Assignment 1

Solomon Gruse

January 26, 2025

1 Refreshing the Rows and Columns: Linear Algebra Review

1.1 Basic Operations

1. $\sum_{i=1}^{n} x_i y_i = 14$
2. $\sum_{i=1}^{n} x_i z_i = 0$
3. $\alpha(\mathbf{x} + \mathbf{y}) = [6\ 10\ 14]^T$
4. $\|\mathbf{x}\| = \sqrt{5}$
5. $\mathbf{x}^T = [0\ 1\ 2]$
6. $A\mathbf{x} = [6\ 5\ 7]^T$
7. $\mathbf{x}^T A\mathbf{x} = 19$

1.2 Matrix Algebra Rules

1. True
2. True
3. False
4. False
5. False
6. True
7. False
8. True
9. True

1.3 Matrix Operations

1. $\mathbf{B}$ is not invertible.

2. $\mathbf{B}$ is diagonalizable. Let $\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ be the diagonalization of $\mathbf{B}$.

2 Taking Chances: Probability Review

2.1 Basic Probability

1. $\mathbb{E}[X] = \$5$

2. $P(B) = 0.55$

3. $P(B) = 0.91\bar{6}$

2.2 Expectations and Variance

1. $\mathbb{E}[X] = 0.6$

2. $\mathbf{Var}[X] = 0.72$

3. $\mathbb{E}[Y] = 0.406$

2.3 A Variance Paradox

$X_1, ..., X_n$ are independent random variables, so their variances add directly. In the case of $X+X$, only one random variable is involved. This scales the variance such that $\mathbf{Var}[X+X]\mathbf{Var}[2X] = 4\sigma^2$. Therefore, there is no contradiction.

3 Calculus Review

3.1 One-variable derivatives

1. $f'(x) = 6x - 2$

2. $f'(x) = 1 - 2x$

3. $f'(x) = 1 - \frac{1}{x}p(x)(1 - p(x))$

3.2 Multi-variable derivatives

1. $\nabla f(\mathbf{x}) = (2x_1, e^{x_2})$

2. $\nabla f(\mathbf{x}) = (e^{x_1+x_2x_3}, x_3e^{x_1+x_2x_3}, x_2e^{x_1+x_2x_3})$

3. $\nabla f(\mathbf{x}) = (a_1, a_2)$

4. $\nabla f(\mathbf{x}) = (4x_1 - 2x_2, -2x_1 + 2x_2)$
5. $\nabla f(\mathbf{x}) = \mathbf{x}$

4 Algorithms and Data Structures Review

1. $O(n\ log\ n)$
2. $O(n)$
3. $O(log\ n)$
4. $O(1)$
5. $O(nd)$
6. $O(d^2)$
7. $O(mnd)$

5 Programming

5.1 PyTorch Basics

N/A

5.2 Sentiment Classification with PyTorch and Word Embeddings

5.2.1 Data Loading and Splits

N/A

5.2.2 Word Embeddings: Representing Meaning in a Computer

N/A

5.2.3 String to Feature: Featurizing Input Text with Word Embeddings

See code.

5.2.4 Dataset and Dataloader

See code.

