Deep Learning Practical Work 2-b: Solutions

Visualizing Neural Networks

Section 1 – Saliency Map

1. Show and interpret the obtained results.

Interpretation: The saliency maps highlight the pixels that have the most influence on the classification score. Visually, these maps align with the object of interest (e.g., the face of an animal, the body of a car), showing that the network focuses on the object rather than the background to make its decision.

2. Discuss the limits of this technique of visualizing the impact of different pixels.

Answer:

- **Noise:** The resulting maps are often noisy and speckled, making it hard to see fine details.
- **Gradient Saturation:** In deep networks, gradients can be unstable or vanishing, meaning the sensitivity map might not fully capture global importance.
- **Lack of "Why":** It tells us *where* the network looks, but not *what* features (curves, textures) it is detecting.
- **Single Instance:** It is specific to one image and does not explain the global logic of the model.

3. Can this technique be used for a different purpose than interpreting the network?

Answer: Yes, it can be used for:

- **Weakly Supervised Segmentation:** Generating segmentation masks (separating object from background) without having pixel-level training labels.
- **Saliency-based Object Localization/Cropping:** Automatically identifying the bounding box of the main object.

4. Test with a different network, for example VGG16, and comment.

Answer: When using VGG16 (a deeper and larger network than SqueezeNet), the saliency maps tend to be sharper and more focused on the object. This suggests that larger, better-performing models learn features that are spatially better aligned with the semantic objects in the image.

Section 2 – Adversarial Examples

5. Show and interpret the obtained results.

Interpretation: The generated adversarial image looks almost identical to the original image to the human eye (the perturbations are subtle high-frequency noise). However, the network confidently misclassifies it as the target class.

6. In practice, what consequences can this method have when using convolutional neural networks?

Answer:

- **Security Vulnerability:** Systems relying on CNNs (e.g., face ID, self-driving cars reading signs) can be tricked by malicious actors.
- **Trustworthiness:** It shows that high accuracy on a test set does not guarantee robust, human-like understanding of images.

7. Discuss the limits of this naive way to construct adversarial images. Can you propose some alternative or modified ways?

Answer:

- **Limits:** The naive gradient ascent does not bound the noise. If run for too many iterations, the image will look like garbage/noise rather than the original image, defeating the purpose of an "imperceptible" attack.
- **Alternatives:**
  - **FGSM (Fast Gradient Sign Method):** Single step update using the sign of the gradient, scaled by a small ϵ.
  - **PGD (Projected Gradient Descent):** Iterative updates with a projection step to ensure the adversarial image stays within a small ϵ-ball (L-infinity norm) of the original image.

Section 3 – Class Visualization

8. Show and interpret the obtained results.

Interpretation: The generated images represent the "platonic ideal" of the class according to the network. They contain characteristic textures and parts (e.g., eyes, fur, specific colors) associated with that class, often arranged in a dream-like, repetitive pattern.

9. Try to vary the number of iterations and the learning rate as well as the regularization weight.

Answer:

- **Iterations:** Too few iterations result in noise; too many can lead to high-frequency artifacts or saturation.
- **Learning Rate:** Higher rates converge faster but can be unstable. Lower rates require more iterations.
- **L2 Regularization:** Higher regularization forces the image to be gray/neutral, suppressing features. Lower regularization allows for high-contrast, high-frequency noise to dominate the visualization.

10. Try to use an image from ImageNet as the source image instead of a random image (parameter init_img). You can use the real class as the target class. Comment on the interest of doing this.

Answer: Initializing with a real image and optimizing for a target class (often different from the original) creates a "Deep Dream" effect. The network hallucinates features of the target class onto the structures of the source image. This is interesting for artistic generation and for understanding how the network interprets existing shapes as potential features for other classes.

11. Test with another network, VGG16, for example, and comment on the results.

Answer: VGG16 produces significantly better class visualizations than SqueezeNet. The images from VGG16 clearly show recognizable object parts (e.g., distinct legs, beaks, faces), whereas SqueezeNet visualizations are often more abstract and texture-based. This reflects VGG16's greater capacity and more detailed feature representations.

General Summary and Personal Understanding

Context and Problem Statement

In this practical work, we addressed the "Black Box" problem inherent to Deep Learning. While Convolutional Neural Networks (CNNs) like `SqueezeNet` and `VGG16` achieve state-of-the-art performance on classification tasks, their internal decision-making processes are often opaque. The primary objective was to explore methods that interpret *why* a network makes a specific prediction and to test the robustness of these predictions.

Approach: Gradient-Based Visualization

The unifying thread across all three methods explored in this practical is the use of **backpropagation to the input**. Unlike training, where we optimize weights to minimize loss, here we froze the network weights and computed the gradient of the loss (or score) with respect to the input image pixels ($\nabla_x \mathcal{L}$).

- **Saliency Maps (Interpretability):** By computing the gradient of the correct class score with respect to the input, we visualized which pixels most influenced the decision. This provided a "sensitivity map," confirming that the network generally focuses on the object's distinct features (e.g., a dog's face) rather than the background.
- **Adversarial Examples (Robustness):** By performing gradient ascent on a *target* (incorrect) class score, we discovered the brittleness of these networks. We showed that imperceptible perturbations, aligned mathematically with the network's gradients, can force the model to hallucinate a wrong class with high confidence. This highlights a significant security vulnerability in modern computer vision systems.
- **Class Visualization (Representation):** By optimizing a noise image to maximize a specific class score (with regularization), we generated "Platonic ideals" of classes. This revealed that the network stores "templates" of textures and shapes (e.g., feathers for a flamingo) to identify objects, offering a glimpse into the feature representations learned by the deep layers.

Personal Synthesis

This practical highlighted a fundamental duality in Deep Learning:

1. **High Accuracy vs. Low Robustness:** The existence of adversarial examples proves that while CNNs generalize well to natural data, they do not "see" widely like humans. They rely on specific, high-frequency patterns that are easily disrupted.
2. **The Versatility of Gradients:** We learned that the backpropagation algorithm is not just a training tool. It is a powerful generic optimization method that can be inverted to synthesize images, attack models, or debug predictions.
3. **Model Capacity Matters:** Comparing `SqueezeNet` and `VGG16` demonstrated that larger capacity models don't just yield better accuracy; they also learn more coherent and visually interpretable features, as seen in the sharper class visualizations.

Conclusion: Ultimately, these visualization techniques are essential tools for an AI engineer. They move us from blindly trusting metrics to understanding the model's behavior, ensuring that the network learns relevant semantic features rather than spurious correlations.