Walking through the projects

News article classification using Linear model, Bidirectional LSTM

Smita Bhattacharya
M.Sc. in Data Science & Artificial Intelligence
Saarland University

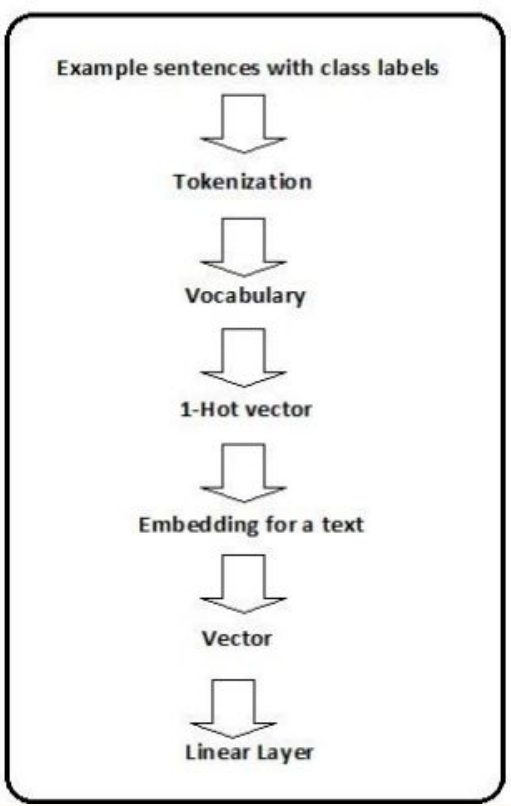
Project 1:

News article classification using Linear model, Bidirectional LSTM

News Article Classification

- **Goal:** News article classification into Topics using a **Linear Model**
- **Dataset:** AG_NEWS - an inbuilt torchtext Text Classification Dataset
- **Labels:** World, Sports, Business, Sci/Tech
- **Model Architecture:**
  - The model consists of- an embedding layer, a fully connected linear layer.
  - The n-grams of the input are obtained and appended at the end of the sentence.
  - This is then passed through an embedding layer to obtain the word embedding.
  - This is then passed to a linear layer to obtain the output

Model Architecture:



News Article Classification (Cont..)

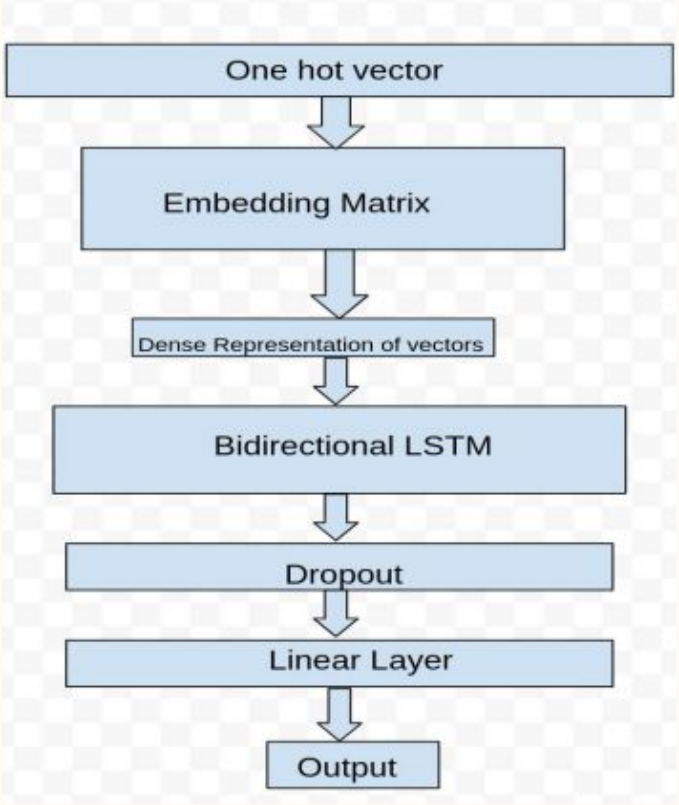
- **Training and evaluation**
  - **Optimizer :** Stochastic Gradient Descent
  - **Loss Function:** Cross Entropy
  - **Accuracy:** Computed as percentage of correct predictions

- **Results and conclusions:**
  - **Test accuracy:** At the end of 5 epochs is approximately 89%
  - **Validation accuracy:** Improved with each epoch
  - **Conclusion:** Linear model does not take into consideration the context

News Article Classification (Cont..)

- **Problem:** Linear model does not take into consideration the context
- **Goal:** To develop a model that provides better accuracy than Linear Model in text classification
- **Model Architecture:**
  - **First layer:** An embedding layer
    - A pre-trained glove embedding is used
  - **Second layer:** A Bidirectional LSTM
  - **Third & Fourth layer:** linear layers which take the input from the LSTM layer and provides the output
  - The regularization method used is **Dropout**

Model Architecture:



News Article Classification (Cont..)

- **Training and evaluation**
  - **Optimizer :** Adam(Adaptive Moment Estimation)
  - **Loss Function:** Cross Entropy
  - **Accuracy:** Computed as percentage of correct predictions
  - **Hyperparameters:** Learning rate, dropout, number of layers etc.

- **Results and conclusions:**
  - **Test accuracy:** At the end of 10 epochs is approximately 91%(improved from 89%)
  - **Reason:** The context in future is also considered for current along with the context in the past
  - Regularization has **improved the accuracy** of the model on unseen data