Out-of-distribution Analysis and Robustness of Deep Neural Networks

June 9, 2023
SEMLA - Montreal

Ettore Merlo, Zhenyu Yang, Mira Marhaba

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Projects

- DEEL
  - Dependable, Certifiable, and Explainable Artificial Intelligence for Critical Systems
  - https://www.deel.ai/

- Adimor (NSERC, CRIAQ, GHGSAT)
  - Tests and Robustness for AI-Based Image Recognition for Emission Monitoring Satellites
  - //www.ghgsat.com/

DNN Testing and Robustness

- Research: white-box profiling of neural net computation
  - Measure / assess neuron coverage profiles in training/test sets
  - Measure / compare neuron coverage profiles during classification

- We want to relate coverage profiles to how results of classification can be trusted
- Detect “unusual reasoning”

Method

- Extract neuron activation levels: Computational Profile
- Non-parametric approach
- Compute bin probabilities using the bin frequencies

$$p(b,i,j,X,k) = \frac{1}{|X| \cdot |K|} \cdot bFreq(b,i,j,X,k)$$

- Estimate the maximum likelihood (joint probability of all neurons in a layer)

$$L(y,j,k,X) = \prod_i p(b,i,j,X,k)$$

- Convert to logarithmic values (since the joint probabilities are small)
- Distance ensures that the numbers are all >= 0

$$dist(y,j,k,X) = -\sum_i log(p(b,i,j,X,k))$$

- High distance low probabilities input not likely to present profile close to training

OOD Detection

- Penultimate layer:

$$archDist(y,k,a) = dist(y, N-1, k, a)$$

- Average and std variation of training set:

$$refArchAvg(k,a) = \frac{1}{\mid X'_{k,\ a} \mid} \cdot \sum_{x \in X'_{k,\ a}} archDist(x,k,a)$$

$$refArchStdVar(k,a) = \sqrt{\frac{\sum_{x \in X'_{k,\ a}} (archDist(x,k,a) - refArchAvg(k,a))^2}{\mid X'_{k,\ a} \mid}}$$

- Sigma-Normalized Units:

$$normArchDist(y,k,a) = \frac{archDist(y,k,a)}{refArchStdVar(k,a)}$$

OOD Detection

- OOD Comparison of a single input:

$$OOD(y,k,a) = normArchDist(y,k,a) > sepTh(k,a)$$

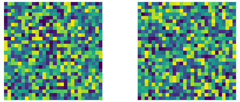
- InD Comparison of a single input:

$$InD(y,k,a) = \neg OOD(y,k,a) = normArchDist(y,k,a) <= sepTh(k,a)$$

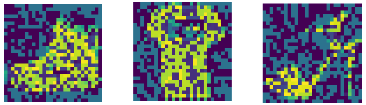
Adversarial Images

- Images from MNIST-Fashion data

- Experiment Datasets:
    1. Training set
    2. Test set
    3. Random set (noise)

