New Approaches to Training and Analysing Deep Networks

John Shawe-Taylor[1]

[1]Department of Computer Science
University College London

CAp, Rouen, 2018

Joint work with: Omar Rivasplata, Gaurav Singh, Csaba Szepesvari, Emilio Parrado-Hernandez, Shiliang Sun

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks

PART II

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks
- Link between stability and data distribution priors that could point the way to further analysis of stable learning, here for SVMs

PART II

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks
- Link between stability and data distribution priors that could point the way to further analysis of stable learning, here for SVMs
- Gives tighter bounds based on data distribution defined prior

PART II

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks
- Link between stability and data distribution priors that could point the way to further analysis of stable learning, here for SVMs
- Gives tighter bounds based on data distribution defined prior

PART II

- Exploiting iid property of training data to learn improved update directions from mini-batches

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks
- Link between stability and data distribution priors that could point the way to further analysis of stable learning, here for SVMs
- Gives tighter bounds based on data distribution defined prior

PART II

- Exploiting iid property of training data to learn improved update directions from mini-batches
- Can be used to improve efficiency of use of data

Background

PART I

- Renewed interest in stability in connection with Stochastic Gradient Descent for training Deep Networks
- Link between stability and data distribution priors that could point the way to further analysis of stable learning, here for SVMs
- Gives tighter bounds based on data distribution defined prior

PART II

- Exploiting iid property of training data to learn improved update directions from mini-batches
- Can be used to improve efficiency of use of data
- Indicates a new approach to obtaining generalisation bounds

Problems aim to address

- Require large amounts of data for training

Problems aim to address

- Require large amounts of data for training
- Require a lot of iterations to reach convergence

Problems aim to address

- Require large amounts of data for training
- Require a lot of iterations to reach convergence
- It is tough to obtain generalization in deep models

Problems aim to address

- Require large amounts of data for training
- Require a lot of iterations to reach convergence
- It is tough to obtain generalization in deep models
- Learned models are not robust to adversarial noise (attacks)

Form of the PAC-Bayes SVM bound

- Note that bound holds for all posterior distributions so that we can choose μ to optimise the bound

Form of the PAC-Bayes SVM bound

- Note that bound holds for all posterior distributions so that we can choose μ to optimise the bound
- If we define the inverse of the KL by

$$\mathrm{KL}^{-1}(q, A) = \max\{p : \mathrm{KL}(q\|p) \leq A\}$$

then have with probability at least $1 - \delta$ over the choice of the m sample

$$Pr\left(\langle \mathbf{w}, \phi(\mathbf{x})\rangle \neq y\right) \leq 2 \min_{\mu} \mathrm{KL}^{-1}\left(\mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x}, y))], \frac{\mu^2/2 + \ln \frac{m+1}{\delta}}{m}\right)$$

Definition of the Prior

- In PAC-Bayes we are free to choose the prior as long as it doesn't depend on the training data

Definition of the Prior

- In PAC-Bayes we are free to choose the prior as long as it doesn't depend on the training data
- Bound corresponds to prior at the origin

Definition of the Prior

- In PAC-Bayes we are free to choose the prior as long as it doesn't depend on the training data
- Bound corresponds to prior at the origin
- Can use part of the data to estimate a better prior and then evaluate the bound on the remaining data

Model Selection with the new bound: setup

- Comparison with X-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound

Model Selection with the new bound: setup

- Comparison with X-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets

Model Selection with the new bound: setup

- Comparison with X-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)

Model Selection with the new bound: setup

- Comparison with X-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)
  - For X-F XV select the pair that minimize the validation error

Model Selection with the new bound: setup

- Comparison with X-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)
  - For X-F XV select the pair that minimize the validation error
  - For PAC-Bayes Bound and Prior PAC-Bayes Bound select the pair that minimize the bound

