Capabilities and Training of Feedforward Nets

Eduardo D. Sontag
Rutgers University

1 Introduction

This paper surveys recent work by the author on learning and representational capabilities of feedforward nets. The learning results show that, among two possible variants of the so-called backpropagation training method for sigmoidal nets, both of which variants are used in practice, one is a better generalization of the older perceptron training algorithm than the other. The representation results show that nets consisting of sigmoidal neurons have at least twice the representational capabilities of nets that use classical threshold neurons, at least when this increase is quantified in terms of classification power. On the other hand, threshold nets are shown to be more useful when approximating implicit functions, as illustrated with an application to a typical control problem.

1.1 Classification Problems

One typical application of neural nets is in binary classification problems. During a training or supervised learning stage, examples and counterexamples, labeled *true* and *false* respectively, are presented, and parameters are adjusted so as to make the network's numerical output consistent with the desired labels. (Various conventions can be used: for instance, a positive output may be understood as *true* and a negative one as *false*.) Later, during actual operation, the output given by the net when a new input is presented will be taken as the network's guess at a classification.

One set of issues to be addressed when implementing the above ideas deals with the algorithms used for the adjustment of parameters during the training stage. Typically, a *cost* function is proposed, that measures the discrepancy between the desired output and the output of a net having a given set of parameters. One then attempts to minimize this cost, over the space of all parameters, to arrive at a network that best matches the given training data. This last step involves a hard nonlinear minimization problem, and several variants of gradient descent have been proposed to solve it; in fact, the use of sigmoidal activation functions –as opposed to simpler thresholds– is to a great extent motivated by the requirement that the cost should be differentiable as a function of the parameters, so that gradient descent can be employed. There is one major exception to the use of gradient descent and sigmoids, though, and that is the case of networks having no (or equivalently from the viewpoint of representation, just one) hidden unit; in that case one can apply either linear programming techniques or the classical "perceptron learning rule," because the problem is then essentially linear. Section 2 deals with minimization problems; we show that among two possible variants of the cost functions, the gradient descent approach applied to one restricts perfectly to the older perceptron rule, while the other results in serious potential problems, including convergence to inconsistent parameters. (It would appear that in practice, both techniques are used without regard to these dangers; in fact, the "bad" variant is the one most often described in papers.)

A second and equally important set of issues deals with the architecture of the network: type of activation functions, number of units, interconnection pattern. Typically, in classification problems, one uses either "threshold" or "sigmoidal" units and these are arranged in a feedforward manner. A good rule of thumb is that the number of units (and hence of tunable parameters) should be small; otherwise the network will tend to "memorize" the training data, with no data compression or "feature extraction," and it will not be able to classify correctly inputs which had not been seen during the training stage. (This informal rule can be justified in various manners, theoretically through the use of "PAC" learning, or ex-

perimentally through the generation of random training and testing inputs based on given probability distributions.) Thus it is of interest to study the minimal number of neurons needed in order to attain a certain classification objective; Section 3 below deals with this problem. Some of the main conclusions quantify the difference between the use of sigmoidal versus threshold nets. Section 4 deals with the problem of interpolation as opposed to classification, but for sigmoidal nets one obtains similar estimates as before. Finally, Section 5 shows that multiple hidden layers increase approximation capabilities, if one is interested in approximating not just continuous functions but also inverses of such functions, as illustrated with an example from control theory.

1.2 Basic Definitions

Let N be a positive integer. A *dichotomy* (S_-, S_+) on a finite set $S \subseteq \mathbb{R}^N$ is a partition $S = S_- \bigcup S_+$ of S into two disjoint subsets. (One often expresses this as a "coloring" of S into two colors.) A function $f : \mathbb{R}^N \to \mathbb{R}$ will be said to *implement* this dichotomy if it holds that

$$f(u) > 0 \text{ for } u \in S_+ \text{ and } f(u) < 0 \text{ for } u \in S_- \ .$$

We define a "neural net" as a function of a certain type, corresponding to the idea of feedforward interconnections, via additive links, of neurons each of which has a scalar response θ. For any fixed function $\theta : \mathbb{R} \to \mathbb{R}$, we say that f is a *single hidden layer neural net with k hidden neurons of type θ and no direct input to output connections* (or just that f is a "(k, θ)-net") if there are real numbers

$$w_0, w_1, \ldots, w_k, \tau_1, \ldots, \tau_k$$

and vectors

$$v_1, \ldots, v_k \in \mathbb{R}^N$$

such that, for all $u \in \mathbb{R}^N$,

$$f(u) \ = \ w_0 \ + \ \sum_{i=1}^{k} \ w_i \, \theta(v_i.u - \tau_i) \tag{1}$$

where the dot indicates inner product.

For fixed θ, and under mild assumptions on θ, such neural nets can be used to approximate uniformly arbitrary continuous functions on compacts. See for instance [6], [8]. In particular, they can be used to implement arbitrary dichotomies on finite sets.