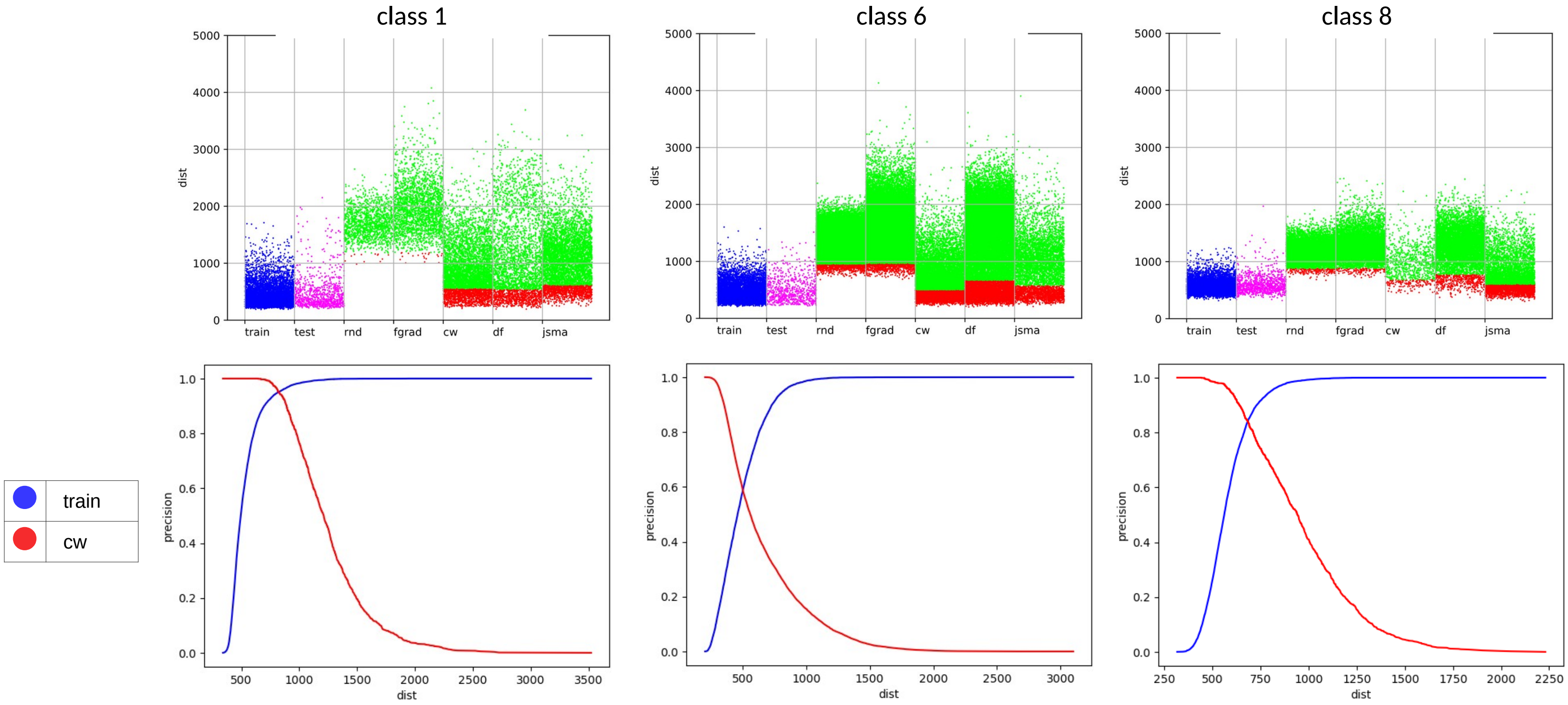


    4. Adversarial Images – Fast Gradient Method, Carlini & Wagner, DeepFool, Jacobian-Based Saliency Maps

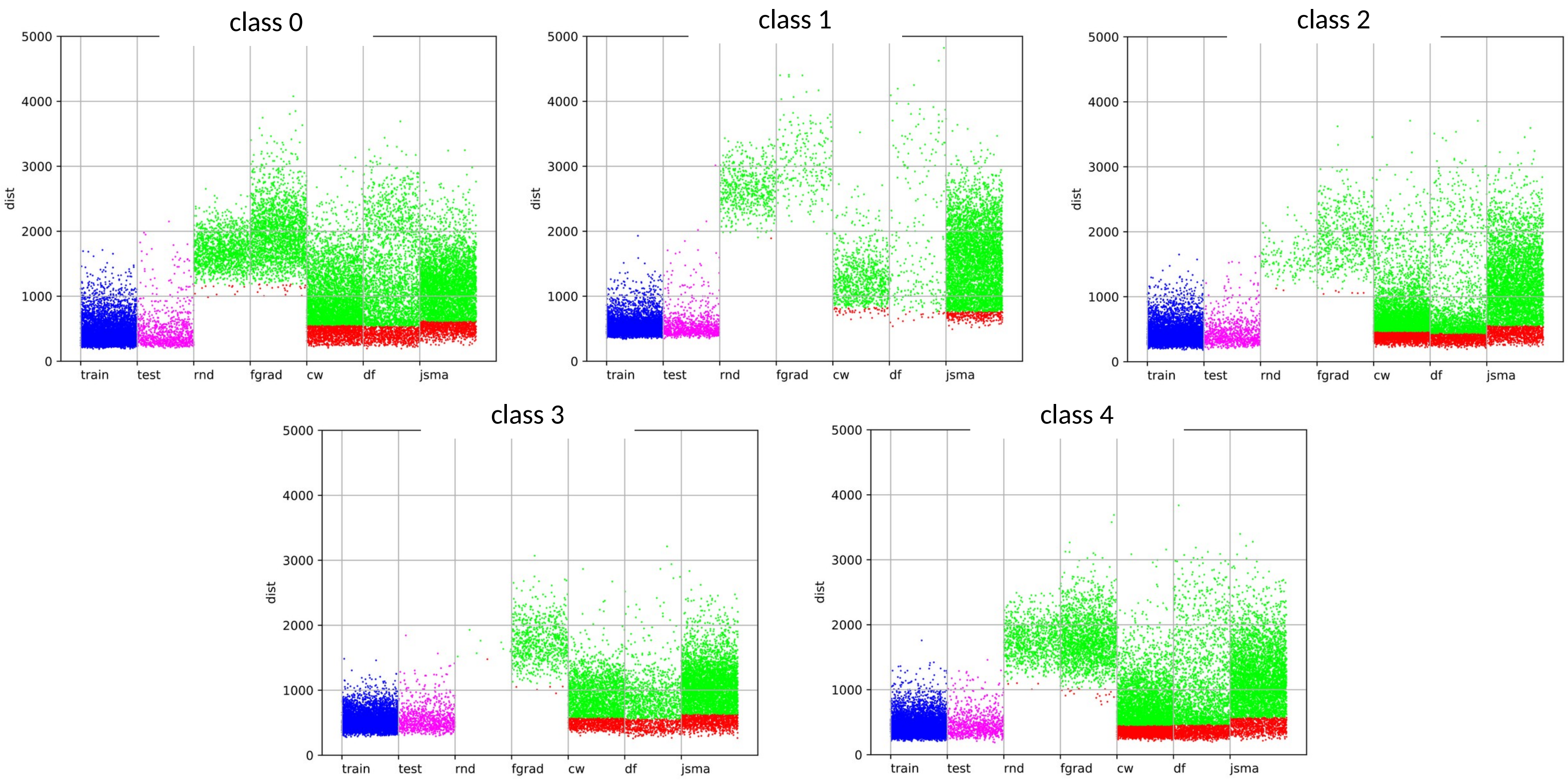


- Considered only the last layer before the output layer

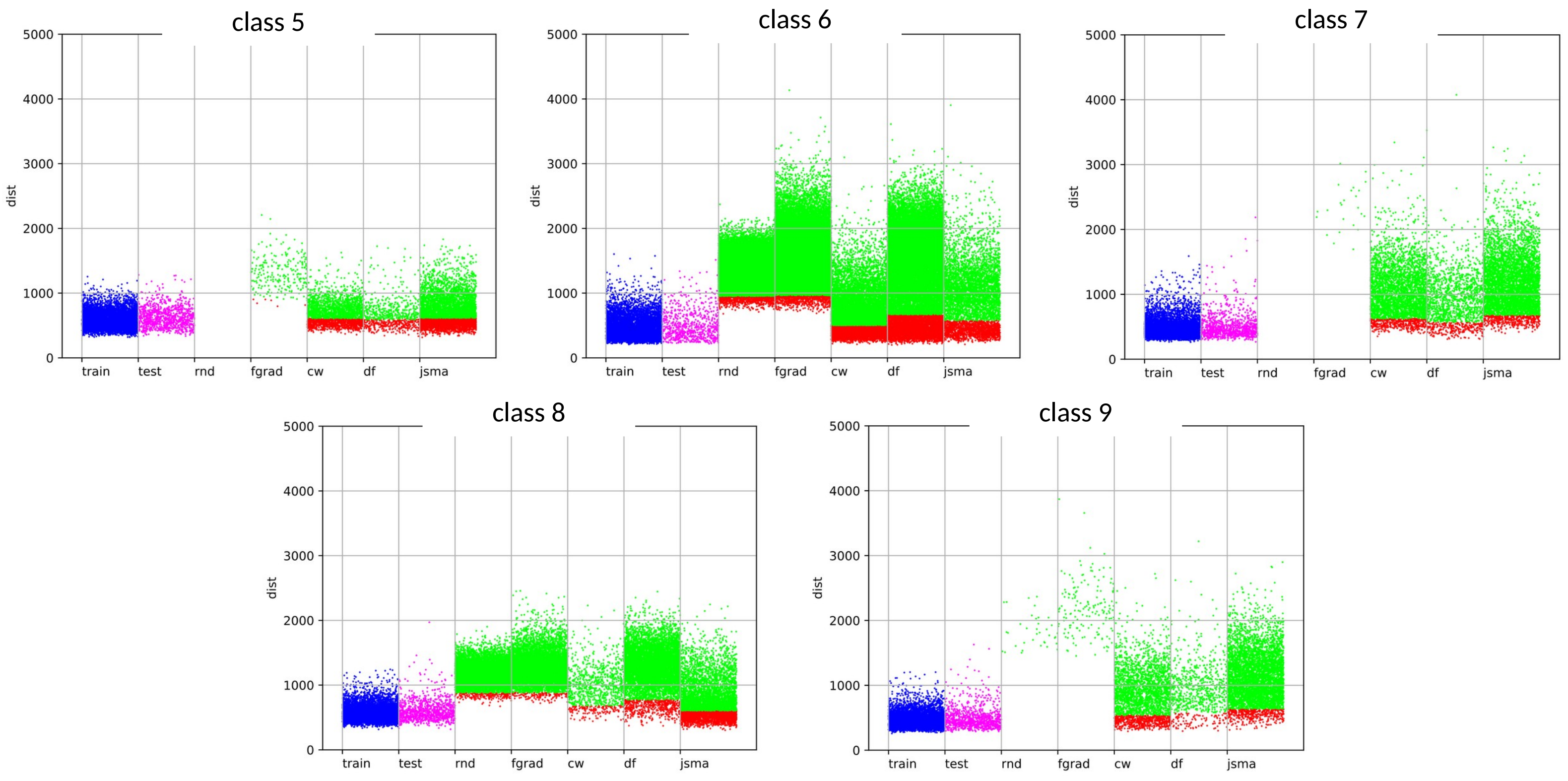
Visualization Examples: Classes 1, 6, 8



Distance Visualization: Classes 0 – 4



Distance Visualization: Classes 5 – 9



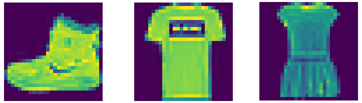
Linear Best Joint Separability

