Lifelong Learning: A Case Study

Sebastian Thrun

November 1995

CMU-CS-95-208

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

The author is also affiliated with the Computer Science Department III of the University of Bonn, Germany, where part of this research was carried out.

This research is sponsored in part by the National Science Foundation under award IRI-9313367, and by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and the Advanced Research Projects Agency (ARPA) under grant number F33615-93-1-1330. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of NSF, Wright Laboratory or the United States Government.

Keywords: Artificial neural networks, bias, concept learning, knowledge transfer, lifelong learning, machine learning, object recognition, relevance, supervised learning

Abstract

Machine learning has not yet succeeded in the design of robust learning algorithms that generalize well from very small datasets. In contrast, humans often generalize correctly from only a single training example, even if the number of potentially relevant features is large. To do so, they successfully exploit knowledge acquired in previous learning tasks, to bias subsequent learning.

This paper investigates learning in a lifelong context. Lifelong learning addresses situations where a learner faces a stream of learning tasks. Such scenarios provide the opportunity for synergetic effects that arise if knowledge is transferred across multiple learning tasks. To study the utility of transfer, several approaches to lifelong learning are proposed and evaluated in an object recognition domain. It is shown that all these algorithms generalize consistently more accurately from scarce training data than comparable "single-task" approaches.

1 Introduction

Supervised learning (pattern classification and regression) is concerned with approximating unknown functions based on examples. More specifically, given a set of input-output tuples of an unknown function which might be distorted by noise, the goal of supervised learning is to construct a generalization of the data that minimizes the weighted prediction error on future data.

Since deducing the output of unseen, future data is impossible without making further assumptions [31, 68, 19, 73], every learning algorithm makes inherent assumptions concerning the nature of the data. These assumptions—often referred to as *hypothesis space*, *preferences*, or *prior*, and henceforth called *bias* [30]—enables an algorithm to favor one particular generalization over all others, hence to generalize. The choice of bias is crucial in machine learning, as it represents both the designer's knowledge and his/her ignorance about the domain. In some approaches, bias is obtained explicitly through the expertise of a human expert of the domain, communicated by symbolic if-then rules [33, 12, 65, 41, 40, 38]. In others, it arises from an uninformed set of equations, as is the case in neural network Back-Propagation [72, 71, 48] or inductive tree learning [45, 17, 22], to name two popular examples.

All these approaches have in common that the available data consists exclusively of input-output examples of the target function. While this framework facilitates the precise study and evaluation of machine learning approaches, it dismisses important aspects that are crucial for the way humans learn. One of the key aspects of human learning is the fact that they face a stream of learning problems over their entire lifetime. When learning a skill as complex as driving a car, for example, years of learning experience with basic motor skills, typical traffic patterns, communication, logical reasoning, language, and much more precede and influence this learning task. To date, virtually all approaches studied in machine learning are concerned with learning a single function based on a single data set only, isolated from a more general learning context.

Studying learning in a "lifelong" context provides the opportunity to transfer knowledge between learning tasks. For example, in [1, 2] psychological experiments are reported in which humans acquire complex language concepts based on a single training example. The learning problem studied there involves the distinction of relevant from irrelevant features to generalize the training example. It is shown that humans can spot relevant features very well, even if the number of potentially relevant features is huge and the target concept is rather complex. As argued in [1, 2], the ability to do so relies on previously learned knowledge, which had been acquired earlier in the lifetime of the tested subjects. Another recent study

[37] illustrates that humans employ very specific routines for the robust recognition of human faces, so that they are able to learn to recognize new faces from very few training examples. In these experiments, it is shown empirically that the recognition rate of faces in an upright position is significantly better than that of faces in an inverted position. As argued there and in [26], this finding provides evidence that humans can transfer knowledge for the recognition of faces across different face recognition tasks—unless the human visual system is genetically pre-biased to the recognition of upright human faces (in which case evolution learned a good strategy for us).

This paper studies machine learning algorithms that can transfer knowledge across multiple learning tasks. We are interested in situations where a learner faces a collection or learning tasks over its entire lifetime. If these tasks are appropriately related, such a lifelong learning problem provides the opportunity for synergy. When faced with the n-th learning task, there is the opportunity to transfer knowledge acquired in the previous $n-1$ learning tasks, to save data in the n-th one. In other words, the first $n-1$ learning tasks may be used to acquire a knowledgeable, domain-specific bias for the n-th learning task. The acquisition, representation and use of bias are therefore the key scientific issues that arise in the lifelong learning framework.

Instead of the general problem, this paper considers a restricted version of the lifelong learning problem. In particular, the following assumptions are made throughout the paper:

1. **Concept learning.** We assume that the learner only encounters concept learning (pattern classification) tasks, which are defined over a d-dimensional feature space. A concept learning task is a supervised learning task in which there are only two possible output values, 1 and 0. The k-th concept learning tasks (with $k = 1, \ldots, n$) involves learning a classification function $f^k : \Re^d \longrightarrow \{0, 1\}$ that maps patterns in $\Re^d$ to two classes, 1 and 0. The set of training data for the k-th learning tasks is denoted by

$$X^k \quad = \quad \{\langle x_i^k, y_i^k \rangle \mid i = 1 \ldots N^k\}. \tag{1}$$

   Here x_i^k denotes the i-th input pattern in X^k, y_i^k the corresponding class label, and N^k the cardinality of the training set. A pattern x is member of the k-th concept, if and only if $f^k(x) = 1$.

2. **Support sets.** All data is assumed to be available at all time. Therefore, when learning the n-th concept, the learner is given a training set X^n of examples and counterexamples of the concept defined by f^n (which might

be distorted by noise), and $n-1$ data sets $X^1, X^2, \ldots, X^{n-1}$ that stem from previous concept learning tasks.

Notice that data in $X^1, X^2, \ldots, X^{n-1}$ can generally not be used directly to augment the training set X^n, since they carry the wrong class labels. However, they may support learning f^n, and are therefore called *support sets*.

3. **Relatedness.** The functions $f^1, f^2, \ldots, f^n$ are drawn from a family of functions, denoted by F. The nature of F is not completely known in the beginning of lifelong learning.

A practical example of this framework is a mobile robot whose task is to find and fetch various objects, using its camera for object recognition. Each object defines a recognition function, $f : \Re^d \longrightarrow \{0, 1\}$, which maps camera images $x \in \Re^d$ to 1, if and only if the object is contained in the image. Consequently, the set F is the set of all recognition functions, one for each (potential) object. When learning to recognize the n-th object, the training set X^n consists of positive and negative examples of that object. The support sets $X^1, X^2, \ldots, X^{n-1}$ contain labeled examples and counterexamples of other objects. Notice that all functions in F are invariant with respect to rotation, translation, scaling in size, change of lighting, and so on. Identifying F involves the identification of these invariances. Hence, given that the learning algorithm is able to learn these and use them to bias subsequent learning, the support sets can reduce the need for training data when learning to recognize the n-th object.

The goal of this paper is to demonstrate that more complex functions can be learned from less training data, when embedded in a lifelong learning context. Lifelong learning goes beyond the intrinsic bounds associated with learning single functions in isolation. The remainder of this paper is organized as follows. The following section introduces the basic terminology of base-level and meta-level learning, and sheds light onto the relation of conventional function fitting and learning bias. Sections 3 and 4 present four approaches to lifelong learning, which extend conventional memory-based and artificial neural network algorithms by a strategy for learning bias. Subsequently, in Sections 5 and 6, lifelong learning is investigated empirically in the context of object recognition, and theoretically in the context of PAC-Learning. The results support our claim that independently of the particular learning approach, lifelong learning approaches are superior to conventional algorithms. The final sections review relevant literature and discuss open problems of the approach taken here.

Figure 1: Meta-level learning—an example. The circles $H_0, H_1, \ldots$ represent different base-level hypothesis spaces. Target functions are drawn from F.

2 Learning Bias

Transferring knowledge across learning tasks involves learning bias. If a learner would approach the n-th learning task with the same, static bias as by which it learns its first one, there would be no way to improve its ability to learn. A simple example of learning bias is shown in Figure 1. Different biases are represented by different hypothesis sets [32] (preferences within these hypothesis sets are ignored to simplify the presentation). Suppose that all target functions are sampled from a specific class of functions F, and suppose the learner can chose its bias from $\{H_0, H_1, \ldots, H_4\}$ prior to the arrival of the training examples for the n-th target function f^n. Of the biases shown in Figure 1, H_4 is superior to all others. H_4 is more appropriate than H_2 and H_3, since it includes F completely while the latter ones do not. It is also more appropriate than H_0 and H_1, since it is more specific than those. Consequently, if the learner starts learning a function sampled from F using the hypothesis space H_4, it will conceivably require less training data than if it had used H_0 or H_1 as initial hypothesis space, and generalize more accurately than with H_2 or H_3. Since previous learning tasks also are sampled from F, learning that H_4 is the best bias in $\{H_0, H_1, \ldots, H_4\}$ appears to be feasible.

