NLP Final Project Report
VAE based sentiment analysis

312581006 張宸瑋

1. Motivation

Because I believe that the embedding vectors obtained from these sentences through a pretrained BERT model reflect a certain relationship between adjacent vectors in the embedding space. Moreover, there exists a transformation that maps these vectors into a two-dimensional space composed of Arousal and Valence, preserving their positional relationship. Conversely, this relationship also holds true in reverse (Figure 1).

Therefore, I think this concept is similar to the encoder-decoder architecture. The encoder is responsible for transforming each sentence's embedding into a latent space composed of Arousal and Valence. Then, through the decoder, the latent space is converted back into the original embedding.

In our dataset, we have calculated the mean and standard deviation for each sentence through manual annotations. Therefore, I think that this task is well-suited for a VAE model because its concept involves the encoder predicting a set of mean and standard deviation values. By using these standard deviations along with values sampled from a normal distribution, we obtain the output of our final latent space. This allows for smoother results from the decoder. Additionally, we can use this architecture to enable the model to input sentence embeddings and output corresponding mean and standard deviation values. This approach ensures that adjacent points in the latent space, when decoded back to the original embedding by the decoder, maintain their original relationships.

In this project, I hope to accomplish this sentiment analysis task using such a model architecture.

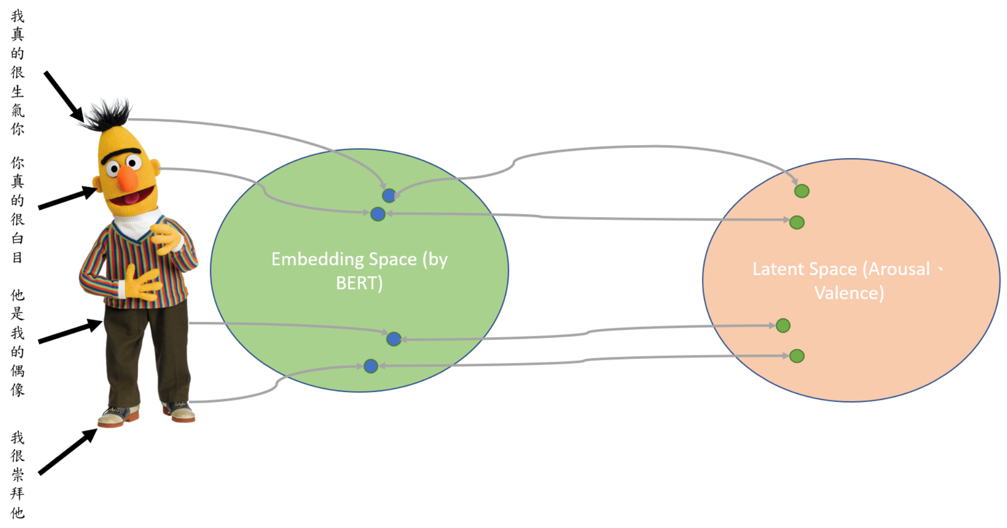


Figure 1

2. Related work

A Variational Autoencoder (VAE) is a type of generative model that learns to encode data into a latent space and then decode it back to reconstruct the original input (Figure 2). The architecture of a VAE consists of two main components: the encoder and the decoder.

1. **Encoder**: The encoder is a neural network that takes the input data and maps it to a latent space. Unlike a traditional autoencoder, a VAE's encoder outputs two vectors: the mean and the variance, which define a Gaussian distribution in the latent space. This distribution allows the VAE to sample points from the latent space during training.
2. **Latent Space**: The latent space is a lower-dimensional representation of the input data. In a VAE, the latent space is characterized by a probabilistic distribution, typically a Gaussian distribution. This probabilistic approach allows for smoother interpolation and better generalization in generating new data (Figure 3).
3. **Decoder**: The decoder is another neural network that takes points sampled from the latent space and maps them back to the original data space. The decoder aims to reconstruct the input data from the sampled points, ensuring that the generated data resembles the original input.

