Learning to Label Stack Exchange Questions

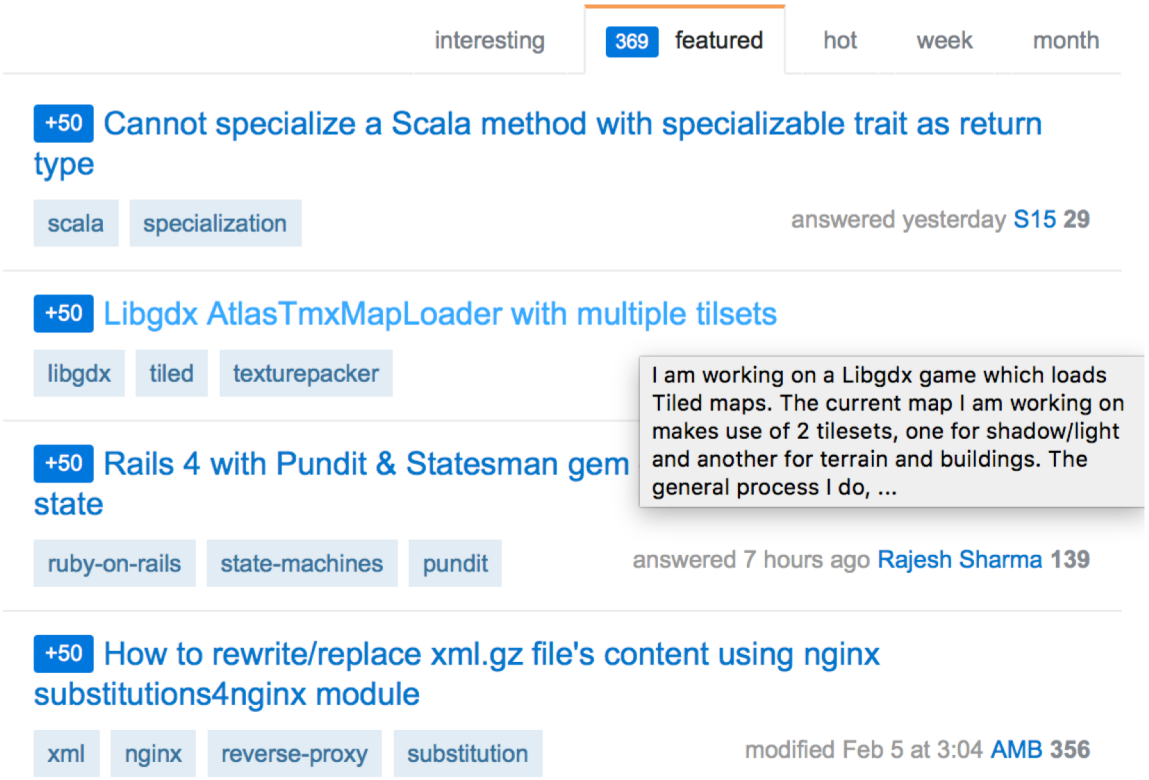
Rafael Leano

Jose Picado

Zahra Iman

Problem

- Q/A sites get thousands of questions daily
- Questions must be labeled for easy retrieval



Task

- Automatically assign labels to questions using deep learning and natural language processing

How to efficiently iterate over each Entry in a Map?

