Learning Internal Representations*

Jonathan Baxter
Department of Mathematics
London School of Economics
and
Department of Computer SCience
Royal Holloway College
University of London

Abstract

Probably the most important problem in machine learning is the preliminary biasing of a learner's hypothesis space so that it is small enough to ensure good generalisation from reasonable training sets, yet large enough that it contains a good solution to the problem being learnt. In this paper a mechanism for *automatically* learning or biasing the learner's hypothesis space is introduced. It works by first learning an appropriate *internal representation* for a learning environment and then using that representation to bias the learner's hypothesis space for the learning of future tasks drawn from the same environment.

An internal representation must be learnt by sampling from *many similar tasks*, not just a single task as occurs in ordinary machine learning. It is proved that the number of examples m *per task* required to ensure good generalisation from a representation learner obeys $m = O(a+b/n)$ where n is the number of tasks being learnt and a and b are constants. If the tasks are learnt independently (*i.e.* without a common representation) then $m = O(a+b)$. It is argued that for learning environments such as eech and character recognition $b \gg a$ and hence representation learning in these environments can potentially yield a drastic reduction in the number of examples required per task. It is also proved that if $n = O(b)$ (with $m = O(a+b/n)$) then the representation learnt will be good for learning novel tasks from the same environment, and that the number of examples required to generalise well on a novel task will be reduced to $O(a)$ (as opposed to $O(a+b)$ if no representation is used).

It is shown that gradient descent can be used to train neural network representations and the results of an experiment are reported in which a neural network representation was learnt for an environment consisting of *translationally invariant* Boolean functions. The experiment provides strong qualitative support for the theoretical results.

*Appeared in *Proceedings of the 8th International ACM Workshop on Computational Learning Theory* (long talk).

1 Introduction

It has been argued elsewhere (for example, see [2]) that the main problem in machine learning is the biasing of the learner's hypothesis space sufficiently well to ensure good generalisation from a relatively small number of examples. Once suitable biases have been found the actual learning task is relatively trivial. Despite this conclusion, much of machine learning theory is still concerned only with the problem of quantifying the conditions necessary for good generalisation *once* a suitable hypothesis space for the learner has been found; virtually no work appears to have been done on the problem of how the learner's hypothesis space is to be selected in the first place. This paper presents a new method for automatically selecting a learner's hypothesis space: *internal representation learning*.

The idea of automatically learning internal representations is not new to machine learning. In fact the huge increase in Artificial Neural Network (henceforth ANN) research over the last decade can be partially attributed to the promise—first given air in [5]—that neural networks can be used to *automatically* learn appropriate internal representations. However it is fair to say that despite some notable isolated successes, ANNs have failed to live up to this early promise. The main reason for this is not any inherent deficiency of ANNs as a machine learning model, but a failure to realise the true source of information necessary to generate a good internal representation.

Most machine learning theory and practice is concerned with learning a *single* task (such as "recognise the digit '1'") or at most a handful of tasks (recognise the digits '0' to '9'). However it is unlikely that the information contained in a small number of tasks is sufficient to determine an appropriate internal representation for the tasks. To see this, consider the problem of learning to recognise the handwritten digit '1'. The most extreme representation possible would be one that completely solves the classification problem, i.e. a representation that outputs 'yes' if its input is an image of a '1', regardless of the position, orientation, noise or writer dependence of the original digit, and 'no' if any other image is presented to it. A learner equipped with such a representation would require only one positive and one negative example of of the digit to learn to recognise it perfectly. Although the representation in this example certainly reduces the complexity of the learning problem, it does not really seem to capture what is meant by the term *representation*. What is wrong is that although the representation is an excellent one for learning to recognise the digit '1', *it could not be used for any other learning task*. A representation that is appropriate for learning to recognise '1' should also be appropriate for other character recognition problems—it should be good for learning other digits, or the letters of the alphabet, or Kanji characters, or Arabic letters, and so on. Thus the information necessary to determine a good representation is not contained in a single learning problem (recognising '1'), but is contained in many examples of similar learning problems. The same argument can be applied to other familiar learning domains, such as face recognition and speech recognition. A representation appropriate for learning a single face should be appropriate for learning all faces, and similarly a single word representation should be good for all words (to some extent even regardless of the language).

In the rest of this paper it is shown how to formally model the process of sampling from many similar learning problems and how information from such a sample can be used to learn an appropriate representation. If n learning problems are learnt independently then the number of examples m required per problem for good generalisation obeys $m = O(a + b)$, whereas if a common representation is learnt for all the problems then $m = O(a + b/n)$. The origin of the constants a and b will be explained in section 3 and it is argued in section 2 that for common learning domains such as speech and image recognition $b \gg a$, hence representation learning in such environments can potentially yield a drastic reduction in the number of examples required for good generalisation. It will also be shown that if a representation is learnt on $n = O(b)$ tasks then with high probability it will be good for learning novel tasks and that the sampling burden for good generalisation on novel tasks will be reduced $m = O(a)$, in constrast to $m = O(a+b)$ if no representation is used.

In section 4 it is shown how to use gradient descent to train neural network representations, and the results of an experiment using the technique are reported in section 5. The experiment provides strong qualitative support for the theoretical results.

2 Mathematical framework

Haussler's [3] *statistical decision theoretic* formulation of ordinary machine learning is used throughout this paper as it has the widest applicability of any formulation to date. This formulation may be summarised as follows. The learner is provided with a *training set* $\vec{z} = (z_1, \ldots, z_m)$ where each *example* $z_i = (x_i, y_i)$ consists of an *input* $x_i \in X$ and an *outcome* $y_i \in Y$. The training set is generated by m independent trials according to some (unknown) joint probability distribution P on $Z = X \times Y$. In addition the learner is provided with an *action* space A, a *loss function* $l\colon Y \times A \to [0, M]$ and a *hypothesis space* $\mathcal{H}$ containing functions $h\colon X \to A$. Defining the *true loss* of hypothesis h with respect to distribution P as

$$E(h, P) = \int_{X \times Y} l\,(y, h(x))\; dP(x, y), \qquad (1)$$

the goal of the learner is to produce a hypothesis $h \in \mathcal{H}$ that has true loss as small as possible. $l(y, h(x))$ is designed to give a measure of the *loss* the learner suffers, when, given an input $x \in X$, it produces an action $h(x) \in A$ and is subsequently shown the outcome $y \in Y$.

If for each $h \in \mathcal{H}$ a function $l_h\colon Z \to [0, M]$ is defined by $l_h(z) = l(y, h(x))$ for all $z = (x, y) \in Z$, then $E(h, P)$ can be expressed as the expectation of l_h with respect to P,

$$E(h, P) = \mathbb{E} l_h P = \int_Z l_h(z)\, dP(z).$$

Let $l_{\mathcal{H}} = \{l_h\colon h \in \mathcal{H}\}$. The measure P and the σ-algebra on Z are assumed to be such that all the $l_h \in l_{\mathcal{H}}$ are P-measurable (see definition 3).

To minimize the true loss the learner searches for a hypothesis minimizing the *empirical loss* on the sample $\vec{z}$,

$$E(h, \vec{z}) = \mathbb{E} l_h \vec{z} = \frac{1}{m} \sum_{i=1}^{m} l_h(z_i). \qquad (2)$$

To enable the learner to get some idea of the environment in which it is learning and hopefully then extract some of the bias inherent in the environment, it is assumed that the environment consists of a *set* of probability measures $\mathcal{P}$ and an *environmental measure* Q on $\mathcal{P}$. Now the learner is not just supplied with a single sample $\vec{z} = (z_1, \dots, z_m)$, sampled according to some probability measure $P \in \mathcal{P}$, but with n such samples $(\vec{z}_1, \dots, \vec{z}_n)$. Each sample $\vec{z}_i = (z_{i1}, \dots, z_{im})$, for $1 \le i \le n$, is generated by first sampling from $\mathcal{P}$ according to the environmental measure Q to generate P_i, and then sampling m times from P_i to generate $\vec{z}_i = (z_{i1}, \dots, z_{im})$. Denote the entire sample by $\mathbf{z}$ and write it as an $n \times m$ (n rows, m columns) matrix over Z:

$$\mathbf{z} = \begin{matrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{matrix}$$

Denote the $n \times m$ matrices over Z by $Z^{(n,m)}$ and call a sample $\mathbf{z} \in Z^{(n,m)}$ generated by the above process an *(n,m)-sample*.

