DisturbLabel: Regularizing CNN on the Loss Layer
Lingxi Xie\textsuperscript{1}, Jingdong Wang\textsuperscript{2}, Zhen Wei\textsuperscript{3}, Meng Wang\textsuperscript{4} and Qi Tian\textsuperscript{5}
\textsuperscript{1}University of California, Los Angeles
\textsuperscript{2}Microsoft Research
\textsuperscript{3}Shanghai Jiao Tong University
\textsuperscript{4}Hefei University of Technology
\textsuperscript{5}University of Texas at San Antonio

\textbf{ABSTRACT}
During a long period of time we are combating over-fitting in the CNN training process with model regularization, including weight decay, model averaging, data augmentation, etc. In this paper, we present DisturbLabel, an extremely simple algorithm which randomly replaces a part of labels as incorrect values in each iteration. Although it seems weird to intentionally generate incorrect training labels, we show that DisturbLabel can serve as the first work which adds noises on the network training, it can be explained as two regularization functions. Experiments demonstrate competitive recognition results on several popular image recognition datasets.

\textbf{THE PROPOSED ALGORITHM}

Pseudo codes for DisturbLabel
\begin{verbatim}
1. Input: a dataset D = \{(x\textsubscript{m}, y\textsubscript{m})\textsubscript{m=1}\}
2. Initialization: a network M: f(x; \theta) \in R\textsuperscript{C},
3. for each mini-batch D\textsubscript{t} = \{(x\textsubscript{m}, y\textsubscript{m})\textsubscript{m=1}\}
4. for each training sample (x\textsubscript{m}, y\textsubscript{m}) do
5. Generate a disturbed label y\textsubscript{m}^D,
6. end for
7. Update the parameter \theta with SGD,
8. end for
9. Output: the trained model M: f(x; \theta) \in R\textsuperscript{C}.
\end{verbatim}

Implementation details of label disturbance
- Each sample (x, y) is sent into an extra sampling process, in which a disturbed label vector \(y^D = [y_1^D, y_2^D, \ldots, y_n^D]\) in randomly generated from a Multinoulli function (generalized Bernoulli) distribution \(P(x)\). Suppose that the sampled integer is \(\tilde{c}\), then we have \(y_{\tilde{c}}^D = 1\) and \(y_m^D = 1\) for all \(m \neq \tilde{c}\).
- An illustration of the DisturbLabel algorithm (\(\alpha = 10\%\)). A mini-batch of 10 training samples is used as the toy example. Each sample is disturbed with the probability \(\alpha\). A disturbed training sample is marked with a red frame and the disturbed label is written below the frame. Even if a sample is disturbed, the label may remain unchanged (e.g., the digit 3 in the 3rd mini-batch).

\textbf{RESULTS}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
Dataset & OA & OA & OA & OA \\
\hline
\textit{A} & 82 & 78 & 86 & 92 \\
\hline
\textit{B} & 97 & 93 & 100 & 100 \\
\hline
\textit{C} & 83 & 80 & 86 & 92 \\
\hline
\textit{D} & 100 & 100 & 100 & 100 \\
\hline
\end{tabular}
\caption{Results on some small datasets}
\end{table}

\textbf{CONTRIBUTION}

In this paper, we present a novel algorithm named DisturbLabel, which regularizes CNNs by intentionally introducing incorrect labels in the training process. Each training iteration, each training sample is randomly picked up with probability \(\alpha\), then assigned a random label. To the best of our knowledge, this is the first work to add noise on the loss layer.

DisturbLabel prevents over-fitting in the CNN training process. It can be explained as two different ways, i.e., model ensemble and data augmentation, both of which are performed in a latent manner.

\begin{itemize}
\item Like Dropout, a popular regularization algorithm, DisturbLabel can be explained as a latent way of using a large number of models. In Dropout, models are trained with the same data and different network structures, but in DisturbLabel, models are trained with the same network structure and different data. Therefore, DisturbLabel can be used with Dropout.
\item DisturbLabel can also be explained as data augmentation (see the figure to the right). It is equivalent to generating many different training samples that increase the ability of the network. Therefore, DisturbLabel can be used in the scenarios with fewer training data or with imbalanced training data, since it shares data among different categories.
\end{itemize}

The implementation of DisturbLabel is very easy, with only few lines of codes. Experimental results on several standard benchmark datasets verify that DisturbLabel produces competitive recognition results.

A: Dropout, B: DisturbLabel. A-C: + both, A-D: + Dropout, B-D: + Dropout, C-D: + Dropout, D-D: no regularization

\textbf{REFERENCES}


\textbf{ACKNOWLEDGEMENTS}
This work was done when Lingxi Xie and Zhen Wei were interns at MSR. This work was supported by ARQ grant W911NF-15-1-0290, Faculty Research Gift Awards by NEC Labs of America and Bllarp, and NSF 61322201, 61420021 and 61420190. We thank Prof. Alan Yuille and the anonymous reviewers for instructive discussions.