Communicated by Steve Suddarth

Hints and the VC Dimension

Yaser S. Abu-Mostafa
California Institute of Technology, Pasadena, CA 91125 USA

Learning from hints is a generalization of learning from examples that allows for a variety of information about the unknown function to be used in the learning process. In this paper, we use the VC dimension, an established tool for analyzing learning from examples, to analyze learning from hints. In particular, we show how the VC dimension is affected by the introduction of a hint. We also derive a new quantity that defines a VC dimension for the hint itself. This quantity is used to estimate the number of examples needed to "absorb" the hint. We carry out the analysis for two types of hints, invariances and catalysts. We also describe how the same method can be applied to other types of hints.

1 Introduction

Learning from examples deals with an unknown function f that is represented by examples to the learning process. The process uses the examples to infer an approximate implementation of f. Learning from hints (Abu-Mostafa 1990) generalizes the situation by allowing other information that we may know about f to be used in the learning process. Such information may include invariance properties, symmetries, correlated functions (Suddarth and Holden 1991), explicit rules (Omlin and Giles 1992), minimum-distance properties (Al-Mashouq and Reed 1991), or any other fact about f that narrows down the search. In many practical situations, we do have some prior information about f, and the proper use of this information (instead of just using "blind" examples of f) can make the difference between feasible and prohibitive learning.

In this paper, we develop a theoretical analysis of learning from hints. The analysis is based on the VC dimension (Blumer *et al.* 1989), which is an established tool for analyzing learning from examples. Simply stated, the VC dimension $VC(G)$ furnishes an upper bound for the number of examples needed by a learning process that starts with a set of hypotheses G about what f may be. The examples guide the search for a hypothesis $g \in G$ that is a good replica of f.

Since f is unknown to begin with, we start with a relatively big set of hypotheses G to maximize our chances of finding a good approximation of f among them. However, the larger G is, the more examples of f we need to pinpoint the good hypothesis. This is reflected in a bigger value of $VC(G)$. How do we make G smaller without the risk of losing good approximations of f? This is where the hints come in. Since a hint is a *known* property of f, we can use it as a litmus test to weed out bad gs thus shrinking G without losing good hypotheses. The main result of this paper is the application of the VC dimension to hints in two forms.

1. The VC dimension provides an estimate for the number of examples needed to learn f. When a hint H is given about f, the number of examples of f can be reduced. This is reflected in a smaller "VC dimension given the hint" $VC(G \mid H)$.
2. If H itself is represented to the learning process by a set of examples, we would like to estimate how many examples are needed to absorb the hint. This calls for a generalization of the VC dimension to cover examples of the hint as well as examples of the function, which is reflected in a "VC dimension for the hint" $VC(G;H)$.

We will study two types of hints in particular, invariances and catalysts. We will discuss how the same framework can be used to study other types of hints.

A detailed account of the VC dimension can be found in (Blumer *et al.* (1989) and Vapnik and Chervonenkis (1971). We will provide a brief background here to make the paper self-contained. The setup for learning from examples consists of an *environment* X and an unknown function $f : X \rightarrow \{0,1\}$ that we wish to learn. The goal is to produce a *hypothesis* $g : X \rightarrow \{0,1\}$ that approximates f. To do this, the learning process starts with a set of hypotheses G and tries to select a good $g \in G$ based on a number of examples $[x_1, f(x_1)]$; ... ; $[x_N, f(x_N)]$ of f. To generate the examples, we assume that there is a probability distribution $P(x)$ on the environment X. Each example is picked independently according to $P(x)$. The hypothesis g that results from the learning process is considered a good approximation of f if the probability [w.r.t. $P(x)$] that $g(x) \neq f(x)$ is small. The learning process should have a high probability of producing a good approximation of f when a sufficient number of examples is provided. The VC dimension helps determine what is "sufficient."

