

Neural POS Tagging

FEED FORWARD NEURAL NETWORK

1. Hyperparameters used to train the model are:

Hidden layer size, number of layers, embedding dimensions and activation functions

2. Dev set accuracy of all the configurations and test set accuracy for best model configuration.



```
test_correct = 0
test_total = 0
with torch.no_grad():
    test_outputs = best_model(test_input_tensors)
    _, test_predicted = torch.max(test_outputs, 1)
    test_total += test_label_tensors.size(0)
    test_correct += (test_predicted == test_label_tensors).sum().item()

test_accuracy = test_correct / test_total
print(f'Test Set Accuracy for Best Configuration: {test_accuracy}')
```

Dev Set Accuracy for config: {'hidden_size': 64, 'num_layers': 1, 'embedding_dim': 50, 'activation': ReLU()}, Accuracy: 0.9835942203491872
Dev Set Accuracy for config: {'hidden_size': 128, 'num_layers': 2, 'embedding_dim': 100, 'activation': ReLU()}, Accuracy: 0.9825406381697772
Dev Set Accuracy for config: {'hidden_size': 128, 'num_layers': 3, 'embedding_dim': 200, 'activation': Tanh()}, Accuracy: 0.9820891029500302
Dev Set Accuracy for config: {'hidden_size': 256, 'num_layers': 4, 'embedding_dim': 200, 'activation': ReLU()}, Accuracy: 0.9817000794701986
Best Hyperparameter Configuration: {'hidden_size': 64, 'num_layers': 1, 'embedding_dim': 50, 'activation': ReLU()}, Dev Set Accuracy: 0.9835942203491872
Test Set Accuracy for Best Configuration: 0.9822188449848024

3. Evaluation matrix of best model on dev set and test set

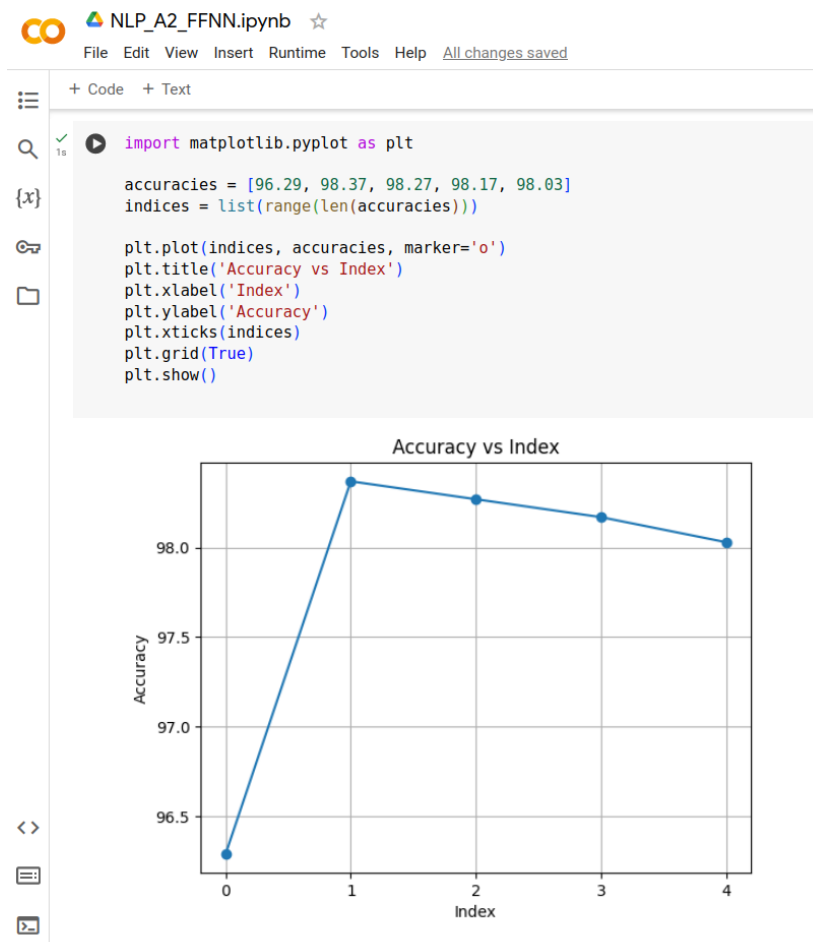
```
CO NLP_A2_FFNN.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

0s ✓
Dev Set Evaluation:
Accuracy: 0.9739614689945816
Recall (Micro): 0.9739614689945816
Recall (Macro): 0.8893533540178932
F1-Score (Micro): 0.9739614689945816
F1-Score (Macro): 0.8815304661180864
Confusion Matrix:
[[ 186  1  23  0  4  0 10  0  0  0  2  0  0]
 [  0 1393  2  0  4  0  1  0  7  8  0  0  0]
 [  2  0  50  1  0  0  3  1  0  0  1  0  1]
 [  0  0  1  254  0  0  0  0  0  0  0  0 11]
 [  0  0  0  0 107  0  0  0  0  0  0  0  0]
 [  0 14  0  0  0 542  0  0  0  0 12  0  0]
 [  0  0  0  0  0  0 35  0  0  0  0  0  0]
 [  0  0  2  0  0  0  0 1128  0  0  6  0  7]
 [  0  0  1  0  0  0  0  1 129  0  0  0  0]
 [  0  1  0  0  1  0  0  0  0 71  0  0  0]
 [  0  0  2  0  0  1  0  0  0  0 411  0  0]
 [  0  0  1  0  0  0  0 13  0  0  0 1536  0]
 [  0  0  0  0  0  0  0  2  0  0  0  0  0]
 [  1 15  3  3  0  0  0  2  0  0  0  0 629]]

Test Set Evaluation:
Accuracy: 0.9791793313069909
Recall (Micro): 0.9791793313069909
Recall (Macro): 0.9627416749933008
F1-Score (Micro): 0.9791793313069909
F1-Score (Macro): 0.9585247511284103
Confusion Matrix:
[[ 199  0 16  0  0  1  1  2  0  0  0  1  0]
 [  0 1413  2  1  0  6  0  0  0  2 10  0  0]
 [  5  1  63  0  0  1  0  0  0  0  1  3  2]
 [  0  0  0 254  0  0  0  0  0  0  0  0  2]
 [  0  0  1  0 107  0  0  0  0  0  1  0  0]
 [  0  0  0  0  0 507  0  0  0  0  5  0  0]
 [  0  0  0  0  0  0 35  1  0  0  0  0  0]
 [  0  1  2  0  0  0  0 1151  0  0  0  6  6]
 [  0  0  0  0  0  1  0  1 119  0  0  6  0]
 [  0  0  0  0  0  0  0  0  0 56  0  0  0]
 [  0  1  0  0  0  0  0  0  1  0 390  0  0]
 [  1  0  0  0  0  1  0 12  2  0  0 1550  1]
 [  0 16  5  3  0  0  1  3  1  0  1  0 599]]
```

