Finding Relevant Subspaces in Neural Network Learning

Avrim L. Blum* and **Ravi Kannan**
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891

Introduction

Consider a layered neural network with many inputs but only a small number of nodes k in its hidden layer (the layer immediately above the inputs if there is more than one). In such a network, hidden node i computes the dot product of an example $\mathbf{x}$ with a vector of weights $\mathbf{w}_i$ and then applies some function such as a sigmoid or threshold to that quantity; the final result gets sent on to the layer above. An implication of this structure is that the output of the network on some example $\mathbf{x}$ depends only on the values $\mathbf{x} \cdot \mathbf{w}_1, \ldots, \mathbf{x} \cdot \mathbf{w}_k$, and not on any other information about $\mathbf{x}$. One way of looking at this fact is that even though the examples may lie in a high dimensional input space, there exists some low-dimensional "relevant subspace", namely the span of the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$, such that the network output depends only on the projection of examples into that space.

Algorithms such as Backpropagation, when applied to a network of the form described above, work by directly searching for good vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$. We consider here a somewhat different approach. Suppose the function we are trying to learn can be represented by a layered neural network with only a small number of hidden nodes. (The fact that neural networks of this form have been quite successful in many situations implies that at least many interesting functions can be approximated well by such a representation.) In that case, we know there exists (somewhere) a relevant subspace of low dimension as described above. So, instead of searching for the exact vectors $\mathbf{w}_i$ right away, why not relax our goal and just look for some set of vectors whose span is (approximately) the relevant space: even a slightly larger space would do. This might be an easier task than finding the weight vectors. And, if we could accomplish it, we would be able to significantly simplify the learning problem since we could then project all our examples onto this low-dimensional space and learn inside that space.

*This material is based upon work supported under NSF National Young Investigator grant CCR-9357793 and a National Science Foundation postdoctoral fellowship. Authors' email addresses: {avrim,kannan}@cs.cmu.edu

A common technique for searching for relevant subspaces is that of "principal component analysis". Our main result is a proof that under an interesting set of conditions, a variation on this approach will find at least one vector (nearly) in the space we are looking for. A recursive approach can then be used to get additional such vectors. All proofs of theorems described here appear in [Blum and Kannan, 1993].

Assumptions and Definitions

In principal component analysis [Morrison, 1990], given a sample of unlabeled data, one computes the directions along which the variance of the data is maximized. This computation can be done fairly quickly by finding the eigenvectors of largest eigenvalue of a matrix defined by the examples. The usual reason one wants these directions is that if the distribution of examples is some sort of "pancake", we want to find the main axes of this pancake.

In our work, we consider the following two assumptions.

1. We assume examples are selected uniformly at random in the n-dimensional unit ball B_n. While this condition will not hold in practice, this kind of uniform distribution seems a reasonable one to examine because it is "unbiased". Performing any kind of clustering or principal-component-analysis on the *unlabeled* examples will provide no useful information.
2. We assume that examples are classified as either positive or negative by a simple form of two-layer neural network. This network has k nodes in its hidden layer, each one using a strict threshold instead of a sigmoid, and (the main restriction) an output unit that computes the AND function of its inputs. In geometrical terms, this means that the region of positive examples can be described as an intersection of k halfspaces.

Let $\mathbf{w}_1 \cdot \mathbf{x} \leq a_1, \ldots, \mathbf{w}_k \cdot \mathbf{x} \leq a_k$ be the k halfspaces from assumption (2). Let P be the intersection of the unit ball with the positive region of the target concept; i.e., $P = \{\mathbf{x} \in B_n : \mathbf{w}_i \cdot \mathbf{x} \leq a_i, \text{ for } i = 1, 2, \ldots, k\}$.

Assume that all the halfspaces defining the target concept are non-redundant in the sense that their removal would enlarge P (otherwise, we could think of the target function as an even simpler network). Then, formally, the *relevant subspace of* P, denoted $\mathbf{V}_{\text{rel}}(P)$, is the span of the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$. The *irrelevant subspace of* P, denoted $\mathbf{V}_{\text{irrel}}(P)$, is the collection of all vectors orthogonal to $\mathbf{V}_{\text{rel}}(P)$. Vectors in $\mathbf{V}_{\text{rel}}(P)$ are called *relevant* vectors. A *direction* is a unit vector.