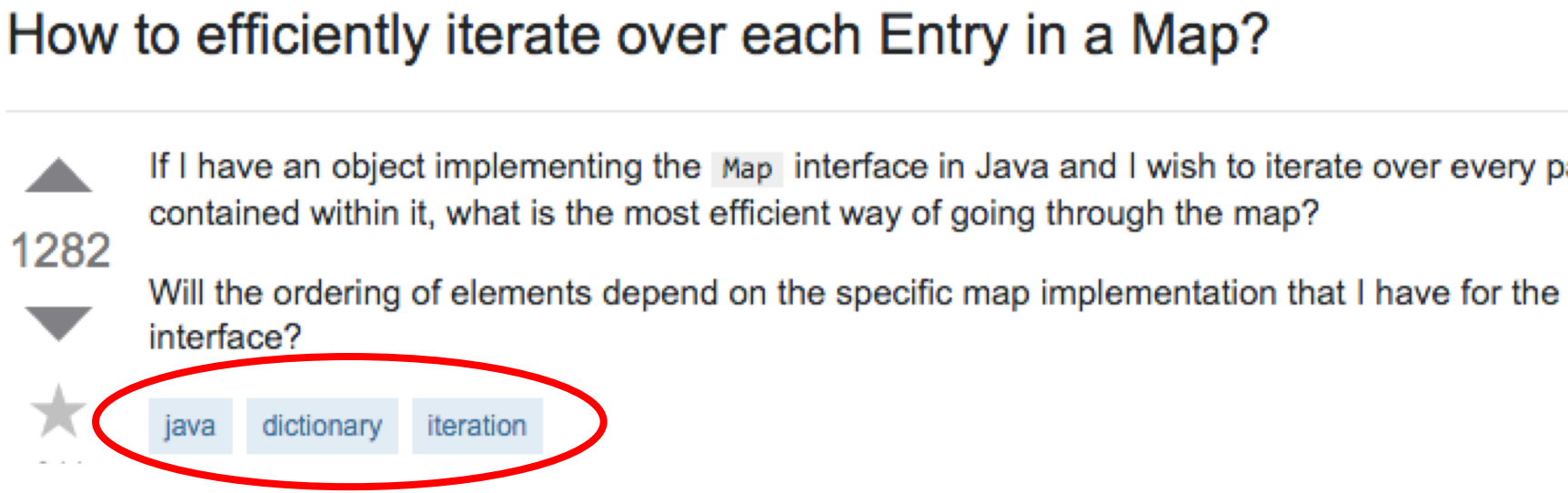
1282

If I have an object implementing the `Map` interface in Java and I wish to iterate over every pair contained within it, what is the most efficient way of going through the map?

Will the ordering of elements depend on the specific map implementation that I have for the interface?

Task

- Automatically assign labels to questions using deep learning and natural language processing

