CS4650 Final Presentation Interpret, visualize BERT

Xueren Ge

Haibei Zhu

Goal

- Understand BERT
  - whether BERT create reasonable embeddings
  - explore each layer’s necessity
- Find out possible improvement to solve “anistropy” problem

Method

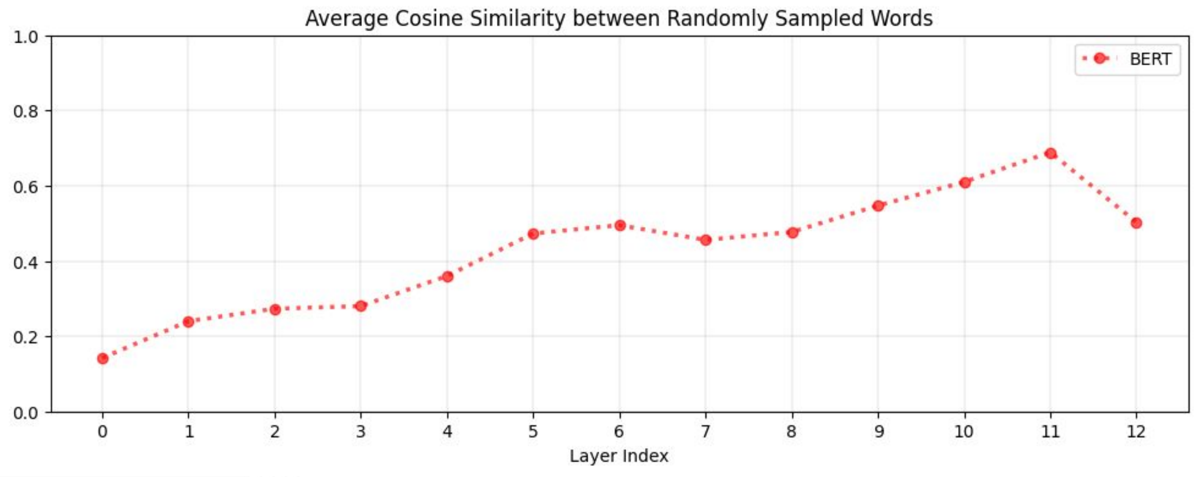
- Remove Layer 12 in BERT
- Biased Dataset Analysis
- Contrastive Learning

Data

- Stanford Natural Language Inference (SNLI) Corpus
  - https://nlp.stanford.edu/projects/snli/

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Anisotropy



We randomly select 2 words and calculate the cosine similarity, and find that as layer goes deeper, any words cosine similarity grows □

Any 2 random words have similar representation, which means, the whole vector space tend to be a cone instead of ball.

- How to adjust for it?
  - Use anisotropy baseline

$$\mathit{Baseline}(f_\ell) = \mathbb{E}_{x,y\sim U(\mathcal{O})}\left[\cos(f_\ell(x), f_\ell(y))\right]$$

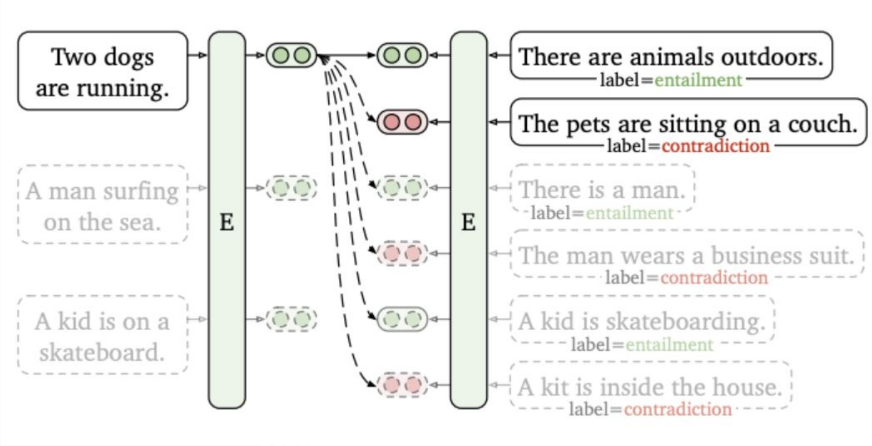
$$\mathit{SelfSim}^*_\ell(w) = \mathit{SelfSim}_\ell(w) - \mathit{Baseline}(f_\ell)$$

