Active Bias: Training a More Accurate Neural Network by Emphasizing High Variance Samples

UMASS CS

IESL

Haw-Shiuan Chang, Erik Learned-Miller, Andrew McCallum

Main Idea

Variance based active learning selects more informative training samples to annotate. Does it help if we emphasize those examples in SGD?

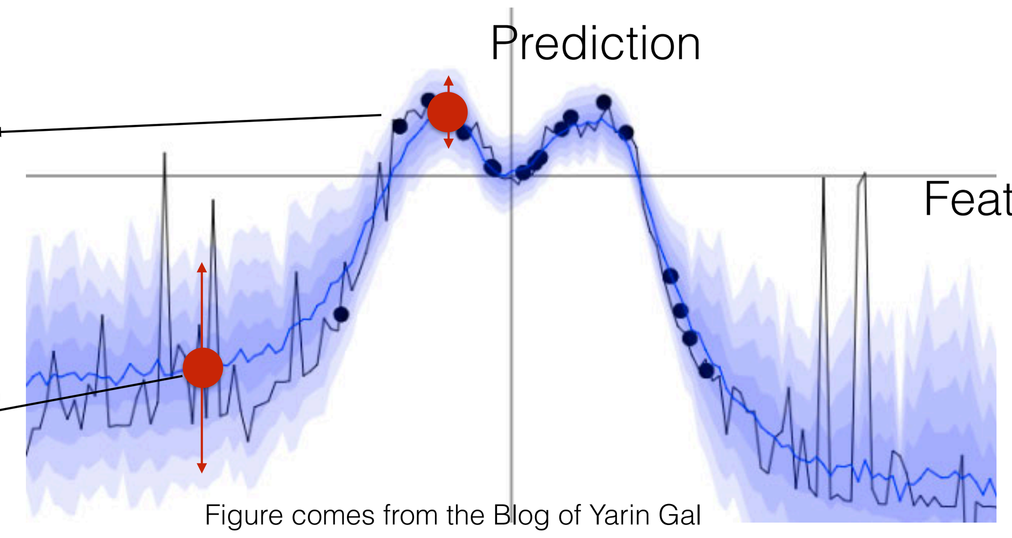
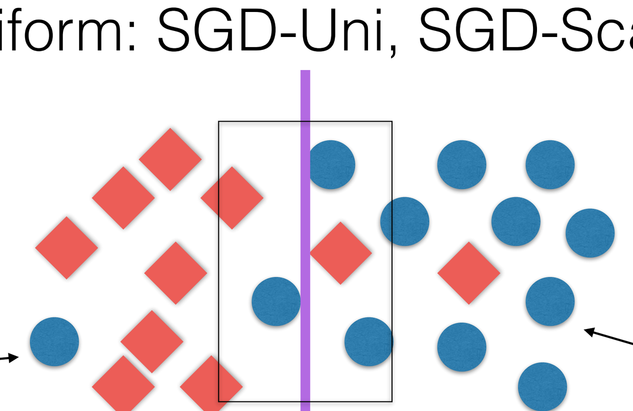