Here is how it works. Let $\pi_g = \Pr[g(x) = f(x)]$, where $\Pr[\cdot]$ denotes the probability of an event. We wish to pick a hypothesis g that has $\pi_g \approx 1$. However, f is unknown and thus we do not know the values of these probabilities. Since f is represented by examples, we can compute the frequency of agreement between each g and f *on the examples* and base our choice of g on the frequencies instead of the actual probabilities. Let hypothesis g agree with f on a fraction ν_g of the examples. We pick a hypothesis that has $\nu_g \approx 1$. The VC inequality asserts that the values of ν_gs will be close to π_gs. Specifically,

$$\Pr\left[\sup_{g \in G} |\nu_g - \pi_g| > \epsilon\right] \leq 4m(2N)e^{-\epsilon^2 N/8}$$

 © 1993 Massachusetts Institute of Technology

where "sup" denotes the supremum, and m is the *growth function* of G. $m(N)$ is the maximum number of different binary vectors $g(x_1)\cdots g(x_N)$ that can be generated by varying g over G while keeping $x_1, \ldots, x_N \in X$ fixed. Clearly, $m(N) \leq 2^N$ for all N. The VC dimension $VC(G)$ is defined as the smallest N for which $m(N) < 2^N$. We assume that G has a finite VC dimension. If $VC(G) = d$, the growth function $m(N)$ can be bounded by

$$m(N) < \sum_{i=0}^{d} \binom{N}{i} \leq N^d + 1$$

When this estimate is substituted in the VC inequality, the right-hand side of the inequality becomes arbitrarily small for sufficiently large N. This means that it is almost certain that each ν_g is approximately the same as the corresponding π_g. This is the rationale for considering N examples sufficient to learn f. We can afford to base our choice of hypothesis on ν_g as calculated from the examples, because it is approximately the same as π_g. How large N needs to be to achieve a certain degree of approximation is affected by the value of the VC dimension.

In this paper, we assume that $f \in G$. This means that G is powerful enough to implement f. We also assume that f strictly satisfies the hint H. This means that f will not be excluded as a result of taking H into consideration. Finally, we assume that everything that needs to be measurable will be measurable.

2 Invariance Hints

It is often the case that we know an invariance property of an otherwise unknown function. For example, speaker identification based on a speech waveform is invariant under time shift of the waveform. Properties such as shift invariance and scale invariance are commonplace in pattern recognition, and dozens of methods have been developed to take advantage of them (e.g., Hu 1969). Invariances have also been used in neural networks, for example, group invariance of functions (Minsky and Papert 1988) and the use of invariances in backpropagation (Abu-Mostafa 1990).

An invariance hint H can be formalized by the partition

$$X = \bigcup_{\alpha} X_\alpha$$

of the environment X into the invariance classes X_α, where α is an index. Within each class X_α, the value of f is constant. In other words, $x, x' \in X_\alpha$ implies that $f(x) = f(x')$.

Some invariance hints are "strong" and others are "weak," and this is reflected in the partition $X = \bigcup_\alpha X_\alpha$. The finer the partition, the weaker the hint. For instance, if each X_α contains a single point, the hint is extremely weak (actually useless) since the information that $x, x' \in X_\alpha$ implies that $f(x) = f(x')$ tells us nothing new as x and x' are the same point in this case. On the other extreme, if there is a single X_α that contains all the points ($X_\alpha = X$), the hint is extremely strong as it forces f to be constant over X (either $f = 1$ or $f = 0$). Practical hints, such as scale invariance and shift invariance, lie between these two extremes.

In what follows, we will apply the VC dimension to an invariance hint H. We will start by assessing the impact of H on the original VC dimension. We will then focus on representing H by examples and address what an example of H is, how to define a VC dimension for H, and what it means to approximate H. Finally, we will discuss relations between different VC dimensions.

2.1 How the Hint Affects $VC(G)$. The VC dimension is used to estimate the number of examples needed to learn an unknown function f. It is intuitive that, with the benefit of a hint about f, we should need fewer examples. To formalize this intuition, let the invariance hint H be given by the partition $X = \bigcup_\alpha X_\alpha$. Each hypothesis $g \in G$ either satisfies H or else does not satisfy it. Satisfying H means that whenever $x, x' \in X_\alpha$, then $g(x) = g(x')$. The set of hypotheses that satisfy H is $\hat{G}$

$$\hat{G} = \{g \in G \mid x, x' \in X_\alpha \Rightarrow g(x) = g(x')\}$$

$\hat{G}$ is a set of hypotheses and, as such, has a VC dimension of its own. This is the basis for defining the VC dimension of G given H

$$VC(G \mid H) = VC(\hat{G})$$

Since $\hat{G} \subseteq G$, it follows that $VC(G \mid H) \leq VC(G)$. Nontrivial hints lead to a significant reduction from G to $\hat{G}$, resulting in $VC(G \mid H) < VC(G)$. On the other hand, some hints may have $VC(G \mid H) = VC(G)$. For instance, in the case of the weak hint we talked about, every g trivially satisfies the hint, hence $\hat{G} = G$.

