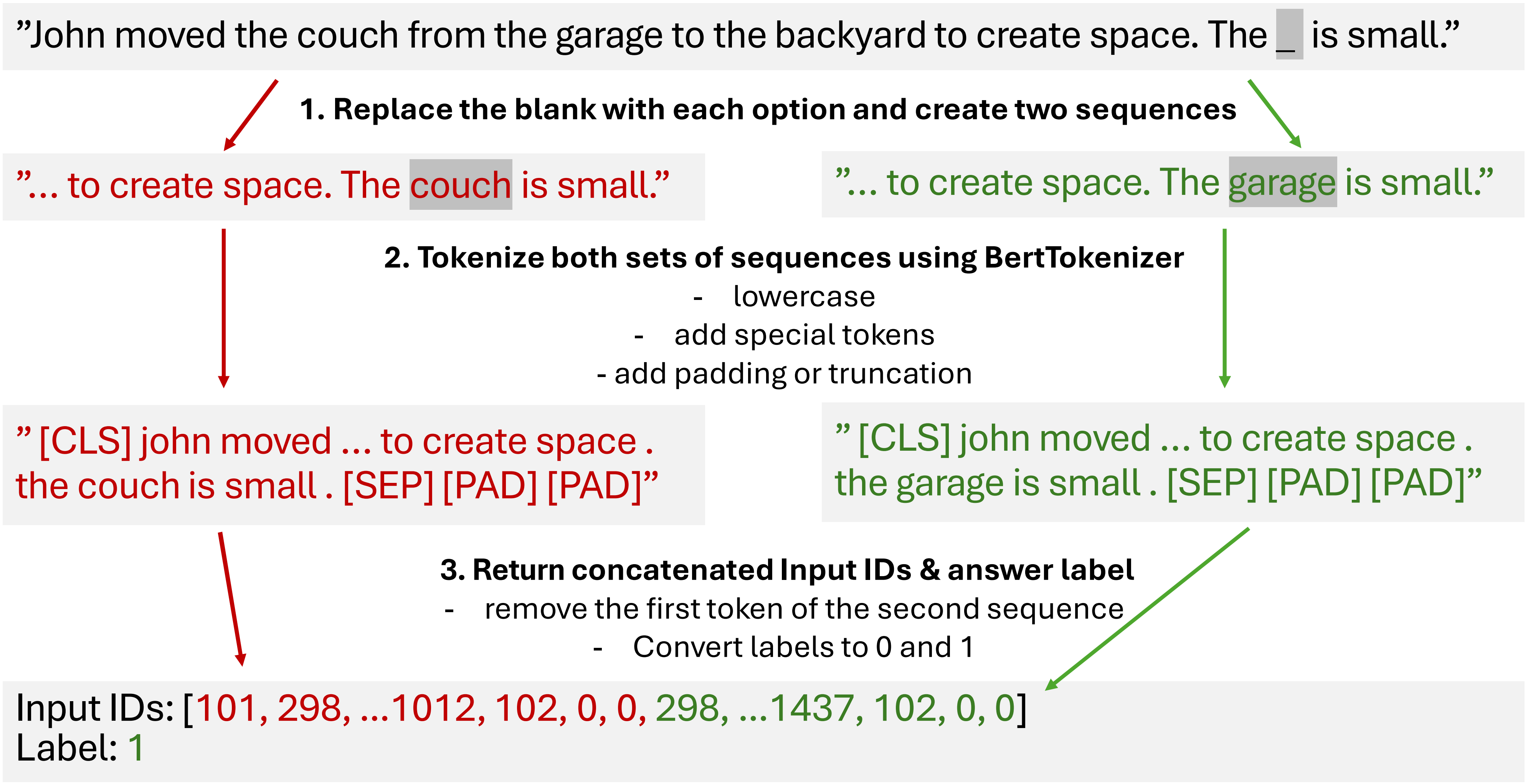
Project 2: RNNs

SW05 – Jannine Meier – FS24

Preprocessing: Winogrande dataset



”John moved the couch from the garage to the backyard to create space. The _ is small.”

