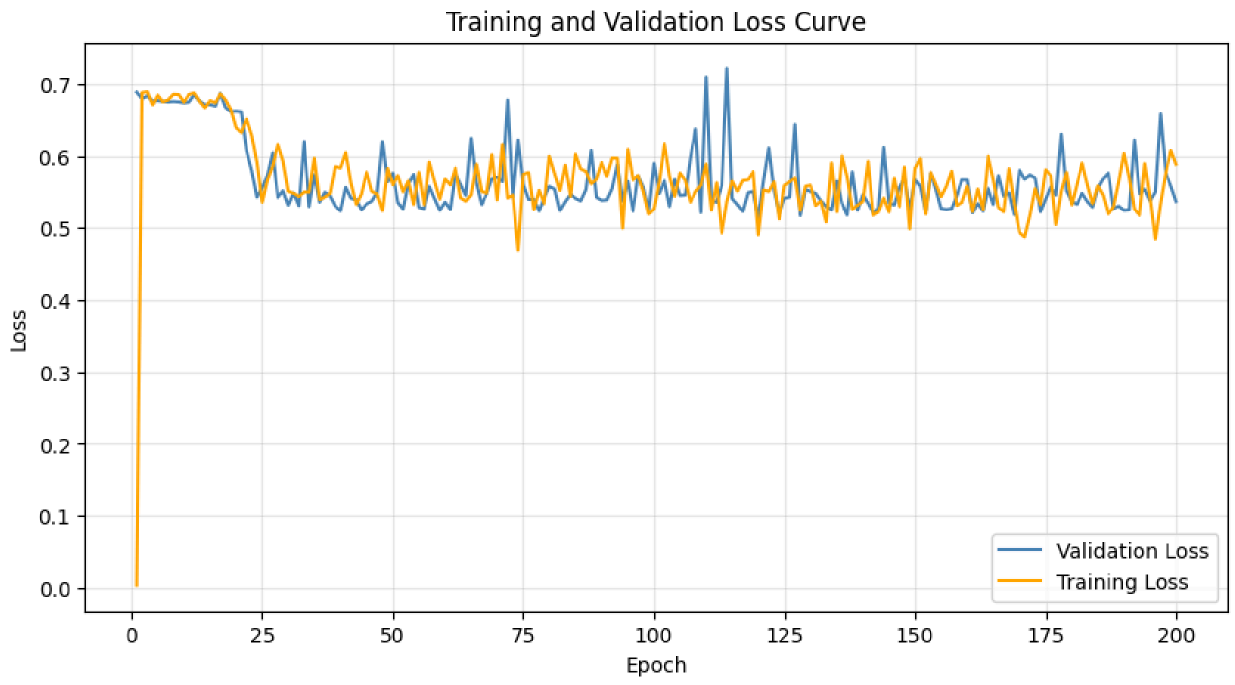
CIS530 HW6 Writeup Template



This training and validation loss curve demonstrates that the model quickly stabilized after an initial steep drop in loss within the first few epochs, indicating rapid early learning. Both the training and validation losses converge toward an average around 0.55, maintaining close proximity throughout the remaining epochs—suggesting minimal overfitting and consistent generalization across datasets. The oscillations in both curves are relatively small and follow similar patterns, reflecting stable optimization behavior. Overall, the model exhibits solid convergence, achieving a good balance between fitting the training data and maintaining performance on the validation set, which supports its reliability for downstream evaluation.

Neural Language Models: City Classification Experiments

Task 2.1: Learning Rate Tuning (6 points)

