A Mathematical Analysis of Learning Loss for Active Learning in Regression

Megh Shukla, Shuaib Ahmed
Mercedes-Benz Research and Development India

Workshop on Fair, ***Data Efficient*** and Trusted Computer Vision

Mercedes-Benz

Motivation

Active Learning for continuous model refinement

Can we recognize model failures *on-the-fly?*

Solution

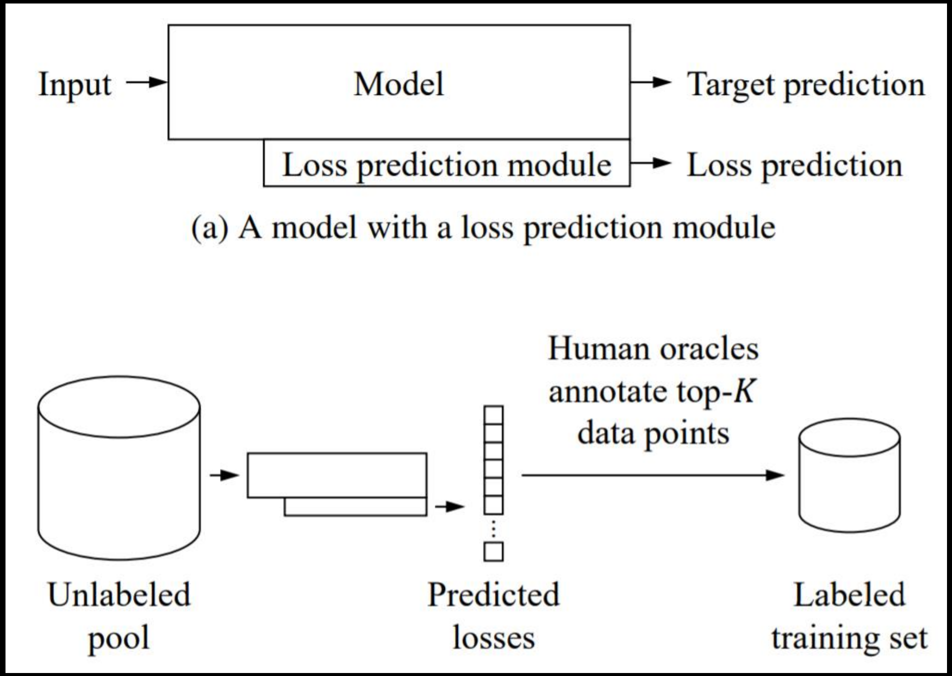
LearningLoss++

A mathematical evolution of Learning Loss to better identify failure cases for deployed models

What is Learning Loss?

Yoo and Kweon, "Learning Loss for Active Learning", CVPR 2019

Architecture:



Auxiliary network appended to the main model to predict the loss for a given image

Objective:

$$\mathbb{L}_{loss} = \max\left(0, -\underbrace{\mathrm{sign}(l_i - l_j)}_{\text{True Loss}}\underbrace{(\hat{l}_i - \hat{l}_j)}_{\text{Predicted Loss}} + \underbrace{\xi}_{\text{Predicted Loss margin}}\right)$$

Compares the true loss and predicted loss

Margin ensures a minimum separation between predicted losses

Why Learning Loss?

Task Agnostic, Real-time active learning
… Lacks rigorous analysis?

How Does Learning Loss Work?

