Mini Project Report

Hong Pengfei 1002949
Gao Yunyi 1002871

Abstract

In this project, we trained a multi-label classifier using Pascal Voc 2012 image dataset. We utilized nn.BCEWithLogitsLoss for the loss function and tested over different threshold values to reach optimum precision, which is 0.9251 with corresponding t value of 0.652.

Choice of Loss Function

The pascal voc multi-labeling problem requires identification of multiple objects from one image. This is actually a multi-label classification problem. To solve this problem,we need to give probabilities of all 20 possible labels, including, cat, dag, boat, etc. Hence, we need to use a loss function which can minimize 20 separate binary classifiers.
It should be noted that because pascal voc has 20 classifications, the final output layer of our network has 20 output values. Hence we need to apply a sigmoid after the last linear layer so that the values are between [0, 1]. There should be no constraint on the sum of the values so that we can have multiple values close to 1.
To actualize this, we used nn.BCEWithLogitsLoss (binary_cross_entropy_with_logits). It will implement Sigmoid activation function internally before calculating the loss. While the alternative is nn.BCELoss (binary_cross_entropy) which requires extra application of sigmoid and hence we did not use.
So our logic goes like this: suppose our multi-hot encoding label of one image is [1, 0, 1] and the last layer of our model is [1.1, -0.1, 0.8]. We then (inside .BCEWithLogitsLoss) apply a sigmoid function to each element and get [1, 0, 0.8]. We then compare the model output [1, 0, 0.8] with the ground truth label [1, 0, 1] using binary cross-entropy, element-wise.The loss will be the errors summed up.

Evaluation using Accuracy & Precision

For precision, the formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

here we take mean average precision over all 20 classes.

The mean average precision (mAP) of a set of queries is defined by Wikipedia as such:

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

where Q is the number of queries in the set and AveP(q) is the average precision (AP) for a given query, q. Hence, we calculate precision for each class and then average by dividing using the number of classes. Comparing precision=TP/(TP+FP) and accuracy=(TP+TN)/(TP+FP+TN+FN), accuracy is not informative as in this problem context, we care more about if the detected class is correct or not, we don’t care about how good we are at predicting the label that is “not the label we want”, the True negatives. Since there are a lot of TN than TP, therefore accuracy may be misleading here.
In our case the accuracy is higher than precision: **mean accuracy overall classes: 0.9661**
mAP results for individual classes and average over all classes: (t=0.652) **mean prec: 0.9251**

aeroplane	bicycle	bottle	car	chair	dining table	horse	person	sheep	train
0.9912	0.9727	1.0	0.9355	0.7984	0.8125	0.8429	0.97	0.9697	1.0
bird	boat	bus	cat	cow	dog	Motor bike	Potted plant	sofa	Tv monitor
0.988	0.9588	0.9908	0.9799	0.8	0.9631	0.9784	0.6667	0.9091	0.9746

Model Architecture

We used resnet 34 pretrained using imagenet as our main model. We then added a linear layer at the top to predict 20 classes.

Learning Rate Schemes

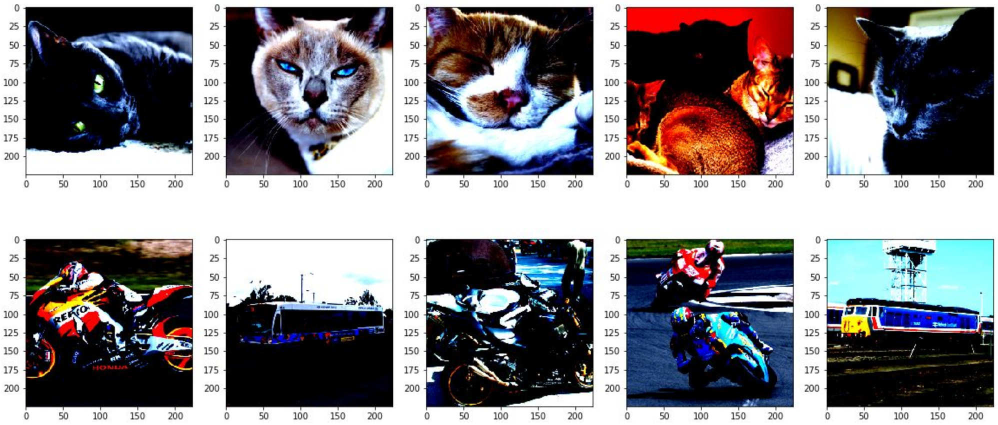
I used a learning rate of 0.01, and kept the learning rate constant during training.

