Project 1: Word Embeddings

SW03 – Jannine Meier – FS24

Preprocessing

”John moved the couch from the garage to the backyard to create space. The _ is small.”

Text Normalization

- converte to lowercase
- remove punctuation

Stopword Removal

- remove stopwords
  - «the», «is», «a», «not», «by», «of» etc.

”john moved couch garage backyard create space _ small”

Network Architecture: Word Embedding

Word2Vec: word2vec-google-news-300

Integrating Dataset with PyTorch for Evaluation

1. Sentence and Options Extraction
2. Sentence Replacement
3. *Vectorization*
4. Stacking and Labeling
   - returns tuple with tensor (stacked vectors) and label

Convert Sentence to Vector

1. Tokenization
2. Vector Accumulation
3. Averaging & Normalizing
   - returns single vector

Network Architecture: Classifier

ComparativeClassifier

Sequential Model Layers

1. Linear Layer: Resize
   - input_size to hidden_layer_size
2. ReLU Activation: Learn
   - non-linearity for complex patterns
3. Linear Layer: Transform
   - hidden_layer_size to single scalar

Forward Pass

- Input
  - sentence pair vectors tensor

1. Vector Separation
2. Scoring
3. Comparison

- Output
  - score difference tensor

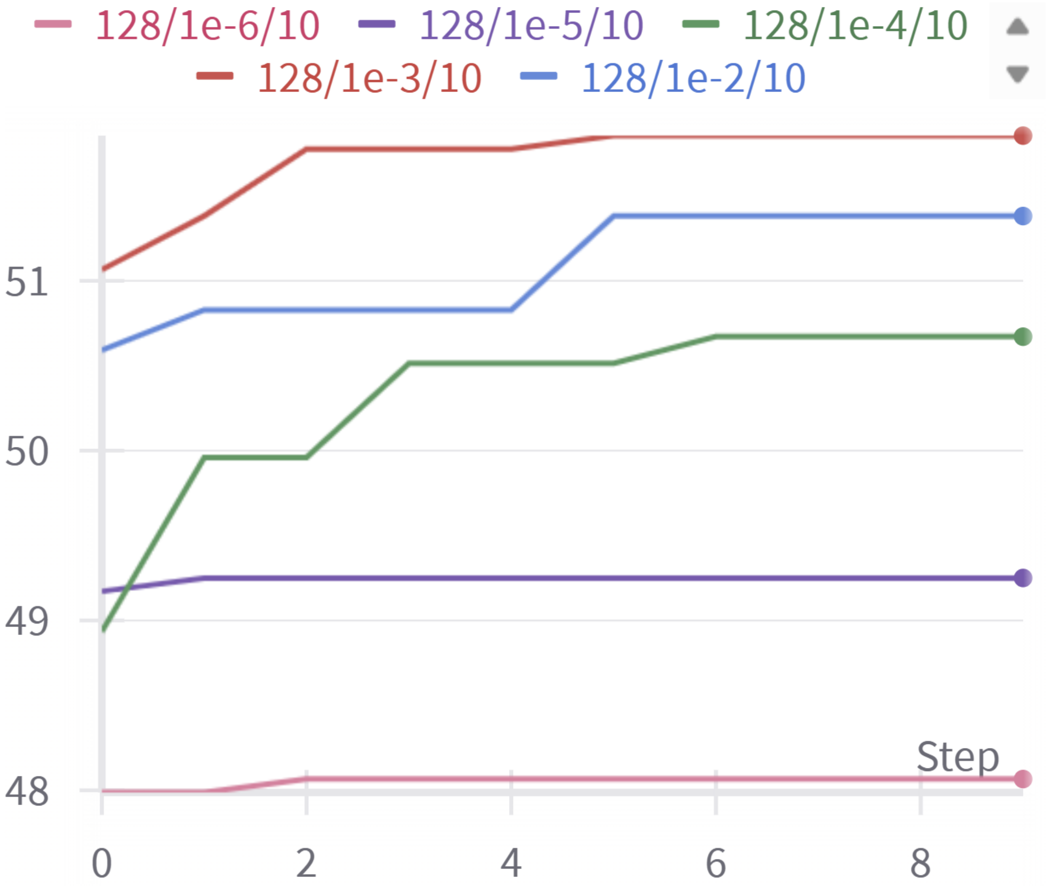
Experiments

Start

- Testing Preprocessing
  - lowercase
  - punctuation
  - remove stopwords
  - stemming / lemmatization

- Short Epochs (<=10)
- Fixed Hidden Layers (128)
- Tuned Learning Rate
  - 1e-2 -> 1e-6