Following the terminology in [46], we will refer to the problem of learning bias as the *meta-level learning problem*. The conventional learning problem, which involves learning functions, will be referred to as the *base-level learning*

	base-level	meta-level
example	$\langle x, f^n(x)\rangle$	$X^k = \{\langle x, f^k(x)\rangle\}$
training set	$X^n = \{\langle x, f^n(x)\rangle\}$	$\{X^k\} = \{\{\langle x, f^k(x)\rangle\}\}$
hypothesis	$h : I \longrightarrow O$	$H \subset \{f \vert f : I \longrightarrow O\}$
hypothesis space	$H \subset \{f \vert f : I \longrightarrow O\}$	$\mathcal{H} \subset \wp(\{f \vert f : I \longrightarrow O\})$
target concept	$f^n \in F$	F
objective function ($\rightarrow$ min)	$\sum_{x \in X^n} Prob_{f^n}(x) \ \vert\vert f^n(x) - h(x)\vert\vert$	$\sum_{x \in X^n} Prob_{f^n}(x) \ \vert\vert f^n(x) - h(x)\vert\vert$

Table 1: The base-level and the meta-level in lifelong supervised learning. Here $\wp$ denotes the power set, and $Prob_{f^n}$ denotes the sampling distribution for the n-th dataset.

problem. Both learning problems are closely related. Simplified speaking, entities at the meta-level are power sets of the corresponding entities at the base-level, as depicted in Table 1. As can be seen there, the base-level is concerned with selecting a function h from a set of hypotheses H. The meta-level involves learning an entire space of functions, since its result is an entire base-level hypothesis space H. Consequently, a meta-level hypothesis space is a set of sets of functions, each of which is a potential base-level hypothesis space. Training examples at the base-level are input-output tuples. Training examples at the meta-level are support sets, which are entire sets such tuples.

Clearly, there can be no useful bias-free learning at the meta-level any more than there can be at the base-level. If nothing is known about the relation between different base-level learning tasks, there will be no reason to believe that meta-level learning will improve base-level learning for reasons other than pure chance. The hypothesis spaces shown in Figure 1 constitutes one example of meta-level bias. If the meta-level is equipped with the bias $\mathcal{H} = \{H_1, H_2, H_3, H_4\}$, it is biased towards picking one of those four sets as base-level hypothesis space, ignoring the myriad of alternative ways of combining sets of functions. To learn successfully at the meta-level, the support sets must provide information as to which base-level bias is most appropriate. If, for example, previous learning tasks involve functions f drawn exclusively from F, the learner could use its support sets to determine the most specific function space in $\mathcal{H}$ that includes all previous functions.

Despite these similarities, there are the differences between meta-level and base-level learning.

1. Given a particular target function, $f^n \in F$, the ultimate goal of learning in the n-th learning task is to minimize the prediction error for f^n. Recognizing F is a secondary goal. It is only useful insofar it supports learning f^n.

2. Each support set X^i ($1 \leq i \leq n$) establishes a single training pattern at the meta-level. However, X^i usually does not specify f^i uniquely. Instead, it provides a potentially small and noisy set of input-output examples of f^i.

3. Support sets may vary in cardinality; thus, training examples at the meta-level may vary in length.

4. Each support set X^i provides a positive example for the "meta-concept" F. Negative examples are not available at the meta-level.

The following sections do not present just one particular approach to lifelong learning. In order to investigate the general principles that are at stake in this paper, several are described, some of which have been motivated by or adopted from recent literature. These approaches are compared with learning algorithms that do not transfer knowledge. The comparison, along with a PAC-learning analysis of lifelong learning, demonstrates that more complex functions can be learned from less training data is bias is learned at the meta-level—independently of the particular learning approach.

3 Memory-Based Approaches

The first two lifelong approaches investigated here are memory-based learning algorithms (MBL). Memory-based approaches memorize all training examples explicitly, and interpolate between them at query-time. Notice that memory-based learning has been applied with significant success to a variety of challenging learning problems [35, 51, 69]. In what follows, we will first sketch two well-known approaches to memory-based learning, then propose meta-level components that take the support sets into account.

3.1 Nearest Neighbor

Probably the most widely used memory-based learning algorithm is *K-nearest neighbor (KNN)* [15, 57]. Suppose x is a query pattern, for which we would like to know the output $y = f^n(x)$. KNN searches the set of training examples X^n for those K examples $\langle x_i^n, y_i^n \rangle \in X^n$ whose input patterns x_i^n are nearest to x

Figure 2: Re-representing the data to better suit memory-based algorithms.

(according to a distance metric, *e.g.*, the Euclidian distance). In the context of concept learning, KNN returns the majority vote of the K nearest neighbors:

$$\sigma\left(\frac{1}{K}\sum y_i^n\right) \qquad \text{where} \qquad \sigma(z) := \begin{cases} 1 & \text{if } z > 0.5 \\ 0 & \text{if } z \leq 0.5 \end{cases} \tag{2}$$

3.2 Shepard's Method

Another popular method is due to Shepard [54]. When computing the y for a query point x, Shepard's method averages the output values of *all* training examples in X^n. However, it weights each example $\langle \hat{x}, \hat{y} \rangle \in X^n$ according to the inverse distance to the query point x.

$$s(x) \quad := \quad \left(\sum_{\langle \hat{x}, \hat{y} \rangle \in X^n} \frac{\hat{y}}{||x - \hat{x}|| + \eta} \right) \cdot \left(\sum_{\langle \hat{x}, \hat{y} \rangle \in X^n} \frac{1}{||x - \hat{x}|| + \eta} \right)^{-1} \tag{3}$$

Here $\eta > 0$ is a small constant that prevents numerical overflows.

Notice that both memory-based learning methods (KNN and Shepard's method) use exclusively the training set X^n for learning. There is no obvious way to incorporate the support sets, since those examples carry the wrong class labels.

3.3 Learning Representations

How can one use the support sets to boost generalization? It is well-known that the generalization accuracy of an inductive learning algorithm depends on the representation of the data. This is especially the case when training data is scarce. Hence, one way to exploit support sets in lifelong learning is to develop data representations that better fit the generalization properties of the inductive learning algorithm. As shown in Figure 2, data can be re-represented by a function, denoted by $g : I \longrightarrow I'$, which maps input patterns in I to a new space, I'. This new space I'

forms the input space for a memory-based algorithm. This raises the questions as to what constitutes a good data representation for memory-based learning algorithms.

Obviously, a good transformation g maps multiple examples of a single concept to similar representations, whereas an an example and a counterexample should have distinctly different representations. This property can directly be transformed into an "energy function" for g [62]:

$$E \; := \; \sum_{k=1}^{n-1} \sum_{\langle x, y=1\rangle \in X^k} \left(\sum_{\langle \hat{x}, \hat{y}\rangle \in X^k, \hat{y}=y} \underbrace{||g(x) - g(\hat{x})||}_{(*)} \; - \right.$$
$$\left. \sum_{\langle \hat{x}, \hat{y}\rangle \in X^k, \hat{y}\neq y} \underbrace{||g(x) - g(\hat{x})||}_{(**)} \right) \tag{4}$$

Adjusting g to minimize E forces the distance $(*)$ between pairs of examples of the same concept to be small, and the distance $(**)$ between an example and a counterexample of a concept to be large. Memory-based learning is then performed on the re-represented training set $\{\langle g(x), y\rangle\}$ (with $X = \{\langle x, y\rangle\}$). In our implementation, g is realized by an artificial neural network and trained using the Back-Propagation algorithm [48].

It is important to notice that the transformation g is obtained using the support sets. In the object recognition example described in Section 1, g will—in the ideal case—map images of the same object to an identical representation, regardless of where in the original image the object appears. Such a g entails knowledge about the invariances in the object recognition domain. Hence, learning data representations is one way to change bias in a domain-specific way.

3.4 Learning To Compare

An alternative way for exploiting support sets in the context of memory-based learning is to learn the distance function. One way to do this is to learn a comparator $d : I \times I \longrightarrow [0, 1]$ [63]. A comparator d accepts two input patterns, say x and $\hat{x}$, and outputs 1 if x and $\hat{x}$ are members of the same concept, and 0 otherwise. Consequently, each training example for d is obtained using a pair of examples $\langle x, y\rangle$ and $\langle \hat{x}, \hat{y}\rangle \in X^k$ taken from an arbitrary support set X^k (for all $k = 1, \ldots, n-1$):

$$\begin{array}{ll} \langle (x, \hat{x}), 1\rangle & \text{if } y{=}1 \text{ and } \hat{y}{=}1 \\ \langle (x, \hat{x}), 0\rangle & \text{if } (y{=}1 \text{ and } \hat{y}{=}0) \text{ or } (y{=}0 \text{ and } \hat{y}{=}1) \end{array} \tag{5}$$

If both examples $\langle x, y\rangle$ and $\langle \hat{x}, \hat{y}\rangle$ belong to the same concept class k, they form a positive example for d (first case in (5)). Negative examples for d are composed of an example and a counterexample of a concept (second case in (5)). Consequently, each support set X^k produces $|X^k|^2$ training examples for d. Since the training examples for d lack information concerning the concept for which they were originally derived, all support sets can be used to train d.

