Assignment 1 CNNs

By: Benjamin Trenery

Introduction

Convolutional neural networks is a type of machine learning that utilizes deep neural networks. Deep neural networks are used to find relationships between data through neurons. This is done by creating a neural network, which is based on how neurons work in the human brain. There are three types of layers in neural networks, the input layer, the hidden layer, and the output layer. The input layer is composed of input data that is separated into different nodes. From the input layer, the hidden layer calculates weighted values for the strength of relationship between the data. This is done by calculating the summation of each input node with an associated random initialized weight for a new node. Then, the bias of the new node is added to the calculated value and put through an activation function. An activation function adds non-linearity to the model, which makes it able to detect more complex relationships that are non linear. The outputs of these new nodes become the inputs for the next layer. This is done until the output layer, which gives a final probability to represent the classification. For a convolutional neural network, this architecture is slightly altered. The input data is separated by pixel into a matrix of weights. A filter matrix is then put over the input matrices to form a feature map. To create a feature map, one must find the dot product of the input matrix and the features. These are called convolutional layers. These layers find features in the image that have a relationship to the classification. Finally, the output layer remains the same.

Background

This assignment tasked with creating a convolutional neural network on an image dataset for a classification model. A deep neural network was needed in order to find the relationship between the pixels in the image. Less sophisticated models, such as the decision trees, cannot capture the relationships between the data. As a test, a decision tree classifier was created to test its effectiveness for classifying the image. The accuracy of the decision tree model was 26.7 percent, which is not a high accuracy rate. The ability to create an accurate image classification model is important in modern society. Some problems that require image classification include automated cars, robotics, and medical diagnostic machines. These are just some areas that use image classification.

Methods

In this assignment, two models were created. The first model did not include techniques for reducing overfitting while the second one did include techniques. For the model, the data came from cisar10 dataset. This dataset contains 60,000 32 by 32 colored images. There are 10 classes of images. These classes are: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The cisar10 dataset library was available through keras. For the first model, in order to train, test, and optimize the data, a train test split was utilized. A train test split works by randomly separating the data into two categories. The training data is used to train the model. Then, the testing data simulates unseen data in order to determine the effectiveness of the model. The train and test sets were then normalized by rescaling the image to be from 0 to 1. In order to normalize the data, the training and test set were divided by 255. The number 255 represents the default range of pixel values. The training set was further divided into a validation set, which is used for metric analysis of the model. Then, the architecture for the convolutional neural network was created. It included convolutional layers with a relu activation function. It also utilized max pooling to improve the efficiency of the model. The model used the Adam optimizer with a sparse categorical cross entropy loss function and an accuracy metric. Finally, the first model was fit to the training data using 20 epochs and compared to the validation data. In the second model, a random horizon flip augmentation was used to increase the size of the dataset as well as improve the overall generalization of the model. Then, in the architecture of the model, batch normalization was used to improve efficiency and reliability of the model. Next, the drop out function was used in order to improve generalizability by removing random nodes from the neural network.

Results

To determine the effectiveness of the two models, the accuracy and loss metric were utilized for analysis. Loss represents the difference between the predicted and actual values in the machine learning models. The accuracy and loss was separated by the training data and the validation data. Each model ran for 20 epochs. An epoch represents one pass through the model. Accuracies closer to 1 and losses closer to 0 are indicative of an effective model. Figure 1 and Figure 2 represent the accuracy and loss of the first model and Figure 3 and Figure 4 represent the accuracy and loss of the second model. In figure 1, the training data set had a loss of about 1.6 on the first epoch and 0.2 on the last epoch. The validation data set had a loss of about 1.3 in the first epoch and 1.2 in the final epoch. Figure 2 displays the training data set had an accuracy of about 0.4 in the first epoch and about 0.9 in the 20th epoch. For the validation data set, it had a training data set accuracy of about 0.5 in the first epoch and about 0.65 in the last epoch. The results in figure 3 for the training data set show a loss of about 0.67 in the first epoch and 0.2 in the 20th epoch. For the validation data set it had a loss of 0.5 in the first epoch and 0.43 in the final epoch. Figure 4 displayed an accuracy of 0.78 and 0.9 in the first and last epoch respectively for the training set. Accuracy for the validation data set included about 0.83 in the initial epoch and about 0.9 in the 20th epoch.

