Deep Supervision with Intermediate Concepts

Chi Li\textsuperscript{1}, M. Zeeshan Zia\textsuperscript{2}, Quoc-Huy Tran\textsuperscript{2}, Xiang Yu\textsuperscript{2}, Gregory D. Hager\textsuperscript{1}, and Manmohan Chandraker\textsuperscript{2,3}

\textsuperscript{1}Johns Hopkins University. \{chi\_li,hager\}@cs.jhu.edu
\textsuperscript{2}NEC Labs America. \{zeeshan,ghtran,xiangyu,manu\}@nec-labs.com
\textsuperscript{3}UC San Diego. \{mkchandraker\}@eng.ucsd.edu

We introduce a novel technique for training convolutional neural networks (CNNs), namely deep supervision with intermediate concepts, leading to improved generalization. Our approach draws inspiration from Deeply Supervised Nets (DSN) \cite{ZagoruykoAndSutskever16}, which supervises each layer by the main task to accelerate training convergence. Our method differs from DSN in that we apply deep supervision with intermediate concepts, intrinsic to the ultimate task, to regularize the network for better generalization. We apply this improved generalization ability to transfer knowledge from synthetic to real images.

\textbf{Toy Example.} To motivate the idea of deep supervision with intermediate concepts, let us consider a simple network with 2 layers $y = \sigma(w_2\sigma(w_1 x + b_1) + b_2)$, where $\sigma$ is ReLU activation. Provided that the true model for a phenomenon is $(w_1, w_2, b_1, b_2) = (3, 1, -2, -7)$ and the training data $\{(x, y)\}$ is $\{(1, 0), (2, 0), (3, 0)\}$. A learning algorithm may obtain a different model $(w_1, w_2, b_1, b_2) = (1, 3, -1, -10)$, which still achieves zero loss over training data but fails to generalize to the case when $x = 4$ or 5. However, if we have additional cues that tell us the values of intermediate layer activations, $\sigma(w_1 x + b_1)$ for each $(x, y)$, we can achieve better generalization. For example, suppose we have training examples with additional intermediate cues $\{(x, y', y)\} = \{(1, 0, 0), (2, 0, 0), (3, 1, 0)\}$, with $y' = \sigma(w_1 x + b_1)$. We find that the incorrect solution above that works for $\{(x, y)\}$ is removed because it does not agree with $\{(x, y', y)\}$. While simple, this example illustrates that deep supervision with intermediate concepts can regularize network training and reduce overfitting.

1 Intermediate Concepts

We consider a supervised learning task to predict $y_m$ from $x$. We have a training set $S = \{(x, (y_1, \cdots, y_m))\}$, sampled from an unknown distribution $D$, where each training tuple consists of multiple task labels $(y_1, \cdots, y_m)$. Without the loss of generality, we analyze the $i$-th concept $y_i$ in the following, with $1 < i \leq m$. Here, $y_{i-k}$ is regarded as an intermediate concept to estimate $y_i$, with $0 < k < i$. Intuitively, the estimate of $y_{i-k}$ constrains that of $y_i$, as in our above example.

Formally, we define an intermediate concept $y_{i-k}$ of $y_i$ as a strict necessary condition such that there exists a deterministic function $T$ which maps $y_i$ to $y_{i-k}$: $y_{i-k} = T(y_i)$. In general, there is no inverse function $T'$ that maps $y_{i-k}$ to $y_i$ because multiple $y_i$ may map to the same $y_{i-k}$. In the context of multi-class classification where tasks $y_i$ and $y_{i-k}$ both contain discrete class labels, task $y_i$ induces a finer partition over the input space $X = \{x\}$ than task $y_{i-k}$ by further partitioning each class in $y_{i-k}$. Figure \ref{fig:toy} shows a fictitious example of hierarchical partitioning over 2D input space created by three concepts $\{y_1, y_2, y_3\}$. As shown in Figure \ref{fig:toy}, a sequence of concepts hierarchically decompose the input space from coarse to fine granularity. Concretely, we denote a concept hierarchy as $\mathcal{Y} = \{y_1, \cdots, y_m\}$ where $y_{i-k}$ is a strict necessary condition of $y_i$ for all $i > 1$.