Best Validation Accuracy: 51.86%



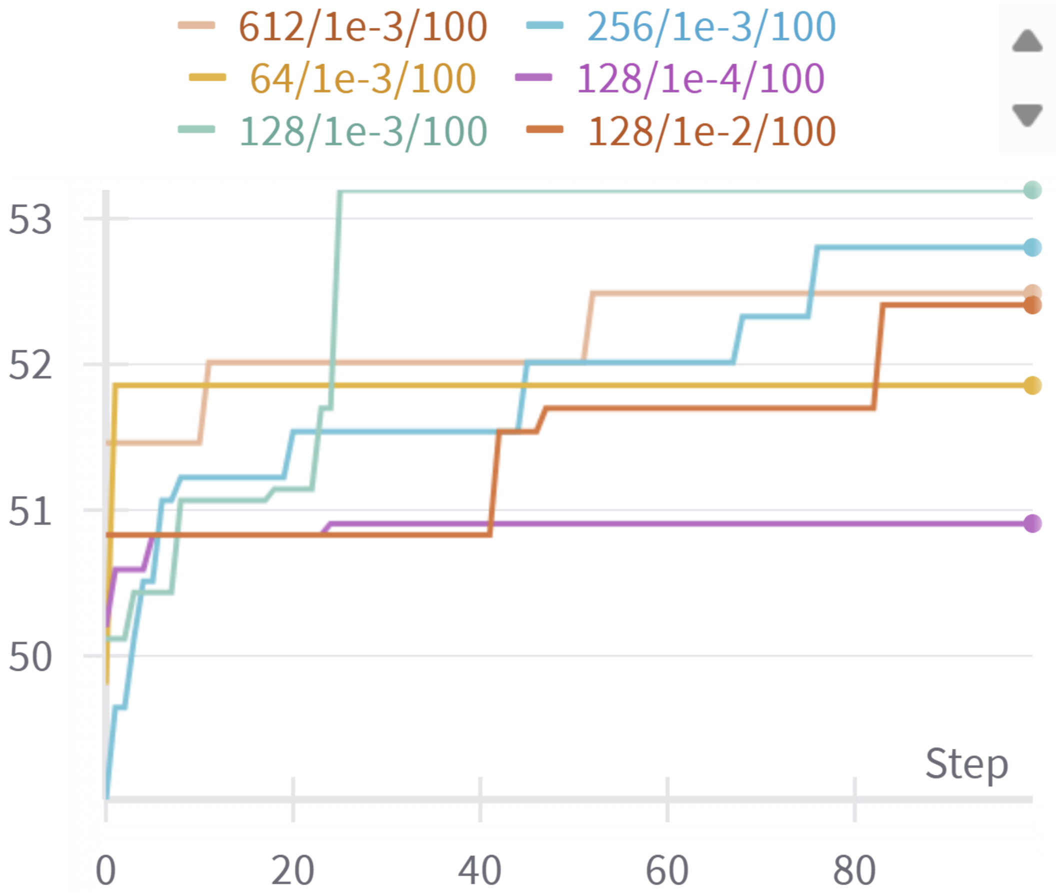
Experiments

Continuation

- Fixed preprocessing
    - lowercase -> yes
    - punctuation -> yes
    - remove stopwords -> no

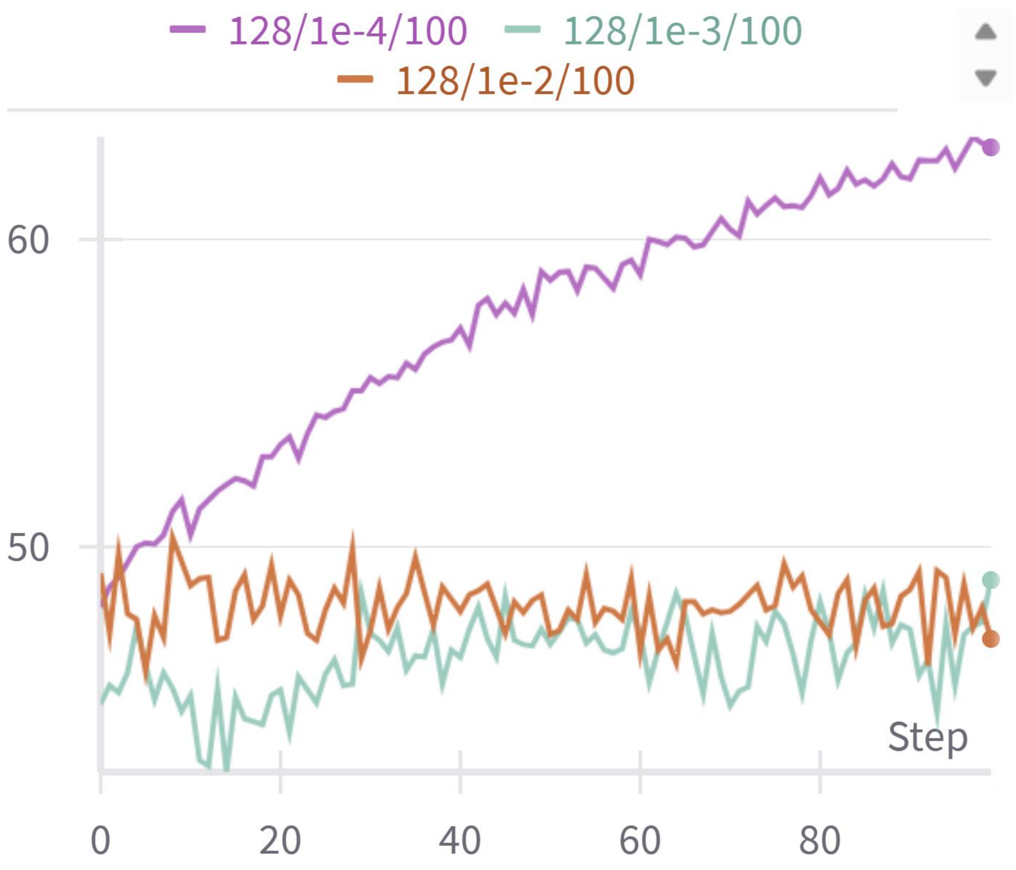
- Large(r) Epochs (>=100)
- Tuned Hidden Layers
    - 64 -> 512
- Tuned Learning Rate
    - 1e-2 -> 1e-4