Experimental Setup

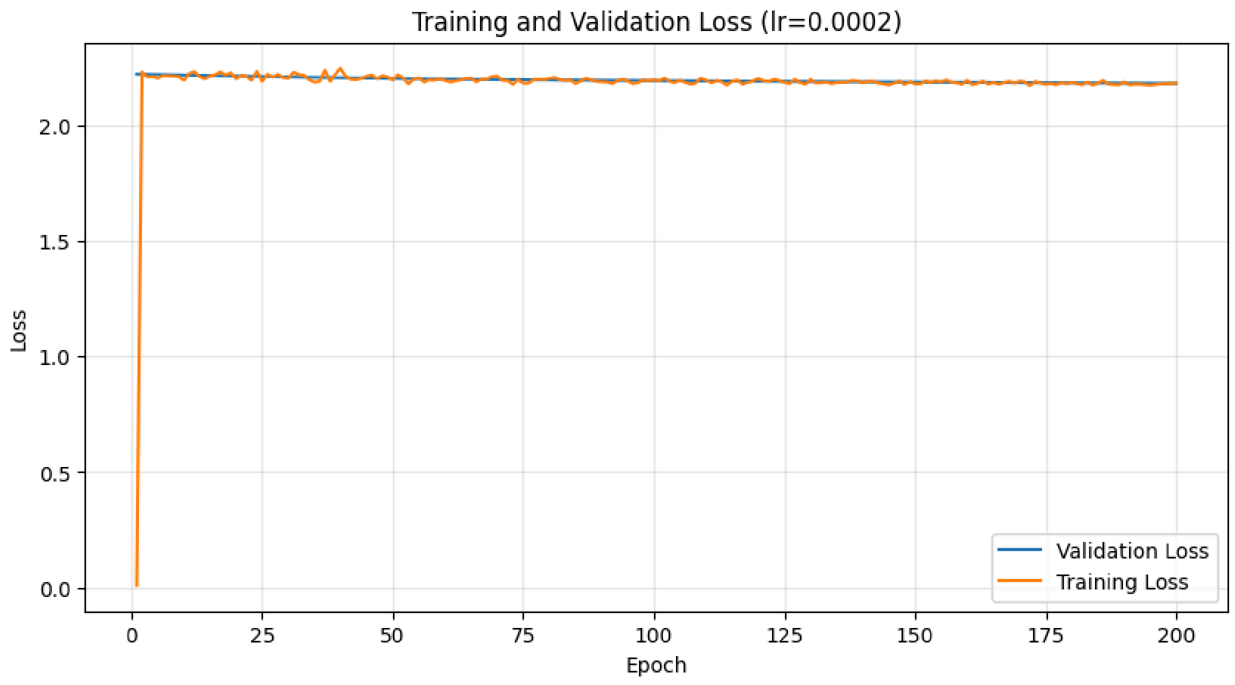
I trained RNN models for city classification using different learning rates while keeping other hyperparameters constant:

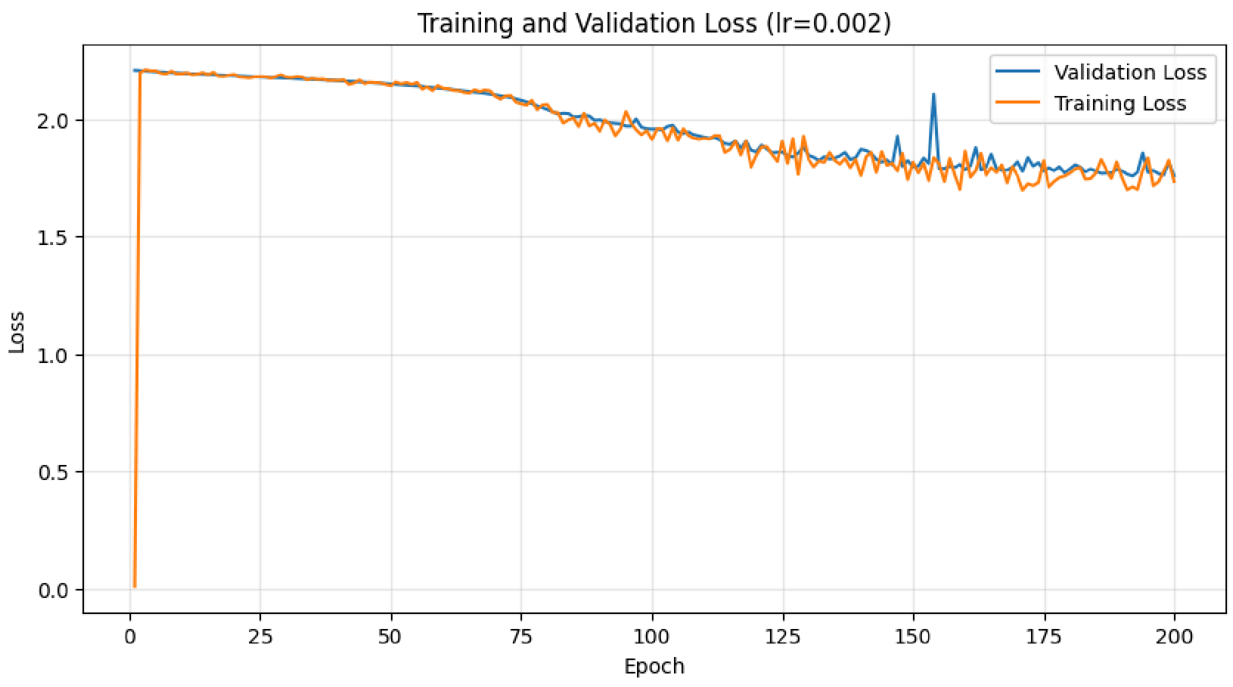
- **Base hyperparameters**: hidden_size = 10, n_epochs = 50000
- **Learning rates tested**: {0.0002, 0.002, 0.02}