When learning a new concept, the comparator d can be used instead of a pre-given, static distance function. For each query point $x \in I$ and each positive training example $\langle \hat{x}, \hat{y}\rangle \in X^n$, the output of the comparator $d(x, \hat{x})$ measures the belief

$$Bel(f^n(x) = 1 \mid f^n(\hat{x}) = \hat{y}) \tag{6}$$

that x is a member of the target concept f^n according to d. Since the value of $d(x, \hat{x})$ depends on the training example $\langle \hat{x}, \hat{y}\rangle$, the belief (6) is conditioned on $\langle \hat{x}, \hat{y}\rangle$.

Obviously, Equation (6) delivers the right answer when only a single positive training example is available. If multiple examples are available in X^n, their votes can be combined using Bayes' rule [42], leading to

$$Bel(f^n(x){=}1) \quad := \quad 1 - \frac{1}{1 \; + \displaystyle\prod_{\langle \hat{x}, \hat{y}=1\rangle \in X^n} \frac{d(x, \hat{x})}{1 - d(x, \hat{x})}} \; . \tag{7}$$

The somewhat lengthy derivation of (7), which is given in [61], is straightforward if one interprets the output of d as a conditional probability for the class of a query point x given a training example $\langle \hat{x}, \hat{y}\rangle$, and if one assumes (conditionally) independent sampling noise X^n. Since (7) combines multiple votes of the comparator d using the training set X^n, the resulting learning scheme is a version of memory-based learning. In the experiments reported below, d is implemented by an artificial neural network. Notice that d is not a distance metric, because the triangle inequality need not hold, and because an example of the target concept $\hat{x}$ can provide evidence that x is *not* a member of that concept (if $d(x, \hat{x}) < 0.5$).

In the context of lifelong learning, learning d can be considered a meta-level learning strategy, since it biases memory-based learning to extrapolate training instances in a domain-specific way. For example, in the object recognition example, d outputs—ideally—the belief that two images show the same object (regardless of the identity of the object). To compare two images, d must possess knowledge about the invariances in the object recognition domain. By learning d, this invariance knowledge is transferred across multiple concept learning tasks.

Figure 3: Re-representing the data to better suit neural network learning.

4 Neural Network Approaches

To make our comparison more complete, we will now describe lifelong approaches that rely exclusively on artificial neural network representations. Neural networks have been applied successfully to a variety of real-world learning problems [47, 43, 49].

4.1 Back-Propagation

Probably the most common way to learn a function $f^n : \Re^d \longrightarrow \{0, 1\}$ with an artificial neural network is to approximate it using the Back-Propagation algorithm (or a variation thereof). The network that approximates f^n might have d input units, one for each of the d input features, and a single output unit that encodes class membership. Such an approach is unable to incorporate the support sets, since their examples carry the wrong concept labels.

4.2 Learning Representations For Neural Networks

As argued in Section 3.3, the generalization accuracy of an inductive learning algorithm depends on the representation of the data. In the context of neural network learning, several researchers have proposed methods for learning data representations that are tailored towards the built-in bias of artificial neural networks [58, 52, 44, 9, 5]. The basic idea here is the same as in Section 3.3. To re-represent the data, these approaches train a neural network, $g : I \longrightarrow I'$, which maps input patterns in I to a new space, I'. This new space I' forms the input space for further, task-specific neural network learning. The overall architecture is depicted in Figure 3.

The question of what representation forms a good basis for neural network learning is not as easily answered as it is in the context of memory-based learning. Basically, all the approaches cited above rely on the observation that the architecture depicted in Figure 3 can be considered a single neural network. Hence, it is possible to use standard Back-Propagation to tune the weights of the transformation network g, along with the weights of the respective classification network. While some authors [52, 44] have proposed to process the support sets and the training set sequentially, others [58, 9, 5] are in favor of training g in parallel, using all n tasks simultaneously. Sequential training offers the advantage that not all training data has to be available at all time. However, it faces the potential burden of "catastrophic forgetting" in Back-Propagation, which basically arises from the fact that the training data in the sequential case is sampled using a non-stationary probability distribution. Both strategies learn at the meta-level through developing new data representations.

4.3 Explanation-Based Neural Network Learning

The remainder of this section describes a hybrid neural network learning algorithm for learning f^n. This algorithm is a special version of both the Tangent-Prop algorithm [56] and the explanation-based neural network learning (EBNN) algorithm [34, 61]. Here we will refer to it as EBNN.

EBNN approximates f^n using an artificial neural network, denoted by $h : I \longrightarrow [0,1]$, just like the conventional Back-Propagation approach to supervised learning. However, in addition to the target values given by the training set X^n, EBNN also constructs the *slopes* (tangents) of the target function f^n at the examples in X^n. More specifically, training examples in EBNN are of the type

$$\langle x, f^n(x), \nabla_x f^n(x) \rangle \ . \tag{8}$$

The first two terms in (8) are just taken from the training set X^n. Obviously, as illustrated by Figure 4, knowing the slope of the target function (third term in (8)) can be advantageous. This is because this slope measures how infinitesimal changes of the features of x will affect its classification, hence can guide the generalization of the training example. However, this raises the question as to how to obtain slope information.

The key to applying EBNN to concept learning lies in the comparator function d described in Section 3.4. In EBNN, d has to be represented by a neural network, hence is differentiable. The slope $\nabla_x f^n(x)$ is obtained using d in the following way. Suppose $\langle \hat{x}, \hat{y} \rangle \in X^n$ is a positive training example in X^n, *i.e.*, $\hat{y} = 1$. Then, the function $d_{\hat{x}} : I \longrightarrow [0,1]$, defined as

$$d_{\hat{x}}(z) \quad := \quad d(z, \hat{x}) \tag{9}$$

Figure 4: Fitting values and slopes. Let f^n be the target function for which three examples $\langle x_1, f^n(x_1)\rangle$, $\langle x_2, f^n(x_2)\rangle$, and $\langle x_3, f^n(x_3)\rangle$ are known. Based on these points the learner might generate the hypothesis h_1. If the slopes are also known, the learner can do much better: h_2.

---

maps a single input z pattern to $[0, 1]$, and is an approximation of the target function f^n. Since $d(z, \hat{x})$ is differentiable, the gradient

$$\frac{\partial d_{\hat{x}}(z)}{\partial z} \tag{10}$$

is defined and is an estimate of the slope of f^n at z. Setting $z := x$ yields the desired estimate of $\nabla_x f^n(x)$ (*cf.* (8)). When refining the weights of the target network that approximates f^n, for each training example $x \in X^n$ both the target value $f^n(x)$ and the slope vector $\nabla_x f^n(x)$ are approximated using the Tangent-Prop algorithm [56].

The slope $\nabla_x f^n$, if correct, provides additional information about the target function f^n. Since d is learned using the support sets, the EBNN approach transfers knowledge from the support sets to the new learning task. To improve the generalization accuracy, d has to be accurate enough to yield helpful sensitivity information. However, since EBNN fits both training patterns (values) and slopes, misleading slopes can be overridden by training examples.

Notice if multiple positive instances are available in X^n, slopes can be derived from each one. In this case, averaged slopes are used to constrain the target function:

$$\nabla_x d(x) \quad := \quad \frac{1}{|X^n_{\text{pos}}|} \sum_{x_{\text{pos}} \in X^n_{\text{pos}}} \frac{\partial d(x, x_{\text{pos}})}{\partial x} \tag{11}$$

Here $X^n_{\text{pos}} \subset X^n$ denotes the set of positive examples in X^n. The application of the EBNN algorithm to learning with invariance networks is summarized in Table 2.

Generally speaking, slope information extracted from the comparator network is a linear approximation to the variances and invariances of F at a specific point

1. *Let* $X^n_{\text{pos}} \subset X^n$ *be the set of positive training examples in* X^n.
2. *Let* $X' = \emptyset$
3. *For each training example* $\langle x, f^n(x) \rangle \in X^n_{\text{pos}}$ *do:*
   (a) *Compute* $\nabla_x d(x) \;=\; \dfrac{1}{|X^n_{\text{pos}}|} \displaystyle\sum_{x_{\text{pos}} \in X^n_{\text{pos}}} \dfrac{\partial d(x)(x_{\text{pos}})}{\partial x}$ *using* d.
   (b) *Let* $X' \;=\; X' \;+\; \langle x, f^n(x), \nabla_x d(x) \rangle$
4. *Fit* X'.

Table 2: Application of EBNN to learning multiple concepts.

in I. Along the invariant directions slopes will be approximately zero, while along others they may be large. For example, in the aforementioned object recognition domain, color might be an important feature for classification while brightness might not be. This is typically the case in situations with changing illumination. In this case, the comparator network ideally ignores brightness, hence the slopes of its classification with respect to brightness will be zero. The slopes for color, however, would be larger, given that color changes imply that the object would belong to a different class.