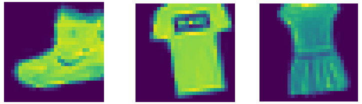
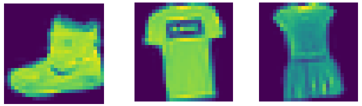
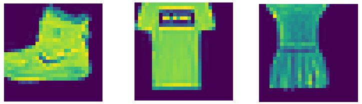
	class 0		class 1		class 2		class 3		class 4	
	train	adv	train	adv	train	adv	train	adv	train	adv
rnd	0.9908	0.9908	0.9998	1	0.9936	0.9938	0.9998	1	0.9956	0.9957
fgrad	0.9911	0.9915	0.9998	1	0.9917	0.9917	0.995	0.995	0.9912	0.9912
cw	0.7554	0.7555	0.9512	0.9518	0.6837	0.6837	0.7108	0.711	0.6132	0.6133
df	0.7378	0.7379	0.933	0.9355	0.6145	0.6145	0.6624	0.6624	0.6354	0.6355
jsma	0.8141	0.8142	0.9333	0.9333	0.8161	0.8163	0.8016	0.8016	0.8292	0.8292

	class 5		class 6		class 7		class 8		class 9	
	train	adv	train	adv	train	adv	train	adv	train	adv
rnd	-	-	0.9808	0.9808	-	-	0.9775	0.9775	0.9998	1
fgrad	0.9763	0.9793	0.9813	0.9814	0.9998	1	0.9805	0.9806	0.9998	1
cw	0.5961	0.5964	0.5879	0.588	0.8726	0.8726	0.8412	0.8419	0.785	0.7851
df	0.5568	0.5569	0.841	0.8411	0.8047	0.8048	0.9335	0.9336	0.8562	0.8562
jsma	0.6133	0.6134	0.7212	0.7214	0.9101	0.9101	0.656	0.656	0.9174	0.9174