Figure comes from the Blog of Yarin Gal

Methods and Related Work

Uniform: SGD-Uni, SGD-Scan



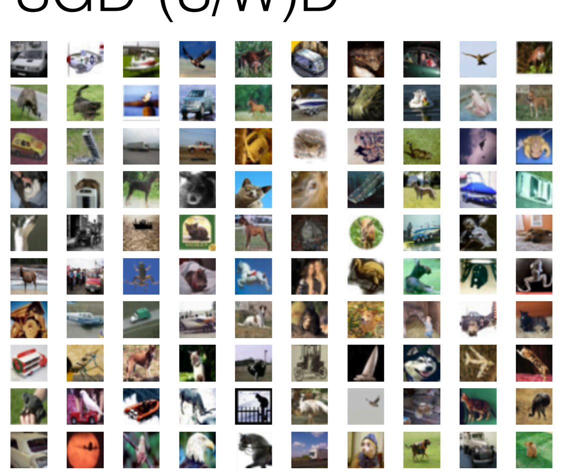


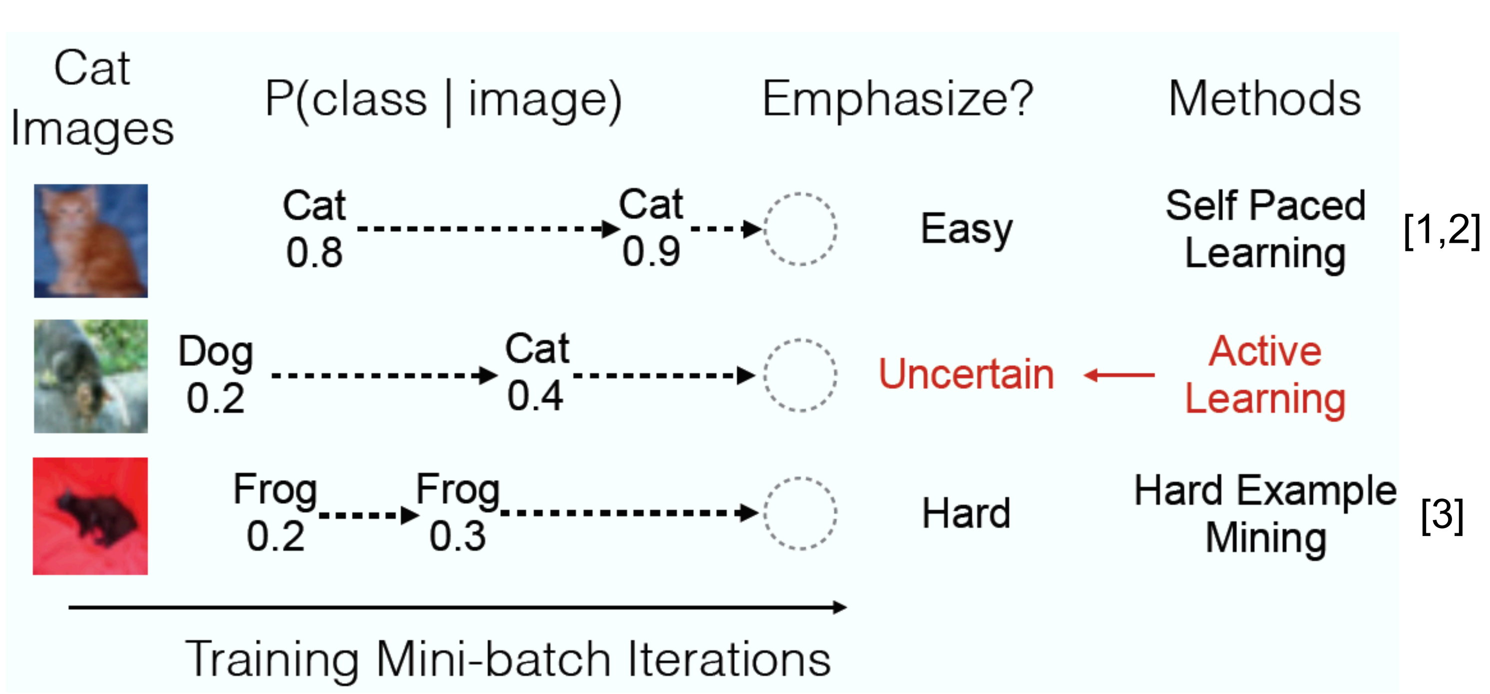
Figure comes from [4]

Figure comes from [4]

Uncertain:

SGD-(S/W)PV: Sampling/Weighting according to Prediction Variance

SGD-(S/W)TC: Sampling/Weighting according to Threshold Closeness



Example: Logistic Regression

Obj func: $-\log(Pr(Y,W=\mathbf{w}|X)) = -\sum_i \log(p(y_i|\mathbf{x_i},\mathbf{w})) - \frac{c}{s_0}||\mathbf{w}||^2$

