FINAL-REPORT: GradCAM

SongSeungWu

1 What is GradCAM?

Grad-CAM is a class-discriminative visualization technique designed to enhance the interpretability of CNN-based models. By utilizing gradient information from the final convolutional layer, Grad-CAM generates a heatmap that highlights image regions critical for the model's prediction. This method does not require modifications to the model's architecture or retraining, making it broadly applicable across tasks such as image classification, captioning, and visual question answering.

2 Grad-CAM Implementation Using Pretrained Models (AlexNet and VGG-16)

This implementation demonstrates the use of Grad-CAM with pretrained CNN architectures, AlexNet and VGG-16. Grad-CAM enhances the interpretability of CNN models by highlighting important regions in the input image that are most influential for a specific class prediction.

Framework of Grad-CAM

- Grad-CAM generates visual explanations by computing the gradient of the target class score with respect to feature maps of the final convolutional layer.
- The gradients are globally average-pooled to compute neuron importance weights. These weights are then used to create a weighted combination of feature maps, followed by ReLU activation, producing the final heatmap.

Pretrained AlexNet

- The architecture consists of five convolutional layers with ReLU activations and max-pooling.
- Pretrained weights are loaded to enhance feature extraction, and dropout layers are added to prevent overfitting.

Pretrained VGG-16

- The VGG-16 model includes multiple convolutional layers interspersed with max-pooling layers.
- Dropout layers are used before the second and final fully connected layers. Batch normalization is excluded to match the original VGG-16 configuration.

The models are initialized with pretrained weights to ensure efficient training and are evaluated using Grad-CAM to visualize class-discriminative regions.

3 Grad-CAM Mask Generation and Visualization

The Grad-CAM implementation generates visual explanations for CNN-based model predictions, highlighting image regions critical for specific class decisions.

GradCAM Class

- The class accepts a pretrained model and extracts feature maps from the last convolutional layer.
- save_gradient: A hook function saves the gradient of the target class score with respect to the convolutional features, enabling visualization of important regions.
- forward_model: Performs a forward pass, extracting the target feature maps and the class score.
- gen_CAM: Generates the Grad-CAM mask.
  - Computes feature maps and class scores.
  - Calculates gradients of the class score with respect to feature maps.
  - Global averages the gradients to obtain weights for each feature map.
  - Creates a weighted sum of feature maps to produce the CAM.
  - Resizes the CAM to the input image size and applies ReLU.
  - Normalizes the mask for visualization.

Preprocessing and Reprocessing

- preprocess_image: Converts an input image into a normalized tensor suitable for the model.
- reprocess_image: Converts the output tensor back into an image array for visualization.

Visualization

- Grad-CAM masks are applied to the input image to visualize the regions most relevant to the model's decision. The normalized mask highlights critical areas, aiding interpretability and transparency.

4 Grad-CAM Visualization with Test Images

This section demonstrates the application of Grad-CAM on pretrained models using test images, generating class-discriminative heatmaps to highlight image regions most relevant to the model's predictions.

Implementation Details

- Instantiation: Grad-CAM objects were created for AlexNet and VGG-16 models.
- One-Hot Encoding: For each test image, a one-hot encoded target vector was generated to specify the class for backpropagation.
- Grad-CAM Mask Generation
  - Gradients and activations from the last convolutional layer were utilized to compute the weighted activation maps.
  - The resulting Grad-CAM heatmaps highlight critical areas that influenced the prediction.
- Visualization
  - The original input image is displayed alongside Grad-CAM heatmaps for AlexNet and VGG-16.
  - Heatmaps are overlaid on the input image using a color map, with transparency to emphasize model focus areas.
- Findings
  - AlexNet and VGG-16 often focus on different parts of the image due to architectural differences, as evidenced by the heatmaps.
  - AlexNet and VGG-16 tend to focus on different parts of the image due to architectural differences.

5 Conclusion

Objective

- The Grad-CAM visualization experiment aimed to analyze how AlexNet and VGG-16 interpret input images by generating class-discriminative heatmaps for test labels such as "king snake" and "bull mastiff."

Process

- Grad-CAM was implemented for both AlexNet and VGG-16 using pretrained weights.
- Test images were processed, and heatmaps were generated by back-propagating gradients to the last convolutional layer.
- The resulting Grad-CAM heatmaps were visualized by overlaying them on the input images.

Findings

- Model-Specific Focus: The experiment revealed that AlexNet and VGG-16 focus on different regions of the same image due to differences in their architectures.
- Class Interpretation: For "king snake," VGG-16 focused more on the distinctive patterns of the snake's body, while AlexNet spread its attention across the image. For "bull mastiff," VGG-16 concentrated on the dog's face, while AlexNet distributed its focus to include both the dog and the surrounding area.

Conclusion

- The Grad-CAM visualization provided valuable insights into the interpretability of CNNs. By revealing the regions each model considered critical for its predictions, this experiment demonstrated the utility of Grad-CAM for understanding model behavior, diagnosing errors, and improving model transparency.