Affine Transformations

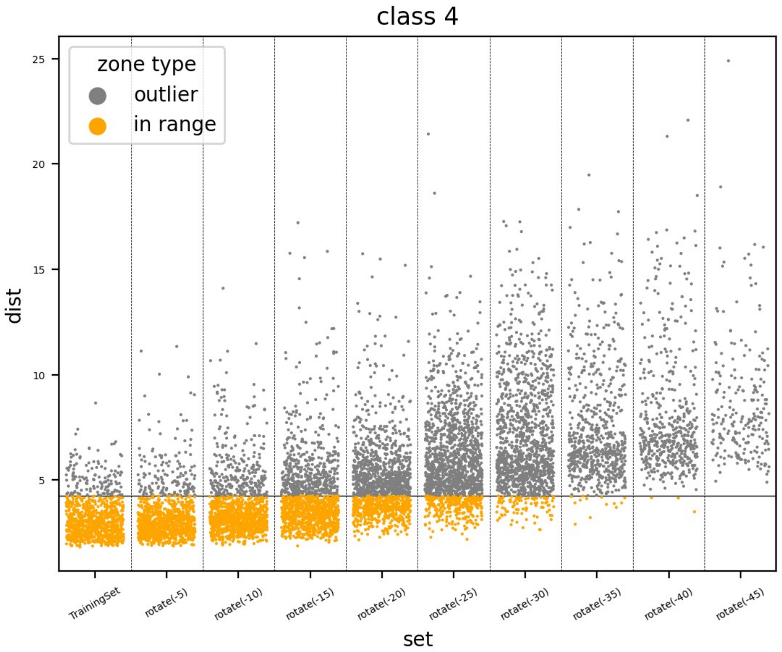
- Images from MNIST-Fashion data



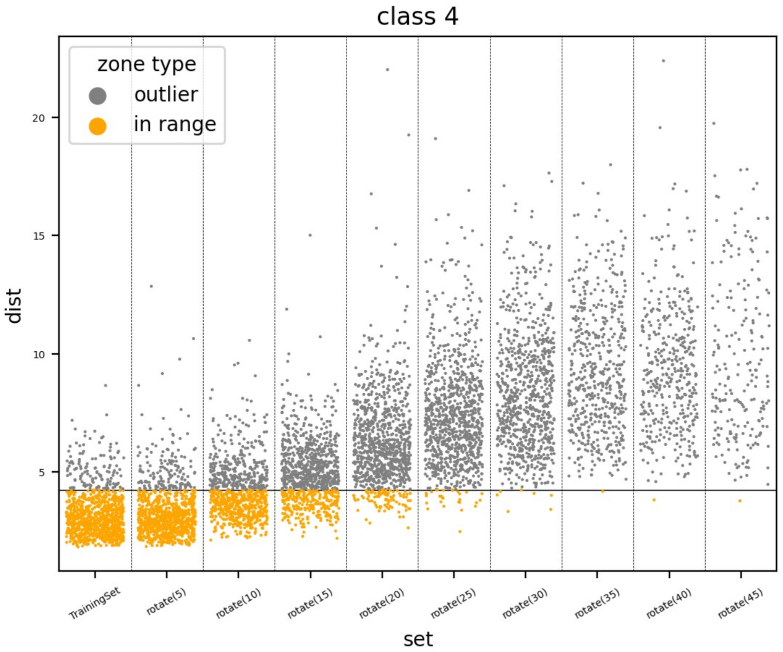
- Experiment Datasets:
  1. Training set
  2. Affine transformations
     - Corner Rotations 
     - Center Rotations 
     - X,Y Translations 

