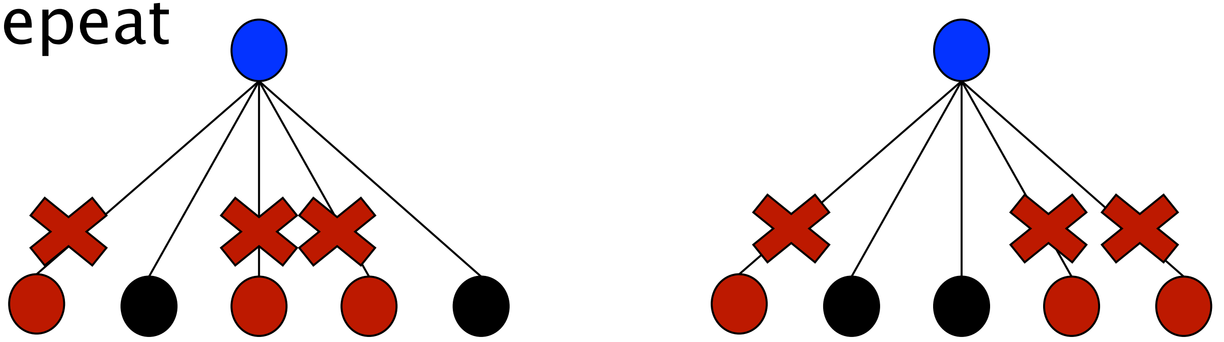
Semi-supervised Dropout Training

Baylearn 2013

Stefan Wager, Sida Wang, Percy Liang

The basics of dropout training

- Introduced by Hinton et al. in "Improving neural networks by preventing co-adaptation of feature detectors"
- For each example, randomly select features
  - zero them
  - compute the gradient, make an update
  - repeat



Empirically successful

- Dropout is important in some recent successes
  - won the ImageNet challenge [Krizhevsky et al., 2012]
  - won the Merck challenge [Dahl et al., 2012]
- Improved performance on standard datasets
  - images: MNIST, CIFAR, ImageNet, etc.
  - document classification: Reuters, IMDB, Rotten Tomatoes, etc.
  - speech: TIMIT, GlobalPhone, etc.

Lots of related works already

Variants

- DropConnect [Wan et al., 2013]
- Maxout networks [Goodfellow et al., 2013]

Analytical integration

- Fast Dropout [Wang and Manning, 2013]
- Marginalized Corrupted Features [van der Maaten et al., 2013]

Many other works report empirical gains

Theoretical understanding?

- **Dropout as adaptive regularization**
  - feature noising -> interpretable penalty term

$$\text{Loss(Dropout(data))}$$
$$= \text{Loss(data)} + \text{Regularizer(data)}$$

- **Semi-supervised learning**
  - feature dependent, label independent regularizer:

$$\text{Regularizer(Unlabeled data)}$$

Dropout for Log-linear Models

- Log likelihood (e.g., softmax classification):

$$\log p(y|x;\theta) = x^T\theta_y - A(x^T\theta)$$

$$\theta = [\theta_1, \theta_2, \ldots, \theta_K]$$

Dropout for Log-linear Models

- Log likelihood (e.g., softmax classification):

$$\log p(y|x;\theta) = x^T\theta_y - A(x^T\theta)$$

$$\theta = [\theta_1, \theta_2, \ldots, \theta_K]$$

- Dropout: $\tilde{x}_j = \begin{cases} 2x_j & \text{with p=0.5} \\ 0 & \text{otherwise} \end{cases}$ $\quad \mathbb{E}[\tilde{x}] = x$

- Dropout objective:

$$\underbrace{\mathbb{E}[\log p(y|\tilde{x};\theta)]}_{\text{-Loss(Dropout(data))}} = \mathbb{E}[\tilde{x}^T\theta_y] - \mathbb{E}[A(\tilde{x}^T\theta)]$$

Loss(data)+Regularizer(data)