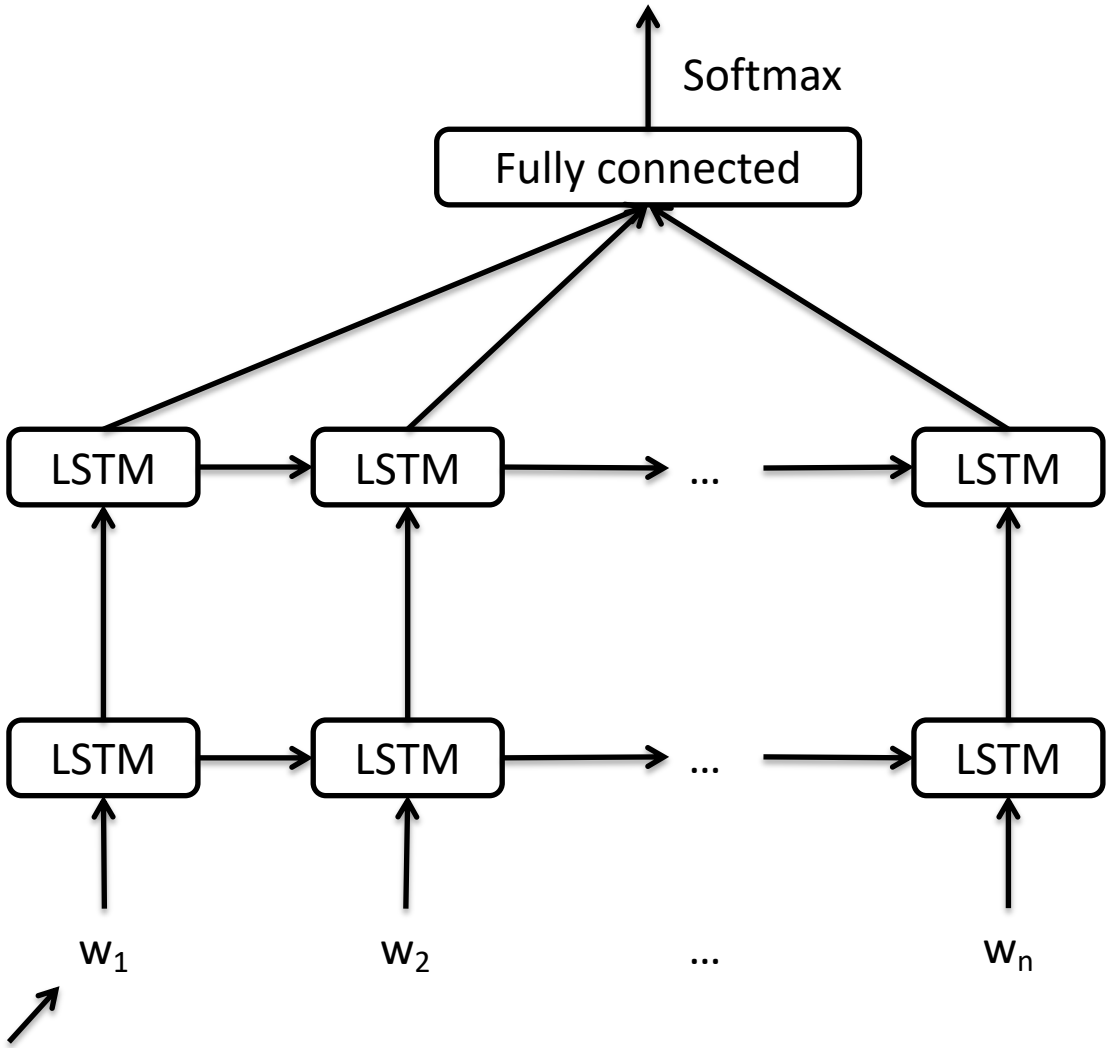


Approach

1. Convert title and questions to word vectors
   - GloVe: Global Vectors for Word Representation*
   - Pre-trained word vectors on Wikipedia 2014 + Gigaword 5
   - Dimension of vectors: 50
2. Train a long-short term memory (LSTM) model
3. Predict multiple labels for each example

*http://nlp.stanford.edu/projects/glove/

Architecture



Data

- Stack Exchange data* (~7GB):
  - Id, title, body, tags
- Statistics:
  - Questions: 6,034,195
  - Unique tags: 42,048
  - Tags/question: 1-5
  - Average no. of tags/question: 2.89

*https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/data

Reduced data

- Reduce number of tags:
  - Keep top 10 tags
  - Keep top 100 co-occurring tags with each top tag
  - Use only questions containing these tags
  - Remove stopwords, numbers and punctuation
- Statistics:
  - Questions: 1,220,004
  - Unique tags: 573
  - Maximum question length: 2694

First approach

- Input data:
  - 1,220,004*2694*50 = 164,334,538,800
- Process in batches, reduce question size:
  - Batch size: 10,000 questions
  - Maximum question length: 512
- No improvement after each batch

Even more reduced data

- Reduce number of tags:
  - Keep top 10 tags
  - Keep top 10 co-occurring tags with each top tag
  - Use only questions containing these tags
- Statistics:
  - Questions: 582,331
  - Unique tags: 72
  - Maximum question length: 2694

Data used in experiments

- Questions: 200,000
- Unique tags: 72
- Maximum question length: 400

jquery	json	asp.net-mvc	eclipse
javascript	jquery-ajax	xml	spring
html	c#	wcf	hibernate
ajaxcss	.net	python	multithreading
php	winforms	java	jsp
jquery-ui	wpf	android	java-ee
asp.net	linq	swing	…

Baseline: N-grams + Logistic Regression

- Extract n-grams from title and body (cleaned text)
- Train a one-vs-rest classifier using Logistic Regression
- Implemented using scikit-learn (Python)

