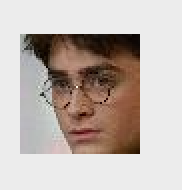
Part 1

Sample 1:

Chopped



Original

图片包含 人员, 室内, 外表, 墙壁

描述已自动生成:

Sample2:

Original:

Cropped:



图片包含 人员, 室内, 就坐, 孩子

描述已自动生成

Sample 3:

Original

Cropped:

图片包含 服装

描述已自动生成

图片包含 人员, 室内, 妇女, 就坐

描述已自动生成

1. The provided data sets are accurate for most bounding box. For example, for image “radcliffe53.jpg” (the first sample above), it successfully cropped character’s face. But for some of the cropped images, they either contain extra “information “or not accurate. For example, for sample 2, it has another person’s face on the right. Also, for sample 3, it doesn’t contain character’s face, so, during the process of data training, it may cause inaccuracy.
2. It is important because without the cropping process, there are a lot of external factor could affect our prediction, like the background, other people’s face, etc. For example, for sample 2, the original image also contains baby’s face, but what we want is the lady’s face as sample during training. By cropping the image, it could prevent the influence of external factors and increase the prediction accuracy.
3. By resizing the image, we could save the compile time without lose an information (because resize an image doesn’t change the amount of data in that image).

Part 2

1. 图片包含 地图

   描述已自动生成
2. and 3.

In the process of data splitting, the preprocessed image and the image label were splitted into X\_train, X\_test, y\_train, y\_test by 0.2 (80% training sets and 20% testing sets). Then for X\_train, and y\_train, it was further splitted by 0.2 (80% training sets and 20% testing sets) to X\_train, X\_valid, y\_train, y\_valid.

Therefore, for X\_valid, it contains 20% to the training data.

For y\_valid, it contains 20% of the testing data.

For X\_train, it contains 80% of the training data.

For X\_test, it contains 20% of preprocessed image data.

Preprocessed the input:

For preprocessed image, firstly convert all the images to greyscale, so that all the image only has one channel. Then, use the get data function to get the information from the image and divide the data by 255. After the division, append the image data to preprocessed image list. Therefore, after the iteration, preprocessed image list will have all images ‘data

For label, it uses file name to identify characters. For example, for file name “bracco0.jpg”, since it contains “bracco”, label it with “0”. After go through all the filename, the label list will have all the characters labeled.

|  |  |
| --- | --- |
| Character Name | Label |
| Bracco | 0 |
| butler | 1 |
| Gilpin | 2 |
| Harmon | 3 |
| Radcliffe | 4 |
| Vartan | 5 |

Activation function used:

1. “relu”: Rectified Linear Unit.
2. “softmax”: Softmax activation function.

Number of hidden layer node selected: 100

Compiling result:

图片包含 文字

描述已自动生成

Part 3:

1. For “block4\_pool”

图片包含 文字

描述已自动生成

For “block5\_pool”

图片包含 文字

描述已自动生成

For VGG16 layer, since “block4\_pool” and “block5\_pool” has similar accuracy (block4: 0.798, block5: 0.771), but “block5\_pool” has significantly lower test loss (block4: 293.26, block5: 1.37). Therefore, I choose “block5\_pool” as VGG16 layer.

1. VGG16 yields better accuracy, because for VGG16, we use a CNN to pre-train the image data set. Also, for VGG16, it drives high level features from the image’s visual content using fulling connected layer (from the lecture slide). Also, for VGG16, it has many weight parameters, so it increases the accuracy of image recognition.

Part 4

Original parameter:

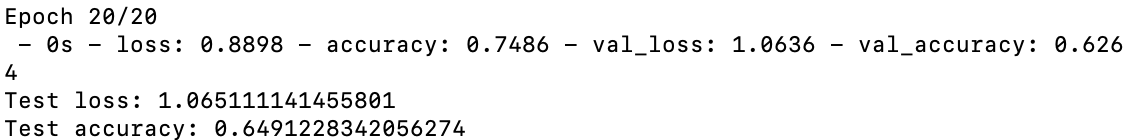
Number of Hidden Layer: 1

Number of Node in hidden layer: 100

Dropout: no dropout

Batch size: 128

Epoch: 20



Add dropout to be 0.1 and keep other parameters the same:

Number of Hidden Layer: 1

Number of Node in hidden layer: 100

Dropout: 0.1

Batch size: 128

Epoch: 20



Base on the output, add dropout doesn’t improve the accuracy at lot, it rather slightly decreases the accuracy (0.649-0.622). That is because when the data set is small, adding dropout rate would decrease its accuracy. Adding dropout would only increase the accuracy if the data set is huge, because since the data set is big enough, overfitting is not a problem.

Add another hidden layer and set the node to be 50:

Number of Hidden Layer: 2

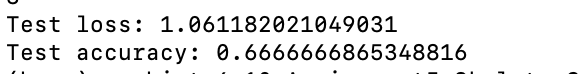
Number of Node in first hidden layer: 100

Number of Node in first hidden layer: 50

Dropout: no dropout

Batch size: 128

Epoch: 20



Base on the output, add another hidden layer does not improve the accuracy a lot. Because by adding more hidden layer, it increases the training cost and subsequent layer would cause its work on different input. Also, add more hidden layer would risk the process getting overfitting.

Add more nodes to hidden layer and keep other parameters the same:

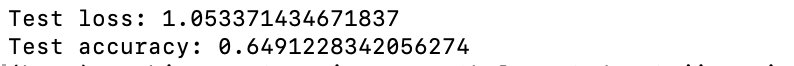
Number of Hidden Layer: 1

Number of Node in hidden layer: 500

Dropout: no dropout

Batch size: 128

Epoch: 20



Base on the output, add more nodes in hidden layer doesn’t increase the accuracy as well. This is because the number of noes control the capability of learning any mapping function, but the learning algorithm did not realize this capability [2].

Add more epochs and keep other parameters the same:

Number of Hidden Layer: 1

Number of Node in hidden layer: 100

Dropout: no dropout

Batch size: 128

Epoch: 100

图片包含 物体

描述已自动生成

图片包含 文字

描述已自动生成

Base on the output, add more epochs would significantly increase accuracy, because the accuracy is increasing overall (see epoch from 1 to 20). Also, adding more epochs would decrease the variance, so that increase the accuracy to a good balance. Therefore, by adding more epochs, it could increase the accuracy to a consistent value.

Decrease the batch size and keep other parameters the same:

Number of Hidden Layer: 1

Number of Node in hidden layer: 100

Dropout: no dropout

Batch size: 20

Epoch: 20



Base on the output, decrease the batch size also increase the accuracy. Because with small number of batch size, it goes through the system more quickly with less variability, which fosters faster learning [1].

Therefore, the best model is:

Number of Hidden Layer: 1

Number of Node in hidden layer: 100

Dropout: no dropout

Batch size: 20

Epoch: 100

图片包含 展示

描述已自动生成

With Lose graph:

图片包含 地图, 文字

描述已自动生成

References

1. <https://www.scaledagileframework.com/visualize-and-limit-wip-reduce-batch-sizes-and-manage-queue-lengths/>
2. <https://machinelearningmastery.com/how-to-control-neural-network-model-capacity-with-nodes-and-layers/>