$VC(G \mid H)$ replaces $VC(G)$ following the "absorption" of the hint. Without the hint, $VC(G)$ provides an estimate for the number of examples needed to learn f. With the hint, $VC(G \mid H)$ provides a new estimate for the number of examples. This estimate is valid regardless of the mechanism for absorbing the hint, as long as it is completely absorbed. If, however, the hint is only *partially* absorbed (which means that some gs that do not strictly satisfy the invariance are still allowed), the effective VC dimension lies between $VC(G)$ and $VC(G \mid H)$.

2.2 Representing the Hint by Examples. What is an example of an invariance hint? If we take the hint specified by $X = \bigcup_\alpha X_\alpha$, an example would be "$f(x) = f(x')$," where x and x' belong to the same invariance class. In other words, an example is a pair (x, x') that belong to the same X_α.

The motivation for representing a hint by examples is twofold. The hint needs to be incorporated in what is already a learning-from-examples process. The example $f(x) = f(x')$ can be directly included in descent methods such as backpropagation along with examples of the function itself. To do this, the quantity $[g(x) - g(x')]^2$ is minimized the same way $[g(x) - f(x)]^2$ is minimized when we use an example of f. In addition, we may represent a hint by examples if it cannot be easily expressed as a global mathematical constraint. For instance, invariance under elastic deformation of images does not readily yield an obvious constraint on the weights of a feedforward network.

In contrast to the function f that is represented by a controlled number of examples and is otherwise unknown, a hint can be represented by as many examples as we wish, since it is a known property and hence can be used indefinitely to generate examples.

Examples of the hint, like examples of the function, are generated according to a probability distribution. One way to generate (x, x') is to pick x from X according to the probability distribution $P(x)$, then pick x' from X_α (the invariance class that contains x) according to the conditional probability distribution $P(x' \mid X_\alpha)$. A sequence of N (pairs of) examples $(x_1, x_1'); (x_2, x_2'); \ldots; (x_N, x_N')$ would be generated in the same way, independently from pair to pair.

2.3 A VC Dimension for the Hint. As we discussed in the introduction, the VC inequality is used to estimate how well f is learned. We wish to use the same inequality to estimate how well H is absorbed. To do this, we transform the situation from hints to functions. This calls for definitions of new **X**, **P**, **G**, and **f**.

Let H be the invariance hint $X = \bigcup_\alpha X_\alpha$. The new environment is defined by

$$\mathbf{X} = \bigcup_\alpha X_\alpha^2$$

(pairs of points coming from the same invariance class) with the probability distribution described above

$$\mathbf{P}(x, x') = P(x)P(x' \mid X_\alpha)$$

where X_α is the class that contains x (hence contains x'). The new set of hypotheses **G**, defined on the environment **X**, contains a hypothesis **g** for every hypothesis $g \in G$ such that

$$\mathbf{g}(x, x') = \begin{cases} 1 & g(x) = g(x') \\ 0 & g(x) \neq g(x') \end{cases}$$

and the function to be "learned" is

$$\mathbf{f}(x, x') = 1$$

The VC dimension of the set of hypotheses **G** is the basis for defining a VC dimension for the hint.

$$VC(G;H) = VC(\mathbf{G})$$

$VC(G;H)$ depends on both G and H since **G** is based on G and the new environment **X** (which in turn depends on H).

2.4 Approximation of the Hint. If the above learning process resulted in the hypothesis $\mathbf{g} = \mathbf{f}$ (the constant 1), the corresponding $g \in G$ would obviously satisfy the hint. Learning from examples, however, results only in **g** that *approximates* **f** well (with high probability). The approximation is in terms of the distribution $\mathbf{P}(x, x')$ used to generate the examples. Thus, w.r.t. to **P**, $\Pr[\mathbf{g}(x, x') \neq 1] \to 0$ as the number of examples N becomes large. Can we translate this statement into a similar one based only on the original distribution $P(x)$? To do this, we need to rid the statement of x'. Let

$$\Pr[g(x) \neq g(x')] = \gamma$$

By definition of **g**, $\Pr[\mathbf{g}(x, x') \neq 1]$ is the same as $\Pr[g(x) \neq g(x')]$. This implies that $\gamma \to 0$ as $N \to \infty$. In words, if we pick x and x' at random according to $\mathbf{P}(x, x')$, the probability that our hypothesis will have different values on these two points is small.