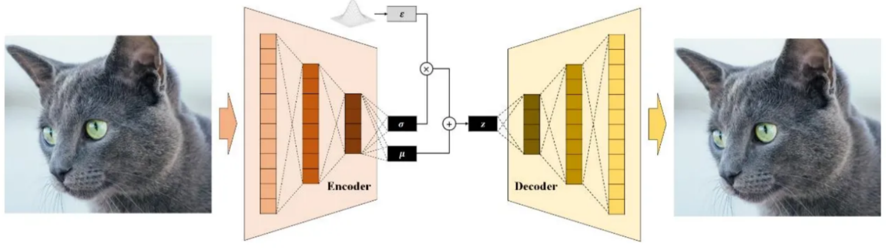
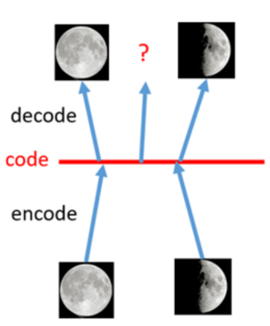
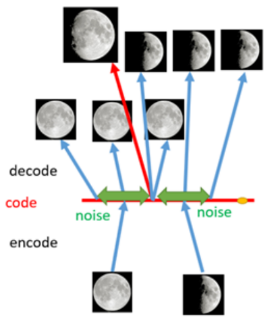


Figure 2 : The architecture of a Variational Autoencoder (VAE), showing an encoder transforming an input image of a cat into a latent space, and a decoder reconstructing the image from the latent representation.

Advantage of VAE



AutoEncoder



VAE

Figure 3 : The advantage of VAEs in generating more diverse and continuous data samples by incorporating noise into the latent space.

3. Method Overview

Based on the architecture of a Variational Autoencoder (VAE), we have the encoder predict the mean and standard deviation of Valence and Arousal separately. Then, we sample a value for Valence and Arousal from a normal distribution, multiply each by their respective standard deviation, and finally add the mean to obtain the final latent space (Figure 4).

Additionally, based on the original VAE loss, we use the mean and standard deviation of each sentence's Valence and Arousal from the dataset as our prior. Through the MSE loss between the mean and standard deviation predicted by the encoder and these priors to ensure that our embedding can produce the correct output (Figure 5).

Through KL annealing, I dynamically adjust the weight of the KL divergence loss term in Variational Autoencoders, enabling smoother optimization and better control over the trade-off between reconstruction fidelity and latent space regularization."

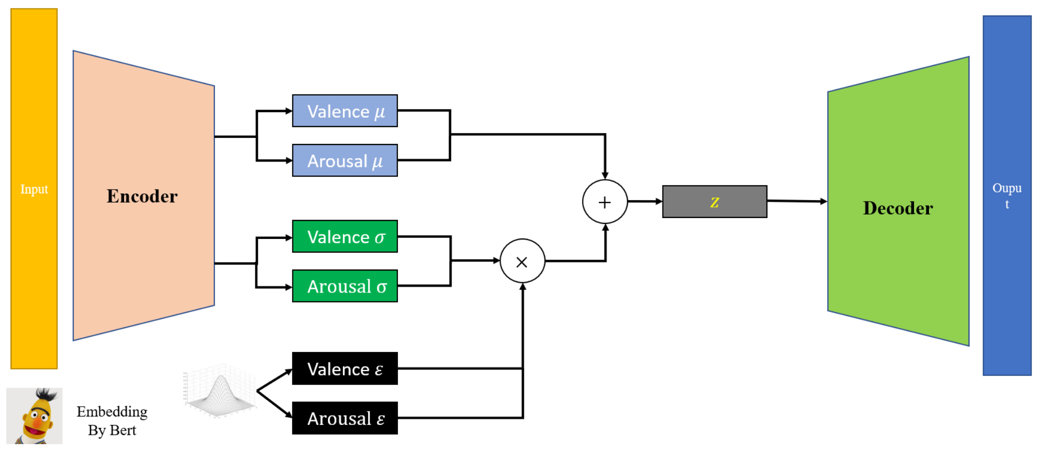


Figure 4 : The architecture of our model

$$\mathcal{L}_{recon} = \sum_i (x_i - \hat{x}_i)^2 \tag{1}$$

$$\mathcal{L}_{KL} = -\frac{1}{2}\sum_{j=1}^{d}\left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right) \tag{2}$$

