Lab: CNN Chihuahua or Muffin

Devin Prikryl-Martin

Houston Community College

ITAI 1378 Computer Vision

Professor Patricia McManus

October 10, 2024

Lab: CNN Chihuahua or Muffin

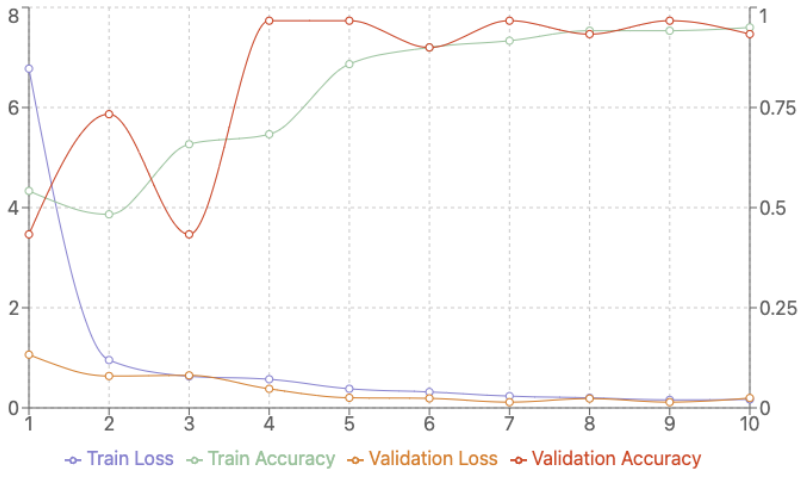
My lab experience today begins with diving into a bit of research about Convolutional Neural Netwoks so that I might better understand what they are and how they differ from the neural network we worked with last week. My takeaway from this preliminary research is that CNNs are a type of neural network that are suited to data that is in a grid-like structure. This makes it especially useful in the field of Computer Vision because digital images are essentially grids of pixels of various dimensions. CNNs are not limited to use on images, other data with grid-like structures can also benefit from the use of CNNs to detect patterns. The architecture of a CNN involves several different types of layers that the input travels through before reaching the traditional neural network, known as the "fully connected layer" which is often used to make the final determination or prediction. The architecture and the layers involved will depend on the task at hand and its complexity, but you will typically find convolutional layers, activation layers, and pooling layers in addition to the fully connected layers. The convolutional layer's process involves passing a small grid, called a kernel, over the image. The kernel represents a matrix of numbers that are used to multiply the pixel values beneath the kernel as it slides over the image. The pixel values are then summed and the resulting value represents a single pixel on a new output image. Different matrices are used depending on what features or patterns you are trying to extract from the image. Typically following each convolutional layer will be an activation layer, which will introduce non-linearity to the model, most commonly by employing the ReLU activation function. Then you find pooling layers, which are used to condense the data in the grid mathematically to reduce the size and computational power required to assess the grid. I will now take this knowledge and begin the lab.

This lab begins, as usual, with preparing the necessary libraries and then moves on to preparing the data that the model will train and validate with. We can see that the training set contains 65 images of chihuahuas and 55 muffins, while the validation set has 17 dogs and 13 pastries. The split between 120 training images and 30 validation images aligns with the 80/20 rule commonly used for splitting data during training. With our datasets in place, we begin the process of applying transforms to our image data, which is the preprocessing step. I chose to input the same 224x224 image dimensions for this lab as I had chosen for the previous lab because it will allow me to more accurately compare the models. The final step in the data prep involves loading the datasets and creating the dataloaders, which feed data into the model. Next we define the model, which looks like laying out the structure of the model including the information about the layers and their actions. With a model framework ready, we prepare to train the model by defining the loss function and optimizer, which includes the learning rate. In this case the optimizer selected

is Adam and the learning rate is set to 0.001. Then we train the model. It will run for 10 epochs and then we will be able to assess its performance.