5 Experimental Results

5.1 Description of the Testbed

To illustrate the utility of meta-level learning when training data is scarce, we collected a database of 700 color camera images of seven different objects described in Table 3. The objects were chosen so as to provide color and size cues helpful for their discrimination. The background of all images consisted of plain, white cardboard. Different images of the same object varied by the relative location and orientation of the object within the image. In 50% of all images, the location of the light source was also changed, producing bright reflections at random locations in various cases. In some of the images the objects were back-lit, in which case they appeared to be black. Example images of all objects are shown in Figure 5 (left columns). Figure 6 shows examples of two of these objects, the *shoe* and the

Object	color	size
bottle	green	medium
hat	blue and white	large
hammer	brown and black	medium
can	red	medium
book	yellow	depending on perspective
shoe	brown	medium
sunglasses	black	small

Table 3: Objects in the image database.

sunglasses, to illustrate the variations in the images. 100 images of each object were available. In all our experiments images were down-scaled to a matrix of 10 by 10 triplets of values. Each pixel of the down-scaled image was encoded by a color value (color is mapped into a cyclic one-dimensional interval), a brightness value and a saturation value. Notice that these values carry the same information as conventional RGB (red/green/blue). Examples of down-scaled images are shown in Figures 5 (right columns) and 6. Although each object appears to be easy to recognize from the original image, in many cases we found it difficult to visually classify objects from the down-sampled images. In this regard, down-scaling makes the learning problem harder. However, down-sampling was also necessary to keep the networks at a reasonable size.

Finding a good approximation to f^n involves recognizing the target object invariant of rotation, translation, scaling in size, change of lighting, and so on. Since these invariances are common to all object recognition tasks, images showing other objects can provide additional information and, thus, boost the generalization accuracy. In all our experiments, the n-th learning task was the task of recognizing one of these objects, namely the *shoe*. The previous $n-1$ learning tasks corresponded to recognizing five other objects, namely the *bottle*, *hat*, *hammer*, *coke can*, and *book*. To ensure that the latter images could not be used simply as additional training data for f^n, the only counterexamples of the *shoe* were images of a seventh object, the *sunglasses*.[1] Hence, the training set for f^n contained images

[1] Since both the positive and negative examples in X^n form a disjunct class of images, it is possible to treat positive and negative examples symmetrically (in all lifelong learning approaches). For example, EBNN derives slopes not only for positive training examples, but also for negative ones. See [63, 61] for more details.

Figure 5: Objects (left) and corresponding input representations (right).

of the *shoe* and the *sunglasses*, and the support sets contained images of the other five objects. Each experiment was performed 100 times under different (random) initial conditions, in order to increase our confidence in the results.

5.2 Results For A Single Training Instance

Transfer of knowledge is most important when training data is scarce. Hence, in an initial experiment we tested all methods using a single image of the *shoe* and the *sunglasses* only. Those methods that are able to transfer knowledge were also provided 100 images of each of the five supporting objects.

The results are intriguing. The generalization accuracies depicted in Table 4 illustrate that all approaches that learn at the meta-level generalize significantly better than those that do not. With the exception of the neural network hint-learning

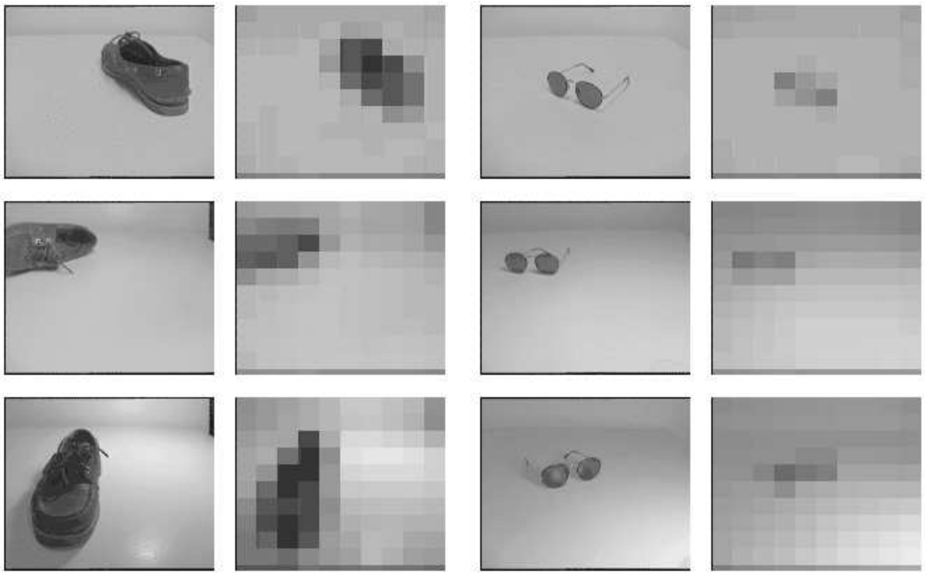

Figure 6: Examples that illustrate some of the variations in the database.

approach, they can be grouped into two categories: Those which generalize approximately 60% of the testing set correctly, and those which achieve roughly 75% generalization accuracy (for comparison: random guessing produces 50% accuracy). The former group contains the conventional supervised learning algorithms, and the latter contains the lifelong approaches. The differences within each group are statistically not significant, while the differences between the groups are (at the 95% confidence level). These results suggest that the generalization accuracy merely depends on the particular choice of the learning algorithm (*e.g.*, memory-based vs. neural networks). Instead, the main factor determining the generalization accuracy is the fact whether or not knowledge is transferred from past learning tasks.

5.3 Increasing the Number of Training Example

What happens as more training data arrives? Figures 7 and 8 show generalization curves with increasing numbers of training examples for some of these methods. As the number of training examples for the n-th learning task increases, the impact of the meta-level learning strategy decreases. After presenting 20 training examples, for example, some of the standard methods (especially Back-Propagation)

	not using support sets				using support sets			
	KNN		Shepard	BP	Shepard	compa-	BP	EBNN
	$K{=}1$	$K{=}2$			repr. g	rator d	repr. g	
Section	3.1	3.1	3.2	4.1	3.3	3.4	4.2	4.3
Accuracy	60.4%	50.0%	60.4%	59.7%	74.4%	75.2%	62.1%	74.8%
Std. deviation	8.3%	0.0%	8.3%	9.0%	18.5%	18.9%	10.2%	11.1%
Conf. interval	59.2%	50.0%	59.2%	57.9%	59.8%	72.6%	59.8%	72.6%
(for the mean)	61.6%	50.0%	61.6%	61.4%	64.3%	77.9%	64.2%	77.0%
statistical confidence in the difference								
KNN, $K{=}1$		100%	**0.0%**	**76.8%**	100%	100%	**90.0%**	100%
KNN, $K{=}2$	100%		100%	100%	100%	100%	100%	100%
Shepard	**0.0%**	100%		**76.8%**	100%	100%	**90.0%**	100%
Backprop.	**76.8%**	100%	**76.8%**		100%	100%	95.4%	100%
Shepard with g	100%	100%	100%	100%		**68.2%**	100%	**60.1%**
comparator d	100%	100%	100%	100%	**68.2%**		100%	**60.2%**
BP with g	**90.0%**	100%	**90.0%**	95.4%	100%	100%		100%
EBNN	100%	100%	100%	100%	**60.1%**	**60.2%**	100%	

Table 4: Statistical comparison for the methods described in this paper, when presenting two training examples and five support sets. The first three rows show the mean accuracy, its standard deviation and the 95% confidence interval for the mean. The bottom table shows the confidence in the statistical difference of the individual approaches. Values smaller than 95% (printed in bold) indicate that the observed performance difference is not statistically significant at the 95% confidence level.

---

generalize about as accurately as those methods that exploit support sets. Here the differences in the underlying learning mechanisms becomes more dominant. However, when comparing lifelong learning methods with their corresponding conventional approaches, the latter ones are still consistently inferior: Back-Propagation (88.4%) is outperformed by EBNN (90.8%), and Shepard's method (70.5%) and

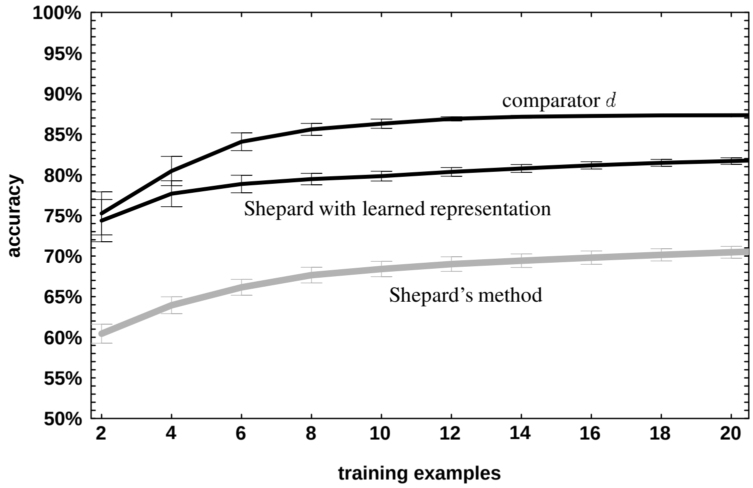

Figure 7: Memory-based approaches: Generalization accuracy as a function of training examples, measured on an independent test set and averaged over 100 experiments. 95%-confidence bars are also displayed.