Best Validation Accuracy: 53.12%

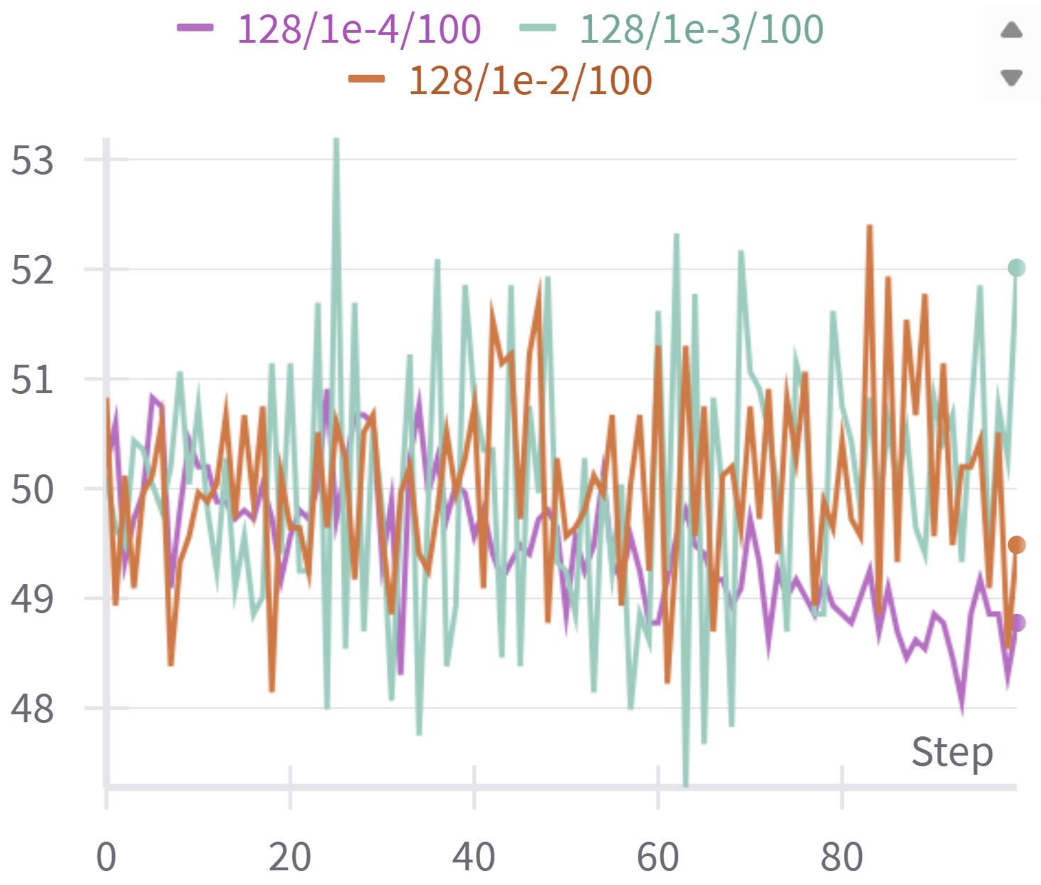


Experiments: Learning Rate Overfitting

Train Accuracy



Validation Accuracy



Results

My Best Configuration

- hidden_dim: 128
- num_epochs: 100
- learning_rate: 1e-3
- batch_size: 32
- preprocessing:
  - lowercase: True
  - remove_punctuation: True
  - remove_stopwords: False

Interpretation of Results

- marginally better than random
  - 50% chance by guessing
- overfitting concerns
  - when using small learning rates
- underfitting concerns
  - both train and validation accuracies are low
  - complex task

Key Take-aways

Increase Efficiency

- choose right device (GPU)
  - shorten run time
  - run longer epochs
- automate tuning of hyperparameters

Proper Logging

- plan what to log
  - train, evaluation, test
  - (best) accuracy, loss
- start logging from start on
- arrange logs structured
  - adjust view/axis of graphics
  - rename runs