Dropout for Log-linear Models

- We can rewrite the dropout log-likelihood

$$\mathbb{E}[\log p(y|\tilde{x};\theta)] = \mathbb{E}[\tilde{x}^T\theta_y] - \mathbb{E}[A(\tilde{x}^T\theta)]$$

$$\log p(y|x;\theta) = x^T\theta_y - A(x^T\theta)$$

$$\underbrace{\mathbb{E}[\log p(y|\tilde{x};\theta)]}_{\text{-Loss(Dropout(data))}} = \underbrace{\log p(y|x;\theta)}_{\text{-Loss(data)}} - (\underbrace{\mathbb{E}[A(\tilde{x}^T\theta)] - A(x^T\theta)}_{\text{Regularizer(data)}})$$

- Dropout reduces to a regularizer

$$R(\theta, x) = \mathbb{E}[A(\tilde{x}^T\theta)] - A(x^T\theta)$$

Second-order delta method

Take the Taylor expansion

$$A(s) \approx A(s_0) + (s - s_0)^T A'(s_0) + (s - s_0)^T \frac{A''(s_0)}{2} (s - s_0)$$

Second-order delta method

Take the Taylor expansion

$$A(s) \approx A(s_0) + (s - s_0)^T A'(s_0) + (s - s_0)^T \frac{A''(s_0)}{2}(s - s_0)$$

Substitute $s = \tilde{s} \stackrel{\text{def}}{=} \theta^T \tilde{x}$, $s_0 = \mathbb{E}[\tilde{s}]$

Take expectations to get the quadratic approximation:

$$\begin{aligned} R^{\mathrm{q}}(\theta, x) &= \frac{1}{2}\mathbb{E}[(\tilde{s} - \mathbf{s})^T \nabla^2 A(\mathbf{s})(\tilde{s} - \mathbf{s})] \\ &= \frac{1}{2}\mathrm{tr}(\nabla^2 A(\mathbf{s})\mathrm{Cov}(\tilde{s})) \end{aligned}$$

Example: logistic regression

- The quadratic approximation

$$R^{\mathrm{q}}(\theta, x) = \frac{1}{2} A''(x^T \theta) \mathrm{Var}[\tilde{x}^T \theta]$$

Example: logistic regression

- The quadratic approximation

$$R^{\mathrm{q}}(\theta, x) = \frac{1}{2} A''(x^T\theta)\mathrm{Var}[\tilde{x}^T\theta]$$

- $A''(x^T\theta) = p(1-p)$ represents uncertainty:

$$p = p(y|x;\theta) = (1 + \exp(-yx^T\theta))^{-1}$$

Example: logistic regression

- The quadratic approximation

$$R^{\mathrm{q}}(\theta, x) = \frac{1}{2} A''(x^T\theta)\mathrm{Var}[\tilde{x}^T\theta]$$

- $A''(x^T\theta) = p(1-p)$ represents uncertainty:

$$p = p(y|x;\theta) = (1 + \exp(-yx^T\theta))^{-1}$$

- $\mathrm{Var}[\tilde{x}^T\theta] = \sum_j \theta_j^2 x_j^2$ is L_2-regularization after normalizing the data

The regularizers

- Dropout on Linear Regression

$$R^q(\theta) = \frac{1}{2} \sum_j \theta_j^2 \sum_i x_j^{(i)2}$$

- Dropout on Logistic Regression

$$R^q(\theta) = \frac{1}{2} \sum_j \theta_j^2 \sum_i p_i(1 - p_i) x_j^{(i)2}$$

- Multiclass, CRFs [Wang et al., 2013]

Dropout intuition

$$R^q(\theta) = \frac{1}{2}\sum_j \theta_j^2 \sum_i p_i(1-p_i)x_j^{(i)2}$$

- Regularizes "rare" features less, like AdaGrad: there is actually a more precise connection [Wager et al., 2013]
- Big weights are okay if they contribute only to confident predictions
- Normalizing by the diagonal Fisher information

Semi-supervised Learning

- These regularizers are label-independent
  - but can be data adaptive in interesting ways
  - labeled dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$
  - unlabeled data $\mathcal{D}_{\text{unlabeled}} = \{u_1, u_2, \ldots, u_n\}$
- We can better estimate the regularizer

$$R_*(\theta, \mathcal{D}, \mathcal{D}_{\text{unlabeled}}) \stackrel{\text{def}}{=} \frac{n}{n + \alpha m} \Big(\sum_{i=1}^{n} R(\theta, x_i) + \alpha \sum_{i=1}^{m} R(\theta, u_i) \Big).$$

for some tunable α.

