Mini project plan

Reporter: Zhengyu Wu

Report Time：2018-09-01

The input/output of the task

- Input image: 32pixel x 32pixel x 3channel

  After unpickling the data from CIFAR-10, we can get a 10000x3072 matrix. Each line represents an image whose first 1024 values stand for red channel, the next 1024 values for green channel, and the final 1024 values for blue channel.

- Output: 10 classes x 1

  The output is 10 classes with their corresponding possibilities.

The general steps of image classification task

An image classification task is divided into two parts, training stage and test stage.

Training stage

Training process is often to improve the parameters of the model. The overall training process of image classification task may be summarized as below:

For example, there are four classes in total. The input image is a cat, then the target probability is 1 for cat class and 0 for other three classes.

- Input Image = Cat
- Output: Target vector [0,1,0,0] ([prob to be Dog, prob to be Cat, prob to be Car, prob to be Bird])

Step1: We initialize all filters and parameters / weights with random values.

Step2: The network takes a training image as input, goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and finds the output probabilities for each class.

Lets say the output probabilities for the cat image above are [0.2, 0.4, 0.1, 0.3]

Step3: Calculate the total error at the output layer.

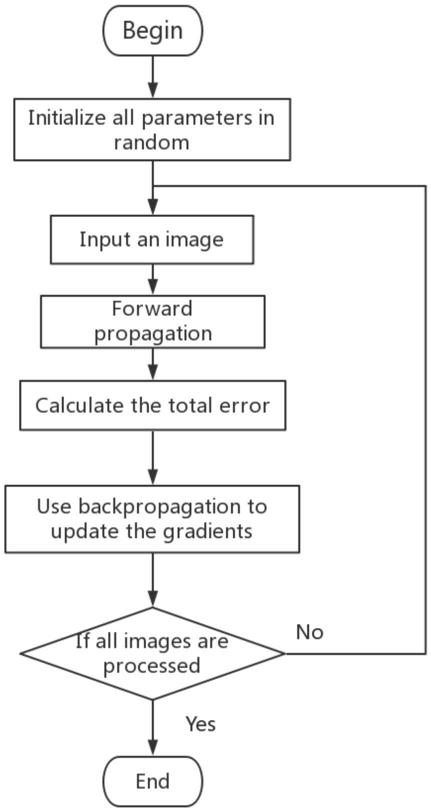
Total Error = $\Sigma(target - output)^2$

Step4: Use Backpropagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent to update all filter values / weights and parameter values to minimize the output error (see reference 21 and 22 for details).

- The weights are adjusted in proportion to their contribution to the total error.
- When the same image is input again, output probabilities might now be [0.1, 0.1, 0.7, 0.1], which is closer to the target vector [0, 0, 1, 0].
- This means that the network has learn to classify this particular image correctly by adjusting its weights / filters such that the output error is reduced.
- Parameters like number of filters, filter sizes, architecture of the network etc. have all been fixed before Step 1 and do not change during training process – only the values of the filter matrix and connection weights get updated.

Step5: Repeat steps 2-4 with all images in the training set.

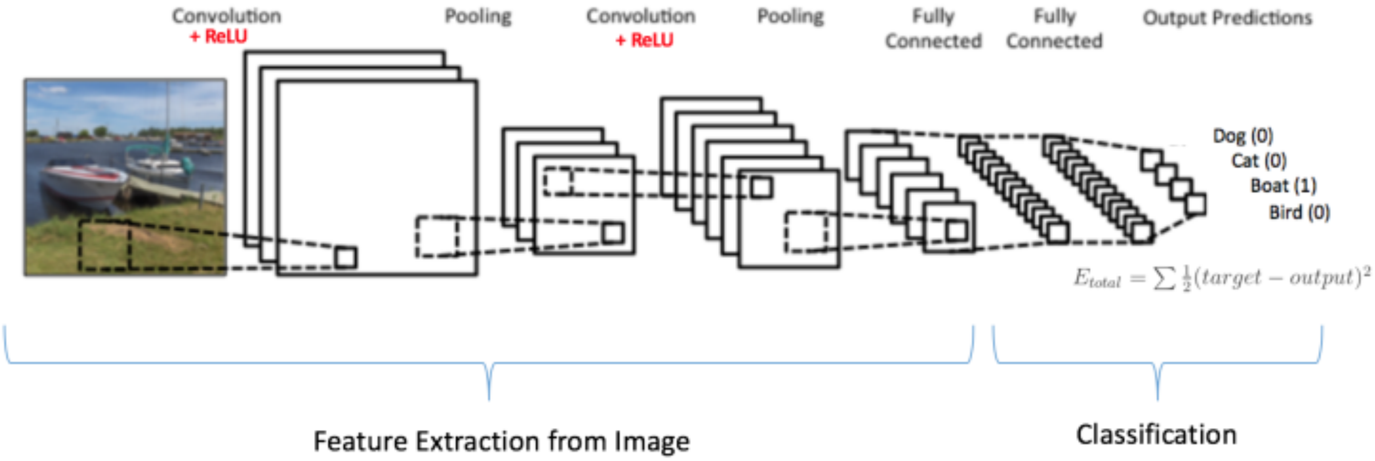
The overflow of the training process is shown as below.


Begin
Initialize all parameters in random
Input an image
Forward propagation
Calculate the total error
Use backpropagation to update the gradients
If all images are processed
No
Yes
End

Test stage

When a new image is input into the model, the model would go through the forward propagation step and output a probability for each class.

Notice that for a new image now, the output probabilities are calculated using the weights which have been optimized to correctly classify all the previous training examples.



The difficulty of image classification task

In my opinion, the difficulty comes from the training stage. Whether the network is large enough, whether the model can be trained effiently, and whether the model is not overfitting will decide whether the model works in a general image classification task.