Making deep neural networks robust to label noise: a loss correction approach

Giorgio Patrini
23 July 2017
CVPR, Honolulu

joint work with
Alessandro Rozza, Aditya Krishna Menon, Richard Nock and Lizhen Qu
ANU, Data61, Waynaut, University of Sydney

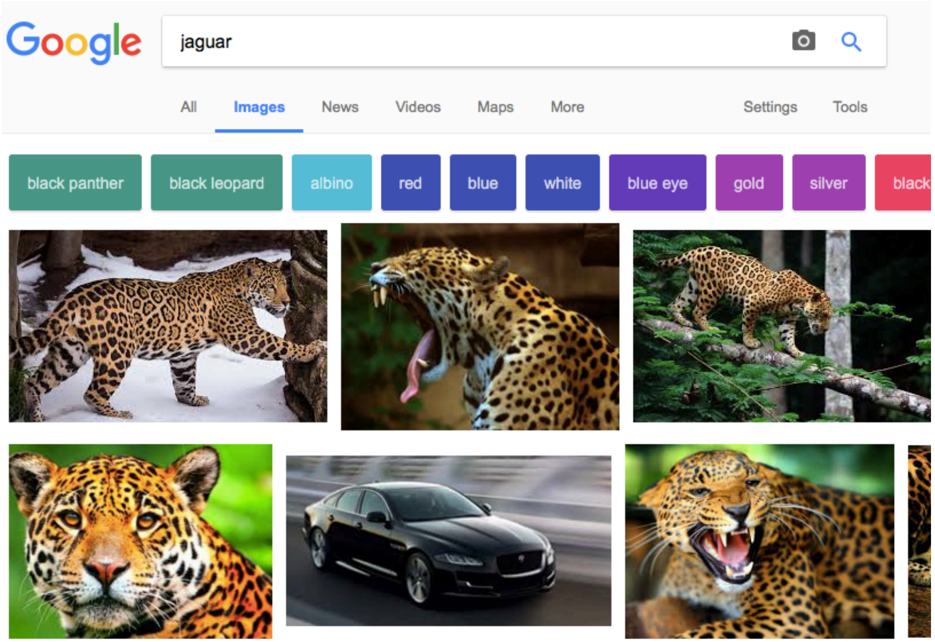
Label noise: motivations

“Data science becomes the art of extracting **labels** out of thin air” [Malach & Shalev-Shwartz 17]

Label noise: motivations

"Data science becomes the art of extracting **labels** out of thin air" [Malach & Shalev-Shwartz 17]

Labels from Web queries



Crowd sourcing

: ? : jaguar : leopard : cheetah

Previous work (sample)

- Noise-aware deep nets (CV)
  - Good performance on specific domains, scalable
  - Heuristics
  - In many cases, need some clean labels

  [Sukhbaatar et al. ICLR15, Krause et al. ECCV16, Xiao et al. CVPR15]

- Theoretically robust loss functions (ML)
  - Theoretically sound
  - Unrealistic assumptions… knowing the noise distribution!

  [Natarajan et al. NIPS13, Patrini et al. ICML16]

- Estimating the noise from noisy data

  [Menon et al. ICML15]

Contributions

- **Two procedures for loss correction**. Loss/architecture/dataset agnostic.
- Theoretical guarantee: same model as without noise (in expectation).
- Noise estimation, by using the same deep net.
- Tests on MNIST, CIFAR10/100, IMDB with multiple nets (CNN, ResNets, LSTM, …). SOTA on data of [Xiao et al. 15].

Supervised learning

- Sample from $p(\boldsymbol{x}, \boldsymbol{y})$
- c-class classification: $\boldsymbol{y} \in \{\boldsymbol{e}^j : j = 1, \ldots, c\}$
- Learn a neural network $p(\boldsymbol{y}|\boldsymbol{x})$

Supervised learning

- Sample from $p(\boldsymbol{x}, \boldsymbol{y})$
- c-class classification: $\boldsymbol{y} \in \{\boldsymbol{e}^j : j = 1, \ldots, c\}$
- Learn a neural network $p(\boldsymbol{y}|\boldsymbol{x})$
- Minimize the empirical risk associated with loss $\ell(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x}))$:

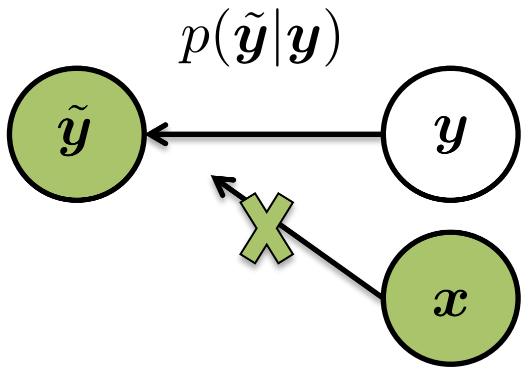
$$\operatorname*{argmin}_{p(\boldsymbol{y}|\boldsymbol{x})} \mathbb{E}_S \ \ell(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x}))$$

- Let $\boldsymbol{\ell}(p(\boldsymbol{y}|\boldsymbol{x})) = \left(\ell(\boldsymbol{e}^1, p(\boldsymbol{y}|\boldsymbol{x})), \ldots, \ell(\boldsymbol{e}^c, p(\boldsymbol{y}|\boldsymbol{x}))\right)^\top$

Asymmetric label noise

- Sample from $p(\boldsymbol{x}, \tilde{\boldsymbol{y}})$
- Corruption by **asymmetric** noise, defined by a transition matrix $T \in [0,1]^{c \times c}$:

$$T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j | \boldsymbol{y} = \boldsymbol{e}^i)$$

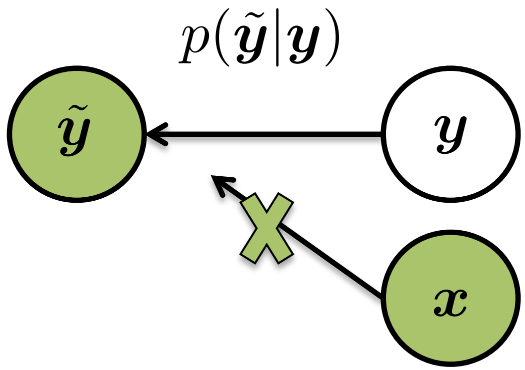


Feature independent noise

Asymmetric label noise

- Sample from $p(\boldsymbol{x}, \tilde{\boldsymbol{y}})$
- Corruption by **asymmetric** noise, defined by a transition matrix $T \in [0,1]^{c \times c}$:

$$T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j | \boldsymbol{y} = \boldsymbol{e}^i)$$



Feature independent noise

- How to be robust to such noise?

Backward loss correction

- c-class version of [Natarajan et al. 13]

$$\boldsymbol{\ell}^{\leftarrow}(p(\boldsymbol{y}|\boldsymbol{x})) = T^{-1}\boldsymbol{\ell}(p(\boldsymbol{y}|\boldsymbol{x}))$$

- **Rationale:** linear combination of losses, weighted by the inverse of the noise probabilities
- "One step back" in the Markov chain T

Backward loss correction: theory

- **Theorem:** if T is non-singular, $\boldsymbol{\ell}^{\leftarrow}$ is **unbiased**. It follows that the models learned with/without noise are the same under noise expectation:

$$\underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{y}}}\, \ell^{\leftarrow}(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x})) = \underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\, \ell(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x}))$$

Forward loss correction

- Inspired by [Sukhbaatar et al. 15]: "absorbs" the noise in a top linear layer, emulating T

$$\boldsymbol{\ell}^{\rightarrow}(p(\boldsymbol{y}|\boldsymbol{x})) = \boldsymbol{\ell}(T^{\top}p(\boldsymbol{y}|\boldsymbol{x}))$$

- **Rationale:** compare noisy labels with "noisified" predictions

Forward loss correction: theory

- **Theorem:** if T is non-singular, $\boldsymbol{\ell}^{\rightarrow}$ is such that the models with/without noise are the same under noise expectation* :

$$\underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{y}}} \; \ell^{\rightarrow}(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x})) = \underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \; \ell(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x}))$$

* Technically, the loss needs to be **proper composite** here. Cross-entropy and square are OK.

Noise estimation

- c-class extension of [Menon et al. 15]

Noise estimation

- c-class extension of [Menon et al. 15]
- **Hp:** there are some "perfect examples", and the net can model $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ very well

Noise estimation

- c-class extension of [Menon et al. 15]
- **Hp:** there are some "perfect examples", and the net can model $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ very well
- First, train and get $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$
- Then estimate $\hat{T}$ by

$$\forall i, j \quad \left[\begin{array}{l} \bar{\boldsymbol{x}}^i = \operatorname*{argmax}_{\boldsymbol{x}} p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^i|\boldsymbol{x}) \\ T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j|\bar{\boldsymbol{x}}^i) \end{array} \right.$$