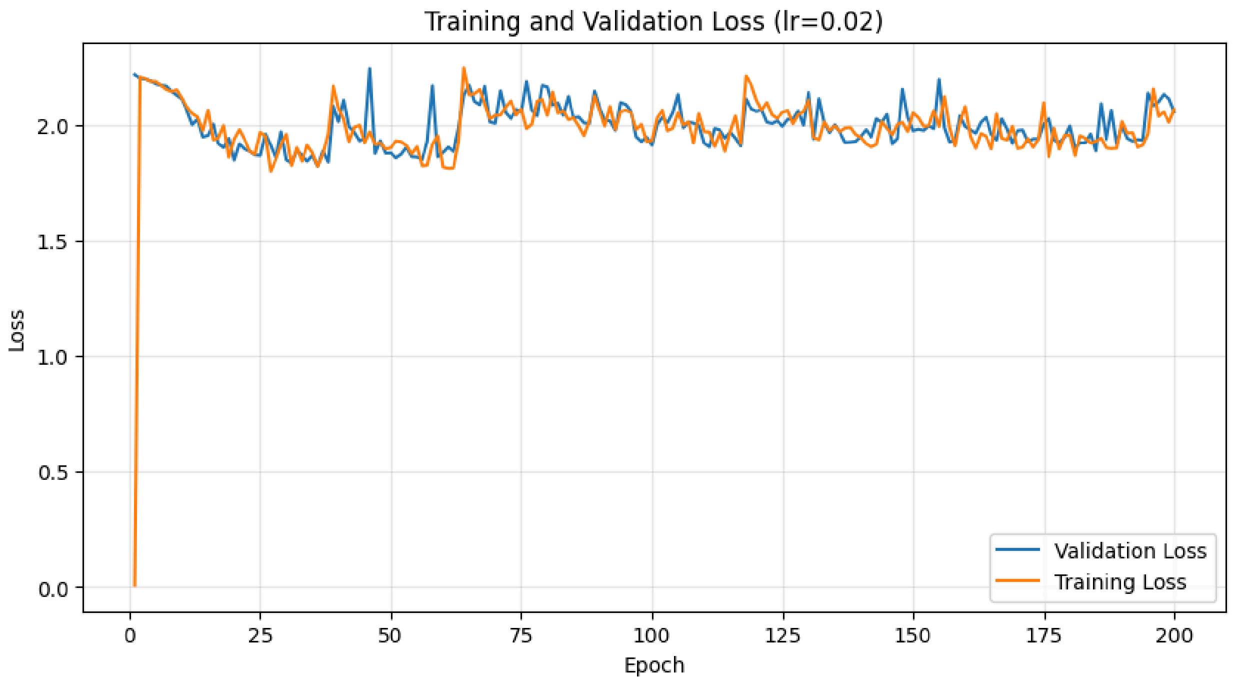
Results

Learning Rate	Validation Accuracy
0.0002	0.1778
0.002	.3289
0.02	.1900

Training and Validation Loss Plots







Observations and Discussion

The model's performance was not statistically significant. At around 75 epochs, the model seems to make a large leap in performance from 2.2 loss to 1.8 loss. It continues to decline toward 1.7 for the remaining epochs. Ultimately, we expect to see more improvement as we continue to modify other components of the system.