1. Replace the blank with each option and create two sequences

”... to create space. The couch is small.”

”... to create space. The garage is small.”

2. Tokenize both sets of sequences using BertTokenizer

- lowercase
- add special tokens
- add padding or truncation

” [CLS] john moved ... to create space . the couch is small . [SEP] [PAD] [PAD]”

” [CLS] john moved ... to create space . the garage is small . [SEP] [PAD] [PAD]”

3. Return concatenated Input IDs & answer label

- remove the first token of the second sequence
- Convert labels to 0 and 1

Input IDs: [101, 298, ...1012, 102, 0, 0, 298, ...1437, 102, 0, 0]
Label: 1

Preprocessing: Synthetic dataset

Anagram dataset

1. Split the sequence between the <sep>

Palindrome dataset

1. Treat the entire sequence as one

2. Tokenize at character level to IDs (no spaces)

ID 0 reserved for [PAD] -> next slide

ID 1 reserved for <sep>

b p k <sep> k p b

b p k

[2, 3, 4]

k p b

[4, 3, 2]

j q f m m s f q j

[7, 8, 4, 9, 9, 4, 8, 7]

3. Return concatenated IDs& label

Input IDs: [2, 3, 4, 1, 4, 3, 2]
Label: 1

3. Return IDs & label

Input IDs: [7, 8, 4, 9, 9, 4, 8, 7]
Label: 0

Preprocessing: Input format

Custom Collate Function

BertTokenizer data

- Convert IDs and labels into tensors

CharTokenizer data

- Convert IDs and labels into tensors
- Pad sequences to the longest sequence in the batch

Returns

tensor([[1, 2, 3, 0], [4, 5, 0, 0]])	**input_ids:** a tensor of padded sequences -> dimensionality: batch_size x max_sequence_length
tensor([1, 0])	**labels:** a tensor containing the labels for each sequence in the batch -> dimensionality: batch_size
tensor([3, 2])	**sequence_lengths:** a tensor containing the actual lengths of each sequence before padding -> dimensionality: batch_size

Network Architecture: RNNClassifier

2-Layer RNN with LSTM and nn.Embedding

Forward Pass Modification

Most of inputs are padded with zero

➔hidden state is meant to accumulate and carry forward information from one step of the sequence to the next

➔padding tokens in this process can introduce bias

➔Solution: ignore the padding by taking the actual lengths of each sequence

Instantiation

input_dim= length of tokenizer's vocab
output_dim= 1
num_layers= 2
embedding_dim
hidden_dim
dropout

loss_function = nn.BCEWithLogitsLoss()