In neural net practice, one often takes θ to be the *standard sigmoid*

$$\theta(x) = \sigma(x) = \frac{1}{1+e^{-x}}$$

or equivalently, up to translations and change of coordinates, the hyperbolic tangent $\theta(x) = \tanh(x)$. Another usual choice is the hardlimiter or *Heaviside* or *threshold* function

$$\theta(x) = \mathcal{H}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

which can be approximated well by $\sigma(\gamma x)$ when the "gain" γ is large. Most analysis, including studies of circuit complexity and the theory of threshold logic, has been done for $\mathcal{H}$, but as explained earlier, in practice one often uses the standard sigmoid.

2 Gradient Descent

Assume that one wishes to find a function f that implements a given dichotomy (S_-, S_+) on a set S, and f is to be chosen from a class of functions parameterized by variables $\mu = (\mu_1, \ldots, \mu_\kappa)$ (for instance, the class of all (k, θ)-nets, for fixed k and θ, parameterized by the possible weights and biases v_i, τ_i). Write for now $F(u, \mu)$ for the function $f(u)$ obtained when using the parameters μ. Thus, we wish to find a set of values for the parameters μ so that $F(u, \mu) > 0$ for $u \in S_+$ and $F(u, \mu) < 0$ for $u \in S_-$. As described in the Introduction, one approach is to set up an error function

$$E(\mu) := \sum_{u \in S} E_u(\mu)$$

where $E_u(\mu)$ measures the failure of $F(u, \mu)$ to be positive, if $u \in S_+$, or the failure of $F(u, \mu)$ to be negative, if $u \in S_+$. The question to

be treated next has to do with the precise choice of these measures $E_u(\mu)$. Since the point that we are interested in discussing is already well illustrated by the particular case of $(1,\theta)$-nets, we shall restrict for the remainder of this section to $k = 1$.

As far as recognition properties are concerned, nets with $k = 1$ implement linearly separable dichotomies. More precisely, assume that θ is a nondecreasing function, and that the following property holds:

(P) $t_+ := \lim_{x\to+\infty}\theta(x)$ and $t_- := \lim_{x\to-\infty}\theta(x)$ exist, $t_- < t_+$.

Fix any real number λ in the interval (t_-, t_+). Then, the following properties are equivalent, for any given finite dichotomy (S_-, S_+):

1. There is some $(1,\theta)$-net implementing this dichotomy.
2. There exist $\overline{v} \in \mathbb{R}^N$ and $\overline{\tau} \in \mathbb{R}$ such that

$$\overline{v}.u > \overline{\tau} \text{ for } u \in S_+ \text{ and } \overline{v}.u < \overline{\tau} \text{ for } u \in S_- \,. \qquad (2)$$

3. There exist $v \in \mathbb{R}^N$ and $\tau \in \mathbb{R}$ such that

$$\theta(v.u-\tau) > \lambda \text{ for } u \in S_+ \text{ and } \theta(v.u-\tau) < \lambda \text{ for } u \in S_- \,. \quad (3)$$

The equivalence is trivial from the definitions: Property 1 means that there are v, τ, w_0, w such that

$$w_0 + w\theta(v.u - \tau) \qquad (4)$$

has the right sign; if $w > 0$, this implies that $\theta(v.u_+-\tau) > \theta(v.u_--\tau)$ whenever $x_+ \in S_+$ and $x_- \in S_-$, which in turn implies (because θ is nondecreasing) that $v.x_+ > v.u_-$ for such pairs, and therefore Property 2 holds too. On the other hand, if Property 2 holds, then

$$\lim_{\gamma\to+\infty} \gamma(\overline{v}.u - \overline{\tau}) = +\infty$$

uniformly on the finite set S_+, which implies that $\theta(\gamma(\overline{v}.u - \overline{\tau})) > \lambda$ for all $u \in S_+$ if γ is large enough, and for the same reason $\theta(\gamma(\overline{v}.u - \overline{\tau})) < \lambda$ for all $u \in S_-$ for such γ, so Property 3 holds too. Finally, if

this latter Property holds then the net in Equation (4) implements the dichotomy, using $w = 1$ and $w_0 = -\lambda$.

Given a dichotomy, the problem of determining if there exist a vector $\overline{v} \in \mathbb{R}^N$ and a number $\overline{\tau} \in \mathbb{R}$ such that Equation (2) holds (that is, if the sets S_+ and S_- are *linearly separable*) is a simple linear programming question, and there are very efficient methods for solving it as well as for finding explicit solutions $(\overline{v}, \overline{\tau})$. More in connection with nets, the classical *perceptron learning procedure* (see e.g. [7]) provides a recursive rule for finding one such solution provided that any exist. The perceptron rule is very simple, and it is worth recalling next, since we shall later compare it to gradient descent. We write

$$\hat{u} := (u, -1) \tag{5}$$

for each element of S, and use the notations $\hat{S}$, $\hat{S}_+$, and $\hat{S}_-$ for the points of the form $(u, -1)$ with u in S, S_+, and S_- respectively. Using these notations, the question becomes that of finding a vector