To get rid of x' from this statement, we introduce hint-satisfying versions of the gs. For each $g \in G$, let $\hat{g}$ be the best approximation of g that strictly satisfies the hint. This means that, within each invariance class X_α, $\hat{g}(x)$ is constant and its value is the more probable of the two values of $g(x)$ within X_α (ties are broken either way). We will argue that

$$\Pr[g(x) \neq \hat{g}(x)] \leq \gamma$$

Since $\gamma \to 0$, this statement [which is solely based on $P(x)$] implies that "g approximately satisfies the hint" in a more natural way.

Here is the argument. Let $\hat{\gamma}$ be the probability that $g(x) \neq \hat{g}(x)$. Given X_α, let $\hat{\gamma}_\alpha$ be the conditional probability that $g(x) \neq \hat{g}(x)$, and let γ_α be the conditional probability that $g(x) \neq g(x')$. From the definition of $\hat{g}$, $\hat{\gamma}_\alpha$ must be $\leq \frac{1}{2}$ (otherwise, the value of $\hat{g}$ in X_α should be flipped). Within each X_α, since $\hat{g}$ is constant, $g(x) \neq g(x')$ if, and only if, g agrees with $\hat{g}$ on either x or x' and disagrees on the other. This means that

$$\begin{aligned} \gamma_\alpha &= 2\hat{\gamma}_\alpha(1 - \hat{\gamma}_\alpha) \\ &\geq \hat{\gamma}_\alpha \end{aligned}$$

(since $1 - \hat{\gamma}_\alpha \geq \frac{1}{2}$). This is true for every class X_α. Averaging over α, we get $\gamma \geq \hat{\gamma}$, hence

$$\begin{aligned} \Pr[g(x) \neq \hat{g}(x)] &= \hat{\gamma} \\ &\leq \gamma \\ &\to 0 \end{aligned}$$

This establishes the more natural notion of approximating the hint.

2.5 A Bound on $VC(G;H)$. As in the case of the set G and its growth function $m(N)$, the VC dimension $VC(G;H) = VC(\mathbf{G})$ is defined based on the growth function $\mathbf{m}(N)$ of the set $\mathbf{G}$. $\mathbf{m}(N)$ is the maximum number of patterns of 1s and 0s that can be obtained by applying the $\mathbf{g}$'s to (fixed but arbitrary) N examples $(x_1, x_1'); (x_2, x_2'); \ldots; (x_N, x_N')$. $VC(G;H)$ is the smallest N for which $\mathbf{m}(N) < 2^N$.

The value of $VC(G;H)$ will differ from hint to hint. Consider our two extreme examples of weak and strong hints. The weak hint has $VC(G;H)$ as small as 1 since each g always agrees with each example of the hint [hence every $\mathbf{g}$ is the constant 1, and $\mathbf{m}(N) = 1$ for all N]. The strong hint has $VC(G;H)$ as large as it can be. How large is that? In Fyfe (1992), it is shown that for any invariance hint H,

$$VC(G;H) < \lambda\, VC(G)$$

where $\lambda \approx 4.54$. The argument goes as follows. For each pattern generated by the g's on

$$x_1, x_1', x_2, x_2', \ldots, x_N, x_N'$$

there is at most one distinct pattern generated by the $\mathbf{g}$'s on

$$(x_1, x_1'); (x_2, x_2'); \ldots; (x_N, x_N')$$

because $\mathbf{g}(x_n, x_n')$ is uniquely determined by $g(x_n)$ and $g(x_n')$. Therefore,

$$\mathbf{m}(N) \leq m(2N)$$

If $VC(G) = d$, we can use Chernoff bounds (Feller 1968) to estimate $m(2N)$ for $N \geq d$ as follows

$$\begin{aligned} m(2N) &< \sum_{i=0}^{d} \binom{2N}{i} \\ &\leq 2^{\mathcal{H}(d/2N)\times 2N} \end{aligned}$$

where $\mathcal{H}(\theta) = -\theta \log_2 \theta - (1-\theta)\log_2(1-\theta)$ is the binary entropy function. Therefore, once $\mathcal{H}(d/2N) \leq \frac{1}{2}$, $\mathbf{m}(N)$ will be less than 2^N and N must have reached, or exceeded, the VC dimension of $\mathbf{G}$. This happens at $N/d \approx 4.54$.