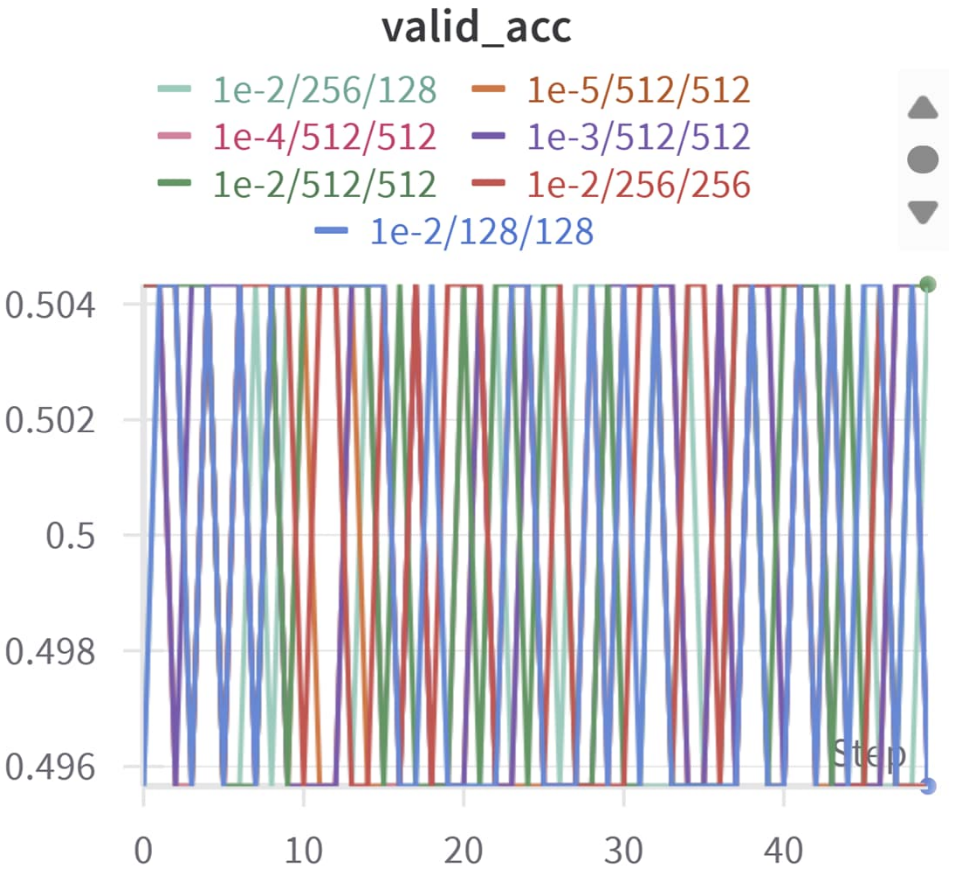
optimizer = optim.Adam()