Contrastive Learning

- Loss Function

$$\mathcal{L} = -\sum_{i,j=1}^{b} t_{i,j} \log p_{i,j} = -\sum_{i,j=1}^{b} t_{i,j} \log \frac{e^{s_{i,j}}}{\sum_{j} e^{s_{i,j}}} = -\sum_{i,j=1}^{b} t_{i,j} s_{i,j}$$

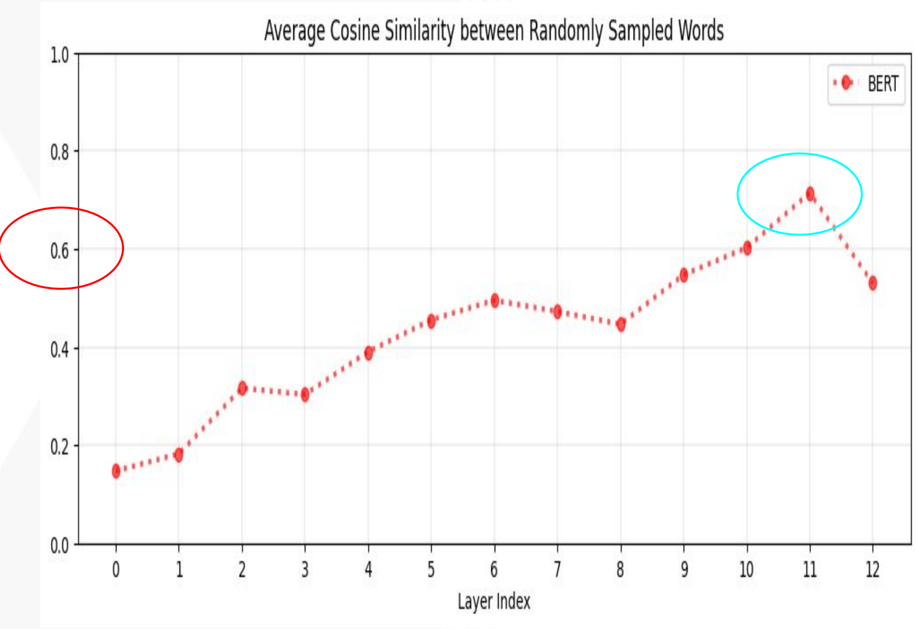
b is batch size, t_i,j is the label,- one hot matrix. s_i,j is the similarity score between sample i and j

