Invariant Representation Learning for Robust Deep Networks

Julian Salazar[1,*] Davis Liang[1,*] Zhiheng Huang[1] Zachary C. Lipton[1,2]
[1]Amazon AI [2]Carnegie Mellon University *equal contribution

Workshop on Integration of Deep Learning Theories, NeurIPS 2018

Introduction

Invariant representation learning (IRL) [1] *expresses the inductive bias that a deep network's intermediate representations should also exhibit invariance to noise.* Per sample x, we create noisy samples $\tilde{x}^1, \ldots, \tilde{x}^K \sim \nu_x$ and modify the objective to penalize the distance d_l between their intermediate activations a_l:

$$\mathcal{L}_{\mathrm{IRL}}(\boldsymbol{x}, y) = \alpha\mathcal{L}(\boldsymbol{x}, y) + \beta\mathcal{L}_{\mathrm{noise}}(\boldsymbol{x}, y) + \gamma\mathcal{L}_{\mathrm{dist}}(\boldsymbol{x})$$

$$\mathcal{L}_{\mathrm{noise}}(\boldsymbol{x}, y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}(\tilde{\boldsymbol{x}}^{(k)}, y),$$

$$\mathcal{L}_{\mathrm{dist}}(\boldsymbol{x}) = \sum_{\ell=1}^{L}\gamma_\ell\, d_\ell(a_\ell(\boldsymbol{x}), \{a_\ell(\tilde{\boldsymbol{x}}^{(k)})\}_{k=1}^{K})$$

We take d_l to be a weighted sum of L2 and cosine distance (reduces degs. of freedom). We show IRL:

- can be interpreted from vicinal and structural risk minimization theory
- generalizes known stochastic and analytic regularizations for noise/adversarial robustness
- enables semi-supervised learning
- also applies to deep networks for computer vision and language modeling

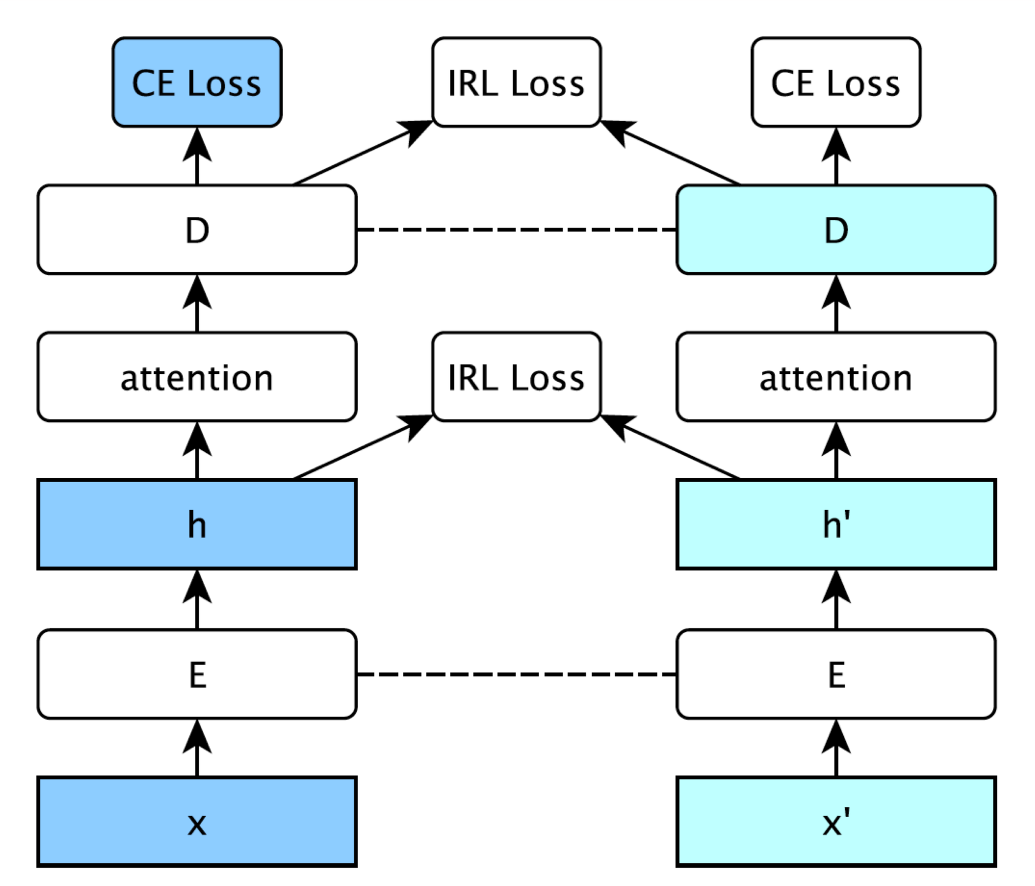

Figure 1. IRL loss with $K = 1$ for a sequence-to-sequence model.

Theory

Empirical risk minimization approximates P_{data}:

$$\hat{P}_{\mathrm{data}} = \frac{1}{N}\sum_{i=1}^{N}(\delta_{\boldsymbol{x}^{(i)}} \times \delta_{y^{(i)}}).$$

Optimization in deep learning approximates ERM:

$$R(\boldsymbol{\theta}) \approx \int L(\mathrm{x}, \mathrm{y}; \boldsymbol{\theta})\,\mathrm{d}\hat{P}_{\mathrm{data}}(\mathrm{x}, \mathrm{y}) \approx \int \mathcal{L}(\mathrm{x}, \mathrm{y}; \boldsymbol{\theta})\,\mathrm{d}\hat{P}_{\mathrm{data}}(\mathrm{x}, \mathrm{y}) \approx \tfrac{1}{B}\textstyle\sum_{b=1}^{B}\mathcal{L}(\boldsymbol{x}^{(b)}, y^{(b)}; \boldsymbol{\theta})$$

Vicinal risk minimization [2]: The estimate is improved by linear interpolation of δ_x with a noise model ν_x:

$$\int \mathcal{L}(\mathrm{x}, \mathrm{y}; \boldsymbol{\theta})\,\mathrm{d}[\alpha\delta_{\boldsymbol{x}} + \beta\nu_{\boldsymbol{x}}](\mathrm{x})\,\mathrm{d}\delta_y(\mathrm{y}) \approx \alpha\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{\theta}) + \frac{\beta}{K}\sum_{k=1}^{K}\mathcal{L}(\tilde{\boldsymbol{x}}^{(k)}, y; \boldsymbol{\theta})$$

To mitigate catastrophic forgetting due to batching, this mix is interpolated with the $\hat{y}$ predicted by the model:

$$\int (\mathcal{L}(\mathrm{x}, \mathrm{y}; \boldsymbol{\theta}) + \gamma\mathcal{L}(\mathrm{x}, f(\boldsymbol{x}; \boldsymbol{\theta}); \boldsymbol{\theta}))\,\mathrm{d}\nu_{\boldsymbol{x}}(\mathrm{x})\,\mathrm{d}\delta_y(\mathrm{y}) \approx \cdots + \frac{\gamma}{K}\sum_{k=1}^{K}\mathcal{L}_{\mathrm{c.e.}}(f(\tilde{\boldsymbol{x}}^{(k)}; \boldsymbol{\theta}), f(\boldsymbol{x}; \boldsymbol{\theta}))$$

for which symmetrizing and supervising multiple layers corresponds to L_{dist}. This stochastic form generalizes stability training [4], adversarial training [5], and logit pairing [7], although at best they all supervise at the logit level.

Structural risk minimization: As in [3], assuming a Gaussian noise model $\tilde{\mathrm{x}} = x + \xi$ allows for analytic approximation. IRL and cross-entropy give:

$$\mathcal{L}_{\mathrm{IRL}}(\boldsymbol{x}, y) \approx (\alpha + \beta)\mathcal{L}_{\mathrm{c.e.}}(\boldsymbol{x}, y) + \beta\Omega_R + \gamma\Lambda_R,$$

$$\Omega_R = -\frac{1}{2}\sum_{j=1}^{J_L} y_j(\nabla_{\boldsymbol{x}}\log f_j(\boldsymbol{x}))^\top\boldsymbol{\Sigma}(\nabla_{\boldsymbol{x}}\log f_j(\boldsymbol{x})),$$

$$\Lambda_R = \sum_{\ell=1}^{L}\gamma_\ell\sum_{j=1}^{J_\ell}(\nabla_{\boldsymbol{x}}(a_\ell)_j(\boldsymbol{x}))^\top\boldsymbol{\Sigma}(\nabla_{\boldsymbol{x}}(a_\ell)_j(\boldsymbol{x})),$$

where $\Sigma = \mathrm{Cov}(\xi)$. This analytic form generalizes works like [3] which assume isotropy/diagonal Σ, or structural gradient regularization [6] which computes a running estimate.