Analysing the gradient response

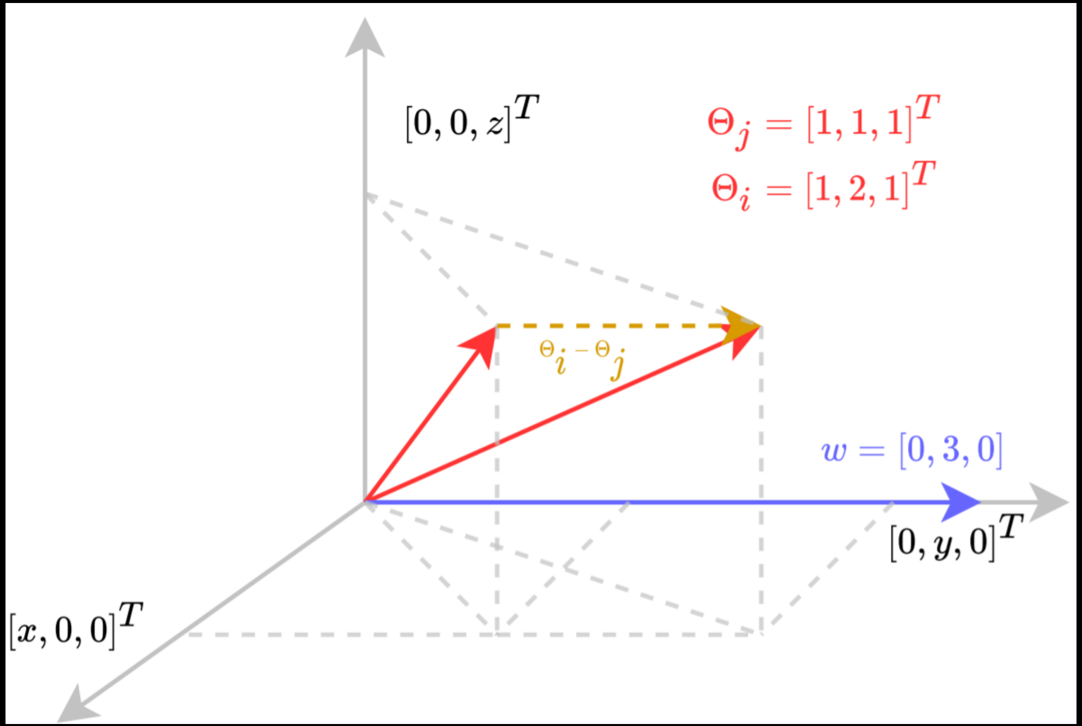
$$\mathbb{L}_{loss} = \max\left(0, -\operatorname{sign}(l_i - l_j)(\theta_i^T w - \theta_j^T w) + \xi\right)$$

$$\nabla_w \mathbb{L}_{loss} \in \{0, \pm(\theta_i - \theta_j)\}$$
$$\nabla_\theta \mathbb{L}_{loss} \in \{0, \pm w\}$$

Case 1: $l_i > l_j$

Case 2: $l_i < l_j$

… Role of the margin?



The weights of the learning loss network align along the most discriminative component between the embedding pair

LearningLoss++

We show that a KL divergence based objective is equivalent to the original empirical formulation:

$$\mathbb{L}_{loss}(w, \theta_i, \theta_j) = \mathrm{KL}(p||q) = p_i \log\frac{p_i}{q_i} + p_j \log\frac{p_j}{q_j}$$

p: probabilistic interpretation corresponding to the true losses for a pair of images
q: softmax over the predicted losses for a pair of images

Learning Loss gradient

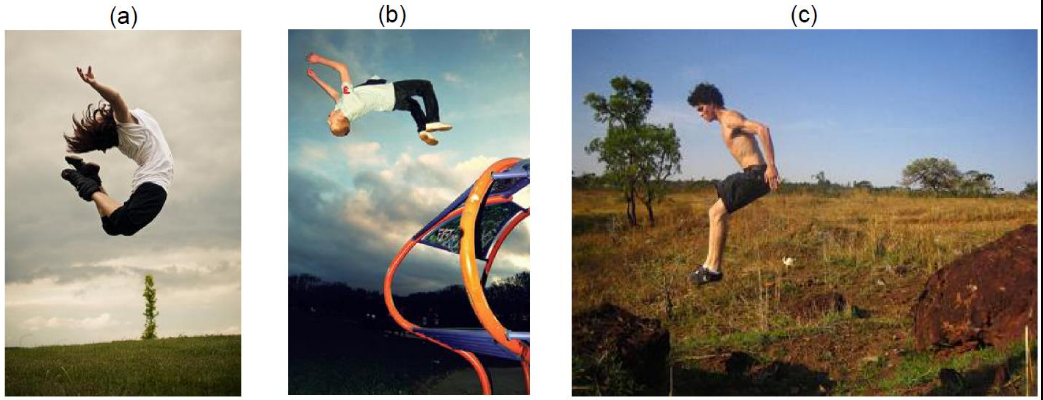
$$\nabla_w \mathbb{L}_{loss} \in \{0, \pm(\theta_i - \theta_j)\}$$
$$\nabla_\theta \mathbb{L}_{loss} \in \{0, \pm w\}$$

