A Benchmark for Interpretability Methods in DNNs

(Google Brain)

Sam Laing

University of Tuebingen

June 19, 2024

A Bit of Background

- Which features in my input are most affecting the model's output?

A Bit of Background

- Which features in my input are most affecting the model's output?
- Deep image classification: "features" = pixels .

A Bit of Background

- Which features in my input are most affecting the model's output?
- Deep image classification: "features" = pixels .
- Ensembling also possible

A Bit of Background

- Which features in my input are most affecting the model's output?
- Deep image classification: "features" = pixels .
- Ensembling also possible
- Goal: help the engineer understand how their model is performing ...

A Bit of Background

- Which features in my input are most affecting the model's output?
- Deep image classification: "features" = pixels .
- Ensembling also possible
- Goal: help the engineer understand how their model is performing ...
- Austensibly. But are they even right?

Some of the included interpretability methods

* make individual slides and add pictures*

- Gradients (sensitivity heatmaps). i.e literally considering the gradient of output wrt. input at each pixel value. $e = \partial_{x_i} A_n^\ell$
- Guided Backprop (sort of a tidied up sensitivity map by only looking at positive values)
- Integrated Gradient: for each pixel, compare to baseline x^0 (often elected to be black pixel)
- Ensembling: Effectively injecting inputs with Gaussian noise J times and considering mean/variance. Then apply a gradient method. $\eta_i \sim N(\mu, \sigma)$
- SmoothGrad (SG) $e = \sum_{i=1}^J f(x + \eta_i, A_n^\ell)$
- SmoothGrad Squared (SG-SQ) $e = \sum_{i=1}^J f(x + \eta_i, A_n^\ell)^2$
- VarGrad (VAR) $e = \text{Var}\left(\sum_i^J f(x + \eta_i, A_n^\ell)\right)$

Notice the similarity between SG-SQ and VarGrad

Motivation

- How do we know how well a feature an interpretability method is really perfomative?

Motivation

- How do we know how well a feature an interpretability method is really perfomative?
- Different methods may consider different features important

Motivation

- How do we know how well a feature an interpretability method is really perfomative?
- Different methods may consider different features important
- How can I really say that interpretability method A has chosen better features than interpretabiilty method B?
- If only there was a benchmarking framework to do this...

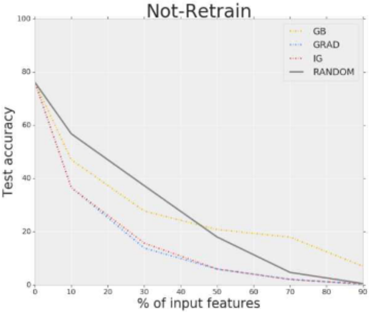
ROAR (RemOve And Retrain)

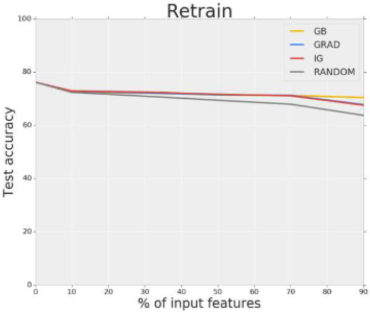
The idea behind ROAR

- For each method being considered, sort the feature (pixels) in order of ranked importance by the interpretability method.
  Creating an ordered tuple $(e_j)_{j=1}^{D}$ of pixel coordinates for each image in the training dataset.
- for $j \in \{0, 10, \ldots, 100\}$, we replace the top j percent of ranked pixels with the per channel mean of the image for every image and then retrain.
- Consider the affect of having dropped the "most informative pixels" as determined by each interpretability.
- Investigate how much their removal from the training process effects accuracy.
- Also a no-retraining variant

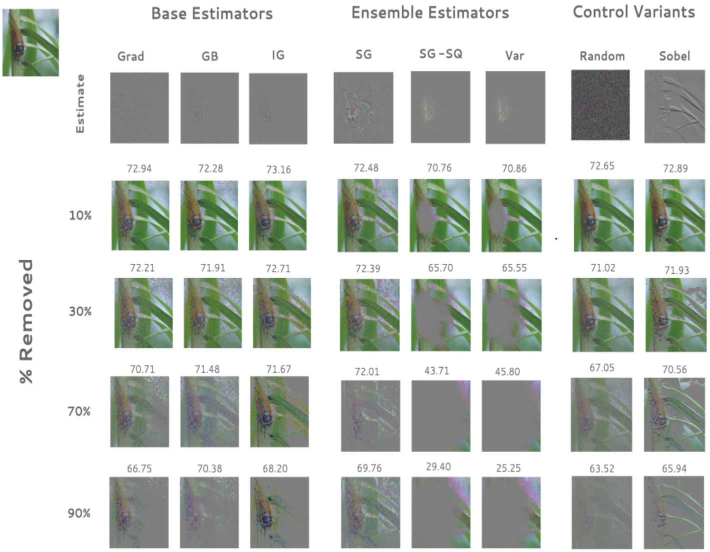
To Retrain Or not To Retrain

- Without retraining the model, train and test come from different distributions... violates key assumption of ML
- Paper therefore argues it is necessary





ROAR in Action



An Outline of the Experiment

Really just a refinement of above.

- Experiment was performed using a ResNet50 classifier on Imagenet, Birdsnap and Food 101

An Outline of the Experiment

Really just a refinement of above.

- Experiment was performed using a ResNet50 classifier on Imagenet, Birdsnap and Food 101
- Along with a number of different interpretability methods, a random ranking was also included: This tells us if the method outperforms random.

An Outline of the Experiment

Really just a refinement of above.