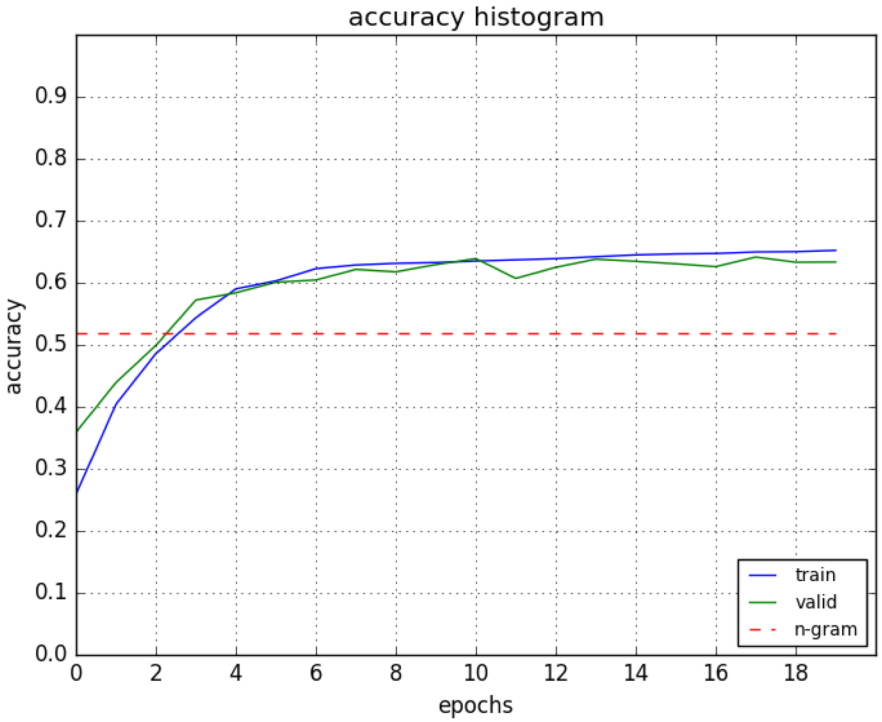
N	Number of features
1-gram	1,171,964
{1,2}-gram	5,551,226

Results

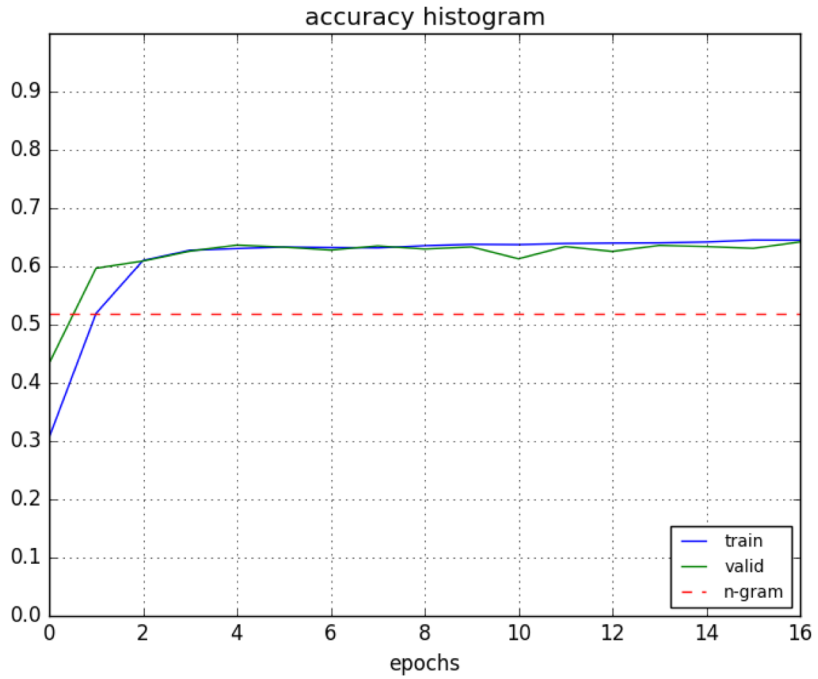
- Training set: 180,000 questions (90%)
- Validation set: 20,000 questions (10%)
- Deep models:
  - 128 batch size
  - 18 epochs
  - Rmsprop
- Subset accuracy:
  - the set of predicted labels must *exactly* match the true set of labels

Method	Validation Accuracy	Categorical Loss Entropy
1-gram + Logistic Regression	0.487	16.66
{1,2}-gram + Logistic Regression	0.518	16.09
2 layer LSTM	0.642	3.32
3 layer LSTM	0.641	3.48

Results (accuracy)