4. Context window vs dev set accuracy plot



5. Analysis

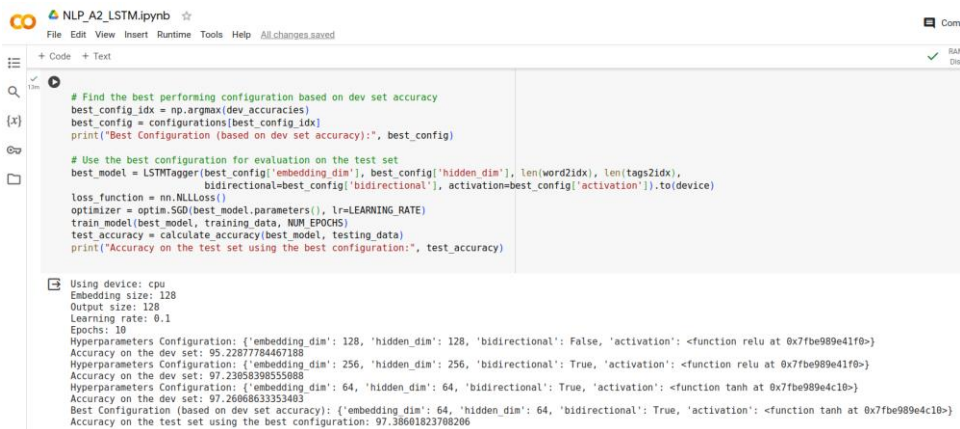
1. Configuration 1, with lower complexity (single layer, lower hidden size, and smaller embedding dimension), outperformed configurations with higher complexity.
2. The use of ReLU activation function seems beneficial, as it resulted in higher accuracy compared to Tanh activation.
3. Increasing the number of layers and hidden size did not necessarily lead to improved performance. It's possible that the model may have overfit the training data with higher complexity configurations.
4. The performance on the test set is consistent with the dev set, indicating that the model generalizes well.