$$\mathcal{L}_{arousal} = \sum_i (a_i - \hat{a}_i)^2 \tag{3}$$

$$\mathcal{L}_{valence} = \sum_i (v_i - \hat{v}_i)^2 \tag{4}$$

$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \beta\mathcal{L}_{KL} + \mathcal{L}_{\mu} + \mathcal{L}_{\sigma} \tag{5}$$

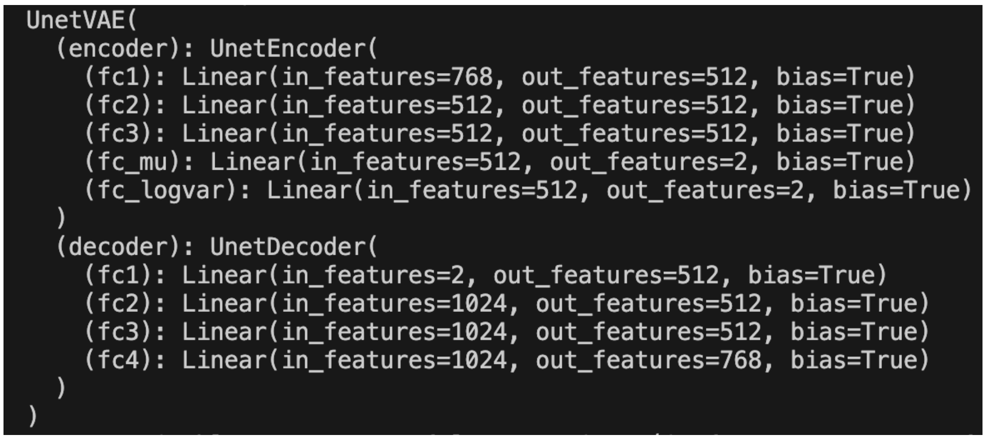
Figure 5

4. Experiment

In the data preprocessing stage, I saved the output of each sentence processed through pre-trained BERT as CSV files. This approach minimizes the time required for obtaining embeddings from BERT during the subsequent training phase.

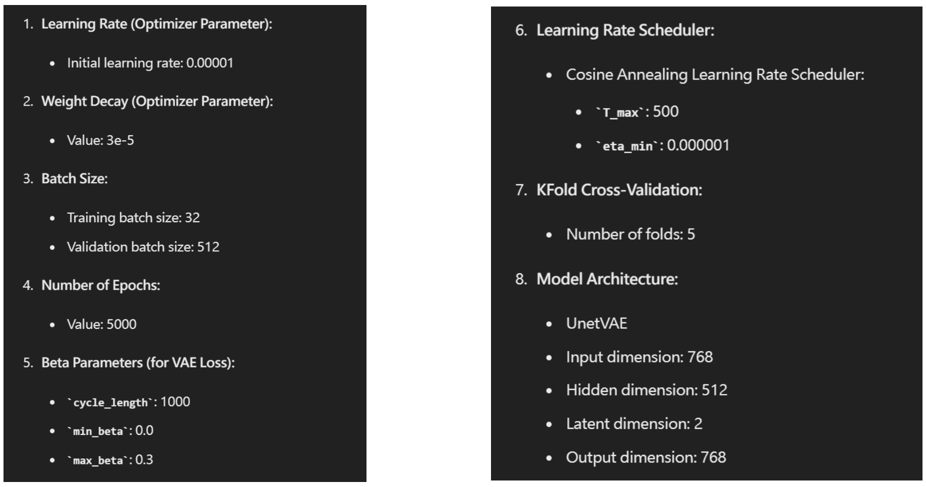
The implementation details of our model (Figure 6), which is designed based on the U-Net architecture. This model leverages the strengths of U-Net, particularly its encoder-decoder structure with skip connections, to effectively process and learn from our data.