KNN (81.0%) generalize less accurately when the representation is learned (81.7%) or when the distance function is learned (87.3%). All these differences are significant at the 95% confidence level.

5.4 Degradation

All results reported up to this point employ all five supporting objects at the meta-level. They all show that across the board, learning at the meta-level improves the generalization accuracy when all five support sets are used. However, a natural question to ask is how the different approaches degrade as fewer support sets are available. Will the base-level approach be powerful enough to override wrong (and thus misleading) meta-level knowledge? Or will a poorly trained meta-level make successful generalization impossible at the base-level?

The answers differ for different lifelong learning approaches. To investigate the degradation with the quality of the meta-level knowledge, two different lifelong learning approaches were evaluated: (a) EBNN and (b) memory-based learning using the comparator as distance function. Both these approaches rely on the (identical) comparator network d. However, they trade off their meta-level and base-level component quite differently. When using the comparator in memory-based learning, a poorly trained comparator can prohibit successful generalization,

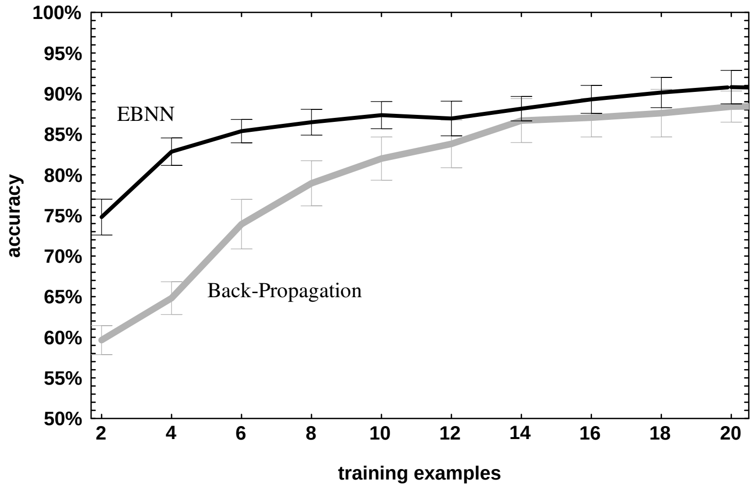

Figure 8: Neural network approaches: Generalization accuracy as a function of training examples.

---

even if the training set X^n is huge and noise-free. This is basically because such a network might not even recognize that two identical input patterns are in fact identical. Consequently, there are cases in which regular memory-based learning (without a meta-level strategy) is expected to outperform the lifelong learning approach. EBNN, on the other hand, uses Back-Propagation as its base-level learning strategy. Hence, even in the presence of a poor comparator d, the built-in bias of neural network Back-Propagation is conceivably able to override errors in the meta-level knowledge—an effect that was confirmed by extensive studies in other application domains [39, 34].

The results shown in Figure 9 confirm our expectations. The results for EBNN, shown in the left diagram, are approximately the same as long as support sets are available (approximately 74% generalization accuracy). Hence, even a poorly trained comparator d still improves the overall generalization accuracy in EBNN. When d is untrained, *i.e.*, its weights are random, the generalization accuracy of EBNN (60.7%) does not differ significantly from that of Back-Propagation (59.7%).

The generalization accuracy of the comparator d (right diagram) depends stronger on the number of support sets and does not degrade as gracefully. While with two support sets, the comparator d generalizes approximately 65.3% of all test examples correctly, it classifies 75.2% of them correctly when given all five support sets. When no support sets are available, the comparator produces random

Figure 9: Generalization accuracy as a function of the support sets, (a) for EBNN, and (b) for the comparator network d. Two training examples were used at the base-level. The error bars indicate a 95% confidence interval for the statistical mean. For comparison, the corresponding conventional approaches are also depicted. Every experiment was repeated 100 times using different base-level training and testing sets.

---

results (50% generalization accuracy), hence is clearly inferior to all other methods studied here, including conventional memory-based approaches with a fixed distance metric (*e.g.*, KNN and Shepard: 60.4%).

It is somewhat surprising that d generalizes better when given three support sets than when given four. This difference is statistically significant at the 95% level. At first glance, one might interpret this finding as evidence that seeing images of the red coke can is counter-supportive. However, this conclusion is questionable in the light of the following two observations. Firstly, the same phenomenon does not appear in EBNN, despite the fact that the same training and testing data were used. Secondly, the performance difference disappears when more than two training examples are available. This can be seen in Figure 10, which depicts the generalization accuracy of the comparator approach with varying numbers of training examples and support sets. This figure clearly illustrates that the generalization accuracy of comparator d increases (a) with the number of available support sets, and (b) with the number of training examples in the n-th learning task. Notice that the upper graph in Figure 10, which is obtained when using all five support sets, is also shown in Figure 7 (upper curve).

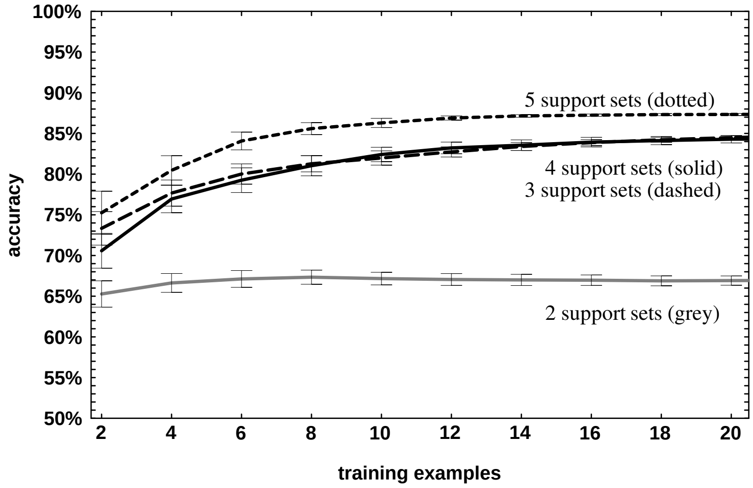

Figure 10: Comparator network g: Generalization curves for different numbers of support sets.

---

6 Analysis

The empirical study provides one example of the successful transfer of knowledge across multiple learning tasks. Why does it work? What are the general mechanisms at work, and when will they succeed?

In the object recognition domain, the function family F, from which all target functions $f^1, f^2, \ldots$ were drawn, had a variety of properties. Some of these properties, such as the invariances with respect to orientation and illumination in object recognition, are unknown in the beginning of lifelong learning. Therefore, the meta-level seeks to recognize these properties. For every property that has been recognized, the meta-level can bias the base-level learning accordingly, which reduces the sample complexity when learning a new concept $f \in F$. In this sense, the object recognition domain is an instance of a more general problem class, which involves the recognition of unknown properties of function classes at the meta-level.

6.1 The Learning Model

To make meta-level learning amendable to a formal analysis, more specific assumptions must be made concerning the nature of hypothesis spaces on both levels. Suppose the learner has an initial hypothesis space, denoted by H, which contains

F. The properties of F are unknown in the beginning of learning. Instead, let us assume there is a pool of m *candidate properties*, denoted by $P_1, P_2, \ldots, P_m$, which the learner is willing to consider. Thus, the task of the meta-level is to learn which of its candidate properties is a property of F.

To facilitate our analysis, let us assume that each property P_j (with $j = 1, \ldots, m$) is only valid for a subset of all functions in H. Let p denote the fraction of functions in H which have property P_j (for reasons of simplicity we assume p is the same for all P_j, $j = 1, \ldots, m$). For example, if a tenth of all functions have property P_j (*e.g.*, only a tenth of all functions in H are invariant with respect to rotation), then $p = 0.1$. Let us also assume that all properties $P_1, P_2, \ldots, P_m$ are independent, *i.e.*, that knowledge about certain properties of a function f does not tell us anything about the correctness of any other property. To further simplify the analysis, let us make the somewhat unrealistic assumption that we have an algorithm that can check (without error and in polynomial time) the correctness of every property P_j (with $j = 1, \ldots, m$) for a support set X^k—notice that in practice, where support sets might contain noisy examples, this could require that the support sets have to be unreasonably large. This simplistic model allows to make assertions about the reduction of the initial base-level hypothesis space when learning f^n.

> **Lemma.** *Any set of l properties that is consistent with all $n-1$ support sets $Y = \{X^k\}$ will reduce the size of the base-level hypothesis space by a factor of p^l. The probability that this reduction removes the target function f^n from the base-level hypothesis space, which will be considered a failure, is bounded above by $p^{n-1} \cdot m^l$.*

Hence, if F has l properties, the meta-level algorithm will identify the correct ones with probability p^l. The resulting reduction of the hypothesis space can be enormous, as illustrated by the following example.

> **Numerical Example 1.** If $p = 0.01$, *i.e.*, every property applies only to 1% of the functions in H (and in F, unless a property is a property of F), and if $l = 3$ properties of $m = 100$ candidate properties are known to be properties of F, the resulting base-level hypothesis space will be reduced by a factor of 10^{-6}. If 10 support sets are available (*i.e.*, $n = 11$), the probability of removing f^n accidently from the base-level hypothesis space (a failure at the meta-level) is bounded above by 10^{-14}.