Best learning rate for subsequent experiments: 0.002 Learning Rate

---

Task 2.2: Hidden Layer Size Tuning (6 points)

Experimental Setup

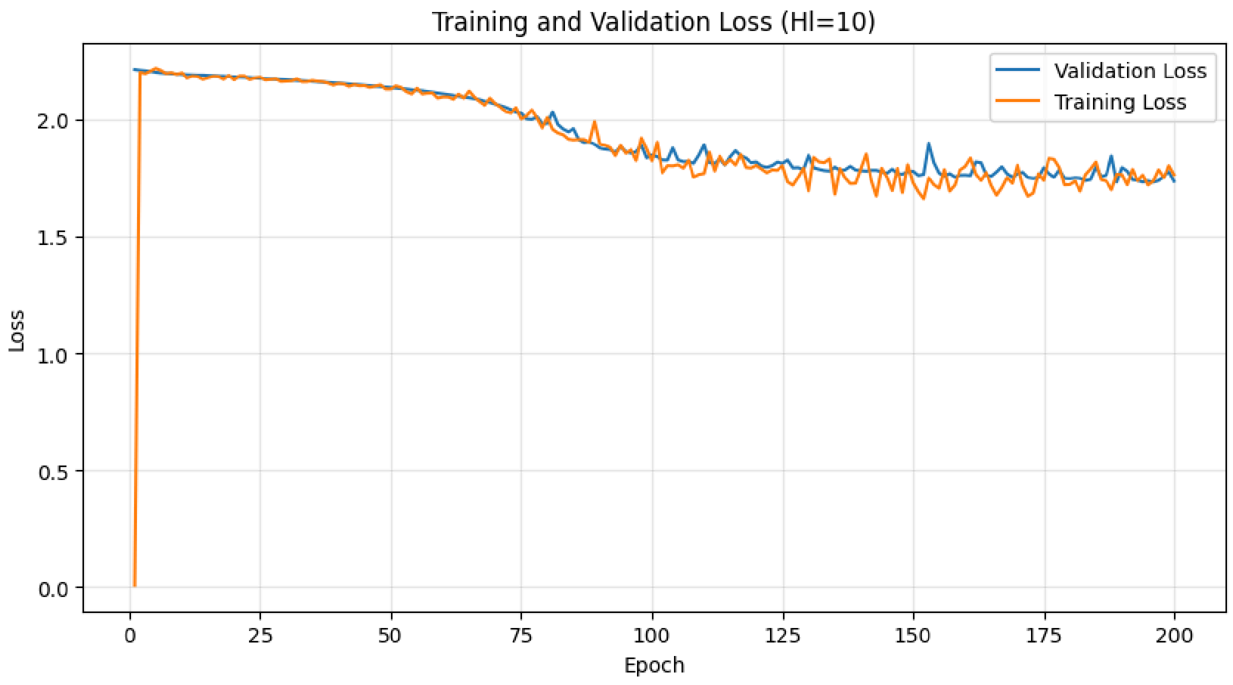
Using the best learning rate from Task 2.1, I experimented with different hidden layer sizes:

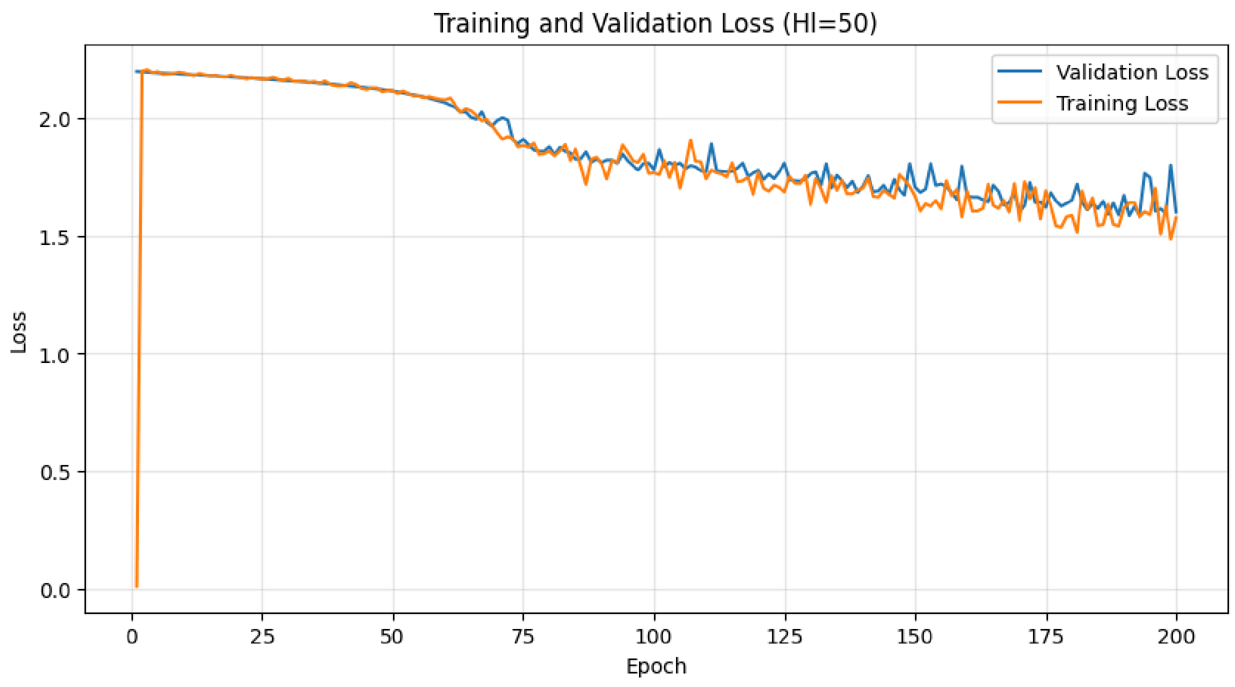
- **Fixed hyperparameters**: learning_rate = [0.02], n_epochs = 50000
- **Hidden sizes tested**: {10, 50, 100}

