Assignment: Compare Naive Bayes and ANN (One Hidden Layer)

Natural Language Processing — Fall 2025

Instructor: Ayesha Enayet

Total: 5 points

Overview

Your task is to **train from scratch** two text classifiers on the **SMS Spam Collection** (ham vs spam):
(1) a **Naive Bayes (NB)** model; and (2) an **Artificial Neural Network (ANN)** with **exactly one hidden layer**.
You may explore different preprocessing methods, hyperparameters, and training setups to maximize performance. Treat this as a **challenge/competition**.

Dataset: SMS Spam Collection (5,574 SMS), labeled *ham*/*spam*. Download:
https://archive.ics.uci.edu/dataset/228/sms+spam+collection

Learning Objectives

- Build and evaluate Naive Bayes and a Artificial Neural Network (1-hidden-layer ANN).
- Compare models fairly under a fixed protocol; analyze errors and robustness.

Tasks

A. Data Preparation (0.5 pt)

A.1 Load and clean the SMS dataset. Split data into train and test set (80/20). Report counts per class and split.

A.2 Text features: Use one-hot encoding for ANN.

B. Naive Bayes (1.0 pt)

B.1 Train a **Naive Bayes** classifier. **Train from scratch:** Train from scratch and use statistics from the training set only. No pre-trained components.

B.2 You must cache statistics that can be computed once and reused (e.g., class priors, word likelihoods, vocabulary, and any preprocessed features). Recomputing these at every run or for every prediction will be considered a computationally expensive implementation, and I will deduct marks accordingly.

B.3 Tune smoothing α (try a small grid, e.g., $\{0.1, 0.5, 1.0\}$) using validation.

B.4 Report test results (Accuracy, Precision, Recall, F1). Include a confusion matrix on the test set.

C. ANN with One Hidden Layer (1.5 pt)

C.1 Architecture: Input = your text vector; **exactly one hidden layer** with your chosen size (e.g., 64/128/256), activation of your choice (ReLU, tanh, etc.), and an output layer with sigmoid (binary) or 2-way softmax.

C.2 **Train from scratch:** initialize weights randomly (no pre-trained embeddings, no transfer learning).

C.3 Report test results (Accuracy, Precision, Recall, F1). Include a confusion matrix on the test set.

D. Fair Comparison & Analysis (1 pt)

D.1 Use the **same train/test split** and base features when comparing NB and ANN (you may additionally show ablations).

D.2 Present a **results table** and a **short analysis** (5–8 bullet points): When does NB win/lose? What preprocessing mattered? Any overfitting signs? What kinds of messages are misclassified?

Competition Bonus (1.0 pt out of 5)

Leaderboard Bonus (1.0 pt): The **best test accuracy in class** earns the full 1.0 bonus point.
"-3 policy" (near-best also rewarded): Any submission within **3 percentage points** of the best test accuracy also receives the full 1.0 bonus. Ties are allowed.
Example: if the top accuracy is 98.2%, then any submission $\geq 95.2\%$ gets the bonus.
Update your resulsts https://docs.google.com/spreadsheets/d/1yv7LhDDZcqbmYb9_Ps1hY6Y3hWjGbHdZTV6Fp
edit?usp=sharing
Competition Threshold: There is a hidden performance threshold for the leaderboard bonus; submissions below this threshold receive 0 competition marks, regardless of ranking.

Academic Integrity (Read Carefully)

Train and implement on your own. Any **significant overlap** with either other humans' work or AI-generated code/text will result in a **zero grade**. Do not share code, write-ups, or prompts. You may consult official library docs and general references but **all code and analysis must be your own**. We are not discussing how overlap is detected; **focus on implementing it yourself.**

Protocol Details & Constraints

- **Allowed:** Different preprocessing pipelines (tokenization, lowercasing, stopwords, n-grams), feature types, hyperparameters, and class-prior adjustments.
- **Not allowed:** Any pre-trained embeddings or models; more than one hidden layer; external datasets (beyond SMS Spam).
- **Evaluation:** Report Accuracy, Precision, Recall, and F1 on validation and test splits; include a confusion matrix for the test set.

From-Scratch Implementation Rule

No prebuilt ML libraries. Implement both **Naive Bayes** and the **ANN (one hidden layer)** entirely from scratch. Do *not* use scikit-learn, TensorFlow, PyTorch, Keras, JAX, statsmodels, or any NB/NN helper packages. No automatic differentiation or library optimizers.

You must implement:

- Text vectorization coded by you.
- Naive Bayes parameter estimation (priors, likelihoods).
- ANN components: weight initialization, forward pass, activation functions (e.g., sigmoid, tanh, ReLU), loss, **backpropagation/gradients** implemented by you.

Allowed utilities: Standard Python for I/O and light helpers only (e.g., `csv`, `json`, `re`, `math`, `random`, `collections.Counter`). If used, `numpy` is limited to basic array/matrix arithmetic; no ML/NN/NB functions or autograd. For plots/tables only: `matplotlib`, `pandas` (optional). These must not perform any model training or inference.

Prohibited shortcuts: Any function that trains a classifier, provides NB/NN layers, activation function, or computes gradients automatically.

Integrity: Any significant overlap with human or AI work results in a **zero**. Focus on implementing it yourself.

What to Submit

1. **PDF report** (max 3 pages):
   - Your report must use the official ACL Conference LaTeX template. Length: 3–4 pages of main content.
   - Dataset handling and split statistics; feature choices.
   - Model specs: NB variant and α; ANN architecture and hyperparameters.
   - Results table for both models; test confusion matrix.
   - 5–8 bullet analysis and 3 misclassification examples with your interpretation.
2. **Code** (zip): Submit a single .zip with all source code and input/output files (train/test splits, saved NB stats/ANN weights, tables, figures) plus a short README with exact run commands and dependencies.`requirements.txt`/`environment.yml`. Your code should regenerate your results on our machine.

Grading Rubric (5 points total).

Data/Features (0.5)	Clean split; justified feature pipeline; class distribution reported.
Naive Bayes (1.0)	Correct training from scratch; metrics + confusion matrix.
ANN (1 hidden) (1.5)	Correct architecture; train-from-scratch; reasonable tuning; metrics + plot.
Comparison (1.0)	Fair protocol; results table; insightful analysis with examples.
Competition Bonus (1.0)	Best or within 3 pp of best test accuracy (**bonus beyond the 4 base points**).

Notes: Submissions not in the correct template/style (ACL style) will receive a grade of 0 for the entire assignment. Keep your report concise and focused on evidence. Clearly label all figures/tables. If you deviate from the protocol, justify why and show the effect.