Semi-supervised intuition

$$R^q(\theta) = \frac{1}{2} \sum_j \theta_j^2 \sum_i p_i(1 - p_i) x_j^{(i)2}$$

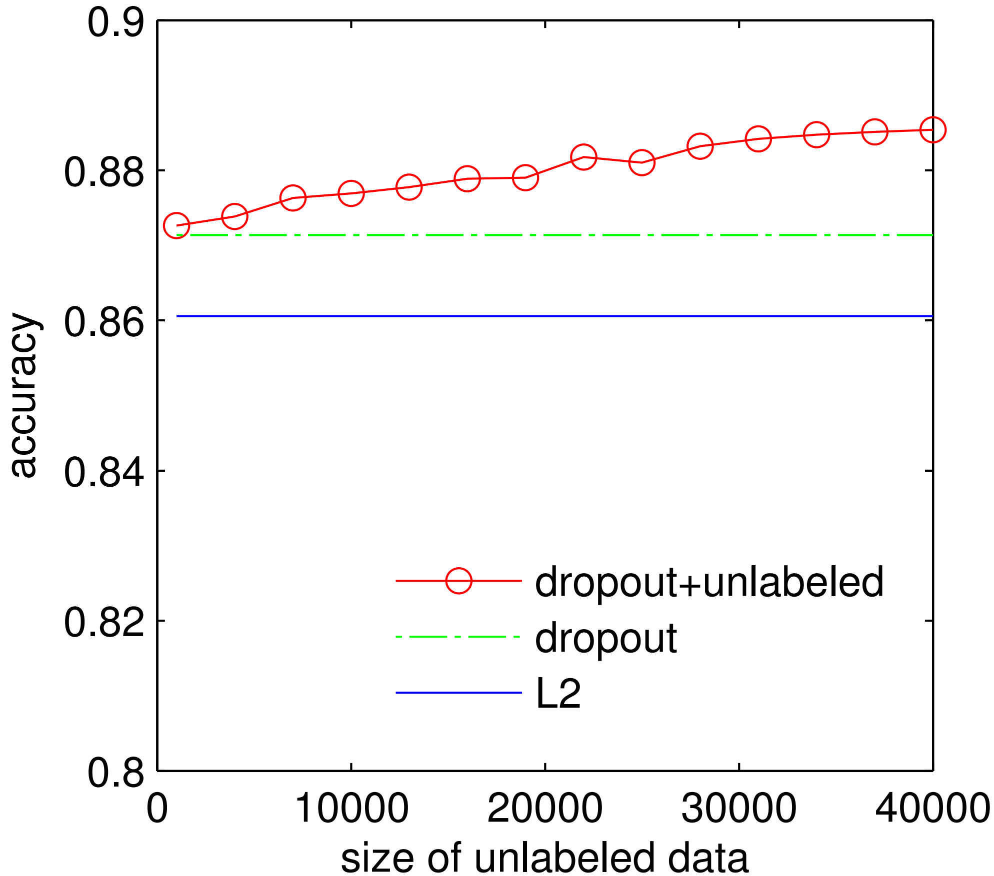
- Like other semi-supervised methods:
  - transductive SVMs [Joachims, 1999]
  - entropy regularization [Grandvalet and Bengio, 2005]
  - EM: guess a label [Nigam et al., 2000]
  - want to make confident predictions on the unlabeled data
- Get a better estimate of the Fisher information

IMDB dataset [Maas et al., 2011]

- 25k examples of positive reviews
- 25k examples of negative reviews
- Half for training and half for testing
- 50k unlabeled reviews also containing neutral reviews
- 300k sparse unigram features
- ~5 million sparse bigram features