2 Deep Supervision with Intermediate Concepts

Given a concept hierarchy $\mathcal{Y}$ and the corresponding training set $S$, we formulate a new deeply supervised architecture to jointly learn the main task along with its intermediate concepts. Consider a CNN with $N$ hidden layers that receives input $x$ and outputs $m$ predictions for $y_1, \cdots, y_m$. The $i$-th concept $y_i$ is applied to supervise the intermediate hidden layer at depth $d_i$ by adding a side output branch at the $d_i$-th hidden layer. We denote the function represented by the $k$-th hidden layer as $h_k(\cdot, W_k)$, with parameters $W_k$. The output branch at depth $d_i$ constructs a function $g_{d_i}(\cdot, V_{d_i})$, with parameters $V_{d_i}$. Further, we denote $f_{y_i}$ as the function for predicting concept $y_i$ such that
\begin{equation}
f_{y_i} = g_{d_i} \circ h_{d_i} \circ \cdots \circ h_1.
\end{equation}
Figure 2 shows a schematic diagram of our deep supervision framework with three concepts. We formulate the following objective function to encapsulate these ideas:

\begin{equation}
W^*, V^* = \underset{W,V}{\text{argmin}} \sum_{(x, (y_i))} \sum_{i=1}^m \lambda_i l_i(y_i, f_{y_i}(x; W_{1:d_i}, V_{d_i})),
\end{equation}

with \(W_{1:d_i} = \{W_1, \cdots, W_{d_i}\}\), \(W = W_{1:d_m}\), and \(V = \{V_{d_1}, \cdots, V_{d_m}\}\). In addition, \(l_i\) is the loss for task \(y_i\), scaled by the loss weight \(\lambda_i\). We optimize Equation (1) over \(S\) by simultaneously backpropagating the loss of each supervisory signal all the way back to the first layer.

We note that Equation (1) is a generic supervision framework, which represents many existing supervision schemes. For example, the standard CNN with a single task supervision is a special case when \(m = 1\). Additionally, the multi-task learning framework places all task supervisions on the last hidden layer when \(d_i = N\) for all \(i\). DSN [5] framework is obtained when \(m = N\) and \(y_i = y_m\) for all \(i\). In this work, we propose to apply \(m\) different concepts \(\{y_i\}\) in a concept hierarchy \(\mathcal{Y}\) at locations with growing depths \(d_{i-k} < d_i\) with \(0 < k < i\). In addition, please see [6] for a detailed analysis on improved generalization by deep supervision with intermediate concepts.

### 3 Experimental Results

Here we present an empirical study of image classification on CIFAR100 [4], where a strict concept hierarchy is applied to boost fine-grained object classification performance. In addition, please see [6] for extensive evaluation on 2D/3D keypoint localization on several datasets such as KITTI-3D, PASCAL VOC, PASCAL3D+ [9], and IKEA [7], where our method trained exclusively on synthetic data (without any pre-training or fine-tuning on real data) outperforms state-of-the-art methods trained on real data, thanks to improved generalization by deep supervision with intermediate concepts.

Image classification has a natural concept hierarchy where object categories can be progressively partitioned from coarse to fine granularity. We leverage coarse-grained labels (20 classes) as an intermediate concept in our formulation to assist fine-grained recognition (100 classes). We use a CNN with 20 layers and employ coarse-grained supervision at layer 16. Table 1 shows the results of our method (DISCO) and its variants. We use plain-single and plain-all to denote the networks with supervision of single fine-grained label and both labels at the last layer respectively. DISCO-random uses a (fixed) random coarse-grained label for each training image. We observe that plain-all achieves roughly the same performance as plain-single, which shows that intermediate supervision signal applied at the same layer as the main task helps relatively little in generalization. However, DISCO is able to reduce the error of plain-single by roughly 0.6% using the intermediate supervision signal. Further, DISCO-random is significantly inferior to DISCO as a random intermediate concept makes the training more difficult. Finally, DISCO slightly outperforms the current state-of-the-art on image classification but with only half of the network parameters as compared to [4].

<table>
<thead>
<tr>
<th>Methods</th>
<th>Errors (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN [5]</td>
<td>34.37</td>
</tr>
<tr>
<td>FitNet, LSUV [8]</td>
<td>27.66</td>
</tr>
<tr>
<td>ResNet-1001 [2]</td>
<td>27.82</td>
</tr>
<tr>
<td>plain-single</td>
<td>23.31</td>
</tr>
<tr>
<td>plain-all</td>
<td>23.26</td>
</tr>
<tr>
<td>DISCO-random</td>
<td>27.53</td>
</tr>
<tr>
<td>DISCO</td>
<td>22.46</td>
</tr>
</tbody>
</table>

Table 1: Image classification on CIFAR100 [4]. The first four are previous methods with [4] being the current state-of-the-art. The remaining four are our method (DISCO) and its variants.
References


