Hierarchical Interpretations of Neural Network Predictions

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Chandan Singh, W. James Murdoch, Bin Yu

July 2019

Introduction

- Introduces a DNN interpretation method called "Agglomerative Contextual Decomposition(ACD)"
- hierarchical interpretations to explain DNN predictions
- hierarchical clustering of the input features, with a contribution score for each cluster to the final prediction

ACD Overview

- Hierarchical clustering of features given the prediction from a DNN
- hierarchy optimized to indetify clusters of features identified by a DNN that are predictive
- CD+Hierarchical Agglomerative Clustering

Method: Contextual Decomposition for General DNNs

- Generalize CD for general DNNs
- A general DNN $f(x) = Softmax(g(x))$

$$f(x) = Softmax(g_L(g_{L-1}...(g_1(x)))) \quad (1)$$

- Given a group of features S $\{x_j\}j \in S$, decompose $g(x) = \beta(x) + \gamma(x)$
- $g^{CD}(x) = (\beta(x), \gamma(x))$
- $\beta(x)$ is contribution from S, and $\gamma(x)$ is contribution from rest

Method: Contextual Decomposition for General DNNs

- To get this final value, recompute decomposition for every layer
- For every layer i, $\beta_i(x) + \gamma_i(x) = g_i(x)$
- By compositing all these decompositions

$$f(x) = Softmax(g_L^{CD}(g_{L-1}^{CD}...(g_1^{CD}(x)))) \tag{2}$$

- Recomputing these for different types of layers: Conv, Pool, Relu

Method: Contextual Decomposition for General DNNs

- Conv Layer

$$\beta_i = W\beta_{i-1} + \frac{|W\beta_{i-1}|}{|W\beta_{i-1}| + |W\gamma_{i-1}|} \cdot b \tag{3}$$

$$\gamma_i = W\gamma_{i-1} + \frac{|W\gamma_{i-1}|}{|W\beta_{i-1}| + |W\gamma_{i-1}|} \cdot b \tag{4}$$

- MaxPool Layer

$$max\_idxs = \underset{idxs}{\text{argmax}} \left[\text{maxpool}(\beta_{i-1} + \gamma_{i-1}; idxs)\right] \tag{5}$$

$$\beta_i = \beta_{i-1}[max\_idxs] \tag{6}$$

$$\gamma_i = \gamma_{i-1}[max\_idxs] \tag{7}$$

- ReLU Layer

$$\beta_i = \text{ReLU}(\beta_{i-1}) \tag{8}$$

$$\gamma_i = \text{ReLU}(\beta_{i-1} + \gamma_{i-1}) - \text{ReLU}(\beta_{i-1}) \tag{9}$$

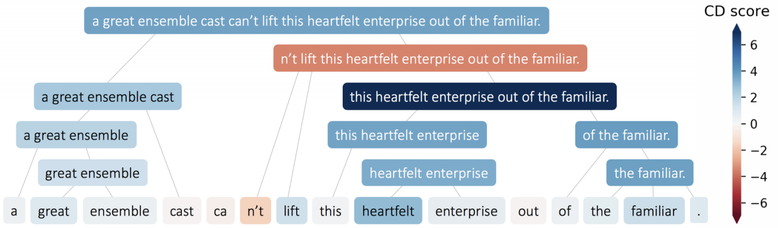
- Gives importance scores β_i for all feature groups

Method: Agglomerative Contextual Decomposition

Algorithm 1 Agglomeration algorithm.

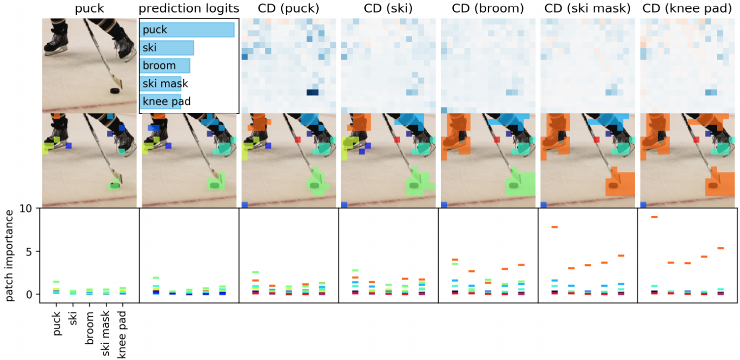
```
ACD(Example x, model, hyperparameter k, function CD(x, blob; model))
  # initialize
  tree = Tree()                                              # tree to output
  scoresQueue = PriorityQueue()                              # scores, sorted by importance
  for feature in x :
      scoresQueue.push(feature, priority=CD(x, feature; model))

  # iteratively build up tree
  while scoresQueue is not empty :
      selectedGroups = scoresQueue.popTopKPercentile(k)      # pop off top k elements
      tree.add(selectedGroups)                               # Add top k elements to the tree

      # generate new groups of features based on current groups and add them to the queue
      for selectedGroup in selectedGroups :
          candidateGroups = getCandidateGroups(selectedGroup)
          for candidateGroup  in candidateGroups :
              scoresQueue.add(candidateGroup, priority=CD(x, candidateGroup;model)-CD(x,selectedGroup;
  model))
  return tree
```

Experiments: Diagonosis of Incorrect Predictions



- Incorrect Combination of a positive sentiment vs a negative semtiment

Results: Dataset Bias



- Orange area very large : both skates and puck to classify as puck

Quantitative Results: Robust to Adversarial Perturbation

- Compute two ACDs: one for original image and the other for perturbed image
- Compute ranking of each pixel in each ACD based on when it was added into the hierarchy
- Compute correlation between adversarial vs original image

Attack Type	**ACD**	**Agglomerative Occlusion**
Saliency (Papernot et al., 2016)	0.762	0.259
Gradient attack	0.662	0.196
FGSM (Goodfellow et al., 2014)	0.590	0.131
Boundary (Brendel et al., 2017)	0.684	0.155
DeepFool (Moosavi Dezfooli et al., 2016)	0.694	0.202

Table 2: Correlation between pixel ranks for different adversarial attacks. ACD achieves consistently high correlation across different attack types, indicating that ACD hierarchies are largely robust to adversarial attacks. Using occlusion in place of CD produces substantially less stable hierarchies.

Quantitative Results: How much trust in the model: human study

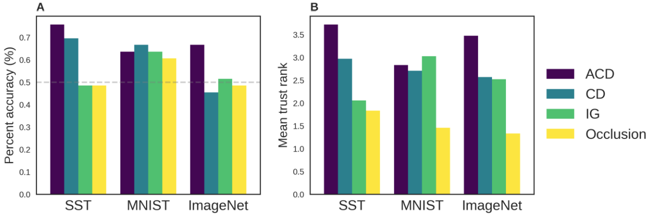


Figure 4: Results for human studies. **A.** Binary accuracy for whether a subject correctly selected the more accurate model using different interpretation techniques **B.** Average rank (from 1 to 4) of how much different interpretation techniques helped a subject to trust a model, higher ranks are better.