Jay Patel

jrp2212

AI

1. **What is one-hot encoding? Why is this important and how do you implement it in keras?**
    a. One-hot encoding is a method used to represent categorical data in a format that is compatible with machine learning algorithms. It is important because many machine learning algorithms require the input data to be in a numerical format, and cannot handle data in the categorical format directly. You can implement one-hot encoding using the "to_categorical" function. Example of an implementation is as follows: "one_hot_labels = to_categorical(labels)". Essentially this function takes an array of integers and converts it to a "readable" matrix.
2. **What is dropout and how does it help overfitting?**
    a. Dropout is a technique used to prevent overfitting (as hinted by the question). It works by randomly dropping out a certain number of output features of a layer during training. This reduces the inter-dependency of the remaining features and instead has the model you are training to learn new features. Again, this prevents overfitting because the model is not as reliant on a single feature.
3. **How does ReLU differ from the sigmoid activation function?**
    a. ReLU is a very intuitive activation function. It takes all negative values and turns them into zero and the positive values stay the same. Not only does this help in comprehension to a degree but it also assists in improving the convergence of a model. The sigmoid activation function is a little bit more complex. All the values passed through are mapped to a value between 0 and 1 and the function itself if graphed represents a s-shaped curve.
4. **Why is the softmax function necessary in the output layer?**
    a. The softmax function is used in the output layer of a neural network to map the output of the network to a probability distribution over the classes, which makes it easier to interpret and compare the output to the true labels.
5. **What are the dimensions of the outputs of the convolution and max pooling layers?**
    a. The size of the convolution layer is 96 while the output dimension of the max pooling layer will be 48x48x16.

About the Model:
- This model was mostly created via trial and error. I would add and delete layers and change the value for the dropout and then measure the accuracy of the model. If I thought that I was going somewhere with a specific layout, I would keep that layout and then make small tweaks. I continued to do this until I got to around an average of a 91-92% accuracy rate.