Measuring Representational Robustness of Neural Networks Through Shared Invariances

Vedant Nanda 🐢 🧞, Till Speicher 🧞, Camila Kolling 🧞, John P. Dickerson 🐢, Krishna P. Gummadi 🧞, Adrian Weller 🕍

- University of Maryland, College Park
- Max Planck Institute for Software Systems
- University of Cambridge



Motivation

How to define *desired* invariances that a model should follow?



work



Whatever is same for another

High shared invariance necessary for robustness!

✓ Instead of using approximations of human perception (e.g., class labels), we have full access to representations of a neural network

✓ Allows us to investigate interesting questions about Deep Learning

✓ Useful for a future society with multiple agents controlled by neural nets where all networks should be similarly robust (e.g., driverless cars)

Problem Statement

X

Given: network m_2 : $\mathbb{R}^m \to \mathbb{R}^{d2}$, a reference network m_1 : $\mathbb{R}^m \to \mathbb{R}^{d1}$, inputs **X** How to measure shared invariances of m_2 wrt m_1 on **X** *i.e.*, $S_{inv}(m_2 | m_1, X)$?

STIR – Similarity Through Inverted Representations

1. Find Identically Represented Inputs (IRIs) X, X' s.t. $m1(X) \cong m1(X')$ $m_1(X)$ $\operatorname{argmin}_{\mathbf{X}} \mathcal{L}(\mathbf{X})$ $\mathcal{L}(\boldsymbol{X}) = \| \boldsymbol{m}_{1}(\boldsymbol{X}) - \boldsymbol{m}_{1}(\boldsymbol{X}) \|_{2}$ $\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{\alpha} \nabla_{\boldsymbol{Y}} \boldsymbol{\mathcal{L}}$ $m_1(X')$ 2. Measure similarity of $m_2(X)$ and $m_2(X')$ Use a representation similarity measure S_{rep}! $S_{inv}(m_2 | m_1, X, S_{rep}) =$ $S_{rep} (m_2(X), m_2(X')) =$

LinearCKA($m_2(\mathbf{X}), m_2(\mathbf{X}')$)

We propose STIR, a method to measure shared invariance between neural networks.



Explicitly captures invariances by interacting with underlying models



Offers additional insights beyond representation similarity measures



Useful tool to understand DL pipelines



pip install stir-invariance

Paper: tinyurl.com/stir-paper **Code:** github.com/nvedant07/stir

STIR offers insights beyond representation similarity

2 ResNet18 only differing in initial weights

Vanilla Training

Adversarial Training (AT)

	$m_1 \mid m_2$	$m_2 m_1$		$m_1 \mid m_2$	$m_2 m_1$					
STIR	0.605 _{± 0.013}	0.562 ± 0.023	STIR	0.934 _{± 0.003}	0.939 _{± 0.002}					
CKA	0.967	± 0.000	СКА	0.937 _{± 0.000}						

✓ Models trained with AT have higher STIR scores than Vanilla

 \checkmark AT explicitly induces invariance to ℓ_p perturbations (p=2 in this experiment), hence higher STIR scores are expected

✓ CKA is high in both cases and hence does not offer such insights

Losses & network architectures

STIR										CKA											
Vanilla, vgg19	- 0.32	0.26	0.21	0.14	0.43	0.35	0.41	1.00		1.0	Vanilla, vgg19	0.55	0.55	0.54	0.61	0.88	0.89	0.90	1.00		1.0
Vanilla, vgg16	- 0.38	0.26	0.30	0.28	0.41	0.34	1.00	0.38	- (0.8	Vanilla, vgg16	0.54	0.55	0.54	0.58	0.94	0.94	1.00	0.90		- 0.8
Vanilla, resnet34	- 0.44	0.36	0.28	0.28	0.60	1.00	0.43	0.34			Vanilla, resnet34	0.56	0.57	0.55	0.58	0.95	1.00	0.94	0.89		
Vanilla, resnet18	- 0.38	0.41	0.29	0.29	1.00	0.68	0.43	0.42	- (0.6	Vanilla, resnet18	0.55	0.56	0.54	0.58	1.00	0.95	0.94	0.88		- 0.6
AT, vgg19	- 0.78	0.78	0.84	1.00	0.41	0.39	0.35	0.39	- (0.4	AT, vgg19	0.84	0.84	0.86	1.00	0.58	0.58	0.58	0.61		0.4
AT, vgg16	0.85	0.82	1.00	0.82	0.39	0.41	0.36	0.35	Ì	0.1	AT, vgg16	0.92	0.90	1.00	0.86	0.54	0.55	0.54	0.54		0.4
AT, resnet34	- 0.94	1.00	0.93	0.92	0.56	0.58	0.55	0.56	- (0.2	AT, resnet34	0.94	1.00	0.90	0.84	0.56	0.57	0.55	0.55		0.2
AT, resnet18	- 1.00	0.94	0.93	0.92	0.51	0.52	0.52	0.56			AT, resnet18	1.00	0.94	0.92	0.84	0.55	0.56	0.54	0.55		
	AT, resnet18	AT, resnet34 -	AT, vgg16-	AT, vgg19 -	Vanilla, resnet18-	Vanilla, resnet34	Vanilla, vgg16-	Vanilla, vgg19-		0.0		AT, resnet18-	AT, resnet34 -	AT, vgg16 -	AT, vgg19-	Vanilla, resnet18-	Vanilla, resnet34	Vanilla, vgg16 -	Vanilla, vgg19-		⊥ 0.0

✓ Adversarial training leads to higher STIR scores

✓ Higher STIR when residual networks are reference models

Training datasets



✓ Shared invariance drops for later layers

✓ Drop lesser for AT than Vanilla

✓ Drop-off for STIR higher than that of CKA

Other investigations

[Effect of Random Init Across Layers] Models trained with AT consistently lead to higher STIR scores across layers

✓ [Comparing TRADES MART and AT] Despite similar l_{p} ball robustness, only moderate STIR scores, thus indicating different behavior of these models outside the l_{p} ball

[Updating Models w/ More Data] STIR monotonically increases as we add more data, but with diminishing returns

References

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. In CVPR 2015

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In ICML 2019

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In ICML 2019

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks In ICLR 2014