TP1 Solution: Transfer Learning with VGG16

Deep Learning Practical Work 2-a

January 13, 2026

Section 1: VGG16 Architecture

Question 1

Knowing that the fully-connected layers account for the majority of the parameters in a model, give an estimate on the number of parameters of VGG16 (using the sizes given in Figure 1).

Solution: The vast majority of parameters are in the first Fully Connected (FC) layer. The transition is from the last convolutional feature map ($7 \times 7 \times 512$) to the first FC vector (4096).

- **Input to FC1:** $7 \times 7 \times 512 = 25,088$ neurons.
- **FC1 Weights:** $25,088 \times 4096 = 102,760,448$ parameters (≈ 102 Million).
- **FC2 Weights:** $4096 \times 4096 = 16,777,216$ parameters (≈ 16 Million).
- **FC3 (Output) Weights:** $4096 \times 1000 = 4,096,000$ parameters (≈ 4 Million).

Total Estimate: $102M + 16M + 4M \approx$ **122 Million parameters** (just for the FC layers). The actual total for VGG16 is approximately **138 Million**.

Question 2

What is the output size of the last layer of VGG16? What does it correspond to?

Solution: The output size is a vector of dimension **1000**.

- It corresponds to the **1000 class scores** (logits) for the ImageNet dataset categories (e.g., Persian cat, Golden Retriever, etc.).

Question 3

Apply the network on several images of your choice and comment on the results.

- **What is the role of the ImageNet normalization?**
- **Why setting the model to eval mode?**

Solution:

- **Normalization:** The pre-trained weights were learned on images normalized with specific Mean (μ) and Standard Deviation (σ). Applying the same normalization to input images ensures that the data distribution matches what the network expects, leading to correct activations. Without it, predictions would be incorrect.
- **Eval Mode (`vgg16.eval()`):** This acts on layers like **Dropout** and **Batch Normalization**.
  - *Dropout:* Disables the random dropping of neurons (used only during training).
  - *Batch Norm:* Uses the global running statistics (mean/var) instead of the batch statistics.

Question 4

Bonus: Visualize several activation maps obtained after the first convolutional layer. How can we interpret them?

Solution: The activation maps of the first layer typically act as **edge detectors** or **color filters**.

- Some maps will highlight vertical or horizontal edges.
- Others might activate on specific colors (like green vegetation or blue sky).
- These are "low-level" features similar to Gabor filters.

Section 2: Transfer Learning with VGG16 on 15 Scene

Question 5

Why not directly train VGG16 on 15 Scene?

Solution:

- **Data Scarcity:** The *15 Scene* dataset is very small (4485 images) compared to ImageNet (1.2 million images).
- **Overfitting:** VGG16 has = 138 million parameters. Training such a huge model on a small dataset would immediately lead to severe overfitting (memorizing the training data without generalizing).

Question 6

How can pre-training on ImageNet help classification for 15 Scene?

Solution:

- **Feature Reusability:** The earlier layers of VGG16 have learned robust, generic features (edges, textures, shapes) that are useful for almost any visual task.
- **Transfer Learning:** By using these learned weights, we start with a "knowledgeable" feature extractor instead of random noise, requiring much less data to learn the specific high-level classes of the 15 Scene dataset.

Question 7

What limits can you see with feature extraction?

Solution:

- **Domain Shift:** If the source dataset (ImageNet) is too different from the target dataset (e.g., medical X-rays), the learned features might not be relevant.
- **Fixed Representation:** By freezing the network, the feature extractor cannot adapt to specific nuances of the new dataset (unlike Fine-Tuning).
- **High Dimensionality:** The extracted feature vectors (4096 dimensions) are large, which can be computationally expensive for the subsequent classifier (SVM).

Question 8

What is the impact of the layer at which the features are extracted?

Solution:

- **Early Layers (Conv):** Contain spatial, low-level information (edges). Dimensions are very large ($512 \times 7 \times 7$).
- **Later Layers (FC6/FC7):** Contain semantic, high-level information invariant to position. Dimensions are smaller (4096).
- **Impact:** Extracting from FC layers usually yields better classification performance for semantic tasks, while Conv layers might be better for tasks requiring spatial localization.

Question 9

The images from 15 Scene are black and white, but VGG16 requires RGB images. How can we get around this problem?

Solution: VGG16 expects an input tensor of depth 3 ($C = 3$). For grayscale images ($C = 1$):

- **Replication:** We duplicate the single channel 3 times to create a pseudo-RGB image.
- **Code:** `img = img.convert('RGB')` in PIL, or `np.stack((img,)*3, axis=-1)` in NumPy.

Question 10

Rather than training an independent classifier (SVM), is it possible to just use the neural network? Explain.

Solution: Yes, it is entirely possible and is known as **Fine-Tuning** (or transfer learning with end-to-end training).