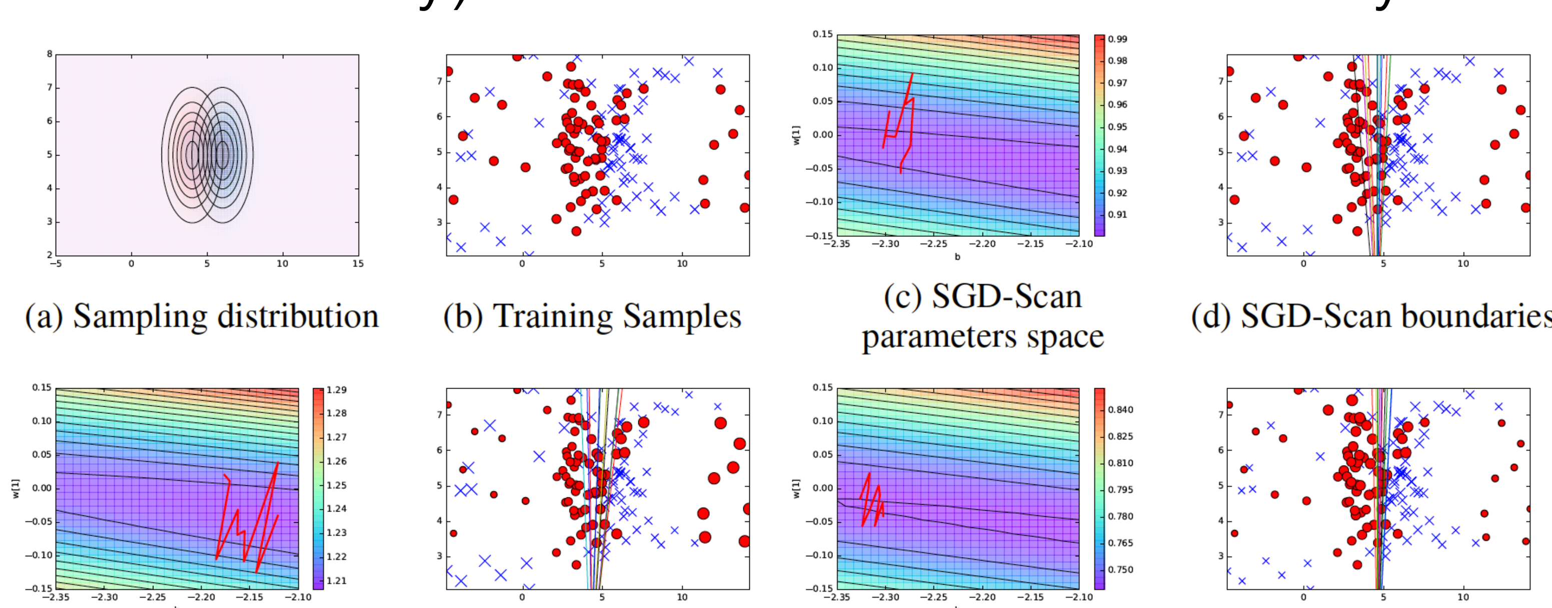
Assumption 1: $Pr(W = \mathbf{w}|Y,X) \approx \mathcal{N}(\mathbf{w}|\mathbf{w_N}, S_N)$

Assumption 2: $p(y_i|\mathbf{x_i},W) \approx p(y_i|\mathbf{x_i},\mathbf{w}) + g_i(\mathbf{w})^T(W-\mathbf{w})$

$$Var(p(y_i|\mathbf{x_i},W)) \approx g_i(\mathbf{w})^T S_N g_i(\mathbf{w})$$

$$g_i(\mathbf{w}) = p(y_i|\mathbf{x_i},\mathbf{w})\,(1-p(y_i|\mathbf{x_i},\mathbf{w}))\,\mathbf{x_i} \qquad S_N^{-1} = \sum_i p(y_i|\mathbf{x_i})\,(1-p(y_i|\mathbf{x_i}))\,\mathbf{x_i}\mathbf{x_i}^T + \frac{2c}{s_0}I$$

Weighting more uncertain examples (with high prediction variance or close to decision boundary) reduces classifier uncertainty.



