

Assignment 4

Roll Number : 140050002, 140050008, 140050073

Model consists of Multilayer RNN, where layers can be added of '*RNN*' class defined in RNN.lua. Last step output of last layer is passed through linear layer of size 2 x (number of hidden variables in last layer) for classification. Cross Entropy Criterion is used.

Description of nontrivial variables in Model :

1. `mapTable` : Since input (system call numbers) are not in range 1 to 153, one hot encoding needs to be remembered. It is stored in `mapTable` (key, value) pair of input to encoded tensor.
2. `finalIndexInOneHotEncoding` : It turns out that test file contains inputs (system call numbers in a particular sequence) which are not seen in train data. To accomodate such cases, this variable is used. (see `testModel.lua` for details)

Description of implementation Backward Propagation :

Gradients are clamped and backward propagation is truncated to avoid vanishing gradient.

Since this is “many to one” classification, loss is calculated at last step of last layer and is propagated through last layer in obvious sense. Propagation of gradient from one layer to another layer is done **only at last step**, and then gradient is back propagated within the layer.

It is expected to call `forward()` before `backward()` using same input because hidden variables and outputs are stored during `forward()` and are used during `backward()`. This design is incorporated to avoid calculation of forward propagation in `backward()`.

BestModel :

Saved best model consists of one hidden layer which has 64 hidden variables. It gives accuracy of **0.73096** on public leaderboard on kaggle with team name '140050002'.

Hyper parameters of the training are as following. Learning rate : 0.001, clamp size : 2, number of epoch : 130, truncate length : 20