INLP Assignment-2 Report

Chekkapalli Naveen, 2021101025

Hyperparameter tuning:

FFNN Hyperparameter Tuning

Embedding Dimension	Hidden Dimension	prev context size	succesive context size	Activation	Accuracy
100	50	1	1	'relu'	96.58491863396745
100	50	1	1	'tanh'	97.61631904652762
100	100	1	1	'relu'	97.11207884483154
100	100	1	1	'tanh'	97.22667889067155
200	50	1	1	'relu'	97.13499885399955
200	50	1	1	'tanh'	96.03483841393536
200	100	1	1	'relu'	96.63075865230346
200	100	1	1	'tanh'	96.65367866147146
100	50	0	0	'tanh'	95.89102950030103
100	50	1	1	'tanh'	97.8731139792765
100	50	2	2	'tanh'	97.38711895484758
100	50	3	3	'tanh'	96.45909645909646
100	50	4	4	'tanh'	95.42454905411351

The top 3 best accuracy metrics from the FFNN Hyperparameter Tuning table are:

1. Embedding Dimension: 100, Hidden Dimension: 50, p: 1, s: 1, Activation: 'tanh', Accuracy: 97.8731139792765
2. Embedding Dimension: 100, Hidden Dimension: 50, p: 1, s: 1, Activation: 'tanh', Accuracy: 97.61631904652762
3. Embedding Dimension: 100, Hidden Dimension: 100, p: 1, s: 1, Activation: 'tanh', Accuracy: 97.22667889067155

LSTM Hyperparameter Tuning

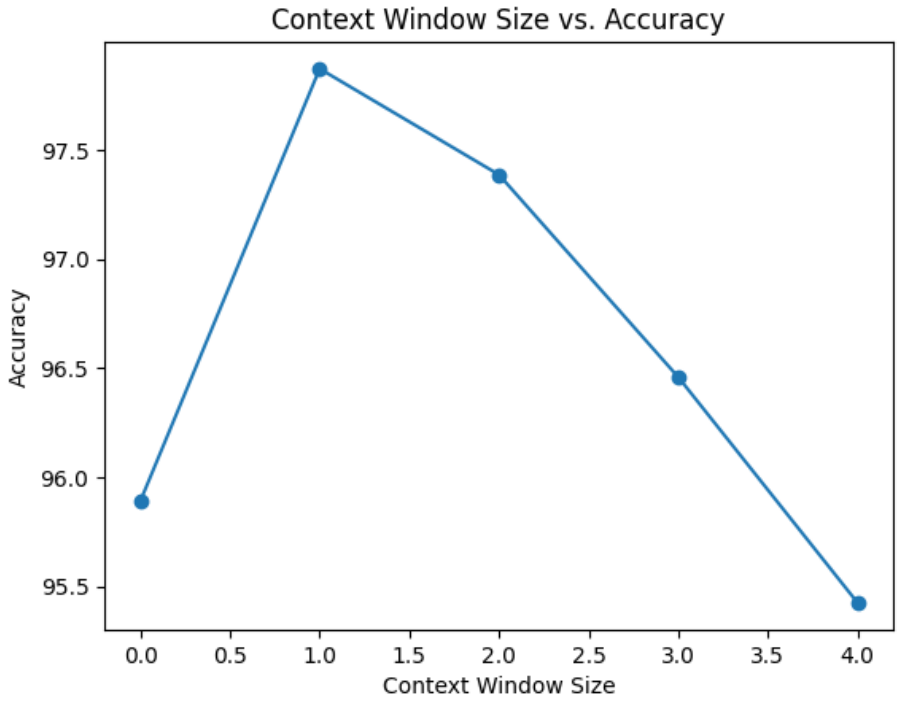
Input Dimension	Hidden Dimension	Num Layers	Activation	Accuracy
100	50	1	'relu'	96.8543046357616
100	50	1	'tanh'	97.17037928958459
100	50	2	'relu'	97.12522576760988
100	50	2	'tanh'	97.17037928958459
100	100	1	'relu'	97.20048163756773
100	100	1	'tanh'	97.1101745936183
100	100	2	'relu'	97.03491872366044