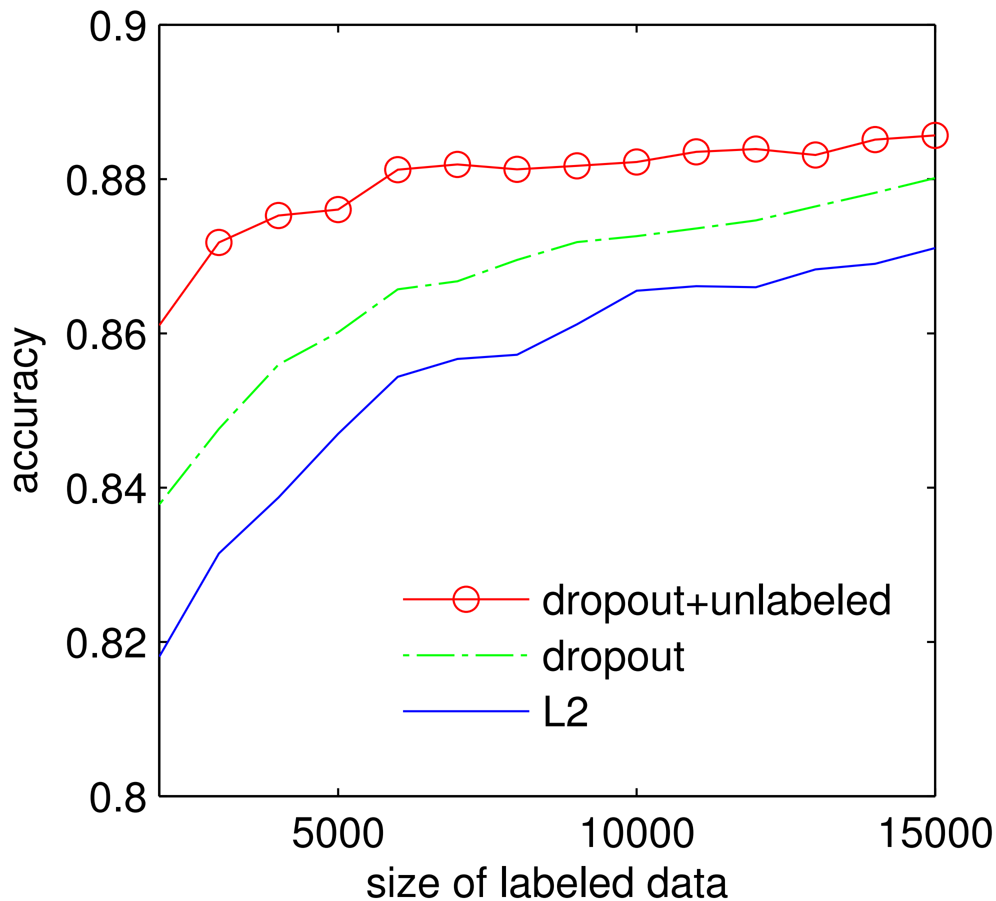
Experiments: semi-supervised

- Add more unlabeled data (10k labeled) improves performance



Experiments: semi-supervised

- Add more labeled data (40k unlabeled) improves performance



Quantitative results on IMDB

Method \ Settings	Supervised	Semi-sup.
MNB - unigrams with SFE [Su et al., 2011]	83.62	84.13
Vectors for sentiment analysis [Maas et al., 2011]	88.33	88.89
This work: dropout + unigrams	87.78	89.52
This work: dropout + bigrams	91.31	**91.98**

Experiments: other datasets

Dataset \ Settings	L_2	Drop	+Unlbl
Subjectivity [Peng and Lee, 2004]	88.96	90.85	91.48
Rotten Tomatoes [Peng and Lee, 2005]	73.49	75.18	76.56
20-newsgroups	82.19	83.37	84.71
CoNLL-2003	80.12	80.90	81.66

Advertisements

- Our arXiv paper [Wager et al., 2013] has more details, including the relation to AdaGrad
- Our EMNLP paper [Wang et al., 2013] extends this framework to structured prediction
- Our ICML paper [Wang and Manning, 2013] applies a related technique to neural networks and provides some negative examples

CRF sequence tagging

- CoNLL 2003 Named Entity Recognition
  - Facebook[ORG] is[O] hosting[O] Baylearn[MISC] in[O] Menlo[LOC] Park[LOC]

