Farhad Kazemi

Predicting Supreme Court Decision Making

First Question

SVM with RBF Kernel	SVM with Linear Kernel	MLP	Knn	Decision tree
0.77	0.51	0.35	0.70	0.77

Improvement

1- we need more data to get better accuracy.

2- We need more uncorrelated variables.

3- we can ues ensemble learning for improving the accuracy.

4- we can visualized the dataset for more information about the data. Is the data balanced or imbalanced.

5- The models can be trained and tested by applying a stratified 10-fold cross validation, which uses a held-out 10% of the data at each stage to measure predictive performance.

Second Question

We derive textual features from the text extracted from each section (or subsection) of each case. These are either N-gram features, i.e., contiguous word sequences and abstract semantic topics.

LSTM is a popular semantic representation of text used in NLP and Information Retrieval.

Our goal is to predict whether, in the context of a particular case, a court will classify a worker as an employee or an independent contractor.

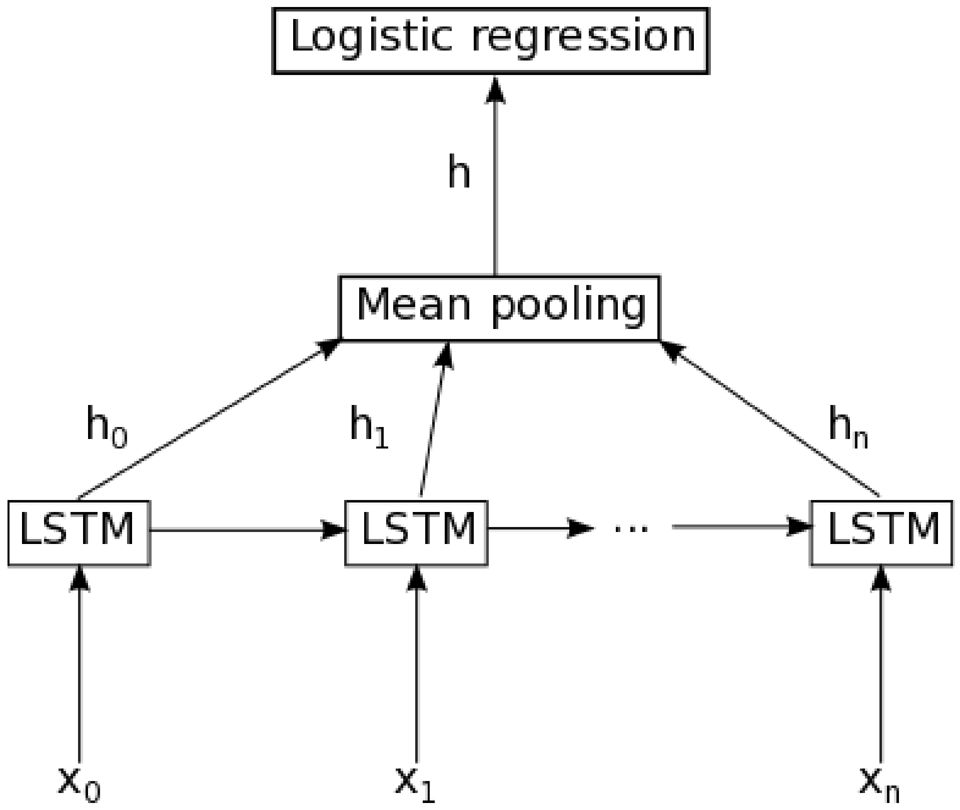
For this purpose, we use each set of textual features from different sections in our case, i.e., N-grams and topics, to train Support Vector Machine (SVM) classifier.

