Kyi Win
Prof. Jafari
Dec 4, 2015
Individual Final Report

Introduction:

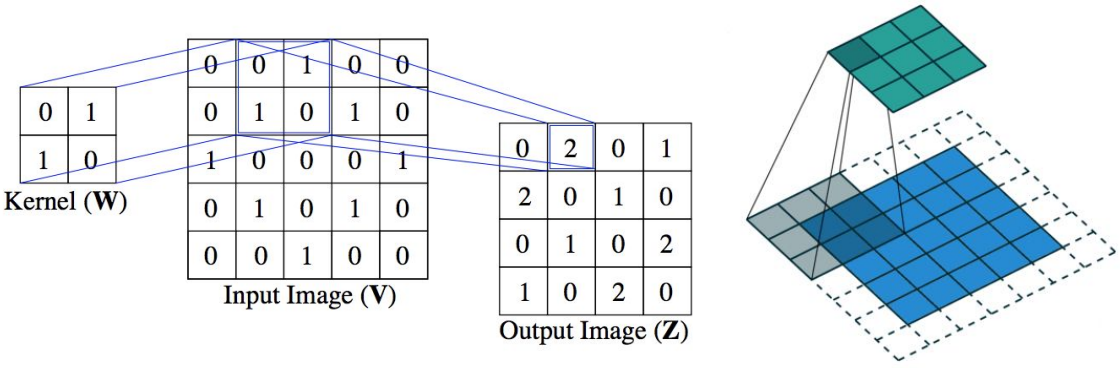
In this project, Street2shop dataset is used for image classification. The dataset includes 404,683 images in the original dataset with 11 classes. Due to the large number of images, 70500 images with 6 classes from this dataset is used in the project. Data Preprocessing is performed by splitting into train and test, convert to tensor, resize, and centercopped. Convolutional neural network is applied for this project with two different techniques, bottleneck and straight. Four different optimizers, different kernel size and different layers are used in the project to see the performance of the model, and looked for the best model.

Description of your individual work:

Convolutional neural network

This network is a multilayer feedforward neural network that has two, or three dimensional inputs. The primary layer for this network is the convolutional layer and and the weight functions of this network perform convolution operation on the image, using the convolution kernel. This is done while still preserving the relationship between pixels. To insure that our code runs correctly the following equation was used to determine feature-map size:

$$z_{i,j} = \sum_{k=1}^{r} \sum_{l=1}^{c} w_{k,l} v_{i+k-1,j+l-1}$$



In Convolution, it reduces the number of rows in the image by r-1 and the number of columns by c-1. We can pad the outside of the image with zeros before convolving to keep the image size, and the output image can be made smaller by taking larger strides or kernel movements. Normally the kernel moves one at a time, however if the stride is 2, the output image size is reduced by a factor of 2. The pooling layer often follows the convolutional layer and it reduces the spatial size of the feature map, which in other words reduces the number of parameters in the network.

Describe the portion of the work that you did on the project in detail:
In this project, I used different optimizers and trained this neural network model. I used Adam, SGD, Rprop, and Adagrad in training the neural network, and compared the results of it to see if there is a difference in accuracy rates due to the different optimizer.

Results:

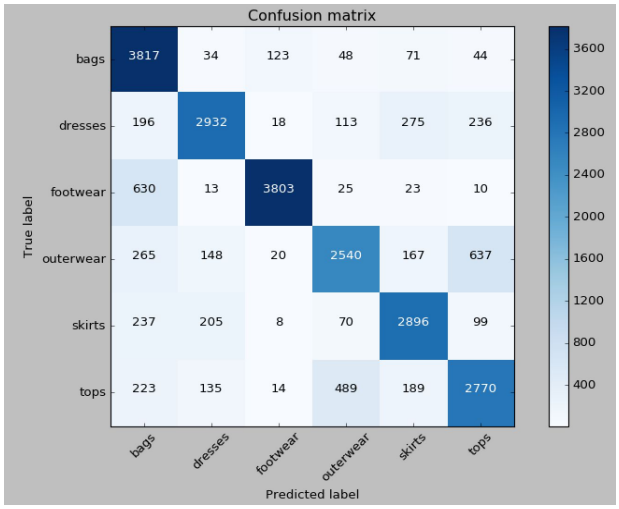
With channel of 3 and the same random sample overtime with set.seed (1122), our results are summarized as below. The following are the 5 training that I did for the project, and the results are summarized as below.

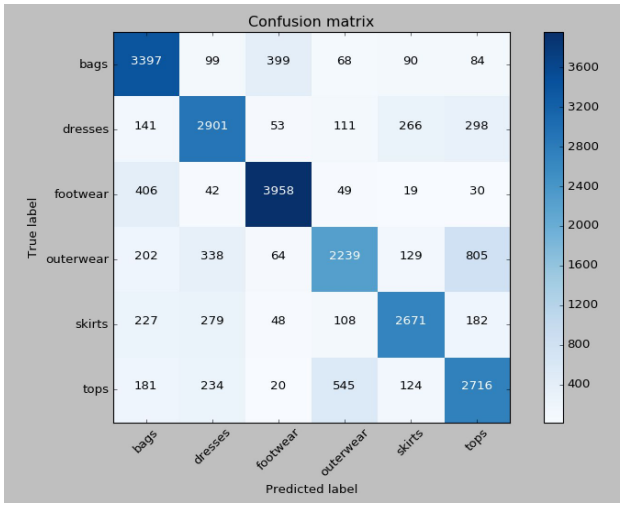
Epoch	Iteration	Batchsize	Layers	Accuracy	Time	Image Size	Optimizer
1	4	100	3	56%	104.281539 sec	224x224	Adam
5	7	64	5 (bottleneck) 4 (maxpool)	72%	717.42504 sec	224x224	Adam
5	7	64	5 (bottleneck) 4 (maxpool)	80%	473.503 sec	224x224	SGD
5	7	64	5 (bottleneck) 4 (maxpool)	79%	481.23 sec	224x224	Adagrad
5	7	64	5 (bottleneck) 4 (maxpool)	54%	1025.029 sec	224x224	Rprop