Results

		Classifier					
		SVM				ηPrior SVM	
Problem		2FCV	10FCV	PAC	PrPAC	PrPAC	τ-PrPAC
digits	Bound	–	–	0.175	0.107	0.050	0.047
	CE	0.007	0.007	0.007	0.014	0.010	0.009
waveform	Bound	–	–	0.203	0.185	0.178	0.176
	CE	0.090	0.086	0.084	0.088	0.087	0.086
pima	Bound	–	–	0.424	0.420	0.428	0.416
	CE	0.244	0.245	0.229	0.229	0.233	0.233
ringnorm	Bound	–	–	0.203	0.110	0.053	0.050
	CE	0.016	0.016	0.018	0.018	0.016	0.016
spam	Bound	–	–	0.254	0.198	0.186	0.178
	CE	0.066	0.063	0.067	0.077	0.070	0.072

Defining the prior through the data distribution

- The idea of using a data distribution defined prior was pioneered by Catoni who looked at these distributions:

Defining the prior through the data distribution

- The idea of using a data distribution defined prior was pioneered by Catoni who looked at these distributions:
- P and Q are Gibbs-Boltzmann distributions

$$p(h) := \frac{1}{Z'} e^{-\gamma \mathrm{risk}(h)} \qquad q(h) := \frac{1}{Z} e^{-\gamma \widehat{\mathrm{risk}}_{\mathcal{S}}(h)}$$

Defining the prior through the data distribution

- The idea of using a data distribution defined prior was pioneered by Catoni who looked at these distributions:
- P and Q are Gibbs-Boltzmann distributions

$$p(h) := \frac{1}{Z'} e^{-\gamma \mathrm{risk}(h)} \qquad q(h) := \frac{1}{Z} e^{-\gamma \widehat{\mathrm{risk}}_S(h)}$$

- These distributions are hard to work with since we cannot apply the bound to a single weight vector, but the bounds can be very tight:

$$KL_+(\hat{Q}_S(\gamma) || Q_{\mathcal{D}}(\gamma)) \leq \frac{1}{m}\left(\frac{\gamma}{\sqrt{m}} \sqrt{\ln \frac{8\sqrt{m}}{\delta}} + \frac{\gamma^2}{4m} + \ln \frac{4\sqrt{m}}{\delta} \right)$$

as it appears we can choose γ small even for complex classes.

Data distribution dependent prior

- Let's try something simple to motivate the idea

Data distribution dependent prior

- Let's try something simple to motivate the idea
- Consider the Gaussian prior centred on the weight vector:

$$\mathbf{w}_p = \mathbb{E}[y\phi(\mathbf{x})]$$

Data distribution dependent prior

- Let's try something simple to motivate the idea
- Consider the Gaussian prior centred on the weight vector:

$$\mathbf{w}_p = \mathbb{E}[y\phi(\mathbf{x})]$$

- Note that we do not know this vector, but it is nonetheless fixed independently of the training sample.

Data distribution dependent prior

- Let's try something simple to motivate the idea
- Consider the Gaussian prior centred on the weight vector:

$$\mathbf{w}_p = \mathbb{E}[y\phi(\mathbf{x})]$$

- Note that we do not know this vector, but it is nonetheless fixed independently of the training sample.
- We can compute a sample based estimate of this vector as

$$\hat{\mathbf{w}}_p = \mathbb{E}_S[y\phi(\mathbf{x})]$$

Estimating the KL divergence

- With probability $1 - \delta/2$ we have

$$\|\hat{\mathbf{w}}_p - \mathbf{w}_p\| \leq \frac{R}{\sqrt{m}} \left(2 + \sqrt{2 \ln \frac{2}{\delta}}\right).$$

Estimating the KL divergence

- With probability $1 - \delta/2$ we have

$$\|\hat{\mathbf{w}}_p - \mathbf{w}_p\| \leq \frac{R}{\sqrt{m}} \left(2 + \sqrt{2 \ln \frac{2}{\delta}}\right).$$

- Proof relies on independence of examples and the fact the vector is a simple sum

Estimating the KL divergence

- With probability $1 - \delta/2$ we have

$$\|\hat{\mathbf{w}}_p - \mathbf{w}_p\| \leq \frac{R}{\sqrt{m}} \left(2 + \sqrt{2 \ln \frac{2}{\delta}}\right).$$

