# IS 784 Deep Learning for Text Analysis

## Assignment 2 (Deadline 18 May 2021)

In this assignment, you are going to create a word-embedding network based on children's essays about a book that they have read.

For the solution, you are required to implement a skip-gram model with a context window size of one using logistic regression.

### 1. Data Preparation(50pts)

The dataset given with this document includes The Children's Books Test(CBT) dataset, which was created by the Facebook team. Select a book from the BOOK_SPLIT file among 108 different children's books or if you are feeling adventurous (or your selected book has fewer pages) you can select more than one book. Note which book(s) you have selected.

After selecting the book, retrieve it from the dataset. Then filter out the irrelevant sections or tokens from the data such as chapter titles. You are required to prepare the data for the model using subsampling and negative sampling.

You are expected to show and explain your steps clearly and save your processed data as a CSV file. Your data should contain a center word- context word pairs and their class (positive or negative samples).

**Note:** Negative sampling can take a long time especially with increasing data sizes and with non-optimal algorithms.[1]

### 2. Preprocessing and Data Loader(20pts)

Now that your dataset is ready, we need to preprocess the data and create a data loader. You are expected to use the csv file you have previously created. Your codes from now on should run independently from the first part.

Here you have few options: (1) You can either give your CSV file to Torchtext (check legacy version documentation),(2) you can process it with external libraries and finally use Pytorch's DataLoader (check Torhtext's migration tutorial) (3) you can create your own data loader.

Don't forget to split your data into train-validation-test sets. This will become handy later on.

Because we are creating a language model, **do not use any pre-trained word vectors.**

[1] While using semi optimized algorithm and Google Colab: 1 book with ~20k tokens it took around 44 seconds and while using all of the cbt validation set -5 books- ~300k tokens it took around 42 min)

## 3. Skip-gram with Logistic Regression(30 pts)

While creating the network consider;

- Logistic regression model.[2]
- How you will create word embeddings.
- How are you are going to handle data coming from the data loader?

Train your network using the data loader created on the second part and show validation and test results using necessary metrics such as loss, accuracy, confusion matrix, etc. and comment on the results.

State which optimizers and loss functions you considered for this task. Describe how we can improve this network further briefly.

Note: You can save the trained network parameters to do the bonus part later on.

## 4. Bonus:Word Vectors(20pts)

How can we obtain embeddings from the network we trained in the previous part? From the trained network obtain word embeddings. Using these embeddings select few words from your dataset and show them on a graph.

Then, using similarity metrics, show word similarities between some example words you selected. (for example similarities between King and Queen ..) (if you are feeling adventurous you can also do the equations like "King-male+female = Queen). Assess whether your results are satisfactory. Give your reasons.

## 5. Deliverables and Late submission

You are expected to submit one '*.ipynb' file and one CSV file that contains the edited dataset. Your file names should contain your student number. Add your name and student number inside the ipynb file as well. Your code blocks should be adequately explained. Add explanations by creating text blocks.

Late submission is accepted until 23 May 2021 with a -5 points penalty for each day.

[2] You can check example logistic regression implementation in pytorch using MNIST dataset at https://blog.goodaudience.com/awesome-introduction-to-logistic-regression-in-pytorch-d13883ceaa90