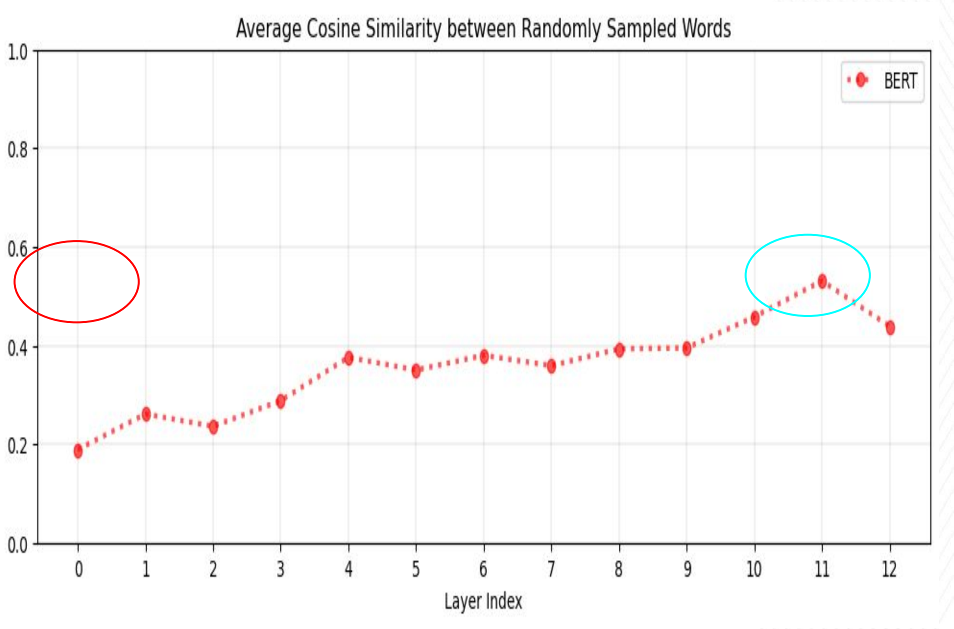


Use cross entropy loss of negative samples with in a batch.
Make the feature of x more similar to the positive sample and less similar to the feature of the negative sample

Results

- Contrastive Learning





As the layer goes deeper, contrastive-learning BERT model's any words cosine similarity grows slower than the original BERT. Anisotropy issue alleviate

Remove Layer 12 Experiment

Experiment 1:

Use SNLI to do **classification**:

```
length of train loader: 17168
length of val loader: 308
```

whether premise can infer hypothesis, if so, label is entailment. if not, label is contradiction. If not sure, label is neutral.

Premise: A man inspects the uniform of a figure in some East Asian country.

Hypothesis: The man is sleeping.

Label: *contradiction*

Original BERT

```
      (11): BertLayer(
        (attention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
  (pooler): BertPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
(dropout): Dropout(p=0.1, inplace=False)
(classifier): Linear(in_features=768, out_features=3, bias=True)
)
```