(a) Sampling distribution (b) Training Samples (c) SGD-Scan parameters space (d) SGD-Scan boundaries

(e) SGD-WD parameters space (f) SGD-WD sample weights and boundaries (g) SGD-WPV parameters space (h) SGD-WPV sample weights and boundaries

Results

Table 3: The average of the best testing error rates for different sampling methods and datasets (%). The confidence intervals are standard errors. LR means logistic regression.

Datasets	Model	SGD-Uni	SGD-SD	SGD-ISD	SGD-SE	SGD-SPV	SGD-STC
MNIST	CNN	0.55±0.01	0.52±0.01	0.57±0.01	0.54±0.01	**0.51** ±0.01	**0.51**±0.01
Noisy MNIST	CNN	0.83±0.01	1.00±0.01	0.84±0.01	0.69 ±0.01	0.64±0.01	**0.63**±0.01
CIFAR 10	LR	62.49±0.06	63.14±0.06	62.48±0.07	60.87±0.06	**60.66**±0.06	61.00±0.06
QT	CNN	17.70±0.07	17.61±0.07	17.66±0.08	17.92±0.08	**17.49**±0.08	17.55±0.08

Table 4: The average of the best testing error rates and their standard errors for different weighting methods (%). For CoNLL 2003 and OntoNote 5.0, the values are 1-(F1 score). CNN, LR, RN 27, RN 63 and FC mean convolutional neural network, logistic regression, residual networks with 27 layers, residual network with 63 layers, and fully-connected network, respectively.

Datasets	Model	SGD-Scan	SGD-WD	SGD-WE	SGD-WPV	SGD-WTC
MNIST	CNN	0.54±0.01	**0.48**±0.01	0.56±0.01	**0.48**±0.01	**0.48**±0.01
Noisy MNIST	CNN	0.81±0.01	0.92±0.01	0.72±0.01	**0.61**±0.02	0.63±0.01
CIFAR 10	LR	62.48±0.06	63.10±0.06	60.88±0.06	**60.61**±0.06	61.02±0.06
CIFAR 100	RN 27	34.04±0.06	34.55±0.06	33.65±0.07	33.69±0.07	**33.64**±0.07
CIFAR 100	RN 63	30.70±0.06	31.57±0.09	**29.92**±0.09	30.02±0.08	30.16±0.09
QT	CNN	17.79±0.08	17.70±0.08	17.87±0.08	**17.57**±0.07	17.61±0.08
CoNLL 2003	CNN	11.62±0.04	11.50±0.05	11.73±0.04	11.24±0.06	**11.18**±0.03
OntoNote 5.0	CNN	17.80±0.05	17.65±0.06	18.40±0.05	17.82±0.03	**17.51**±0.05
MNIST	FC	2.85±0.03	**2.17**±0.01	3.08±0.03	2.68±0.02	2.34±0.03
MNIST (distill)	FC	2.27±0.01	2.13±0.02	2.35±0.01	2.18±0.02	**2.07**±0.02

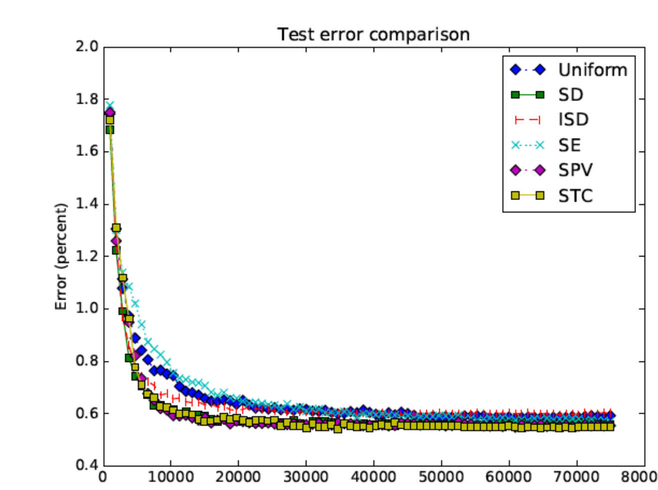


Figure 3: MNIST error rate (%)