$$\nu = (\overline{v}, \overline{\tau}) \tag{6}$$

such that $\nu.\hat{u} < 0$ if $\hat{u} \in \hat{S}_+$ and this inner product is negative on $\hat{S}_-$. We first give an arbitrary starting value for ν. Now the possible elements in $\hat{S}$ are presented one after the other in an infinite sequence, with the only restriction that every element must appear infinitely often. For each element of this sequence, the corresponding inequality is tested. If the sign of the inequality is wrong, the estimate for ν is updated as follows: $\nu := \nu + \hat{u}$ if $\hat{u}$ is in $\hat{S}_+$, and $\nu := \nu - \hat{u}$ if $\hat{u}$ is in $\hat{S}_-$; if the sign was right, no change is made. It is a very old result that this procedure converges in finitely many steps to a solution ν, when starting at any initial guess for ν, if the original sets are linearly separable.

Since the existence of $v \in \mathbb{R}^N$ and $\tau \in \mathbb{R}$ such that Equation (3) holds is equivalent to the solution of the linear separability problem, it would seem useless to study directly Property 3. However, we wish to do so in order to exhibit in this simplified case some problems that may arise in the general case of hidden units ($k > 1$), which is not so easy to analyse. (We have observed the same problems

experimentally, but there seems to be no easy way to give a theorem in the general case.) We deal with that Property next.

The restriction to the case $k = 1$ of the popular "backpropagation" technique for solving this problem –or at least of one often-used variant of it,– is essentially as follows. First pick two numbers α and β so that $t_- < \alpha < \lambda < \beta < t_+$. Using again the notation in Equations (5) and (6), now consider the cost

$$E(\nu) := (1/2) \sum_{\hat{u} \in \hat{S}} (\delta_{\hat{u}} - \theta(\nu.\hat{u}))^2 \tag{7}$$

where $\delta_{\hat{u}}$ equals β if $\hat{u} \in \hat{S}_+$ and equals α otherwise. (The terminology "target values" is standard for α and β.) One now attempts to minimize E as a function of ν. If a small value results, it will follow that $\theta(\nu.\hat{u})$ will be approximately equal to α, and hence less than λ, when $\hat{u} \in \hat{S}_-$, and similarly for elements of $\hat{S}_+$, and the classification problem is solved. Unfortunately, there are local minima problems associated to this procedure; see for instance [17], [4], [11]. False (i.e., non-global) local minima can occur *even if the sets are separable*. Moreover, even if there is only one local (and hence global) minimum, the resulting solution may fail to separate (hence the title of [4]). (For a theoretical study of local minima in a more general hidden-unit case, $k > 1$, see for instance [3] and references there.)

The reason for this difficulty is very simple to understand, and is as follows: In trying to minimize E, one is trying to fit the values α and β *exactly*, when it would be sufficient for classification that $\theta(\nu.\hat{u})$ be less than α for $\hat{u}$ in $\hat{S}_-$, and bigger than β for $\hat{u}$ in $\hat{S}_+$. The precise fitting may force the parameters ν to be chosen so as to make most terms small at the cost of leaving one term large. This can be illustrated with an example. Assume that $N = 1$ and S consists of five points u_i so that u_1 and u_2 are very close to -1, u_3 and u_4 are very close to 1, and $u_5 = -0.9$. The dichotomy is given by $S_- := \{u_1, u_2\}$ and $S_+ := \{u_3, u_4, u_5\}$. Obviously these are linearly separable (pick $\overline{v} = 1, \overline{\tau} = -0.95$). We now pose the minimization problem, using the standard sigmoid $\theta = \sigma$, $\lambda := 0.5$, and target values $\alpha := 0.2$, $\beta := 0.8$. That is, we must minimize the error

$$(0.8 - \sigma(-0.9v - \tau))^2 + 2\,(0.8 - \sigma(v - \tau))^2 + 2\,(0.2 - \sigma(-v - \tau))^2$$

as a function of v and τ. There is a unique local (and global) minimum, attained at the unique values $v = 0.971$ and $\tau = -0.516$. It turns out that these parameter values do not separate: for $\hat{u} = (-0.9, 1)$, $\sigma(\nu.\hat{u}) = 0.411 < 0.5$. The classification of x_5 has been traded-off for a better perfect fit of the other points. Note that the cutoff between classes happens approximately at $\hat{u} = -1/2$. (Of course, redefining λ as 0.4 one could say that this solution f separates, but this redefinition can only be done a fortriori, once we know the solution. In any case, it is easy to give examples in dimension $N = 2$ in which the same pathology happens but where there is no possible redefinition of λ that helps; see the references cited earlier.) Figure 1 (function f in the plot, darker one) shows the resulting $(1, \sigma)$-net.