Noise estimation

- c-class extension of [Menon et al. 15]
- **Hp:** there are some "perfect examples", and the net can model $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ very well
- First, train and get $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$
- Then estimate $\hat{T}$ by

$$\forall i, j \quad \left[\begin{array}{l} \bar{\boldsymbol{x}}^i = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^i|\boldsymbol{x}) \\ T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j|\bar{\boldsymbol{x}}^i) \end{array} \right.$$

- **Rationale:** mistakes on "perfect examples" must be due to the noise

Recap: the algorithm

(1) Train the network on noisy data to obtain $\hat{T}$

$$\underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{y}}} \; \ell(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x})) = p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) \to \hat{T}$$

(2) Re-train the network correcting with backward/forward loss, e.g.

$$\underset{p(\boldsymbol{y}|\boldsymbol{x})}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{y}}} \; \ell^{\leftarrow}(\boldsymbol{y}, p(\boldsymbol{y}|\boldsymbol{x}))$$

no change in back-propagation

Empirics: models and datasets

- **Goal:** show robustness independently from architecture and dataset

Simulated noise:

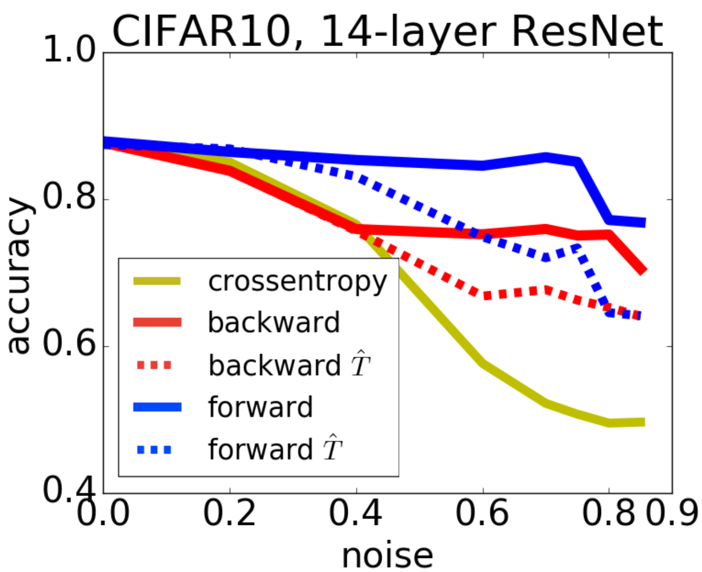
- MNIST: 2 x fully connected, dropout
- IMDB: word embedding + LSTM
- CIFAR10/100: various ResNets

Real noise:

- Clothing1M [Xiao et al. 15], 50-ResNet

Inject sparse, asymmetric T

$$T \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{.3} & 0 & 0 & 0 & 0 & \mathbf{.7} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{.3} & 0 & 0 & 0 & 0 & \mathbf{.7} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{.3} & \mathbf{.7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{.7} & \mathbf{.3} & 0 & 0 & 0 \\ 0 & \mathbf{.7} & 0 & 0 & 0 & 0 & 0 & \mathbf{.3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



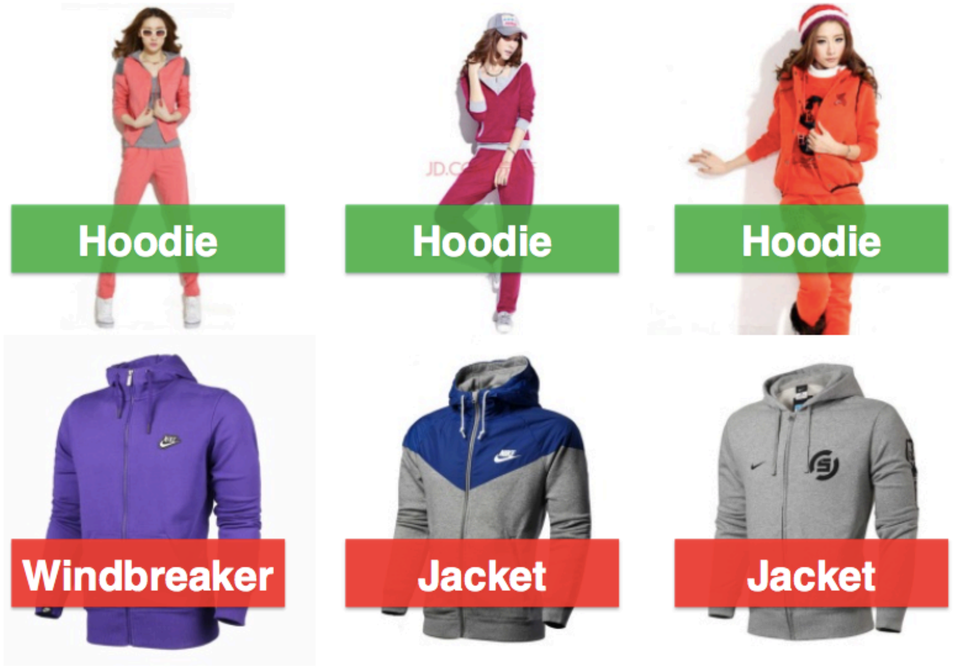
$$\hat{T} \quad \begin{bmatrix} 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \mathbf{.33} & \epsilon & \epsilon & \epsilon & \epsilon & \mathbf{.67} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \mathbf{.35} & \epsilon & \epsilon & \epsilon & \epsilon & \mathbf{.65} & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \mathbf{.29} & \mathbf{.71} & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \mathbf{.73} & \mathbf{.26} & \epsilon & \epsilon & \epsilon \\ \epsilon & \mathbf{.75} & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \mathbf{.25} & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 \end{bmatrix}$$

