Mateo W. Racca

UNIVERSITAT ROVIRA i VIRGILI
Escola Tècnica
Superior d'Enginyeria

Artificial Vision and Pattern Recognition

Assignment #2: Action Recognition From Still Images Using Deep Learning Networks

Introduction

In this report, we examine the task of recognizing human actions from still images, a challenge that deviates from the traditional use of video footage in action recognition.

The focus is on understanding and categorizing human activities through single photographs using deep learning techniques. We utilize the Stanford 40 Action Dataset, which includes a wide range of human actions, for this purpose.

The report presents two approaches: developing a custom Convolutional Neural Network (CNN) and adapting pre-existing deep learning models like ResNet and VGG for this task.

Through rigorous training and validation processes, we aim to identify the most effective approach for action recognition from still images.

This study is designed to provide a real word experience from a regular student.

Methodology

The first step was to analyze the dataset and to work on the data loader.

The DataLoader plays a crucial role in streamlining the process of loading and preprocessing the data for the neural network. It ensures that the model receives the data in a format that's suitable for training and evaluation. Here are some key aspects of the DataLoader:

- ➔ Data Handling: It efficiently handles large datasets by loading data in batches, which is essential for training deep learning models on large datasets like Stanford 40.
- ➔ Batch Processing: The DataLoader divides the dataset into batches. Batch processing is crucial for optimizing the training process, allowing for more efficient gradient computations and updates.
- ➔ Shuffling: In the training phase, shuffling the data before each epoch helps in reducing overfitting and ensures that the model does not learn specific patterns related to the data ordering.
- ➔ Transformations and Augmentations: The DataLoader applies predefined transformations and augmentations to the data. This includes resizing images for uniformity, normalizing pixel values, and potentially augmenting the dataset to improve the model's generalization capabilities.

- ➔ Parallel Processing: It leverages parallel processing to load data, significantly speeding up the process, which is particularly beneficial when working with large datasets.
- ➔ Customization for Specific Tasks: The DataLoader is customized to suit the specifics of the Stanford 40 Actions dataset, ensuring that each image-label pair is correctly processed and presented to the model during training and testing.

In summary, the DataLoader is a critical component that efficiently manages data flow from the dataset to the neural network, ensuring acceptable data processing for effective model training and evaluation.

In the CNN approach for action recognition from still images, we constructed a custom Convolutional Neural Network (CNN) model. This model was designed to process the Stanford 40 Action Dataset, which includes a wide array of human actions in diverse settings.

CNN Architecture: the custom CNN model is comprised of several layers, each playing a critical role in feature extraction and classification:

- ➔ Convolutional Layers: these layers are the core building blocks of the CNN, responsible for extracting features from the images. We used multiple convolutional layers with varying filter sizes to capture different aspects of the input images.
- ➔ Activation Functions: we experimented with different activation functions like ReLU, Sigmoid, and Tanh in separate models to compare their impacts on model performance. ReLU is generally preferred for its ability to mitigate the vanishing gradient problem, while Sigmoid and Tanh are explored for their unique characteristics in data transformation.
- ➔ Pooling Layers: following the convolutional layers, pooling layers (specifically max pooling) were used to reduce the spatial dimensions of the feature maps, leading to a decrease in computational complexity and an increase in the receptive field.
- ➔ Batch Normalization: this technique was applied after the convolutional layers to normalize the activations and speed up the training process.
- ➔ Fully Connected Layers: at the end of the network, fully connected layers were employed to classify the extracted features into one of the 40 action categories.
- ➔ Dropout: in order to prevent overfitting, dropout layers were introduced, particularly before the fully connected layers.
- ➔ Training and Evaluation: the model was trained using a cross-entropy loss function and an Adam optimizer. The training process involved feeding batches of images into the network, calculating the loss, and updating the weights through backpropagation. The model's performance was regularly evaluated on a validation set to monitor its accuracy and generalization capability.

Key aspects of the training process included:

- ➔ Number of Epochs: The models were trained for a sufficient number of epochs to balance the risk of underfitting against overfitting.
- ➔ Learning Rate (LR): The LR was carefully chosen to ensure effective convergence without overshooting the optimal point in the loss landscape.
- ➔ LR Scheduler: We used ReduceLROnPlateau, which dynamically adjusted the LR based on validation performance, enhancing the fine-tuning during the later stages of training.
- ➔ Activation Functions: ReLu, Sigmoid and Tanh has been implemented in order to explore different options.

In our CNN approach, due to the computational limitations of the available hardware, we adopted specific training strategies. The number of epochs was capped to ensure manageability on the available system, avoiding excessive computational demands. We opted for smaller learning rates, which, although resulting in slower convergence, allowed for more precise model tuning within the limited number of epochs. Additionally, the 'ReduceLROnPlateau' learning rate scheduler was employed, effectively adjusting the learning rate in response to validation performance. This approach was instrumental in achieving the best outcomes possible under the given hardware constraints, resulting in notable accuracies with different activation functions.

In parallel to our custom CNN approach, we adopted the ResNet model, specifically ResNet18, for action recognition from still images. ResNet18 was chosen for its depth and efficiency, along with its proven track record in image classification tasks, making it an ideal candidate for our requirements.

The ResNet18 architecture utilizes deep residual learning, which involves shortcut connections that skip one or more layers. These connections help address the vanishing gradient problem, allowing for effective training of deeper networks.

We adapted ResNet18 to fit our specific task by modifying its final fully connected layer. This customization allowed the model to output 40 distinct action categories, aligning with our dataset.

Training Strategy and Hyperparameter Tuning

- ➔ The learning rate was carefully chosen to be small (0.0001), ensuring gradual and precise weight updates. This was crucial to refine the pre-learned features without causing drastic changes that could negate the benefits of pre-training.

- ➔ The AdamW optimizer, known for its improved weight decay management, was employed. This choice provided a balance between maintaining the stability of the pre-learned features and adapting them to the new task.
- ➔ ReduceLROnPlateau was used, reducing the learning rate when the validation loss stopped improving. This dynamic adjustment helped in overcoming plateaus during training.

Results

By varying activation functions of the CNN, we aimed to identify which one best suits the task of action recognition in static images. The comparative analysis of these models provided insights into the influence of activation functions on the learning dynamics and performance of CNNs in the context of action recognition.

This CNN approach represents a comprehensive attempt to tackle the challenging task of recognizing and categorizing human actions from static images using deep learning, offering a detailed exploration into the capabilities and limitations of CNN architectures in this domain.

The CNN model's performance with different activation functions can be compared based on the test accuracies obtained:

- Sigmoid Activation: the exact accuracy figure is not available due to an error in accessing the file, but it was noted earlier as being around 9.38%. This lower accuracy suggests that the Sigmoid function may not be ideal for this task, possibly due to issues like vanishing gradients, which are common with Sigmoid in deeper networks.
- ReLU Activation: the model achieved a test accuracy of approximately 15.06%. ReLU seems to perform better than the other two tested functions. Its ability to provide a linear, non-saturating response might have contributed to more effective learning and higher accuracy.
- Tanh Activation: the accuracy of the Tanh activation model is approximately 13.25%. However, Tanh, like Sigmoid, can suffer from vanishing gradients in deep networks, though it's zero-centered, which can be an advantage over Sigmoid.

Conclusions for the CNN Approach:

The ReLU activation function results to be more effective for this particular task of action recognition from still images, as evidenced by its higher test accuracy compared to Sigmoid and Tanh. Sigmoid and potentially Tanh might not be as suitable for deep CNN architectures due to their characteristics leading to potential issues in gradient flow. The overall

accuracies indicate that there is room for improvement. Future work might involve experimenting with deeper or more complex architectures, incorporating advanced techniques like transfer learning, or fine-tuning hyperparameters.

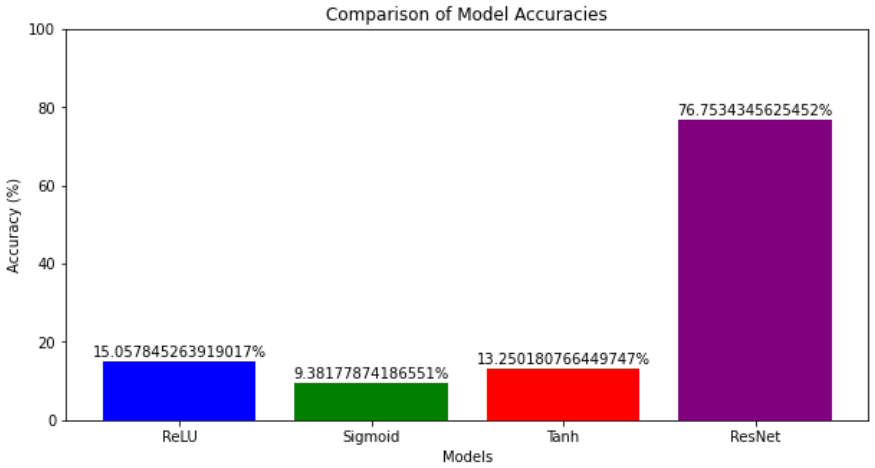
Considering the challenging nature of action recognition from still images, the results are encouraging and open avenues for further exploration and optimization of CNN models for this task.

On the other hand, the conclusions drawn from the implementation of the ResNet model in our action recognition project are:

- Effective Transfer Learning: the ResNet model, with its pre-trained architecture on ImageNet, demonstrated an impressive ability to adapt to the specific requirements of action recognition from still images. This underscores the effectiveness of transfer learning in leveraging pre-existing complex models for specialized tasks.
- Fine-Tuning and Parameter Optimization: the careful tuning of the learning rate, the use of AdamW optimizer, and the dynamic adjustments made by the `ReduceLROnPlateau` scheduler were instrumental in achieving high accuracy. These decisions highlight the importance of hyperparameter optimization in the successful application of deep learning models.
- High Accuracy and Robust Performance: the ResNet model reached a peak validation accuracy of 76.48%, showcasing its robust performance in the task. This high level of accuracy indicates the model's capability to effectively recognize a wide range of human actions from static images.
- Implications for Future Research: the success of the ResNet model in this context opens up new possibilities for employing pre-trained models in various domains of computer vision, especially where data or computational resources are limited.

These conclusions emphasize the potential of using advanced, pre-trained models like ResNet in specific and complex tasks, such as action recognition from still images, and provide a strong foundation for future explorations in this area.

Comparative Analysis



The CNN model's exploratory approach with various activation functions provided valuable insights into model behavior and feature extraction capabilities. However, the limitations posed by hardware constraints and the simpler architecture became evident in comparison to the more sophisticated ResNet model.

The ResNet model's approach, utilizing a complex pre-trained architecture and fine-tuned to the task, illustrates the efficacy of leveraging existing advanced models for specialized applications, achieving significantly higher accuracy.

Concluding Observations

These results underscore the potential and versatility of using pre-trained models like ResNet for complex tasks in computer vision, especially in scenarios with limited computational resources. On the other hand, the findings from the CNN models highlight the importance of activation function selection and suggest potential areas for model optimization and further research.

Future work could explore integrating more advanced techniques, experimenting with different architectures, or employing ensemble methods to enhance model performance and accuracy.

In conclusion, this work provides a comprehensive exploration of the capabilities and limitations of both custom-built and pre-trained models in the field of action recognition from still images, offering a foundation for future advancements in computer vision and deep learning.