MNIST FASHION DATA IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK VGG16.

A BRIEF REPORT ON DIFFERENT OPTIMIZATION ALGORITHM USED AND HYPERPARAMETER TUNING.

We have performed the task of image classification of MNIST Fashion dataset using Visual Geometry group VGG16 like convolutional neural network and implemented it using keras. Different optimizers have been used to train the model ADAM, RMSprop, AdaGrad, SGD, momentum SGD and NAG to minimize the error rate of the learning function. EarlyStopping function has been implemented using keras to set up the callbacks and other hyperparameter tuning was performed (learning rate and dropout rate) using a standard batch size of 16 and it was run for a standard 200 epochs.

RMSprop optimizer was used to train the model and hyperparameter learning rate was tuned it uses plain momentum giving loss and accuracy of 0.6023 and 0.796 respectively of Train and 0.6139 loss and 0.798 accuracy of validation data .

Followed by Adagrad optimizer was used to train the model and hyperparameter leraning ratewhich is parameter specific was tuned and the losss and accuracy was found to be 0.6023 and 0.795 of Train and 0.6139 and 0.7897 for validation data

Subsequently, SGD optimizer was used to train the model in minibatch and hyperparameter learning rate was tuned and the loss and accuracy was found to be 0.6022 and 0.7977 respectively and 0.6137 and 0.7899 for validataion data.

Momentum SGD was used to train the model and hyperparameter learning rate was tuned and momentum was set at 0.9, the loss and accuracy was found to be 0.785 and 0.7362 respectively and 0.7962 and 0.7324 for training set.

Nestor accelerated gradient optimizer was then used with momentum 0.9 and hyperparameter leraning rate was tuned, the loss and accuracy was found to be 0.7310 and 0.753 for Train and 0.7401 and 0,7529 for validation data.

ADAM optimier was used which is a stochastic gradient descent method and it takes account on first order second order moments, hyperparameter learning rate was tuned and the loss and accuracy was found to be 0.6024 and 0.7975 for Train and 0.6140 and 0.7092 for validation data.