LearningLoss++ gradient

$$\nabla_w \mathbb{L}_{loss}(w, \theta_i, \theta_j) = (q_i - p_i)(\theta_i - \theta_j)$$
$$\nabla_\theta \mathbb{L}_{loss}(w, \theta_i, \theta_j) = (q_i - p_i)w$$

LearningLoss++ introduces a smoothness to the objective, absorbing the predicted loss margin!
How does this smoothness lead to better learning of failures?

LearningLoss++

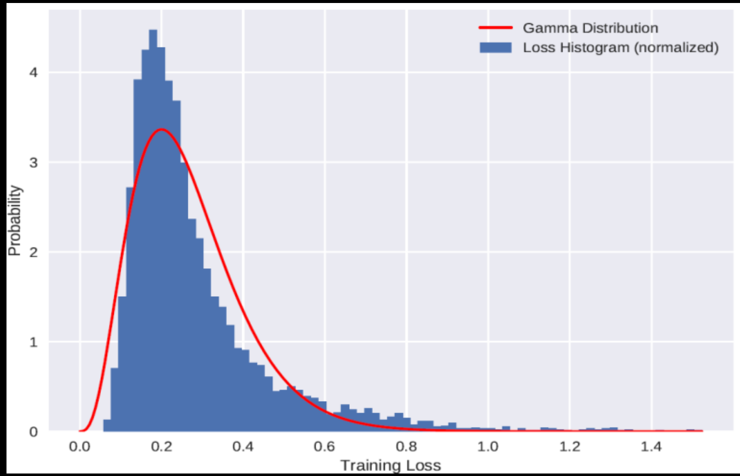


	(a)	(b)	(c)
True Loss: l	0.45	0.47	2.87
Pred Loss: $\hat{l}$	1.48	1.51	1.63
Image Pairs(i,j)		$(i=(a), j=(b))$	$(i=(a), j=(c))$
$\lvert\lvert\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\rvert\rvert$: (LearningLoss)		$\lvert\lvert\theta_i - \theta_j\rvert\rvert$	$\lvert\lvert\theta_i - \theta_j\rvert\rvert$
$\lvert\lvert\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\rvert\rvert$: (LearningLoss++)		≈ 0	$0.32\lvert\lvert\theta_i - \theta_j\rvert\rvert$

Case 1: True loss and predicted losses are similar

Learning loss incorrectly penalizes the network!

Q . How likely are we to sample a pair of images with similar true losses?



