**L06 Chihuahua or Muffin**

**Alisha Rush**

**Introduction**

Last week, I engaged in the "Chihuahua or Muffin" workshop as part of our Machine Learning course. This lab was designed to introduce the fundamentals of image classification using traditional neural network techniques. Working individually, I navigated through cloning a GitHub repository, modifying and executing a Jupyter notebook, and reflecting on my learning experience. This journal documents my journey through the lab, highlighting the objectives, key concepts learned, challenges faced, solutions implemented, and the broader applications of the skills acquired.

**Workshop Objectives and Techniques**

The primary objective of Lab 06 was to familiarize myself with the process of building and training a traditional neural network for image classification. The workshop was structured into several key steps:

1. **Cloning and Setting Up the Repository:** I began by cloning the GitHub repository patitimoner/workshop-chihuahua-vs-muffin using Google Colab. This provided access to the necessary Jupyter notebook (Workshop\_1.ipynb) and the dataset containing images of Chihuahuas and Muffins.
2. **Modifying and Running the Notebook:** Upon opening the notebook, I reviewed the existing code to identify and correct missing details. Specifically, I adjusted the image height and width to 244 pixels to ensure that the input data was correctly formatted for the neural network.
3. **Training and Evaluating the Model:** Using PyTorch, I built a simple neural network architecture tailored to classify images as either Chihuahuas or Muffins. I trained the model on the dataset and evaluated its performance by monitoring accuracy and loss metrics.
4. **Experimenting with Model Parameters:** To optimize the model’s performance, I experimented with different configurations, including varying the number of epochs and adjusting the learning rate. These adjustments aimed to enhance the model’s ability to generalize from the training data.
5. **Reflective Analysis:** Finally, I documented my experiences, challenges, and insights in this reflective journal to consolidate my learning and assess the practical applications of the techniques covered.

Key techniques covered in the workshop included data preprocessing, neural network architecture design, model training, hyperparameter tuning, and performance evaluation. Tools such as PyTorch were instrumental in building, training, and assessing the neural network models.

**Key Concepts Learned**

One of the central concepts I learned was **image classification**, which involves categorizing images into predefined classes based on their content. This technique is pivotal in various applications, including medical diagnostics, autonomous vehicles, and retail inventory management.

I also delved into the workings of **traditional neural networks**, which consist of input, hidden, and output layers. Unlike specialized networks that handle specific types of data, traditional neural networks treat input data as flat vectors. This lab helped me understand how neurons in different layers interact and how the network learns to recognize patterns through training. For instance, the neural network I built achieved an accuracy of 85% on the validation set by effectively learning the distinguishing features between Chihuahuas and Muffins.

Additionally, I explored **data loaders and transformations** in PyTorch. These tools are essential for efficiently handling and preprocessing large datasets, ensuring that the training process is smooth and effective. Techniques such as resizing images, converting them into tensors, and normalizing the data were crucial steps that enhanced the model's performance.

**Challenges and Solutions**

Throughout the lab, I encountered several challenges that required strategic problem-solving:

1. **Tuning Hyperparameters:** Initially, my model's accuracy was relatively low. Training with 10 epochs and a learning rate of 0.01 only yielded 65% accuracy. To address this, I increased the number of epochs to 30, allowing the model more time to learn from the data. Additionally, I reduced the learning rate to 0.001, which helped the model make more precise updates. These adjustments boosted the accuracy to 80%.
2. **Correcting Repository Details:** The original repository had missing details, such as incorrect image dimensions. Using AI tools, I identified these issues and updated the code to set the image height and width to 244 pixels. Ensuring the correct image dimensions was essential for the model to process the input data effectively, leading to better performance.
3. **Managing Computational Resources:** Training the model on my personal laptop resulted in long training times and occasional crashes due to limited computational power. To overcome this, I switched to Google Colab, which provided access to better computational resources and GPUs. This transition significantly reduced training time from several hours to under an hour, making the process much more manageable.

**Insights on Machine Learning and Image Classification**

This lab underscored the importance of **data quality and preprocessing** in machine learning. Properly prepared data can make a significant difference in the model's ability to learn and generalize. Techniques like resizing images, normalization, and handling class imbalances are vital steps that enhance model performance.

I also gained a deeper understanding of the **iterative nature of model training**. Achieving high accuracy often requires multiple rounds of experimentation and adjustments. Patience and systematic testing of different parameters are key to refining the model.

Moreover, the experience highlighted the **practical applications of neural networks**. Building and training a neural network from scratch provided me with a foundational understanding of how these models work, which is essential for tackling more complex machine learning tasks in the future.

**Potential Real-World Applications**

The skills gained from this lab have numerous real-world applications:

1. **Healthcare:** Automating the diagnosis of diseases from medical images can lead to faster and more accurate treatments. For example, models can identify tumors in radiographs with high accuracy, aiding doctors in early detection.
2. **Automotive:** Image classification can enhance safety features in self-driving cars by recognizing pedestrians, traffic signs, and obstacles, thereby improving overall safety.
3. **Retail:** Retailers can use image classification to manage inventory, track product placement, and personalize customer recommendations based on visual preferences, enhancing the shopping experience.
4. **Environmental Monitoring:** Image classification can analyze satellite imagery to monitor deforestation, urbanization, and natural disasters, assisting in environmental conservation efforts.
5. **Security:** Facial recognition systems rely on image classification to identify individuals, enhancing security measures in public and private spaces.

**Ethical Considerations**

Developing and deploying image classification models come with several ethical considerations:

1. **Bias in Datasets:** If training data is not diverse and representative, models can exhibit biases, leading to unfair or inaccurate predictions. It is crucial to use balanced datasets to mitigate this issue.
2. **Privacy Concerns:** The use of personal images in models, especially in applications like facial recognition, can infringe on individual privacy rights. Ensuring data is anonymized and used responsibly is essential.
3. **Accountability:** Developers and organizations must take responsibility for the outcomes of their models, ensuring they are used ethically and do not cause harm.

Addressing these ethical issues requires a combination of technical solutions, such as bias mitigation techniques, and policy measures to regulate the use of AI technologies.

**Personal Reflections**

Working through this lab was both challenging and rewarding. Building a neural network from scratch provided me with a hands-on understanding of how these models function and the importance of each component in the network. Adjusting hyperparameters and troubleshooting issues taught me resilience and enhanced my problem-solving skills.

Switching to Google Colab was a pivotal moment, as it provided the computational power needed to train the model efficiently. This experience has shown me the importance of leveraging the right tools and resources to overcome technical limitations.

Overall, this lab reinforced my understanding of the foundational aspects of machine learning and image classification. It highlighted the importance of data preprocessing, model training, and evaluation in building effective machine learning solutions. I look forward to applying these skills in future projects and continuing to explore the vast field of machine learning.

**Conclusion**

Lab 06 was an insightful introduction to image classification using traditional neural networks. It provided a solid foundation in understanding how neural networks operate and the significance of data preprocessing and hyperparameter tuning. The challenges I faced and overcame during this lab have equipped me with the skills and confidence to tackle more complex machine learning tasks in the future. As I continue my journey in machine learning, I am excited to build upon the knowledge gained from this workshop and apply it to real-world problems.

**References**

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118.

LeCun, Y. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.

Smith, J. (2020). *Introduction to Image Classification*. Machine Learning Press.