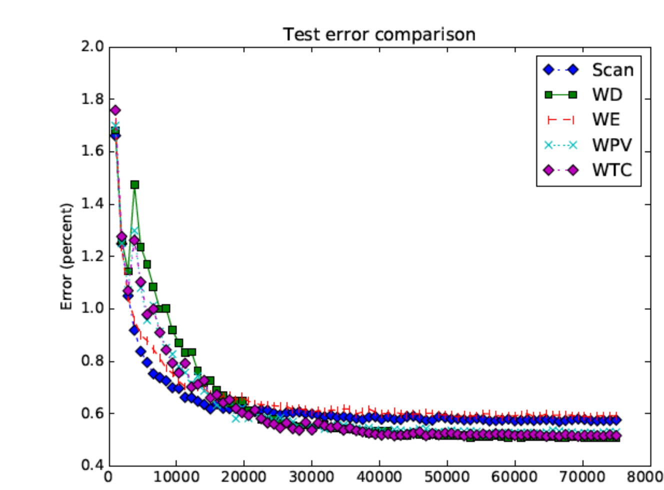


Figure 4: MNIST error rate (%)

Experimental setup

Dataset	# Class	Instance	Input dimensions	# Training	# Testing
MNIST	10	Image	28x28	60,000	10,000
CIFAR 10	10	Image	32x32x3	50,000	10,000
CIFAR 100	100	Image	32x32x3	50,000	10,000
Question Type	6	Sentence	$50 \times L$	5492	500
CoNLL 2003	17	Word	$50 \times L$	204,567	46,666
OntoNote 5.0	74	Word	$50 \times L$	1,088,503	152,728

Dataset	# Conv layers	Filter size	Filter number	# Pooling layers	# BN layers	# FC layers	Dropout keep probs	L2 reg
MNIST	2	5x5	32, 64	2	0	2	0.5	0.0005
CIFAR 10	0	N/A	N/A	0	0	1	1	0.01
CIFAR 100	26 or 62	3X3	16, 32, 64	0	13 or 31	1	1	0
Question Type	1	(2,3,4)x1	64	1	0	1	0.5	0.01
CoNLL 2003 OntoNote 5.0	3	3x1	100	0	0	1	0.5, 0.75	0.001
MNIST	0	N/A	N/A	0	0	2	1	0

Table 2: Optimization hyper-parameters and experiment settings

Dataset	Optimizer	Batch size	Learning rate	Learning rate decay	# Epochs	# Burn-in epochs	# Trials
MNIST	Momentum	64	0.01	0.95	80	2	20
CIFAR 10	SGD	100	1e-6	0.5 (per 5 epochs)	30	10	30
CIFAR 100	Momentum	128	0.1	0.1 (at 80, 100, 120 epochs)	150	90 or 50	20
Question Type	ADAM	64	0.001	1	250	50	100
CoNLL 2003 OntoNote 5.0	ADAM	128	0.0005	1	200	30	10
MNIST	SGD	128	0.1	1	60	20	10

Conclusion and Future Work

Lightweight supervised training trick motivated by active learning.

Training error	Validation error	Emphasize
Low	Low	Uncertain or hard examples
High	High	Uncertain or easy examples
Low	High	Future work

Can we apply this trick to reinforcement learning?

References

[1] Bengio, Yoshua, Louradour, Jérôme, Collobert, Ronan, and Weston, Jason. Curriculum learning. In ICML , 2009.

[2] Kumar, M Pawan, Packer, Benjamin, and Koller, Daphne. Self-paced learning for latent variable models. In NIPS , 2010.

[3] Shrivastava, Abhinav, Gupta, Abhinav, and Girshick, Ross. Training region-based object detectors with online hard example mining. In CVPR , 2016.

[4] Avramova, Vanya. Curriculum learning with deep convolutional neural networks, 2015.