Statistical results allow us to model training losses with a Gamma distribution

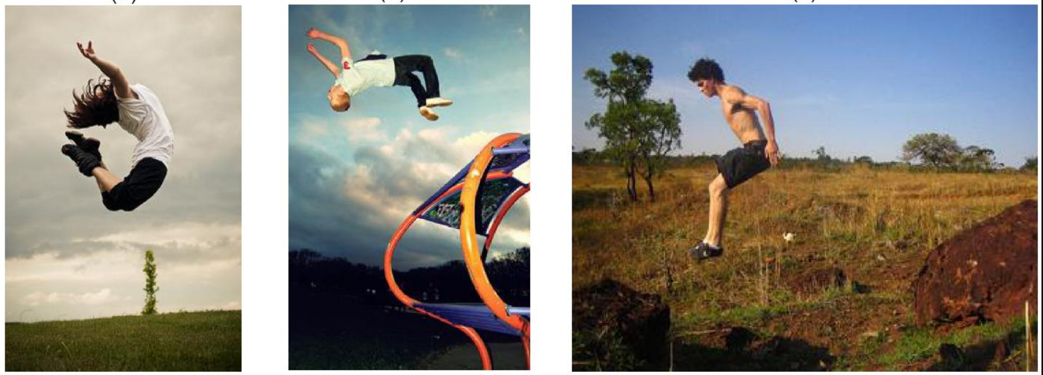
$$P(|X-Y| \leq \delta) = \int_0^{\delta} \gamma(x,k,\Theta) \int_0^{x+\delta} \gamma(y,k,\Theta)\mathrm{d}y\mathrm{d}x + \int_{\delta}^{\infty} \gamma(x,k,\Theta) \int_{x-\delta}^{x+\delta} \gamma(y,k,\Theta)\mathrm{d}y\mathrm{d}x$$

Sampling a pair of images with true loss within δ

δ	0.02	0.04	0.06	0.08	0.1	0.125	0.15
$P_{X,Y,\gamma}$	0.094	0.185	0.274	0.358	0.437	0.527	0.607

Ans . A sufficiently trained Learning Loss network imposes a penalty for a non trivial number of image pairs with true loss margin <= δ

LearningLoss++



	(a)	(b)	(c)
True Loss: l	0.45	0.47	2.87
Pred Loss: $\hat{l}$	1.48	1.51	1.63

Image Pairs(i, j)	$(i=(a), j=(b))$	$(i=(a), j=(c))$
$\lvert\lvert\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\rvert\rvert$: (LearningLoss)	$\lvert\lvert\theta_i - \theta_j\rvert\rvert$	$\lvert\lvert\theta_i - \theta_j\rvert\rvert$
$\lvert\lvert\nabla_w(l_i, l_j, \hat{l}_i, \hat{l}_j)\rvert\rvert$: (LearningLoss++)	≈ 0	$0.32\lvert\lvert\theta_i - \theta_j\rvert\rvert$

Case 2: True loss different, predicted losses similar

Learning loss does not scale with the degree of error!

Q . Can we prove that LearningLoss++ implicitly absorbs both:
1) δ (true loss margin) 2) ε (predicted loss margin)?

LearningLoss++ gradient:

$$\nabla_w \mathbb{L}_{loss}(w, \theta_i, \theta_j) = (q_i - p_i)(\theta_i - \theta_j)$$

The expected gradient given the true loss margin δ is:

$$\mathbb{E}_{x,y|\delta_2}\Big[\nabla_w \mathbb{L}(w, \theta_i, \theta_j)\Big] = \lim_{\delta_1 \to \delta_2} \int_{x=0}^{x=\infty} \int_{y=x+\delta_1}^{y=x+\delta_2} (q_i - \underbrace{\frac{x}{2x+\delta_2}}_{})(\theta_i - \theta_j) \frac{\gamma(x,k,\Theta)\gamma(y,k,\Theta)}{p(y-x=\delta_2)} \mathrm{d}y\mathrm{d}x$$

