Download zip from <https://github.com/scpd-proed/XCS224N-A4>

Unzip content into

C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master

* Open Anaconda Prompt

(base) C:\Users\ADMIN>python --version

Python 3.7.3

* navigate to your project

(base) C:\Users\ADMIN>cd C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master

# Environment Setup Start

* Create a virtual environment using conda

(base) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>conda env create --file local\_env.yml

|  |
| --- |
| Collecting package metadata (repodata.json): done  Solving environment: done  Downloading and Extracting Packages  torchvision-0.5.0 | 6.4 MB | #################################### | 100%  wheel-0.34.1 | 65 KB | #################################### | 100%  six-1.14.0 | 27 KB | #################################### | 100%  docopt-0.6.2 | 23 KB | #################################### | 100%  scipy-1.3.2 | 11.2 MB | #################################### | 100%  numpy-base-1.18.1 | 3.8 MB | #################################### | 100%  pip-20.0.2 | 1.7 MB | #################################### | 100%  tqdm-4.42.0 | 55 KB | #################################### | 100%  pytorch-1.4.0 | 472.8 MB | #################################### | 100%  nltk-3.4.5 | 1.7 MB | #################################### | 100%  cudatoolkit-10.1.243 | 300.3 MB | #################################### | 100%  numpy-1.18.1 | 6 KB | #################################### | 100%  setuptools-45.1.0 | 536 KB | #################################### | 100%  Preparing transaction: done  Verifying transaction: done  Executing transaction: done  #  # To activate this environment, use  #  # $ conda activate local\_nmt  #  # To deactivate an active environment, use  #  # $ conda deactivate |

Note: The conda env is located in C:\ProgramData\Anaconda3\envs\local\_nmt.

* Activate the virtual environment using conda

(base) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>conda activate local\_nmt

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master> |

* Check all the python packages in this virtual environment

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>conda list  # packages in environment at C:\ProgramData\Anaconda3\envs\local\_nmt:  #  # Name Version Build Channel  blas 1.0 mkl  certifi 2019.11.28 py36\_0  cffi 1.13.2 py36h7a1dbc1\_0  cudatoolkit 10.1.243 h74a9793\_0  docopt 0.6.2 py36\_0  freetype 2.9.1 ha9979f8\_1  icc\_rt 2019.0.0 h0cc432a\_1  intel-openmp 2019.4 245  jpeg 9b hb83a4c4\_2  libpng 1.6.37 h2a8f88b\_0  libtiff 4.1.0 h56a325e\_0  mkl 2019.4 245  mkl-service 2.3.0 py36hb782905\_0  mkl\_fft 1.0.15 py36h14836fe\_0  mkl\_random 1.1.0 py36h675688f\_0  ninja 1.9.0 py36h74a9793\_0  nltk 3.4.5 py36\_0  numpy 1.18.1 py36h93ca92e\_0  numpy-base 1.18.1 py36hc3f5095\_1  olefile 0.46 py36\_0  pillow 7.0.0 py36hcc1f983\_0  pip 20.0.2 py36\_1  pycparser 2.19 py36\_0  python 3.6.10 h9f7ef89\_0  pytorch 1.4.0 py3.6\_cuda101\_cudnn7\_0 pytorch  scipy 1.3.2 py36h29ff71c\_0  setuptools 45.1.0 py36\_0  six 1.14.0 py36\_0  sqlite 3.30.1 he774522\_0  tk 8.6.8 hfa6e2cd\_0  torchvision 0.5.0 py36\_cu101 pytorch  tqdm 4.42.0 py\_0  vc 14.1 h0510ff6\_4  vs2015\_runtime 14.16.27012 hf0eaf9b\_1  wheel 0.34.1 py36\_0  wincertstore 0.2 py36h7fe50ca\_0  xz 5.2.4 h2fa13f4\_4  zlib 1.2.11 h62dcd97\_3  zstd 1.3.7 h508b16e\_0 |

# Environment Setup End

# Coding/Testing Start

1. (2 points) (coding) In order to apply tensor operations, we must ensure that the sentences in a

given batch are of the same length. Thus, we must identify the longest sentence in a batch and

pad others to be the same length. Implement the pad\_sents function in utils.py, which shall

produce these padded sentences.

1. (3 points) (coding) Implement the \_\_init\_\_ function in model\_embeddings.py to initialize the necessary source and target embeddings.
2. (4 points) (coding) Implement the \_\_init\_\_ function in nmt\_model.py to initialize the necessary model embeddings (using the ModelEmbeddings class from model\_embeddings.py) and layers (LSTM, projection, and dropout) for the NMT system.
3. (8 points) (coding) Implement the encode function in nmt\_model.py. This function converts the padded source sentences into the tensor X, generates , …, , and computes the initial state and initial cell for the Decoder. You can run a non-comprehensive sanity check by executing:

python sanity\_check.py 1d

|  |
| --- |
| (base) C:\Users\ADMIN>cd C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master  (base) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>conda activate local\_nmt  (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python sanity\_check.py 1d  Running Sanity Check for Question 1d: Encode  --------------------------------------------------------------------------------  enc\_hiddens Sanity Checks Passed!  dec\_init\_state[0] Sanity Checks Passed!  dec\_init\_state[1] Sanity Checks Passed!  --------------------------------------------------------------------------------  All Sanity Checks Passed for Question 1d: Encode!  -------------------------------------------------------------------------------- |