Semi-Supervision

For unlabeled data, one can continue viewing ν_x as kernel density estimate of P_{data}. Furthermore, [2] notes the model's best estimate can be used to approximate the unknown y. This gives

$$\int \mathcal{L}(\mathrm{x}, f(\boldsymbol{x}; \boldsymbol{\theta}); \boldsymbol{\theta})\,\mathrm{d}\nu_{\boldsymbol{x}}(\mathrm{x})$$

which corresponds to L_{dist} as before. Using the same CIFAR model and in-domain noise on 4,000 labeled gives **21.5%** and **60.7%** test error. Semi-supervised training with L_{dist} on unlabeled samples improves this to **18.2%** and **59.6%** respectively.

Activations and Gradients

In our CIFAR models, data augmentation already induces nearby intermediate representations, further improved by IRL. As predicted, per-layer Jacobians are reduced, especially in the last four layers relative to augmentation and IRL batching.

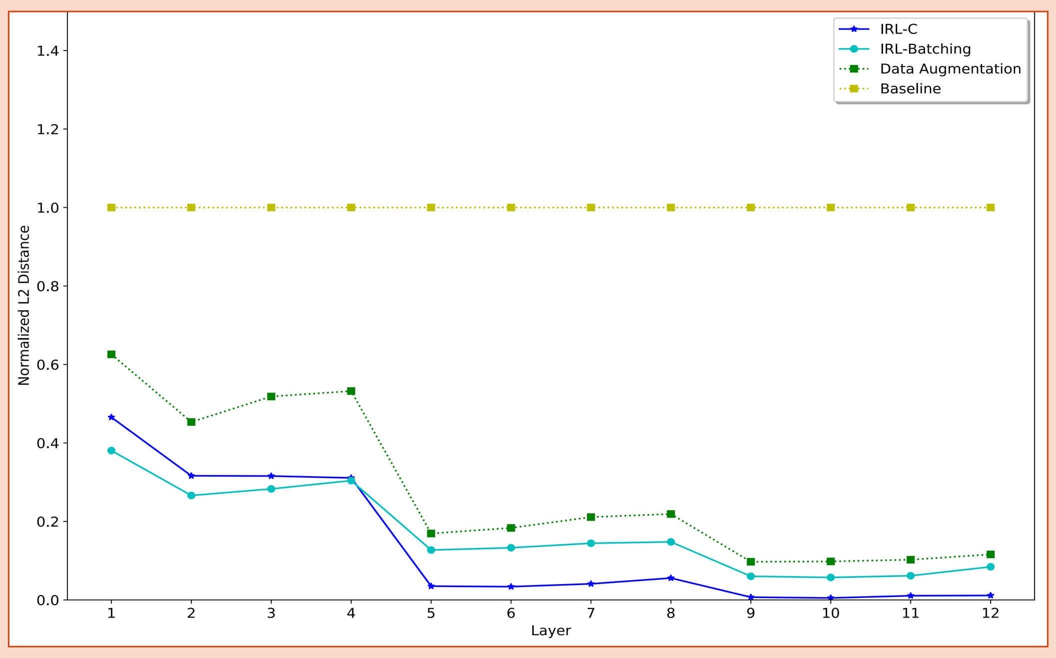

Figure 2. Normalized avg. L2 distance between *x* and *x'*

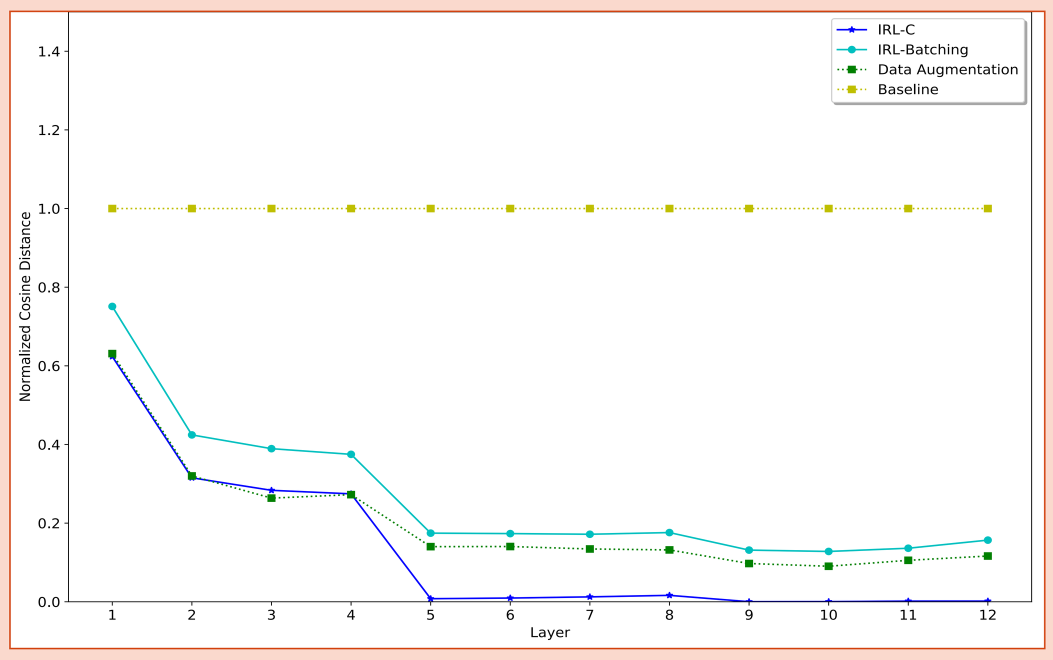

Figure 3. Normalized avg. cosine distance between *x* and *x'*

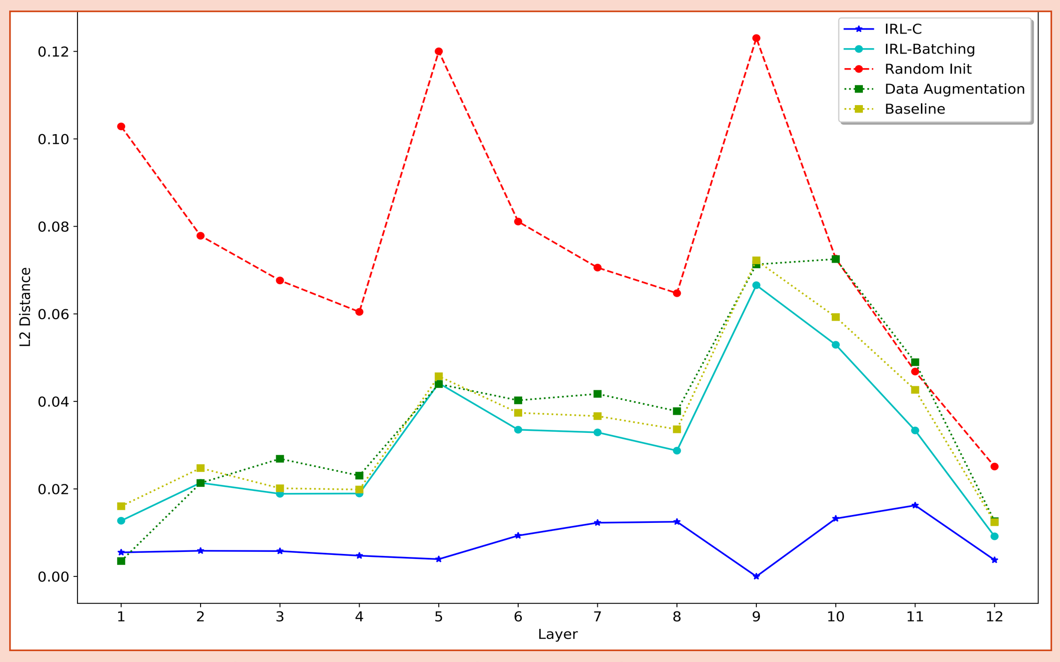

Figure 4. Jacobian norms per layer on the test set

Language Modeling

Our noise methods and baselines follow [9]; with probability $p = 0.2$, replace context words with a **unigram** draw or a special **blank** token. Here, IRL loss uniquely improves over the baseline where augmentation and IRL batching do not.