- Proof relies on independence of examples and the fact the vector is a simple sum
- We can therefore w.h.p. upper bound KL divergence between prior P, an isotropic Gaussian at $\mathbf{w}_p$, and posterior Q, an isotropic Gaussian at $\mathbf{w}$ by

$$\frac{1}{2} \left(\|\mathbf{w} - \hat{\mathbf{w}}_p\| + \frac{R}{\sqrt{m}} \left(2 + \sqrt{2 \ln \frac{2}{\delta}}\right) \right)^2$$

Resulting bound

- Giving the following bound on generalisation:

$$KL_+(\hat{Q}_S(\mathbf{w},\mu)||Q_{\mathcal{D}}(\mathbf{w},\mu)) \leq \frac{\frac{1}{2}\left(\|\mu\mathbf{w}-\eta\hat{\mathbf{w}}_p\|+\eta\frac{R}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2}{\delta}}\right)\right)^2+\ln\frac{2(m+1)}{\delta}}{m}$$

  with probability $1-\delta$.

- Values of the bounds for an SVM.

Prob.	PAC-Bayes	PrPAC	τ-PrPAC	$\mathbb{E}$ PrPAC	τ-$\mathbb{E}$ PrPAC
han	0.175 $\pm$ 0.001	**0.107 $\pm$ 0.004**	**0.108 $\pm$ 0.005**	0.157 $\pm$ 0.001	0.176 $\pm$ 0.001
wav	0.203 $\pm$ 0.001	**0.185 $\pm$ 0.005**	**0.184 $\pm$ 0.005**	0.202 $\pm$ 0.001	0.205 $\pm$ 0.001
pim	0.424 $\pm$ 0.003	0.420 $\pm$ 0.015	0.423 $\pm$ 0.014	0.428 $\pm$ 0.003	0.433 $\pm$ 0.003
rin	0.203 $\pm$ 0.000	**0.110 $\pm$ 0.004**	**0.110 $\pm$ 0.004**	0.201 $\pm$ 0.001	0.204 $\pm$ 0.000
spa	0.254 $\pm$ 0.001	**0.198 $\pm$ 0.006**	**0.198 $\pm$ 0.006**	0.249 $\pm$ 0.001	0.255 $\pm$ 0.001

Expected SVM as prior

- Consider the Gaussian prior (with isotropic variance 1) centred on the weight vector:

$$\mathbf{w}_p = \mathbb{E}_{S\sim\mathcal{D}^m}[A_S]$$

Expected SVM as prior

- Consider the Gaussian prior (with isotropic variance 1) centred on the weight vector:

$$\mathbf{w}_p = \mathbb{E}_{S\sim\mathcal{D}^m}[A_S]$$

- Following Bousquet et al we use the SVM with hinge loss:

$$A_S = \arg\min_{\mathbf{w}} \frac{1}{m}\sum_{i=1}^{m} \ell(g_{\mathbf{w}}, (\mathbf{x}_i, y_i)) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \qquad (1)$$

Resulting bound

- We obtain a bound for which the KL term is $O(1/m^2)$: with probability $1-\delta$:

$$\mathrm{KL}_+(\hat{Q}_S(A_S,1)\|Q_{\mathcal{D}}(A_S,1)) \leq \frac{2}{\lambda^2 m^2}\left(1+\sqrt{\frac{1}{2}\ln\frac{1}{\delta}}\right)^2 + \\ + \frac{1}{m}\ln\left(\frac{m+1}{\delta}\right)$$

Resulting bound

- We obtain a bound for which the KL term is $O(1/m^2)$: with probability $1-\delta$:

$$\mathrm{KL}_+(\hat{Q}_S(A_S,1)||Q_{\mathcal{D}}(A_S,1)) \leq \frac{2}{\lambda^2 m^2}\left(1+\sqrt{\frac{1}{2}\ln\frac{1}{\delta}}\right)^2 + \\ +\frac{1}{m}\ln\left(\frac{m+1}{\delta}\right)$$

- Compared with Bousquet et al bound:

$$R \leq R_{\mathrm{emp}} + \frac{1}{\lambda m} + \left(1+\frac{2}{\lambda}\right)\sqrt{\frac{\ln(1/\delta)}{2m}}$$

