姓名: 林宇辰
学号: 2200013211

Homework 1

课程名称: FNLP
2024 年 3 月 31 日

1 Model Implementation

1.1 Feature Extract

I adopt a simple way to extract features from the dataset. The methodology can be summarized as two following stages.

- **Count the High Frequency Words**

  Intuitively, if we encode all the words occurring in the document to a vector, we will achieve a minimum loss of information. However, this is not practical because the vocabulary is really large (about 70000 distinct words) and one piece of news only contains a few words of the vocabulary, resulting in a sparse matrix. Due to the simplicity of our log-linear model, it has limited capacity to fit sparse data with a high dimension. To address this issue, I count the word frequency in the training dataset for the 4 different labels. I select top 200 words for each category so my feature vector is 800 dimensional.

- **Encode a word frequency vector**

  Once we obtain the vocabulary of our word vector in stage 1, we can generate word frequency vector for each piece of the news according to the vocabulary. This is the input that I feed into the model.

1.2 Log-linear Model

Denote $\mathbf{x} \in \mathbb{R}^d$ as our feature vector where d is the dimension of each vector. In this situation, $d = 800$. Denote $\mathcal{W} \in \mathbb{R}^{c \times d}$ as the weight matrix where d is the feature dimension and c is the number of categories (4 in this situation). Denote $f(\cdot)$ as the soft-max function formulated as following:

$$f(\mathbf{x}) = \frac{\exp \mathbf{x}}{\sum_i \exp x_i}, \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is any vector.

1.2.1 Forward

The forward process is simple, we just need to get the predict probability of different category with soft-max function. Denote $\mathbf{s} \in \mathbb{R}^c$ as the predicted score of each class. It can be derived as following:

$$\begin{aligned} \mathbf{s} &= f\left(\mathcal{W}\mathbf{x}\right) \\ &= \frac{\exp \mathcal{W}\mathbf{x}}{\sum_i \exp\left(\mathcal{W}\mathbf{x}\right)_i} \end{aligned} \tag{2}$$

1.2.2 Backward

Our loss function $L(\cdot)$ can be derived by taking the logarithm of soft-max function:

$$L(\mathbf{x}, \mathbf{y}) = -\mathbf{y} \cdot \log f(\mathbf{x}) \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^n$ is any vector, $\mathbf{y} \in \mathbb{R}^n$ is $\mathbf{x}$'s label and it is a one-hot vector. Set the t-th logit of $\mathbf{y}$ to be non-zero. So the gradient is:

$$\begin{aligned} \nabla_{\mathbf{x}} L &= -\log f(\mathbf{x})_t = -\log f_t \\ &= \frac{\partial L}{\partial f_t} \frac{\partial f_t}{\partial \mathbf{x}} \\ &= \left(-\frac{1}{f_t}\right) \left\{f_t[\delta_{it} - f(\mathbf{x})]\right\} \\ &= -\delta_{it} + f(\mathbf{x}) \\ &= -\mathbf{y} + f(\mathbf{x}) \end{aligned} \tag{4}$$

In our situation, $\hat{\mathbf{x}} = \mathcal{W}\mathbf{x}$ where $\hat{\mathbf{x}}$ is the $\mathbf{x}$ in Eq. 3. Therefore, with using the chain rule, our gradient changes into:

$$\begin{aligned} \nabla_{\mathcal{W}\mathbf{x}} L &= -\mathbf{y} + f(\mathcal{W}\mathbf{x}) \\ \nabla_{\mathcal{W}} L &= [-\mathbf{y} + f(\mathcal{W}\mathbf{x})] \cdot \mathbf{x}. \end{aligned} \tag{5}$$

We use the gradient to update our weight matrix.

$$\mathcal{W} = \mathcal{W} - \alpha \nabla_{\mathcal{W}} L \tag{6}$$

表 1:

	Feature Dimension↓	Training Accuracy↑	Training F1-Score↑	Test Accuracy↑	Test F1-Score↑
Naive	68000	0.73379	0.73421	0.715	0.71546
High Frequency	800	0.84531	0.84477	0.83289	0.83222
High Frequency (remove duplicates)	**466**	**0.84593**	**0.84547**	**0.83460**	**0.83411**

2 Experiment

2.1 Settings

We train 10 epochs over all the train data and test accuracy and f1-score on test data. I concatenate titles and descriptions as joint inputs. All matrix operation is based on numpy. We can achieve about 83% accuracy on test data.

2.2 Ablation

I have done three ablation studies to prove the effectiveness of my feature extracting strategy. As shown in Table 1

- **Naive word frequency vector** Count the words over all vocabulary.
- **High frequency word vector** As mentioned in Sec. 1.1.
- **High frequency word vector with duplicates removed** We remove duplicates in four top-200 vocabulary. The intuition behind this is that there are many meaningless word occurring in different categories, such as ”a”, ”the”, ”and”, etc.