**Introduction**

Currently, the convolutional neural networks are state-of-the-art classifiers with near human level accuracies. However, they are less humanlike against changes in the images, especially geometric transformations. Because of this, many solutions were suggested to improve the invariance of the networks, which means we require a way to measure these solutions to find the best one.

**Method**

Given a classifier $k$, the image $I$, and the transformation set $\mathcal{T}$, we define the invariance score of $k$ to the transformation of $I$ by $\mathcal{T}$ as

$$\Delta_T(I,k) = \min_{T \in \mathcal{T}} \frac{d(e,r)}{\|r\|} \quad \text{subject to} \quad k(I) \neq k(Tr)$$

The metric $d$ is the geodesic distance between the no transformation and fooling one. Using this invariance score, the total invariance of $k$ becomes

$$\rho(k) = \mathbb{E}_k[\Delta_T(I,k)]$$

We approach this problem in three steps:

1. Design an algorithm to find a small $\tau$ that can fool $k$
2. Estimate the invariance score of $k$ for $m$ images
3. Estimate the invariance score as

$$\hat{\rho}(k) = \frac{1}{m} \sum_{i=1}^{m} \Delta_T(I_i,k)$$

**Algorithms**

**Manifool**

Start with $k$, $x$, and Lie group $\mathcal{T}$. Iterate by using two main steps:

1. Find movement direction $u \in \mathcal{T} \cdot M$
2. Map this movement to $M$ as $x_{i+1} = R(u)$

Iteration continues until $k$ is fooled. Output $\tau$ that generated this fooling image

**Iterative Projection**

Start with $k$, $x$, and set $\mathcal{T}$. Iterate by using two main steps:

1. Find an adversarial example behind the decision boundary:
   $$\hat{x} = x + (1+\alpha)r$$
2. Find the closest point to this target on set of transformed images:
   $$r_{i+1} = \arg\min_{r} \|x_{i+1} - Tr\|$$

Iteration continues until $k$ is fooled. Output $\tau$ that generated this fooling image

**Results**

- Our method is faster than similar methods such as Manitest[1] however with less accuracy.
- We can confirm that the invariance of neural networks increase with number of layers for many different transformation sets.

![Figure: Invariance scores of ResNet networks with different depths against projective transformations and its subsets. Computed using Manifool.](image1)

![Figure: Invariance score of ResNet networks with different depths against “general” transformations. Computed using Iterative Projection](image2)

**Conclusion/Perspectives**

- Designed two algorithms for finding small fooling transformations.
- Can find the invariance score for high dimensional transformation sets.
- Can help increase the accuracy of the image classification systems as they are used increasingly.

**Acknowledgments**

Many thanks to my supervisor, Seyed-Mohsen Moosavi-Dezfooli and Prof. Pascal Frossard for all their help during the project.

**References**