Results

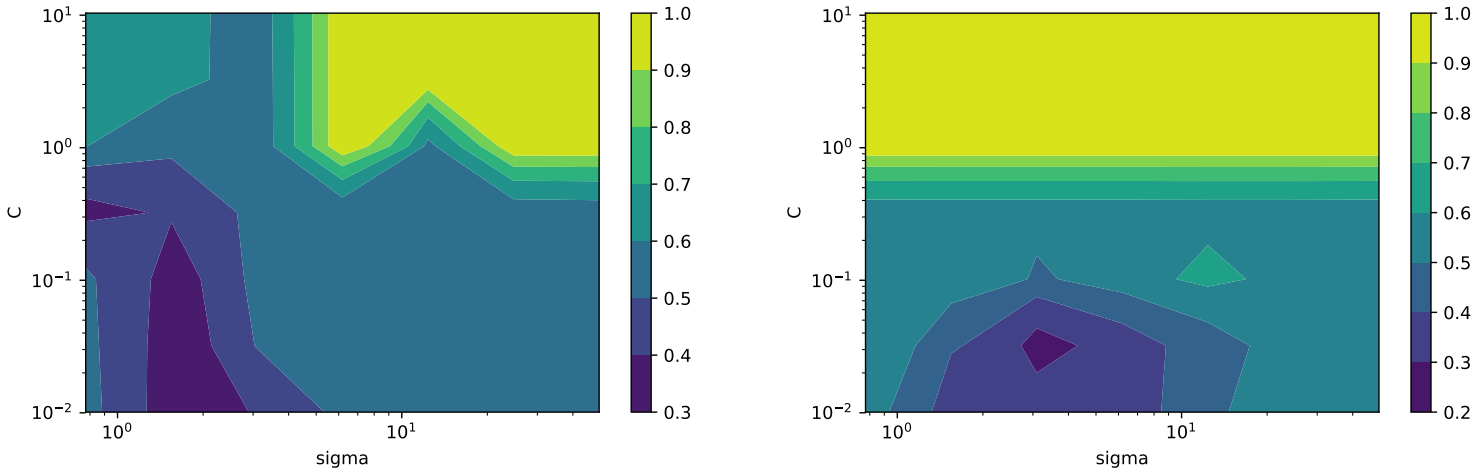
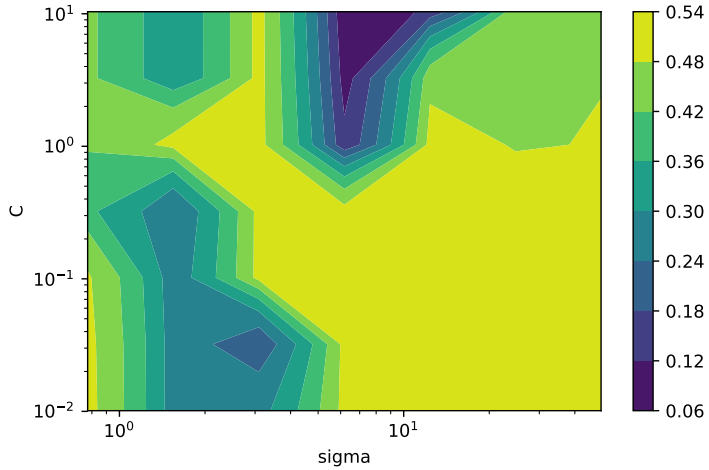


Figure: The PO bound (left) PEW bound (right) and on RIN with below test error



Implications

- Initial bounds depended on complexity of the function class

Implications

- Initial bounds depended on complexity of the function class
- Large margin (luckiness) depended on the link between the data generating distribution and the function class

Implications

- Initial bounds depended on complexity of the function class
- Large margin (luckiness) depended on the link between the data generating distribution and the function class
- Defining a prior in terms of the data generating distributions suggests that all that matters is how stable the learning is – not how complex the function class or margin is

Implications

- Initial bounds depended on complexity of the function class
- Large margin (luckiness) depended on the link between the data generating distribution and the function class
- Defining a prior in terms of the data generating distributions suggests that all that matters is how stable the learning is – not how complex the function class or margin is
- Need more data dependent measures of stability and extension to deep learning

