COMP9444-Project 2: Simpsons Character Classification

Transform:

Preprocessing of picture:

1、Resize: Due to the different sizes of the images in dataset, we have to change the size of the input image. And we finally choose the common size[64,64] after checking the size of images;

2、ToTensor: change the input image to tensor by changing the image shape(H,W,C) into tensor shape(C,H,W). Convert all the parameters to [0,1].

3、Normalize: Reset the value of each color channels of RGB. Normalization helps to get the data within a range and which helps in making training a lot faster.

4、Compose: run the functions above step by step.

Loss function: CrossEntryloss

We used the loss function: CrossEntropyLoss function.

CrossEntropyLoss function is the most common function for multi-class classification. Comparing with another loss function(NLL), it add a log_softmax algorithm, softmax function converts the output into a probability value in a range of 0-1. And the value shows the probability that the image represents the character. The non-linear softmax algorithm allows the function to calculate the loss of the input that are in the middle range more accurately, which means the accuracy of the loss is higher in general class.

$$log\_softmax = log_e\left[\sigma(z)_j\right] \qquad \sigma(z)_j = \frac{e^{zj}}{\sum_{k=1}^{n} e^{zk}}$$

$$nllloss = -\frac{1}{N}\sum_{k=1}^{N} y_k \left(log\_softmax\right)$$

$$cross\_entropy = log\_softmax + nllloss$$

So that we can consider the CrossEntryloss function combines the advantages of both of the NLL loss function and the softmax algorithm.

Optimiser:

The optimiser we used is adam.

There's two kinds of optimiser that we can choose: SGD and Adam. We finally used Adam cause Adam is the second generation, it contains both first and second order momentum while SGD only consider the first order momentum. The speed of SGD optimization is lower than Adam, it would normally stay at the best point and stop optimizing. Adam is also suitable for dealing with large number of dataset. Instead of SGD, Adam is more stable and the process of parameter adjusting is more simple.

Tuning of metaparameters:

Train_val_split: the proportion of trainset and testset. And number is 1 when we trained the model for the last time;

Batch_size: the number of trainset selected for each training. The larger number, the greater the randomness.

Epochs: the times of training. The more training times, the higher the accuracy, but the longer the training time.

Overfitting avoid

the way to avoid ovefitting we used in the code is dropout, it will randomly ignore the nodes, which means the node will probably not be used during the training of the model. This way is sufficient and it can be used in different kinds of situation. The model will have a series of different networks that are constructed with same nodes, due to the random choosing. To solve the overfitting problem, we used this function to make the architecture to be more simple, and this way will also generate more input data.

Model architecture

The model used in the program is CNN model, and the network is constructed as convolution→ batchnormal→relu→pool→convolution→...→pool→linear.

Convolution: apply the feature detectors;

batchnormal: to convert all the input into the same variance and same pixel value in every channel;

Relu: thresholds all the input features to be 0 or greater;

Maxpool2d: make the features more obvious without changing the channel_num of input and output;

Linear: compute all the classes by fully connected layer.