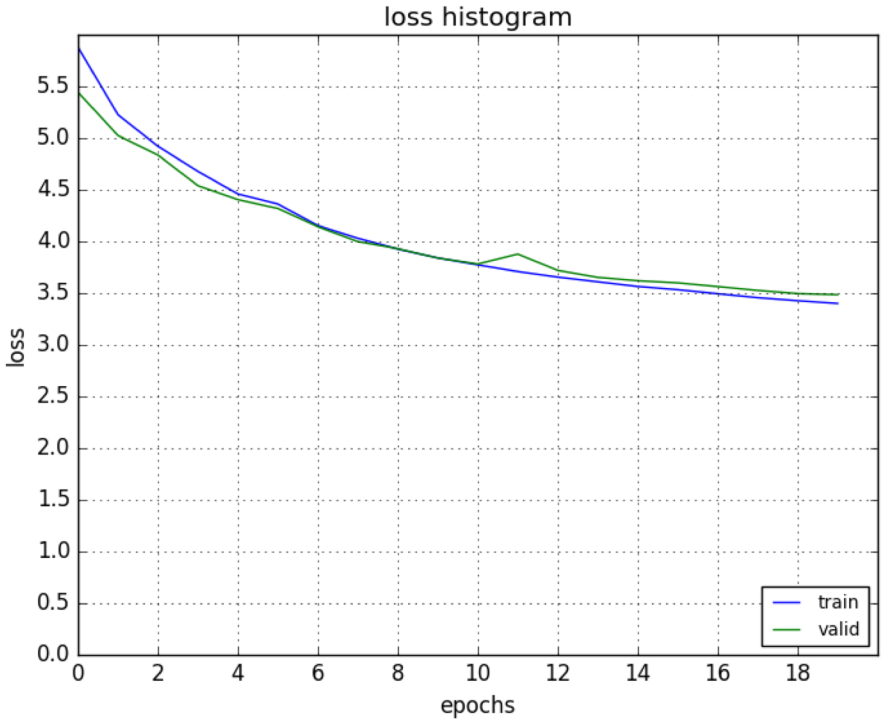


Questions: 100,000
3-layer LSTM

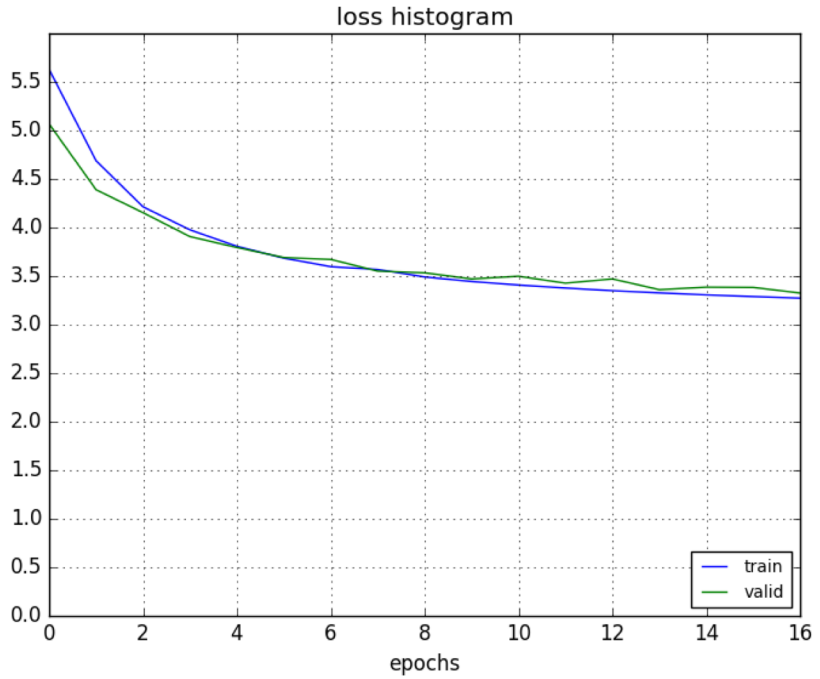


Questions: 200,000
2-layer LSTM

Results (categorical loss)

