Auburn University
Assignment 4

COMP 5630/ COMP 6630/ COMP 6630 - D01 (Fall 2025)
Machine Learning

Deadline: Nov 19, 2025, 11:59 PM CST

1 Word Embeddings (10 Points)

1. You will examine two-word embeddings. You are given the following words.
   Dog
   Bark
   Tree
   Bank
   River
   Money

   (a) Use Glove-twitter-50D word2vec and compute nxn matrices using cosine similarities for the given words. Use the following syntax to import glove
   import gensim.downloader as api
   wv = api.load('glove-twitter-50')
   Use the configuration
   sentences=common_texts, vector_size=50, window=5, min_count=1

   (b) Now use Fasttext Embedding from Genism and compute nxn matrices as question a. Use the following configuration

FastText(vector_size=50, window=5, min_count=1, sentences=common_texts, epochs=10)

(c) Which embedding captures better semantics? Justify your answer.

Link to FastText

2 LSTM (40 Points)

You will be training an LSTM to predict the emoji associated with a short tweet.
Input: Tweet text
Output: Emoji label

1. Load tweet and emoji dataset from here emoji subset. Use only the training examples as your entire dataset.
2. Preprocess the dataset.
   (a) Clean tweets (remove URLs, mentions, hashtags, special characters).
   (b) Tokenize and pad sequences for model input.
   (c) One-hot encode labels.
3. Split the dataset as follows: 80% training, 10% validation, 10% test.
4. **Baseline LSTM**
   (a) Train a simple LSTM with random embeddings of 50D.
   (b) Experiment with hyperparameters:
   Epochs: try 3, 5, 10
   Dropout: try 0.2, 0.3, 0.5
   (c) Record validation accuracy, test accuracy, and training loss for each combination.
5. **LSTM with FastText Embeddings**
   (a) Load pre-trained FastText embeddings (50D).

(b) Replace the embedding layer in your previous LSTM implementation with FastText embeddings.

(c) Train the model with different hyperparameters:
Epochs: 3, 5, 10
Dropout: 0.2, 0.3, 0.5

(d) Report performance metrics and compare with baseline.

6. Hyperparameter Analysis

(a) For each model, create a table of results showing:

- Epochs
- Dropout
- Validation Accuracy
- Test Accuracy
- Notes on convergence/training behavior

(b) Identify best hyperparameter combination for each model.

(c) Justify which embedding works best and why.

7. Compare models and justify your findings.

(a) Which embeddings performed better and why?

(b) How did hyperparameters affect performance?

(c) Any observations on misclassified tweets or common errors.