Recurrent Neural Network

1. Hyperparameters used to train the model are:

Hidden layer size, bidirectional, embedding dimensions and activation functions

2. Dev set accuracy of all the configurations and test set accuracy for best model configuration.



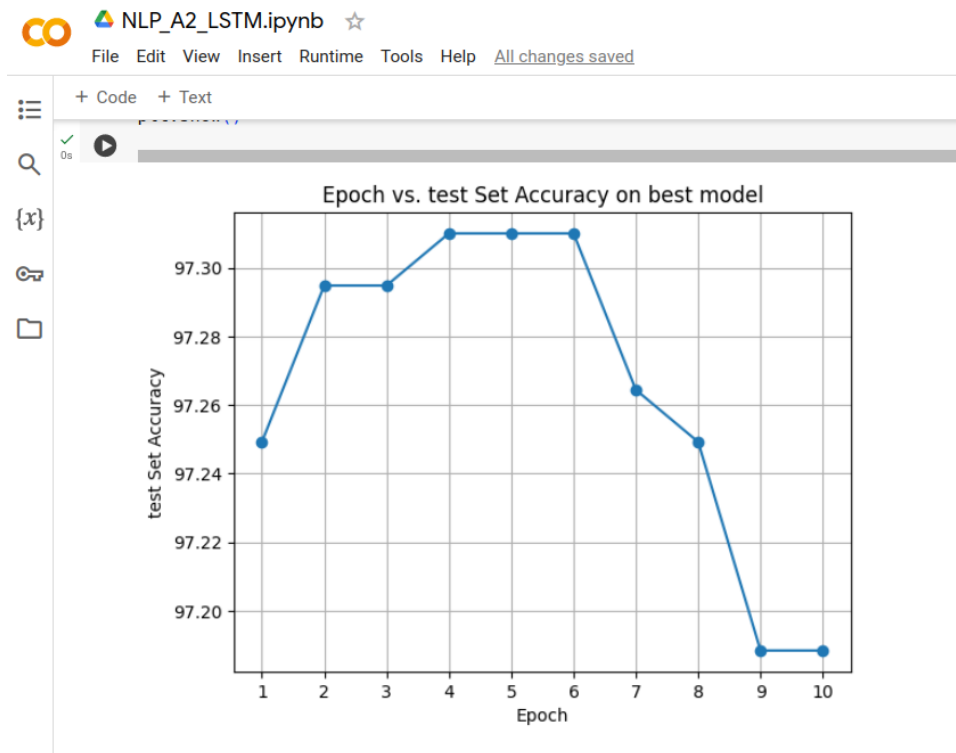
```
NLP_A2_LSTM.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
# Find the best performing configuration based on dev set accuracy
best_config_idx = np.argmax(dev_accuracies)
best_config = configurations[best_config_idx]
print("Best Configuration (based on dev set accuracy):", best_config)

# Use the best configuration for evaluation on the test set
best_model = LSTMTagger(best_config['embedding_dim'], best_config['hidden_dim'], len(word2idx), len(tags2idx),
                        bidirectional=best_config['bidirectional'], activation=best_config['activation']).to(device)

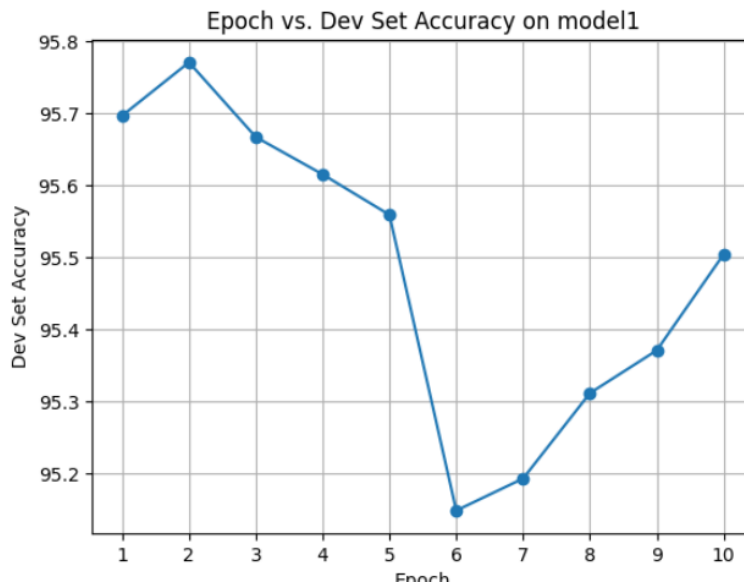
loss_function = nn.NLLLoss()
optimizer = optim.SGD(best_model.parameters(), lr=LEARNING_RATE)
train_model(best_model, training_data, NUM_EPOCHS)
test_accuracy = calculate_accuracy(best_model, testing_data)
print("Accuracy on the test set using the best configuration:", test_accuracy)

Using device: cpu
Embedding size: 128
Output size: 128
Learning rate: 0.1
Epochs: 10
Hyperparameters Configuration: {'embedding_dim': 128, 'hidden_dim': 128, 'bidirectional': False, 'activation': <function relu at 0x7f9e989e41f0>}
Accuracy on the dev set: 95.22877784467188
Hyperparameters Configuration: {'embedding_dim': 256, 'hidden_dim': 256, 'bidirectional': True, 'activation': <function relu at 0x7f9e989e41f0>}
Accuracy on the dev set: 97.23058398555088
Hyperparameters Configuration: {'embedding_dim': 64, 'hidden_dim': 64, 'bidirectional': True, 'activation': <function tanh at 0x7f9e989e4c10>}
Accuracy on the dev set: 97.26068633353403
Best Configuration (based on dev set accuracy): {'embedding_dim': 64, 'hidden_dim': 64, 'bidirectional': True, 'activation': <function tanh at 0x7f9e989e4c10>}
Accuracy on the test set using the best configuration: 97.30601823780206
```