To illustrate this formalism, consider the problem of *character recognition*. In this case X would be the space of all images, Y would be the set $\{0, 1\}$, each probability measure $P \in \mathcal{P}$ would represent a distinct character or character like object in the sense that $P(y|x) = 1$ if x is an image of the character P represents and $y = 1$, and $P(y|x) = 0$ otherwise. The marginal distribution $P(x)$ in each case could be formed by choosing a positive example of the character with probability half from the 'background' distribution over images of the character concerned, and similarly choosing a negative example with probability half. Q would give the probability of occurence of each character. The (n,m)-sample $\mathbf{z}$ is then simply a set of n training sets, each row of $\mathbf{z}$ being a sequence of m classified examples of some character. If the idea of character is widened to include other alphabets such as the greek alphabet and the Japanese Kanji characters then the number of different characters to sample from is very large indeed.

To enable the learner to take advantage of the prior information contained in the (n,m)-sample $\mathbf{z}$, the hypothesis space $\mathcal{H}\colon X \to A$ is split into two sections: $\mathcal{H} = \mathcal{G} \circ \mathcal{F}$ where $\mathcal{F}\colon X \to V$ and $\mathcal{G}\colon V \to A$, where V is an arbitrary set[1]. To simplify the notation this will be written in future as

$$X \xrightarrow{\mathcal{F}} V \xrightarrow{\mathcal{G}} A.$$

$\mathcal{F}$ will be called the *representation space* and an individual member f of $\mathcal{F}$ will be called an *internal representation* or just a *representation*. $\mathcal{G}$ will be called the *output function space*.

[1]That is, $\mathcal{H} = \{g \circ f \colon g \in \mathcal{G}, f \in \mathcal{F}\}$.

Based on the information about the environment Q, contained in $\mathbf{z}$, the learner searches for a good representation $f \in \mathcal{F}$. A good representation is one with a small *empirical loss* $E^*_{\mathcal{G}}(f, \mathbf{z})$ on $\mathbf{z}$, where this is defined by

$$E^*_{\mathcal{G}}(f, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} \inf_{g \in \mathcal{G}} \mathbb{E} l_{g \circ f}\, \vec{z}_i, \tag{3}$$

where $\vec{z}_i = (z_{i1}, \dots, z_{im})$ denotes the ith row of $\mathbf{z}$. The empirical loss of f with respect to $\mathbf{z} \in Z^{(n,m)}$ is a measure of how well the learner can learn $\mathbf{z}$ using f, assuming that the learner is able to find the best possible $g \in \mathcal{G}$ for any given sample $\vec{z} \in Z^m$. For example, if the empirical loss of f on $\mathbf{z} = (\vec{z}_1, \dots, \vec{z}_n)$ is zero then it is possible for the learner to find an output function[2] $g_i \in \mathcal{G}$, for each i, $1 \le i \le n$, such that the ordinary empirical loss $\mathbb{E} l_{g_i \circ f} \vec{z}_i$ is zero. The empirical loss of f is an estimate of the *true loss* of f (with respect to Q), where this is defined by

$$E^*_{\mathcal{G}}(f, Q) = \int_{\mathcal{P}} \inf_{g \in \mathcal{G}} \mathbb{E} l_{g \circ f}\, P \, dQ(P). \tag{4}$$

The true loss of f with respect to Q is the expected best possible performance of $g \circ f$—over all $g \in \mathcal{G}$—on a distribution chosen at random from $\mathcal{P}$ according to Q. If f has a small true loss then learning using f on a random "task" P—drawn according to Q—will with high probability be successful.

Note that the learner takes as input samples $\mathbf{z} \in Z^{(n,m)}$, for any values $n, m \ge 1$, and produces as output hypothesis representations $f \in \mathcal{F}$, so it is a map $\mathcal{A}$ from the space of all possible (n,m) samples into $\mathcal{F}$,

$$\mathcal{A}\colon \bigcup_{n,m \ge 1} Z^{(n,m)} \to \mathcal{F}.$$

It may be that the n tasks $\vec{P} = (P_1, \dots, P_n)$ generating the training set $\mathbf{z}$ are all that the learner is ever going to be required to learn, in which case it is more appropriate to suppose that in response to the (n,m)-sample $\mathbf{z}$ the learner generates n hypotheses $\vec{g} \circ \bar{f} = (g_1 \circ f, \dots, g_n \circ f)$ all using the same representation f and collectively minimizing

$$E(\vec{g} \circ \bar{f}, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E} l_{g_i \circ f} \vec{z}_i. \tag{5}$$

The true loss of the learner will then be

$$E(\vec{g} \circ \bar{f}, \vec{P}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E} l_{g_i \circ f} P_i. \tag{6}$$

[2]Assuming the infimum is attained in $\mathcal{G}$.

Denoting the set of all functions $\vec{g} \circ \bar{f} = (g_1 \circ f, \ldots, g_n \circ f)$ for $g_1, \ldots, g_n \in \mathcal{G}$ and $f \in \mathcal{F}$ by $\mathcal{G}^n \circ \overline{\mathcal{F}}$, the learner in this case is a map

$$\mathcal{A} \colon \bigcup_{n,m \geq 1} Z^{(n,m)} \to \mathcal{G}^n \circ \overline{\mathcal{F}}.$$

If the learner is going to be using the representation f to learn future tasks drawn according to the same environment Q, it will do so by using the *restricted* hypothesis space $\mathcal{G} \circ f = \{g \circ f \colon g \in \mathcal{G}\}$. That is, the learner will be fed samples $\vec{z} \in Z^m$ drawn according to some distribution $P \in \mathcal{P}$, which in turn is drawn according to Q, and will search $\mathcal{G} \circ f$ for a hypothesis $g \circ f$ with small empirical loss on $\vec{z}$. Intuitively, if $\mathcal{F}$ is much "bigger" than $\mathcal{G}$ then the number of examples required to learn using $\mathcal{G} \circ f$ will be much less than the number of examples required to learn using the full space $\mathcal{G} \circ \mathcal{F}$, a fact that is proved in the next section. Hence, if the learner can find a good representation f and the sample $\mathbf{z}$ is large enough, learning using $\mathcal{G} \circ f$ will be considerably quicker and more reliable than learning using $\mathcal{G} \circ \mathcal{F}$. If the representation mechanism outlined here is the one employed by our brains then the fact that children learn to recognise characters and words from a relatively small number of examples is evidence of a small $\mathcal{G}$ in these cases. The fact that we are able to recognise human faces after being shown only a single example is evidence of an even smaller $\mathcal{G}$ for this task. Furthermore, the fact that most of the difficulty in machine learning lies in the initial bias of the learner's hypothesis space [2] indicates that our ignorance concerning an appropriate representation f is large, and hence the entire representation space $\mathcal{F}$ will have to be large to ensure that it contains a suitable representation. Thus it seems that at least for the examples outlined above the conditions ensuring that representation learning is a big improvement over ordinary learning will be satisfied.

The main issue in machine learning is that of quantifying the necessary sampling conditions ensuring good generalisation. In representation learning there are two measures of good generalisation. The first is the proximity of (5) above to the second form of the true loss (6). If the sample $\mathbf{z}$ is large enough to guarantee with high probability that these two quantities are close, then a learner that produces a good performance in training on $\mathbf{z}$ will be likely to perform well on future examples of any of the n tasks used to generate $\mathbf{z}$. The second measure of generalisation performance is the proximity of (3) to the first form of the true loss (4). In this case good generalisation means that the learner should expect to perform well if it uses the representation f to learn a new task P drawn at random according to the environmental measure Q. Note that this is a new form of generalisation, one level of abstraction higher than the usual meaning of generalisation, for within this framework a learner generalises well if, after having learnt many different tasks, it is able to learn new tasks easily. Thus, not only is the learner required to generalise well in the ordinary sense by generalising well on the tasks in the training set, but also the learner is expected to have "learnt to learn" the tasks from the environment in general. Both the number of tasks n generating $\mathbf{z}$ and the number of examples m of each task must be sufficiently large to ensure good generalisation in this new sense.