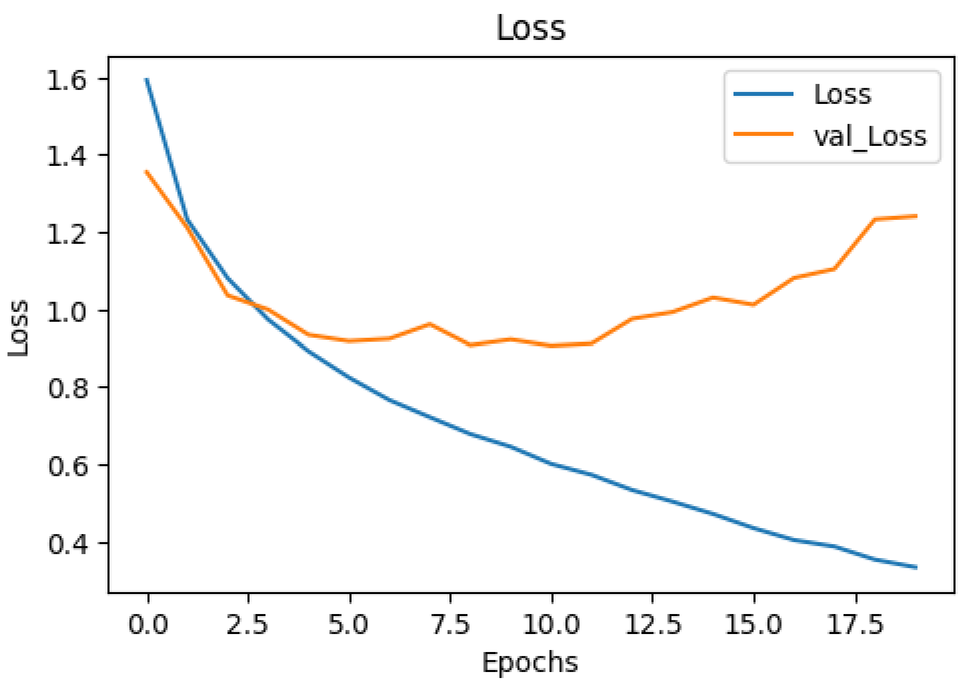

Figure 1 Convolutional Neural Network for model one showing epochs on the x-axis and loss on the y-axis. The loss of the training data set is compared to the loss of the validation data set.

