

CS 280r Final Project

Yuntian Deng, Matthew Finlayson

Segment design

Acknowledgements to Peniel Argaw, Zilin Ma, and Andrew Tran

Prerequisites

Conditional probability (for Markov Chains/HMMs) Basic Neural Networks (feedforward networks, backprop, loss functions, etc.)

Course-wide learning objectives

- Linguistics (L): Model the standard levels of linguistic structure using formal grammars or structured statistical and computational models.
- Experimental methodology (EM): Identify and carry out proper experimental methodology for training and evaluating NLP systems.
- Machine learning (ML): Manipulate probabilities and estimate parameters of structured models using supervised training methods.
- Implementation (I): Implement simple models of language, and employ and adapt them in service of solving NLP problems.

Segment Learning objectives

Sequence task understanding

- [S1] Manually tag parts of speech in a sentence. (L)
- [S2] Describe the limitations of treating sequence labelling problems as a classification problem (the need for context/too many combinations to treat each sequence as a label) (L)

N-grams

- [N1] Describe how n-gram can be applied to text generation.(L)
- [N2] Describe in detail how an n-gram model works. (ML)
- [N3]Estimate a transition table an n-gram model. (ML)
- [N4]Evaluate a language model using perplexity. (EM)
- [N5] Use Laplace smoothing to improve language model performance. (ML)

HMMs

- [H1] Describe how HMM can be applied to text generation and slot filling. (L)

- [H2] Determine the most likely state sequence given an HMM model and sequence of observations using the Viterbi algorithm. (ML)
- [H3] Implement an HMM for slot-filling. (I)
- [H4] Estimate the parameters of an HMM using the forward-backward algorithm. (ML)

RNNs

- [R1] Describe how RNN can be applied to text generation and slot filling. (L)
- [R2] Implement in Python the forward step of an RNN for slot-filling. (ML)
- [R3] Use PyTorch to implement an RNN for slot-filling (I)
- [R4] Explain the details of backpropagation for an RNN. (ML)
- [R5] Explain the benefits of using RNN over HMM on part-of-speech tagging. (I)

LSTMs

- [L1] Explain at a high level how an LSTM works. (ML)
- [L2] Explain why an LSTM improves on an RNN. (ML)
- [L3] Implement an LSTM slot filler using PyTorch. (I)
- [L4] Evaluate and report the performance of HMM, RNN, and LSTM models using standard techniques. (EM)

Learning activities

Instruction in this segment will be broken down into 3 phases: "I do," "we do," and "you do," where "I" refers to the instructor, and "you" refers to the students. During the "I do" phase, students observe and read about the task they will perform. This gives them the necessary background knowledge to attempt the next step. During the "we do" phase, students will be guided through the task with the help of teaching staff. This is where the majority of the learning should occur. During the "you do" students will be asked to apply what they have learned by themselves. This phase is mostly for assessment purposes but does provide learning opportunities.

"I do"

- Readings: In the readings from Jurafsky & Martin students will learn about the n-gram, RNN, and LSTM models and their specific implementations.
- Mini-lectures: During the beginning of class, the instructor will motivate the in-class activities with presentations of real-world problems.

"We do"

- In-class activities: Students will be required to participate in in-class activities where they will gain first hand experience applying the methods learned in the readings under the close guidance and supervision of the instructor. These activities will prepare students for the

segment project.

- Office hours: These will be optional opportunities for students to ask their own questions and get one-on-one help from teaching staff.

"You do"

- Segment Project: Students will be required to complete a segment project where they will apply what they learned in class and in the readings in an unsupervised setting either alone or with a partner. The purpose of the project is to both assess the student's grasp of the material and to give further learning opportunities as they encounter and overcome obstacles.
- Quizzes and exams: These activities will mainly be designed to assess the student's conceptual understanding of the material, and will not require a high level of very specific technical knowledge of, say, the inner workings of RNN backpropagation.

Assessments

"I do"

- Readings - Students will receive full marks if they complete the readings and participate in online discussions of the reading. Completion and participation are measures of engagement with the material and preparation for the in-class activities. In class, the students will depend on each others' collective understanding of the material from the readings to complete the activities.
 - The instructor will provide a platform like us.edstem.org to facilitate discussion of the readings.
- Mini-lectures - There will be no formal assessment for students' attendance at to the mini-lectures at the beginning of class, but students who do show up and listen will probably do better in other aspects of the class and may receive more lenient grading from the instructor.

