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

WAYNE STATE UNIVERSITY

Supervised Learning with Noisy Labels

- Usual supervised learning: minimize empirical loss over training image/label pairs:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$
$$x_i \in \mathbb{R}^d, \;\; y_i \in \{1, \ldots, C\}$$

- Real-world training sets suffer from **incorrect labels**:

$$\mathcal{D}' = \{(x_1, y_1'), (x_2, y_2'), ..., (x_n, y_n')\}$$

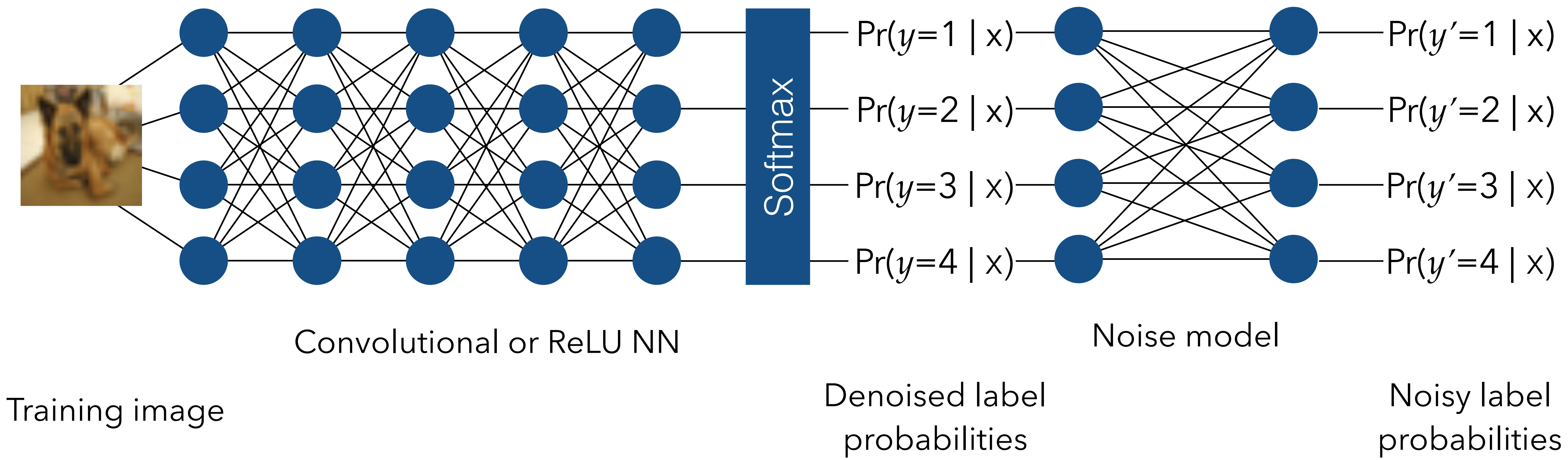
- Performance of SVMs, k-NN, naive Bayes well-studied under noisy labels [1,2]
- What about **deep learning**?

[1] N. Natarajan et al., "Learning with noisy labels," 2014
[2] B. Frenay and M. Verleysen, "Classification in the presence of label noise: a survey," 2014