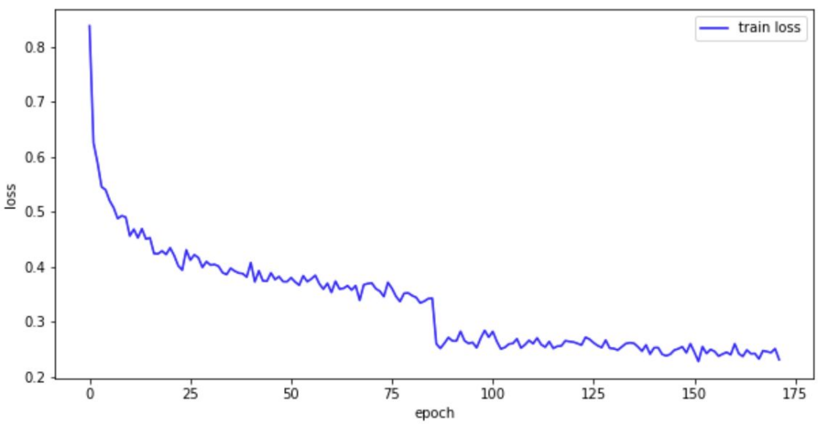
BERT w/o Layer12

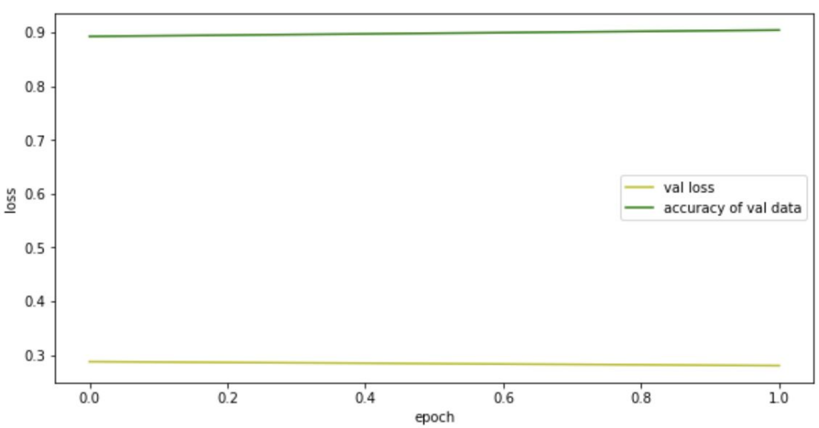
```
        (10): BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (classifier): Sequential(
    (0): Linear(in_features=768, out_features=50, bias=True)
    (1): ReLU()
    (2): Linear(in_features=50, out_features=3, bias=True)
  )
)
```

Remove Layer 12 Experiment

During fine-tuning, I set all parameters as the same
epochs = 2
optimizer = AdamW(model.parameters(), lr=5e-5, eps=1e-8)
loss_fn = nn.CrossEntropyLoss()

Original BERT

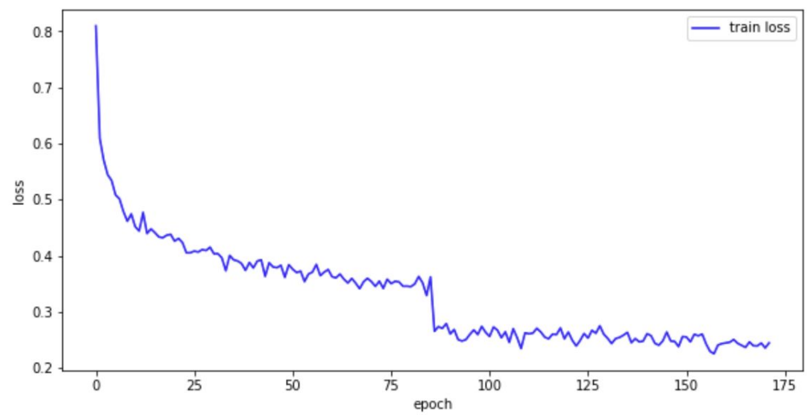


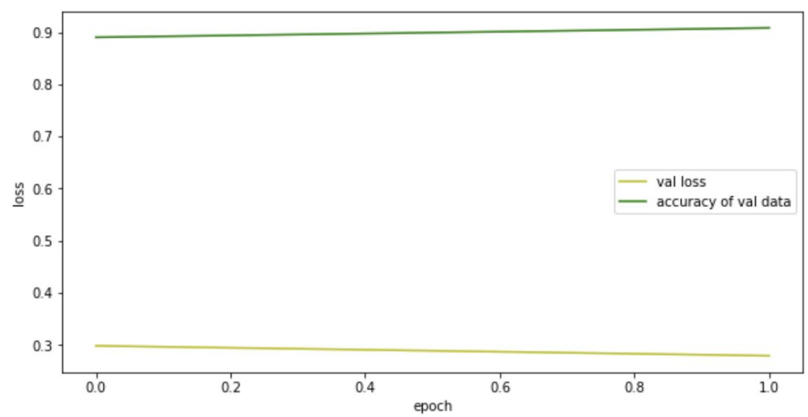


Accuracy on test dataset: 90.45195439739413

total time: 6140.41s

BERT w/o Layer 12





Accuracy on test dataset: 90.32980456026058

total time: 5623.56s

Remove Layer 12 Experiment

Experiment 2:

Use **MedWeb** to do **classification**:

```
length of train loader: 7021
length of val loader: 1505
```

Reviews to predict satisfaction level (1,2,3,4,5,6) for drugs

1	Age	Condition	Date	Drug	DrugId	EaseofUse	Effectivene	Reviews	Satisfactior	Sex	Sides	UsefulCount
2	75 or over	Stuffy Nose	########	25dph-7.5p	146724	5	5	I'm a retire	5	Male	Drowsiness	0