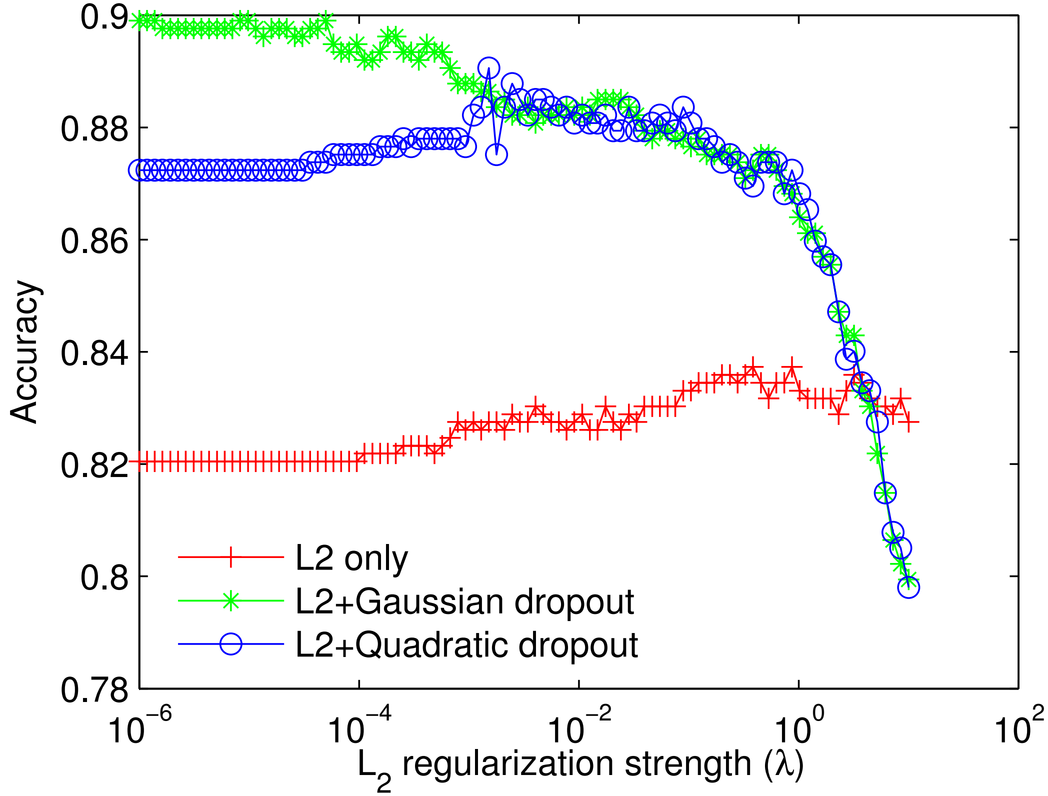
Dataset \ Settings	None	L_2	Drop
CoNLL 2003 Dev	89.40	90.73	91.86
CoNLL 2003 Test	84.67	85.82	87.42

Advertisements

- Our arXiv paper [Wager et al., 2013] has more details, including the relation to AdaGrad
- Our EMNLP paper [Wang et al., 2013] extends this framework to structured prediction
- Our ICML paper [Wang and Manning, 2013] applies a related technique to neural networks and provides some negative examples
- Thanks! Any questions?

Dropout vs. L_2

- Can be much better than all settings of L_2
- Part of the gain comes from normalization



Example: linear least squares

- The loss function is $f(\theta \cdot x) = 1/2(\theta \cdot x - y)^2$
- Let $X = \theta \cdot \tilde{x}$ where $\tilde{x}_j = 2z_j x_j$, $z_j = \mathrm{Bernoulli}(0.5)$

$$\mathbb{E}[f(X)] = f(\mathbb{E}[X]) + \frac{f''(\mathbb{E}[X])}{2}\mathrm{Var}[X]$$

$$= 1/2(\theta \cdot x - y)^2 + 1/2 \sum_j x_j^2 \theta_j^2$$

- The total regularizer is

$$R^q(\theta) = \frac{1}{2} \sum_j \theta_j^2 \sum_i x_j^{(i)2}$$

- This is just L2 applied after data normalization

Quantitative results on IMDB

Method \ Settings	Supervised	Semi-sup.
MNB - unigrams with SFE [Su et al., 2011]	83.62	84.13
MNB – bigrams	86.63	86.98
Vectors for sentiment analysis [Maas et al., 2011]	88.33	88.89
NBSVM – bigrams [Wang and Manning, 2012]	91.22	-
This work: dropout + unigrams	87.78	89.52
This work: dropout + bigrams	91.31	**91.98**