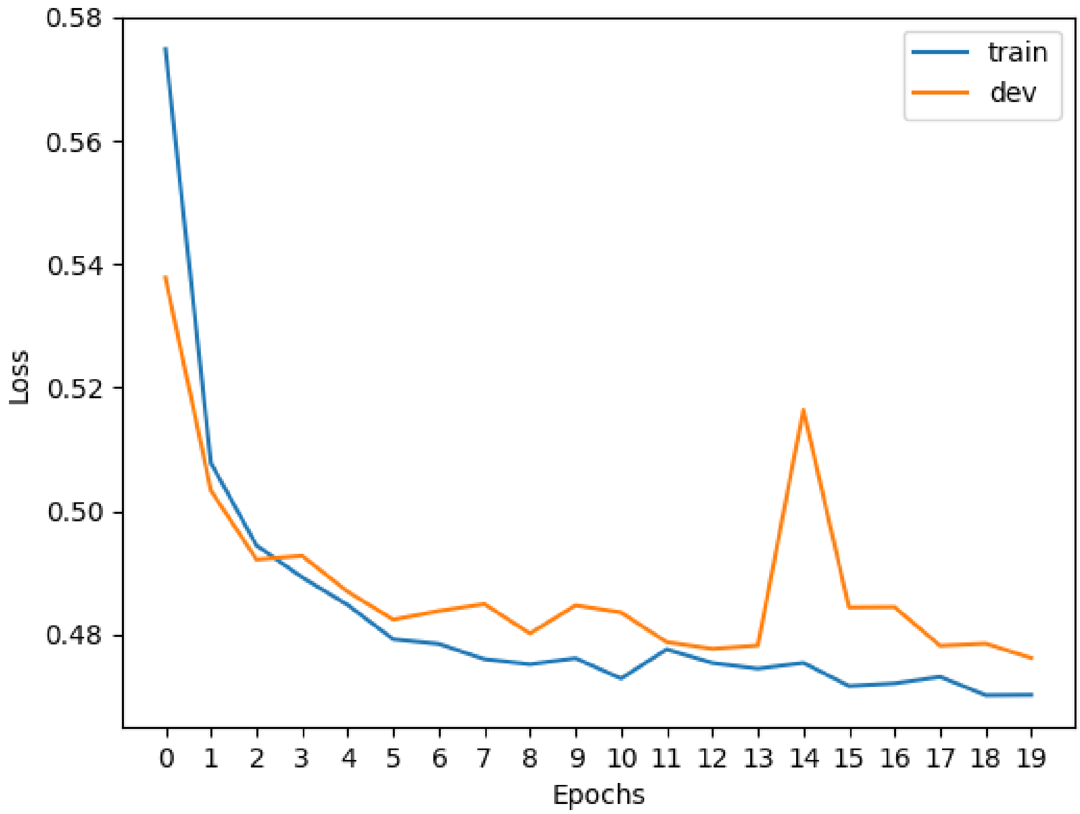
5.2.5 Defining our First PyTorch Model: nn.Module

See code.

5.2.6 Chain Everything Together: Training and Evaluation

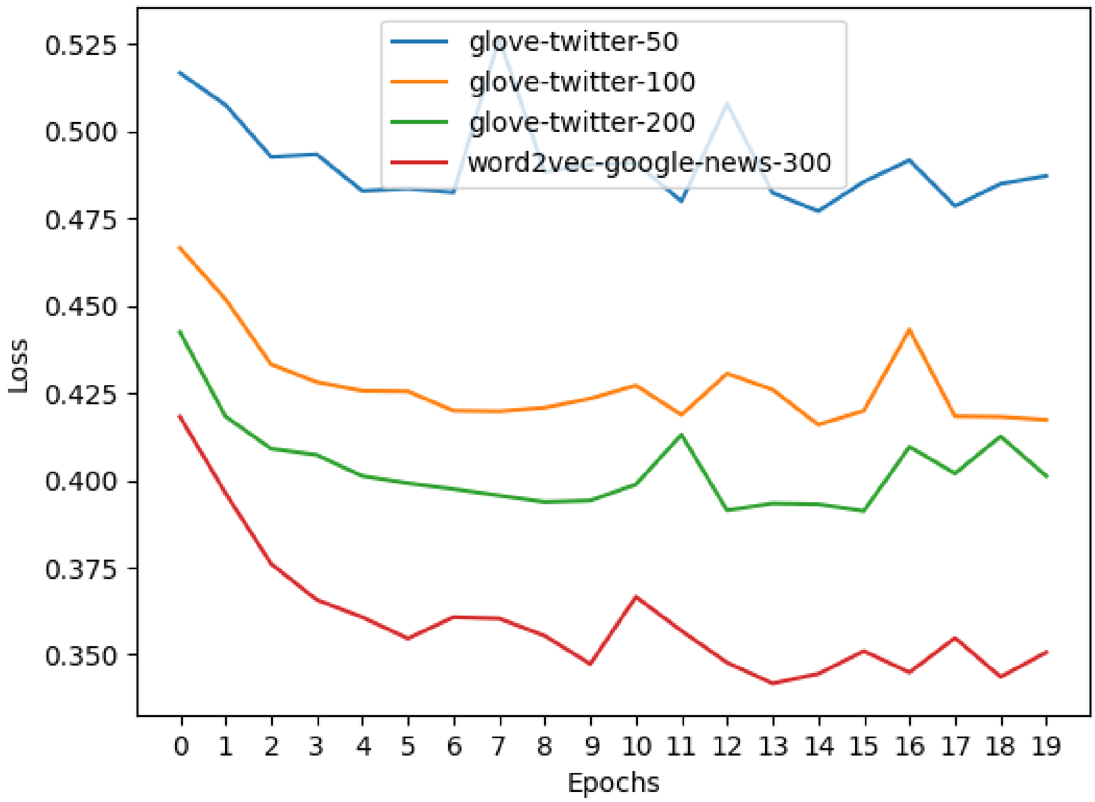
See code.