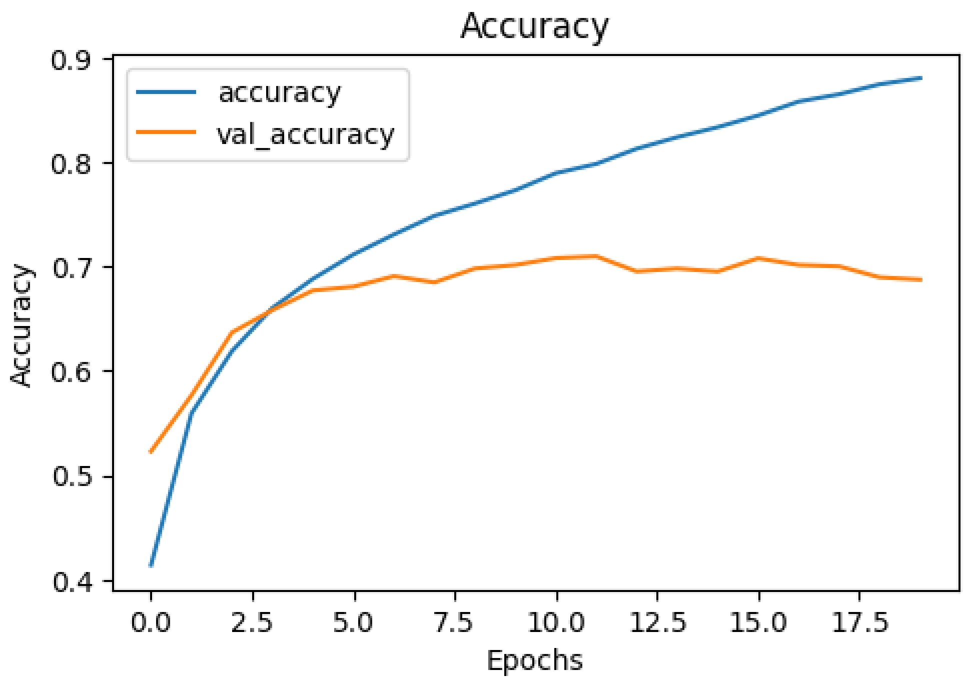

Figure 2 Convolutional Neural Network for model one showing epochs on the x-axis and accuracy on the y-axis. The loss of the training data set is compared to the loss of the validation data set.

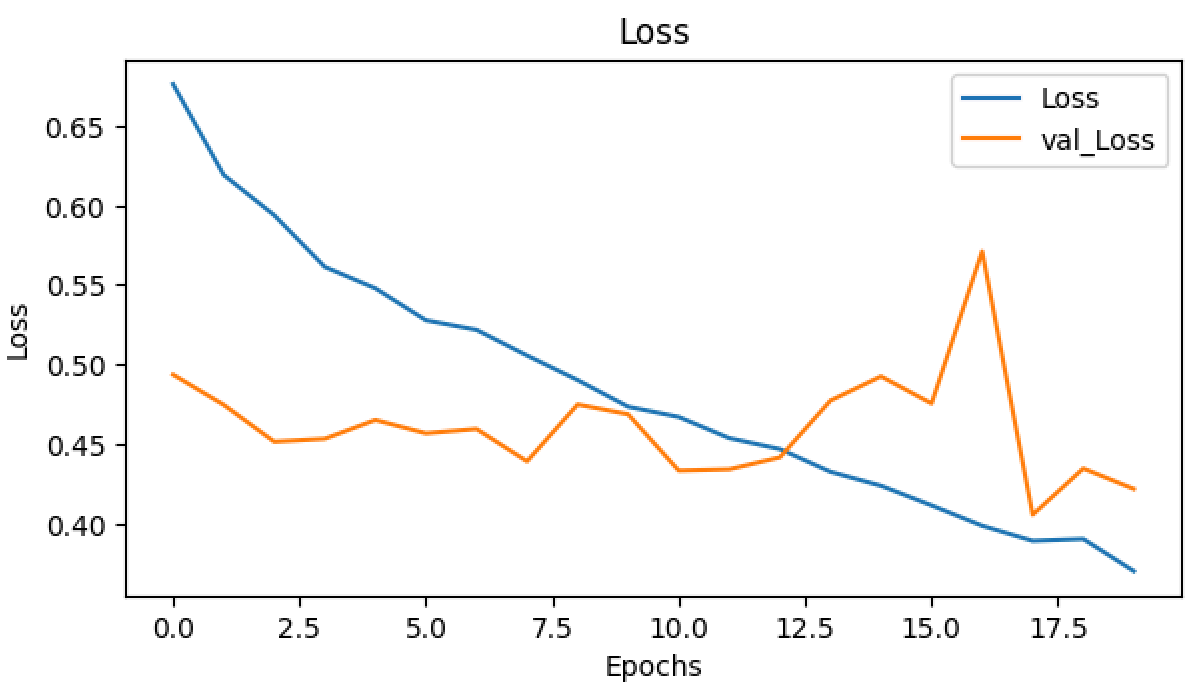

Figure 3 Convolutional Neural Network for model two showing epochs on the x-axis and loss on the y-axis. The loss of the training data set is compared to the loss of the validation data set.

