Ulises Zamarripa

Chihuahua or Muffin

Reflecting on the workshop, I found that switching from traditional neural networks to Convolutional Neural Networks (CNNs) significantly improved my understanding of deep learning. The old fully connected network felt clunky, like trying to make sense of a story from a random list of words, as it missed the spatial relationships between pixels.

The CNN was a game-changer for image analysis. Its design made sense. Convolutional layers acted like detectives, identifying basic features first, then recognizing more complex shapes as they advanced. The pooling layers helped streamline the information, making the model smarter and more adaptable to image variations.

The leap in performance from our first model to the CNN was truly impressive. The traditional network achieved a validation accuracy of around 80%, which seemed pretty good at the time. But when the CNN came along, boasting a validation accuracy of about 96.4%, it really put things into perspective. It showed that it was more accurate without the need to change anything

The training process was also quite eye-opening. The CNN trained significantly faster than its predecessor. This wasn't just a matter of speed; it showcased the efficiency of the design. Thanks to techniques like parameter sharing and pooling, the CNN had far fewer trainable parameters.

Analyzing the misclassifications was equally intriguing. They often occurred with the most ambiguous images. Those tricky ones that even I had to take a second look at. A muffin with a bumpy top or a chihuahua with a crinkled face in the shadows could easily confuse the model. This just goes to show that even a nearly perfect model has its limits.

The lab had its challenges, like grappling with a confusing RuntimeError due to mismatched tensor shapes. Debugging it with some print statements revealed the issue. Just a quick tweak to the in_features parameter solved everything.

I also struggled initially with training, choosing the wrong learning rates. One was too high, causing erratic loss, while another was too low, making progress slow. This trial-and-error process taught me the importance of fine-tuning hyperparameters to find the right learning rate for smooth model training.

The lessons I learned from these labs go beyond just a fun example. The ability to automatically identify images has huge potential in many areas. For instance, in medicine, it can help pathologists spot cancerous cells in biopsies. In self-driving cars, it enables vehicles to recognize pedestrians and road signs. It's thrilling to think about future uses, like farmers using drones to check on crops, and improving visual search on our phones.

But with this ability comes a big responsibility. We can't ignore the ethical issues that come with these technologies. A key concern is data bias. If a facial recognition model is trained on non-diverse data, it might misidentify certain groups, leading to unfair outcomes. As developers, we have to be aware of the societal effects of our work and stick to ethical standards. I'm really passionate about this topic and want to delve deeper into it. Overall, I've not only enhanced my technical skills but also developed a critical outlook on the technology I'm helping create. I'm excited to keep exploring this journey.