1. (8 points) (coding) Implement the decode function in nmt\_model.py. This function constructs and runs the step function over every timestep for the input. You can run a non-comprehensive sanity check by executing:

python sanity\_check.py 1e

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python sanity\_check.py 1e  --------------------------------------------------------------------------------  Running Sanity Check for Question 1e: Decode  --------------------------------------------------------------------------------  combined\_outputs Sanity Checks Passed!  --------------------------------------------------------------------------------  All Sanity Checks Passed for Question 1e: Decode!  -------------------------------------------------------------------------------- |

1. (10 points) (coding) Implement the step function in nmt\_model.py. This function applies the Decoder's LSTM cell for a single timestep, computing the encoding of the target word , the attention scores , attention distribution , the attention output , and finally the combined output . You can run a non-comprehensive sanity check by executing:

python sanity\_check.py 1f

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python sanity\_check.py 1f  --------------------------------------------------------------------------------  Running Sanity Check for Question 1f: Step  --------------------------------------------------------------------------------  dec\_state[0] Sanity Checks Passed!  dec\_state[1] Sanity Checks Passed!  combined\_output Sanity Checks Passed!  e\_t Sanity Checks Passed!  --------------------------------------------------------------------------------  All Sanity Checks Passed for Question 1f: Step!  -------------------------------------------------------------------------------- |

* to generate the necessary vocab file

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python vocab.py --train-src=./en\_es\_data/train.es --train-tgt=./en\_es\_data/train.en vocab.json  read in source sentences: ./en\_es\_data/train.es  read in target sentences: ./en\_es\_data/train.en  initialize source vocabulary ..  number of word types: 172418, number of word types w/ frequency >= 2: 80623  initialize target vocabulary ..  number of word types: 128873, number of word types w/ frequency >= 2: 64215  generated vocabulary, source 50004 words, target 50002 words  vocabulary saved to vocab.json |

* train the model on your local machine in CPU:

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python run.py train --train-src=./en\_es\_data/train.es --train-tgt=./en\_es\_data/train.en --dev-src=./en\_es\_data/dev.es --dev-tgt=./en\_es\_data/dev.en --vocab=vocab.json  Let it run till iter 50 and did not crash |

* train the model on your local machine in GPU:

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python run.py train --train-src=./en\_es\_data/train.es --train-tgt=./en\_es\_data/train.en --dev-src=./en\_es\_data/dev.es --dev-tgt=./en\_es\_data/dev.en --vocab=vocab.json --cuda  ......  epoch 13, iter 83920, avg. loss 28.83, avg. ppl 5.08 cum. examples 61440, speed 4971.98 words/sec, time elapsed 9668.26 sec  epoch 13, iter 83930, avg. loss 29.53, avg. ppl 5.37 cum. examples 61760, speed 4920.32 words/sec, time elapsed 9669.40 sec  epoch 13, iter 83940, avg. loss 29.40, avg. ppl 5.45 cum. examples 62080, speed 4659.11 words/sec, time elapsed 9670.59 sec  epoch 13, iter 83950, avg. loss 28.28, avg. ppl 5.25 cum. examples 62400, speed 4972.68 words/sec, time elapsed 9671.69 sec  epoch 13, iter 83960, avg. loss 31.31, avg. ppl 5.50 cum. examples 62720, speed 5090.12 words/sec, time elapsed 9672.84 sec  epoch 13, iter 83970, avg. loss 28.42, avg. ppl 5.46 cum. examples 63040, speed 4885.95 words/sec, time elapsed 9673.94 sec  epoch 13, iter 83980, avg. loss 31.39, avg. ppl 5.67 cum. examples 63360, speed 5103.18 words/sec, time elapsed 9675.07 sec  epoch 13, iter 83990, avg. loss 29.85, avg. ppl 5.48 cum. examples 63680, speed 4949.73 words/sec, time elapsed 9676.21 sec  epoch 13, iter 84000, avg. loss 28.90, avg. ppl 5.09 cum. examples 64000, speed 5058.72 words/sec, time elapsed 9677.33 sec  epoch 13, iter 84000, cum. loss 29.63, cum. ppl 5.34 cum. examples 64000  begin validation ...  validation: iter 84000, dev. ppl 17.395953  hit patience 5  hit #5 trial  early stop! |

It took about 2 hours and 45 min.

1. (3 points) Once your model is done training (this should take about 4 hours on the VM), execute the following command to test the model:

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python run.py decode model.bin ./en\_es\_data/test.es ./en\_es\_data/test.en outputs/test\_outputs.txt --cuda  load test source sentences from [./en\_es\_data/test.es]  load test target sentences from [./en\_es\_data/test.en]  load model from model.bin  Decoding: 100%|████████████████████████████| 6999/6999 [06:21<00:00, 18.33it/s]  Corpus BLEU: 21.949783912564655  (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master> |

* Run the script evaluation\_output.py to generate the file gradescope\_test\_outputs.txt.

|  |
| --- |
| (local\_nmt) C:\Users\ADMIN\Documents\Stanford\NLP\_DeepLearning\Assignment4\XCS224N-A4-master>python evaluation\_output.py  Decoding: 100%|████████████████████████████| 1065/1065 [00:59<00:00, 17.81it/s] |

Autograde test Corpus BLEU: 28.492845489521283

* Zip all .py files and include all files under ./en\_es\_data ./sanity\_check\_en\_es\_data ./outputs folders to produce assignment4.zip