Results

It is not hard to see that given the assumptions (1) and (2) above, if we draw a large sample and find that the *mean* or *center of gravity* of the positive examples is *far* from the center of gravity of the negative examples, then this difference gives us a vector (nearly) in $\mathbf{V}_{\text{rel}}(P)$. (There will be some error due to sampling.) The reason this vector will approximately lie in the relevant space is that P is symmetric about reflection through the hyperplane normal to any vector $\mathbf{v} \in \mathbf{V}_{\text{irrel}}(P)$. In other words, for any positive example $\mathbf{x} = \mathbf{x}' + \mathbf{x}_{irrel}$ where $\mathbf{x}_{irrel}$ is the component of $\mathbf{x}$ in $\mathbf{V}_{\text{irrel}}(P)$, the example $\mathbf{x}' - \mathbf{x}_{irrel}$ is also positive. Our main result is that if on the other hand these centers of gravity are close, then the direction that minimizes the *second moment* of the positive examples, or equivalently maximizes the second moment of the negative examples, is also guaranteed to be (nearly) in the relevant space. This direction will not necessarily be any of the vectors $\mathbf{w}_i$ but rather may be some linear combination of them.

Formally, we prove the following theorem. Let p denote the probability measure of P (the probability of drawing a positive example). Say that a point x is *τ-central* to a region R if a ball of radius τ about x is contained in R.

Theorem 1 *Let τ be such that the origin is τ-central in P. Let $\mathbf{v}$ be the unit vector such that $\mathbf{E}_{\mathbf{x}\in P}[(\mathbf{v}\cdot\mathbf{x})^2]$ is minimized, and $\mathbf{w}$ any other unit vector with $t = ||\text{proj}(\mathbf{w}, \mathbf{V}_{\text{irrel}}(P))||$. Then:*

1. $\mathbf{v} \in \mathbf{V}_{\text{rel}}(P)$.
2. $\mathbf{E}_P[(\mathbf{w}\cdot\mathbf{x})^2] \geq \mathbf{E}_P[(\mathbf{v}\cdot\mathbf{x})^2] + \frac{t^2 n\Delta}{k(n+2)}$.

where $\Delta = p(\frac{\tau}{8n^2})$ if $p \leq \frac{1}{2}$, and $\Delta = \frac{(1-p)^2}{p}(\frac{\tau}{8n^2})$ if $p > \frac{1}{2}$.

Though Theorem 1 is a bit complicated to state, the key point is the following. If the irrelevant portion t of direction w is large, then the second moment in direction $\mathbf{w}$ will be noticeably larger than the minimum second moment, so long as τ is not too small and p is not too close to 0 or 1. It is not hard to show that if the mean of the positive examples is close to the mean of the negative examples (the case where we would need to apply this theorem) then τ will not be too small. Also, if p is close to either 0 or 1, then it does not bother us that we cannot find the relevant directions since in that case either almost all examples are positive or almost all examples are negative.

We conjecture that a theorem of this form extends to the rest of the relevant subspace. In particular, we conjecture that the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ where $\mathbf{v}_1$ is the center of gravity of P and $\mathbf{v}_i$ is the unit vector among those orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$ that minimizes $\mathbf{E}_{\mathbf{x}\in P}[(\mathbf{v}_i\cdot\mathbf{x})^2]$, will span $\mathbf{V}_{\text{rel}}(P)$. We also conjecture that vectors with a "noticeable" component in an irrelevant direction will have noticeably higher second moment. If these conjectures are true, this would result in a simple polynomial time algorithm for approximating the relevant subspace under assumptions (1) and (2). In addition, it would give a simple polynomial-time *learning* algorithm when k is a constant.

Although we cannot prove the above conjectures, using Theorem 1 we *are* able to prove that a more complicated and less efficient strategy will learn in polynomial time in a probably-approximately-correct sense under assumptions (1) and (2) when k is constant, extending results of [Baum, 1990]. The basic idea of the strategy is this: After finding the first relevant direction as described above, look at "slices" perpendicular to that direction, and recursively find a good hypothesis for each slice. The idea here is that sufficiently thin slices can roughly be treated as $(n-1)$-dimensional balls, with a $(k-1)$-dimensional relevant subspace. In part due to the errors introduced in this approximation, the end hypothesis produced by our algorithm will not be an intersection of halfspaces, but rather a polynomial-time prediction algorithm representing a union of hypotheses for each slice. The running time and number of examples needed by this procedure are, unfortunately, doubly exponential in k.

Remarks

Examples in real learning scenarios are, of course, unlikely to be "selected at random from the unit ball". Labelings are also unlikely to be completely consistent with an intersection of a small number of halfspaces. Nonetheless, we believe that our theorems regarding these conditions shed some light on why examining principal components can be a useful tool when neural networks are being considered. We also believe our techniques should be extendible to cover more general radially-symmetric distributions.

References

E. B. Baum. Polynomial time algorithms for learning neural nets. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 258–272. Morgan Kaufmann, 1990.

A. Blum and R. Kannan. Learning an intersection of k halfspaces over a uniform distribution. In *Proceedings of the 34^{th} Annual IEEE Symposium on Foundations of Computer Science*, pages 312–320, November 1993.

D. F. Morrison. *Multivariate Statistical Methods.* McGraw-Hill, 3rd edition, 1990.