- **Method:** Instead of extracting features and feeding them to an external SVM, we replace the final fully connected layer of VGG16 (which has 1000 outputs for ImageNet) with a new fully connected layer having **15 outputs** (for the 15 Scene classes).
- **Training:** We then continue the training process (Backpropagation) on the 15 Scene dataset. We can either:
  1. **Freeze** the convolutional layers and only train the new classifier layer (similar to the SVM approach but integrated).
  2. **Unfreeze** all layers and update the weights of the entire network (allows the feature extraction part to adapt specifically to the new dataset).
- **Comparison:** This approach is often more powerful than Feature Extraction + SVM because the network learns features *specifically* optimized for the target task, rather than just using generic ImageNet features.

Question 11

For every improvement that you test, explain your reasoning and comment on the obtained results.

1. Tuning the Parameter C (SVM)

- **Reasoning:** The parameter C in a Linear SVM controls the trade-off between the margin width and error penalty. A low C implies high regularization, while a high C aims to classify every training point correctly.
- **Obtained Result (With PCA):** We tested values from 0.001 to 10.0 on the PCA-reduced features (128 dims).
  - **Underfitting:** At $C = 0.001$, accuracy is lower (82.24%), indicating the model is too constrained.
  - **Optimal Zone:** Performance peaks at $C = 1.0$ with an accuracy of 88.14%.
  - **Overfitting:** At $C = 10.0$, accuracy drops to 87.37%, suggesting sensitivity to noise.
- **Obtained Result (No PCA):** Using the full 4096 features, the best accuracy was slightly higher at 88.81% (for $C = 1.0$), confirming that PCA causes a very minor loss of information ($< 0.7\%$) in exchange for speed.

2. Dimensionality Reduction (PCA)

- **Reasoning:** The raw VGG16 feature vector has 4096 dimensions. PCA allows us to compress this information into a smaller subspace to speed up training and reduce memory usage.
- **Obtained Result:**
  - We reduced the dimensions from **4096 to 128** (compression rate of $\approx 97\%$).
  - The accuracy dropped marginally from 88.81% (Full) to 88.14% (PCA).
  - **Conclusion:** The critical information for classifying the *15 Scene* dataset is effectively contained in a 128-dimensional manifold. The trade-off is excellent for computational efficiency.

3. Changing the Extraction Layer

- **Reasoning:** We compared extracting features from the last Fully Connected layer (`ReLU7`) versus the last Convolutional layer (`Conv5`).
  - **ReLU7 (FC):** Semantic, high-level features (objects). Dimension: 4096.
  - **Conv5 (Spatial):** Spatial features before flattening. Dimension: $7 \times 7 \times 512 = 25,088$.
- **Obtained Result:**
  - **ReLU7 Accuracy:** 88.81% (using standard SVM).
  - **Conv5 Accuracy:** 89.82% (using PCA to reduce 25k $\rightarrow$ 512 dims).
  - **Analysis:** Surprisingly, the spatial features from `Conv5` yielded the highest accuracy of all tests. This suggests that preserving some spatial layout information (before the aggressive flattening of FC layers) helps discriminate certain scenes in this specific dataset better than the abstract vectors of `ReLU7`.

General Summary & Conclusion

Context & Problem Statement

In the field of Computer Vision, Deep Convolutional Neural Networks (CNNs) like **VGG16** have achieved state-of-the-art performance on massive datasets such as *ImageNet* (1.2 million images). However, in many real-world applications, we are constrained by **data scarcity**.

In this practical, we faced the *15 Scene* dataset, which contains only approximately 4, 485 images. Training a deep network with 138 million parameters from scratch on such a small dataset is infeasible due to the high risk of **overfitting**—the model would memorize the training data rather than generalizing to new examples.

Proposed Approach: Transfer Learning

To overcome this limitation, we adopted a **Transfer Learning** strategy. The core idea is to leverage the robust, generic visual features (edges, textures, shapes, and objects) learned by VGG16 on ImageNet and apply them to our specific scene classification task.

Our approach consisted of several key steps:

1. **Feature Extraction:** We utilized the pre-trained VGG16 network as a fixed feature extractor. By removing the final classification layer, we treated the network as a function that transforms raw pixels into meaningful high-level feature vectors.
2. **Classification (SVM):** We trained a **Linear SVM** on top of these extracted features. Since the features were already highly discriminative, a simple linear model was sufficient to achieve high accuracy.
3. **Optimization & Analysis:**
   - We explored **Dimensionality Reduction** using **PCA**, compressing the feature space from 4096 to 128 dimensions. This demonstrated that the essential information lies in a lower-dimensional manifold, allowing for faster training with negligible loss in accuracy.

- We compared extraction at different depths (**ReLU7** vs. **Conv5**). Surprisingly, extracting spatial features from the convolutional layers (`Conv5`) combined with PCA yielded the best performance ($\approx 89.8\%$), suggesting that preserving spatial layout is crucial for scene recognition.

Conclusion

This practical highlighted the power and versatility of Transfer Learning. It effectively bridges the gap between massive, pre-trained models and small-scale specific tasks.

I observed that modern Deep Learning is not just about designing architectures but also about efficiently **reusing representations**. The ability to achieve nearly 90% accuracy on a small dataset without training a CNN from scratch demonstrates that learned visual features are highly transferable across different domains. Furthermore, the experiments with PCA and SVM regularization (C) emphasized the importance of hyperparameter tuning and data processing in building robust machine learning pipelines.