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**

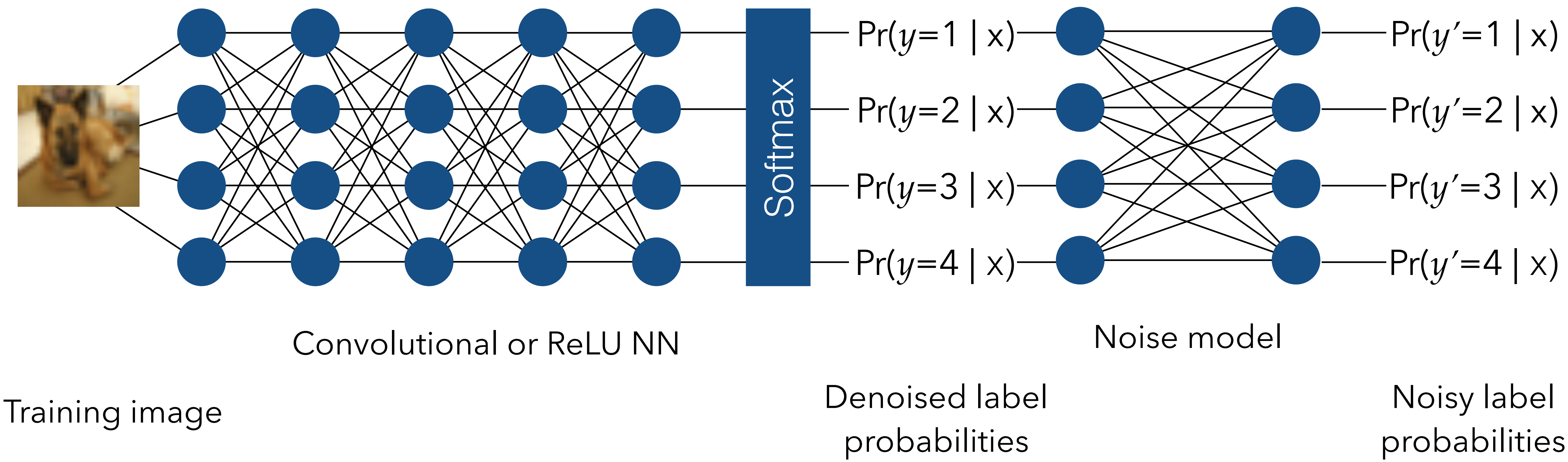
Joint Classifier/Noise Architecture



- 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 CxC matrix $\boldsymbol{\Psi}$

[3] S. Sukhbaatar et al., "Training convolutional networks with noisy labels," 2014.