Training Procedures

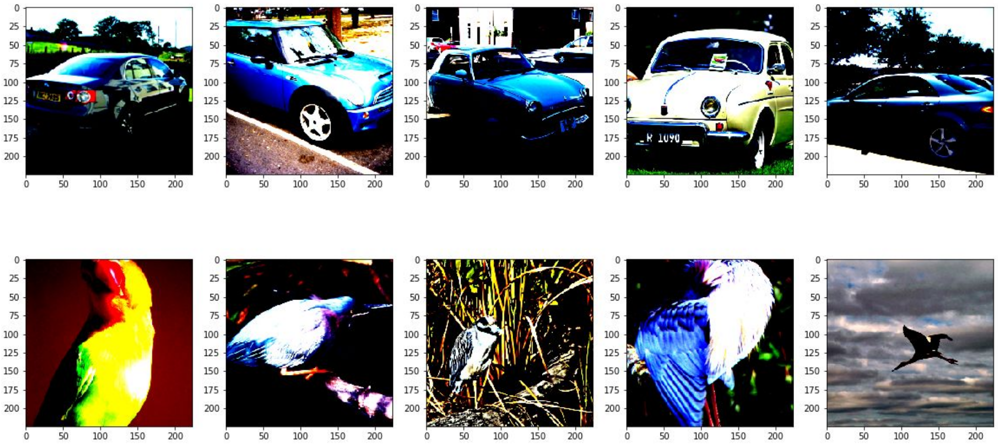
I used BATCH_SIZE = 64 which is able to fit in the GPU memory. And I trained with Stochastic gradient descent with a learning rate of 0.01 with a momentum of 0.9 during training. We trained about 3 epochs because after 3 epochs we noticed that the training loss became much smaller than the validation loss which is an indication of overfitting.

Top 5 Highest/Lowest Scored Images of 5 Classes

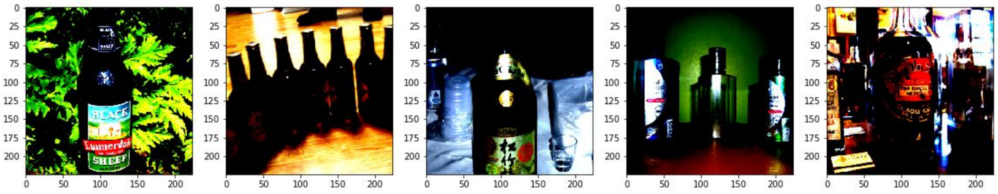
Cat Class (1st row :Top 5; 2nd row: Bottom 5)

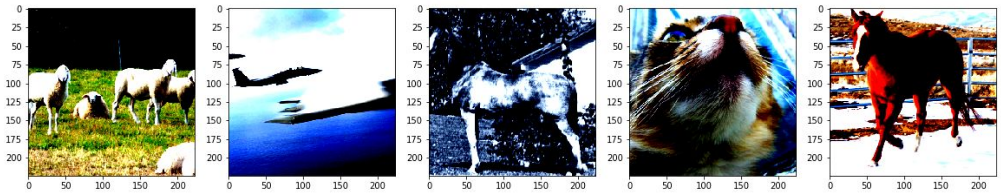


Car Class (1st row :Top 5; 2nd row: Bottom 5)