PART II

Motivation

- Stochastic Gradient Descent uses mini-batches to derive noisy gradient estimates

Motivation

- Stochastic Gradient Descent uses mini-batches to derive noisy gradient estimates
- Simplest approach is average of gradients, but can include longer averages

Motivation

- Stochastic Gradient Descent uses mini-batches to derive noisy gradient estimates
- Simplest approach is average of gradients, but can include longer averages
- Conjugate gradient methods exploit second order information

Motivation

- Stochastic Gradient Descent uses mini-batches to derive noisy gradient estimates
- Simplest approach is average of gradients, but can include longer averages
- Conjugate gradient methods exploit second order information
- Can more be extracted from mini-batch gradients exploiting the fact that they represent an i.i.d. sample of the data distribution?

Key idea

- View weight update direction as a classifier of the mini-batch: correct classification if reducing its error, incorrect if increasing its error

Key idea

- View weight update direction as a classifier of the mini-batch: correct classification if reducing its error, incorrect if increasing its error
- This is ignoring second order effects: i.e. for small weight updates will hold

$$f^j(\mathbf{w} + \delta\mathbf{w}, \mathbf{x}_i) \approx f^j(\mathbf{w}, \mathbf{x}_i) + \langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x}_i) \rangle$$

Key idea

- View weight update direction as a classifier of the mini-batch: correct classification if reducing its error, incorrect if increasing its error
- This is ignoring second order effects: i.e. for small weight updates will hold

$$f^j(\mathbf{w} + \delta\mathbf{w}, \mathbf{x}_i) \approx f^j(\mathbf{w}, \mathbf{x}_i) + \langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x}_i) \rangle$$

- Have a target reduction of ϵ

Key idea

- View weight update direction as a classifier of the mini-batch: correct classification if reducing its error, incorrect if increasing its error
- This is ignoring second order effects: i.e. for small weight updates will hold

$$f^j(\mathbf{w} + \delta\mathbf{w}, \mathbf{x}_i) \approx f^j(\mathbf{w}, \mathbf{x}_i) + \langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x}_i) \rangle$$

- Have a target reduction of ϵ
- In order to minimise second order effects, we need to choose a minimum norm update that has the desired error reductions

Key idea

- View weight update direction as a classifier of the mini-batch: correct classification if reducing its error, incorrect if increasing its error
- This is ignoring second order effects: i.e. for small weight updates will hold

$$f^j(\mathbf{w} + \delta\mathbf{w}, \mathbf{x}_i) \approx f^j(\mathbf{w}, \mathbf{x}_i) + \langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x}_i) \rangle$$

- Have a target reduction of ϵ
- In order to minimise second order effects, we need to choose a minimum norm update that has the desired error reductions
- this can be translated into an optimisation that needs to be solved

Update optimisation to choose $\delta\mathbf{w}$

1: Minimise $\frac{1}{2}\|\delta\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\sum_{j=1}^{K}\xi_{ij}$
2: Subject to: $y_{ij}\langle\delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x}_i)\rangle \geq \epsilon - \xi_{ij}$ $\xi_{ij} \geq 0$, $i = 1, \ldots, \ell; j = 1, \ldots, K$

this is an SVM optimisation (with target margin ϵ).

The dual optimisation is

1: Max $\epsilon\sum_{ij}\alpha_{ij} - \frac{1}{2}\sum_{ijkl}\alpha_{ij}\alpha_{kl}\kappa((\mathbf{x}_i, j), (\mathbf{x}_k, l))$
2: Subject to: $C \geq \alpha_{ij} \geq 0$

where $\kappa((\mathbf{x}_i, j), (\mathbf{x}_k, l)) = \langle\nabla f^j(\mathbf{w}, \mathbf{x}_i), \nabla f^l(\mathbf{w}, \mathbf{x}_k)\rangle$

Analysis

- Since data generated iid, can use generalisation bounds to analyse the effects of the weight update

Analysis

- Since data generated iid, can use generalisation bounds to analyse the effects of the weight update
- The following bound holds

$$\mathbb{E}_{\mathcal{D}}[(\epsilon - y_j\langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x})\rangle)_+] \leq A = \frac{1}{\ell\epsilon}\sum_{ij}\xi_{ij} + \\ + \frac{4\|\delta\mathbf{w}\|}{\epsilon\ell}\sqrt{\mathrm{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