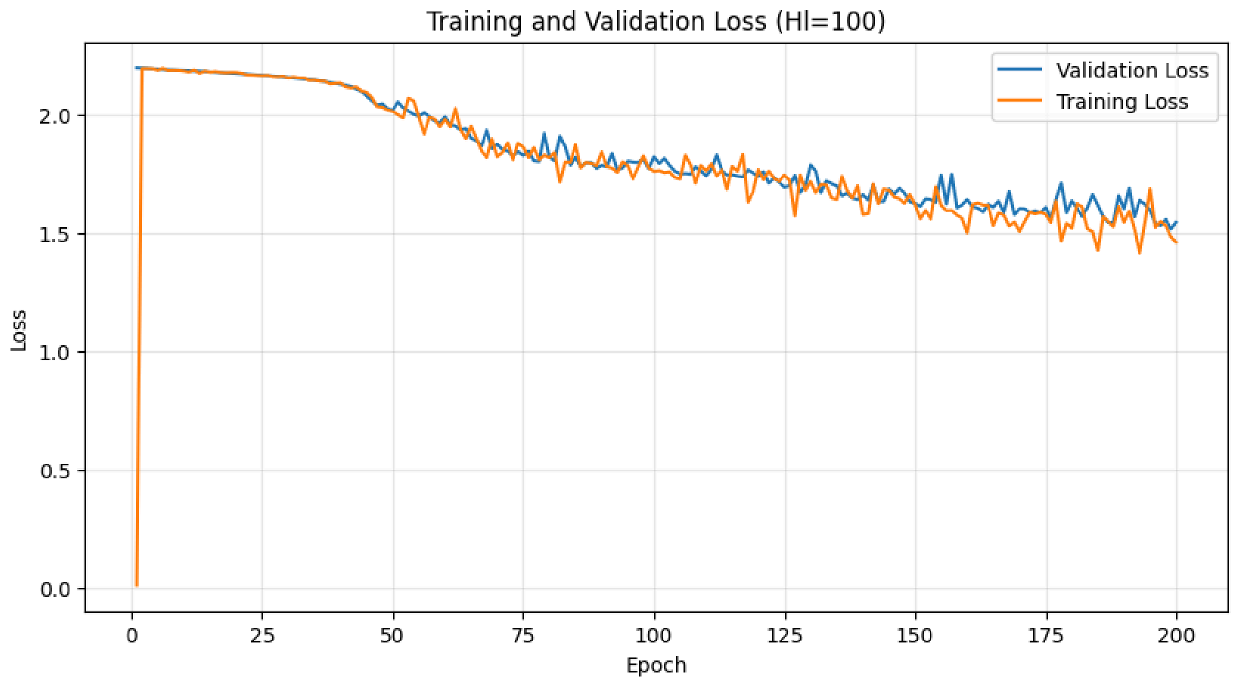
Results

Hidden Size	Validation Accuracy
10	0.3556
50	0.3955
100	0.4178

Training and Validation Loss Plots







Observations and Discussion

Using different hidden layer sizes 10, 50, and 100 directly influences how much representational capacity the RNN has to capture complex patterns in the data. A smaller hidden layer (10 units) limits the model's ability to learn long-range dependencies and nuanced relationships, often resulting in underfitting. A mid-sized layer (50 units) offers a balance between expressive power and generalization, improving performance while keeping computational costs reasonable. A larger hidden layer (100 units) gives the network substantially more parameters and internal memory, allowing it to model more intricate temporal structures and subtle linguistic cues. However, this also increases the risk of overfitting and training time. In this experiment, the model with 100 hidden units achieved the highest validation accuracy, suggesting that the additional capacity helped it better capture the underlying structure of the sequence data without overfitting given the dataset size.

Best hidden size for subsequent experiments: 100 hidden layers.

Task 2.3: Epoch Tuning (6 points)

Experimental Setup

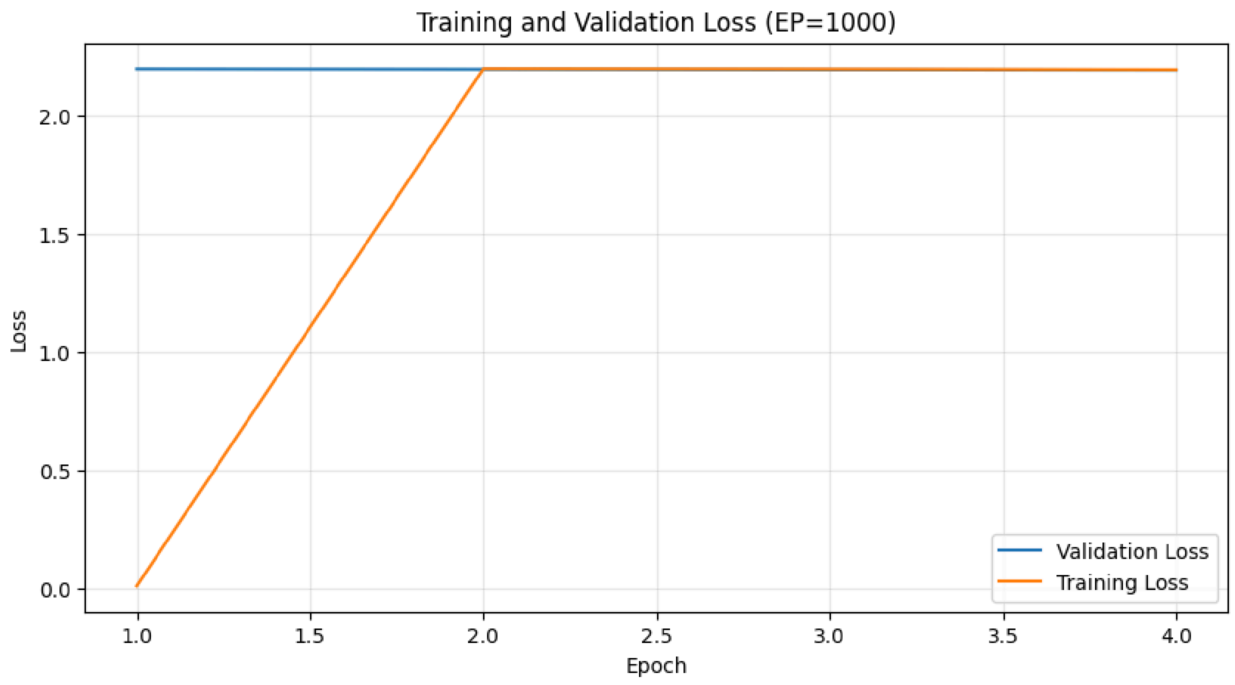
Using the best learning rate and hidden size from previous tasks, I tested different training durations:

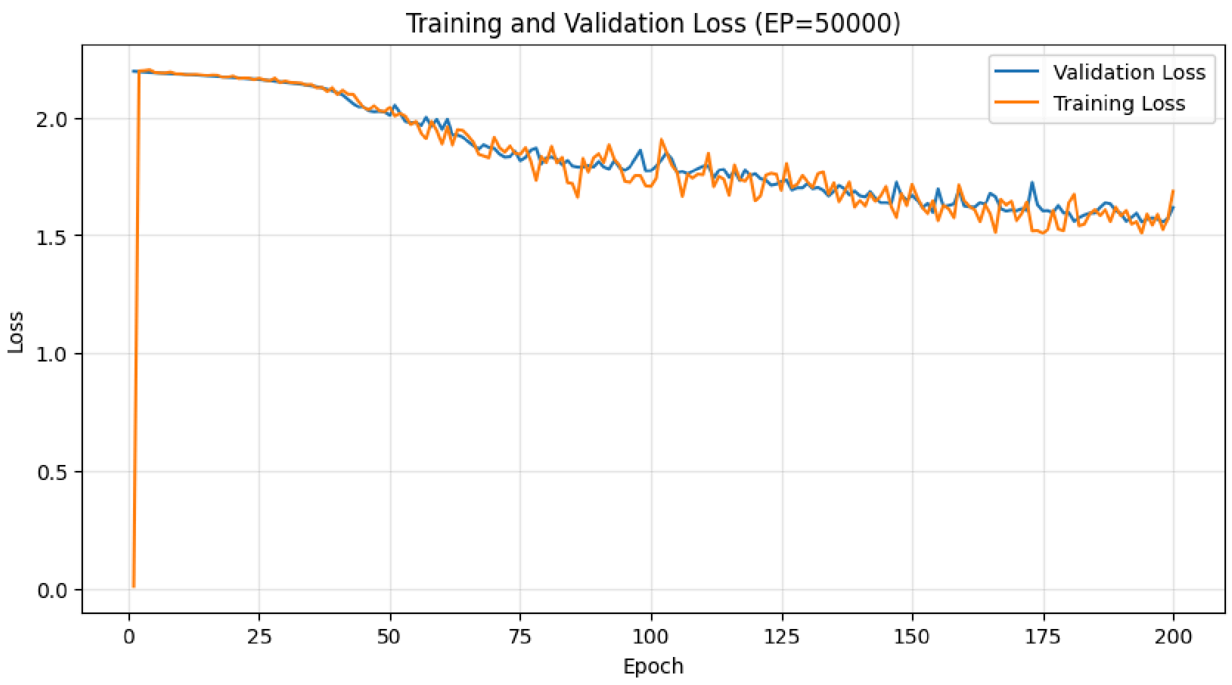
- **Fixed hyperparameters**: learning_rate = .002, hidden_size = 100
- **Epochs tested**: {1000, 50000, 100000}

