1)Framework Used:

1. Pytorch for fine tuning and linear probing
2. Keras for model training from scratch using word2vec

Two source code file.

1) Transfer learning.ipynb

2) word2vec.ipynb

2) Data Preprocessing:

Performed – Downloaded and loaded the imdb dataset in separate training and testing dataset of 25000 each and used tokenizer of the transformer to perform the preprocessing such as padding and encoding the value for the first two methods fine tuning and linear probing. For 3rd method used the word2vec to perform the same.

**Sample Sequence:**

Input : {'text': 'I rented I AM CURIOUS-YELLOW from my video store because of all the controversy that surrounded it when it was first released in 1967. I also heard that at first it was seized by U.S. customs if it ever tried to enter this country, therefore being a fan of films considered "controversial" I really had to see this for myself.<br /><br />The plot is centered around a young Swedish drama student named Lena who wants to learn everything she can about life. In particular she wants to focus her attentions to making some sort of documentary on what the average Swede thought about certain political issues such as the Vietnam War and race issues in the United States. In between asking politicians and ordinary denizens of Stockholm about their opinions on politics, she has sex with her drama teacher, classmates, and married men.<br>.

Target: 'label': 0

Encoding: 'input\_ids': [101, 1045, 12524, 1045, 2572, 8025, 1011, 3756, 2013, 2026, 2678, 3573, 2138, 1997, 2035, 1996, 6704, 2008, 5129, 2009, 2043, 2009, 2001, 2034, 2207, 1999, 3476, 1012, 1045, 2036, 2657, 2008, 2012, 2034, 2009, 2001, 8243, 2011, 1057, 1012, 1055, 1012, 8205, 2065, 2009, 2412, 2699, 2000, 4607, 2023, 2406, 1010, 3568, 2108, 1037, 5470, 1997, 3152, 2641, 1000, 6801, 1000, 1045, 2428, 2018, 2000, 2156, 2023, 2005, 2870, 1012, 1026, 7987, 1013, 1028, 1026, 7987, 1013, 1028, 1996, 5436, 2003, 8857, 2105, 1037, 2402, 4467, 3689, 3076, 2315, 14229, 2040, 4122, 2000, 4553, 2673, 2016, 2064, 2055, 2166, 1012, 1999, 3327, 2016, 4122, 2000, 3579, 2014, 3086, 2015, 2000, 2437, 2070, 4066, 1997, 4516, 2006, 2054, 1996, 2779, 25430, 14728, 2245, 2055, 3056, 2576, 3314, 2107, 2004, 1996, 5148, 2162, 1998, 2679, 3314, 1999, 1996, 2142, 2163, 1012, 1999, 2090, 4851, 8801, 1998, 6623, 7939, 4697, 3619, 1997, 8947, 2055, 2037, 10740, 2006, 4331, 1010, 2016, 2038, 3348, 2007, 2014, 3689, 3836, 1010, 19846, 1010, 1998, 2496, 2273, 1012, 1026, 7987, 1013, 1028, 1026, 7987, 1013, 1028, 2054, 8563, 2033, 2055, 1045, 2572, 8025, 1011, 3756, 2003, 2008, 2871, 2086, 3283, 1010, 2023, 2001, 2641, 26932, 1012, 2428, 1010, 1996, 3348, 1998, 16371, 25469, 5019, 2024, 2261, 1998, 2521, 2090, 1010, 2130, 2059, 2009, 1005, 1055, 2025, 2915, 2066, 2070, 10036

3)

**Fine Tuning:**

Learning rate: 5e-5

Epochs = 5

Transformer by default provides a function to add a classifier layer to add it to the pretrained model. The syntax is given below:

DistilBertForSequenceClassification.from\_pretrained('distilbert-base-uncased', num\_labels=2).to(device)

The trainer model used in the program default takes the adam optimizer and executes with other default settings.

Evaluation Result for the above:

{'eval\_loss': 0.4668221175670624, **'eval\_accuracy'**: 0.93012, 'eval\_runtime': 236.5302, 'eval\_samples\_per\_second': 105.695, 'eval\_steps\_per\_second': 3.306, 'epoch': 5.0}

Test Exp: Ran this code for 100 epoch and still gaves accuracy around the above value, not much improvement.

Total time to execute the above: 54:59 mm:ss

4)

**Linear Probing**:

All the layers are freezed except the classification layer and trained for 5 epochs.

Evaluation Result for the above:

{'eval\_loss': 0.3245013356208801, 'eval\_accuracy': 0.86224, 'eval\_runtime': 235.0447, 'eval\_samples\_per\_second': 106.363, 'eval\_steps\_per\_second': 3.327, 'epoch': 5.0}

Total time to execute the above: 20:59 mm:ss

5) **LSTM model using Word2vec embeddings:**

No of layers = 2

Optimizer = adam

Learning rate = default learning rate 0.01

Evaluation Result for the above:

782/782 [==============================] - 38s 48ms/step - loss: 0.8031 - accuracy: 0.8006

0.8031217455863953 0.8006399869918823

Total time to execute the above: approximately 5 hrs

6) when we compare the model performance in terms of:

i) Accuracy : Pretrained model with fine tuning gave high accuracy compare to other model. Whereas the model developed from the scratch starts to over fit, as we can see the accuracy during the training steps are high

but it is low for testing data. Whereas for the linear probing since it wasn’t able to adjust the internal weights, it is not able to fit the model as per the data compared to fine tuning.

ii) Execution Time: using a pretraining model has less training time compared to developing a model from the scratch especially when we can use linear probing, but it takes lots of time to develop a model from the scratch.

7)

Advantages:

1. Less training time
2. Can train with less resources
3. Can train to get more accuracy.

Limitations:

1. Pretrained model are developed for specific tasks and may not be suitable for other applications or tasks. Example: model trained on images are not suitable for languages.
2. Limited accuracy: It may not well perform on unseen data and the accuracy gets saturated.