After running for 10 epochs and 4 minutes we can see the results of the training. By the final epoch the training loss was 0.169, the training accuracy was 0.95, the validation loss was 0.197 and the validation accuracy was 0.93. We can then see the visualization of the model's performance, including two examples of the model failing to properly classify the images. Interestingly, both misclassified images were chihuahuas that were incorrectly categorized as muffins. I theorize that this might have to do with the small number of images of muffins the model was trained on.

I'm going to try something a little different this lab to enhance my understanding of the results of the model's performance. I'm going to feed the training result data into Claude and ask for an assessment of the data and suggestions for potential improvements to the model. Given that I am a novice programmer this seems like the most realistic way to enhance my technical understanding in a short period of time. Claude provides interesting insights that I probably wouldn't have picked up on, such as validation accuracy fluctuating between epochs that could indicate some problem with the validation data, or signs of slight overfitting in the final epoch with the training accuracy being slightly higher than the validation accuracy. Claude also suggests that the learning rate might be too high, which can be seen in a large drop in training loss from epoch 1 to 2. The recommendations Claude makes to help improve the model based on the very limited amount of data I gave it are: implementing a learning rate scheduler to reduce the learning rate over time, adding regularization techniques such as dropout layers or data augmentation, increasing the dataset size, BOTH stopping the epochs around 7 or 8 to prevent overfitting AND training the model for more epochs to see if it improves on its own, and tuning the architecture. It also produced, without me specifically asking, this wonderful graph to help visualize the training data. (Claude, 2024)



I will now attempt to experiment with the model by adjusting parameters. I begin with epochs, first reducing to 7, which shows even more instability in the validation accuracy at the end. Then upping it to a lengthy 15 epochs, which takes 6 minutes and yields the same sort of yoyo-ing of the validation accuracy. I return the epochs to 10 and turn to adjusting the learning rate to see if that helps. I adjust it to 0.0001, slowing the learning rate in hopes of stabilizing the training output. This seems to result in overfitting, with the test accuracy quickly reaching 100 and validation accuracy peaking at epoch 6 and then lessening to 90. I then adjust the learning rate to 0.01, which absolutely tanks the loss and accuracy scores, with a final validation accuracy of 0.76. Resetting the lr to 0.001, I then implement architecture changes, adding batch normalization and another convolutional layer. This yields a validation accuracy of .93, which is right where we started. This model seems much more finicky, more time consuming and less accurate than the deep learning neural network we used to solve this same problem last week. I am sure I just haven't found the right hyperparameters to adjust to improve it.

Deploying models like this are seemingly innocuous but the technology raises several ethical concerns. A main issue is the using of unethically sourced data sets, either copyrighted material or private in nature and the misuse of that data. What comes to mind immediately is an article I saw last week in which Harvard students built a computer vision model that worked with the Meta glasses to send the live video to a model that searched Instagram for facial recognition matches and then used that information to find that person's publicly available information online via sources like voting registration information. Within seconds of meeting a stranger, the person wearing the glasses could have all of the stranger's information sent to their phone. The demonstration, while impressive, is clearly meant as a cautionary example against how easy it is to build and deploy something that can be used for incredibly nefarious means. (Choo, 2024)

References:

3Blue1Brown. (2022, November 18). But what is a convolution? [Video]. YouTube. https://www.youtube.com/watch?v=KuXjwB4LzSA

The Julia Programming Language. (2020, September 3). Convolutions in Image Processing | Week 1, lecture 6 | MIT 18.S191 Fall 2020 [Video]. YouTube. https://www.youtube.com/watch?v=8rrHTUzyZA

Futurology — An Optimistic Future. (2020, December 19). Convolutional Neural Networks Explained (CNN Visualized) [Video]. YouTube. https://www.youtube.com/watch?v=pj9-rr1wDhM

Anthropic. (2024, October 9). Claude. "Assess the results of this CNN model's performance and give suggestions for improvement."

Choo, L. (2024, October 7). *How 2 students used the meta ray-bans to access personal information*. Forbes. https://www.forbes.com/sites/lindseychoo/2024/10/04/meta-ray-bans-ai-privacy-surveillance/