Learning Many Related Tasks at the Same Time With Backpropagation

Rich Caruana
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
caruana@cs.cmu.edu

Abstract

Hinton [6] proposed that generalization in artificial neural nets should improve if nets learn to represent the domain’s underlying regularities. Abu-Mostafa’s hints work [1] shows that the outputs of a backprop net can be used as inputs through which domain-specific information can be given to the net. We extend these ideas by showing that a backprop net learning many related tasks at the same time can use these tasks as inductive bias for each other and thus learn better. We identify five mechanisms by which multitask backprop improves generalization and give empirical evidence that multitask backprop generalizes better in real domains.

1 INTRODUCTION

You and I rarely learn things one at a time, yet we often ask our programs to—it must be easier to learn things one at a time than to learn many things at once. Maybe not. The things you and I learn are related in many ways. They are processed by the same sensory apparatus, controlled by the same physical laws, derived from the same culture, ... Perhaps it is the similarity between the things we learn that helps us learn so well. What happens when a net learns many related functions at the same time? Will the extra information in the teaching signal of the related tasks help it learn better?

Section 2 describes five mechanisms that improve generalization in backprop nets trained simultaneously on related tasks. Section 3 presents empirical results from a road-following domain and an object-recognition domain where backprop with multiple tasks improves generalization 10–40%. Section 4 briefly discusses when and how to use multitask backprop. Section 5 cites related work and Section 6 outlines directions for future work.
2 MECHANISMS OF MULTITASK BACKPROP

We identified five mechanisms that improve generalization in backprop nets trained simultaneously on multiple related tasks. The mechanisms all derive from the summing of error gradient terms at the hidden layer from the different tasks. Each exploits a different relationship between the tasks.

2.1 Data Amplification

Data amplification is an effective increase in sample size due to extra information in the training signal of related tasks. There are two types of data amplification.

2.1.1 Statistical Data Amplification

Statistical amplification, occurs when there is noise in the training signals. Consider two tasks, \( T \) and \( T' \), with independent noise added to their training signals, that both benefit from computing a feature \( F \) of the inputs. A net learning both \( T \) and \( T' \) can, if it recognizes that the two tasks share \( F \), use the two training signals to learn \( F \) better by averaging \( F \) through the noise. The simplest case is when \( T = T' \), i.e., when the two outputs are independently corrupted versions of the same signal.

2.1.2 Blocking Data Amplification

The 2nd form of data amplification occurs even if there is no noise. Consider two tasks, \( T \) and \( T' \), that use a common feature \( F \) computable from the inputs, but each uses \( F \) for different training patterns. A simple example is \( T = A OR F \) and \( T' = NOT(A) OR F \). \( T \) uses \( F \) when \( A = 0 \) and provides no information about \( F \) when \( A = 1 \). Conversely, \( T' \) provides information about \( F \) only when \( A = 1 \). A net learning just \( T \) gets information about \( F \) only on training patterns for which \( A = 0 \), but is blocked when \( A = 1 \). But a net learning both \( T \) and \( T' \) at the same time gets information about \( F \) on every training pattern; it is never blocked. It does not see more training patterns, it gets more information for each pattern. If the net learning both tasks recognizes the tasks share \( F \), it will see a larger sample of \( F \). Experiments with blocked functions like \( T \) and \( T' \) (where \( F \) is a hard but learnable function of the inputs such as parity) indicate backprop does learn common subfeatures better due to the larger effective sample size.

2.2 Attribute Selection

Consider two tasks, \( T \) and \( T' \), that use a common subfeature \( F \). Suppose there are many inputs to the net, but \( F \) is a function of only a few of the inputs. A net learning \( T \) will, if there is limited training data and/or significant noise, have difficulty distinguishing inputs relevant to \( F \) from those irrelevant to it. A net learning both \( T \) and \( T' \), however, will better select the attributes relevant to \( F \) because data amplification provides better training signals for \( F \) and that allows it to better determine which inputs to use to compute \( F \). (Note: data amplification occurs even when there is no attribute selection problem. Attribute selection is a consequence of data amplification that makes data amplification work better when a selection problem exists.) We detect attribute selection by looking for connections to relevant inputs that grow stronger compared to connections for irrelevant inputs when multiple tasks are trained on the net.
2.3 Eavesdropping

Consider a feature $F$, useful to tasks $T$ and $T'$, that is easy to learn when learning $T$, but difficult to learn when learning $T'$ because $T'$ uses $F$ in a more complex way. A net learning $T$ will learn $F$, but a net learning just $T'$ may not. If the net learning $T'$ also learns $T$, $T'$ can eavesdrop on the hidden layer learned for $T$ (e.g., $F$) and thus learn better. Moreover, once the connection is made between $T'$ and the evolving representation for $F$, the extra information from $T'$ about $F$ will help the net learn $F$ better via the other mechanisms. The simplest case of eavesdropping is when $T = F$. Abu-Mostafa calls these catalytic hints[1]. In this case the net is being told explicitly to learn a feature $F$ that is useful to the main task. Eavesdropping sometimes causes non-monotonic generalization curves for the tasks that eavesdrop on other tasks. This happens when the eavesdropper begins to overtrain, but then finds something useful learned by another task, and begins to perform better as it starts using this new information.

2.4 Representation Bias

Because nets are initialized with random weights, backprop is a stochastic search procedure; multiple runs rarely yield identical nets. Consider the set of all nets (for fixed architecture) learnable by backprop for task $T$. Some of these generalize better than others because they better “represent” the domain’s regularities. Consider one such regularity, $F$, learned differently by the different nets. Now consider the set of all nets learnable by backprop for another task $T'$ that also learns regularity $F$. If $T$ and $T'$ are both trained on one net and the net recognizes the tasks share $F$, search will be biased towards representations of $F$ near the intersection of what would be learned for $T$ or $T'$ alone. We conjecture that representations of $F$ near this intersection often better capture the true regularity of $F$ because they satisfy more than one task from the domain.