \cdots p_k\}$ (since q_j can be chosen to be any of $p_1, \cdots p_k$ that do not occur in $q_1, \cdots, q_{j-1}$).

Since each $q_i \geq 1$, the product $q_1 \cdots q_{j-1} q_j \geq q_1 \cdots q_{j-1}$. Hence

$$\left[\sum_{q_1 \cdots q_j \in M_j} q_1 \cdots q_j \right] \geq \left[(k-j+1) \sum_{q_1 \cdots q_{j-1} \in M_{j-1}} q_1 \cdots q_{j-1}. \right]$$

But

$$\left[\sum_{q_1 \cdots q_j \in M_j} q_1 \cdots q_j \right] = (j!) S_j, \text{ and } \left[\sum_{q_1 \cdots q_{j-1} \in M_{j-1}} q_1 \cdots q_{j-1} \right] = (j-1)! S_{j-1},$$

since each M_i contains $i!$ permutations of each term in S_i. Hence

$$(j!) S_j \geq (k-j+1)[(j-1)!] S_{j-1},$$

$$i.e., \quad \frac{S_j}{S_{j-1}} \geq \frac{(k-j+1)}{j} = \frac{C_j^k}{C_{j-1}^k}.$$

Since j above is arbitrary, we have:

$$\frac{S_j}{S_i} = \frac{S_j}{S_{j-1}} \cdot \frac{S_{j-1}}{S_{j-2}} \cdot \cdots \cdot \frac{S_{i+1}}{S_i} \geq \frac{C_j^k}{C_{j-1}^k} \cdot \frac{C_{j-1}^k}{C_{j-2}^k} \cdot \cdots \cdot \frac{C_{i+1}^k}{C_i^k} = \frac{C_j^k}{C_i^k}.$$

In terms which we are interested in (to show that θ is positive), we have:

$$\frac{S_{d-i}}{S_i} \geq \frac{C_{d-i}^k}{C_i^k}.$$

By the nature of the binomial distribution, it is known that $C_j^k \geq C_i^k$ whenever $k-i \geq j \geq i$. Hence $C_{d-i}^k \geq C_i^k$ if $k-i \geq d-i \geq i$. Since these conditions are implied by $k \geq d$ and $d-i \geq i$, we have the desired result:

$$S_{d-i} \geq S_i,$$

which implies that θ is positive, hence

$$\frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} \geq \frac{d}{2}.$$

This gives us the desired lower bound $d/2$ for A_k/R_k. The assumption that d is even is just a matter of convenience of explanation, and can be trivially relaxed. In the above analysis, we assumed that $k \geq d$. The contrary case is separately analyzed below.

We now consider the case when $k \leq d$. By theorem 3 and equation (4), we have

$$\frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} = \frac{\sum_{i=1}^{k} [p_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 + p_j)]}{\prod_{i=1}^{k} (1 + p_i)} = \sum_{i=1}^{k} \frac{p_i}{1 + p_i}.$$

Since each $p_i \geq 1$, we have $1/2 \leq p_i/(1 + p_i) < 1$ for each p_i. Hence:

$$\frac{k}{2} \leq \frac{A_k(p_1, \cdots, p_k, d)}{R_k(p_1, \cdots, p_k, d)} < k \leq d.$$

□