- Experiment was performed using a ResNet50 classifier on Imagenet, Birdsnap and Food 101
- Along with a number of different interpretability methods, a random ranking was also included: This tells us if the method outperforms random.
- New train and test sets are generated for each $j \in \{0, 10, 30, 50, 70, 90\}$

An Outline of the Experiment

Really just a refinement of above.

- Experiment was performed using a ResNet50 classifier on Imagenet, Birdsnap and Food 101
- Along with a number of different interpretability methods, a random ranking was also included: This tells us if the method outperforms random.
- New train and test sets are generated for each $j \in \{0, 10, 30, 50, 70, 90\}$
- For fairness, the model is retrained 5 times for each method (DNN training is noisy)

An Outline of the Experiment

Really just a refinement of above.

- Experiment was performed using a ResNet50 classifier on Imagenet, Birdsnap and Food 101
- Along with a number of different interpretability methods, a random ranking was also included: This tells us if the method outperforms random.
- New train and test sets are generated for each $j \in \{0, 10, 30, 50, 70, 90\}$
- For fairness, the model is retrained 5 times for each method (DNN training is noisy)

Results

- Surprisingly, replacing large numbers of pixels doesn't remove that much predictive power!

Results

- Surprisingly, replacing large numbers of pixels doesn't remove that much predictive power!
- For ImageNet, after 90% of the pixels are randomly removed, still 63.53% accuracy relative to the original 78.68%

Results

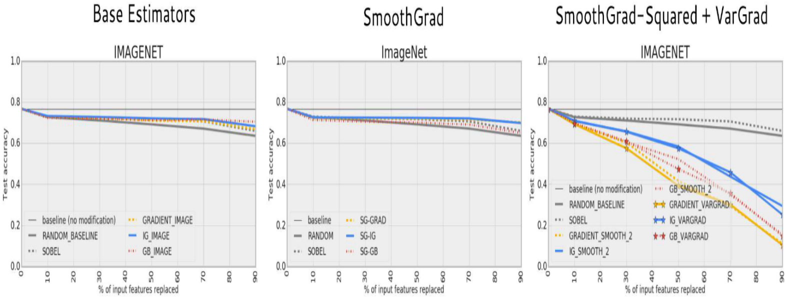
- Surprisingly, replacing large numbers of pixels doesn't remove that much predictive power!
- For ImageNet, after 90% of the pixels are randomly removed, still 63.53% accuracy relative to the original 78.68%
- Don't worry... this paper is from 2018, they don't stink at training networks :)

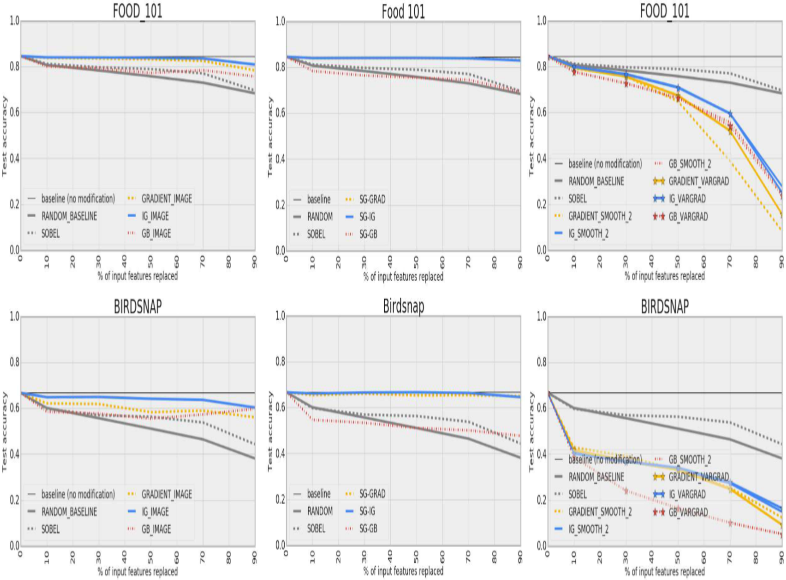
Results

- Surprisingly, replacing large numbers of pixels doesn't remove that much predictive power!
- For ImageNet, after 90% of the pixels are randomly removed, still 63.53% accuracy relative to the original 78.68%
- Don't worry... this paper is from 2018, they don't stink at training networks :)
- According to the paper, SG-SQ and VarGrad are the real heros

Results

- Surprisingly, replacing large numbers of pixels doesn't remove that much predictive power!
- For ImageNet, after 90% of the pixels are randomly removed, still 63.53% accuracy relative to the original 78.68%
- Don't worry... this paper is from 2018, they don't stink at training networks :)
- According to the paper, SG-SQ and VarGrad are the real heros




FOOD_101
Food 101
FOOD_101
BIRDSNAP
Birdsnap
BIRDSNAP
Test accuracy
% of input features replaced
baseline (no modification)
RANDOM_BASELINE
SOBEL
GRADIENT_IMAGE
IG_IMAGE
GB_IMAGE
baseline
RANDOM
SG-GRAD
SG-IG
SG-GB
GRADIENT_SMOOTH_2
IG_SMOOTH_2
GB_SMOOTH_2
GRADIENT_VARGRAD
IG_VARGRAD
GB_VARGRAD

Results (cont.)

- The dataset affects which

A Few Possible Issues in the Approach

- This experiment replaces the top j pixels with the mean of the image.
- Is this really the best way?
- The mean still conveys possibly useful information
- Alternative methods are sometimes advised but most have their own issues.

Another possible Issue

- In practice retraining a large image classifier several times is pretty unfeasible computationally speaking. (ImageNet with ResNet50 can take
- Without retraining, you run into theoretical violations of ML!

A

Yet Another Possible Issue

Why I chose this paper

Questions?