$\epsilon < 10^{-6}$

Experiments with real noise

Clothing1M [Xiao et al. CVPR15]

- Trainset:
  1M noisy label +
  50k clean labels
- Testset:
  10k clean labels



Experiments with real noise

Clothing1M					
#	model	loss	init	training	accuracy
1	AlexNet	cross-.	ImageNet	$50k$	72.63
2	AlexNet	cross-.	#1	$1M, 50k$	76.22
3	2x AlexNet	cross-.	#1	$1M, 50k$	78.24
4	50-ResNet	cross-	ImageNet	$1M$	68.94
5	50-ResNet	backward	ImageNet	$1M$	69.13
6	50-ResNet	forward	ImageNet	$1M$	69.84
7	50-ResNet	cross-.	ImageNet	$50k$	75.19
8	50-ResNet	cross-.	#6	$50k$	**80.38**

Our method

Recipe for SOTA:

- Pre-train: "forward loss" on 1M noisy labels
- Fine-tune: cross-entropy on 50k clean labels

Conclusions

Contributions

- End to end
- Theoretical guarantees
- In pair/better than previous work, SOTA on Clothing1M
- Forward better than backward (easier to optimize)

Limitations

- Noise estimation: **hard with massively multiclass**

Potential improvements

- Couple noise estimation with training [Xiao et al. 15, Goldberger & Ben-Reuven 17, Veit et al. 17]

References

H. Masnadi-Shirazi, N. Vasconcelos, **On the design of loss function for classification: theory, robustness to outliers, and savageboost**, NIPS09

N. Natarajan, I. S. Dhillon, P. Ravikumar, A. Tewari, **Learning with noisy labels**, NIPS13

S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, A. Rabinovich, **Training deep neural networks on noisy labels with bootstrapping**, arXiv14

A. Ghosh, N. Manwani, P. S. Sastry, **Making risk minimization tolerant to label noise**, Neurocomputing15

S, Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, **Training convolutional neural networks with noisy labels**, ICLR15 workshop

A. K. Menon, B. van Rooyen, C. S. Ong, R. C. Williamson, **Learning from corrupted binary labels via class-probability estimation**, ICML15

T. Xiao, T. Xia, T. Yang, X. Huang, X. Wang, **Learning from massive noisy labeled data for image classification**, CVPR15

B. Van Rooyen, A. K. Menon, R. C. Williamson, **Learning with symmetric label noise: the importance of being unhinged**, NIPS15

G. Patrini, F. Nielsen, R. Nock, M. Carioni, **Loss factorization, weakly supervised learning and label noise robustness**, ICML16

J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, L. Fei-Fei, **The unreasonable effect of noisy data for fine-grained recognition**, ECCV16

References, 2017

A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, S. Belongie, **Learning from noisy large-scale datasets with minimal supervision**, CVPR17

S. Yeung, V. Ramanathan, O. Russakovsky, L. Shen, G. Mori, L. Fei-Fei, **Learning to learn from noisy web videos**, CVPR17

J. Goldberger, E. Ben-Reuven, **Training deep neural-networks using a noise adaptation layer**, ICLR17

R. Wang, T. Liu, **Multiclass learning with partially corrupted labels**, IEEE transactions on neural networks and learning systems 17.