Transformations: Example Visualizations

Center Rotations



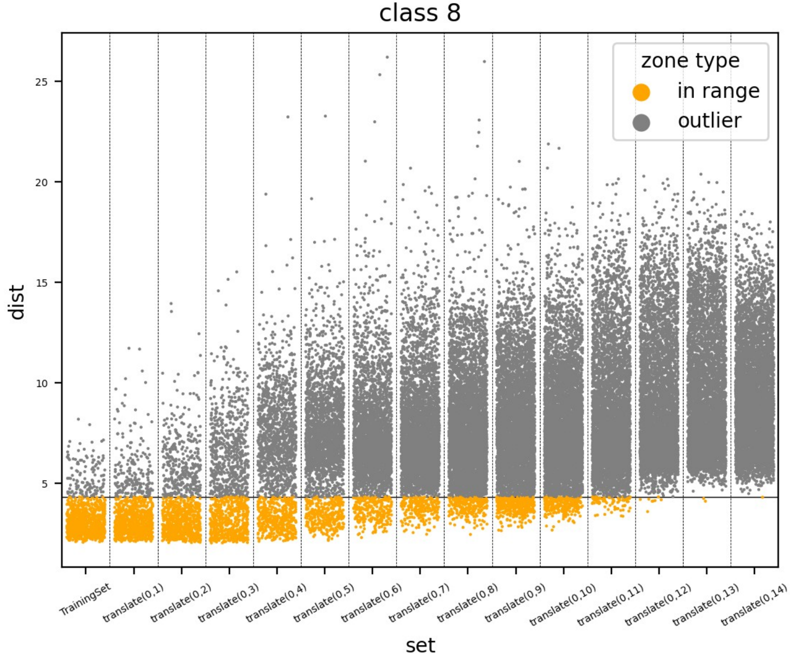
Clockwise rotation



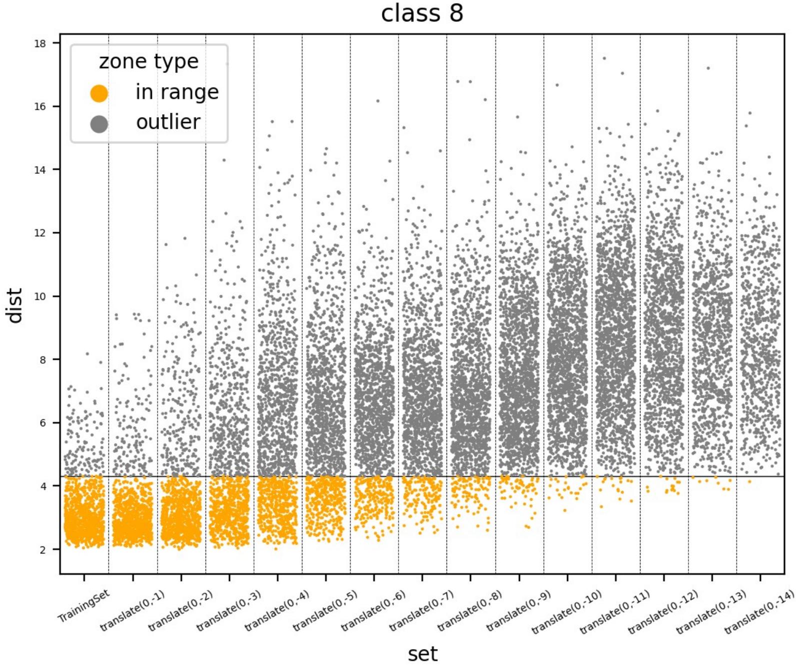
Counter-clockwise rotation

Transformations: Example Visualizations

Translations



Upward translation



Downward translation

Discussion

- Adversarial Attacks
  - Different distribution for each type of attack
  - Differences in behaviour between each class

- Affine Transformations:
  - The more we transform, the further we get from the average of the train set
  - We are able to identify the cases that are the most aggressive and thus have very similar characteristics to our starting images
  - Differences in behaviour depending on true class and on predicted class

Remarkable Points

- Correctly identified 70%-90% of OOD cases (with 10%-30% unrecoverable misclassifications)
- Comparable with previous approaches (SADL)
- Preserves performance without need of a secondary classifier trained on outliers or on errors

Technical Conclusions

- Adversarial Attacks: relatively high recognition of adversarial cases
- Affine transformations: exercise different network paths/profiles
- Both can be considered as “aggressive” test cases
- Potential advantages
  - Measurable robustness
    - Unlikely and unusual coverage profiles are detected
  - Linearly separable categories of coverage profiles
    - Robustness evaluation could be based on separability thresholds and risk levels in avionics domain

Applications

- ML reliability assessment
- Combined Human-Machine Interaction
- ML artefacts evaluation, assessment, testing
- Performance auditing

Future Research

- Multiple architectures
- Multiple datasets
- Investigate different schemes of likelihood based separability (best, fixed, training set variance, etc.)
- Different hyperparameters for attacks
- Ensemble methods on multiple models: Boosting, Bagging, Stacking

Future Research

- Deeper investigation of class-dependent separation
- Disregard reasoning coming from inactive or weakly activated neurons, or from unusual profiles during training
- Combining OOD of network input data with OOD of computational profiles
- Investigate transferability of adversarial attacks across networks and OOD

Questions? Comments?