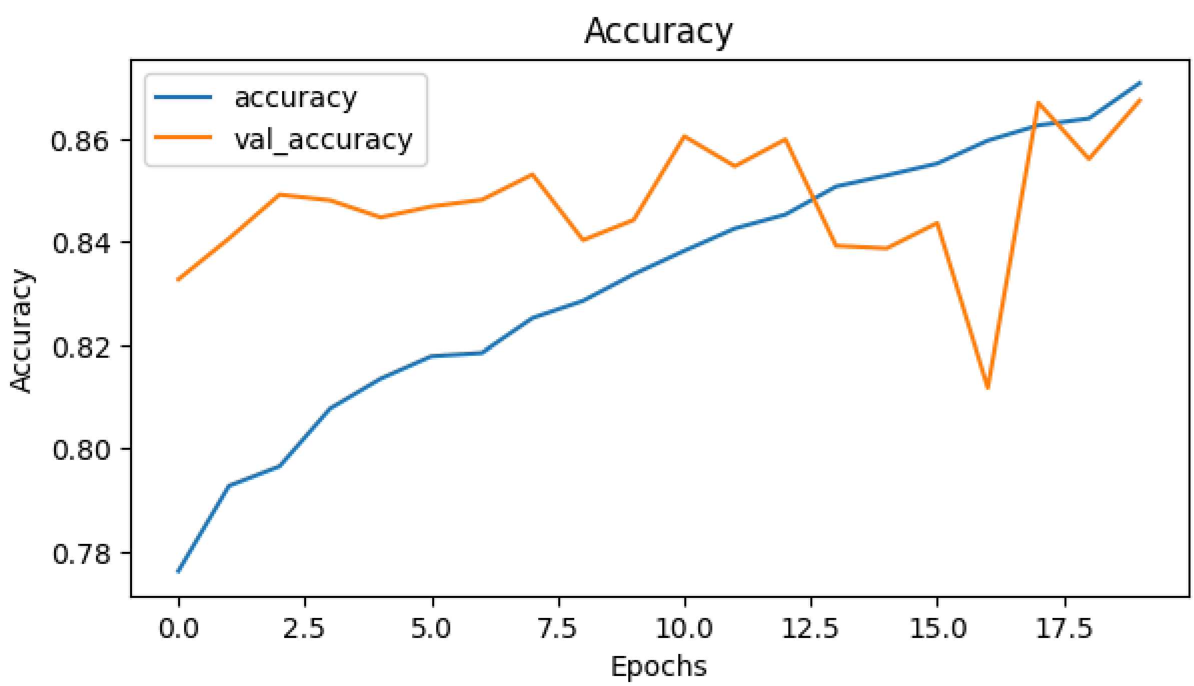

Figure 4 Convolutional Neural Network for model two showing epochs on the x-axis and accuracy on the y-axis. The loss of the training data set is compared to the loss of the validation data set.

Discussion

Overall, there was an increase in the initial and final accuracy of the validation data set in the second data set over the first. The loss was also considerably less for the second model over the first for the validation set. Like the testing data, the validation set represents unseen data to the model. The ability to accurately predict new data outside the training set is the goal of prediction modeling. Generalizing the model helped to make it more effective in this regard. When a model overfits, it becomes less effective when working with new data. By augmenting the data, batch normalizing the convolutional layers, and using the drop out method, the model was more effective at classifying new data. However, in the real world, an overall accuracy of about 90 percent is not effective enough. This would mean that about 1 in 10 images are incorrectly classified. Having an accuracy closer to 1 and a loss of 0 would be a more effective model. Some ways to improve the accuracy of the model would be to use more training data, fit the model over more epochs, adjust the learning rate and add more convolutional layers. Having more data allows for more relationships and characteristics to be found. Using epochs to optimize the model is an effective way to improve the accuracy. However, it can also lead to overfitting if one does not add more generalizability to the model. Learning rate represents the speed that the model learns. This is a hyperparameter by the data analyst. Adjusting the rate could improve the speed in which the model becomes more accurate. The learning rate is used in back propagation by adjusting the loss function. Finally, adding more convolutional layers gives the model the ability to find more relationships between the data. One downside to this approach is that it can make the model take longer to train. This assignment helped to better understand the principles of convolutional layers and how to make them more effective. Overfitting was one of the problems I had with my initial model, which is why I decided to incorporate more generalization to the model.