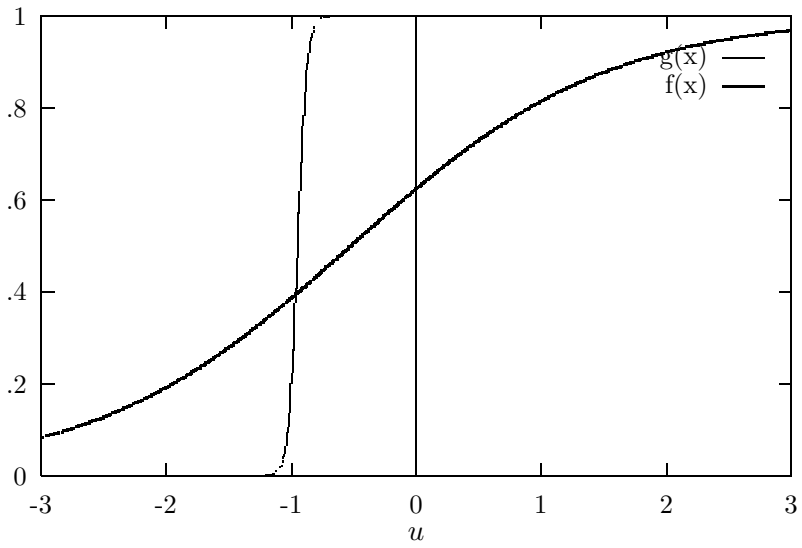

Figure 1: *Non-separating minimum*

The second plot in Figure 1, corresponding to the function

$$g(x) = \sigma(27.73x + 26.34) \ ,$$

does satisfy Property 3 (the values for -1, -0.9 and 1 are at 0.2, 0.8, and approximately 1, respectively; the cutoff with $\lambda = 0.5$ happens at $u = -0.95$). The parameters $(27.73, 26.34)$ were obtained as described below; they are *not* obtained by minimizing the above error function.

The solution to the difficulties raised by the above discussion, namely that gradient descent may give wrong results even when the data is separable, lies in not adding a penality if $\theta(\nu.\hat{u})$ is already less than α and $\hat{u} \in \hat{S}_-$ (and similarly for $\hat{S}_+$). In other words, the corresponding term in Equation (7) is taken to be zero. Thus one has to minimize the new function

$$E^*(\nu) \;:=\; (1/2) \sum_{\hat{u} \in \hat{S}} (\delta_{\hat{u}} - \theta(\nu.\hat{u}))^2_{\varepsilon(\hat{u})} \tag{8}$$

where we are using the notations

$$(r)^2_+ := \begin{cases} 0 & \text{if } r \leq 0 \\ r^2 & \text{if } r > 0 \end{cases}$$

and $(r)^2_- := (-r)^2_+$ for any number r, and $\varepsilon(\hat{u}) = +$ if $\hat{u} \in \hat{S}_+$ or $\varepsilon(\hat{u}) = -$ if $\hat{u} \in \hat{S}_-$.

The new error function E^* will be called, for lack of a better term, the "nonstrict" error measure corresponding to the problem at hand; note that E is differentiable (if θ is), but in general is not second-order differentiable. To motivate the use of this measure, a comparison with the perceptron procedure is useful. A discrete gradient descent step for minimizing E^* takes the form of updating each parameter ν_i (where ν_i denotes the ith coordinate of ν) by iterating the rule:

$$\nu_i \;:=\; \nu_i + \rho \hat{u}_i$$

($\hat{u}_i$ is the ith coordinate of $\hat{u}$), where

$$\rho = (\delta_{\hat{u}} - \theta(\nu.\hat{u}))\theta'(\nu.\hat{u}) \tag{9}$$

if the classification is incorrect and $\rho = 0$ otherwise. This is the precise analogue of the perceptron rule (for which ρ is always either

zero or ± 1). When using E instead of E^* one would use Equation (9) always, even if the classification was correct. The function g in Figure 1 was obtained by minimizing E^* (we used a numerical technique, as a test of the method, but the solution can be obtained in closed form: just fit exactly the values at -1 and -0.9 to obtain $v = 20 \ln 4$ and $\tau = -19 \ln 4$; the value at 1 has been relaxed to about 1, which is greater than 0.8 but contributes zero error in the nonstrict error measure).

The use of E^* was first suggested in [17], who also proved a convergence result under restrictive hypotheses (which do not allow for sigmoids). In [16], we proved that there are no false local minima for E^* if the data is separable, and that the gradient descent differential equation converges in finite time to a separating solution, from a random initial state. Note that the unique minimum of E^* is zero, for separable data, and it is achieved at any separating solution. (Actually, the result proved there is considerably more general, as it deals with a wider class of optimization problems. The only difficulty in the proof has to do with the fact that ρ will tend to zero; one has to use a dynamical-systems argument involving LaSalle invariance.)

Thus we conclude that the use of a nonstrict error function provides the correct generalization of the perceptron learning rule. This provides strong evidence that one should use nonstrict error functions also in the general (hidden unit) case. In [16] we also compared the sigmoid results to known facts in pattern recognition, where nonstrict measures had also been proposed (for linear activation functions σ). This paper also showed why, *even if using the nonstrict measure*, there may be false local minima (for nonseparable data), even for $k = 1$ (no hidden units) and binary training vectors; the necessary counterexample was based on a construction in [15].