Figure 7 provides an overview of the hyperparameter settings employed during model training. Figure 8 presents the trend of the loss during the training of our model. Figure 9 showcases the validation results of our model following 5-fold cross-validation training.



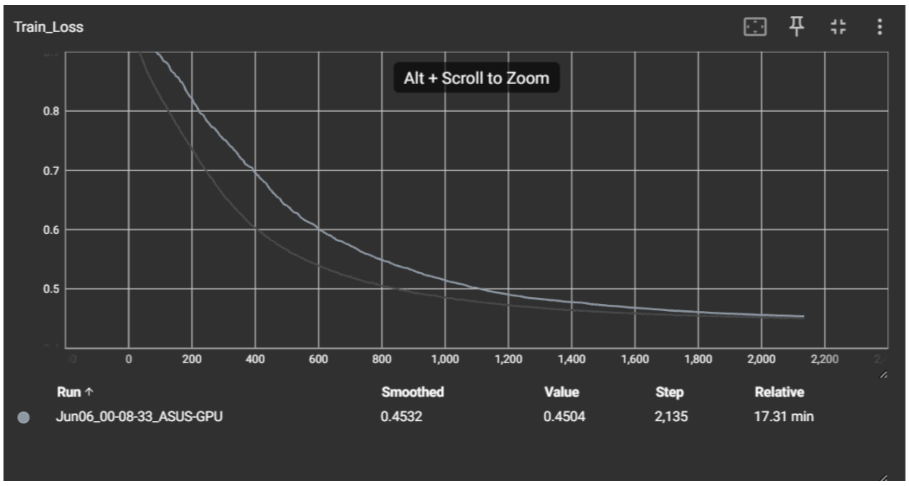
```
UnetVAE(
  (encoder): UnetEncoder(
    (fc1): Linear(in_features=768, out_features=512, bias=True)
    (fc2): Linear(in_features=512, out_features=512, bias=True)
    (fc3): Linear(in_features=512, out_features=512, bias=True)
    (fc_mu): Linear(in_features=512, out_features=2, bias=True)
    (fc_logvar): Linear(in_features=512, out_features=2, bias=True)
  )
  (decoder): UnetDecoder(
    (fc1): Linear(in_features=2, out_features=512, bias=True)
    (fc2): Linear(in_features=1024, out_features=512, bias=True)
    (fc3): Linear(in_features=1024, out_features=512, bias=True)
    (fc4): Linear(in_features=1024, out_features=768, bias=True)
  )
)
```

Figure 6 : The architecture of the U-Net variational autoencoder we designed.



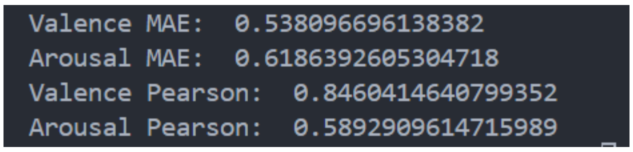
1. **Learning Rate (Optimizer Parameter):**
   - Initial learning rate: 0.00001
2. **Weight Decay (Optimizer Parameter):**
   - Value: 3e-5
3. **Batch Size:**
   - Training batch size: 32
   - Validation batch size: 512
4. **Number of Epochs:**
   - Value: 5000
5. **Beta Parameters (for VAE Loss):**
   - `` `cycle_length` ``: 1000
   - `` `min_beta` ``: 0.0
   - `` `max_beta` ``: 0.3
6. **Learning Rate Scheduler:**
   - Cosine Annealing Learning Rate Scheduler:
     - `` `T_max` ``: 500
     - `` `eta_min` ``: 0.000001
7. **KFold Cross-Validation:**
   - Number of folds: 5
8. **Model Architecture:**
   - UnetVAE
   - Input dimension: 768
   - Hidden dimension: 512
   - Latent dimension: 2
   - Output dimension: 768

Figure 7 : Hyperparameter config



```
Run ↑                    Smoothed   Value    Step    Relative
Jun06_00-08-33_ASUS-GPU  0.4532     0.4504   2,135   17.31 min
```

