Learning Deep Networks from Noisy Labels with Dropout Regularization

Ishan Jindal*, Matthew Nokleby*, Xuewen Chen**
*Department of Electrical and Computer Engineering
**Department of Computer Science
Wayne State University
Supervised Learning with Noisy Labels

- Usual supervised learning: minimize empirical loss over training image/label pairs:
  \[ D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]
  \[ x_i \in \mathbb{R}^d, \quad y_i \in \{1, \ldots, C\} \]

- Real-world training sets suffer from incorrect labels:
  \[ D' = \{(x_1, y'_1), (x_2, y'_2), \ldots, (x_n, y'_n)\} \]

- Performance of SVMs, k-NN, naive Bayes well-studied under noisy labels [1,2]

- What about deep learning?

Our Contributions

• **Joint** learning of deep *image classifier* and a *label noise model*
• Innovation: Use *dropout regularization* to ensure that we learn a non-trivial noise model
• For i.i.d. label noise on CIFAR-10/MNIST, dropout outperforms the state of the art, even *outperforms a genie-aided architecture* that knows the noise statistics
• Dropout leads to a *pessimistic noise model*, encourages unsupervised learning
Joint Classifier/Noise Architecture

- Want to predict **denoised labels**
- Learn deep classifier **simultaneously** with noise model
- Implicit assumption: label “flip” probability depends only on \((y, y')\), described by a stochastic \(C \times C\) matrix \(\Psi\)

Joint Classifier/Noise Architecture

- Train via backpropagation/SGD on the noisy training set
- Noise model “denoises” labels from the training set during SGD
- Joint model is underdetermined: will learn a trivial noise model without regularization

**Dropout Regularization**

- For each SGD mini batch, disconnect some fraction $q$ of the units for denoised label probabilities.
- Forces the learning “action” on the remaining labels, encourages a pessimistic noise model.
- In practice, aggressive dropout ($q \geq 0.8$) works best.

Dropout Regularization

For each SGD mini batch, disconnect some fraction $q$ of the units for denoised label probabilities.

Forces the learning “action” on the remaining labels, encourages a pessimistic noise model.

In practice, aggressive dropout ($q \geq 0.8$) works best.

Dropout Regularization

- For each SGD mini batch, disconnect some fraction $q$ of the units for denoised label probabilities
- Forces the learning “action” on the remaining labels, encourages a pessimistic noise model
- In practice, aggressive dropout ($q \geq 0.8$) works best

Implementation + Simulation setup

- Implement in MATLAB using MatConvNet [1]
- Two deep network architectures:
  - **Three-layer CNN**, similar to “AlexNet”
  - Three-layer fully-connected DNN, with ReLUs
- Datasets: **CIFAR-10** and MNIST (C = 10)
- Generate synthetic label noise:
  - Uniform (i.i.d. label flips): \( \Psi = (1 - p)I + \frac{p}{C}11^T \)
  - Non-uniform (\( \Delta \) drawn from unit simplex): \( \Psi = (1 - p)I + p\Delta \)
- Use cross-entropy loss + dropout regularization

Simulation Results: Uniform Noise

- Uniform noise model: \( \Psi = (1 - p)I + \frac{p}{C}11^T \)
- Error probability on CIFAR-10, compared against:
  - Noise-blind/standard CNN, trace regularization [3], CNN + genie-aided true noise model, noise-free learning

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>29.78%</td>
<td>24.43%</td>
<td>26%</td>
<td>25.76%</td>
<td>20.49%</td>
</tr>
<tr>
<td>50%</td>
<td>38.76%</td>
<td>32.64%</td>
<td>35%</td>
<td>29.63%</td>
<td>20.49%</td>
</tr>
<tr>
<td>70%</td>
<td>48.34%</td>
<td>33.00%</td>
<td>63%</td>
<td>36.24%</td>
<td>20.49%</td>
</tr>
</tbody>
</table>

- Dropout outperforms other methods, usually beats the genie-aided solution!

Simulation Results: Non-uniform Noise

- Non-uniform noise model: $\Psi = (1 - p)\mathbf{I} + p\Delta$
- Error probability on CIFAR-10, compared against:
  - Noise-blind/standard CNN, trace regularization [3], CNN + genie-aided true noise model, noise-free learning

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>30.49%</td>
<td>25.4%</td>
<td>26%</td>
<td>24.95%</td>
<td>20.49%</td>
</tr>
<tr>
<td>50%</td>
<td>39.47%</td>
<td>31.28%</td>
<td>35%</td>
<td>29.9%</td>
<td>20.49%</td>
</tr>
<tr>
<td>70%</td>
<td>65.6%</td>
<td>63.04%</td>
<td>63%</td>
<td>63.91%</td>
<td>20.49%</td>
</tr>
</tbody>
</table>

- Dropout performs well, less competitive with non-uniform noise
- Still learns a nearly-uniform noise model

Conclusion

• Studied deep learning with **noisy labels** in the training set
• Proposed **dropout regularization** for noise model learning
• Encourages the learning of a nearly-uniform, **pessimistic** noise model
• Competitive performance, especially when the label noise is uniform
• Upshot: with label noise, we should encourage the model to cluster the training data as well as to classify it

**Code:** [github.com/ijindal/Noisy_Dropout_regularization](https://github.com/ijindal/Noisy_Dropout_regularization)