Input Dimension	Hidden Dimension	Num Layers	Activation	Accuracy
100	100	2	'tanh'	96.95966285370258
200	50	1	'relu'	97.3208910295003
200	50	1	'tanh'	97.00481637567731
200	50	2	'relu'	97.17037928958459
200	50	2	'tanh'	96.71884406983744
200	100	1	'relu'	97.26068633353401
200	100	1	'tanh'	96.44792293798916
200	100	2	'relu'	97.24563515954245
200	100	2	'tanh'	96.41782059000602

The top 3 best accuracy metrics from the LSTM Hyperparameter Tuning table are:

1. Input Dimension: 200, Hidden Dimension: 50, Num Layers: 1, Activation: 'relu', Accuracy: 97.3208910295003
2. Input Dimension: 200, Hidden Dimension: 100, Num Layers: 1, Activation: 'relu', Accuracy: 97.26068633353401
3. Input Dimension: 200, Hidden Dimension: 100, Num Layers: 2, Activation: 'relu', Accuracy: 97.24563515954245

Plot context vs accuracy for FFNN:



Based on the provided data for the bottom five configurations with varying context sizes in FFNN, here's a focused analysis on how the context size affects the accuracy:

1. **Variation in Context Size**:

- The configurations with varying context sizes are specified as (100, 50, 0, 0, 'tanh'), (100, 50, 1, 1, 'tanh'), (100, 50, 2, 2, 'tanh'), (100, 50, 3, 3, 'tanh'), and (100, 50, 4, 4, 'tanh').
- These configurations represent different context sizes for the model, where `pre` and `suc` parameters denote the size of the context window before and after the current token, respectively.
- The context size determines how much information the model can consider from the surrounding words when making predictions, which can impact the model's ability to capture contextual dependencies.

2. **Impact on Accuracy**:
   - As the context size increases from 0 to 4, there seems to be a fluctuation in accuracy.
   - Configurations with smaller context sizes (e.g., (100, 50, 1, 1, 'tanh') and (100, 50, 2, 2, 'tanh')) tend to achieve higher accuracies compared to those with larger context sizes.
   - The configuration with the smallest context size, (100, 50, 1, 1, 'tanh'), achieves the highest accuracy of 97.87%, indicating that a balanced context size may lead to better performance.
3. **Considerations**:
   - While larger context sizes may theoretically allow the model to capture more contextual information, excessively large context sizes may introduce noise or irrelevant information, leading to decreased performance.
   - It's essential to strike a balance between context size and model complexity, considering factors such as dataset characteristics, task requirements, and computational resources.
   - Further experimentation with different context sizes and validation on larger datasets can provide deeper insights into the optimal context size for the given task and model architecture.

Analysis:

1. **FFNN vs. LSTM**:
   - Both FFNN and LSTM models are evaluated for various configurations.
   - LSTM configurations generally achieve higher accuracies compared to FFNN configurations. For instance, the mean accuracy for LSTM configurations ranges from around 96.41% to 97.32%, while for FFNN configurations, it ranges from around 95.42% to 97.87%.
2. **Effect of Configuration Parameters**:
   - Within the FFNN configurations, changing the context size (`pre` and `suc` parameters) seems to have a noticeable impact on accuracy. For instance, configurations with context sizes of (100, 50, 1, 1, 'tanh') and (100, 50, 2, 2, 'tanh') achieve higher accuracies compared to those with larger context sizes such as (100, 50, 4, 4, 'tanh').
   - There's no explicit indication of context size variations in the LSTM configurations provided. However, it's possible that certain parameters in the LSTM configurations indirectly affect the context size, such as the hidden layer size or the number of layers.
3. **Effect of Embedding Dimension**:

- Within both FFNN and LSTM configurations, there's a trend of higher embedding dimensions (e.g., 200) achieving slightly lower accuracies compared to lower embedding dimensions (e.g., 100). This trend is observed across different activation functions and other parameters.

4. **Effect of Activation Function**:
   - Both ReLU and tanh activation functions are used in the configurations.
   - There isn't a clear dominance of one activation function over the other. The performance seems to depend on other factors such as the specific configuration parameters and the nature of the dataset.
5. **Considerations**:
   - The provided analysis is based on the given data and may not capture the full complexity of the models and their configurations.
   - Further analysis, such as statistical tests or experimentation with different datasets, may be needed to validate these observations and draw more robust conclusions.
   - It's important to consider factors like computational cost, model interpretability, and generalization performance when selecting the final model configurations.