# A Tale of Two Oracles: Defining and Verifying when AI Systems are Safe

Edoardo Manino

University of Manchester (UK)
This work is funded by the EPSRC grant EP/T026995/1 entitled "EnnCore: End-to-End Conceptual Guarding of Neural Architectures" under *Security for all in an AI enabled society*

# The Oracle Problem

## Testing a Black-Box System Requires

- Many test cases (inputs)
- Their ground-truth (outputs)

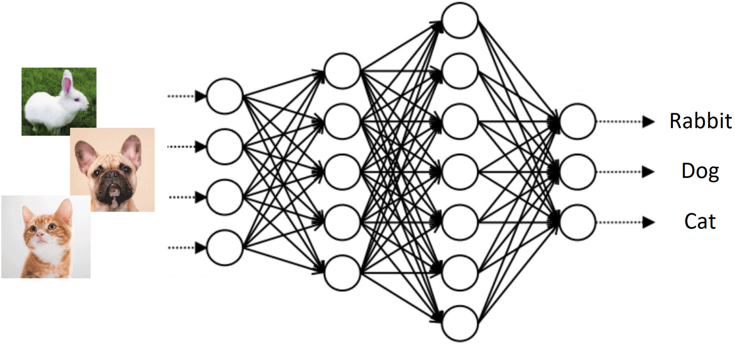
## "Exhaustive" Testing Would Require

- The presence of an oracle
- That can gives us the ground-truth
- For **any** possible input

## A Safety Paradox

- If such oracle exists, we do not need the black box system!
- This talk: two ML-specific variants of this paradox

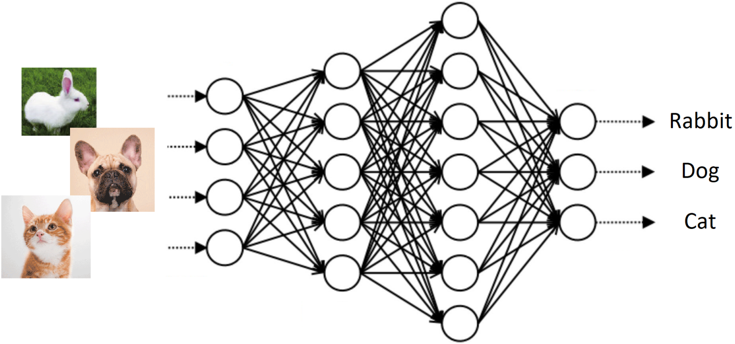
# Back to the Basics: The Data Scientist's View




## ML "Ingredients"

- A (possibly large) dataset of examples
- A ML model and an algorithm to train it

# Back to the Basics: Empirical Risk Minimisation



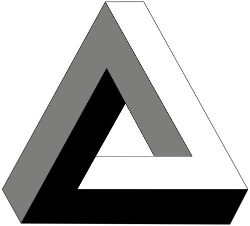

## What's The Requirement?

- Minimise the empirical loss $\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i), y_i)$
- That is, mimic the training set in some statistical sense

# The Requirements Paradox

## No Formal Requirements in ML

- Minimise the loss function
- Perform "well" on test set
- No constraints on OOD behaviour



## A ML Safety Paradox (1)

- If we have a full set of requirements we do not need ML at all
- I.e., just use the oracle

# Popular Safety Requirements

## Research Challenge

- Empirical risk minimisation is not strong enough
- We need to augment it with additional requirements

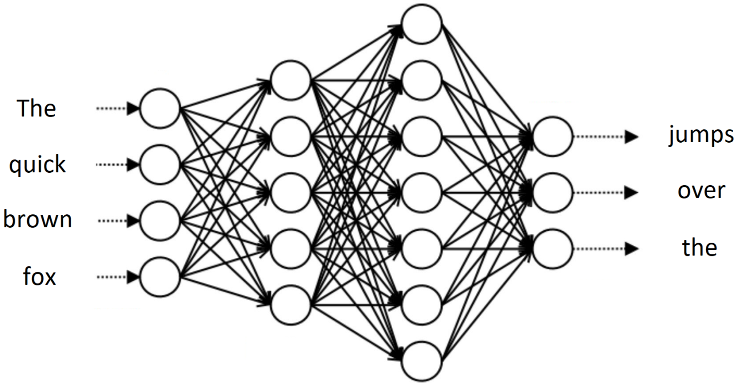
## Popular Safety Properties

- Deterministic: robustness*, monotonicity, equivalence, stability
- Probabilistic: robustness*, fairness
- System-Level: privacy-preserving ML, absence of backdoors

