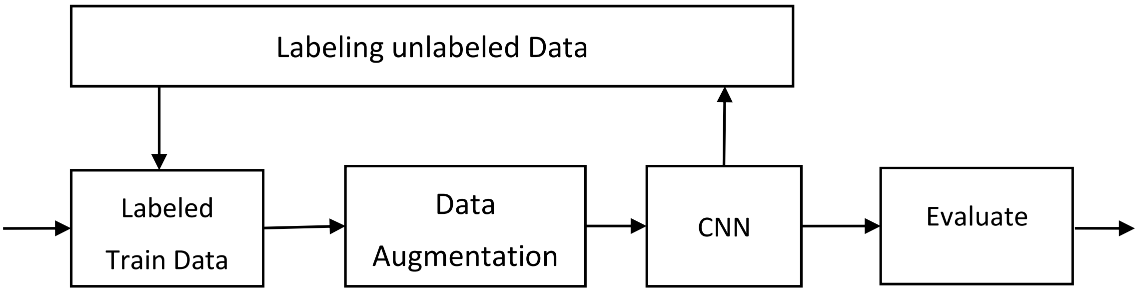
Machine Learning Homework 3 Report

Student ID: r05922063   Name: 陳啟中

CNN Method with Self Training Overview



Data Augmentation

Before feeding pictures into training models, I manually augment the data set by flipping pictures horizontally. It is by instinct that objects remain the same even if we flip the picture. Though, learning models don't learn this common sense. We need to give the hint to our training model.

Supervised Learning - CNN

Topology

Input (32x32 RGB Image)
CONV3-64 with BatchNorm
CONV3-64 with BatchNorm
MaxPool2
CONV3-128 with BatchNorm
CONV3-128 with BatchNorm
MaxPool2
CONV3-256 with BatchNorm
CONV3-256 with BatchNorm
FC-512 with BatchNorm, Dropout(0.5)
FC-512 with BatchNorm, Dropout(0.5)
FC-256 with BatchNorm
Output softmax (10)

Early stopping

To prevent overfitting, early stopping is an easy and efficient way. The key point is to find the ending point just before overfitting occurs. I firstly split the training data into 90% subset training data and 10% validation data. Upon each epoch, the program calculates the testing loss on validation data. Once the testing loss grows, we then find the stopping point.

Performance

This method reaches 65% accuracy on validation data.

Semi-supervised Learning(1) – CNN with Self-Training

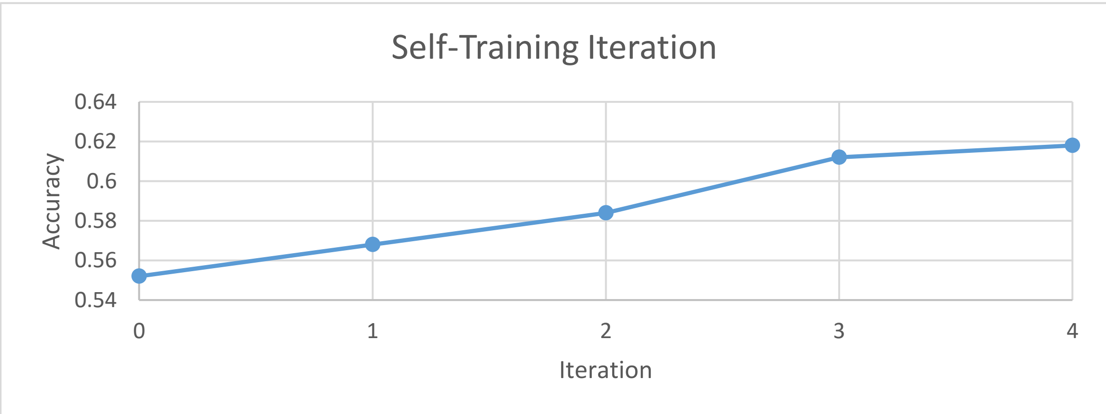
Description

I use self-training method to leverage structure information of unlabeled data. Firstly, I build an initial CNN with only labeled data. Secondly, I evaluate unlabeled data to get the prediction score on each label, and then choose those unlabeled data into labeled data set with 90% belief belonging to a particular label. Thirdly, repeat the procedure to make the model better and label more unlabeled data.

The threshold 90% is tuned by balancing time efficiency and performance. If this value is set too high, labeling unlabeled data will be too difficult to be leveraged, leading to more iteration and time consumption to reach the performance. If too low, ill-predicted data will be added in and downgrade the performance.

Self-Training Iteration Discussion

The following is the figure showing the performance gain of self-training. Iteration #0 is training on labeled data.



Performance

The final performance reaches 75% accuracy on validation data. Far better than supervised CNN.

Semi-supervised Learning(2) – AutoEncoder + DNN

Topology

Pre-train phase:

Input (32x32 RGB Image)
Encoder: FC-512, relu
Encoder: FC-256, relu
Decoder: FC-512, relu
Output (32x32 RGB Image)

Fine-train phase:

Input (32x32 RGB Image)
Encoder: FC-512, relu
Encoder: FC-256, relu
FC-256, relu
Output softmax (10)

Description

I use all unlabeled data to pre-train the stacked autoencoder (SAE) to get a better representation of input images. Then, use the encoder trained by SAE and one fully-connected layer to perform classification on labeled data.

Performance

Poorly 37%... I think the reason is that stacked autoencoder is not a good method to encode images, comparing to CNN.