Assignment - 2 Language Modeling using LSTM

Prakash Kumar Uttam
SR-15247
https://github.com/pkuttam/LSTM_LM

1 Character Level Language Modeling:

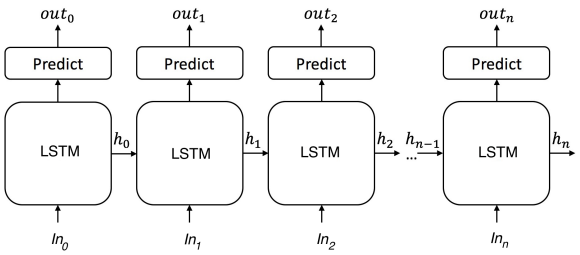

Figure 1: LSTM model for character model

In Network given in Figure 1, $in_0, ..., in_n$ are inputs as character. Each character has one hot encoding corresponding to it.these one hot encoding are fed through LSTM. After output from LSTM, a fully connected layer has been included, which is connected to output layer by softmax. The output will be probability of all the characters which can occurs for the particular input.

input size = number of characters as one hot
Hidden unit size = 100
Output size = Input size = number of characters

perplexity measure

Let $out_n = [p_1, p_2, ..., p_V]$,
here V = Number of distinct character

$$p = \prod_{test\ data} p_V$$

$$perplexity = p^{-1/N}$$

1.1 Results:

- **Perplexity Measure:**

  Tr-G = train on gutenberg dataset (80%)
  Val-G = train on gutenberg dataset (10%)
  Ts-G = test on gutenberg dataset (10%)

Model	LSTM
perplexity	5

- **Generated sentence by character level LSTM model**

  "**ty of the same old from the conscious and seeing the door, and they were a serio**"

2 Word Level Language Modelling:

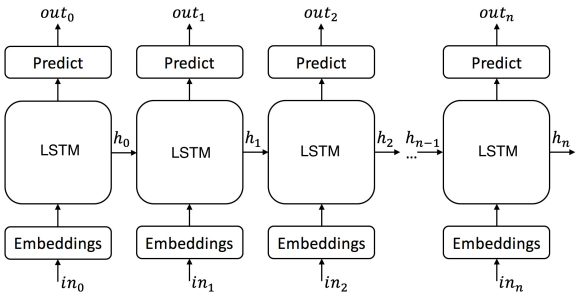

Figure 2: LSTM model for word level model

In Network given in Figure 1, $in_0, ..., in_n$ are inputs as word index. Each word is converted to one hot before feeding to embeddings layer. The embedding layer will generated embeddings for each word. These embeddings are fed through LSTM. After output from LSTM, a fully connected layer has been included, which is connected to output layer by softmax. The output

will be probability of all the words which can occurs for the particular input word.

input size = word as one hot [dimension V]
embedding layer output size = 300
Hidden unit size = 100
Output size = Input size = vocabulary size

perplexity measure

Let $out_n = [p_1, p_2, ..., p_V]$,
here V = vocabulary

$$p = \prod_{test\ data} p_V$$
$$perplexity = p^{-1/N}$$

Results:

- **Perplexity Measure:**

  Tr-G = train on gutenberg dataset (80%)
  Val-G = train on gutenberg dataset (10%)
  Ts-G = test on gutenberg dataset (10%)
  KN = Bayes N-gram with Keynsar-smoothing

model	KN bi-gram	KN tri-gram	LSTM
Gutenberg	98	160	67

- **Generated Sentences by word level LSTM model**

  "**be very case of her own concluded prized blue perhaps to her wretchedness he here**"