## A Property of ML Safety Properties (1)

- We only tell the ML system what **not** to do

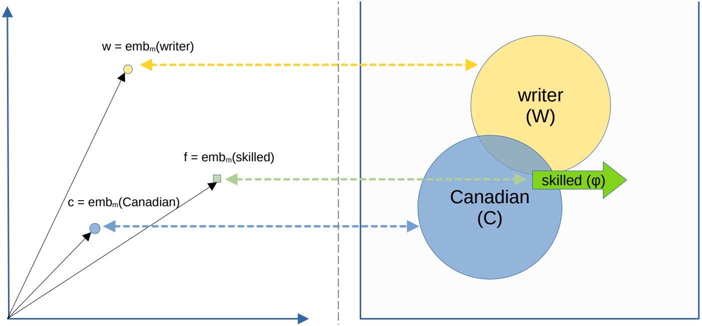
# NLP Safety Properties




## A Few Crucial Differences

- NLP inputs (tokens) are discrete not continuous
- Rich tradition of linguistic analysis, often grounded in logic
- Recent successes suggest the presence of shallow reasoning

# Montague Semantic Properties




## Contribution: formal translation from sets to vectors

- Left: ML models map sentences to points in a high-dim space
- Right: only some adjectives have set-intersective semantics
- in Carvalho et al., *Montague semantics and modifier consistency measurement in neural language models*, 2023

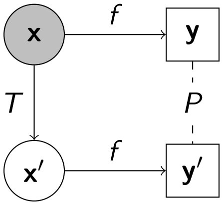
# Metamorphic Safety Properties

## A Property of ML Safety Properties (2)

- They are independent from the ground truth
- They establish behavioural constraints across inputs
- They measures internal consistency rather than correctness
- They are **metamorphic** properties

## Robustness-like Properties

- A "noise" perturbation $T$
- Output equivalence relation $P$
- It must hold for every input $x$




## Research Question

- Can we encode high-level linguistic properties this way?

# NLP Metamorphic Properties

| Pairwise systematicity metamorphic relations | | |
|---|---|---|
| Input: | $\mathbf{x}_1 =$ | $\boxed{\text{Light, cute and forgettable.}}$ |
| | $\mathbf{x}_2 =$ | $\boxed{\text{A masterpiece four years in the making.}}$ |
| | $\mathbf{x}_1' =$ | $\boxed{\text{Thank you.}}$ $\boxed{\text{Light, cute and forgettable.}}$ |
| | $\mathbf{x}_2' =$ | $\boxed{\text{Thank you.}}$ $\boxed{\text{A masterpiece four years in the making.}}$ |
| $T$: | | *concatenate the text* $\boxed{\text{Thank you.}}$ *at the beginning of the input.* |
| $P$: | | $s_{pos}(f(\mathbf{x}_1)) > s_{pos}(f(\mathbf{x}_2)) \iff s_{pos}(f(\mathbf{x}_1')) > s_{pos}(f(\mathbf{x}_2'))$ |

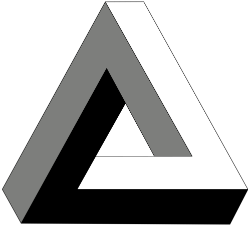
## Empirical results

- 112M+ relations from a dataset with 11K+ unlabelled entries!
- RoBERTa exhibits from 5% to 10% violations depending on $T$
- in Manino et al., *Systematicity, Compositionality and Transitivity [...]: a Metamorphic Testing Perspective*, 2022

# The Equivalence Paradox

## NNs have High Redundancy

- Opportunity for compression
- Pruning, quantisation, distillation
- Different arch. similar behaviour



## A ML Safety Paradox (2)

- Inference with the original NN (the oracle!) is expensive
- The compressed network may introduce unwanted behaviour

# Quantisation and NN Equivalence

| Safety Prop. | | Number of bits | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | 28 | 29 | 30 | 31 | 32 |
| Set. | $R_{40}$ | S | S | F | S | S | S | S | S | ... | S | S | S | S | S |
| | $R_{50}$ | S | S | F | F | F | F | F | F | ... | F | F | F | F | S |
| Vers. | $R_{20}$ | S | F | S | S | S | S | S | S | ... | S | S | S | S | S |
| | $R_{30}$ | S | F | S | S | S | S | S | S | ... | S | S | S | S | S |
| | $R_{40}$ | S | F | S | F | F | F | S | S | ... | S | S | S | S | S |
| | $R_{50}$ | S | F | F | F | F | F | F | F | ... | F | F | F | F | F |
| Virg. | $R_{20}$ | S | F | S | S | S | S | S | S | ... | S | S | S | S | S |
| | $R_{30}$ | S | F | S | S | S | S | S | S | ... | S | S | S | S | S |
| | $R_{40}$ | S | F | S | S | F | S | S | S | ... | S | S | S | S | S |
| | $R_{50}$ | S | F | F | F | F | F | F | F | ... | F | F | F | F | F |

Table: Effects of quantization on the safety of a NN trained on Iris data.

