**Homework 2 Description: Document Classification with Neural Networks**

**STEP 1: FEATURE ENGINEERING AND INITIAL EVALUATION**

**1(a) – Neural Network with CountVectorizer**

* I preprocessed the text using lowercasing, punctuation removal, and stopword filtering.
* Used CountVectorizer with unigram + bigram (ngram\_range=(1,2)) and max 5000 features.
* A 2-hidden-layer neural network (128 neurons each) was trained and evaluated using 5-fold cross-validation.
* **Avg Train Accuracy:** 1.00
* **Avg Val Accuracy:** 0.970
* **Val Std Dev:** 0.0071

**1(b) – Neural Network with TF-IDF Features**

* Used TfidfVectorizer with the same settings as above.
* TF-IDF gave a slightly lower training variance, improving generalization.
* **Avg Train Accuracy:** 0.9998
* **Avg Val Accuracy:** 0.970
* **Val Std Dev:** 0.0063

**1(c) – Feature Summary**

I explored two methods to convert text into numerical features for training neural networks:

CountVectorizer (Baseline):

Used CountVectorizer with unigrams and bigrams (ngram\_range=(1,2)) and a vocabulary size limited to 5000. Each document is represented as a sparse vector of word counts.

TF-IDF (Enhanced Features):

Applied TfidfVectorizer with the same n-gram range and feature limit. This method down-weights common words by using inverse document frequency, helping to highlight more informative terms.

Both feature sets were evaluated using 5-fold cross-validation with a neural network (2 hidden layers, 128 neurons each). TF-IDF provided better validation accuracy than CountVectorizer.

**1(d) – Results Table**

| **Feature Method** | **Train Accuracy** | **Val Accuracy** | **Train Std Dev** | **Val Std Dev** |
| --- | --- | --- | --- | --- |
| CountVectorizer | 1.0000 | 0.9700 | 0.0000 | 0.0071 |
| TF-IDF | 0.9998 | 0.9700 | 0.0005 | 0.0063 |

**STEP 2: MODEL TUNING**

**2(a) – Learning Rate Tuning with 5-Fold CV**

* Evaluated: [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1]
* Best learning rate: **0.001**
  + **Train Acc:** 1.00
  + **Val Acc:** 0.976
  + **Val Std Dev:** 0.0066

**2(b) – Learning Rate Table & Plot**

| **Learning Rate** | **Train Acc** | **Val Acc** | **Train Std Dev** | **Val Std Dev** |
| --- | --- | --- | --- | --- |
| 0.0001 | 0.7742 | 0.725 | 0.1739 | 0.1674 |
| 0.0003 | 0.9868 | 0.943 | 0.0076 | 0.0169 |
| **0.001** | 1.0000 | 0.976 | 0.0000 | 0.0066 |
| 0.003 | 1.0000 | 0.972 | 0.0000 | 0.0051 |
| 0.01 | 1.0000 | 0.965 | 0.0000 | 0.0110 |
| 0.03 | 1.0000 | 0.964 | 0.0000 | 0.0066 |
| 0.1 | 0.9990 | 0.950 | 0.0009 | 0.0122 |

**2(c) – Optimizer Tuning**

* Evaluated: SGD, Adam, RMSprop
* Best: **RMSprop**
  + **Train Acc:** 1.00
  + **Val Acc:** 0.974
  + **Val Std Dev:** 0.0066

| **Optimizer** | **Train Accuracy** | **Val Accuracy** | **Train Std Dev** | **Val Std Dev** |
| --- | --- | --- | --- | --- |
| SGD | 0.204 | 0.210 | 0.0190 | 0.0207 |
| Adam | 1.000 | 0.970 | 0.0000 | 0.0045 |
| RMSprop | 1.000 | 0.974 | 0.0000 | 0.0066 |

**STEP 3: TEST PREDICTION USING FINAL MODEL**

**3(a) – Test Data Preprocessing**

The test data was preprocessed using the same pipeline as the training data:

- Lowercased text

- Removed punctuation and stopwords

- Converted text to TF-IDF vectors using the vectorizer trained on the training set

This ensures consistent feature representation between training and testing sets.

**3(b) – Train Final Model**

The final model was chosen based on the best-performing configuration from the previous experiments. Specifically:

- **Feature Representation**: TF-IDF (unigram + bigram, max 5000 features)

- **Model Type**: Neural Network with two hidden layers of 128 neurons each

- **Optimizer**: Adam

- **Learning Rate**: 0.001

This combination was selected because it consistently provided high validation accuracy and low variance across 5-fold cross-validation, indicating strong generalization and stability.

**3(d) – Model Selection Summary**

The final model was selected based on validation performance during tuning. TF-IDF with RMSprop provided the best generalization with low variance and consistent high accuracy.

**3(d)(1) – Performance on Training Data**

* Final model trained on entire data achieved **100% training accuracy**
* Cross-validation average validation accuracy: **~97%**
* Low standard deviation across folds **(∼0.006**), indicating reliable and consistent performance.