Table 1. Perplexities of a two-layer, 1500 unit, word-level LSTM.

		PTB		WikiText-2	
Method	Noise	val.	test	val.	test
Baseline	none	81.6	77.5	–	–
(ours)	none	80.5	77.5	96.9	92.1
Data aug.	uni.	85.8	82.9	106.7	100.0
IRL bat.	uni.	80.2	77.4	97.5	92.6
IRL loss	uni.	77.2	74.4	99.2	93.7
Data aug.	blank	78.8	75.3	98.6	92.3
IRL bat.	blank	80.3	76.7	97.7	93.8
IRL loss	blank	**75.1**	**71.8**	**94.0**	**88.6**

Computer Vision

We use IRL with Wide ResNet [8]. Their random crops, flips are applied before noising. We categorize noise into **in-domain** (brightness and contrast jitters, PCA noise), **out-of-domain** (hue and saturation jitters), **adversarial** (via FGSM [5]). IRL improves **5-8%** on the baseline trained in-domain, realized incrementally over data augmentation, IRL batching, and IRL loss. IRL also improves on regular and weighted adversarial training.

Table 2. Test set error on Wide ResNet-28-10 model. We apply IRL loss on the last four blocks.

			CIFAR-10						CIFAR-100				
(α, β, γ)	Method	Noise	none	in.	out.	ϵ=0.2	ϵ=0.8	Noise	none	in.	out.	ϵ=0.1	ϵ=0.4
$(1, 0, 0)$	Baseline	std.	3.89	–	–	–	–		18.85	–	–	–	–
$(1, 0, 0)$	(ours)	std.	3.73	5.43	9.24	40.62	82.22		18.77	23.23	39.70	55.87	96.96
$(0, 1, 0)$	Data aug.	in.	3.79	4.67	9.48	42.58	86.19		18.68	20.33	38.69	57.57	96.88
$(0.5, 0.5, 0)$	IRL bat.	in.	3.66	4.55	9.50	42.94	86.98		18.30	20.18	**38.54**	57.60	97.57
$(0.5, 0.5, 1)$	IRL loss	in.	**3.60**	**4.46**	**8.90**	48.43	87.72		**17.64**	**19.78**	38.89	57.70	97.65
$(0.5, 0.5, 0)$	Adv. trn.	ϵ=0.4	4.92	7.74	9.42	4.79	69.51	ϵ=0.2	23.83	31.48	47.51	20.90	83.46
$(0.8, 0.2, 0)$	IRL bat.	ϵ=0.4	5.14	8.70	11.45	5.04	71.61	ϵ=0.2	20.52	25.60	39.85	21.06	**80.40**
$(0.8, 0.2, 1)$	IRL loss	ϵ=0.4	4.55	7.81	10.32	**4.63**	**69.47**	ϵ=0.2	22.49	28.84	45.20	**20.26**	88.63

Speech Recognition

Our noise model is additive from MUSAN, as in [1]. Here we use WSJ, for which IRL training improves accuracy in three of five out-of-domain conditions.

Table 3. Character errors of 4-4 Enc.-Dec. on WSJ on noise

	WSJ (eval92)					
Method	none	RIR	speech	v.up	v.down	tel.
Regular	**11.2**	67.5	130.0	33.8	**28.0**	70.0
IRL bat.	13.0	**45.9**	83.5	29.7	29.4	**64.7**
IRL loss	12.3	48.2	**58.3**	**29.6**	30.0	73.0

References

1. Liang, D., et al. (2018). Learning noise-invariant representations for robust speech recognition. IEEE SLT
2. Chapelle, O., et al. (2001). Vicinal risk minimization. NeurIPS.
3. Bishop, C. (1995). Training with noise is equivalent to Tikhonov regularization. Neural Computation.
4. Zheng, S., et al (2016). Improving the robustness of deep neural networks via stability training. CVPR.
5. Goodfellow, I., et al (2015). Explaining and harnessing adversarial examples. ICLR.
6. Roth, K., et al (2018). Adversarially robust training through structured gradient regularization. ArXiv.
7. Kannan, H., et al (2018). Adversarial logit pairing. ArXiv.
8. Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. BMVC
9. Xie, Z., et al. (2017). Data noising as smoothing in neural network language models. ICLR

Contact

Julian Salazar
Amazon AI
julsal@amazon.com