3 Representational Capabilities

One may express the classification power of a class of functions, such as those computable by neural nets with a fixed architecture and a fixed number of neurons, in terms of set shattering. In this approach, a class of functions is considered to be more powerful than another

if it can be used to implement arbitrary partitions on sets of larger cardinality.

Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^N$ to $\mathbb{R}$, assumed to be nontrivial, in the sense that for each point $u \in \mathbb{R}^N$ there is some $f_1 \in \mathcal{F}$ so that $f_1(u) > 0$ and some $f_2 \in \mathcal{F}$ so that $f_2(u) < 0$. This class *shatters* the set $S \subseteq R^N$ if each dichotomy on S can be implemented by some $f \in \mathcal{F}$.

As in [12], we consider for any class of functions $\mathcal{F}$ as above, the following two measures of classification power, dealing with "best" and "worst" cases respectively: $\overline{\mu}(\mathcal{F})$ denotes the largest integer $1 \leq l$ (possibly ∞) so that there is at least *some* set S of cardinality l in $\mathbb{R}^N$ which can be shattered by $\mathcal{F}$, while $\underline{\mu}(\mathcal{F})$ is the largest integer $1 \leq l$ (possibly ∞) so that *every* set of cardinality l can be shattered by $\mathcal{F}$. Note that $\underline{\mu}(\mathcal{F}) \leq \overline{\mu}(\mathcal{F})$ for every class $\mathcal{F}$. The integer $\overline{\mu}$ is the same as the *Vapnik-Chervonenkis (VC) dimension* of the class $\mathcal{F}$ (see for instance [2] for VC dimension).

Some of the results obtained in [12] are as follows. We use the notation $\underline{\mu}(k, \theta, N)$ for $\underline{\mu}(\mathcal{F})$ if $\mathcal{F}$ is the class of (k, θ)-nets in $\mathbb{R}^N$, and similarly for $\overline{\mu}$.

Theorem 1 *For each k, N, $\underline{\mu}(k, \mathcal{H}, N) = k + 1$, $\underline{\mu}(k, \sigma, N) \geq 2k$.* ∎

Theorem 2 *For each k, $2k + 1 \leq \overline{\mu}(k, \mathcal{H}, 2)$, $4k - 1 \leq \overline{\mu}(k, \sigma, 2)$.* ∎

The main conclusion from the first result is that sigmoids at least double recognition power for arbitrary sets. We conjecture that

$$\frac{\overline{\mu}(k, \sigma, N)}{\overline{\mu}(k, \mathcal{H}, N)} \;=\; 2 + O\left(\frac{1}{k}\right)$$

for all N; this is true for $N = 1$ and is strongly suggested by Theorem 2 (the first bound appears to be quite tight). Unfortunately the proof of this theorem is based on a result from [1] regarding arrangements of points in the plane, a fact which does not generalize to dimension three or higher. Other results in [12] deal with the effect of direct connections from inputs to outputs.

Finally, we also gave in [12] results valid specifically for sets of binary vectors. For example, it is a trivial consequence from the given

results that parity on n bits can be computed with $\lceil \frac{n+1}{2} \rceil$ hidden sigmoidal units (rather than the n that –apparently, though it is still