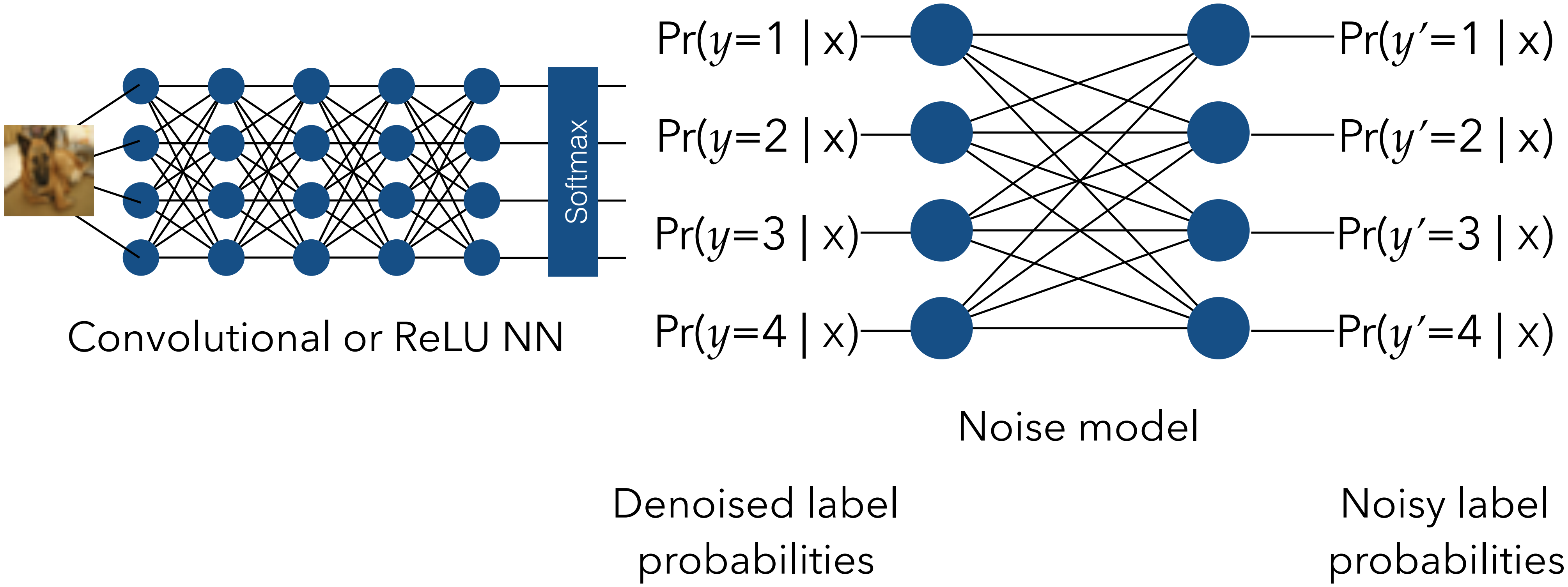
Joint Classifier/Noise Architecture



- 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

[3] S. Sukhbaatar et al., "Training convolutional networks with noisy labels," 2014.

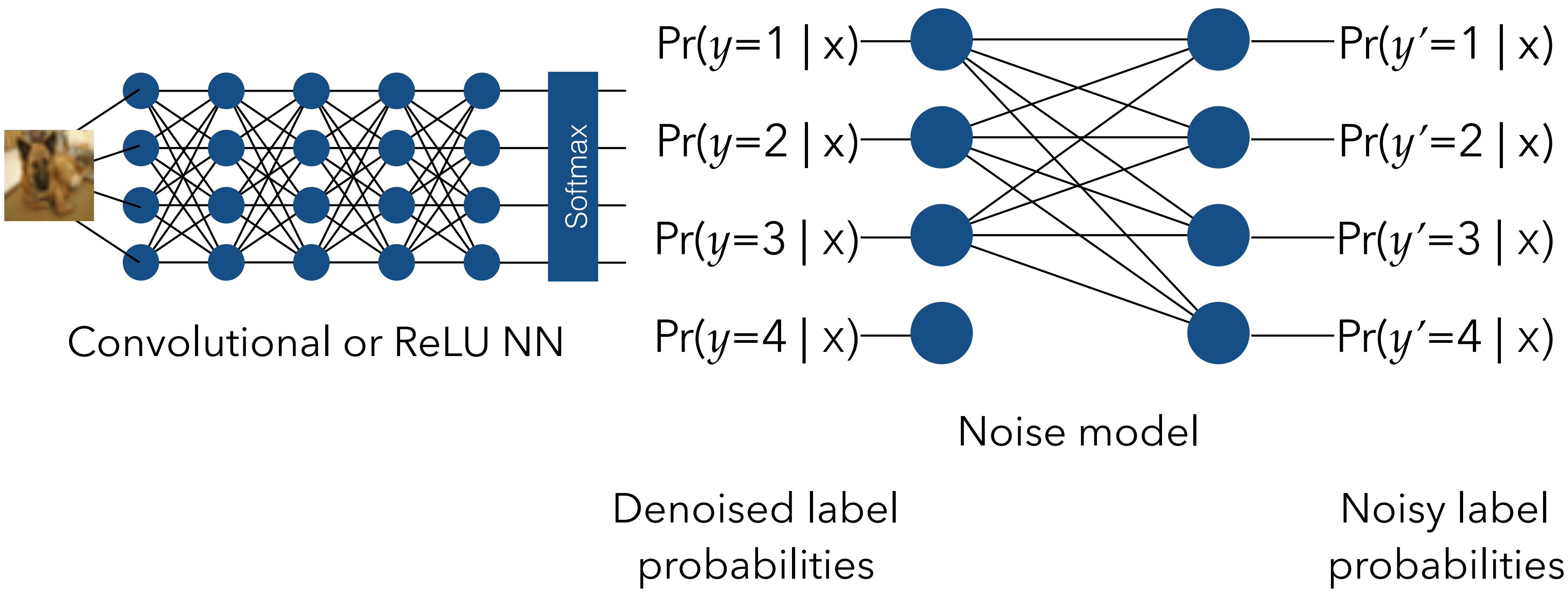
Dropout Regularization



- 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

[4] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," 2014.