To measure the deviation between (5) and (6), and (3) and (4), the following one-parameter family of metrics on $\mathbb{R}^+$, introduced in [3], will be used:

$$d_\nu(x, y) = \frac{|x - y|}{\nu + x + y},$$

for all $\nu > 0$ and $x, y \in \mathbb{R}^+$. Thus, good generalisation in the first case is governed by the probability

$$\Pr\left\{\mathbf{z} \in Z^{(n,m)} : d_\nu\left(E(\mathcal{A}(\mathbf{z}), \mathbf{z}), E(\mathcal{A}(\mathbf{z}), \vec{P})\right) > \alpha\right\}, \tag{7}$$

where the probability measure on $Z^{(n,m)}$ is $P_1^m \times \cdots \times P_n^m$. In the second case it is governed by the probability

$$\Pr\left\{\mathbf{z} \in Z^{(n,m)} : d_\nu\left(E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), \mathbf{z}), E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), Q)\right) > \alpha\right\}. \tag{8}$$

This time the probability measure on $Z^{(n,m)}$ is

$$\mu(S) = \int_{\mathcal{P}^n} P_1^m \times \cdots \times P_n^m(S) \, dQ^n(P_1, \ldots, P_n)$$

for any measurable subset[3] S of $Z^{(n,m)}$.

Conditions on the sample $\mathbf{z}$ ensuring the probabilities (7) and (8) are small are derived in the next section.

3 Conditions for good generalisation

To state the main results some further definitions are required.

Definition 1 For the structure $X \xrightarrow{\mathcal{F}} V \xrightarrow{\mathcal{G}} A$ and loss function $l \colon Y \times A \to [0, M]$ define $l_g \colon V \times Y \to [0, M]$ for any $g \in \mathcal{G}$ by $l_g(v, y) = l(y, g(v))$. Let $l_{\mathcal{G}} = \{l_g \colon g \in \mathcal{G}\}$. For any probability measure P on $V \times Y$ define the pseudo-metric d_P on $l_{\mathcal{G}}$ by

$$d_P(l_g, l_{g'}) = \int_{V \times Y} |l_g(v, y) - l_{g'}(v, y)| \, dP(v, y). \tag{9}$$

[3]The minimal σ-algebra for Z is $\sigma_{l_{\mathcal{G}\circ\mathcal{F}}}$, as defined in definition 3.

Let $\mathcal{N}(\varepsilon, l_{\mathcal{G}}, d_P)$ be the size of the smallest ε-cover of $(l_{\mathcal{G}}, d_P)$ and define the *ε-capacity* of $l_{\mathcal{G}}$ to be

$$\mathcal{C}(\varepsilon, l_{\mathcal{G}}) = \sup \mathcal{N}(\varepsilon, l_{\mathcal{G}}, d_P) \tag{10}$$

where the supremum is over all probability measures P. For any probability measure P on Z define the pseudo-metric $d^*_{[P,l_{\mathcal{G}}]}$ on $\mathcal{F}$ by[4]

$$d^*_{[P,l_{\mathcal{G}}]}(f, f') = \int_Z \sup_{g \in \mathcal{G}} |l_{g \circ f}(z) - l_{g \circ f'}(z)| \, dP(z).$$

Let $\mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon, \mathcal{F})$ be the corresponding ε-capacity.

3.1 Generalisation on n tasks

The following theorem bounds the number of examples m of each task in an (n, m)-sample $\mathbf{z}$ needed to ensure with high probability good generalisation from a representation learner on average on all the tasks. It uses the notion of a *hypothesis space family*—which is just a set of hypothesis spaces—and a generalisation of Pollard's concept of *permissibility* [4] to cover hypothesis space families, called *f-permissibility*. The definition of f-permissibility is given in appendix B.

Theorem 1 *Let $\mathcal{F}$ and $\mathcal{G}$ be families of functions with the structure $X \xrightarrow{\mathcal{F}} V \xrightarrow{\mathcal{G}} A$, let l be a loss function $l\colon Y \times A \to [0, M]$ and suppose $\mathcal{F}, \mathcal{G}$ and l are such that the hypothesis space family $\{l_{\mathcal{G} \circ f}\colon f \in \mathcal{F}\}$ is f-permissible. Let $P_1, \ldots, P_n$ be n probability measures on $Z = X \times Y$ and let $\mathbf{z} \in Z^{(n,m)}$ be an (n, m)-sample generated by sampling m times from Z according to each P_i. For all $0 < \alpha < 1, 0 < \delta < 1, \nu > 0$ and any representation learner*

$$\mathcal{A}\colon \bigcup_{n,m \geq 1} Z^{(n,m)} \to \mathcal{G}^n \circ \overline{\mathcal{F}},$$

if

$$m \geq \frac{8M}{\alpha^2 \nu} \left[\ln \mathcal{C}(\varepsilon_1, l_{\mathcal{G}}) + \frac{1}{n} \ln \frac{4\mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon_2, \mathcal{F})}{\delta} \right] \tag{11}$$

where $\varepsilon_1 + \varepsilon_2 = \frac{\alpha\nu}{8}$, then

$$\Pr\left\{ \mathbf{z} \in Z^{(n,m)}\colon d_\nu \left(E(\mathcal{A}(\mathbf{z}), \mathbf{z}), E(\mathcal{A}(\mathbf{z}), \vec{P}) \right) > \alpha \right\} \leq \delta.$$

Proof. See appendix A. □

Theorem 1 with $n = 1$ corresponds to the ordinary learning scenario in which a single task is learnt. Setting $a = \frac{8M}{\alpha^2\nu} \ln \mathcal{C}(\varepsilon_1, l_{\mathcal{G}})$ and $b = \frac{8M}{\alpha^2\nu} \ln \frac{4\mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon_2, \mathcal{F})}{\delta}$ gives $m = O(a+b)$ for a single task while $m = O(a+b/n)$ for n tasks learnt using a common representation[5]. Note also that if the n tasks are learnt independently then the learner is a map from the space of all (n, m)-samples into $\mathcal{G}^n \circ \mathcal{F}^n$ rather than $\mathcal{G}^n \circ \overline{\mathcal{F}}$ and so (not surprisingly) the number of examples m per task required will be the same as in the single task case: $m = O(a + b)$. Thus for hypothesis spaces in which $b \gg a$ learning n tasks using a representation will be far easier than learning them independently.

If $\mathcal{F}$ and $\mathcal{G}$ are Lipschitz bounded neural networks with $W_{\mathcal{F}}$ weights in $\mathcal{F}$ and $W_{\mathcal{G}}$ weights in $\mathcal{G}$ and l is one of many loss functions used in practice (such as Euclidean loss, mean squared loss, cross entropy loss—see [3], section 7), then simple extensions of the methods of [3] can be used to show:

$$\mathcal{C}(\varepsilon, l_{\mathcal{G}}) \leq \left[\frac{k}{\varepsilon}\right]^{2W_{\mathcal{G}}}$$
$$\mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon, \mathcal{F}) \leq \left[\frac{k'}{\varepsilon}\right]^{2W_{\mathcal{F}}}$$

where k and k' are constants not dependent on the number of weights or ε. Substituting these expressions into (11) and optimizing the bound with respect to $\varepsilon_1 + \varepsilon_2 = \alpha\nu/8$ gives:

$$\varepsilon_1 = \frac{\min(W_{\mathcal{F}}, nW_{\mathcal{G}})}{W_{\mathcal{F}} + nW_G} \frac{\alpha\nu}{8}$$
$$\varepsilon_2 = \frac{\max(W_{\mathcal{F}}, nW_{\mathcal{G}})}{W_{\mathcal{F}} + nW_{\mathcal{G}}} \frac{\alpha\nu}{8}$$

which yields[6]

$$m = O\left(\frac{1}{\alpha^2\nu}\left[\ln \frac{1}{\alpha\nu}\left(W_{\mathcal{G}} + \frac{W_{\mathcal{F}}}{n}\right) + \frac{1}{n} \ln \frac{1}{\delta}\right]\right). \tag{12}$$

Although (12) results from a worst-case analysis and some rather crude approximations to the capacities of $\mathcal{F}$ and $\mathcal{G}$, its general *form* is intuitively very appealing—particularly the behaviour of m as a function of n and the size of $\mathcal{F}$ and $\mathcal{G}$. I would expect this general behaviour to survive in more accurate analyses of specific representation learning scenarios. This conclusion is certainly supported by the experiments of section 5.

[4]For $d^*_{[P,l_{\mathcal{G}}]}$ to be well defined the supremum over $\mathcal{G}$ must be P-measurable. This is guaranteed if the hypothesis space *family* $\{l_{\mathcal{G} \circ f}\colon f \in \mathcal{F}\}$ is *f-permissible* (see appendix B).

