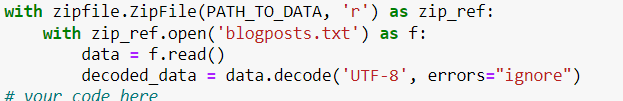
Python

zipFile.ZipFile : to open the file

zipFile.read : read the file , it gives the binary,

* before using we need to decode them into utf-8



# Numpy

Argsort:

a = np.array([13, 11, 14,1 2])

np.argsort(a) : it will return sorted array indices

np.transpose : convert row into column

np.repeat?

np.arrage?

np.argmax?

np.linalg.norm?

print((a - a < 0.001).all()) : return true if all the diff are less then value.

# Torch

Torch.mm vs Torch.mutual?

---------------------

Torch.reshape vs torch.view both work same.

Torch.from\_numpy(arr)

Torch.pow(y-o, 2)/2

t = torch.tensor([[1,2,3],[4,5,6]], dtype=torch.long)

--------------------------------

t.unsqueeze(0) : adding extra dimension at position o

t.squeeze(0) : removing extra dimension at position o

---------------------------------

# Transpose swaps two axes

print(t.transpose(0,1))

# Permute allows more general permutations of axes

print(t.permute(1,0))

both work approximately same.

-----------------------------\

W = init(num\_classes, num\_features)

Num\_classes is no of unique target

Num \_feature is no number of column for train data.

--------------------------------------\

print(np.log(np.exp(M))) : this will implement logic on the all of the individuals values of the matrix

Glove emdedding:

if you have 10,000 unique words in your vocabulary and you set EMBEDDING\_SIZE to 300, then the shape of your embedding matrix would be 10,000 x 300.