Experiments: Winogrande

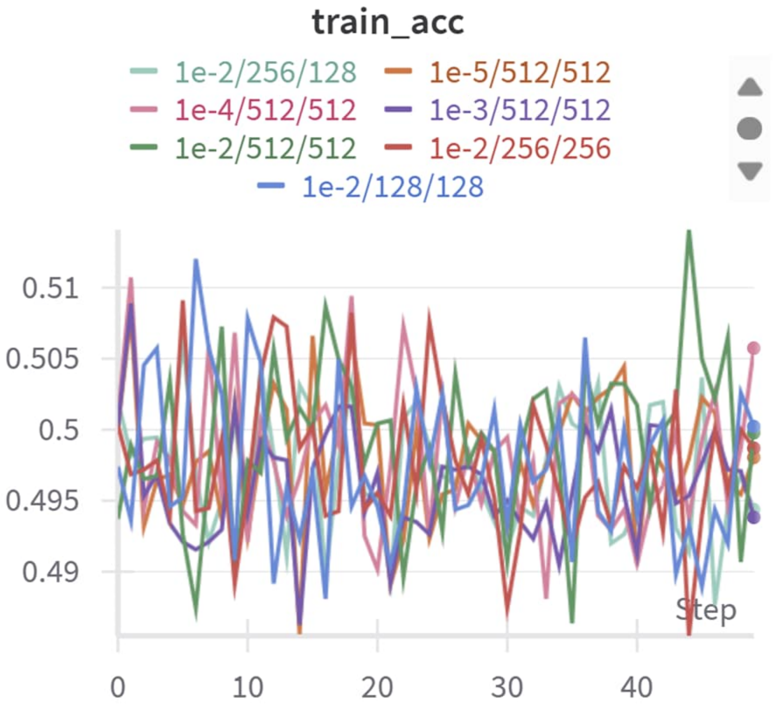
Results

- Best Validation Accuracy: 50.40%
- Test Accuracy: 50.00%

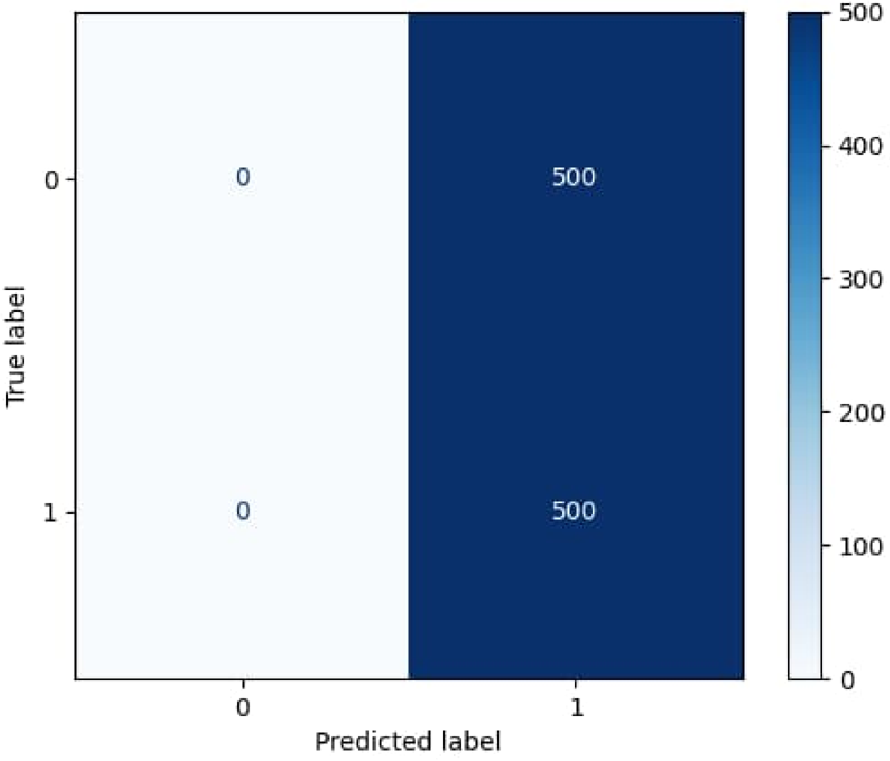
- Epochs: 50
- Hidden Layers: 128 -> 512
- Embedding Layers: 128 -> 512
- Learning Rate: 1e-2 -> 1e-5