Probability of sampling an image is the ratio of true losses

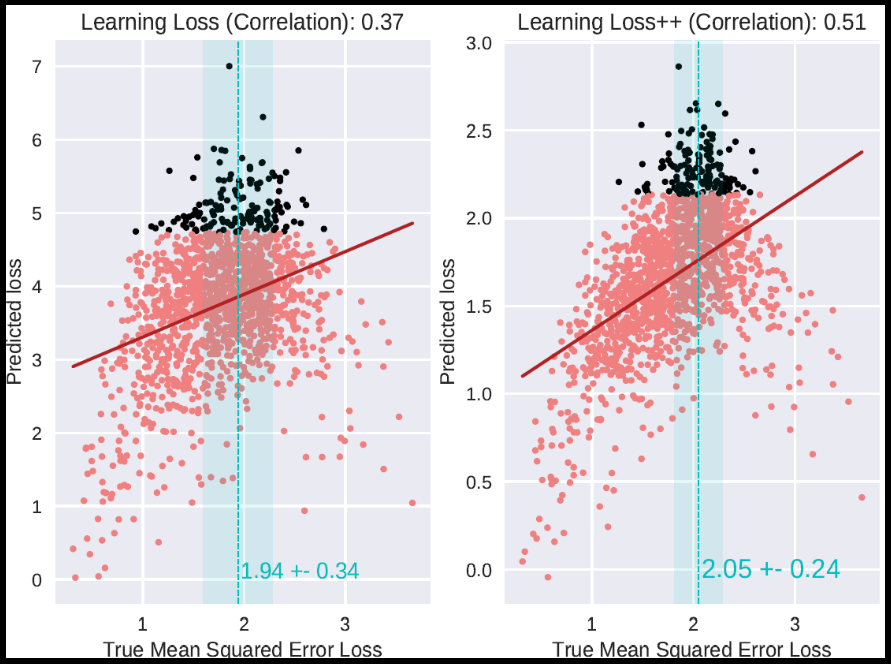
$\delta \rightarrow$	0.0	0.1	0.2	0.3	0.4	0.5
LL++	q_i-0.5	q_i-0.39	q_i-0.3	q_i-0.25	q_i-0.21	q_i-0.18
LL			$\leftarrow$ constant c_1 $\rightarrow$			

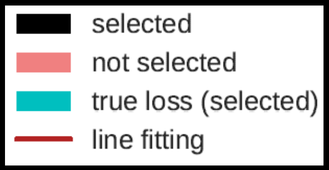
The softmax in LearningLoss++ forces the network to correctly identify *lossy* images as the true loss margin increases

LearningLoss++: Results and Discussion

(a) Failure Detection: PCK scores for the images sampled at Stage n. (Lower PCK values indicate better identification of faulty inferences.)

	LSP-LSPET (PCK@0.2)					MPII (PCKh@0.5)				
# images	2000	3000	4000	5000	6000	1000	2000	3000	4000	5000
Random	0.430 ±0.017	0.527 ±0.012	0.593 ±0.007	0.624 ±0.009	0.645 ±0.007	0.663 ±0.012	0.739 ±0.013	0.766 ±0.003	0.792 ±0.007	0.797 ±0.006
Coreset	0.288 ±0.017	0.438 ±0.020	0.447 ±0.017	**0.493** ±0.013	**0.556** ±0.010	0.384 ±0.014	0.522 ±0.009	0.608 ±0.012	0.697 ±0.009	0.755 ±0.029
LL	0.305 ±0.013	0.253 ±0.021	**0.358** ±0.025	0.520 ±0.011	0.617 ±0.017	0.311 ±0.036	0.465 ±0.024	0.621 ±0.017	0.735 ±0.012	0.777 ±0.010
LL++	**0.250** ±0.011	**0.186** ±0.022	0.385 ±0.011	0.533 ±0.020	0.627 ±0.012	**0.291** ±0.022	**0.439** ±0.018	**0.610** ±0.020	**0.705** ±0.023	**0.762** ±0.014
LL++conv	**0.209** ±0.018	**0.214** ±0.028	0.400 ±0.010	0.545 ±0.011	0.635 ±0.012	**0.309** ±0.029	**0.439** ±0.011	**0.603** ±0.016	**0.704** ±0.022	0.777 ±0.008





LearningLoss++ has a higher correlation with the true loss

This aids in better identification of images with high losses

Why use LearningLoss++?

1. Rigorous analysis for better explainability
2. Recognize real world failures on-the-fly!
3. Eliminates the margin hyperparameter!
4. The revised objective results in a smoother gradient to identify *lossy* images

Thank you!

Megh Shukla
Computer Vision Research Engineer
Mercedes-Benz R&D India

megh.shukla@daimler.com

https://www.linkedin.com/in/megh-shukla/