Name: Stephen Pasco

Collaborators: Google Gemini Flash for Q&A – Understanding tasks, Kera API options

CSCI S-89B Introduction to Natural Language Processing

Assignment 1

Problem 1: SOLUTION:

(a)

Plot Generation ✅

- Content: Dual-panel plot showing:

- Training accuracy increasing from ~49% to ~95%

- Validation accuracy peaking at 82.8% at epoch 14

- Clear overfitting pattern after epoch 14

- Optimal epoch marked with green star and dashed line

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.  
Image: reuters\_training\_history.png

**Python Files:**

- 1/part-1/reuters\_text\_classifier.py

(b)

**See File:** 1/part-1/part1b\_results.txt

**Test Accuracy Results ✅**

- File: part1b\_results.txt

- Results:

- Test Set: 2,246 samples

- Optimal Epochs: 14

- Validation Accuracy at Optimal: 82.8%

- TEST ACCURACY: 78.36%

- Random Baseline: 18.61%

- Improvement: 59.75 percentage points

PART 1B - TEST ACCURACY RESULTS

================================

Configuration:

Random Seed: 42

Dataset: Reuters (46 classes)

Vocabulary Size: 10,000 words

Training Set Size: 7982 samples

Validation Set Size: 1000 samples

Test Set Size: 2246 samples

Model Selection:

Selection Criterion: Maximum Validation Accuracy

Optimal Epochs: 14

Validation Accuracy at Optimal Epoch: 0.8280

Validation Loss at Optimal Epoch: 0.8713

Minimum Validation Loss Epoch: 12 (loss: 0.8575)

Test Set Performance:

TEST ACCURACY: 0.7836 (78.36%)

TEST LOSS: 1.0059

Comparison:

Random Baseline: 0.1950 (19.50%)

Improvement over Baseline: 58.86 percentage points

**Problem 2 (25 points)**

Modify the network in Problem 1 in a way that the test accuracy reported in Problem 1(b) is improved.

1. Plot the results for training and validation accuracy versus number of epochs.
2. Report the test accuracy of the model when trained with the optimal number of epochs.

SOLUTION to (a) and (b):

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

AI-generated content may be incorrect.

EXPERIMENTAL RESULTS

--------------------

Multiple configurations were tested during optimization:

1. ORIGINAL ARCHITECTURE (Baseline)

- Configuration: 64 → 64 hidden units with dropout

- Test Accuracy: 76.27%

- Random Baseline: 19.50%

2. IMPROVED ARCHITECTURE (Current Best)

- Configuration: 256 → 128 → 64 with batch norm and progressive dropout

- Test Accuracy: 80.01% ✓

- Test Loss: 1.0639

- Validation Accuracy: 82.00%

- Optimal Epochs: 20

- Batch Size: 256

3. WIDER ARCHITECTURE (Tested)

- Configuration: 512 → 384 → 256 → 128

- Test Accuracy: 80.59%

- Note: Only marginal improvement despite 4x parameters

4. HEAVY REGULARIZATION (Tested)

- Configuration: L2 regularization + MaxNorm constraints

- Test Accuracy: 78.41%

- Note: Over-regularization decreased performance

5. ADVANCED TECHNIQUES (Tested and Removed)

- TF-IDF vectorization, GELU activation, AdamW, label smoothing

- Test Accuracy: 79.74%

- Note: Advanced techniques couldn't overcome one-hot encoding limitations

PERFORMANCE IMPROVEMENT

-----------------------

- Absolute Improvement: +3.74 percentage points (76.27% → 80.01%)

- Relative Improvement: 4.90% increase in accuracy

- Improvement over Random Baseline: 60.51 percentage points

- Better Generalization: Validation accuracy consistently ~2% higher than test

KEY FINDINGS

------------

1. Moderate capacity increase (256-128-64) provides optimal balance

2. Batch normalization significantly improves training stability

3. Progressive dropout (0.4→0.3→0.2) more effective than uniform dropout

4. Batch size 256 outperforms both smaller (64) and larger (512) sizes

5. One-hot encoding creates fundamental limitation around 80-82% accuracy

6. Network-only optimizations have diminishing returns beyond 80%

7. Achieving 90%+ accuracy would require more research

BATCH SIZE IMPACT

-----------------

- Batch size 256: 80.01% accuracy (optimal)

- Batch size 512: 79.61% accuracy

- Smaller batch size provides better generalization through gradient noise

LIMITATIONS DISCOVERED

----------------------

1. One-hot encoding information bottleneck (sparse 10,000-dim binary vectors)

2. Simply adding more layers/neurons shows diminishing returns

3. Heavy regularization can hurt performance (optimal is moderate)

4. Validation-test gap of ~2% indicates some overfitting to validation set

(b)

**Problem 3 (25 points)**

Consider your network from Problem 2 and experiment with various optimizers and corresponding hyperparameters. Please try at least four optimizers and plot the validation/train accuracy in each case. Present the best validation accuracy found in each case as a table indicating considered optimizers and hyperparameters.

Finally, use the remaining 2,246 test examples to compute the test accuracy for your best model. Compare the test accuracy with the ones obtained in Problems 1 and 2.

SOLUTION:

The task successfully identified RMSprop with lr=0.001 as the optimal optimizer for this Reuters text classification task, achieving 82.80% validation accuracy. The top 5 optimizers all performed within a narrow range, suggesting the improved architecture (with BatchNorm and Dropout) is relatively robust to optimizer choice. The significant drop in performance with lr=0.0001 across all optimizers highlights the importance of proper learning rate selection.

**Experiments ConductedA group of graphs with different colored lines

AI-generated content may be incorrect.**

* **Total Configurations Tested**: 14 (reduced from original 20)
* **Optimizers Evaluated**: 5 (RMSprop, Adam, Adamax, AdamW, Nadam)
* **Hyperparameters Varied**:
  + Learning rates: 0.0001, 0.001, 0.002, 0.01
  + Optimizer-specific: rho, beta values, weight decay
* **Training Duration**: ~30 epochs per experiment with early stopping

### Key Findings

1. **Performance Range**: Top 5 optimizers achieved 82.20-82.80% validation accuracy (only 0.6% spread)
2. **Optimal Learning Rate**: 0.001 was consistently best across most optimizers
3. **Convergence Speed**:
   * Fastest: Nadam (lr=0.002) at epoch 10
   * Average: Most optimizers converged around epochs 20-24
   * Slowest: Low learning rates (0.0001) required 27-30 epochs
4. **Generalization**: All top optimizers showed good generalization with ~2-3% gap between validation and test accuracy
5. **Winner**: RMSprop with lr=0.001 achieved the best validation accuracy (82.80%) and strong test performance (80.19%)

A screenshot of a table

AI-generated content may be incorrect.