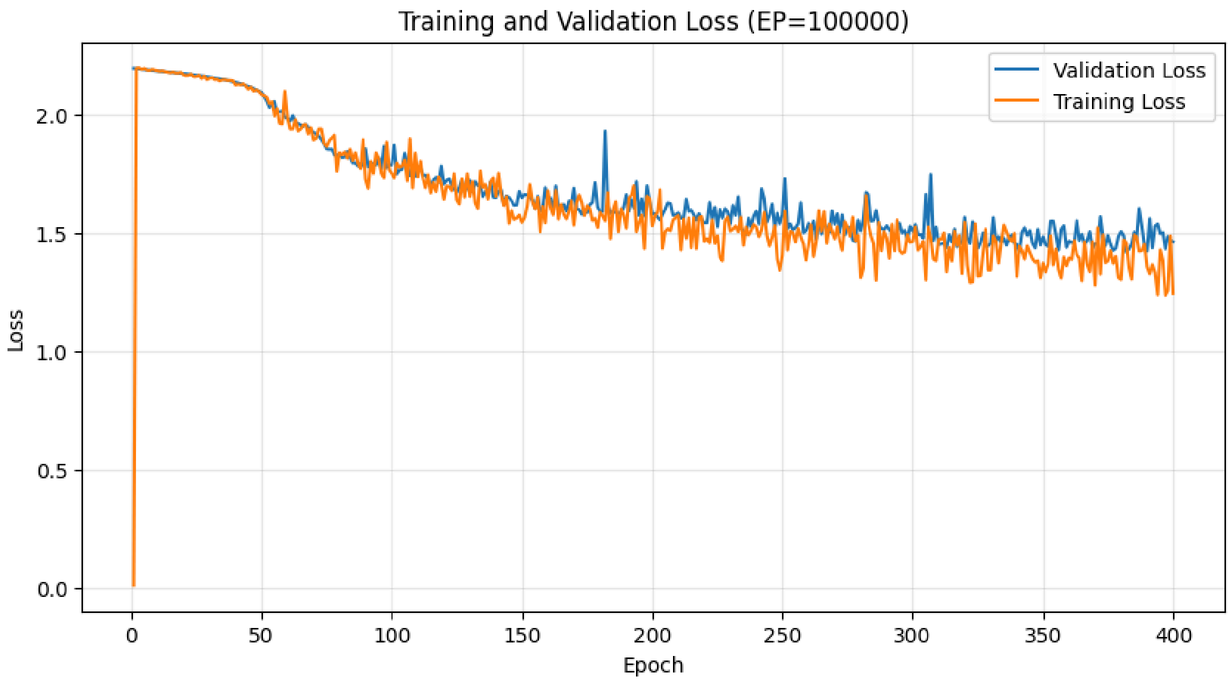
Results

Number of Epochs	Validation Accuracy
1000	0.1256
50000	0.4111
100000	0.4856

Training and Validation Loss Plots







Observations and Discussion

Increasing the number of training epochs from 1,000 to 50,000 and eventually to 100,000 allowed the model to make more complete use of the training data and gradually refine its internal weight representations, leading to a clear positive trend in validation accuracy. Early in training, the model is still learning basic patterns and associations, so performance is limited. As epochs increase, the network continues adjusting its parameters through repeated exposure to the data, reducing both training and validation loss as it converges toward a more optimal set of weights. This extended training enables the RNN to better capture complex temporal dependencies and improve its generalization on unseen validation data. The observed increase in validation accuracy across epochs indicates that the model benefited from the additional optimization time, with each phase of training uncovering progressively more stable and accurate feature representations before reaching its peak performance near 100,000 epochs.

Optimal number of epochs: 100,000. The more optimal training the better the model will perform.

Task 2.4: Leaderboard Competition (Optional, 5 bonus points)

Final Model Configuration

Based on my hyperparameter tuning experiments, my best model uses:

- **Learning rate**: .00065
- **Hidden size**: 128
- **Number of epochs**: 90,000
- **Additional modifications**:
    - LSMT
    - Embedding - Dimensionality of character embeddings - 96
    - Dropout - .3

Final Performance

- **Validation Accuracy**: .7266 (Gradescope)

Additional Experiments Log

[Optional: Include a table or description of other experiments you tried]

Experiment Description	**Hyperparameters**	**Validation Accuracy**	**Notes**
Base Measurement Size	0.0005, 128, 50,000	.6812	Modifying the batch size, hidden layer size, enabling bidirectional support, choosing LSMT.
Performance Validation	0.0005, 128, 90,000	.71	Significant improvement in validation accuracy. The entended time spent validating proved to be beneficial.
...	...	...	...

Discussion

In this experiment, I worked to implement new models and to observe how extended optimization affected model performance in a neural language classification task. Beginning with a more basic model, the model exhibited limited learning capacity, showing relatively low validation accuracy and signs of underfitting. As I created new constructors for improved models and increased epoch size from 50,000 to 90,000, the model demonstrated a clear improvement in its ability to capture meaningful linguistic patterns, with validation accuracy rising steadily. This confirmed that additional training iterations and slow but more effective models allowed the network to refine its internal representations and better generalize to unseen data.

Summary and Conclusions

- The key components we saw within this experiment were epoch size and learning rates. Without a substantial number of epochs, the model doesn't have time to properly learn from the data set. Additionally, if the model has a learning rate that isn't tuned properly, it cannot achieve the proper gradient when learning.
- It seems that by trading off hidden layers with epochs, we can find a balance between the components and allow our model to make statistical predictions with improved accuracy.
- With a proper gold label data set it's quite possible to tune a machine to accomplish a very high validation accuracy rating.
- Referencing sources like PyTorch for constructors of more advanced neural nets became required to build models that passed .70 accuracy. It became necessary to research them all as each model iteration improved upon the last.