Analysis

- Since data generated iid, can use generalisation bounds to analyse the effects of the weight update
- The following bound holds

$$\mathbb{E}_{\mathcal{D}}[(\epsilon - y_j\langle \delta\mathbf{w}, \nabla f^j(\mathbf{w}, \mathbf{x})\rangle)_+] \leq A = \frac{1}{\ell\epsilon}\sum_{ij}\xi_{ij} + \\ + \frac{4\|\delta\mathbf{w}\|}{\epsilon\ell}\sqrt{\mathrm{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

- Hence, ignoring second order effects, for an $\eta\delta\mathbf{w}$ weight update, the average hinge loss across the whole training (and test) set will with high probability reduce by at least

$$\eta\,(\epsilon - A)$$

Optimization

$\eta = 0.1$; $r = 1.0$; $\ell =$ initial batch size
for $i \in [1, 2, \cdots, num\_iter]$ **do**
  $\mathcal{B}_s \leftarrow$ generateMinibatches($\mathcal{D}$, ℓ)
  $\delta\mathbf{w} \leftarrow$ trainMinibatch($\mathcal{B}_s$, $\mathcal{C}$, r)
  bound_term $= \|\delta\mathbf{w}\|\sqrt{tr(\mathbf{K})}/\ell$
  while bound_term $>$ threshold **do**
    $\ell \leftarrow 2 * \ell$ #minibatch size
    $r \leftarrow 0.1 * r$ #SVM Regularizer
    $\mathcal{B}_s \leftarrow$ generateMinibatches($\mathcal{D}$, ℓ)
    $\delta\mathbf{w} \leftarrow$ trainMinibatch($\mathcal{B}_s$, $\mathcal{C}$, r)
    bound_term $= \|\delta\mathbf{w}\|\sqrt{tr(\mathbf{K})}/\ell$
  end while
  $\mathbf{w} \leftarrow \mathbf{w} + \eta * \delta\mathbf{w}$
end for

Datasets

- MNIST: It consists of images of handwritten digits in binary. It has a training set of 60,000 examples, and a test set of 10,000 examples.
- CIFAR-10: It consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.