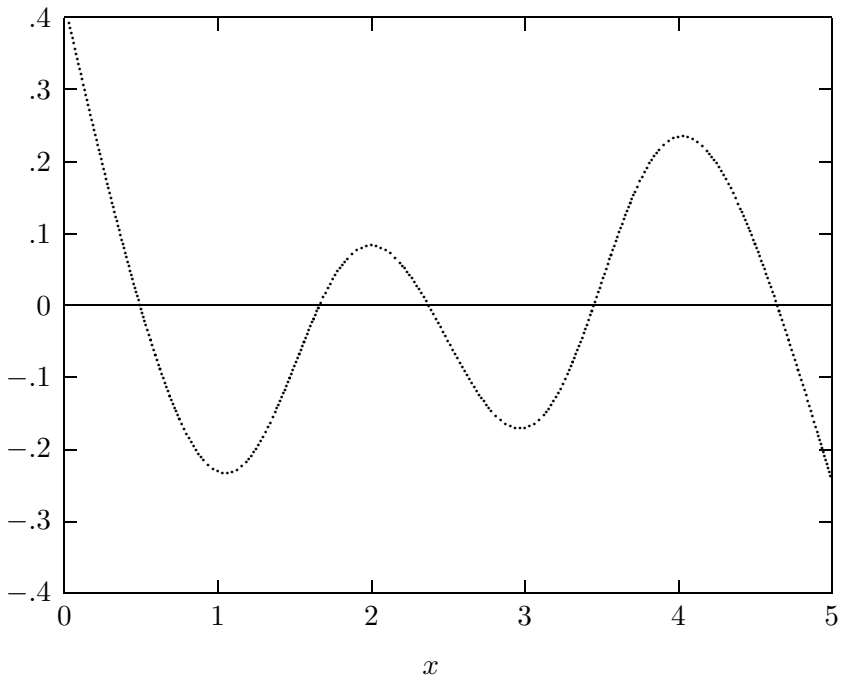

Figure 2: 5-bit parity with 3 sigmoids

an open problem– are needed when using Heavisides). For instance,

$$f(x) := 10.4 + 2\sigma(3x - 4.5) + 2\sigma(3x - 10.5) - 20\sigma(x/5)$$

computes 5-bit parity with 3 sigmoidal neurons, $x := \sum x_i$. (See Figure 2.)

It is also shown in [12] that the function of $2n$ variables (say, with n odd),

$$\mathrm{XOR}\,(\mathrm{MAJ}\,(x_1, \ldots, x_n), \mathrm{MAJ}\,(y_1, \ldots, y_n))$$

(where $\mathrm{MAJ}\,(x_1, \ldots, x_n)$ is the "majority" function that equals "1" if the majority of the x_i's are one, and zero otherwise,) can be implemented by nets with a fixed number (four) of sigmoidal neurons, that

is, a $(4, \sigma)$-net, independently of n, but it can be shown –personal communication by W. Maass– that it is impossible to implement such functions with $(k, \mathcal{H})$-nets, k independent of n.

4 Interpolation

Assume now that θ satisfies (P) with $t_- = -1$ and $t_+ = 1$ (this can be always assumed after rescaling) that in addition it is continuous and there is some point c such that θ is differentiable at c and $\theta'(c) \neq 0$. For instance, the standard response function tanh (or σ after rescaling) satisfies these properties. For such a θ we have:

Theorem 3 *Given any $2n+1$ (distinct) points $x_0, \ldots, x_{2n}$ in R^N, any $\varepsilon > 0$, and any sequence of real numbers $y_0, \ldots, y_{2n}$, there exists some $(n+1, \theta)$-net f such that $|f(x_i) - y_i| < \varepsilon$ for each i.*

Before proving this theorem, we establish an easy technical result:

Lemma 4.1 Assume given real numbers $p, q, \alpha, \beta, \varepsilon, \delta$ so that $\varepsilon > 0$, $\delta > 0$, and $\alpha < q < \beta$. Then, there exists some real numbers a, b, c, d so that, if $f(x) := d + a\theta(bx + c)$, then the following properties hold:

1. $f(p) = q$.
2. $|f(x) - \alpha| < \varepsilon$ for all $x \leq p - \delta$.
3. $|f(x) - \beta| < \varepsilon$ for all $x \geq p + \delta$.

Proof. Let $\rho > 0$ be smaller than $\beta - q$, $q - \alpha$, and ε. Consider the function

$$g(\xi) := \frac{\beta - \alpha}{2}\theta(\xi) + \frac{\beta + \alpha}{2} .$$

Note that $g(\xi)$ approaches α, β at $-\infty, +\infty$, so there is some $K > 0$ so that $|g(\xi) - \alpha| < \rho$ if $\xi \leq -K$ and $|g(\xi) - \beta| < \rho$ if $\xi \geq K$. Pick any $\gamma > 2K/\delta$ and define for this γ, $f_0(x) := g(\gamma x)$. Then,

$$|f_0(x) - \alpha| < \rho \text{ if } x \leq -\delta/2$$

and

$$|f_0(x) - \beta| \ < \ \rho \text{ if } x \geq \delta/2 \ .$$

As $f_0(\delta/2) > \beta - \rho > q$ and $f_0(-\delta/2) < \alpha + \rho < q$, by continuity of f_0 (here we use that θ is continuous) there must be some $u \in (-\delta/2, \delta/2)$ so that $f_0(u) = q$. Finally, we let

$$f(x) \ := \ f_0(x + u - p) \ .$$

Clearly this satisfies $f(p) = q$. For any $x \leq p - \delta$ it holds that $z := x + u - p \leq -\delta/2$, so $|f(x) - \alpha| = |f_0(z) - \alpha| < \rho < \varepsilon$, as desired. The property for $x \geq \delta/2$ is proved analogously. ∎

Now we prove Theorem 3. Note first that it is sufficient to prove it for $N = 1$, as one can perform the usual reduction of first finding a vector v whose inner products with the x_i's are all distinct. Now assume that we have already proved that for any two *increasing* sequences of real numbers

$$x_0 < x_1 < \ldots < x_{2n} \ \text{ and } \ z_0 < z_1 < \ldots < z_{2n} \tag{10}$$

there is some (k, θ)-net so that

$$|f(x_i) - z_i| < \varepsilon/2 \tag{11}$$

for each i. The result then follows from here. Indeed, given the original data, we may assume that the x_i are already in increasing order (reorder them, if necessary). Now pick any real d so that

$$d \ > \ \frac{y_i - y_{i+1}}{x_{i+1} - x_i}$$

for all $i = 0, \ldots, 2n-1$. Letting $z_i := x_i d + y_i$, these are now in increasing order. Let f be so that equation (11) holds for each i. By Lemma 7.2 in [12], there are some numbers a, b, c so that

$$|a + \theta(bx_i + c) + dx_i| \ < \ \varepsilon/2$$

for each i. Then $|f(x_i) + a + \theta(bx_i + c) - y_i| < \varepsilon$, as wanted.

Thus, we must prove the result for the particular case of increasing sequences (10), which we do via an argument somewhat analogous to that used in [5] for showing the (weaker) fact that one can approximately interpolate n points using $n-1$ neurons. We show inductively:

Given data (10) and any $\varepsilon > 0$, there exists an (n, θ)-net f so that

$$|f(x_i) - z_i| < \varepsilon \text{ for each } i = 0, \ldots, 2n \tag{12}$$

and

$$|f(x) - z_{2n}| < \varepsilon \text{ for all } x \geq x_{2n} . \tag{13}$$

For $n = 1$ this follows from Lemma 4.1, by choosing $p = x_1$, $q = z_1$, $\alpha = z_0$, $\beta = z_2$, and δ less than $x_1 - x_0$ and $x_2 - x_1$. Assume now that an $(n-1, \theta)$-net f_1 has been obtained for $x_0, \ldots, x_{2n-2}$ and $z_0, \ldots, z_{2n-2}$, and so that

$$|f_1(x_i) - z_i| < \varepsilon/2 \text{ for each } i = 0, \ldots, 2n-2 \tag{14}$$

and

$$|f_1(x) - z_{2n-2}| < \varepsilon/2 \text{ for all } x \geq x_{2n-2} . \tag{15}$$

Note that this last inequality holds in particular for x_{2n-1} as well as for all $x \geq x_{2n}$. Now let f_2 be as in Lemma 4.1, with δ less than $x_{2n-1} - x_{2n-2}$ and $x_{2n} - x_{2n-1}$, $\alpha = 0$, $\beta = z_{2n} - z_{2n-2}$, $q = z_{2n-1} - z_{2n-2}$, and $p = x_{2n-1}$, and so that

$$|f_2(x)| < \varepsilon/2 \text{ for all } x < x_{2n-1} - \delta \tag{16}$$

and

$$|f_2(x) - \beta| < \varepsilon/2 \text{ for all } x > x_{2n-1} + \delta . \tag{17}$$

It follows that $f := f_1 + f_2$ is as desired for the inductive step. This completes the proof of the Theorem. ∎

Thus we can approximately interpolate any $2n-1$ points using n sigmoidal neurons. It is not hard to prove as a corollary that, for the standard sigmoid, this "approximate" interpolation property holds in the following stronger sense: for an open dense set of $2n-1$ points, one can achieve an open dense set of values; the proof involves looking first at points with rational coordinates, and using

that on such points one is dealing basically with rational functions (after a diffeomorphism), plus some theory of semialgebraic sets. We conjecture that one should be able to interpolate at $2n$ points. Note that for $n = 2$ this is easy to achieve: just choose the slope d so that some $z_i - z_{i+1}$ becomes zero and the z_i are allowed to be nonincreasing or nondecreasing. The same proof, changing the signs if necessary, gives the wanted net.

5 Inverting Functions

Until now, we dealt only with single-hidden layer nets. As remarked above, such nets are "universal" approximators, in the sense that they are dense in the space of continuous functions on any compact subset of $\mathbb{R}^n$, with uniform norm, and they are also dense in L^p, on compacts, for any finite p. These approximations hold for almost arbitrary functions θ. However, in a certain sense, which we describe next, such nets are less powerful than nets with two hidden layers. We will say that a function $f : \mathbb{R}^N \to \mathbb{R}$ is *computable by a two-hidden-layer net* if there exist l functions $f_1, \ldots, f_l$, $l \geq 0$, so that $f(u) = w_0 + \sum_{i=1}^{l} w_i \theta(f_i(u))$ for some $w_1, \ldots, w_l \in \mathbb{R}$ and the f_i's are single-hidden-layer nets.

It is proved in [14] that for each integers m, p, any continuous function $f : \mathbb{R}^m \to \mathbb{R}^p$, any compact subset $C \subseteq \mathbb{R}^p$ included in the image of f, and any $\varepsilon > 0$, there exists some function $\phi : \mathbb{R}^m \to \mathbb{R}^p$, each of whose coordinates is computable by a two-hidden-layer net, so that $\|f(\phi(u)) - u\| < \varepsilon$ for all $u \in C$. That is to say, one may always obtain approximations of (one-sided) inverses of continuous maps. (The proof is not hard, and follows closely the ideas in [9].) But it is also proved there that there are examples of functions f as above whose inverses cannot be approximated by single-hidden-layer nets. Essentially, the difficulty has to do with the fact that the universal approximation results do not hold in L^∞.