Figure 8 : This plot shows how the loss changes during training. The x-axis represents the number of training iterations or epochs, while the y-axis indicates the loss value.



```
Valence MAE:  0.538096696138382
Arousal MAE:  0.6186392605304718
Valence Pearson:  0.8460414640799352
Arousal Pearson:  0.5892909614715989
```

Figure 9 : Validation results

5. Conclusion

In this implementation, I did not perform data cleaning, which could result in some sentences with larger standard deviations. When sampling each sentence's latent space in the VAE, those with larger standard deviations might overlap with the latent space of other sentences, causing the model to become confused and uncertain about the predictions.

I did not use the Category information for each sentence from the dataset. I think incorporating this information for each sentence, similar to the CVAE architecture, by adding extra features based on these Categories (as shown in Figure 10), could potentially enhance the model's performance.

The simple Gaussian assumption is too simplistic: Instead of assuming a simple Gaussian distribution for the latent variables, maybe we can explore the utilization of Gaussian Mixture Models, leveraging the mean and standard deviation derived from each sentence in the dataset, to model a more intricate distribution of the latent space.

Unable to capture important features from the input data : Perhaps we can enhance the model's capability by incorporating an infoVAE, which introduces an additional term based on the mutual information between the latent variables and the input data. This approach encourages the model to learn latent representations that better retain information from the input data.

In this implementation, we did not fine-tune BERT. Fine-tuning BERT typically requires significant computational resources and time. However, even without fine-tuning, we could still leverage BERT's ability to extract features from text. Moving forward, we might explore fine-tuning BERT as a means to further enhance our model's performance. This approach could potentially improve the model's adaptability to specific tasks and deepen its understanding of our data.

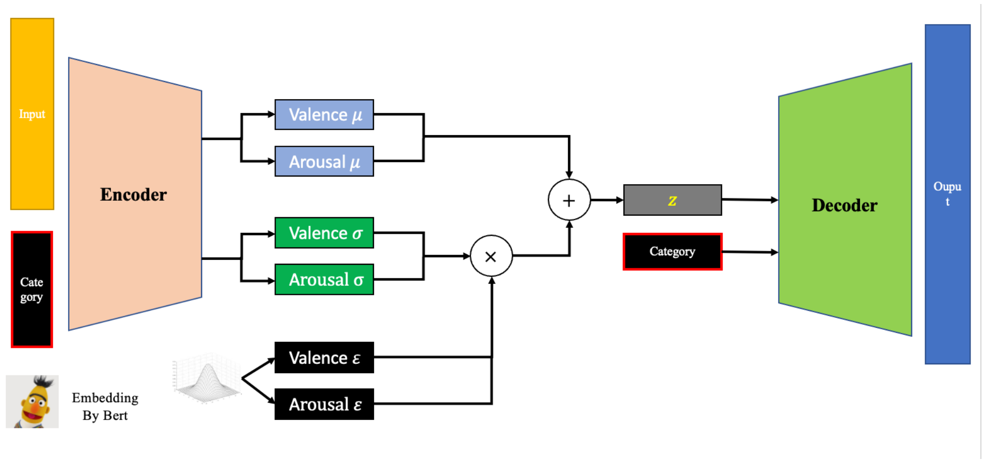


Figure 10 : The improved version based on CVAE.

6. Reference

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*. https://doi.org/10.48550/arXiv.1810.04805
- Kingma, D.P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*. https://doi.org/10.48550/arXiv.1312.6114
- Wu, C., Wu, F., Wu, S., Yuan, Z., Liu, J., & Huang, Y. (2018). Semi-supervised dimensional sentiment analysis with variational autoencoder. *Knowledge-Based Systems*, 164, 206-215. https://doi.org/10.1016/j.knosys.2018.11.018

- Zhao, S., Song, J., & Ermon, S. (2017). InfoVAE: Information Maximizing Variational Autoencoders. *arXiv preprint arXiv:1706.02262*. https://doi.org/10.48550/arXiv.1706.02262