CSC7343 HW2

Due Nov 12th by the end of the day. Do all the work (code, plot, analysis text) on a jupyter notebook. On the notebook, you should clearly mark the code segment that implements the model. Upload the notebook with all the results/text in moodle on or before the due time. [All implementation should use PyTorch.]

(Also upload the weights of the trained model to your google drive. Define a function "load_model_x()" on your notebook. Calling this function should download the trained weights into the current directory and then load them into your model. **For each task *x*, you should have one "load_model_*x*()"**.

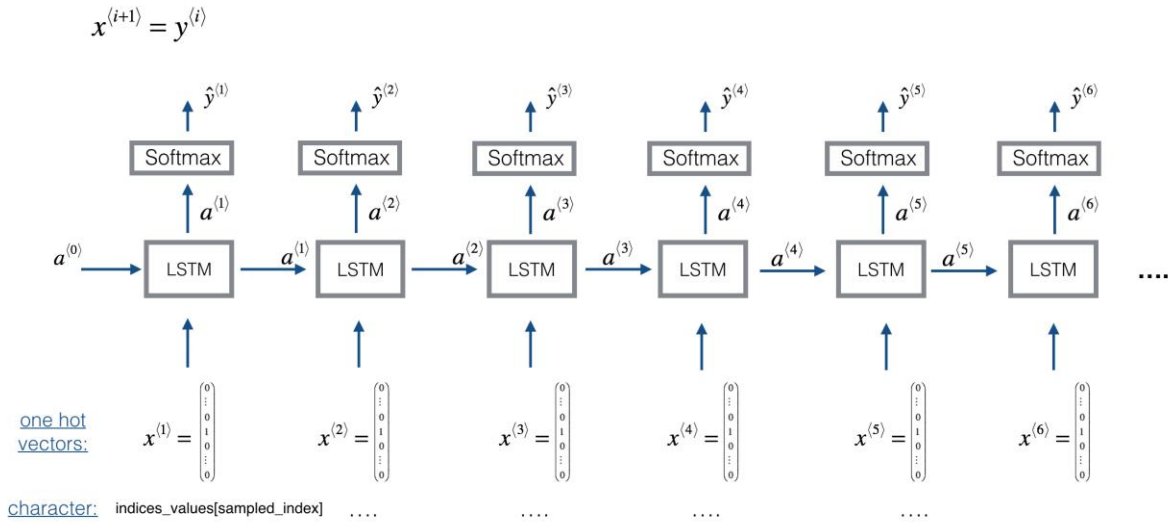
Your tasks in this homework are to build a character level LSTM network: 1) to generate (English) names; 2) to determine whether a string of characters is a (English) name.

Download training data:

https://catalog.data.gov/dataset/baby-names-from-social-security-card-applications-national-level-data

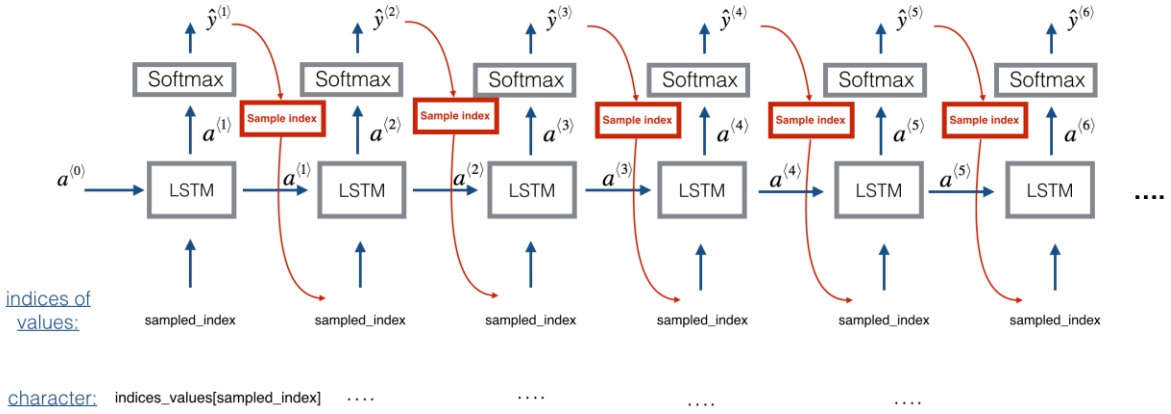
Use the names in the file for the year 2018 as your training data. (You may further reduce the number of names by randomly sampling 5000 names to use as training data.)

Task 1: Implement a LSTM network with the following structure. (The LSTM layer has 128 cells).



Note that a name (string of characters) is used both as input and output in training. (You should add a special character to each name string, which serves as a mark to indicate the end of the string.) Use an all zero vector for $X^{<0>}$ and the initial state of the LSTM should be all zeros too. Train the model with cross-entropy loss from the softmax outputs at all (valid) steps.

1) Describe in the notebook how you train the model using batches of names. (Note that different names may have different length.)
2) After training the model, you can sample (generate names) using the model. Implement the following process to generate a name using the trained model. You should have a function "generate_name()" defined on your notebook. Once the trained parameters are loaded into your model, each call to the function should generate a name (string).



Task 2: You can also use the LSTM model as a string classifier by removing the softmax layer and feeding the LSTM output from the last character of the name string to a sigmoid neuron, which then gives the probability whether the string is a name. Construct a dataset where the positive samples are the names you used in Task 1 and make same number of random strings as negative samples. Train the classifier model on the dataset using binary cross entropy.

1) Define a function "is_real_name(S)" on the notebook, where S is a string. Once the trained parameters are loaded into your model, this function can be called to determine whether a name S is real.