Property Inference for Deep Neural Networks

Aleksei Zhuravlev

27.01.2023

Problems with neural networks

- Lack of robustness. Small (imperceptible) changes to an input lead to misclassifications
- Lack of explainability: it is not well understood why a network makes a certain prediction
- Lack of intent when designing NNs: only learn from examples, often without a high-level requirements specification (crucial for safety-critical software systems)

Goal of the paper

- Automatically infer formal properties of feed-forward neural networks: Pre $\Rightarrow$ Post
- Capture input properties, layer properties based on features
- Define partitions on the input space, grouping together inputs that yield the same output by the network
- Two techniques to extract network properties

Input and layer properties

- Input property - a predicate over the input space, such that, all inputs satisfying it follow the same on/off activation pattern up to some layer and define convex regions in the input space
- Layer property - encode common properties at an intermediate layer that imply the desired output behavior. Can be seen as a grouping of several input properties as dictated by an internal layer.
- Decision pattern σ - specifies an activation status (on or off) for some subset of neurons. Minimal - dropping any neuron from the pattern invalidates it

Interpreting Inferred Network Properties

- Define regions in the input space in which the network is guaranteed to give the same label
- Understand why the network makes a certain prediction on an input.
- Interpret input properties using under-approximation boxes (bounds on each dimension)
- Distill large networks using layer patterns with high support

Results

- Two techniques: iterative relaxation and decision tree
- ACASXU:
  $36000 \leq range \leq 60760$, $0.7 \leq \theta \leq 3.14$, $-3.14 \leq \psi \leq -3.14 + 0.01$, $900 \leq v\_own \leq 1200$, $600 \leq v\_int \leq 1200$: turning advisory as COC
- MNIST:

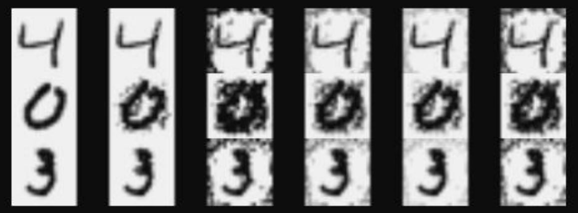

Fig. 4: Visualization of MNIST input properties using under-approximation boxes.

Fig. 5: Visualization of MNIST layer properties using under-approximation boxes.

- MNIST, distillation: 22% saving in inference time for 0.5% decrease in accuracy

Further work

- Compare the performance with other Explainable AI methods, e.g. feature permutation
- Estimate the influence of individual neurons on predictions made by the network
- Assess how random perturbation or permutation of the original dataset influence the output of the network