Second Question

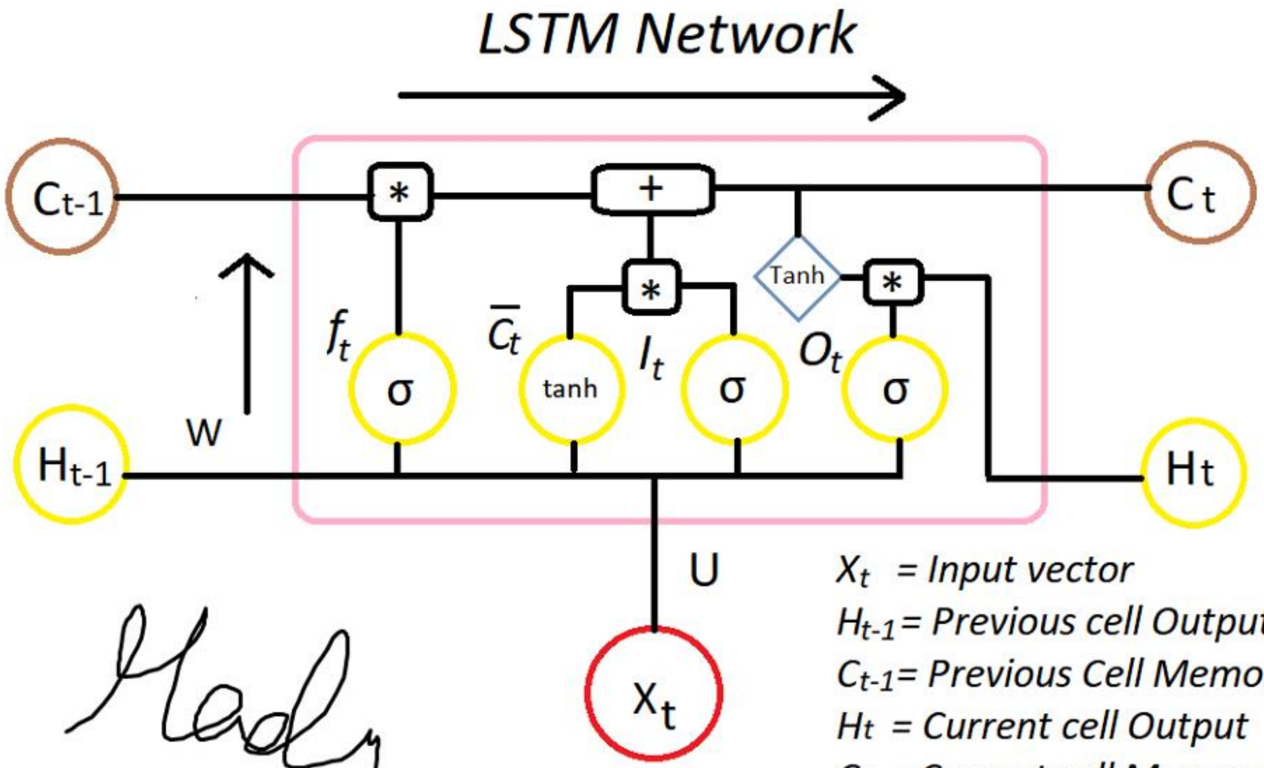
Our model is composed of a single LSTM layer followed by an average pooling and a SVM classifier layer as illustrated in Figure 1.

Thus, from an input sequence x0, x1, x2, ..., xn, the memory cells in the LSTM layer will produce a representation sequence h0, h1, h2, ...,hn.

This representation sequence is then averaged over all time steps resulting in representation h. Finally, this representation is fed to a Logistic regression layer whose target is the class label associated with the input sequence.



LSTM



X_t = Input vector
H_{t-1} = Previous cell Output
C_{t-1} = Previous Cell Memory
H_t = Current cell Output
C_t = Current cell Memory

Mady

$\boxed{*}$ = Element-wise multiplication
$\boxed{+}$ = Element-wise addition

$$f_t = \sigma\,(X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C}_t = \tanh\,(X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma\,(X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma\,(X_t * U_o + H_{t-1} * W_o)$$

$$C_t = f_t * C_{t-1} + I_t * \bar{C}_t$$
$$H_t = O_t * \tanh(C_t)$$

W, U = weight vectors for forget gate (f), candidate (c), i/p gate (I) and o/p gate (O)

Note : These are different weights for different gates, for simpicity's sake, I mentioned W and U