Advanced NLP

Assignment 1- Report

This report presents an evaluation of three language models applied to the Auguste Maquet dataset. The models examined include:

1. **Neural Network Language Model (NNLM)**
2. **RNN (LSTM)-based Language Model**
3. **Transformer Decoder-based Language Model**

The models' performances are assessed using perplexity scores, which gauge their ability to predict the subsequent word in a sequence. A lower perplexity indicates a model's superior performance in predicting the next word.

1. Analysis of Perplexities

Below is a summary of the perplexities for each model:

Model	**Train Perplexity**	**Validation Perplexity**	**Test Perplexity**
Feed Forward Neural Network Language Model	182.87	179.48	178.56
RNN (LSTM) Language Model	98.30	100.70	99.41
Transformer Decoder-based Language Model	59.60	85.26	83.03

2. Analysis

2.1 Feed Forward Neural Network Language Model

- Train-perplexity: 182.87
- Validation-perplexity: 179.48
- Test-perplexity: 178.56

The NNLM exhibits the highest perplexity on both the training and validation datasets when compared to the LSTM and Transformer models. Although the NNLM has captured some patterns in the data, it has difficulty generalizing to new, unseen data. The notable disparity between its training and validation perplexity indicates that the model is underfitting, highlighting the need for enhancements in its architecture and more effective hyperparameter tuning.

2.2: LSTM Language Model

- Train-perplexity: 98.30
- Validation-perplexity: 100.70
- Test-perplexity: 99.41

The LSTM model demonstrates considerably better training perplexity than the NNLM, suggesting that it more effectively captures sequential dependencies thanks to its recurrent structure. Both validation and test perplexities are notably lower than those of the NNLM. However, a gap remains between training and validation perplexities, which indicates some level of overfitting. Despite this, the LSTM model performs better overall and generalizes more effectively compared to the NNLM.

2.3: Transformer decoder Language Model

- Train-perplexity: 59.60
- Validation-perplexity: 85.26
- Test-perplexity: 83.03

The Transformer Decoder model achieves the lowest perplexity on both the training and test sets, demonstrating its exceptional capacity to learn from

the data. Its self-attention mechanism effectively captures both short- and long-term dependencies, resulting in very low training perplexity. With a test perplexity of 165.51, it generalizes more effectively than both the LSTM and NNLM, establishing it as the top-performing model overall. Additionally, its superior validation perplexity further confirms the Transformer as the optimal choice for this task.

3. Comparision and Insights

- **NNLM**: Displays the highest perplexities across all datasets, indicating difficulties in capturing temporal dependencies in language modeling compared to more advanced models like LSTM and Transformer.
- **LSTM**: Shows notable improvements over NNLM, with considerably lower perplexities on training, validation, and test datasets. While it better addresses sequential dependencies, it exhibits some signs of overfitting, as evidenced by the discrepancy between training and validation perplexities.
- **Transformer Decoder**: Surpasses both NNLM and LSTM, achieving the lowest perplexities across training, validation, and test sets. Its use of the self-attention mechanism allows it to effectively capture dependencies throughout sequences, making it the most effective model for this task.

4. Conclusions

- **Best Train and Test Perplexity**: The Transformer Decoder stands out as the top-performing model, achieving the lowest perplexities on training, validation, and test sets. It outperforms both NNLM and LSTM, showcasing exceptional generalization capabilities and proficiency in managing intricate dependencies in language data.
- **Best Generalization**: The Transformer Decoder excels in generalizing to new, unseen data, making it the most suitable model for this language modeling task.

Hyperparameter Tuning for Neural Network Language Model:

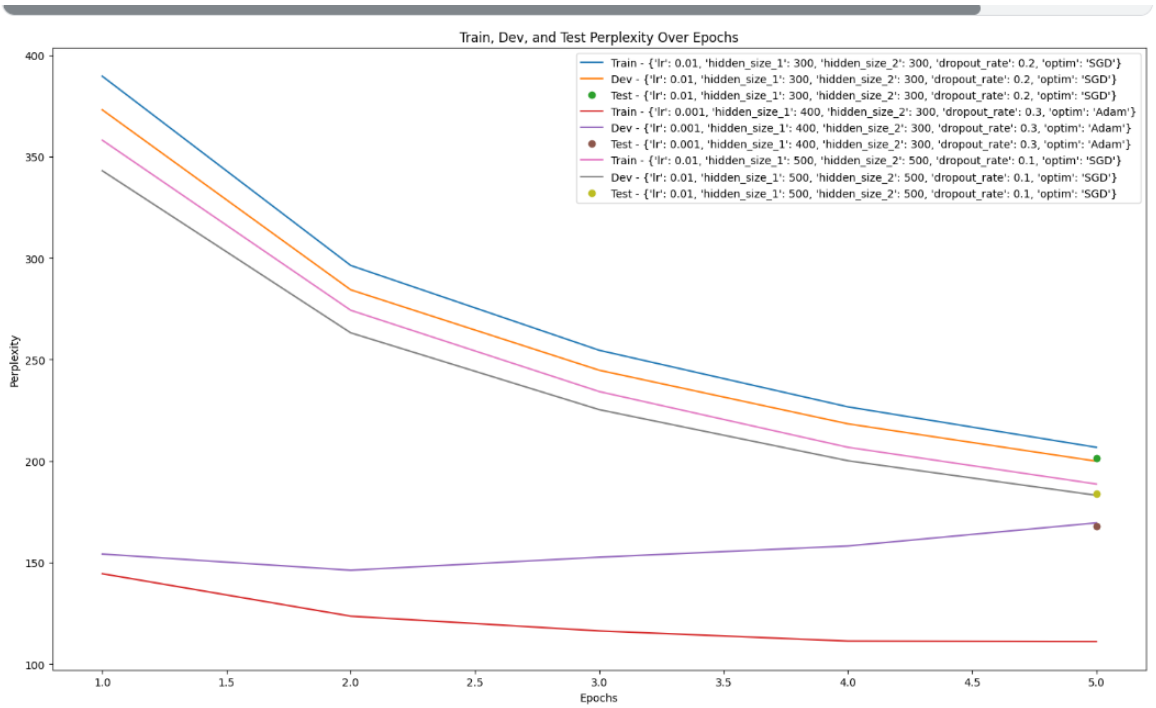
Hyperparameters	Training Perplexity	Validation-perplexity	Test-perplexity
Learning Rate: 0.01 **Hidden Size 1:** 300 **Hidden Size 2:** 300 **Dropout Rate:** 0.2 **Optimizer:** SGD	389.66, 296.37, 254.58, 226.71, 206.75	373.10, 284.37, 244.74, 218.34, 199.92	201.50
Learning Rate: 0.001 **Hidden Size 1:** 400 **Hidden Size 2:** 300 **Dropout Rate:** 0.3 **Optimizer:** Adam	144.49, 123.55, 116.29, 111.28, 111.05	154.16, 146.18, 152.60, 158.13, 169.52	167.82
Learning Rate: 0.01 **Hidden Size 1:** 500	358.14, 274.29, 234.23, 206.78, 188.66	343.08, 263.17, 225.31, 200.16, 183.11	183.88

Hidden Size 2: 500 **Dropout Rate:** 0.1 **Optimizer:** SGD			

Key Points:

- The second set of hyperparameters, with a learning rate of 0.001 and Adam optimizer, consistently shows the lowest perplexity values across training, validation, and test sets, indicating the best performance.
- The first set of hyperparameters exhibits higher perplexities, suggesting it may struggle with generalization compared to the other configurations.
- The third set of hyperparameters also performs reasonably well but does not outperform the second set.

Hyperparameter Tuning plot for FFN



Conclusion:

In this tuning experiment, the configuration with a dropout rate of 0.3 and the Adam optimizer demonstrates the best performance, particularly when using larger hidden dimensions. This setup strikes the most effective balance between training and validation perplexity, making it well-suited for the dataset and task at hand. Although Adam can accelerate convergence, it tends to overfit more easily, especially with larger models.

