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?

Prior work: Whatever is same for 💁

Our 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

Given: network $m_2 : \mathbb{R}^m \rightarrow \mathbb{R}^{d2}$, a reference network m_1: $\mathbb{R}^m \rightarrow \mathbb{R}^{d1}$, inputs $\mathbf{X}$

How to measure shared invariances of m_2 wrt m_1 on $\mathbf{X}$ *i.e.*,$S_{inv}(m_2 \mid m_1, \mathbf{X})$?

STIR – Similarity Through Inverted Representations

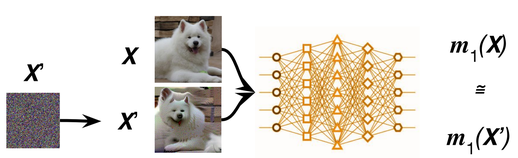
1. Find Identically Represented Inputs (IRIs)

$\mathbf{X}, \mathbf{X'}$ s.t. $m1(\mathbf{X}) \cong m1(\mathbf{X'})$

$\text{argmin}_{\mathbf{X'}} \ \mathcal{L}(\mathbf{X'})$

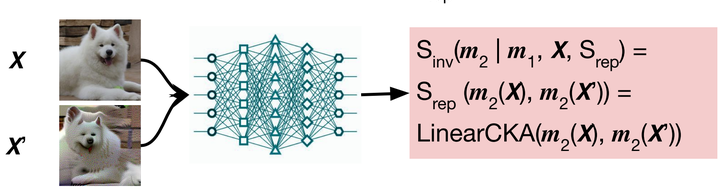
$\mathcal{L}(\mathbf{X'}) = \| m_1(\mathbf{X}) - m_1(\mathbf{X'}) \|_2$

$\mathbf{X'} = \mathbf{X'} - \alpha \nabla_{\mathbf{X'}} \mathcal{L}$



2. Measure *similarity* of $m_2(\mathbf{X})$ and $m_2(\mathbf{X'})$

Use a representation similarity measure S_{rep}!



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

	$m_1 \mid m_2$	$m_2 \mid m_1$
STIR	0.605 ± 0.013	0.562 ± 0.023
CKA	0.967 ± 0.000	

Adversarial Training (AT)

	$m_1 \mid m_2$	$m_2 \mid m_1$
STIR	0.934 ± 0.003	0.939 ± 0.002
CKA	0.937 ± 0.000	

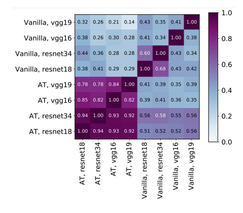
✓ **Models trained with AT have higher STIR scores than Vanilla**

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

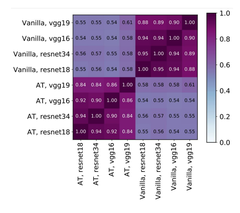
✓ **CKA is high in both cases and hence does not offer such insights**

Losses & network architectures

STIR



CKA

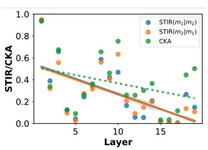


✓ **Adversarial training leads to higher STIR scores**

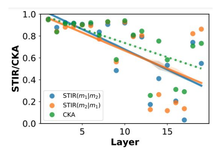
✓ **Higher STIR when residual networks are reference models**

Training datasets

Vanilla



AT



✓ **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_∞ 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