The proof of the lemma is straightforward.

Proof. According to the definition of p, a single property cuts the hypothesis space H by a factor of p. Therefore, l independent properties cuts the base-level hypothesis space in p^l which proves the first part of the Lemma.

It remains to be shown that the probability of error is bounded above by $p^{n-1} \cdot m^l$. Without loss of generality, consider a specific set of l properties, say $\{P_1, P_2, \ldots, P_l\}$. The probability that these properties are correct for all $n-1$ support sets, although at least one of them is not a property of f^n, is bounded above by p^{n-1}. This is because there must be at least one property in $\{P_1, P_2, \ldots, P_l\}$ which is not property of F. Let P_j denote this property. Then the probability that all $n-1$ support functions have this property just by chance is p^{n-1}.

This argument applies to one specific set of l properties. There are

$$\binom{m}{l} \leq m^l$$

ways to select l out of m candidate properties. The bound $p^{n-1} \cdot m^l$ follows from the subadditivity of probability measures. □

Notice that none of the above arguments depends on the particular learning algorithm used at the meta-level. It is only required that the result of this algorithm, a set of l properties, be consistent with the support sets Y. Hence, *any* learning algorithm that is capable of detecting l properties will exclude f^n accidentally with a probability bounded above by $p^{n-1} \cdot m^l$.

6.2 Relation to PAC-Learning

To illustrate the advantage of smaller hypothesis spaces, let us now combine the bound of the Lemma with known results for base-level learning. It is well-known that the complexity of the base-level hypothesis space is related to the number of training examples required for base-level learning (see *e.g.*, [32, 68, 19, 24]). One learning model, which recently has received considerable attention in the computational learning theory community, is Valiant's PAC-learning model [67] (PAC stands for *probably approximately correct*). PAC-Learning extends Vapnik's approach to empirical risk minimization [68] by an additional computational complexity argument. The following standard result by Blumer and colleagues relates the size of the hypothesis space and the number of (noise-free) training examples required for learning a function:

Theorem [8]. *Given a function f^n in a space of functions H, the probability that any hypothesis $h \in H$ with error larger than ε is consistent with f^n on a (noise-free) dataset of size N is less than $(1-\varepsilon)^N \; |H|$. In other words,*

$$N \geq \frac{1}{-\ln(1-\varepsilon)} \left(\ln(|H|) + \ln\left(\frac{1}{\delta}\right) \right) \tag{12}$$

training examples suffice to ensure, with probability $1-\delta$, that any *hypothesis consistent with the data will not produce an error larger than ε on future data.*

This bound is independent of the learning algorithm—it is only required that the learning algorithm produces a hypothesis that is consistent with the data. It also holds independently of the choice of f^n and the sampling distribution, as long as this distribution is the same during training and testing. Notice that (12) is logarithmic in the hypothesis set size $|H|$. An analogous logarithmic lower bound can be found in [13, 24].

By applying the Lemma to Blumer et al.'s Theorem (12), the advantage of smaller hypothesis spaces can be expressed as the reduction in the sampling complexity when learning the n-th function.

Corollary. *Under the conditions of the Lemma, the upper bound on the number of training examples according to Blumer et al.'s Theorem is reduced by a factor of*

$$1 - \frac{l \;\ln\left(\frac{1}{p}\right)}{\ln\left(\frac{1}{\delta}\right) + \ln|H|} \tag{13}$$

The probability that this reduction erroneously removes the target function f^n from F is bounded above by $p^{n-1} \cdot m^l$.

Equation (13) is obtained from (12) and the Lemma. The following example illustrates the Corollary numerically.

Numerical Example 2. Under the conditions of the first numerical example ($l = 3$, $p = 0.01$, $n = 11$, $m = 100$) and with $|H| = 10^8$ and $\delta = 0.1$, the upper bound (12) is reduced by a factor of $\frac{1}{3}$ (*e.g.*, from 2061.9 to 687.3, if $\varepsilon = 0.01$). That means the guaranteed upper bound on the sample complexity when learning the eleventh function is only a third of the sample complexity when learning the first. The probability that learning might now fail (by erroneously removing the correct function from the hypothesis space) is bounded above by 10^{-14}.

These results shed further light onto the role of meta-level learning in lifelong learning. The more properties of F an algorithm discovers at the base-level, the more dramatic the reduction of the sample complexity when learning a new thing. On the other hand, there is the danger of accidentally assuming false properties. This danger increases with the richness of the meta-level hypothesis space, and with the sparseness of the support sets. Falsely assuming the existence of properties can be considered a meta-level analogue to over-fitting. Hence, to improve base-level learning, care has to be taken to pick the "right" meta-level bias. If the meta-level bias is appropriate, however, base-level learning can be improved greatly.

7 Related Approaches

Sampling complexity is currently one of the main obstacles for applying machine learning to real-world problems. Recent research has produced a variety of approaches that aim to reduce the sampling complexity, in order to overcome this fundamental scaling problem. They can roughly be grouped into the following categories.

- **Choosing learning parameters and algorithms.** One of the earliest approaches that is able to learn at the meta-level is the VBMS system [46]. VBMS chooses the most appropriate algorithm out of a pool of conventional inductive learning algorithms based on previous, related learning tasks. A related approach, the STABB algorithm [66], is able shift gradually towards weaker bias. Bias is represented by a restriction on the hypothesis space [32]. Whenever the hypothesis class cannot match the training examples exactly, STABB analyzes this failure and enlarges the hypothesis space correspondingly. STABB could potentially be applied to noise-free lifelong concept learning tasks. In [36] an approach is described that estimates a variety of learning parameters using cross-validation. In particular their approach used yesterday's training data to tune the learning parameters for today's learning experiments. Some of these parameters address different memory-based generalization methods, others influence the relative weight of instance features in a memory-based approach.

All these approaches are capable of transferring knowledge, hence learn at the meta-level. However, not much can be learned at the meta-level. This is because their meta-level hypothesis spaces comprise only of a considerably small number base-level learning parameters.

- **Learning invariances in face recognition.** In the face recognition context, techniques exist for learning the "directions" (sub-manifolds) along which face images are invariant. In [26], this is done by learning changes in activations when faces are rotated or translated, in a specific internal representational space. These changes are assumed to be equivalent for all faces—hence they can be used to project new faces back into a canonical (frontal) view, in which they are easier to recognize. Beymer and his co-authors [7] propose to learn the parameters for the rotation and change in face expression directly, using a supervised learning scheme. Both approaches are in fact powerful lifelong learning approaches. They illustrate how a carefully designed meta-level bias can improve the recognition rate dramatically, in the domain of face recognition.

- **Learning distance metrics.** Various researchers have proposed methods for adapting the distance metric in memory-based learning [3, 36, 16, 20]. Methods for spotting irrelevant features also fall into this category [27, 10]. With the exception of the (aforementioned) algorithm proposed in [36], all these approaches focus exclusively on single learning tasks. However, they could potentially be applied to lifelong learning, and so provide a good basis for research on lifelong learning. As discussed above, the amount of knowledge that can be transferred by these methods in their current form is limited.

- **Knowledge-Based Approaches.** Knowledge-based approaches to machine learning investigate the feasibility of hand-coding prior knowledge into inductive learning approaches. Various systems have been proposed for inductively refining hand-coded domain theories (see *e.g.*, [6, 41]). For example, EITHER [40] inductively refines an initial domain theory based on noisy training data using ID3 [45] as the inductive component. Neural network-based methods [53, 18, 28, 65] basically initialize neural network weights using domain knowledge, then train the network using conventional neural network training algorithms.

  All these approaches are related to the work reported here, since they employ prior knowledge to reduce the sample complexity. However, knowledge-based learning approaches require that an initial domain theory be available, which is usually provided by a human expert. Lifelong learning approaches can be viewed as knowledge-based approaches that instead *learn* domain knowledge.

- **Other methods.** Other methods, that fit neither of these categories, improve the generalization accuracy of an inductive machine learning algorithm by generating additional training data based on domain knowledge [43], adapt data of multiple tasks to fit a single-task description [21], or provide more flexible mechanisms to encode known invariances of the domain [56].

8 Conclusion

This paper studies approaches to lifelong learning. In lifelong learning, the learner faces a collection of learning tasks over its entire lifetime. When faced with the n-th thing to learn, knowledge acquired in the previous $n - 1$ learning tasks can be used to bias learning the n-th. To elucidate mechanisms for the transfer of knowledge, it is convenient to conceptually split lifelong learning algorithms into two levels: the base-level and the meta-level. Base-level learning corresponds to regular function fitting, using a single dataset. Meta-level learning addresses learning bias for the base-level based on multiple datasets.

To illustrate the advantage of a lifelong perspective over conventional approaches to machine learning, four approaches were described and systematically evaluated. All these approaches process multiple datasets, some of which stem from previous learning tasks.

1. The first algorithm gradually learns a domain-specific data representation, which improves the generalization in memory-based learning.

2. The second algorithm replaces the fixed distance metric in memory-based learning by a domain-specific comparator function, which is learned using previous datasets.