[5]The choice of ε_1 and ε_1 subject to $\varepsilon_1 + \varepsilon_2 = \alpha\nu/8$ giving the best bound on m will differ between the $n = 1$ and $n > 1$ cases, so strictly speaking *at worst* $m = O(a + b/n)$ in the latter case.

[6]Bound (12) does not contradict known results on the VC dimension of *threshold* neural networks of $W \log W$ because it only applies to Lipschitz bounded neural networks, and the Lipschitz bounds are part of the bound on m (but they are not shown).

3.2 Generalisation when learning to learn

The following theorem bounds the number of tasks n and the number of examples m of each task required of an (n,m)-sample $\mathbf{z}$ to ensure with high probability that a representation learnt on that sample will be good for learning future tasks drawn according to the same environment.

Theorem 2 *Let $\mathcal{F}, \mathcal{G}$ and l be as in theorem 1. Let $\mathbf{z} \in Z^{(n,m)}$ be an (n,m)-sample from Z according to an environmental measure Q. For all $0 < \alpha, \delta, \varepsilon_1, \varepsilon_2 < 1$, $\nu > 0$, $\varepsilon_1 + \varepsilon_2 = \frac{\alpha\nu}{16}$, if*

$$n \geq \frac{32M}{\alpha^2\nu} \ln \frac{8\mathcal{C}^*_{l_\mathcal{G}}\left(\frac{\alpha\nu}{16}, \mathcal{F}\right)}{\delta},$$

$$\text{and} \quad m \geq \frac{32M}{\alpha^2\nu}\left[\ln \mathcal{C}\left(\varepsilon_1, l_\mathcal{G}\right) + \frac{1}{n}\ln \frac{8\mathcal{C}^*_{l_\mathcal{G}}\left(\varepsilon_2, \mathcal{F}\right)}{\delta}\right],$$

and $\mathcal{A}$ is any representation learner,

$$\mathcal{A}\colon \bigcup_{n,m\geq 1} Z^{(n,m)} \to \mathcal{F},$$

then

$$\Pr\left\{\mathbf{z} \in Z^{(n,m)} \colon d_\nu\left(E^*_\mathcal{G}(\mathcal{A}(\mathbf{z}), \mathbf{z}), E^*_\mathcal{G}(\mathcal{A}(\mathbf{z}), Q)\right) > \alpha\right\} \leq \delta.$$

Proof. See appendix A. □

Apart from the odd factor of two, the bound on m in this theorem is the same as the bound in theorem 1. Using the notation introduced following theorem 1, the bound on n is $n = O(b)$, which is very large. However it results again from a worst-case analysis (it corresponds to the worst possible environment Q) and is only an approximation, so it is likely to be beaten in practice. The experimental results of section 5 verify this. The bound on m now becomes $m = O(a)$, while the *total* number of examples $mn = O(ab)$. This is far greater than the $O(a+b)$ examples that would be required to learn a single task using the full space $\mathcal{G} \circ \mathcal{F}$, however a representation learnt on a single task cannot be used to reliably learn novel tasks. Also the representation learning phase is most likely to be an *off-line* process to generate the preliminary bias for later *on-line* learning of novel tasks, and as learning of novel tasks will be done with the biased hypothesis space $\mathcal{G} \circ f$ rather than the full space $\mathcal{G} \circ \mathcal{F}$, the number of examples required for good generalisation on novel tasks will be reduced to $O(a)$.

4 Representation learning via backpropagation

If the function classes $\mathcal{F}$ and $\mathcal{G}$ consist of "backpropagation" type neural networks then it is possible to use a slight variation on ordinary gradient descent procedures to learn a neural network representation f.

In this case, based on an (n,m)-sample

$$\mathbf{z} = \begin{matrix} (x_{11}, y_{11}) & (x_{12}, y_{12}) & \cdots & (x_{1m}, y_{1m}) \\ (x_{21}, y_{21}) & (x_{22}, y_{22}) & \cdots & (x_{2m}, y_{2m}) \\ \vdots & \vdots & \ddots & \vdots \\ (x_{n1}, y_{n1}) & (x_{n2}, y_{n2}) & \cdots & (x_{nm}, y_{nm}), \end{matrix}$$

the learner searches for a representation $f \in \mathcal{F}$ minimizing the *mean-squared* representation error:

$$E^*_\mathcal{G}(f, \mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} \inf_{g\in\mathcal{G}} \frac{1}{m}\sum_{j=1}^{m}\left(g \circ f(x_{ij}) - y_{ij}\right)^2.$$

The most common procedure for training differentiable neural networks is to use some form of gradient descent algorithm (vanilla backprop, conjugate gradient, etc) to minimize the error of the network on the sample being learnt. For example, in ordinary learning the learner would receive a single sample $\vec{z} = ((x_1, y_1), \ldots, (x_m, y_m))$ and would perform some form of gradient descent to find a function $g \circ f \in \mathcal{G} \circ \mathcal{F}$ such that

$$E(g \circ f, \vec{z}) = \frac{1}{m}\sum_{i=1}^{m}\left(g \circ f(x_i) - y_i\right)^2 \tag{13}$$

is minimal. This procedure works because it is a relatively simple matter to compute the gradient, $\nabla_\mathbf{w} E(g \circ f, \mathbf{z})$, where $\mathbf{w}$ are the parameters (weights) of the networks in $\mathcal{F}$ and $\mathcal{G}$.

Applying this method directly to the problem of minimising $E^*_\mathcal{G}(f, \mathbf{z})$ above would mean calculating the gradient $\nabla_\mathbf{w} E^*_\mathcal{G}(f, \mathbf{z})$ where now $\mathbf{w}$ refers only to the parameters of $\mathcal{F}$. However, due to the presence of the infimum over $\mathcal{G}$ in the formula for $E^*_\mathcal{G}(f, \mathbf{z})$, calculating the gradient in this case is much more difficult than in the ordinary learning scenario. An easier way to proceed is to minimize $E(\vec{g} \circ \bar{f}, \mathbf{z})$ (recall equation (5)) over all $\vec{g} \circ \bar{f}$, for if $\vec{g} \circ \bar{f}$ is such that

$$\left|E(\vec{g} \circ \bar{f}, \mathbf{z}) - \inf_{\vec{g}\circ\bar{f}\in\mathcal{G}^n\circ\overline{\mathcal{F}}} E(\vec{g} \circ \bar{f}, \mathbf{z})\right| \leq \varepsilon,$$

then so too is

$$\left|E^*_\mathcal{G}(f, \mathbf{z}) - \inf_{f\in\mathcal{F}} E^*_\mathcal{G}(f, \mathbf{z})\right| \leq \varepsilon.$$

The advantage of this approach is that essentially the same techniques used for computing the gradient in ordinary learning can be used to compute the gradient of $E(\vec{g} \circ \bar{f}, \mathbf{z})$. Note that in the present framework $E(\vec{g} \circ \bar{f}, \mathbf{z})$ is:

$$E(\vec{g} \circ \bar{f}, \mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\left(g_i \circ f(x_{ij}) - y_{ij}\right)^2, \tag{14}$$

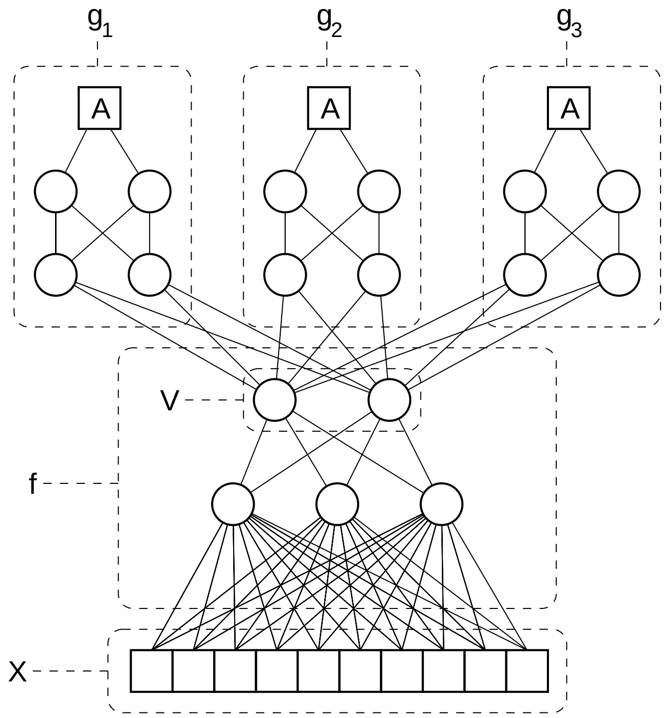

Figure 1: A neural network for representation learning.

which is simply the average of the mean-squared error of each $g_i \circ f$ on $\vec{z}_i$.