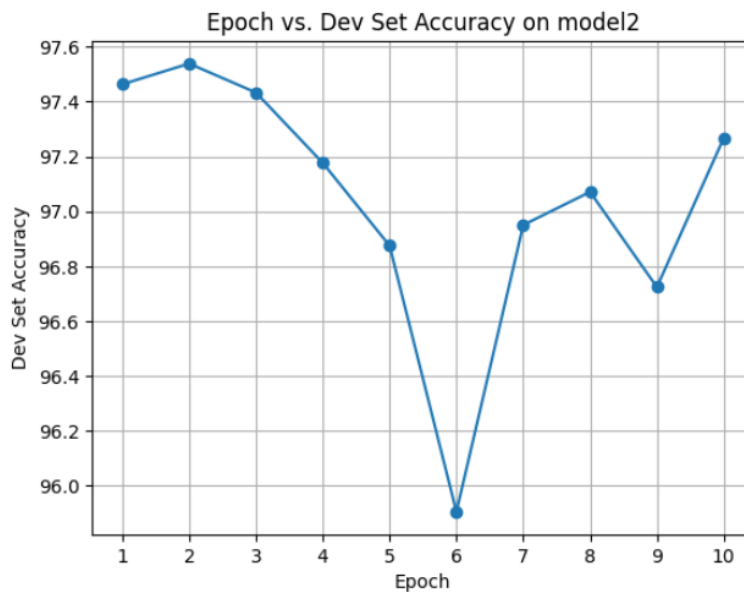
3. Plot of epoch vs test set accuracy on best model



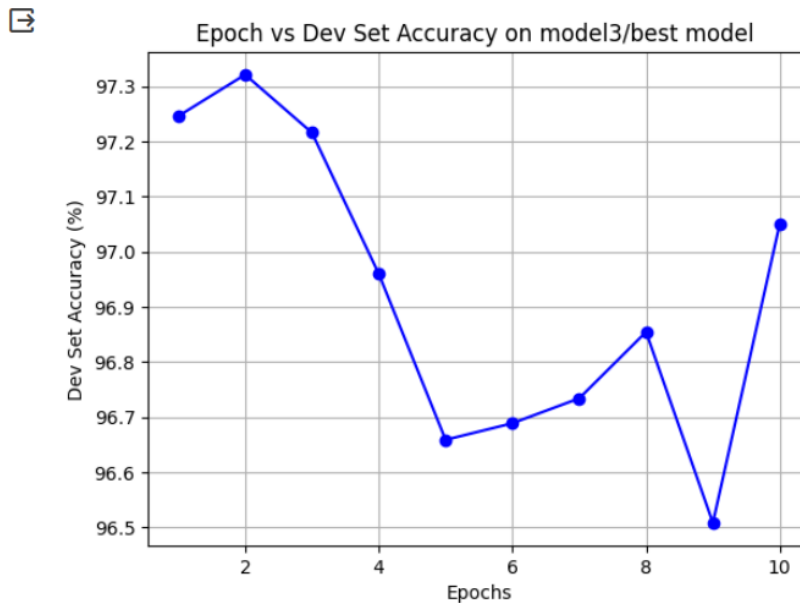
4. Plot of epoch vs Dev set accuracy on model 1



5. Plot of epoch vs Dev set accuracy on model 2



6. Plot of epoch vs Dev set accuracy on model3/best model



6. Analysis

- Configuration 3, with lower complexity (smaller hidden size and embedding dimension) and bidirectional LSTM, outperformed other configurations.
- The use of the Tanh activation function in Configuration 3 seems beneficial, as it resulted in higher accuracy compared to ReLU activation.
- The bidirectional LSTM in Configuration 3 might have helped capture more contextual information, contributing to its improved performance.
- Configuration 1, with a single directional LSTM and higher complexity, performed the worst among the configurations, indicating that increasing model complexity does not always lead to better performance.