3. The third algorithm (see also [58, 52, 44, 9, 5, 55]) learns a domain-specific representation, like the first algorithm, but this representation is tailored towards neural network learning.

4. Finally, the fourth algorithm, called EBNN, uses the comparator network to derive slopes for the target function, which are fit along with the conventional target values.

All these algorithm integrate standard base-level learning with a meta-level component, that allows them to transfer knowledge across multiple learning tasks. In an empirical evaluation, it was shown that when facing the n-th learning task the sample complexity can be reduced drastically by re-using knowledge acquired in previous learning tasks. For example, after seeing a single image of each class in the object recognition domain, the new approaches consistently generalized approximately 75% of unseen images correctly. Conventional approaches achieve only approximately 60% generalization accuracy. This finding appears to be independent of the particular learning approach: Across the board, all approaches generalize better if knowledge is transferred from previous learning tasks—an observation that is well in tune in what we know about human learning.

Despite these intriguing results, the reader should notice that this paper does not provide a final answer to the lifelong learning problem, neither does it cover the issues exhaustively. All approaches rest on several restrictive assumptions (see also Section 1) that warrant further research:

1. **Concept learning.** This paper exclusively address concept learning problems, which are a version of supervised learning involving only two output values. While it seems feasible to extend these approaches to supervised learning in general, little is known about the transfer of knowledge in other learning paradigms, such as unsupervised learning [29, 50, 14, 25] or reinforcement learning [70, 59, 4, 23]. Some recent results for applying EBNN to reinforcement learning can be found elsewhere [60, 61].

2. **Support sets.** In all experiments, it was assumed that all data be available when learning the n-th function. This is clearly impractical if the number of support sets is large. Designing incremental lifelong learning algorithms is an important issue of future research. At first glance, it appears that training neural networks incrementally provides the desired solution. However, when trained with non-stationary data, neural networks may quickly "forget" previously learned knowledge, which can negatively affect the results.

3. **Relatedness.** It was explicitly assumed that all learning tasks were related in the same way. This assumption enabled our algorithms to incorporate all support sets with equal weight when learning at the meta-level. However, it narrows the applicability of the methods to cases where all learning problems are very similar. To give a simple example that does not meet this assumption suppose in the object recognition domain, some tasks require a machine to learn *where* in the image the object is, whereas others require it to determine *what* object it sees. Clearly, both families of tasks exhibit quite different

invariances. In the latter case, shape and color matter but location does not, whereas in the former case the opposite is the case.

A key open problem in lifelong learning is the problem of *discovering* the concrete relation between multiple learning tasks. The current algorithms can handle only a single type relation, and produce only a single base-level bias. Algorithms that can handle a whole hierarchy of relations (relations among points, among functions (or sets of points), and among sets of functions) are clearly desirable and subject of ongoing research (see also [64]).

Despite these open questions, we envision a variety of practical application domains for the methods and ideas presented here. Meta-level learning is particular relevant to learning problems in which the cost of collecting training data is the dominating factor when applying machine learning techniques. Such domains include, for example, autonomous service robots, which are desired to learn and improve over their entire lifetime. They include personal software agents which have to perform various tasks for various users (hence can transfer knowledge among them). Speech recognition, financial forecasting, and database mining are other, promising application domains for the methods presented here.

The fundamental goal of this research is to scale up machine learning. Most of machine learning has narrowly studied the problem of learning from single datasets, isolated from a more general learning context. Learning single functions in isolation imposes intrinsic scaling limitations. The central claim of this paper is that learning becomes easier when embedded in a lifelong context. Recognizing a complex concept in a high-dimensional feature space based on a single training example is only possible if the learner is biased in the right way. The lifelong learning provides the opportunity to learn the right bias, hence to "learn how to learn." As argued in the introduction, the transfer of knowledge within the lifetime of an individual has been found to be one of the dominating factors of human learning and intelligence. If computers are ever to exhibit rapid learning capabilities similar to that of humans, they will most likely have to follow the same principles.

Acknowledgment

The author wishes to express his gratitude to Tom Mitchell, Armin B. Cremers, and Joseph O'Sullivan for fruitful and enlightening discussions.

References

[1] Ahn, W.-K. and Brewer, W. F. *Psychological Studies of Explanation-Based Learning*. in: **Investigating Explanation-Based Learning**, edited by G. DeJong. Kluwer Academic Publishers, Boston/Dordrecht/London, 1993.

[2] Ahn, W.-K., Mooney, R., Brewer, W. F., and DeJong, G. F. *Schema Acquisition from One Example: Psychological Evidence for Explanation-Based Learning*. in: **Proceedings of the Ninth Annual Conference of the Cognitive Science Society**, edited by . Seattle, WA, 1987, p. .

[3] Atkeson, C. A. *Using Locally Weighted Regression for Robot Learning*. in: **Proceedings of the 1991 IEEE International Conference on Robotics and Automation**, edited by . Sacramento, CA, 1991, pp. 958–962.

[4] Barto, A. G., Bradtke, S. J., and Singh, S. P. *Learning to Act using Real-Time Dynamic Programming*. **Artificial Intelligence**, vol. (to appear), p. .

[5] Baxter, J. *Learning Internal Representations*. in: **Proceedings of the Conference on Computation Learning Theory**, edited by . 1995, p. . *To appear*.

[6] Bergadano, F. and Giordana, A. *Guiding Induction with Domain Theories*. in: **Guiding Induction with Domain Theories**, by F. Bergadano and A. Giordana. Morgan Kaufmann, San Mateo, CA, 1990, pp. 474–492.

[7] Beymer, D., Shashua, A., and Poggio, T. *Example Based Image Analysis and Synthesis*. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, November 1993. *A.I. Memo No. 1431*.

[8] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. *Occam's Razor*. **Information Processing Letters**, vol. 24 (1987), pp. 377–380.

[9] Caruana, R. *Multitask Learning: A Knowledge-Based of Source of Inductive Bias*. in: **Proceedings of the Tenth International Conference on Machine Learning**, edited by P. E. Utgoff. Morgan Kaufmann, San Mateo, CA, 1993, pp. 41–48.

[10] Caruana, R. and Freitag, D. *Greedy Attribute Selection*. in: **Proceedings of the Eleventh International Conference on Machine Learning**, edited by . Morgan Kaufmann, San Mateo, CA, 1994, p. .

[11] **Investigating Explanation-Based Learning**. edited by G. DeJong. Kluwer Academic Publishers, Boston, 1993.

[12] DeJong, G. and Mooney, R. *Explanation-Based Learning: An Alternative View*. **Machine Learning**, vol. 1 (1986), pp. 145–176.

[13] Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L. *A general lower bound on the number of examples needed for learning*. **Information and Computation**, vol. 82 (1989), pp. 247–261.

[14] Fisher, D. H. *Knowledge Acquisition Via Incremental Conceptual Clustering*. **Machine Learning**, vol. 2 (1987), pp. 139–172.

[15] Franke, R. *Scattered Data Interpolation: Tests of Some Methods*. **Mathematics of Computation**, vol. 38 (1982), pp. 181–200.

[16] Friedman, J. H. *Flexible Metric Nearest Neighbor Classification*. Department of Statistics and Linear Accelerator Center, Stanford University, Stanford. CA 94305, November 1994.

[17] Friedman, J. H. *Multivariate Adaptive Regression Splines*. **Annals of Statistics**, vol. 19 (1991), pp. 1–141.

[18] Fu, L.-M. *Integration of Neural Heuristics into Knowledge-Based Inference*. **Connection Science**, vol. 1 (1989), pp. 325–339.

[19] Geman, S., Bienenstock, E., and Doursat, R. *Neural Networks and the Bias/Variance Dilemma*. **Neural Computation**, vol. 4 (1992), pp. 1–58.

[20] Hastie, T. and Tibshirani, R. *Discriminant Adaptive Nearest Neighbor Classification*. Dept. of Statistics and Biostatistics, Stanford University, Stanford, CA, December 1994. *Submitted for publication*.

[21] Hild, H. and Waibel, A. *Multi-Speaker/Speaker-Independent Architectures for the Multi-State Time Delay Neural Network*. in: **Proceedings of the International Conference on Acoustics, Speech and Signal Processing**, IEEE, edited by . 1993, pp. II 255–258.

[22] Jordan, M. I. and Jacobs, R. A. *Hierarchies of adaptive experts*. in: **Advances in Neural Information Processing Systems 4**, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann. Morgan Kaufmann, San Mateo, CA, 1992, pp. 985–992.

[23] Kaelbling, L. P., Littman, M. L., and Moore, A. W. *An Introduction to Reinforcement Learning*. in: **The Biology and Technology of Intelligent Autonomous Agents**, edited by L. Steels. Springer Publishers, Berlin, Heidelberg, 1995, pp. 90–127.

[24] Kearns, M. and Vazirani, U. **Introduction to Computational Learning Theory**. MIT Press, Cambridge, MA, 1994.

[25] Kohonen, T. **Self-Organization and Associative Memory, 2nd. edition**. Springer, Berlin New York, 1988.