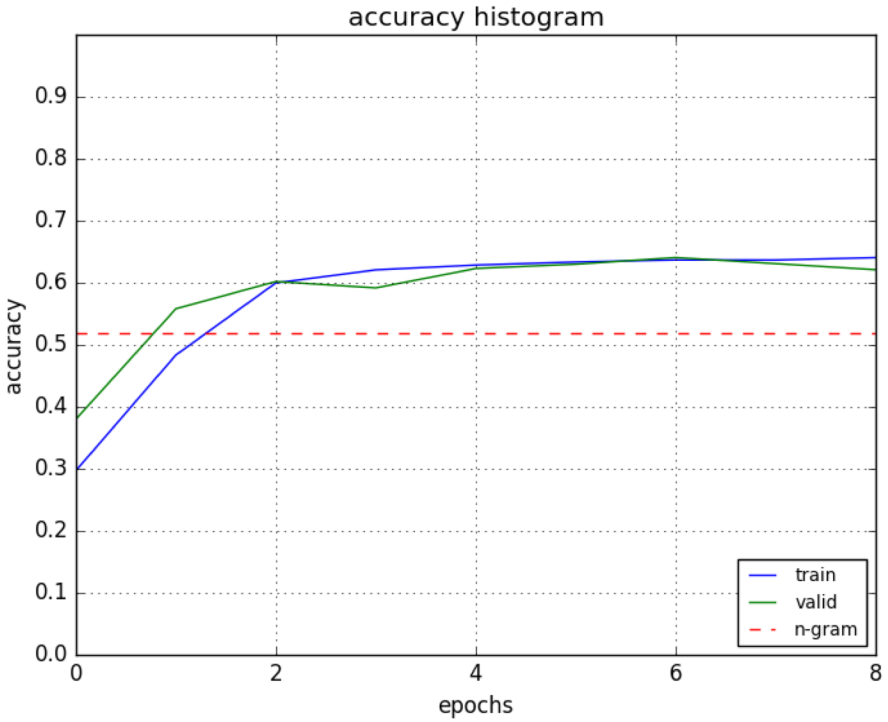


Questions: 100,000
3-layer LSTM

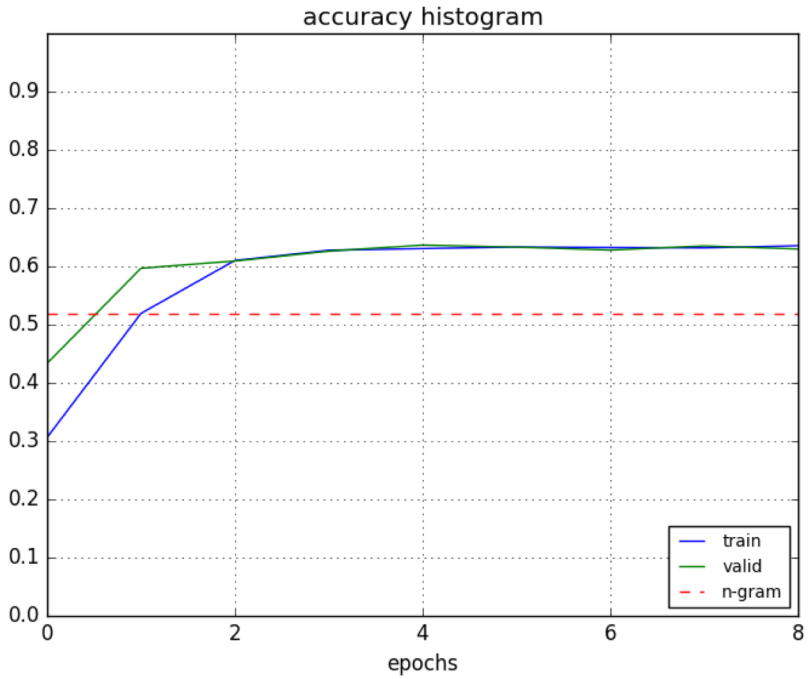


Questions: 200,000
2-layer LSTM

Results II (accuracy)