Convergence

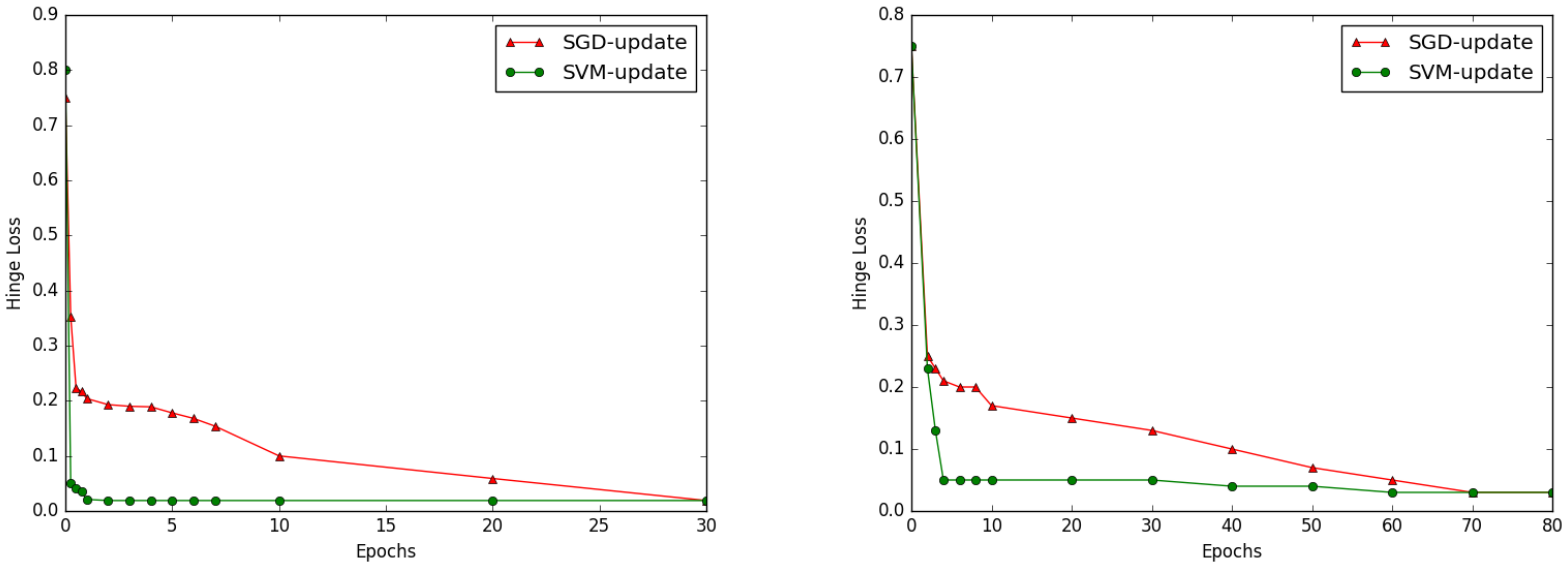


Figure: **Left**: MNIST & **Right**: CIFAR. We plot the hinge loss over train set versus epochs.

Convergence

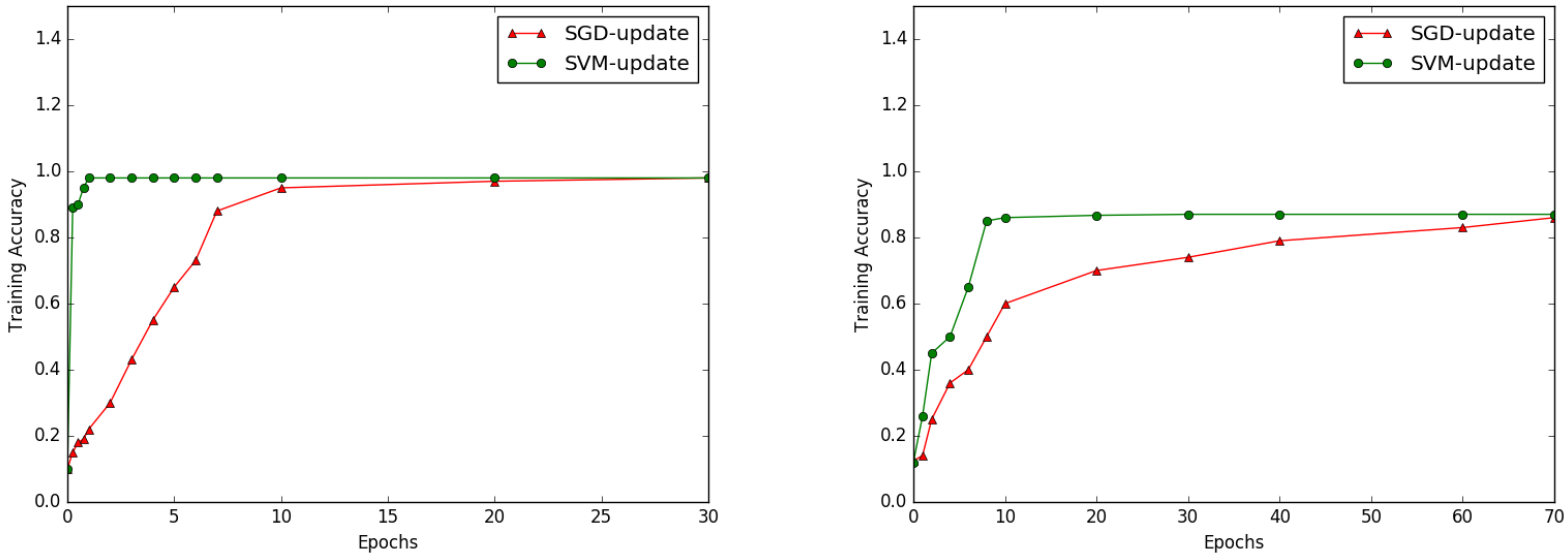
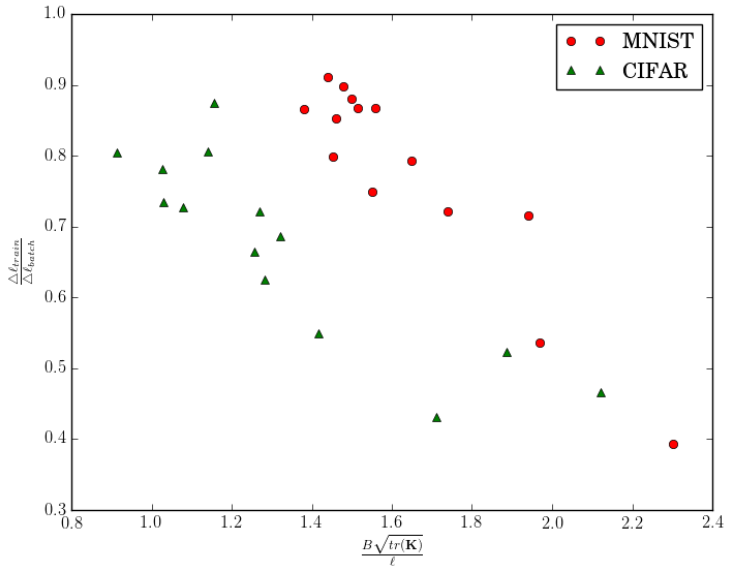