Confusion Matrix

When comparing the confusion matrix models using of SGD optimizer and Adam optimizer with straight technique, we find that SGD optimizer has accuracy scores that are higher than the accuracy scores of Adam optimizer.

(straight SGD 5 layers, and straight Adam 7 layers)



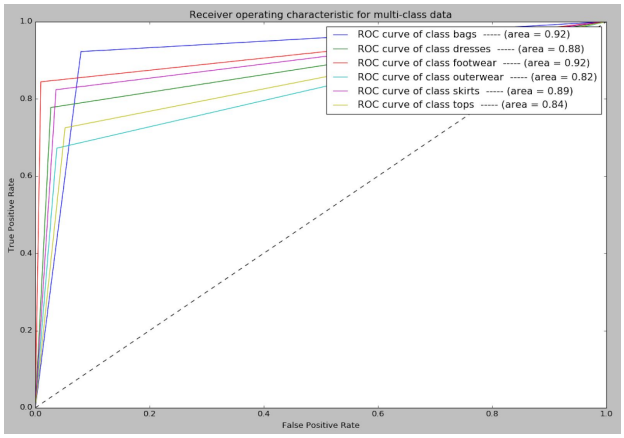


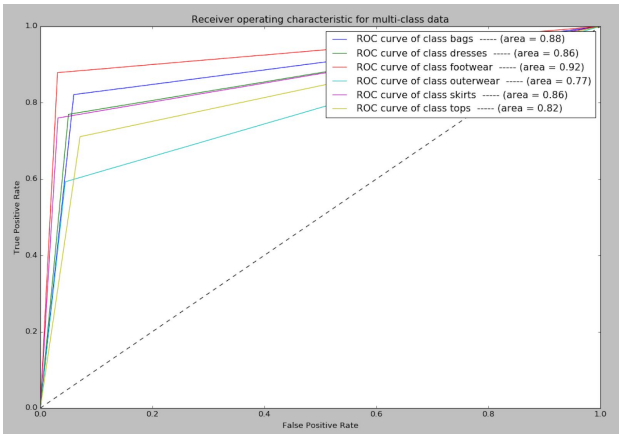
The ROC curve

Receiver Operating Characteristic (ROC) curve is defined as "the true positive rate (Sensitivity) is plotted function of the false positive rate (100-Specificity) for different cut-off points". From the two graphs below, we can see that the right graph has the curve which is closer to the upper corner and we can interpret that it has more accuracy. On the other hand, we can also interpret using the area under the curve (AUC) to determine and compare the accuracy of this bottleneck and straight approach. The areas under the graphs are specified in the box located at the upper corner for each item, and we find that the footwears has the highest accuracy in our image classification when comparing the 5 layers with bottleneck technique and straight technique.

The two graphs below shows the SGD and Adam ROC graphs and the graphs show that SGD performs better than Adam optimizer.

(left-SGD, right-Adam)





Accuray rates on each class (SGD best model)

Test Accuracy of the model on the test images: 80 %

Accuracy of bags : 85 %
Accuracy of dresses : 83 %
Accuracy of footwear : 90 %
Accuracy of outerwear : 74 %
Accuracy of skirts : 88 %
Accuracy of tops : 62 %

Classification Report (SGD best model)

	precision	recall	f1-score	support
bags	0.84	0.85	0.85	4137
dresses	0.81	0.79	0.80	3770
footwear	0.94	0.89	0.92	4504
outerwear	0.69	0.77	0.73	3777
skirts	0.75	0.86	0.80	3515
tops	0.79	0.65	0.71	3820
avg / total	0.81	0.81	0.81	23523

From the classification report of the model with SGD optimizer, we can observed that the f1 score of the footwear is the highest among all the classes. Precision represents how precise or accurate our model is - it says out of those predicted positive, how many of them are actually positive. In footwear, the precision is 94%. Recall calculates how many of the actual positive our model capture through labeling it as Positive. We can interpret from this results that this model

classified 94% correctly out of all the footwear images that are labeled as footwear and out of all the footwear images, it classified 89% of images with accurate labels.

Summary and conclusion:

From the results of our experiments, we found that footwear, bags and skirts performed the best in all of the experiments we conducted out of the categories we chose. In conclusion, the model performed better than random guess; however, there are ways this model could be improved going forward. One way to do this would be to simply gather more data and possessing more processing power. We could also try tuning the hyper-parameter differently. Moreover, we tried only with 4 optimizers and the model could be improved more with different optimizers and different layers. Moreover, having a better computing power when running the model would help us able to modify to see the outcome with better accuracy.

Calculate Code:

14 modify. 0.31% overall.

Reference:

https://pytorch.org/docs/stable/index.html
Book - Neural Network Design by Hagan Demuw Beale De Jesus
Book - Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville