iNLP A4: ELMO

[Naitik Kariwal – 2023201044]

Code explanation:

1. ELMo Model Architecture and Components (elmo_class.py)

The ELMo class implements a contextual word representation model based on a bidirectional LSTM. Below are the key architectural components and functionalities:

Architecture Overview

- **Embedding Layer:** Uses a pretrained word embedding matrix that is passed as an argument. This layer is trainable by default.
- **Projection Layer:** A linear layer (nn.Linear) that projects word embeddings into the appropriate size required for LSTM processing.
- **Bidirectional LSTMs:** Two stacked LSTM layers that capture deep contextual information.
  - The first LSTM processes word sequences bidirectionally.
  - The second LSTM refines contextual embeddings using the outputs from the first LSTM.
- **Lambda Weighting Mechanism (λ):** The model provides different modes to control layer weighting. This forms the task of hyperparameter tuning of ELMo embeddings on downstream task.
  - **Trainable (trainable)**: Learns a set of three lambda (γ) parameters to weigh different LSTM layers.
  - **Frozen (frozen)**: Uses fixed weights, ensuring λ parameters do not update during training.
  - **Function (function)**: A learnable MLP (gamma_mlp) predicts layer weights dynamically.

Key Functionalities

- **freeze_initial()**: Allows freezing of certain layers to prevent gradient updates.
- **freeze_gamma()**: Freezes lambda parameters to maintain static layer weighting.
- **forward(x)**: Computes ELMo representations and outputs weighted word embeddings.
- **get_sequence_embeddings(x)**: Extracts contextual word embeddings for downstream tasks.

2. AGNewsClassifier and Implementation (agnews_classifier.py)

The **AGNewsClassifier** is a deep learning model designed for text classification using ELMo embeddings. Below are the key components:

Architecture

- **Embedding Extractor:** Uses ELMo to extract contextual word embeddings.
- **Bidirectional GRU:** Processes text sequences using a bidirectional GRU layer.
- **Fully Connected Layer:** A linear classifier that maps GRU outputs to class probabilities.
- **Dropout:** Helps in regularization to prevent overfitting.

Key Functionalities

- **forward(x, lengths)**:
  - Extracts ELMo embeddings.
  - Packs and processes sequences using **GRU**.
  - Applies a fully connected classifier to obtain predictions.
- **Fine-tuning Capability:**
  - The classifier can operate with frozen or trainable ELMo embeddings.
  - fine_tune_elmo determines whether the ELMo model updates during training.

3. Dataset Preparation (agnews_dataset.py)

The AGNews dataset preparation involves several steps to preprocess text data for training. The key components include:

Dataset Class (AGNewsDataset)

- **Text Tokenization:** Converts raw text into tokenized words.
- **Indexing:** Maps words to integer indices using a predefined Brown corpus vocabulary.
- **Padding/Truncation:** Ensures all sequences have a fixed length of **100**.

DataLoader Functions

- **collate_fn_agnews(batch)**:
  - Pads sequences dynamically.
  - Stores sequence lengths for later use in packed sequences.
- **prepare_agnews_dataloaders()**:
  - Splits data into training, validation, and test sets.
  - Converts datasets into PyTorch DataLoader objects.

Results & Analysis:

1. The results from **hyperparameter tuning** downstream task with ELMo embeddings are as follows:

    a. Frozen

```
Classifier model path: ../classifier/classifier_elmo_frozen.pth
Classifier model loaded.

Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.93      0.93      0.93     27000
      Sports       0.96      0.98      0.97     27000
    Business       0.93      0.87      0.90     27000
    Sci/Tech       0.89      0.92      0.90     27000

    accuracy                           0.93    108000
   macro avg       0.93      0.93      0.93    108000
weighted avg       0.93      0.93      0.93    108000


Confusion Matrix:
[[25032   647   663   658]
 [  265 26480    70   185]
 [  849   199 23567  2385]
 [  755   224  1049 24972]]

✅Accuracy: 0.9264
✅Precision: 0.9267
✅Recall: 0.9264
✅F1-score: 0.9262
```

```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.86      0.86      0.86      1900
      Sports       0.92      0.94      0.93      1900
    Business       0.84      0.78      0.81      1900
    Sci/Tech       0.80      0.84      0.82      1900

    accuracy                           0.85      7600
   macro avg       0.85      0.85      0.85      7600
weighted avg       0.85      0.85      0.85      7600


Confusion Matrix:
[[1633   80   93   94]
 [  52 1781   22   45]
 [ 117   39 1475  269]
 [  99   40  172 1589]]

✅Accuracy: 0.8524
✅Precision: 0.8525
✅Recall: 0.8524
✅F1-score: 0.8520
```

    b. Trainable

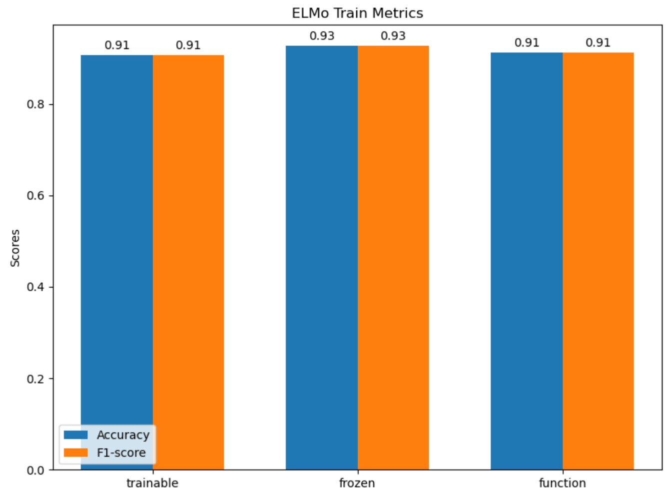
```
Classifier model path: ../classifier/classifier_elmo_trainable.pth
Classifier model loaded.

Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.92      0.89      0.91     27000
      Sports       0.94      0.98      0.96     27000
    Business       0.89      0.87      0.88     27000
    Sci/Tech       0.87      0.89      0.88     27000

    accuracy                           0.91    108000
   macro avg       0.91      0.91      0.91    108000
weighted avg       0.91      0.91      0.91    108000


Confusion Matrix:
[[24127   871  1067   935]
 [  280 26404   111   205]
 [  796   312 23405  2487]
 [  930   428  1680 23962]]

✅Accuracy: 0.9065
✅Precision: 0.9064
✅Recall: 0.9065
✅F1-score: 0.9062
```

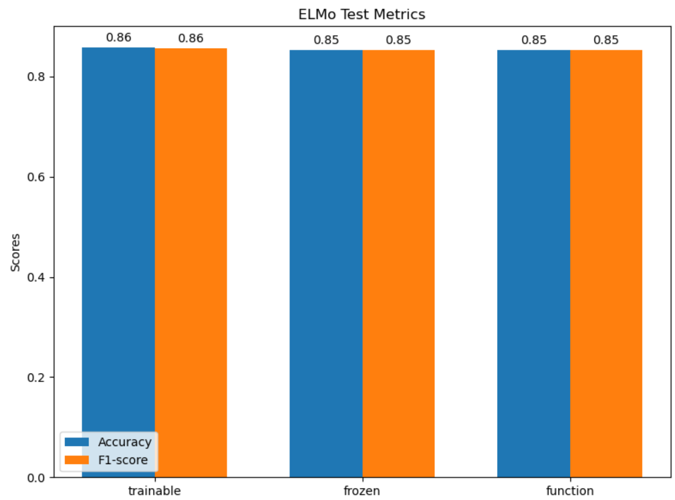
```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.88      0.86      0.87      1900
      Sports       0.91      0.95      0.93      1900
    Business       0.82      0.79      0.81      1900
    Sci/Tech       0.81      0.83      0.82      1900

    accuracy                           0.86      7600
   macro avg       0.86      0.86      0.86      7600
weighted avg       0.86      0.86      0.86      7600


Confusion Matrix:
[[1627   84  109   80]
 [  38 1812   23   27]
 [  85   48 1506  261]
 [  92   43  195 1570]]

✅Accuracy: 0.8572
✅Precision: 0.8567
✅Recall: 0.8572
✅F1-score: 0.8567
```

    c. Function

```
h_l0_reverse', 'rnn.bias_ih_l0_reverse', 'rnn.bias_hh_
Classifier model loaded.

Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.92      0.91      0.92     27000
      Sports       0.96      0.98      0.97     27000
    Business       0.93      0.83      0.88     27000
    Sci/Tech       0.84      0.93      0.89     27000

    accuracy                           0.91    108000
   macro avg       0.91      0.91      0.91    108000
weighted avg       0.91      0.91      0.91    108000


Confusion Matrix:
[[24613   656   699  1032]
 [  261 26334    64   341]
 [ 1097   250 22356  3297]
 [  822   180   801 25197]]

✅Accuracy: 0.9120
✅Precision: 0.9143
✅Recall: 0.9120
✅F1-score: 0.9118
```

```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.86      0.86      0.86      1900
      Sports       0.92      0.93      0.93      1900
    Business       0.86      0.76      0.81      1900
    Sci/Tech       0.77      0.86      0.81      1900

    accuracy                           0.85      7600
   macro avg       0.86      0.85      0.85      7600
weighted avg       0.86      0.85      0.85      7600

Confusion Matrix:
[[1628   80   78  114]
 [  49 1776   13   62]
 [ 113   33 1446  308]
 [  96   36  136 1632]]

✅Accuracy: 0.8529
✅Precision: 0.8553
✅Recall: 0.8529
✅F1-score: 0.8527
```





We observe that the metrics on test set across all three hyperparameter settings are **very close**, i.e. 85-86%. However, on trainset, the metrics follow Frozen (~92%), Function (~91%) and Trainable (~90%).

2. The results from downstream task with **static embeddings** are as follows:
   a. SVD

```
Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.91      0.88      0.89     27000
      Sports       0.96      0.94      0.95     27000
    Business       0.86      0.86      0.86     27000
    Sci/Tech       0.83      0.88      0.86     27000

    accuracy                           0.89    108000
   macro avg       0.89      0.89      0.89    108000
weighted avg       0.89      0.89      0.89    108000


Confusion Matrix:
[[23782   714  1326  1178]
 [  588 25435   235   742]
 [  817   206 23175  2802]
 [ 1044   154  2056 23746]]

✅Accuracy: 0.8902
✅Precision: 0.8913
✅Recall: 0.8902
✅F1-score: 0.8905
```

```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.86      0.83      0.85      1900
      Sports       0.93      0.91      0.92      1900
    Business       0.81      0.79      0.80      1900
    Sci/Tech       0.77      0.82      0.79      1900

    accuracy                           0.84      7600
   macro avg       0.84      0.84      0.84      7600
weighted avg       0.84      0.84      0.84      7600


Confusion Matrix:
[[1584   66  124  126]
 [  57 1723   31   89]
 [ 102   33 1506  259]
 [ 104   29  203 1564]]

✅Accuracy: 0.8391
✅Precision: 0.8410
✅Recall: 0.8391
✅F1-score: 0.8397
```

   b. CBOW

```
Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.90      0.85      0.87     27000
      Sports       0.93      0.95      0.94     27000
    Business       0.84      0.84      0.84     27000
    Sci/Tech       0.82      0.85      0.83     27000

    accuracy                           0.87    108000
   macro avg       0.87      0.87      0.87    108000
weighted avg       0.87      0.87      0.87    108000


Confusion Matrix:
[[22934  1010  1686  1370]
 [  503 25694   295   508]
 [  864   346 22640  3150]
 [ 1160   543  2475 22822]]

✅Accuracy: 0.8712
✅Precision: 0.8717
✅Recall: 0.8712
✅F1-score: 0.8712
```

```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.87      0.83      0.85      1900
      Sports       0.91      0.93      0.92      1900
    Business       0.79      0.80      0.80      1900
    Sci/Tech       0.79      0.81      0.80      1900

    accuracy                           0.84      7600
   macro avg       0.84      0.84      0.84      7600
weighted avg       0.84      0.84      0.84      7600


Confusion Matrix:
[[1572   82  142  104]
 [  54 1762   36   48]
 [  96   33 1514  257]
 [  88   56  216 1540]]

✅Accuracy: 0.8405
✅Precision: 0.8409
✅Recall: 0.8405
✅F1-score: 0.8405
```

   c. Skip-gram

```
Evaluating on Train Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.90      0.93      0.91     27000
      Sports       0.95      0.98      0.97     27000
    Business       0.90      0.87      0.89     27000
    Sci/Tech       0.90      0.88      0.89     27000

    accuracy                           0.91    108000
   macro avg       0.91      0.91      0.91    108000
weighted avg       0.91      0.91      0.91    108000


Confusion Matrix:
[[25015   711   783   491]
 [  289 26482    63   166]
 [ 1067   291 23601  2041]
 [ 1438   285  1637 23640]]

✅Accuracy: 0.9142
✅Precision: 0.9139
✅Recall: 0.9142
✅F1-score: 0.9139
```

```
Evaluating on Test Set:

Classification Report:
              precision    recall  f1-score   support

       World       0.83      0.87      0.85      1900
      Sports       0.91      0.94      0.92      1900
    Business       0.82      0.78      0.80      1900
    Sci/Tech       0.81      0.79      0.80      1900

    accuracy                           0.84      7600
   macro avg       0.84      0.84      0.84      7600
weighted avg       0.84      0.84      0.84      7600

Confusion Matrix:
[[1646   86  101   67]
 [  67 1781   17   35]
 [ 141   39 1474  246]
 [ 138   51  214 1497]]

✅Accuracy: 0.8418
✅Precision: 0.8410
✅Recall: 0.8418
✅F1-score: 0.8411
```

We observe that the metrics on testset are again very close: Skip-gram **(~84.1%),** CBOW (~84%) and SVD (83.9%) while those on train follow Skip-gram (~91%), SVD (89%) and SBOW (87%).

3. Some description text tested during inference are as follows:

```
PS C:\Users\NAITIK\Desktop\Semester-4\INLP\Assignments\A4\code> python inference.py "../classifier/classifier_elmo_frozen.pth" "Police in Burundi #39;s capital, Bujumbura, used tear gas to break up a demonstration Wednesday held to protest the massacre of Congolese Tutsi refugees."
class-1 99.77%
class-2 0.11%
class-3 0.06%
class-4 0.05%
PS C:\Users\NAITIK\Desktop\Semester-4\INLP\Assignments\A4\code> python inference.py "../classifier/classifier_elmo_frozen.pth" "AP - Trying to get the best possible ballpark deal for the Montreal Expos, major league baseball instructed its lawyers to press ahead with negotiations involving four of the areas bidding for the team."
class-1 0.16%
class-2 99.81%
class-3 0.02%
class-4 0.02%
PS C:\Users\NAITIK\Desktop\Semester-4\INLP\Assignments\A4\code> python inference.py "../classifier/classifier_elmo_frozen.pth" "Health care and consumer products maker Johnson  amp; Johnson (JNJ.N: Quote, Profile, Research) is in negotiations to acquire medical-device maker Guidant Corp."
class-1 1.38%
class-2 0.01%
class-3 95.11%
class-4 3.50%
PS C:\Users\NAITIK\Desktop\Semester-4\INLP\Assignments\A4\code> python inference.py "../classifier/classifier_elmo_frozen.pth" "AUGUST 18, 2004 (IDG NEWS SERVICE) - A majority of US home Internet users now have broadband, according to a survey by NetRatings Inc. "
class-1 0.22%
class-2 0.08%
class-3 4.40%
class-4 95.30%
```

Analysis:

1. Analysis and Ranking of Embedding Methods

Observed Performance

- **Test Set** Metrics:
  - **ELMo (Frozen):** ~85–86%

- **Skip-gram:** ~84.1%
- **CBOW:** ~84.0%
- **SVD:** ~83.9%

- **Training Set** Metrics**:**
  - **Skip-gram:** ~91%
  - **CBOW:** ~87%
  - **SVD:** ~89%
  - **ELMo (Frozen):** ~92%

Rankings:

1. **ELMo (Frozen):**
   - **Overall Performance:** Highest on the training set (~92%) and among the top performers on the test set (~85–86%).
   - **Why It Excels:**
     - The pre-trained ELMo model provides robust, contextually rich representations that capture both syntactic and semantic nuances.
     - Freezing the λ weights preserves these robust representations during fine-tuning, ensuring minimal deviation from the strong pre-trained signals.
2. **Skip-gram:**
   - **Overall Performance:** Test performance of ~84.1% and training performance of ~91%.
   - **Why It Excels:**
     - Skip-gram's objective to predict context words helps capture word relationships effectively, particularly for less frequent words.
     - Its performance is competitive with ELMo, although it lacks the deep contextualization provided by a model like ELMo.
3. **SVD-based Embeddings:**
   - **Overall Performance:** Test performance of ~83.9% and training performance of ~89%.
   - **Why It Lags:**
     - SVD embeddings are based on global co-occurrence statistics and are inherently linear.

        - This method may not capture complex, nonlinear language patterns as effectively as predictive models like Skip-gram or contextual models like ELMo.

4. **CBOW:**
    - **Overall Performance:** Test performance of ~84.0% and training performance of ~87%.
    - **Why It Lags:**
        - The averaging mechanism in CBOW can **smooth out** important distinctions between words.
        - While effective and **computationally efficient**, this averaging leads to slightly lower performance compared to Skip-gram and ELMo.

2. Hyperparameter Settings in the ELMo Model

All three hyperparameter settings achieve **very similar** performance (~85–86%) on test set.

Rankings based on train set:

1. Frozen:
    - Training Performance: Slightly highest at ~92%.
    - Rationale: By keeping the λ weights fixed, the model leverages robust pre-trained representations without risking overfitting to the training data. This helps the model retain its **generalization** ability.
2. Function:
    - Training Performance: Next best at ~91%.
    - Rationale: The dynamic function introduces a controlled amount of **flexibility**. It allows the model to adjust layer contributions without significantly altering the pre-trained structure, yielding performance nearly as good as the frozen setting.
3. Trainable:
    - Training Performance: ~90%.
    - Rationale: Although allowing λ parameters to be fine-tuned could potentially adapt the model more closely to the task, it also introduces extra degrees of freedom that might lead to slight overfitting on the training data. The modest difference suggests that the fine-tuning of λ parameters does not have a dramatic effect on the overall performance.

Impact on Downstream Task Performance

- **Generalization:**
  On the test set, the performance across all three hyperparameter settings is very close. This indicates that while different modes slightly affect training dynamics, their impact on generalization is minimal. The model's ability to leverage pre-trained representations appears **robust to variations** in how layer weights are treated.
- **Trade-Offs:**
  - **Frozen mode** minimizes the risk of overfitting by keeping the strong pre-trained signals intact.
  - **Function mode** provides moderate flexibility, potentially adapting better to specific nuances in the training data without significant instability.
  - **Trainable mode** offers the most freedom but may require additional regularization or tuning to avoid overfitting.