In many cases, the relationship between $VC(G \mid H)$ and $VC(G;H)$ can be roughly stated as follows: the smaller one is, the bigger the other is. Strong hints generally result in a small value of $VC(G \mid H)$ and a large value of $VC(G;H)$, while weak hints result in the opposite situation [the loose similarity with the average mutual information $I(X;Y)$ and the conditional entropy $H(X \mid Y)$ in information theory is the reason for choosing this notation for the various VC dimensions].

This relationship between $VC(G \mid H)$ and $VC(G;H)$ may suggest that we do not save when we use examples of a hint and, as a result, use fewer examples of the function. However, it should be noted that examples of the hint can be generated at will, while examples of the function may be limited in number or expensive to generate.

3 Catalyst Hints

Catalyst hints (Suddarth and Holden 1991) were introduced as a means of improving the learning behavior of feedforward networks. The idea is illustrated in Figure 1. A network attempting to learn the function $g = f$ is augmented by a catalyst neuron out of the last hidden layer. $g = f$ is augmented by a catalyst neuron out of the last hidden layer. This neuron is trained to learn a related function $g' = f'$. In doing so, the hidden layers of the network are influenced in a way that helps the main learning task $g = f$. After the learning phase is completed, the catalyst neuron is removed.

The catalyst function f' is typically a "well-behaved version" of f that can be learned more easily and more quickly. When f' is learned, the internal representations in the hidden layers of the network will be suited for the implementation of the main function f.

As a hint, namely a piece of information about f, the catalyst is the assertion that there is a way to set the weights of the network that simultaneously implements $g = f$ and $g' = f'$. Unlike invariances, catalysts are very particular to the network we use.

To formalize the catalyst hint, let $\mathcal{G}$ be the set of pairs of hypotheses (g, g') that can be simultaneously implemented by the network (when the catalyst neuron is present). The values of the weights in the different

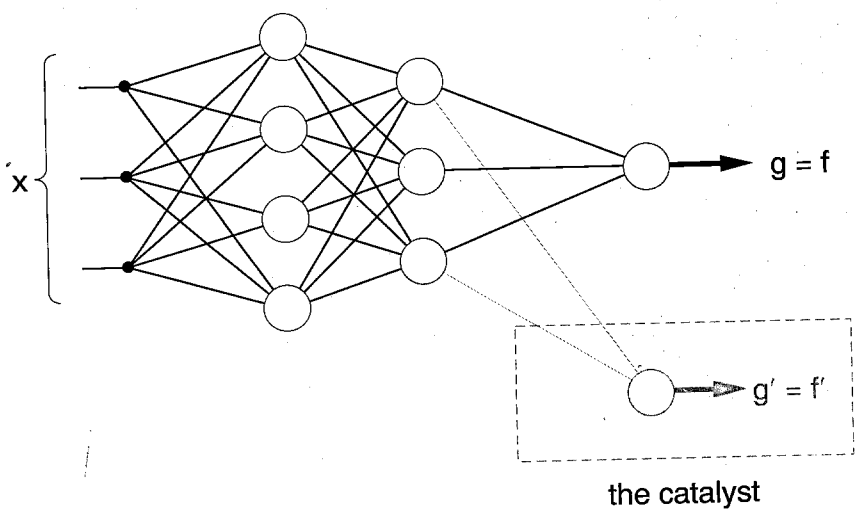

Figure 1: A network that uses a catalyst hint.

layers of the network determine (g, g'). A particular g may appear in different pairs (g, g') and, similarly, a particular g' may appear in different pairs (g, g'). Since the catalyst hint puts a condition on g', its impact on g is indirect through these pairings of (g, g').