Figure: **Left**: MNIST & **Right**: CIFAR. We plot the training accuracy over the entire train set versus epochs.

Generalization



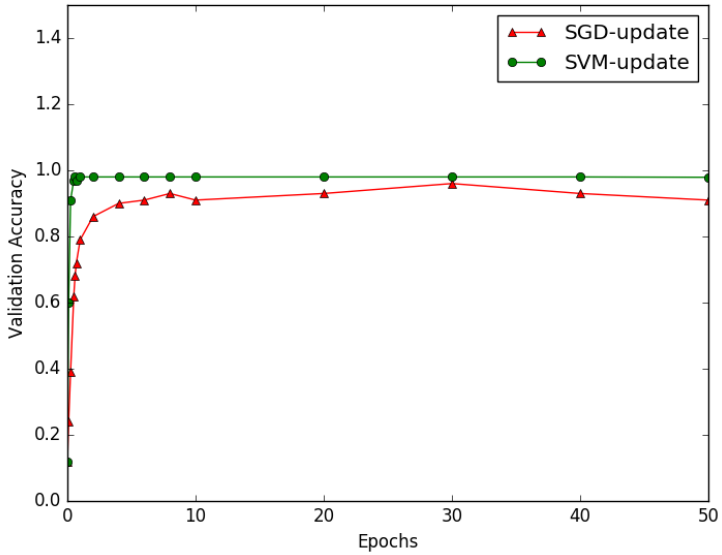


Figure: **Left**: We see that the ratio of decrease in loss over train set and mini-batch decreases with increase in the bound, implying that the updates become less generalized. **Right**: We observe that the validation accuracy initially increases and then stabilizes for mnist using our algorithm, as opposed to sgd.

Robustness: Adversarial Noise

	MNIST		CIFAR	
Norm	sgd-based	svm-based	sgd-based	svm-based
frobenius	5.12	7.32	7.95	9.25
infinity	5.90	8.11	6.50	8.57
nuclear	7.29	8.11	8.50	9.12
1-norm	6.23	7.59	7.40	9.10

Table: We give details of the additive adversarial noise learned for **left**: MNIST and **right**: CIFAR using traditional back-propagation and svm-based updates. Additive adversarial noise is the minimum amount of noise to be added to images such that the network misclassifies them.

Conclusion

- Have explored two novel approaches to analysing deep learning:
  - first explores data distribution defined priors that suggest new approach to understanding function class complexity
  - second looks at new way of selecting an update direction using a mini-batch of data

Conclusion

- Have explored two novel approaches to analysing deep learning:
  - first explores data distribution defined priors that suggest new approach to understanding function class complexity
  - second looks at new way of selecting an update direction using a mini-batch of data
- Provide new analysis of deep learning that potentially throw light on how it achieves good performance despite its very high complexity