Experiments: Winogrande



-> no learning progress



-> guessing only one label

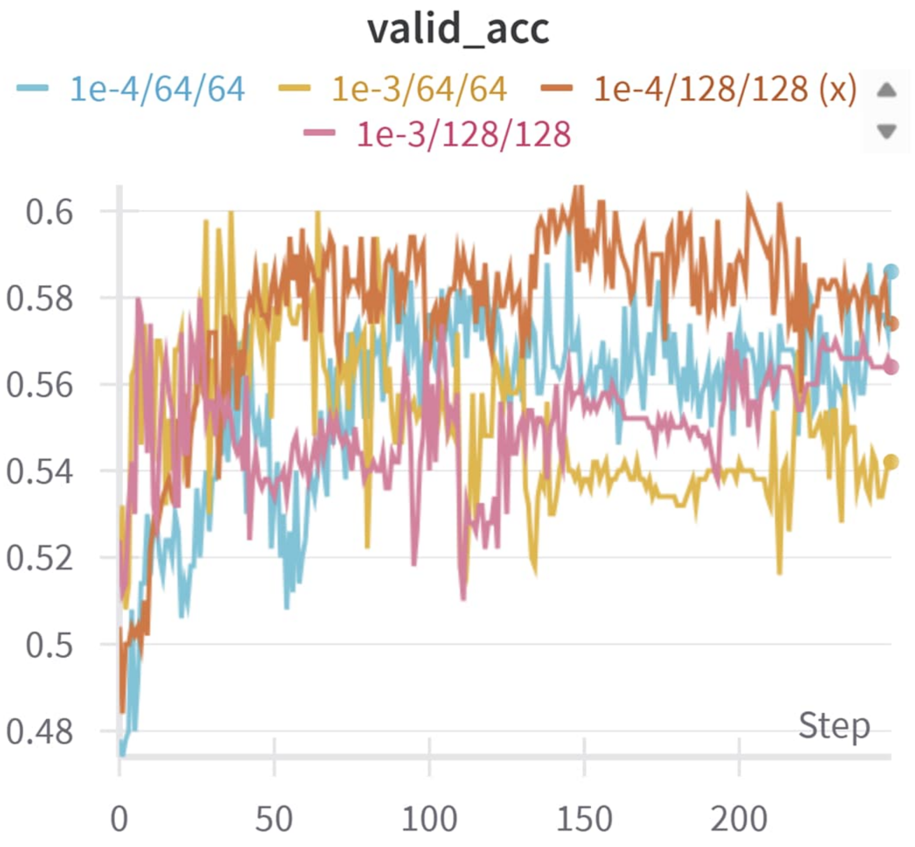
Experiments: Anagram

Results

- Best Validation Accuracy: 60.60%
- Test Accuracy: 50.07%

- Epochs: 250
- Hidden Layers: 64 -> 512
- Embedding Layers: 64 -> 512
- Learning Rate: 1e-2 -> 1e-5
- Dropout 0.1 -> 0.3

