Homework 7-8: Face Recognition

Nathan Stallings

11/22/2021

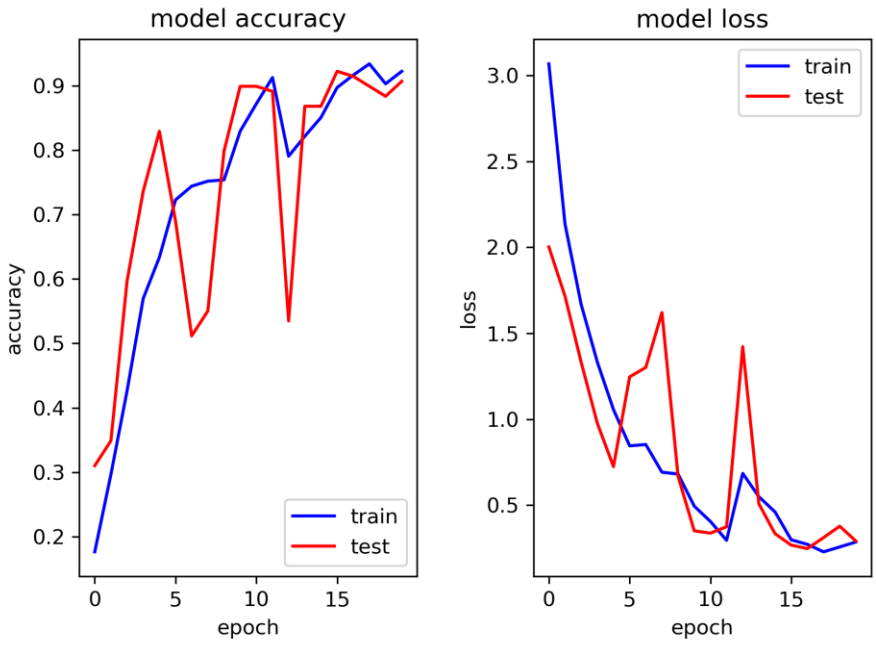
To generate a face recognition model, I used a CNN model with dropout and a categorical_crossentropy loss function. I based my first pass of the model on previous code we had in class, specifically the CNN_MNIST model. My initial model looked like:

```
model = Sequential()
model.add(Convolution2D(32, 5, 5, input_shape=(192, 168, 3), activation='relu'))
model.add(Convolution2D(32, 5, 5, activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.25))

model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dropout(.50))
model.add(Dense(numPeople, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

I trained this model with a batch size of 5 over 20 epochs. It had the following performance:

CNN Error: 9.30%



Homework 7-8: Face Recognition

Nathan Stallings

11/22/2021

The spikes in accuracy and loss signify that the model isn't converging, and the error is deceptive for model performance. To improve this model, I decided to add another set of Convolution2D layers to the model. My updated model looked like this:

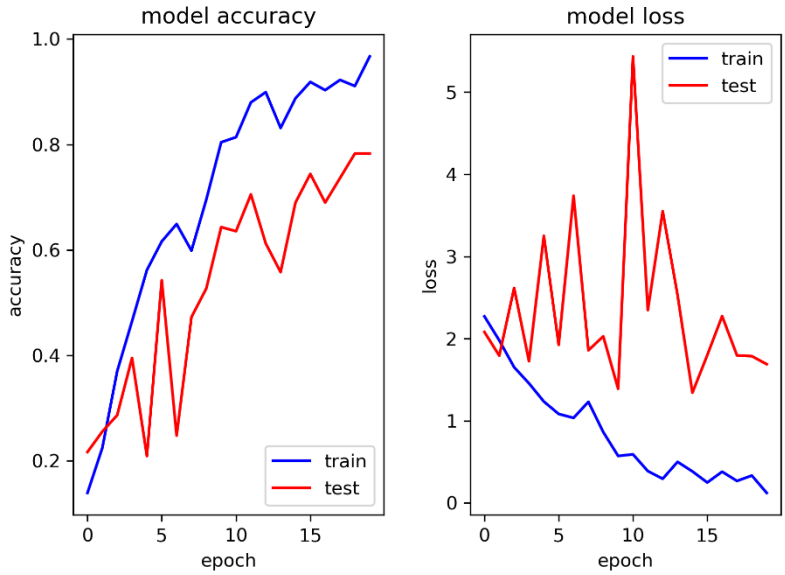
```
model = Sequential()
model.add(Convolution2D(32, 5, 5, input_shape=(192, 168, 3), activation='relu'))
model.add(Convolution2D(32, 5, 5, activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.25))

model.add(Convolution2D(64, 5, 5, activation='relu', padding='same'))
model.add(Convolution2D(64, 5, 5, activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.25))

model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dropout(.50))
model.add(Dense(numPeople, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Attempting to improve performance, I decrease the bath size to 1. I trained this model with a batch size of 1 over 20 epochs. It had the following performance:

CNN Error: 21.71%



Homework 7-8: Face Recognition

Nathan Stallings

11/22/2021

The accuracy of the last model seemed more consistent towards the end, so I tried adding another dense layer to the model. My updated model looked like this:

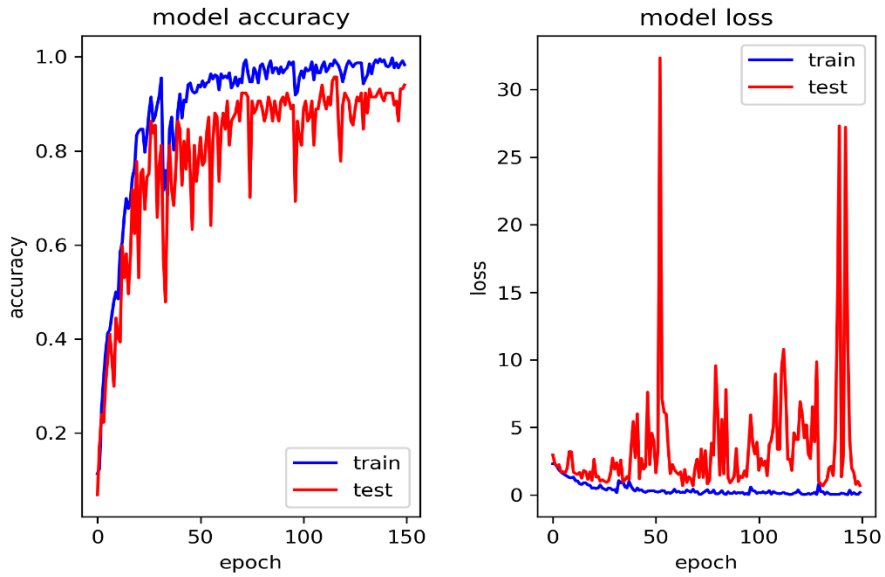
```
model = Sequential()
model.add(Convolution2D(32, 5, 5, input_shape=(192, 168, 3), activation='relu'))
model.add(Convolution2D(32, 5, 5, activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.25))

model.add(Convolution2D(64, 5, 5, activation='relu', padding='same'))
model.add(Convolution2D(64, 5, 5, activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
model.add(Dropout(.25))

model.add(Flatten())
model.add(Dense(200, activation='relu'))
model.add(Dropout(.50))
model.add(Dense(100, activation='relu'))
model.add(Dropout(.50))
model.add(Dense(numPeople, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Attempting to improve performance, I increased dropout rates and trained this model with a batch size of 1 over 150 epochs. It had the following performance:

CNN Error: 5.98%

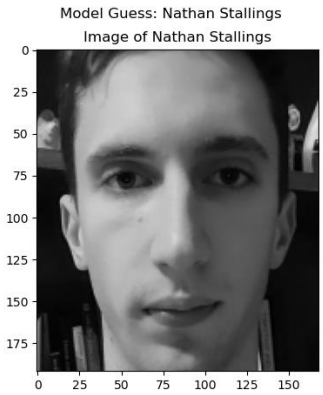


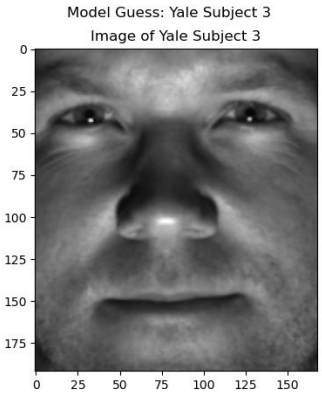
Homework 7-8: Face Recognition

Nathan Stallings

11/22/2021

While the model loss had some spikes, the accuracy performed much better for this final model. An error rate of 5.98% is good for the created model, and many of the misclassifications are due to very dark images in the dataset where much of the participants face is obscured. On the display data set, which has been filtered to remove many of the completely dark images, the model reports an error of 1.96%. Below are two example outputs from the model:





The code for this homework assignment can be found at:
https://github.com/nastallings/Deep_Learning-Face_Recognition