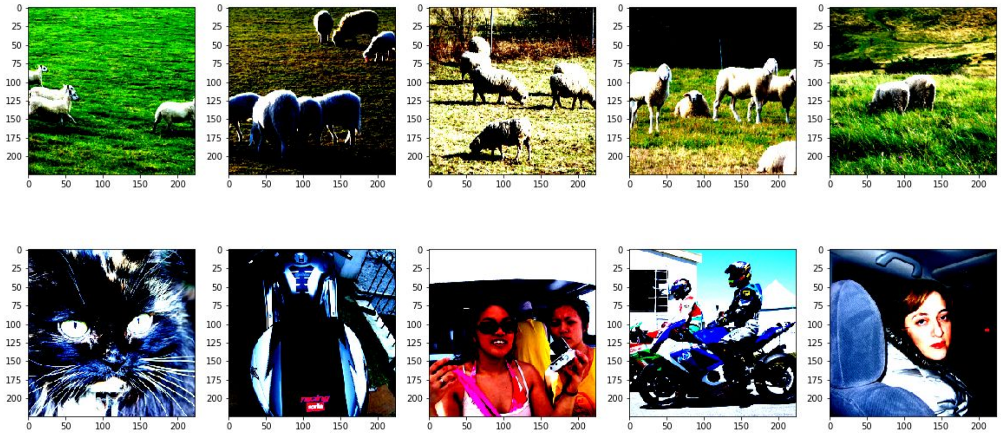


Bottle Class (1st row :Top 5; 2nd row: Bottom 5)

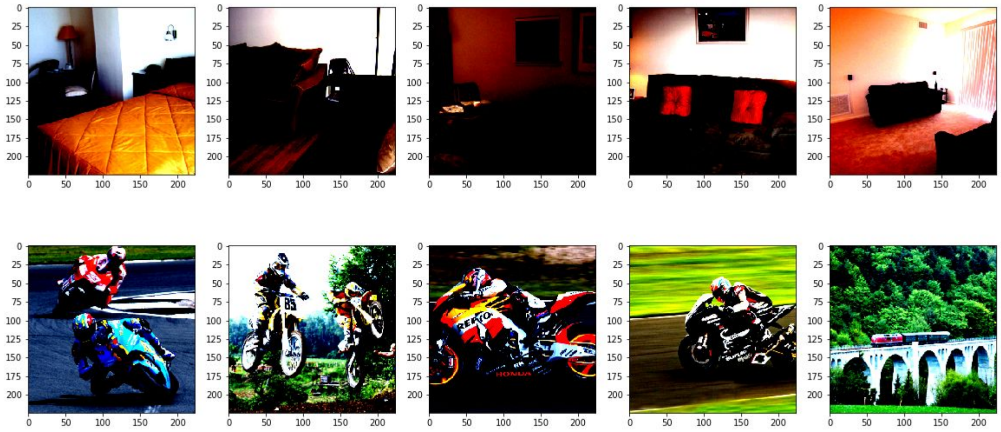




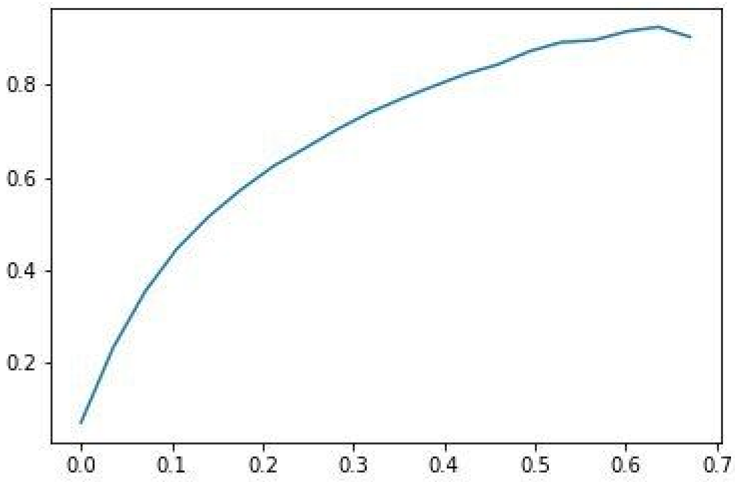
Sheep Class (1st row :Top 5; 2nd row: Bottom 5)



Sofa Class (1st row :Top 5; 2nd row: Bottom 5)



Plot of Taillac averaged over all 20 classes against t



Y-axis: mAP(mean average precision) of top-ranked samples of all 20 classes
X-axis: threshold from 0 to 0.67 (20 threshold points chosen) (0.67 is the lowest of highest f(x)s of all classes, so that prevent from having 0 as precision for some of the classes)
The trend keeps increasing from 0 to 0.65 because as higher the value of the prediction, the model is more confident of the prediction score, therefore precision is higher.