An example of a neural network of the form $\vec{g} \circ \bar{f}$ is illustrated in figure 1. With reference to the figure, consider the problem of computing the derivative of $E(\vec{g} \circ \bar{f}, \mathbf{z})$ with respect to a weight in the i^{th} output network g_i. Denoting the weight by w_i and recalling equation (14), we have

$$\frac{\partial}{\partial w_i} E(\vec{g} \circ \bar{f}, \mathbf{z}) = \frac{1}{n} \frac{\partial}{\partial w_i} \frac{1}{m} \sum_{j=1}^{m} (g_i \circ f(x_{ij}) - y_{ij})^2 \tag{15}$$

which is just $1/n$ times the derivative of the ordinary learning error (13) of $g_i \circ f$ on sample $\vec{z}_i = ((x_{i1}, y_{i1}), \ldots, (x_{im}, y_{im}))$ with respect to the weight w_i. This can be computed using the standard backpropagation formula for the derivative [5]. Alternatively, if w is a weight in the representation network f then

$$\frac{\partial}{\partial w} E(\vec{g} \circ \bar{f}, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w} \frac{1}{m} \sum_{j=1}^{m} (g_i \circ f(x_{ij}) - y_{ij})^2 \tag{16}$$

which is simply the *average* of the derivatives of the ordinary learning errors over all the samples $(\vec{z}_1, \ldots, \vec{z}_n)$ and hence can also be computed using the backpropagation formula.

5 Experiment: Learning translation invariance

In this section the results of an experiment are reported in which a neural network like the one in figure 1 was trained to perform a very simple "machine vision" task where it had to learn certain *translationally invariant* Boolean functions. All simulations reported here were performed on the 32 node CM5 at The South Australian centre for Parallel Supercomputing.

The input space X was viewed as a one-dimensional "retina" in which all the pixels could be either on (1) or off (0) (so in fact $X = \{0,1\}^{10}$). However the network did not see all possible input vectors during the course of its training, the only vectors with a non-zero probability of appearing in the training set were those consisting of from one to four active adjacent pixels placed somewhere in the input (wrapping at the edge was allowed).

The functions in the environment of the network consisted of all possible *translationally invariant* Boolean functions over the input space (except the trivial "constant 0" and "constant 1" functions). The requirement of translation invariance means that the environment consisted of just 14 different functions—all the Boolean functions on four objects (of which there are $2^4 = 16$) less the two trivial ones. Thus the environment was highly restricted, both in the number of different input vectors seen—40 out of a possible 1024—and in the number of different functions to be learnt—14 out of a possible 2^{1024}. (n, m) samples were generated from this environment by firstly choosing n functions (with replacement) uniformly from the fourteen possible, and then choosing m input vectors (with replacement again), for each function, uniformly from the 40 possible input vectors.

The architecture of the network was similar to the one shown in figure 1, the only difference being that the output networks $g \in \mathcal{G}$ for this experiment had only one hidden layer, not two. The network in figure 1 is for learning $(3, m)$ samples (it has 3 output networks), in general for learning (n, m) samples the network will have n output networks.

The network was trained on (n, m) samples with n ranging from 1 to 21 in steps of four and m ranging from 1 to 151 in steps of 10. Conjugate-gradient descent was used with exact line search with the gradients for each weight computed using the backpropagation algorithm according to the formulae (16) and (15). Further details of the experimental procedure may be found in [1].

Once the network had sucessfully learnt the (n, m) sample its generalization ability was tested on all n functions in the training set. In this case the generalisation error (i.e *true* error—$E(\mathcal{A}(\mathbf{z}), \vec{P})$) could be computed

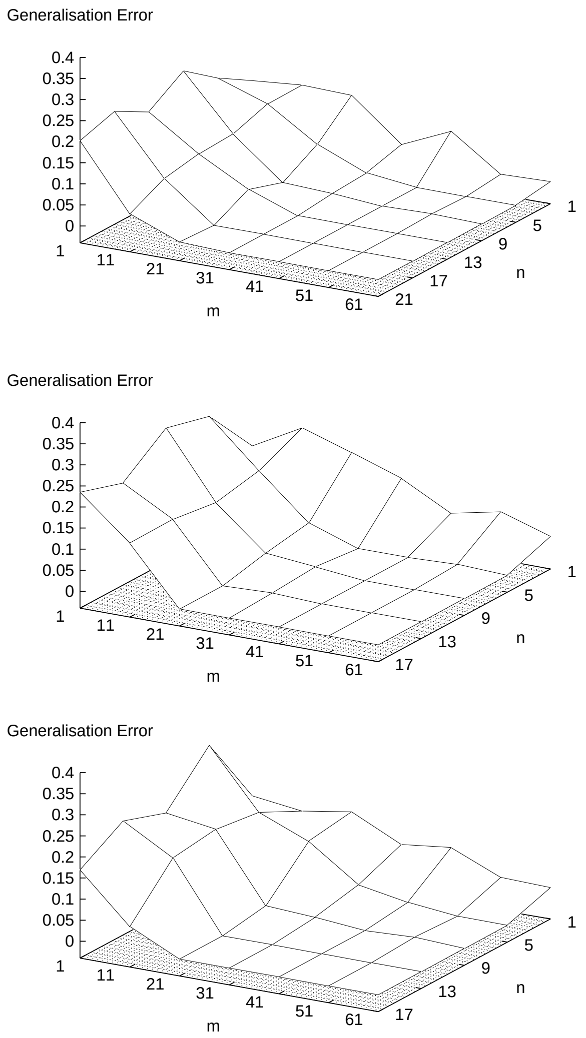

Figure 2: Learning *surface* for three independent simulations.

exactly by calculating the network's output (for all n functions) for each of the 40 input vectors, and comparing the result with the desired output.

In an ordinary learning situtation the generalisation error of a network would be plotted as a function of m, the number of examples in the training set, resulting in a *learning curve*. For representation learning there are two parameters m and n so the learning curve becomes a *learning surface*. Plots of the learning surface are shown in figure 2 for three independent simulations. All three cases support the theoretical result that the number of examples m required for good generalisation decreases with increasing n (*cf* theorem 1).

For (n, m) samples that led to a generalisation error of less than 0.01, the representation network f was extracted and tested for its *true error*, where this is defined as in equation (4) and in the current framework translates to

$$E^*_{\mathcal{G}}(f, Q) = \frac{1}{560} \sum_{i=1}^{14} \inf_{g \in \mathcal{G}} \sum_{j=1}^{40} (g \circ f(x_j) - f_i(x_j))^2$$

where $x_1, \ldots, x_{40}$ are the 40 input vectors seen by the learner and $f_1, \ldots, f_{14}$ are all the functions in the environment. $E^*_{\mathcal{G}}(f, Q)$ measures how useful the representation f is for learning all functions in the environment, not just the ones used in generating the (n, m) sample f was trained on. To measure $E^*_{\mathcal{G}}(f, Q)$, entire training sets consisting of 40 input-output pairs were generated for each of the 14 functions in the environment and the training sets were learnt individually by fixing f and performing conjugate gradient descent on the weights of g. To be (nearly) certain that a minimal solution had been found for each of the functions, learning was started from 32 different random weight initialisations for g (this number was chosen so that the CM5 could perform all the restarts in parallel) and the best result from all 32 recorded. For each (n, m) sample giving perfect generalisation, $E^*_{\mathcal{G}}(f, Q)$ was calculated and then averaged over all (n, m) samples with the same value of n, and finally averaged over all three simulations, to give an indication of the behaviour of $E^*_{\mathcal{G}}(f, Q)$ as a function of n. This is plotted in figure 3, along with the L^∞ representation error for the three simulations (i.e, the maximum error over all 14 functions and all 40 examples and over all three simulations). Qualitatively the curves support the theoretical conclusion that the representation error should decrease with an increasing number of tasks n in the (n, m) sample. However, note that the representation error is very small, even when the representation is derived from learning only one function from the environment. This can be explained as follows. For a representation to be a good one for learning in this environment it must be translationally invariant and distinguish all the four objects it sees (i.e. have different values on all four objects). For small values of n, to achieve perfect generalisation the representation is forced to be at least approximately translationally invariant and so half of the problem is already solved. However depending upon the particular functions in the (n, m) sample the representation may not have to distinguish all four objects, for example it may map two objects to the same element of V if none of the functions in the sample distinguish those objects (a function distinguishes two objects if it has a different value on those two objects). However, because the representation network is continuous it is very unlikely that it will map different objects to *exactly* the same element of V—there will always be slight differences. When the representation is used to learn a function that does distinguish the objects mapped to nearly the same element of V, often an output network g with sufficiently large weights can be found to amplify this difference and produce a function with small error. This is why a representation that is simply translationally invariant does quite well in general. This argument is supported by a plot in figure 4 of the representation's behaviour for minimum-sized samples leading to a generalisation error of less than 0.01 for $n = 1, 5, 13, 17$. The four different symbols marking the points in the plots correspond to the four different input objects. For the $(1, 111)$ plot the three and four pixel objects are well separated by the representation while the one and two pixel objects are not, except that a closer look reveals that there is a slight separation between the representation's output for the one and two pixel objects. This separation can be exploited to learn a function that at least partially distinguishes the two objects. Note that the representation's behaviour improves dramatically with increasing n, in that all four objects become well separated and the variation in the representation's output for individual objects decreases. This improvement manifests itself in superior *learning curves* for learning using a representation from a high n (n, m)-sample, although it is not necessarily reflected in the representation's error because of the use of the infimum over all $g \in \mathcal{G}$ in the definition of that error.

5.1 Representation vs. no representation.

As well as reducing the sampling burden for the n tasks in the training set, a representation learnt on sufficiently many tasks should be good for learning novel tasks and should greatly reduce the number of examples required of new tasks. This was experimentally verified by taking a representation f known to be perfect for the environment above and using it to learn all the functions in the environment in turn. Hence the hypothesis space of the learner was $\mathcal{G} \circ f$, rather than the full space $\mathcal{G} \circ \mathcal{F}$. All the functions in the environment were also learnt using the full space. The learning curves (i.e. the generalisation error as a function of the number of examples in the training set) were calculated for all 14 functions in each case. The learning curves for all the functions were very similar and two are plotted in figure 5, for learning with a representation (Gof in the graphs) and without (GoF). These curves are the average of 32 different simulations obtained by using different random starting points for the weights in g (and f when using the full space to learn). Learning with a good representation is clearly far superior to learning without.

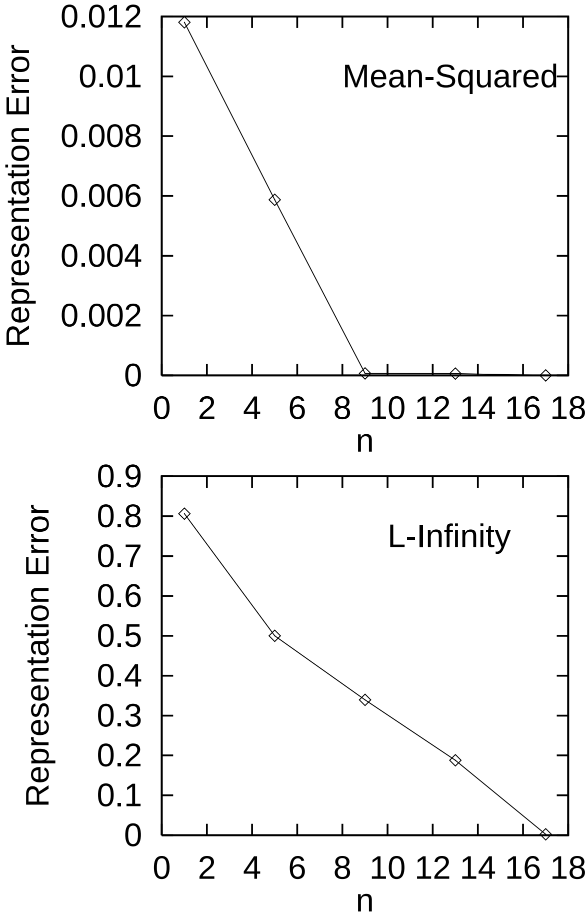

Figure 3: Mean-Squared and L^∞ representation error curves.

A Sketch proofs of theorems 1 and 2

Definition 2 Let $\mathcal{H}_1, \ldots, \mathcal{H}_n$ be n sets of functions mapping Z into $[0, M]$. For any $h_1 \in \mathcal{H}_1, \ldots, h_n \in \mathcal{H}_n$, let $h_1 \oplus \cdots \oplus h_n$ or simply $\vec{h}$ denote the map $\vec{z} \mapsto 1/n \sum_{i=1}^{n} h_i(z_i)$ for all $\vec{z} = (z_1, \ldots, z_n) \in Z^n$. Let

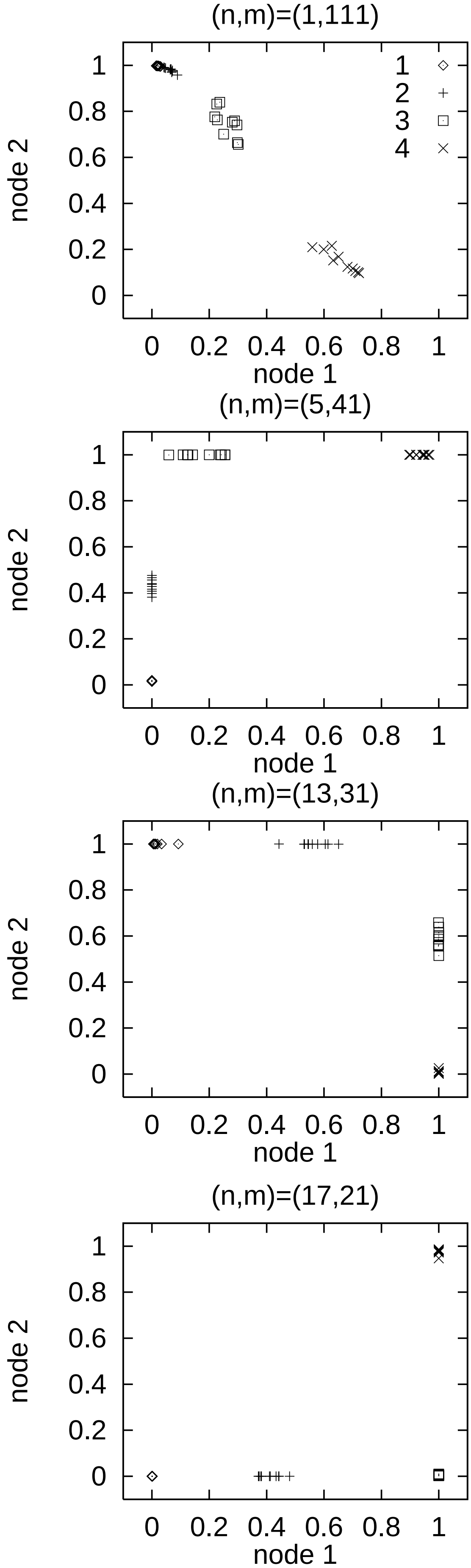

Figure 4: Plots of the output of a representation generated from the indicated (n, m) sample.

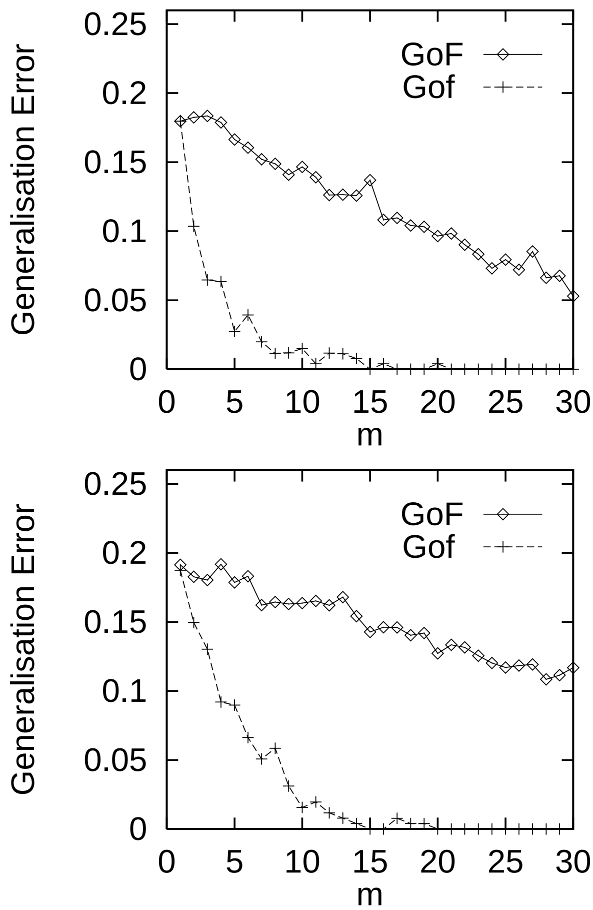

Figure 5: Two typical learning curves for learning with a representation (Gof) vs. learning without (GoF).

$\mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_n$ denote the set of all such functions. Given m elements of Z^n, or equivalently an element of $Z^{(m,n)}$, $\mathbf{z} = (\vec{z}_1, \ldots, \vec{z}_m)$, let $\mathbb{E}\vec{h}\mathbf{z} = 1/m \sum_{i=1}^m \vec{h}(\vec{z}_i)$. For any product probebility measure $\vec{P} = P_1 \times \cdots \times P_n$ on Z^n let $\mathbb{E}\vec{h}\vec{P} = \int_{Z^n} \vec{h}(\vec{z})\, d\vec{P}(\vec{z})$. For any set $\mathcal{H} \subseteq \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_n$ define the pseudo-metric $d_{\vec{P}}$ and ε-capacity $\mathcal{C}(\varepsilon, \mathcal{H})$ as in (9) and (10).

The following lemma is a generalisation of a similar result (theorem 3) for ordinary learning in [3], which is in turn derived from results in [4]. It is proved in [1] where it is called *the fundamental theorem*. The definition of permissibility is given in appendix B.

Lemma 3 *Let $\mathcal{H} \subseteq \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_n$ be a permissible set of functions mapping Z^n into $[0, M]$. Let $\mathbf{z} \in Z^{(m,n)}$ be generated by $m \geq \frac{2M}{\alpha^2 \nu}$ independent trials from Z^n according to some product probability measure $\vec{P} = P_1 \times \cdots \times P_n$. For all $\nu > 0$, $0 < \alpha < 1$,*

$$\Pr\left\{\mathbf{z} \in Z^{(m,n)} \colon \exists \vec{h} \in \mathcal{H} \colon d_\nu\left(\mathbb{E}\vec{h}\mathbf{z}, \mathbb{E}\vec{h}\vec{P}\right) > \alpha\right\} \leq 4\mathcal{C}\left(\alpha\nu/8, \mathcal{H}\right) \exp\left(-\alpha^2 \nu n m / 8M\right).$$

A.1 Proof sketch of theorem 1

Let $l_{\mathcal{G}^n \circ \overline{\mathcal{F}}}$ denote the set of all functions $\vec{z} \mapsto 1/n \sum_{i=1}^n l_{g_i \circ f}(z_i)$ where $g_1, \ldots, g_n \in \mathcal{G}, f \in \mathcal{F}$ and $\vec{z} \in Z^n$, and let $l_{\vec{g} \circ \bar{f}}$ denote an individual such map. Recall that in theorem 1 the learner $\mathcal{A}$ maps (n, m)-samples into $\mathcal{G}^n \circ \overline{\mathcal{F}}$ and note from definition 2 and equation (5) that $E(\vec{g} \circ \bar{f}, \mathbf{z}) = \mathbb{E} l_{\vec{g} \circ \bar{f}} \mathbf{z}^T$ where T denotes matrix transposition. This gives,

$$\begin{aligned}
&\Pr\left\{\mathbf{z} \in Z^{(n,m)} \colon d_\nu\left(E(\mathcal{A}(\mathbf{z}), \mathbf{z}), E(\mathcal{A}(\mathbf{z}), \vec{P})\right) > \alpha\right\} \\
&\leq \Pr\left\{\mathbf{z} \in Z^{(n,m)} \colon \exists \vec{g} \circ \bar{f} \in \mathcal{G}^n \circ \overline{\mathcal{F}} \right. \\
&\qquad \left. \colon\ d_\nu\left(E(\vec{g} \circ \bar{f}, \mathbf{z}), E(\vec{g} \circ \bar{f}, \vec{P})\right) > \alpha\right\} \\
&= \Pr\left\{\mathbf{z} \in Z^{(m,n)} \colon \exists l_{\vec{g} \circ \bar{f}} \in l_{\mathcal{G}^n \circ \overline{\mathcal{F}}} \right. \\
&\qquad \left. \colon\ d_\nu\left(\mathbb{E} l_{\vec{g} \circ \bar{f}} \mathbf{z}, \mathbb{E} l_{\vec{g} \circ \bar{f}} \vec{P}\right) > \alpha\right\} \\
&\leq 4\mathcal{C}\left(\alpha\nu/8, l_{\mathcal{G}^n \circ \overline{\mathcal{F}}}\right) \exp\left(-\alpha^2 \nu n m / 8M\right).
\end{aligned}$$

The probability measure on $Z^{(n,m)}$ is $P_1^m \times \cdots \times P_n^m$ while on $Z^{(m,n)}$ it is $(P_1 \times \cdots \times P_n)^m$. With these measures the map $\mathbf{z} \mapsto \mathbf{z}^T$ is measure preserving, hence the equality above. The last inequality follows from lemma 3 noting that $l_{\mathcal{G}^n \circ \overline{\mathcal{F}}} \subseteq l_{\mathcal{G} \circ \mathcal{F}} \oplus \cdots \oplus l_{\mathcal{G} \circ \mathcal{F}}$ and that $l_{\mathcal{G}^n \circ \overline{\mathcal{F}}}$ is permissible by the assumption of f-permissibility of $\{l_{\mathcal{G} \circ f} \colon f \in \mathcal{F}\}$ and lemma 4 (permissibility of $[H^n]_\sigma$ in that lemma).

Recalling definition 1, it can be shown (see [1], appendix C) that

$$\mathcal{C}\left(\alpha\nu/8, l_{\mathcal{G}^n \circ \overline{\mathcal{F}}}\right) \leq \mathcal{C}(\varepsilon_1, l_{\mathcal{G}})^n \mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon_2, \mathcal{F})$$

where $\varepsilon_1 + \varepsilon_2 = \alpha\nu/8$. Substituting this into the expression above and setting the result less than δ gives theorem 1 □

A.2 Proof sketch of theorem 2

To prove theorem 2 note that in the (n, m)-sampling process a list of probability measures $\vec{P} = (P_1, \ldots, P_n)$ is implicitly generated in addition to the (n, m)-sample $\mathbf{z}$. Defining $E^*_{\mathcal{G}}(f, \vec{P}) = 1/n \sum_{i=1}^n \inf_{g \in \mathcal{G}} \mathbb{E} l_{g \circ f} P_i$ and using the triangle inequality on d_ν, if

$$\Pr\left\{(\mathbf{z}, \vec{P}) \colon d_\nu\left(E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), \mathbf{z}), E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), \vec{P})\right) > \frac{\alpha}{2}\right\} \leq \frac{\delta}{2}, \tag{17}$$

and

$$\Pr\left\{(\mathbf{z}, \vec{P}) \colon d_\nu\left(E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), \vec{P}), E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), Q)\right) > \frac{\alpha}{2}\right\} \leq \frac{\delta}{2}, \tag{18}$$

then

$$\Pr\left\{\mathbf{z} \colon d_\nu\left(E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), \mathbf{z}), E^*_{\mathcal{G}}(\mathcal{A}(\mathbf{z}), Q)\right) > \alpha\right\} \leq \delta.$$

Inequality (17) can be bounded using essentially the same techniques as theorem 1, giving

$$m \geq \frac{32M}{\alpha^2 \nu}\left[\ln \mathcal{C}\left(\varepsilon_1, l_{\mathcal{G}}\right) + \frac{1}{n} \ln \frac{8\mathcal{C}^*_{l_{\mathcal{G}}}\left(\varepsilon_2, \mathcal{F}\right)}{\delta}\right],$$

where $\varepsilon_1 + \varepsilon_2 = \frac{\alpha\nu}{16}$. Note that the probability in inequality (18) is less than or equal to

$$\Pr\left\{\vec{P} \colon \exists f \in \mathcal{F} \colon d_\nu\left(E^*_{\mathcal{G}}(f, \vec{P}), E^*_{\mathcal{G}}(f, Q)\right) > \frac{\alpha}{2}\right\} \tag{19}$$

Now, for each $f \in \mathcal{F}$ define $l^*_f \colon \mathcal{P} \to [0, M]$ by $l^*_f(P) = \inf_{g \in \mathcal{G}} \mathbb{E} l_{g \circ f} P$ and let $l^*_{\mathcal{F}}$ denote the set of all such functions. Note that the expectation of l^*_f with respect to $\vec{P} = (P_1, \ldots, P_n)$ satisfies $\mathbb{E} l^*_f \vec{P} = E^*_{\mathcal{G}}(f, \vec{P})$ and similarly $\mathbb{E} l^*_f Q = E^*_{\mathcal{G}}(f, Q)$. Hence (19) is equal to

$$\Pr\left\{\vec{P} \in \mathcal{P}^n \colon \exists l^*_f \in l^*_{\mathcal{F}} \colon d_\nu\left(\mathbb{E} l^*_f \vec{P}, \mathbb{E} l^*_f Q\right) > \frac{\alpha}{2}\right\}. \tag{20}$$

For any probability measure Q on $\mathcal{P}$ define the pseudo-metric d_Q on $l^*_{\mathcal{F}}$ by

$$d_Q(l^*_f, l^*_{f'}) = \int_{\mathcal{P}} |l^*_f(P) - l^*_{f'}(P)|\, dQ(P)$$

and let $C(\varepsilon, l^*_{\mathcal{F}})$ be the corresponding ε-capacity. Lemma 3 with $n = 1$ can now be used to show that (20) is less than or equal to $4C(\alpha\nu/16, l^*_{\mathcal{F}}) \exp(-\alpha^2 \nu n / 32M)$

(permissibility of $l^*_{\mathcal{F}}$ is guaranteed by f-permissibility of $\{l_{\mathcal{G}\circ f}: f \in \mathcal{F}\}$—see lemma 4 (permissibility of H^* in that lemma)). For any probability measure Q on $\mathcal{P}$, let Q_Z be the measure on Z defined by $Q_Z(S) = \int_{\mathcal{P}} P(S)\,dQ(P)$ for any S in the σ-algebra on Z. It is then possible to show that $d_Q(l^*_f, l^*_{f'}) \le d^*_{[Q_Z, l_{\mathcal{G}}]}(f, f')$ (recall definition 1) and so $\mathcal{C}(\varepsilon, l^*_{\mathcal{F}}) \le \mathcal{C}^*_{l_{\mathcal{G}}}(\varepsilon, \mathcal{F})$ which gives the bound on n in theorem 2 □

B F-permissibility and measurability

In this section all the measurability conditions required to ensure theorems 1 and 2 carry through are given. They are presented without proof—the interested reader is referred to [1] for the details.

Definition 3 Given any set of functions $\mathcal{H}: Z \to [0, M]$, where Z is any set, let $\sigma_{\mathcal{H}}$ be the σ-algebra on Z generated by all inverse images under functions in $\mathcal{H}$ of all open balls (under the usual topology on $\mathbb{R}$) in $[0, M]$. Let $\mathcal{P}_{\mathcal{H}}$ denote the set of all probability measures on $\sigma_{\mathcal{H}}$.

This definition ensures measurability of any function $h \in \mathcal{H}$ with respect to any measure $P \in \mathcal{P}_{\mathcal{H}}$. Use this definition anywhere in the rest of the paper where a set needs a σ-algebra or a probability measure

The following two definitions are taken (with minor modifications) from [4].

Definition 4 $\mathcal{H}: Z \to [0, M]$ is *indexed* by the set T if there exists a function $f: Z \times T \to [0, M]$ such that

$$\mathcal{H} = \{f(\,\cdot\,, t): t \in T\}.$$

Definition 5 $\mathcal{H}$ is *permissible* if it can be indexed by a set T such that T is an *analytic* subset of a Polish space $\overline{T}$, and the function $f: Z \times T \to [0, M]$ indexing $\mathcal{H}$ by T is measurable with respect to the product σ-algebra $\sigma_{\mathcal{H}} \otimes \sigma(T)$, where $\sigma(T)$ is the Borel σ-algebra induced by the topology on T.

For representation learning the concept of permissibility must be extended to cover *hypothesis space families*, that is, sets $H = \{\mathcal{H}\}$ where each $\mathcal{H} \in H$ is itself a set of functions mapping Z into $[0, M]$. Let $H_\sigma = \{h: h \in \mathcal{H}: \mathcal{H} \in H\}$.

Definition 6 H is *f-permissible* if there exist sets S and T that are analytic subsets of Polish spaces $\overline{S}$ and $\overline{T}$ respectively, and a function $f: Z \times T \times S \to [0, M]$, measurable with respect to $\sigma_{H_\sigma} \otimes \sigma(T) \otimes \sigma(S)$, such that

$$H = \{\{f(\,\cdot\,, t, s): t \in T\}: s \in S\}.$$

Definition 7 For any hypothesis space family H, define $H^n = \{\mathcal{H} \oplus \cdots \oplus \mathcal{H}: \mathcal{H} \in H\}$. For all $h \in H_\sigma$ define $\overline{h}: \mathcal{P}_{H_\sigma} \to [0, M]$ by $\overline{h}(P) = \mathbb{E}hP$. Let $\overline{H}_\sigma = \{\overline{h}: h \in H_\sigma\}$ and for all $\mathcal{H} \in H$ let $\overline{\mathcal{H}} = \{\overline{h}: h \in \mathcal{H}\}$. For all $\mathcal{H} \in H$ define $\mathcal{H}^*: \mathcal{P}_{H_\sigma} \to [0, M]$ by $\mathcal{H}^*(P) = \inf_{h \in \mathcal{H}} \overline{h}(P)$. Let $H^* = \{\mathcal{H}^*: \mathcal{H} \in H\}$. For any probability measure $Q \in \mathcal{P}_{\overline{H}_\sigma}$, define the probability measure Q_Z on Z by $Q_Z(S) = \int_{\mathcal{P}_{H_\sigma}} P(S)\,dQ(P)$, for all $S \in \sigma_{H_\sigma}$.

Note that if $H = \{l_{\mathcal{G}\circ f}: f \in \mathcal{F}\}$ then $H^* = l^*_{\mathcal{F}}$ and $[H^n]_\sigma = l_{\mathcal{G}^n \circ \mathcal{F}}$. The f-permissibility of a hypothesis space family H is important for the permissibility and measurability conditions it implies on $H^*, [H^n]_\sigma$, etc, as given in the following lemma. Nearly all the results in this lemma are needed to ensure theorems 1 and 2 hold.

Lemma 4 *Let H be a family of hypothesis spaces mapping Z into $[0, M]$. Take the σ-algebra on Z to be σ_{H_σ}, the set of probability measures on Z to be $\mathcal{P}_{H_\sigma}$ and the σ-algebra on $\mathcal{P}_{H_\sigma}$ to be $\sigma_{\overline{H}_\sigma}$. With these choices, if H is f-permissible then*

1. *H_σ, $[H^n]_\sigma$, $\overline{H}_\sigma$ and H^* are all permissible.*
2. *$\overline{\mathcal{H}}$ and $\mathcal{H}$ are permissible for all $\mathcal{H} \in H$.*
3. *H^n is f-permissible.*
4. *$\mathcal{H}^*$ is measurable for all $\mathcal{H} \in H$.*
5. *For all $Q \in \mathcal{P}_{\overline{H}_\sigma}$, $Q_Z \in \mathcal{P}_{H_\sigma}$.*

Acknowledgements

I would like to thank several anonymous referees for their helpful comments on the original version of this paper.

References

[1] J. Baxter. *Learning Internal Representations.* PhD thesis, Department of Mathematics and Statistics, The Flinders University of South Australia, 1995. Draft copy in Neuroprose Archive under "/pub/neuroprose/Thesis/baxter.thesis.ps.Z".

[2] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Comput.*, 4:1–58, 1992.

[3] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inform. Comput.*, 100:78–150, 1992.

[4] D. Pollard. *Convergence of Stochastic Processes.* Springer-Verlag, New York, 1984.

[5] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.