CS671 : Assignment-3
Abhishek Verma
Roll No. - 14026

Algorithm:

- Convert <.conll> data to trainable form(transitions).
- Extract features from each transition.
- Feed the above features to a self-designed neural network or sklearn.SGDClassifier.
- Get the predicted transitions and calculate the accuracy.

Steps of the above mentioned algorithm are discussed in detail below.

Conversion of data from <.conll> format to trainable form:

- Read the file <train.conll> single line and time and converted the dataset to a corpus of the form <list of list>.
- Iterated throught the whole corpus and processed one datapoint at a time.
- For every datapoint: started with sigma(stack) containing only <ROOT> and beta(stack) containing all the ords of the sentence. Then used the following rules to decide the transition:

$$o(c=(\sigma,\beta,E)) = \begin{cases} la(r) & \text{if } (\beta[0], r, \sigma[0]) \in DG \\ ra(r) & \text{if } (\sigma[0], r, \beta[0]) \in DG \text{ and } (\forall w_k, r' \text{ if } (\beta[0], r', w_k) \in DG \\ & \quad \text{then } (\beta[0], r', w_k) \in E) \\ sh & \text{Else} \end{cases}$$

$ra(r)$ says $(\sigma[0], r, \beta[0])$ added only if all outgoing arcs from $\beta[0]$ are already in E - because $\beta[0]$ will exit from σ and β and cannot be part of any more edges.

Source: CS671A slides

- Once a transition was decided, it was executed and the resulting sigma and beta was sent to the next step.

Feature extraction:

- Following features were extracted:

$\sigma[0]$ to $\sigma[2]$, $\beta[0]$ to $\beta[2]$ - 6; the first two leftmost/ rightmost children of $\sigma[0], \sigma[1]$ - 8; leftmost of leftmost and rightmost of rightmost of $\sigma[0], \sigma[1]$ - 4. This gives a total of 18 words, so 18 PoS tags and relation labels corresponding to the 12(=8+4) elements in the DG.

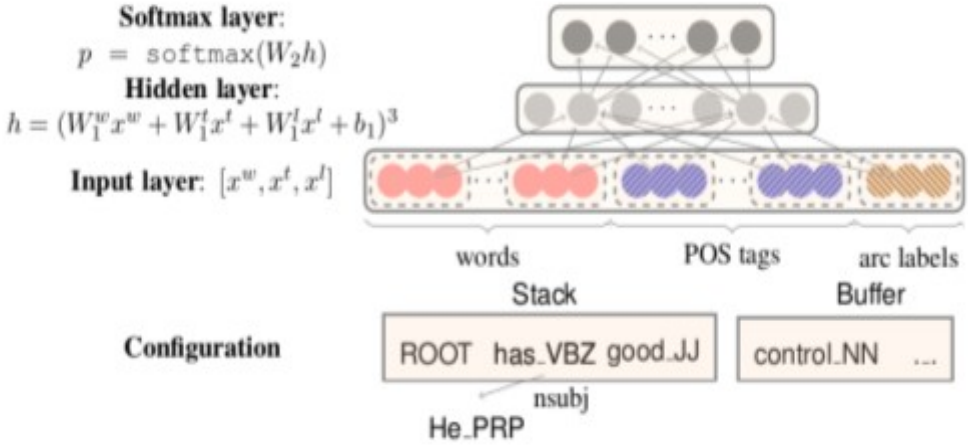
Source: CS671A slides

- So, for every datapoint, our feature matrices were of shape:
  - word - #TRANSITIONS x 18 x VOCAB_SIZE_OF_WORDS
  - PoS - #TRANSITIONS x 18 x #DISTINCT_PoS
  - Relation - #TRANSITIONS x 18 x #DISTINCT_RELATIONS

- Now, for every transition: all of the above mentioned feature matrices were concatenated and fed as input to neural_netwwork/sklearn.SGDClassifier

Neural Network:

- Following is the architecture of the neural network used:



Source: CS671A slides

- It was implemented using <pytorch>.
- More information about weight tensors:
  - W1_w: Initialed with vectors from pre-trained Word2vec.
  - W1_t: Initialized with random values between -0.01 and 0.01.
  - W1_l: Initialized with random values between -0.01 to 0.01
  - W2: Initialized with random values

sklearn.linear_model.SGDClassifier:

- This is the other model that was used. It was initialized with all the default params.

Results:

The model was trained over 1000 datapoints and tested over 100 datapoints, both belonging to the train set. Number of training classes were 92.

- Average test accuracy with neural network: 80.53%
- Average test accuracy with sklearn.linear_model.SGDClassifier: 90.16%

Files attached:

- **<preprocess_new_batch.py>:** Preprocessing with feature extraction + sklearn.linear_model.SGDClassifier + test(92 training classes)
- **<preprocess.py>:** Preprocessing with feature extraction + test(3 training classes)
- **<preprocess_new.py>:** Preprocessing with feature extraction + test(92 training classes)
- **<model.py>:** neural netork used by <preprocess.py> and <preprocess_new.py>