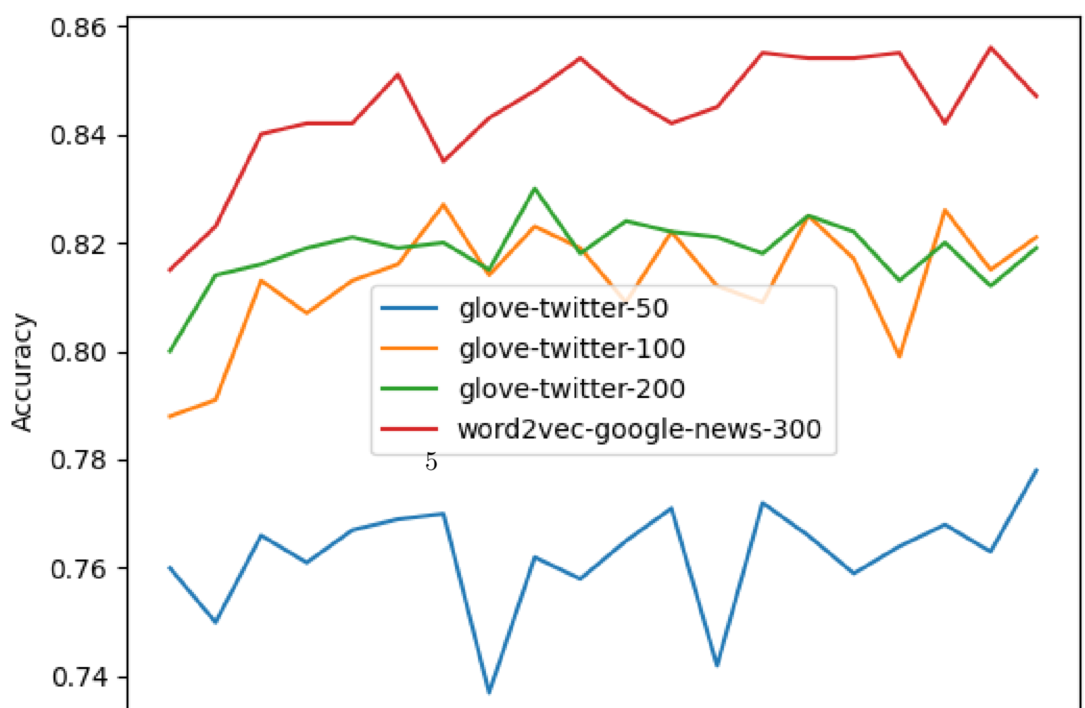
5.2.7 Run the pipeline: Train Loss vs. Dev Loss



The train and dev loss are both (generally) decreasing. In addition, both seem to be approaching a final loss value around 0.47. The dev loss, though, is more jittery than the train loss. For example, there is a large spike in dev loss at epoch 14. This could indicate some level of overfitting within the model.

5.2.8 Run the pipeline: Explore Different Word Embeddings





The `word2vec-google-news-300` embedding model performs best in terms of accuracy and embedding loss. `glove-twitter-50`, on the other hand, was the worst performer. The model discrepancy could be related to differing embedding sizes, differing dataset sizes, and/or differences in dataset robustness. For example, the `word2vec-google-news-300` dataset contains 100 billion words, while the `glove-twitter-50` embedding model was trained on 27 billion tokens. Also, it is plausible that the Google News data are more robust than a dataset of tweets.

6 Optional Feedback

In order to successfully run the provided code, I had to specifically install `scipy==1.12`. This was not included in `requirements.txt`.