This suggests the following notation: $(\mathbf{g}, g')$ denotes the hypothesis g when the catalyst hypothesis is g' and $(g, \mathbf{g}')$ denotes the hypothesis g' when the main hypothesis is g. Applied to a point $x \in X$, we use the convention

$$\begin{aligned}(\mathbf{g}, g')(x) &= g(x) \\ (g, \mathbf{g}')(x) &= g'(x)\end{aligned}$$

Thus $(\mathbf{g}, g')$ and $(g, \mathbf{g}')$ provide an inflated notation for the hypotheses g and g', respectively. In these terms, the set of hypotheses G is defined by

$$G = \{(\mathbf{g}, g') \mid (g, g') \in \mathcal{G}\}$$

To apply the VC dimension to catalyst hints, we will follow the same steps we used for invariance hints. The catalyst hint H is given by the constraint $g' = f'$. When H is absorbed, G is reduced to $\hat{G}$

$$\hat{G} = \{(\mathbf{g}, f') \mid (g, f') \in \mathcal{G}\}$$

Obviously, $\hat{G} \subseteq G$. The VC dimension of G given H is

$$VC(G \mid H) = VC(\hat{G})$$

Again, $VC(G \mid H) \leq VC(G)$. How small $VC(G \mid H)$ will be depends on the catalyst function f'. For instance, the degenerate case of a constant f' results in $VC(G \mid H) = VC(G)$ since the constant can be implemented by the catalyst neuron alone and would not impose any constraint on the weights of the original network. On the other hand, a complex f' will take specific combinations of weights to implement, thus significantly restricting the network and resulting in $VC(G \mid H) \ll VC(G)$. If the hint is only *partially* absorbed, the effective VC dimension lies between $VC(G)$ and $VC(G \mid H)$.

One situation that leads to partial absorption is when the hint is represented by examples. An example of the hint H: $g' = f'$ takes the form $g'(x) = f'(x)$. In this case, examples of H are of the same nature as examples of f; x is picked from X according to $P(x)$ and $f'(x)$ is evaluated. The definition of examples of H leads to the definition of $\mathbf{G}$, the set of agreement/disagreement patterns between the hypotheses and the hint. For each hypothesis $(\mathbf{g}, g') \in G$, there is a hypothesis $\mathbf{g} \in \mathbf{G}$ such that

$$\mathbf{g}(x) = \begin{cases} 1 & (g, \mathbf{g}')(x) = f'(x) \\ 0 & (g, \mathbf{g}')(x) \neq f'(x) \end{cases}$$

The VC dimension of $\mathbf{G}$ is the basis for defining $VC(G; H)$, the VC dimension that will indicate how many examples $[x, f'(x)]$ are needed to absorb H. It is given by

$$VC(G; H) = VC(\mathbf{G})$$

Unlike an invariance hint, the particular choice of a catalyst hint (the function f') does not affect the value of $VC(G; H)$.

The VC inequality asserts that a sufficient number of examples will lead to a hypothesis $(\mathbf{g}, g')$ that satisfies

$$\Pr[(g, \mathbf{g}')(x) \neq f'(x)] \to 0$$

where the probability is taken w.r.t. $P(x)$. Therefore, we will get a hypothesis g that pairs up with a good approximation of f'. This establishes a natural notion of approximating the hint.

4 Conclusion

We have analyzed two different types of hints, invariances and catalysts. The highlight of the analysis is the definition of $VC(G \mid H)$ and $VC(G; H)$. These two quantities extend the VC inequality to cover learning f given the hint, and learning the hint itself.

Other types of hints can be quite different from invariances and catalysts, and will require new analysis. However, the common method for dealing with any type of hint in this framework is as follows.

1. The definition of the hint should determine for each hypothesis in G whether or not it satisfies the hint. The set $\hat{G}$ contains those hypotheses which do satisfy the hint. $VC(G \mid H)$ is defined as $VC(\hat{G})$.

2. A scheme for representing the hint by examples should be selected. Each example is generated according to a probability distribution $\mathbf{P}$ that depends on the original distribution P. Different examples are generated independently according to the same distribution.

3. For every hypothesis and every example of the hint, we should be able to determine whether or not the hypothesis agrees with the example. The agreement/disagreement patterns define the set of hypotheses $\mathbf{G}$, and $VC(\mathbf{G})$ defines $VC(G; H)$. A hypothesis will agree with every possible example if, and only if, it satisfies the hint.