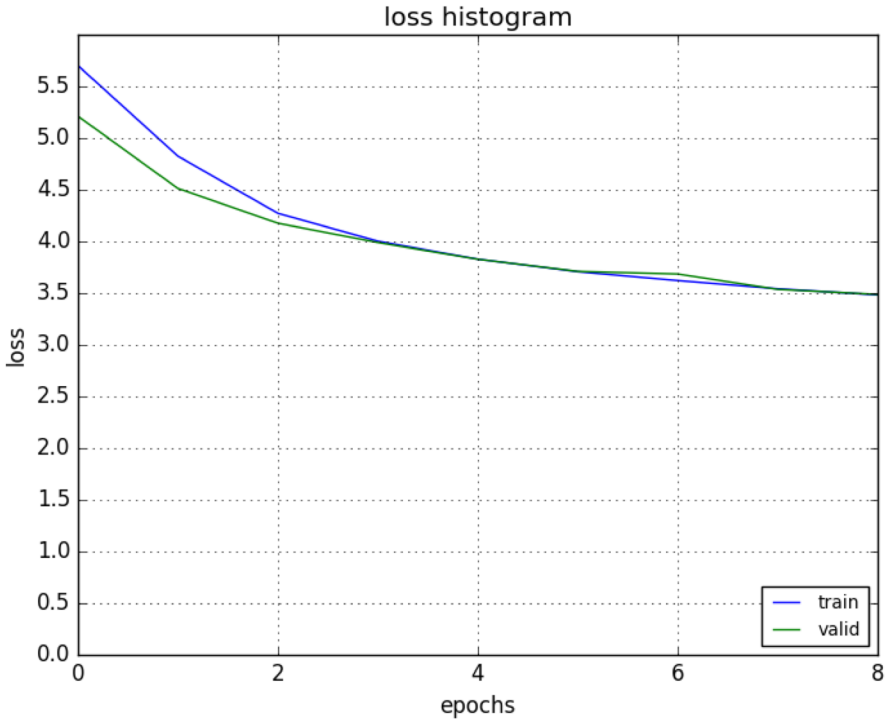


Questions: 200,000
3-layer LSTM

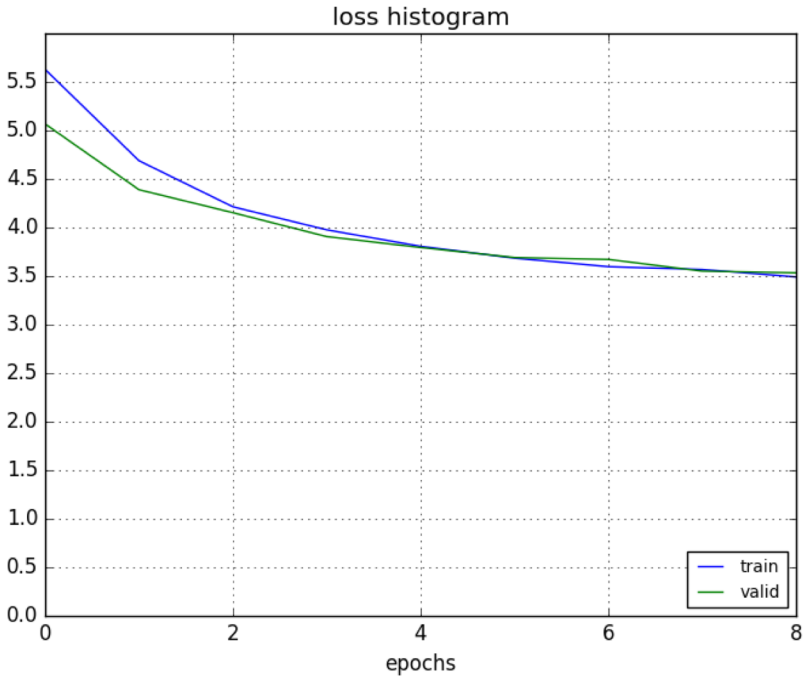


Questions: 200,000
2-layer LSTM

Results II (categorical loss)



Questions: 200,000
3-layer LSTM



Questions: 200,000
2-layer LSTM

References

- S. Hochreiter, J. Schmidhuber. Long short-term memory. Neural computation, 2007.
- J. Pennington, R. Socher, C. Manning. GloVe: Global Vectors for Word Representation, 2014.
- X. Zhang, J. Zhao, Y. LeCun. Character-level Convolutional Networks for Text Classification. ArXiv: 1509.01626v2, 2015.
- LSTM Networks for Sentiment Analysis. deeplearning.net.