ITAI 1378 Computer Vision
Spring 2024
Professor: Patricia McManus
Mayela Miguel

L07 Chihuahua or Muffin with CNN

The architecture of this model is made of 3 convolutional layers with 32 input channels meaning it can process 32 pictures, with 3 kernerls and 1 padding, using ReLU after each layer, in attachment with the pooling layer. CNN architecture is easier to learn features more rapidly and efficiently than the traditional neural network we used in the previous workshop.

In our model performance, the validation accuracy was successful with a score of 0.9333 out of 1. Although the accuracy was high, both missed predictions the model believed were muffins, but instead it was a chihuahua.

If we added more layers, we could learn more complex features, improving our accuracy, but if we added fewer layers, we could simplify the model to prevent overfitting but limit the capacities of the model to learn more slowly and not be able to learn complex patterns.

https://www.camelot-mc.com/blog/deep-learning-adding-layers-to-the-network/#:~:text=Hence%2C%20in%20a%20deep%20learning,features%20learned%20in%20previous%20layers.

Adding more epochs can achieve better training accuracy but the risk again of overfitting if the model begins to learn noise in the training data, a model with less epochs just learns the underlying patterns not the full picture leading to lower accuracy.

https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

Also, the learning rate can be changed to a higher degree in order for the deep learning model to be trained faster but won't properly converge leading to loss and less accuracy. Evidently lowering the learning rate can result in slow convergence which can improve accuracy but takes longer to train.

https://www.purestorage.com/knowledge/what-is-learning-rate.html#:~:text=A%20high%20learning%20rate%20can,get%20stuck%20in%20suboptimal%20solutions.

I wanted to experiment by increasing the learning rate and epoch to improve accuracy to the model and see how it would impact the accuracy overall. I was wondering if it was possible that the accuracy would be 100% with the same layers, telling the model to iterate more times by increasing the epoch one more time, with a faster learning rate of 0.03. If the loss does not decrease accordingly or there is a spike, then I should lower the learning rate.

While I was making all these changes to the model, it turns out the biggest trouble I had was finding which parts of the code did what. Although the comments did help, it was a bit of a pain to look through each part of the code and find what I had to change.

On my first attempt I decided that 3 layers was good enough, so I moved on and ran the module with 14 epoch and the learning rate set to 0.01, yes, the accuracy decreased to 0.5667, half of what it was. It could be due to the low number of layers; if the epoch is increased, the model reaches a limit on how much it actually learns; I could even lower the epoch and the accuracy would be the same; the loss lowers only a little bit, and the accuracy slightly improves then stays the same and never reaches 0.7 accuracy.

On my second try a bit later I kept the three layers, but I lowered the epoch to eleven and also decreased the learning rate to 0.002. The accuracy of the model increased significantly it even surpassed the initial accuracy of the model.

Real-world applications examples are face recognition and object recognition. It could be used for identifying criminals via CCTV and identifying their features so if they’re seen in public through another camera, they can check if they have a match. It’s also used in smartphones. FaceID is a straightforward choice to keep your phone safe from others.