Original BERT

```
    (11): BertLayer(
      (attention): BertAttention(
        (self): BertSelfAttention(
          (query): Linear(in_features=768, out_features=768, bias=True)
          (key): Linear(in_features=768, out_features=768, bias=True)
          (value): Linear(in_features=768, out_features=768, bias=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
        (output): BertSelfOutput(
          (dense): Linear(in_features=768, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
      (intermediate): BertIntermediate(
        (dense): Linear(in_features=768, out_features=3072, bias=True)
      )
      (output): BertOutput(
        (dense): Linear(in_features=3072, out_features=768, bias=True)
        (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (dropout): Dropout(p=0.1, inplace=False)
      )
    )
  )
)
(pooler): BertPooler(
  (dense): Linear(in_features=768, out_features=768, bias=True)
  (activation): Tanh()
)
)
(dropout): Dropout(p=0.1, inplace=False)
(classifier): Linear(in_features=768, out_features=3, bias=True)
)
```

BERT w/o Layer12

```
      (10): BertLayer(
        (attention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
  (pooler): BertPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
(classifier): Sequential(
  (0): Linear(in_features=768, out_features=50, bias=True)
  (1): ReLU()
  (2): Linear(in_features=50, out_features=3, bias=True)
)
)
```

Remove Layer 12 Experiment

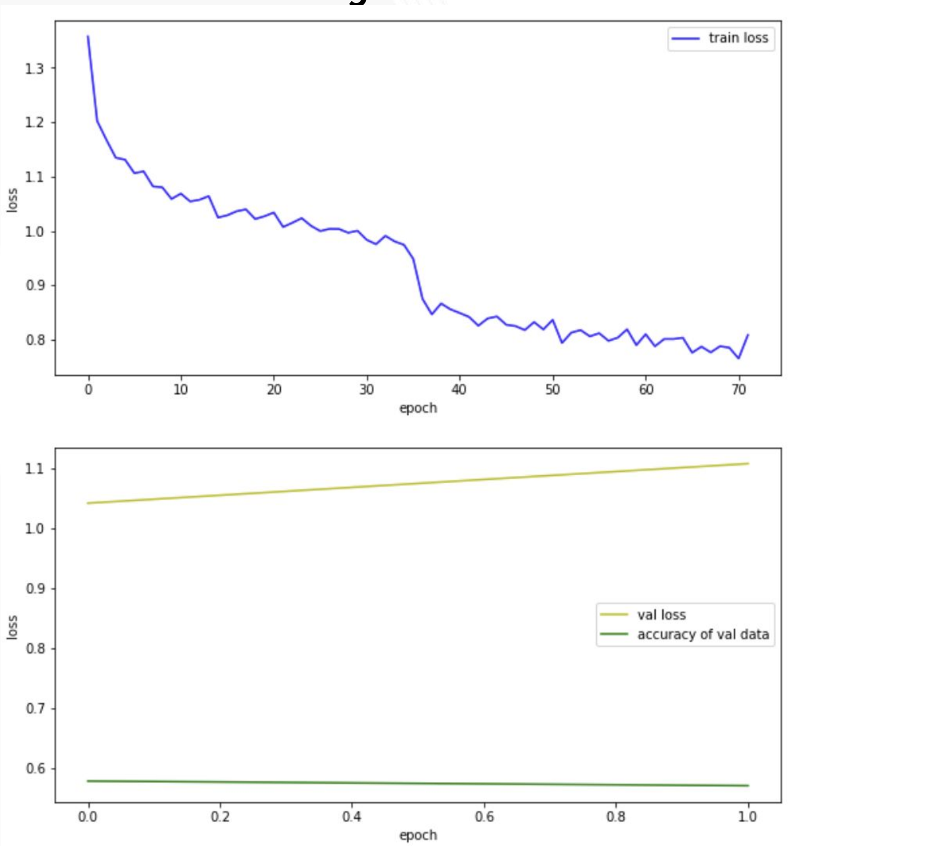
During fine-tuning, I set all parameters as the same

epochs = 2

optimizer = AdamW(model.parameters(), lr=5e-5, eps=1e-8)

loss_fn = nn.CrossEntropyLoss()