The results are applied in [14] to the following problem. Consider nonlinear control systems

$$x(t+1) \;=\; P(x(t), u(t)) \tag{18}$$

whose states evolve in $\mathbb{R}^n$, having controls $u(t)$ that take values in $\mathbb{R}^m$, and with P sufficiently smooth and satisfying $P(0,0) = 0$. We assume that the system can be *locally* stabilized with linear feedback, i.e. there is some matrix F so that the closed loop system with right-hand side $P(x(t), Fx(t))$ is locally asymptotically stable. (See for instance [13], Section 4.8, for more on the topic of nonlinear stabilizability.) The system (18) is *asymptotically controllable* if for each state x_0 there is some infinite control sequence $u(0), u(1), \ldots$ such that the corresponding solution with $x(0) = x_0$ satisfies that $x(t) \to 0$ as $t \to \infty$. This condition is obviously the weakest possible one if any type of controller is to stabilize the system. We say that $K : \mathbb{R}^n \to \mathbb{R}^m$ is a *type-1 feedback* if each coordinate of K has the form $Fx + f(x)$, where F is linear and f is a single-hidden layer net, and that it is a *type-2 feedback* if it can be written in this manner with f is computable by a two-hidden layer net. These are feedback laws that can be computed by nets with one or two hidden layers respectively, and having m output neurons and possible direct connections from inputs to outputs (these connections provide the linear term). The following is also proved in [14]:

Theorem 4 *Let $\theta = \mathcal{H}$. For each system as above, and each compact subset C of the state space, there exists some feedback law of type 2 that globally stabilizes (18) on C, that is, so that C is in the domain of asymptotically stability of $x^+ = P(x, K(x))$. On the other hand, there exist systems as above, and compact sets C, for which no possible feedback law of type 1 stabilizes C.* ∎

The positive result, in this case as well as for continuous time systems under sampling, is based on ideas from [10]. The negative result remains if $\theta = \tanh$, or if θ is one of many other functions used in neural net practice. Basically, the proof is based on the fact that single-layer nets cannot approximate inverses, but two hidden layers are sufficient. Other problems of interest, such as inverse kinematics approximation, have the same flavor.

6 Acknowledgements

This research was supported in part by Siemens Corporate Research, Princeton, NJ, and in part by the CAIP Center, Rutgers University.

Bibliography

[1] T. Asano, J. Hershberger, J. Pach, E.D. Sontag, D. Souivaine, and S. Suri, *Separating Bi-Chromatic Points by Parallel Lines*, Proceedings of the Second Canadian Conference on Computational Geometry, Ottawa, Canada, 1990, p. 46-49.

[2] E.B. Baum, *On the capabilities of multilayer perceptrons*, J.Complexity **4**, 1988, p. 193-215.

[3] E.K. Blum *Approximation of Boolean functions by sigmoidal networks: Part I: XOR and other two-variable functions*, Neural Computation **1**, 1989, p. 532-540.

[4] M. Brady, R. Raghavan and J. Slawny, *Backpropagation fails to separate where perceptrons succeed*, IEEE Trans. Circuits and Systems **36**, 1989, p. 665-674.

[5] D. Chester, *Why two hidden layers and better than one*, Proc. Int. Joint Conf. on Neural Networks, Washington, DC, Jan. 1990, IEEE Publications, 1990, p. I.265-268.

[6] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control, Signals, and Systems **2**, 1989, p. 303-314.

[7] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[8] K.M. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2**, 1989, p. 359-366.

[9] E.D. Sontag, *Remarks on piecewise-linear algebra*, Pacific J.Math., **98**, 1982, p. 183-201.

[10] E.D. Sontag, *Nonlinear regulation: The piecewise linear approach*, IEEE Trans. Autom. Control **AC-26**, 1981, p. 346-358.

[11] E.D. Sontag, *Some remarks on the backpropagation algorithm for neural net learning*, Report SYCON-88-02, *Rutgers Center for Systems and Control*, June 1988.

[12] E.D. Sontag, *Comparing sigmoids and Heavisides*, Proc. Conference Info. Sci. and Systems, Princeton, 1990, p. 654-659.

[13] E.D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Springer, New York, 1990.

[14] E.D. Sontag, *Feedback Stabilization Using Two-Hidden-Layer Nets*, Report SYCON-90-11, *Rutgers Center for Systems and Control*, October 1990.

[15] E.D. Sontag and H.J. Sussmann *Backpropagation can give rise to spurious local minima even for networks without hidden layers*, Complex Systems **3**, 1989, p. 91-106.

[16] E.D. Sontag and H.J. Sussmann, *Backpropagation separates when perceptrons do*, in Proc. IEEE Int. Conf. Neural Networks, Washington, DC, June 1989, p. I-639/642.

[17] B.S. Wittner and J.S. Denker, *Strategies for teaching layered networks classification tasks*, Proc. Conf. Neural Info. Proc. Systems, Denver, 1987, Dana Anderson (Ed.), AIP Press.