4. How well a hypothesis approximates the hint is measured by the probability (w.r.t. $\mathbf{P}$) that it will agree with a new example. An approximation in this sense should imply a partial absorption of the hint.

5. How the hint is represented by examples may not be unique. The choice of representation affects the definition of $VC(G;H)$ and also affects what partial absorption means. A minimum consistency requirement is that no hypothesis that strictly satisfies the hint should be excluded as a result of the partial absorption process. A good process will exclude as many hypotheses as possible without violating this requirement.

Our analysis here dealt with the situation where the unknown function f strictly satisfies the hint, and strictly belongs to G. Relaxing these conditions is worth further investigation. It is also worthwhile to extend this work to cover real-valued functions, as well as average-case measures instead of the worst-case VC dimension. Finally, schedules for mixing examples of f with examples of the hint in learning protocols are worth exploring.

Acknowledgment

This work was supported by AFOSR Grant 92-J-0398 and the Feynman-Hughes fellowship. The author wishes to thank Dr. Demetri Psaltis for a number of useful comments.

References

Abu-Mostafa, Y. 1990. Learning from hints in neural networks. *J. Complex.* **6**, 192–198.

Al-Mashouq, K., and Reed, I. 1991. Including hints in training neural networks. *Neural Comp.* **3**, 418–427.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. 1989. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM* **36**, 929–965.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications*, Vol. 2. Wiley, New York.

Fyfe, A. 1992. Invariance hints and the VC dimension. Ph.D. Thesis, Caltech.

Hu, M. 1962. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory* **IT-8**, 179–187.

Minsky, M., and Papert, S. 1988. *Perceptrons*, expanded edition. MIT Press, Cambridge, MA.

Omlin, C., and Giles, C. 1992. Training second-order recurrent neural networks using hints. In *Machine Learning: Proceedings of the Ninth International Conference (ML-92)*, D. Sleeman and P. Edwards, eds. Morgan Kaufmann, San Mateo, CA.

Suddarth, S., and Holden, A. 1991. Symbolic neural systems and the use of hints for developing complex systems. *Intl. J. Machine Stud.* **35**, 291.

Vapnik, V., and Chervonenkis, A. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probabil. Appl.* **16**, 264–280.

Received 21 April 1992; accepted 15 July 1992.

Communicated by Ralph Linsker

Redundancy Reduction as a Strategy for Unsupervised Learning

A. Norman Redlich
The Rockefeller University, 1230 York Ave., New York, NY 10021 USA

A redundancy reduction strategy, which can be applied in stages, is proposed as a way to learn as efficiently as possible the statistical properties of an ensemble of sensory messages. The method works best for inputs consisting of strongly correlated groups, that is *features*, with weaker statistical dependence between different features. This is the case for localized objects in an image or for words in a text. A local feature measure determining how much a single feature reduces the total redundancy is derived which turns out to depend only on the probability of the feature and of its components, but not on the statistical properties of any other features. The locality of this measure makes it ideal as the basis for a "neural" implementation of redundancy reduction, and an example of a very simple non-Hebbian algorithm is given. The effect of noise on learning redundancy is also discussed.

1 Introduction

Given sensory messages, for example, the visual images available at the photoreceptors, animals must identify those objects or scenes that have some value to them. This problem, however, can be very tricky since the image data (e.g., photoreceptor signals) may underdetermine the scene data (e.g., surface reflectances) needed to find and identify objects (Kersten 1990). In the case of very primitive organisms crude special purpose filters may suffice, such as the "fly detector" in frogs. But for more general object detection and for the reconstruction of physical scenes from noisy image data, some additional clues or constraints are needed. One type of clue is knowledge of the *statistical properties* of scenes and images (Attneave 1954, Barlow 1961, 1989). Such information can be used to recover physical scene data from noisy image data, as shown for example by Geman and Geman (1984). Barlow (1989) has also argued that such information is necessary for object recognition, since it allows objects to be discriminated from irrelevant background data. Also, since objects are encoded *redundantly* in sensory messages, knowing this redundancy can aid in their recognition.

But how can an organism go about *learning* the statistical properties of sensory messages? And second, what is the most efficient way of *stor-*

 © 1993 Massachusetts Institute of Technology