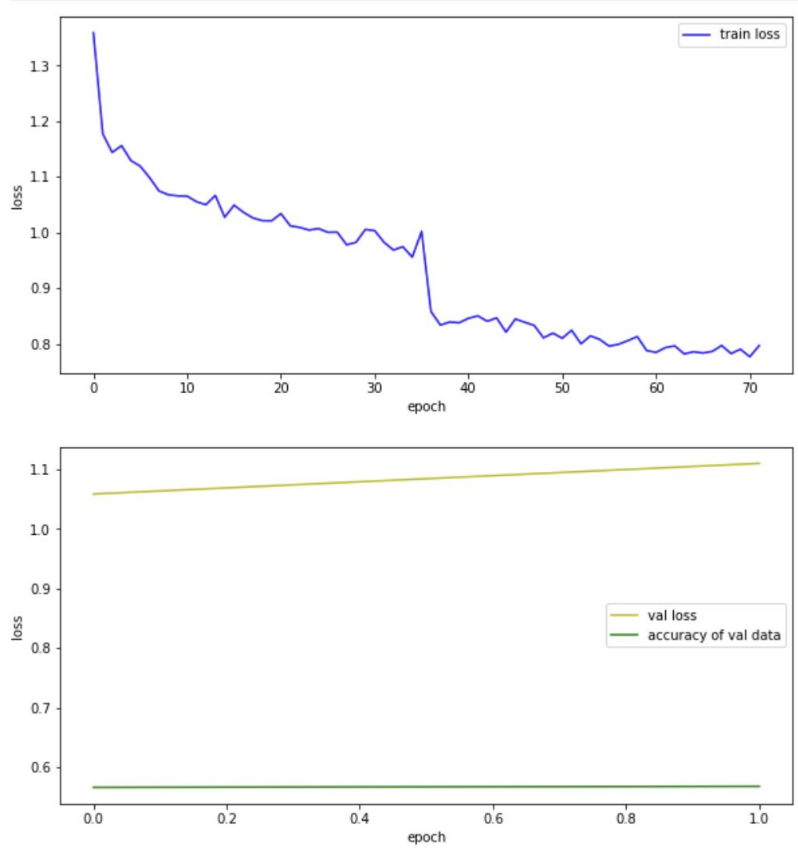
Original BERT



`Accuracy on test dataset: 58.90691741813004`

total time: 2652.56s

BERT w/o Layer 12



`Accuracy on test dataset: 58.66961319411485`

total time: 2447.34s

Biased Dataset Experiment

Experiment:

Use **SNLI** to do **classification**:

we remove sentences which contains words related to “female” in training dataset but keep original validation and test dataset.

```
forbidden_words = ['women', 'woman', 'girl', 'girls', 'mother', 'wife', 'female']
```

Biased SNLI

```
length of train loader: 13197
length of val loader: 308
```

```
Accuracy on test dataset: 59.46661237785016
```

Original SNLI

```
length of train loader: 17168
length of val loader: 308
```

```
Accuracy on test dataset: 90.45195439739413
```

Conclusion:

BERT may not perform well in biased dataset, we should make sure dataset is unbiased.

Conclusion

- Data-biased
  - BERT may not perform well in biased dataset, we should make sure dataset is unbiased.
- Structure
  - We used Hugging Face to freeze layer 12 and found out there is no much difference compare to the original BERT but the process is much faster
- Anisotropy
  - Contrastive learning could alleviate anisotropy by make feature more similar to it's positive sample and less similar to it's negative sample