Introduction to machine learning 2020 – Project 4

Chenwei Xiao

Colab project:
https://colab.research.google.com/drive/1fZuVfErFWA19v6LaIWxRDMT7qUbzT69B?usp=sharing

This task is to use triplets to do a rank similarity learning. Given are triplets of objects {x,x+,x-} whose relative similarity obey a predefined order: x is known to be more similar to x+ than to x-. The goal is to learn a function f such that for any new triplet of objects, it obeys f(x,x+)>f(x,x-). This setup assumes a weaker form of supervision than in regression, because instead of providing an exact measure of similarity, one only has to provide the relative order of similarity. However, we met difficulties when we try to use triplet_loss to optimize our model, especially on the coding. So we use another way.

We decide to use a pre-trained CNN model to get the initial features to do this task to save the time for our training and improve accuracy. We choose ResNet50 model and load weight from imagenet to get features of each food image by using ResNet50 without its output layer. We first do the preprocess of images to convert it into tensorflow format using image_to_array and preprocess_input function and set the target size of (224,224).Then we use the cosine distance to measure the similarity of different images. This reach an accuracy of 0.61 in our training data but only has accuracy of 0.38 on test set.

So we try to improve the features by adding 2 Dense layer of 512 units and 128 units with activation function of ReLU and HE uniform kernel initializer and between each layer we dropout the data with a ratio of 0.3. Then to get a classification we use the sigmoid activation function for output layer.

We train our model in this way, each time we concatenate one triplet with 3 images as input to ResNet50 model without top to get features (so the shape of features is (1,6144)) as input to our adding Dense layer. But the sequence is shuffled and if it is x,x+,x- we sign its category as 1, we also change the sequence to x,x-,x+ and we sign its category as 0. In this way we fit our model and do the prediction for test set. Due to the final output required is actually 1 for (x,x+,x-), we set category as 0 if sigmoid function is smaller than 0.5, or we set category as 1.

We also use the earlystopping to reduce overfitting.
We use epochs of 500 to fit our model and save the model weight to best_model.h5 with ModelCheckpoint method. The model is trained by 2*59392 triplets with batch size of 512 We do the batch here to ensure that we get some training even if we don not have enough time to train the model with all the training dataset. We also calculate the time, accuracy for each epoch and plot the evolution of accuracy to epoch. In this way, we finally fit the model and save the weights. Then we load the final best model and use model to predict the similarity of triplets of images in test set. The output is saved as csv file for each triplet.