[26] Lando, M. and Edelman, S. *Generalizing from a single view in face recognition*. no. CS-TR 95-02, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel, January 1995.

[27] Littlestone, N. *Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm*. **Machine Learning**, vol. 2 (1987), pp. 285–318.

[28] Mahoney, J. J. and Mooney, R. J. *Combining Symbolic and Neural Learning to Revise Probabilistic Theories*. in: **Proceedings of the 1992 Machine Learning Workshop on Integrated Learning in Real Domains**, edited by . Aberdeen Scotland, 1992, p. .

[29] Michalski, R. S. *Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts*. **International Journal of Policy Analysis and Information Systems**, vol. 4 (1980), pp. 219–243.

[30] Mitchell, T. M. *Generalization as Search*. **Artificial Intelligence**, vol. 18 (1982), pp. 203–226.

[31] Mitchell, T. M. *The Need for Biases in Learning Generalizations*. no. CBM-TR-117, Computer Science Department, Rutgers University, New Brunswick, NJ 08904, 1980. *Also appeared in:* Readings in Machine Learning*, J. Shavlik and T.G. Dietterich (eds.), Morgan Kaufmann*.

[32] Mitchell, T. M. *Version Spaces: An approach to concept learning*. Stanford University, California, December 1978. *Also Stanford CS Report STAN-CS-78-711, HPP-79-2*.

[33] Mitchell, T. M., Keller, R., and Kedar-Cabelli, S. *Explanation-Based Generalization: A Unifying View*. **Machine Learning**, vol. 1 (1986), pp. 47–80.

[34] Mitchell, T. M. and Thrun, S. *Explanation-Based Neural Network Learning for Robot Control*. in: **Advances in Neural Information Processing Systems 5**, edited by S. J. Hanson, J. Cowan, and C. L. Giles. Morgan Kaufmann, San Mateo, CA, 1993, pp. 287–294.

[35] Moore, A. W. *Efficient Memory-based Learning for Robot Control*. Trinity Hall, University of Cambridge, England, 1990.

[36] Moore, A. W., Hill, D. J., and Johnson, M. P. *An Empirical Investigation of Brute Force to choose Features, Smoothers and Function Approximators*. in: **Computational Learning Theory and Natural Learning Systems, Volume 3**, edited by S. Hanson, S. Judd, and T. Petsche. MIT Press, 1992.

[37] Moses, Y., Ullman, S., and Edelman, S. *Generalization across changes in illumination and viewing position in upright and inverted faces*. no. CS-TR 93-14, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel, 1993.

[38] Muggelton, S. **Inductive Logic Programming**. Academic Press, New York, 1992.

[39] O'Sullivan, J., Mitchell, T. M., and Thrun, S. *Explanation-Based Neural Network Learning from Mobile Robot Perception*. in: **Symbolic Visual Learning**, edited by K. Ikeuchi and M. Veloso. Oxford University Press, 1995.

[40] Ourston, D. and Mooney, R. J. *Theory Refinement with Noisy Data*. no. AI 91-153, Artificial Intelligence Lab, University of Texas at Austin, March 1991.

[41] Pazzani, M. J., Brunk, C. A., and Silverstein, G. *A knowledge-intensive approach to learning relational concepts*. in: **Proceedings of the Eighth International Workshop on Machine Learning**. Evanston, IL, 1991, pp. 432–436.

[42] Pearl, J. **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[43] Pomerleau, D. A. *Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving*. in: **Robot Learning**, edited by J. H. Connell and S. Mahadevan. Kluwer Academic Publishers, 1993, pp. 19–43.

[44] Pratt, L. Y. *Transferring Previously Learned Back-Propagation Neural Networks to New Learning Tasks*. Rutgers University, Department of Computer Science, New Brunswick, NJ 08904, May 1993. *also appeared as Technical Report ML-TR-37*.

[45] Quinlan, J. R. *Induction of Decision Trees*. **Machine Learning**, vol. 1 (1986), pp. 81–106.

[46] Rendell, L., Seshu, R., and Tcheng, D. *Layered Concept-Learning and Dynamically-Variable Bias Management*. in: **Proceedings of IJCAI-87**. 1987, pp. 308–314.

[47] Rennie, J. *Cancer Catcher: Neural Net catches errors that slip through pap tests*. **Scientific American**, vol. 262 (1990).

[48] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. *Learning Internal Representations by Error Propagation*. in: **Parallel Distributed Processing. Vol. I + II**, edited by D. E. Rumelhart and J. L. McClelland. MIT Press, 1986.

[49] Rumelhart, D. E., Widrow, B., and Lehr, M. A. *The basic Ideas in Neural Networks*. **Communications of the ACM**, vol. 37 (1994), pp. 87–92.

[50] Rumelhart, D. E. and Zipser, D. *Feature Discovery by Competitive Learning*. in: **Parallel Distributed Processing. Vol. I + II**, edited by D. E. Rumelhart and J. L. McClelland. MIT Press, 1986.

[51] Schaal, S. and Atkeson, C. G. *Robot Learning By Nonparametric Regression*. in: **Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems**, edited by . 1994, pp. 478–485.

[52] Sharkey, N. E. and Sharkey, A. J. C. *Adaptive Generalization and the Transfer of Knowledge*. in: **Proceedings of the Second Irish Neural Networks Conference**, edited by . Belfast, 1992, p. .

[53] Shavlik, J. W. and Towell, G. G. *An approach to combining Explanation-based and Neural Learning Algorithms*. **Connection Science**, vol. 1 (1989), pp. 231–253.

[54] Shepard, D. *A Two-Dimensional Interpolation Function for Irregularly Spaced Data*. in: **23rd National Conference ACM**, edited by . 1968, pp. 517–523.

[55] Silver, D. and Mercer, R. *Toward a model of consolidation: The retention and transfer of neural net task knowledge*. in: **Proceedings of the INNS World Congress on Neural Networks**, edited by . Washington, DC, 1995, pp. 164–169, Volume III.

[56] Simard, P., Victorri, B., LeCun, Y., and Denker, J. *Tangent Prop – A Formalism for Specifying Selected Invariances in an Adaptive Network*. in: **Advances in Neural Information Processing Systems 4**, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann. Morgan Kaufmann, San Mateo, CA, 1992, pp. 895–903.

[57] Stanfill, C. and Waltz, D. *Towards Memory-Based Reasoning*. **Communications of the ACM**, vol. 29 (1986), pp. 1213–1228.

[58] Suddarth, S. C. and Holden, A. *Symbolic neural systems and the use of hints for developing complex systems*. **International Journal of Machine Studies**, vol. 35 (1991), p. .

[59] Sutton, R. S. *Integrated Modeling and Control Based on Reinforcement Learning and Dynamic Programming*. in: **Advances in Neural Information Processing Systems 3**, edited by R. P. Lippmann, J. E. Moody, and D. S. Touretzky. Morgan Kaufmann, San Mateo, 1991, pp. 471–478.

[60] Thrun, S. *An Approach to Learning Mobile Robot Navigation*. **Robotics and Autonomous Systems**, 1995. *In press*.

[61] Thrun, S. **Explanation-Based Neural Network Learning: A Lifelong Learning Approach**. Kluwer Academic Publishers, Boston, MA, 1996. *to appear*.

[62] Thrun, S. *Is Learning the n-th Thing Any Easier Than Learning the First?* in: **Advances in Neural Information Processing Systems 8**, edited by D. Touretzky and M. Mozer. MIT Press, Cambridge, MA, 1996, p. . *to appear*.

[63] Thrun, S. and Mitchell, T. M. *Learning One More Thing*. in: **Proceedings of IJCAI-95**, IJCAI, Inc. Montreal, Canada, 1995. *To appear*.

[64] Thrun, S. and O'Sullivan, J. *Clustering Learning Tasks and the Selective Cross-Task Transfer of Knowledge*. no. CMU-CS-95-209, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, November 1995.

[65] Towell, G. G. and Shavlik, J. W. *Knowledge-Based Artificial Neural Networks*. **Artificial Intelligence**, vol. 70 (1994), pp. 119–165.

[66] Utgoff, P. E. **Machine Learning of Inductive Bias**. Kluwer Academic Publishers, 1986.

[67] Valiant, L. G. *A Theory of the Learnable*. **Communications of the ACM**, vol. 27 (1984), pp. 1134–1142.

[68] Vapnik, V. **Estimations of dependences based on statistical data**. Springer Publisher, 1982.

[69] Veloso, M. M. *Learning by Analogical Reasoning in General Problem Solving*. Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, August 1992.

[70] Watkins, C. J. C. H. *Learning from Delayed Rewards*. King's College, Cambridge, England, 1989.

[71] Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in the Behavorial Sciences*. Harvard University, Committee on Applied Mathematics, Cambridge, MA, November 1994.

[72] Widrow, B. and Hoff, M. E. **Adaptive Switching Circuits**. Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part4, 1960.

[73] Wolpert, D. H. *Off-Training set error and a priori distinctions between learning algorithms*. no. SFI TR 95-01-003, Santa Fe Institute, Santa Fe, NM 87501, 1994.