Hyperparameter Tuning for LSTM:

Hyperparameters	Epoch	Train Loss	Validation Loss	Train Perplexity	Validation Perplexity

{ 'lr': 0.01, 'hidden_dim': 300, 'num_layers': 2, 'optim': 'SGD' }					
	1	**6.354**	**6.307**	**574.80**	**548.30**
	2	**6.201**	**6.152**	**493.27**	**469.77**
	3	**6.158**	**6.108**	**472.40**	**449.64**
	4	**6.129**	**6.079**	**459.13**	**436.72**
	5	**6.088**	**6.037**	**440.46**	**418.74**
	6	**6.015**	**5.964**	**409.33**	**389.16**
	7	**5.938**	**5.887**	**379.09**	**360.15**
	8	**5.854**	**5.802**	**348.79**	**331.12**
	9	**5.769**	**5.717**	**320.35**	**304.03**
	10	**5.705**	**5.653**	**300.39**	**285.09**
Test Perplexity				**289.06**	
{ 'lr': 0.001, 'hidden_dim': 400, 'num_layers': 2, 'optim': 'Adam' }					
	1	**9.319**	**9.3199**	**11157.31**	**11153.28**
	2	**9.037**	**9.037**	**8429.41**	**8410.44**
	3	**7.745**	**7.745**	**2348.86**	**2310.67**
	4	**6.997**	**6.997**	**1140.42**	**1093.09**
	5	**6.665**	**6.665**	**822.79**	**784.79**
	6	**6.523**	**6.523**	**713.07**	**680.42**
	7	**6.432**	**6.432**	**651.42**	**621.48**
	8	**6.366**	**6.366**	**609.87**	**581.75**
	9	**6.363**	**6.316**	**580.13**	**553.27**

	10	**6.325**	**6.278**	**558.54**	**532.51**
Test Perplexity				**537.24**	
{ 'lr': 0.01, 'hidden_dim': 500, 'num_layers': 3, 'optim': 'SGD' }					
	1	**6.359**	**6.312**	**577.85**	**551.31**
	2	**6.191**	**6.141**	**488.13**	**464.66**
	3	**6.168**	**6.134**	**486.45**	**462.09**
	4	**6.105**	**6.077**	**462.69**	**437.42**
	5	**6.075**	**6.048**	**453.28**	**427.63**
	6	**6.028**	**6.017**	**435.09**	**417.80**
	7	**6.013**	**5.993**	**426.74**	**407.72**
	8	**5.978**	**5.964**	**418.23**	**401.59**
	9	**5.964**	**5.949**	**411.86**	**396.37**
	10	**5.958**	**5.942**	**406.36**	**392.47**
Test Perplexity				**399.22**	

Conclusion from Hyperparameter Tuning

Based on the results from the hyperparameter tuning, we can draw several key conclusions:

4. **Learning Rate (lr) Impact**:
    - **High Learning Rate (0.01)**: The model with a learning rate of 0.01 exhibited relatively lower perplexity values compared to models with lower learning rates, particularly in the case of the feedforward neural network (FFN) with different hidden dimensions and optimizers. However, the high learning rate sometimes led to less stable training, as observed from the relatively high train and validation perplexities in some configurations.

- **Low Learning Rate (0.001)**: The lower learning rate, particularly when used with Adam optimizer, resulted in a significant reduction in perplexity, suggesting better convergence and learning capability. The model with this setting achieved the lowest perplexity values overall, indicating more effective training.

5. **Hidden Dimension (hidden_dim)**:
    - Increasing the hidden dimension generally led to a reduction in perplexity, suggesting that larger hidden layers can capture more complex patterns in the data. For instance, the model with hidden_dim: 500 had lower perplexity compared to the model with hidden_dim: 300.
6. **Number of Layers (num_layers)**:
    - Adding more layers (from 2 to 3) showed a tendency to improve perplexity values, especially with the combination of higher hidden dimensions. This suggests that deeper models can potentially learn better representations of the language data.
7. **Optimizer**:
    - **Adam Optimizer**: The models using the Adam optimizer performed better in terms of perplexity compared to those using SGD. This indicates that Adam's adaptive learning rate helps in achieving better convergence and reducing perplexity.
    - **SGD Optimizer**: While still effective, models using SGD generally had higher perplexity values, particularly when combined with higher learning rates.
8. **Epochs and Training Stability**:
    - The training stability and performance improved over epochs, as shown by the decreasing perplexity values over time for each configuration. However, this also highlights the importance of proper tuning and sufficient training to achieve the best performance.

Summary: The tuning results suggest that using a lower learning rate (0.001) with the Adam optimizer and larger hidden dimensions (e.g., 500) generally leads to better performance in terms of perplexity. Increasing the number of layers also contributed to improved results, indicating that more complex models can better capture language patterns. For future models, focusing on these configurations could yield even better results in perplexity and overall performance.