-> Overfitting after 20 epochs



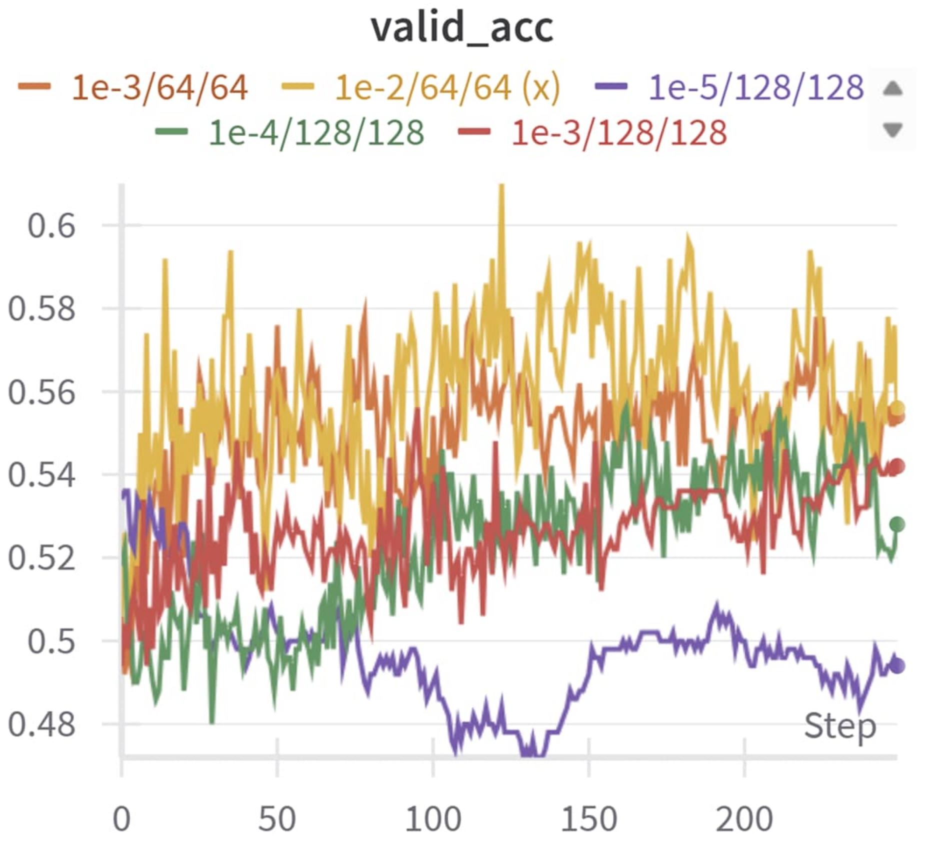
Experiments: Palindrome

Start

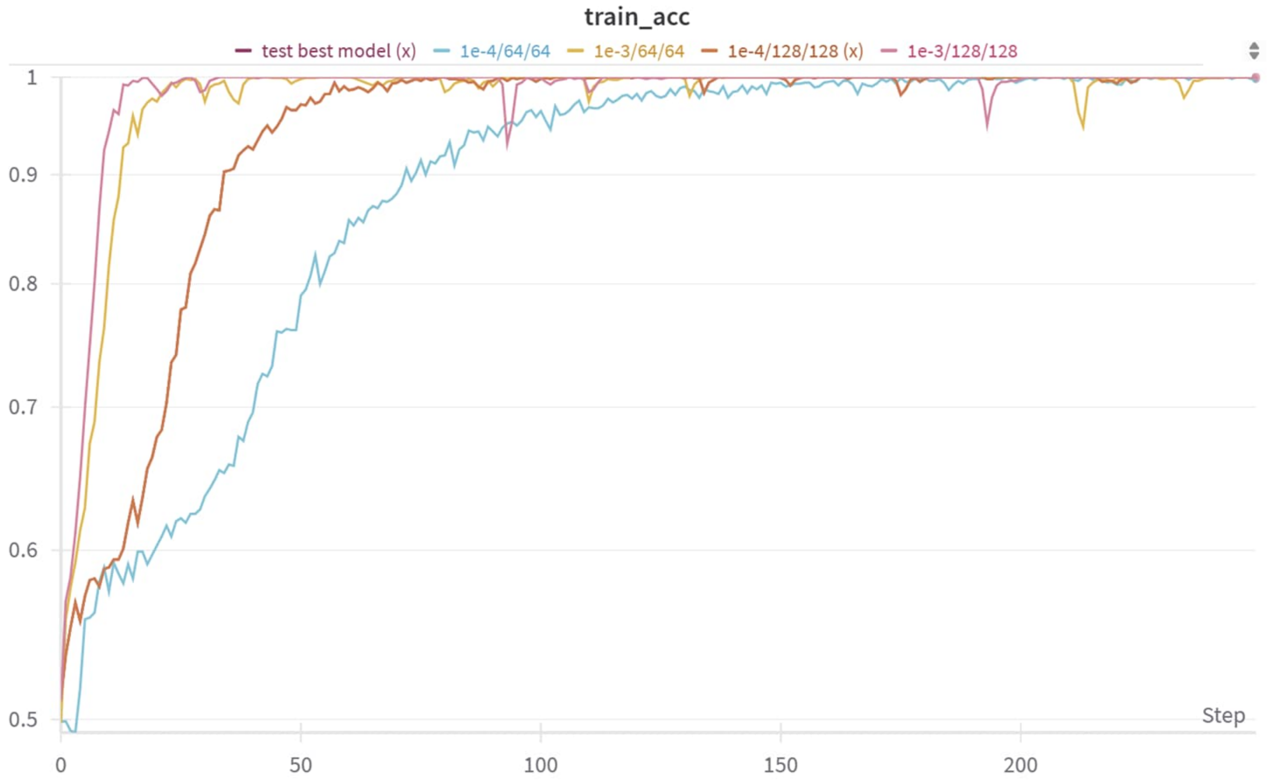
- Best Validation Accuracy: 61.00%
- Test Accuracy: 50.32%

- Epochs: 250
- Hidden Layers: 64 -> 512
- Embedding Layers: 64 -> 512
- Learning Rate: 1e-2 -> 1e-5
- Dropout 0.1 -> 0.3

