

Assignment 1 : Devnagri character recognition

This assignment is to be done on Tensorflow via its Python interface. The links [here](#) might be helpful in building the background.

The dataset contains labeled images of handwritten devnagri characters. The images are in png format. Total 10,000 images are there in train set. Your task is to predict correct label for test images that we will provide.

The images can be read using the python API - scikit-image. The following example code is there to read png image into numpy-ndarray (matrix)

```
from skimage import io
image = io.imread(<filename.png>)
```

Now, image will contain the numpy ndarray (320 * 320 in our case).

Your goal is to design a feed-forward neural network for the task of Devnagri character recognition. Your report should include the following:

1. The accuracy on the validation dataset for increasing number of levels
2. Accuracy with increasing width for a fixed number of levels
3. Effect of changing the hidden unit type from RelU to Sigmoid to Tanh
4. Accuracy with different learning rates
5. Effect of different kinds of regularizers: L2 regularizer vs drop-out.

Download information for the dataset

The training dataset can be downloaded from this link -

<https://drive.google.com/file/d/oBzIqj5JgNb5RRlo1aUwyTDNEdzg/view>

The validation dataset (to test your model) can be downloaded from this link -

<https://drive.google.com/open?id=oBzIqj5JgNb5RaoduNW95UDNHNoU>

Submission format and instruction

The report should contain a link to folder containing all the source codes used for both training and pre-processing (if any). Also, the folder should contain the trained model in a binary file, which can be loaded in Tensor-flow later on. The model when given a 320 x 320 pixel image should output the probabilities of all the classes. The output should only contain probabilities serially separated by space and nothing else. Your model will be evaluated on test data only.

Note: If you pre-process the input before forward-propagation, provide a script which takes an a 320 x 320 image from the location provided as an argument and provide the above mentioned output. The script should do all the pre-processing.