Y. Li, J. Yang, Y. Song, L. Cao, J. Li, **Learning from noisy labels with distillation**, arXiv17

A. Vahdat, **Toward robustness against label noise in training deep discriminative neural networks**, arXiv17

E. Malach, S. Shalev-Shwartz, **Decoupling "when to update" from "how to update"**, arXiv17

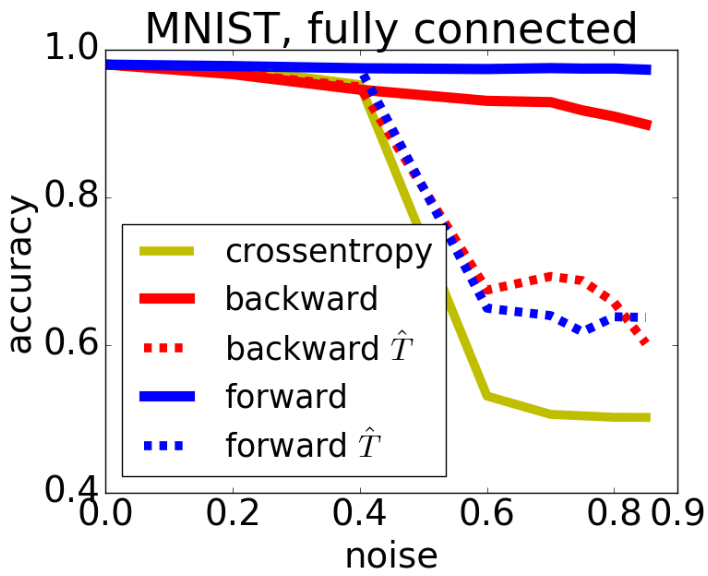
Example: cross-entropy

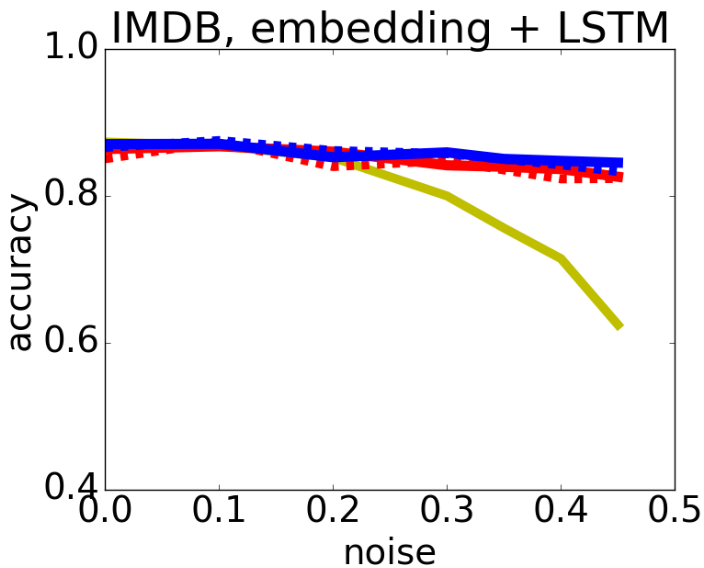
- cross-entropy (multi-class logistic)

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{softmax}(\mathrm{net}(\boldsymbol{x}))$$

$$\boldsymbol{y}^\top \boldsymbol{\ell}(p(\boldsymbol{y}|\boldsymbol{x})) = -\boldsymbol{y}^\top \log p(\boldsymbol{y}|\boldsymbol{x})$$

Inject sparse, asymmetric T





Compare with previous work

	CIFAR-10, 32-layer ResNet			
	NO NOISE	SYMM. $N = 0.2$	ASYMM. $N = 0.2$	ASYMM. $N = 0.6$
cross-entropy	90.1	86.6	89.0	53.6
unhinged [van Rooyen et al., 15]	90.2	86.5	87.1	60.0
sigmoid [Ghosh et al., 15]	81.6	69.6	79.1	61.8
Savage [Masnadi-Shirazi et al., 09]	88.3	86.2	86.3	53.5
bootstrap soft [Reed et al., 14]	90.9	86.9	88.6	53.1
bootstrap hard [Reed et al., 14]	90.4	86.4	88.6	54.7
backward	90.1	83.0	84.4	74.3
backward, $\hat{T}$	90.8	86.9	86.4	66.7
forward	91.2	**87.7**	**89.9**	**87.6**
forward, $\hat{T}$	90.5	**87.9**	**90.1**	**77.6**

- Similar for CIFAR100, but estimating *high-intensity* noise is hard for 100 classes with 50k examples.