-> Overfitting after 15 epochs

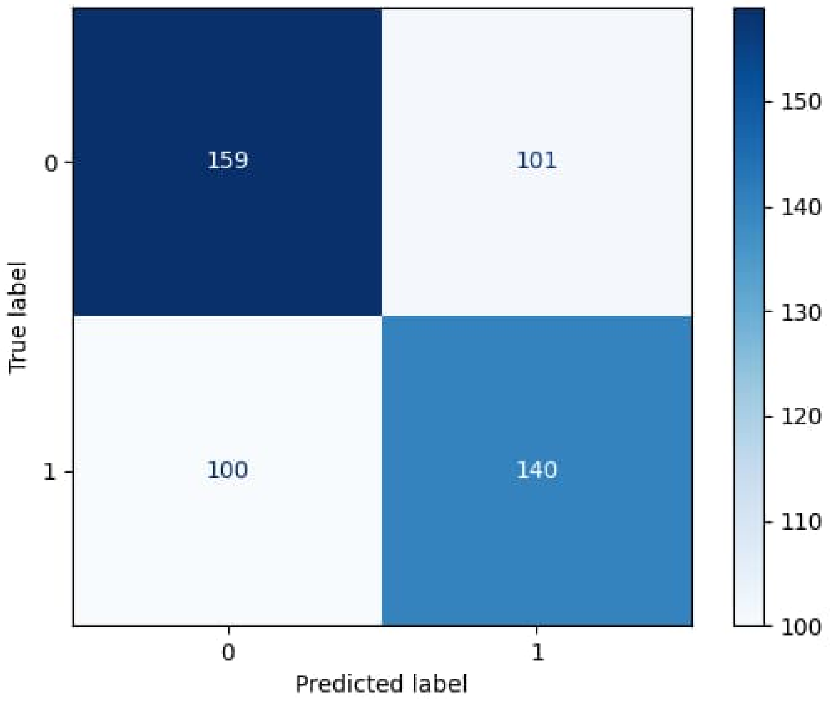


Synthetic dataset (Palindrome): Overfitting

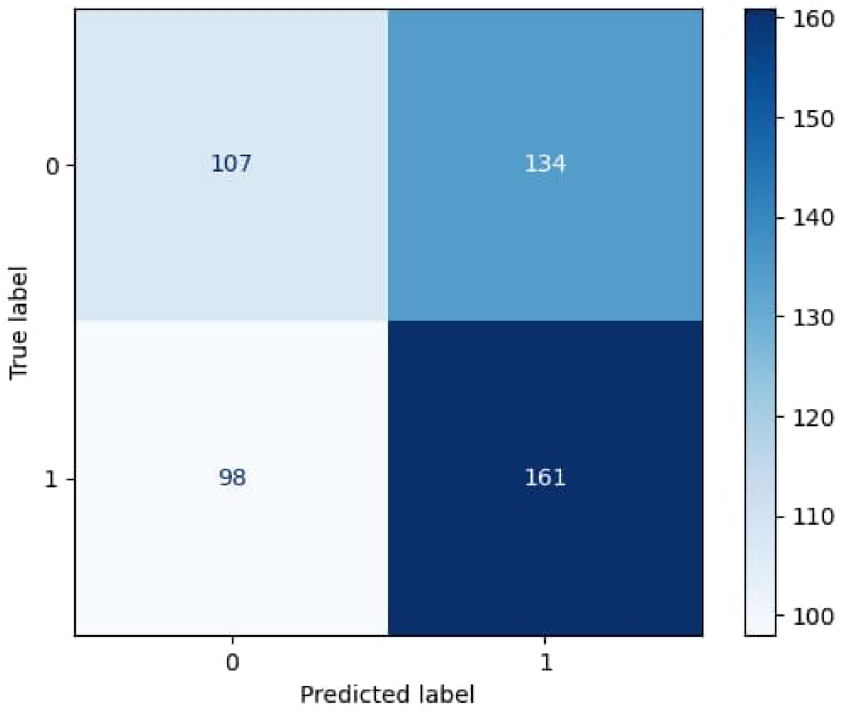


Confusion Matrix

Anagram



Palindrome



Results

Interpretation

Winogrande dataset

- model guesses one label only

Synthetic dataset

- marginally better than random
- extreme overfitting

Considerations

Winogrande dataset

- too complex for 2-Layer RNN

Synthetic dataset

- not enough data
- measures against overfitting
    - higher dropout
    - more hyperparamter tuning