## Effects of Quantisation

- Even if the accuracy does not drop, the behaviour may change

# CEG4N: Counterexample-Guided NN Quantisation

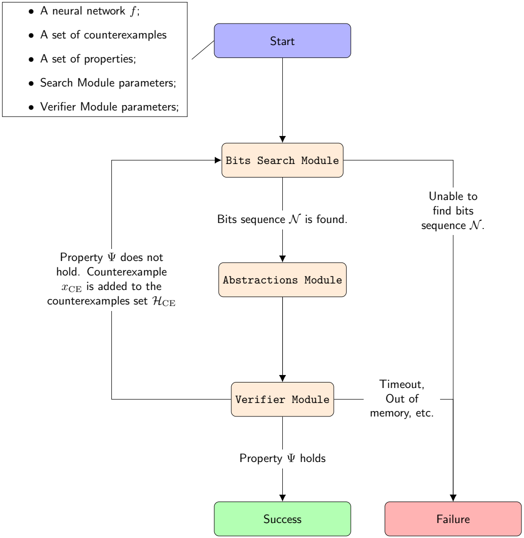
- in Batista et al., FoMLAS 2022

## Quantisation

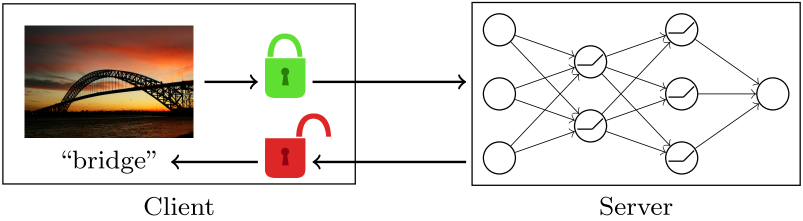
- Genetic algorithm
- Minimise bits
- Test equivalence

## Verification

- Verify equivalence
- If not, generate counterexample
- Augment testset
- Repeat



# Private Inference for Neural Networks



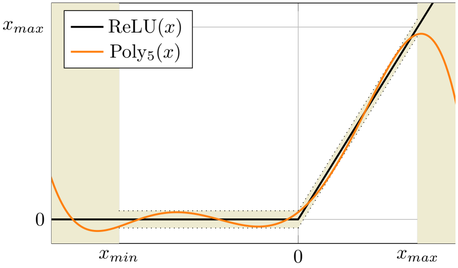

## Inference on Encrypted Data is Hard

- The encrypted computation should not leak information
- The decrypted result should be identical to non-private one
- Encryption primitive only support + and * efficiently
- The whole NN needs to be converted to a large polynomial!
- Can we ensure that the converted NN is equivalent?

# Certified Private Inference on Neural Networks via LiGAR

## Polynomial Approx.

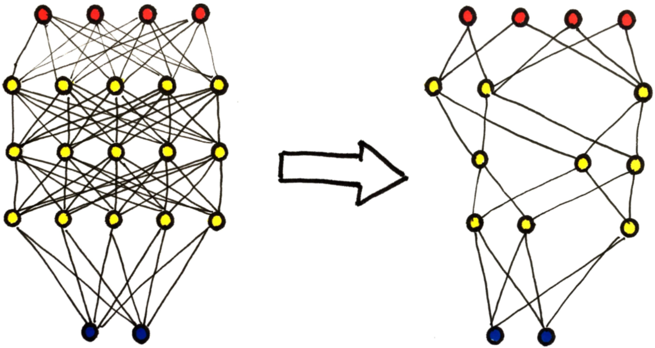
- Replace all activations
- Keep polynomial degree small
- Keep error small




## LiGAR: Lipschitz-Guided Abstraction Refinement

- Compute $x_{min}, x_{max}$ of each activation potential
- Compute Lipschitz constant of each error term
- Compute the polynomial degrees that minimise the error
- Tighten the abstraction bounds and repeat until convergence
- in Manino et al., FoMLAS 2023

# Pruning and NN Equivalence



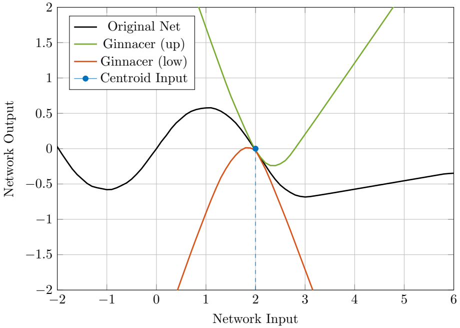
## Effects of Pruning

- In the same way as quantisation, the behaviour may change
- Can we keep certified error bounds on the pruned network?
- Is it possible to keep them relatively tight?

# Towards Global Abstractions with Local Reconstruction

## Pruning is Merging

- Merge neurons with similar $W$
- By taking the max/min of their weights




## Our GINNACER Algorithm

- Do not merge if the activation state changes at the centroid
- The upper and lower bounds are ReLU NNs themselves!
- Orders of magnitude tighter than other global abstractions
- Comparable tightness with SOTA local abstractions
- in Manino et al., Neural Network Journal, 2023

# Summary

## Requirements Paradox

- Formalise as many safety properties as possible
- Our Research: metamorphic definition of linguistic properties

## Equivalence Paradox

- Compressed NN may exhibit unwanted behaviour
- Our Research: NNs that are equivalent by design

## My Collaborators

- João Batista, Iury Bessa, Danilo Carvalho, Lucas Cordeiro, Eddie de Lima Filho, André Freitas, Bernardo Magri, Mustafa Mustafa, Julia Rozanova, Xidan Song

## Any Questions?