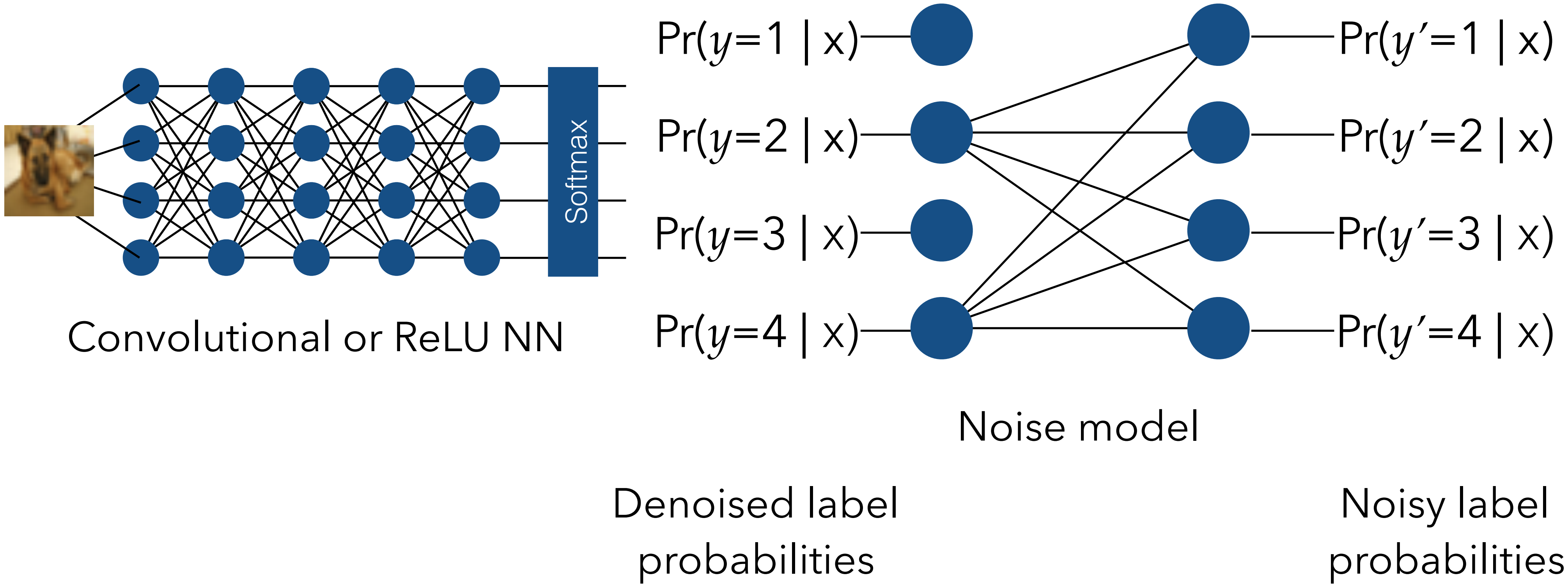
Dropout Regularization



- 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

[4] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," 2014.

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

[4] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," 2014.

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): $\Psi = (1-p)\mathbf{I} + \frac{p}{C}\mathbf{1}\mathbf{1}^T$
  - Non-uniform (Δ drawn from unit simplex): $\Psi = (1-p)\mathbf{I} + p\Delta$
- Use cross-entropy loss + dropout regularization

[4] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," 2015.

Simulation Results: Uniform Noise

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

Noise Level p	Standard CNN	Dropout	[3]	Genie-aided	Noise-free
30%	29.78%	**24.43%**	26%	25.76%	20.49%
50%	38.76%	**32.64%**	35%	29.63%	20.49%
70%	48.34%	**33.00%**	63%	36.24%	20.49%

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

[3] S. Sukhbaatar et al., "Training convolutional networks with noisy labels," 2014.

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

Noise Level *p*	Standard CNN	Dropout	[3]	Genie-aided	Noise-free
30%	30.49%	**25.4%**	26%	24.95%	20.49%
50%	39.47%	**31.28%**	35%	29.9%	20.49%
70%	65.6%	**63.04%**	63%	63.91%	20.49%

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

[3] S. Sukhbaatar et al., "Training convolutional networks with noisy labels," 2014.

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`