"We do"

- In-class activities - participation, completion
 - For in-class coding activities, student's effort and understanding will be assessed through automated unit tests of their code, at least half of which they will be expected to pass. Half correct may seem like a low bar, but correctness is not so much the point as participation. Students will therefore get some credit for submitting even incorrect/partially correct answers.
 - For in-class activities done by hand on paper, students will be assessed based on their ability to attempt all the questions. The hope here is that students will get their feet wet trying out the activities without the stress of getting everything exactly right.
 - For any type of in-class activity, students will be allowed to complete the activity at home or after class and turn it in up to a week after receiving it.

"You do"

This phase of the segment is the most focused on assessing students' fulfillment of the course objectives.

- Project - correctness, code design, experimental design, style.
 - Students' fulfillment of the course objectives relating to machine learning, implementation, and experimental design will be assessed through their delivery on the segment project. Students can demonstrate their fulfillment of these objectives through correct implementations of ML models, proper experimental procedures, and the overall design, style, and correctness of their code.
- Segment quiz/exams - demonstration of understanding
 - The quiz will focus on assessing students' grasp of linguistic concepts, and their understanding of the differences and workings of ML models. Tasks students will be expected to complete in the segment quiz include deriving models, writing pseudocode, explaining the reasoning for the application of different models, and performing sequence labelling tasks by hand such as POS tagging.

High level schedule

	N-gram/Markov	RNN/LSTM
LMs	N-gram	RNN
Sequence labeling	HMM	RNN, LSTM

Based on the relative difficulty of understanding different computational methods compared to understanding the tasks these methods solve, the classes will be split up by method. Within each class, applications of each method for LMs and sequence labeling will be discussed.

Part 1: Probabilistic models. Classes 1-3 will be spent focusing on understanding, implementing, then evaluating N-gram language models and HMMs for sequence labeling.

Part 2: Neural networks. Classes 4-5 will be spent on activities to help prepare students to understand, implement, and evaluate neural networks, RNNs, and finally LSTMs at a high level and then applying them as language models and then to sequence labeling tasks.

Class 1: N-gram models for language modeling, perplexity

Learning Objectives

N1, N2, N3, N4, N5

Mini-lecture

What is language modeling? Why is it useful? In the mini-lecture we will talk about language modeling, sequence labeling, evaluation metrics, give a preview of the methods that will be used to complete the tasks.

Learning activity: Implement and evaluate

Students implement an N-gram model in a pair-programming exercise to predict the next word in a sequence. They evaluate their models on the basis of perplexity, which necessitates Laplacian smoothing when there are unseen words at test time.

Reading

J&M Chapter 3

Class 2: HMMs for sequence labeling

Learning Objectives

H1, H2, H3, S1, S2

Mini-lecture

What is sequence labeling? Why is it useful? In this class we will cover slot-filling, part of speech tagging, HMM, Viterbi

Learning activity: manually tagging sentences

Students manually tag 5 English sentences, explain and investigate reasons for their decisions.

Learning activity: DIY Viterbi

Using an concrete icecream example, students apply Viterbi algorithm for efficiently finding the most likely sequence of labels given transition and emission tables. Then students will derive the general form of this algorithm.

Readings

J&M Chapter 8

Class 3: HMMs for sequence labeling

Learning Objectives

H4

Mini-lecture

Why would we need a forward-backward algorithm for HMMs?

Learning activity: forward-backward algorithm to learn emission and transition tables

Students perform by hand examples of calculations required to perform the forward-backward algorithm to learn the state transition table and the emission table for a toy version of Eisner's ice cream example.

Readings

J&M Appendix A

Class 4: RNNs for language modeling

Learning Objectives

R1, R2, R4

Mini-lecture

What are the shortcomings of n-gram language models? How do RNNs present a solution?

Learning activity: RNN forward

Students perform by hand examples of calculations associated with the forward and backward step for an RNN. Refer to worksheet for more details.

Readings

J&M Chapter 9 up to and including 9.2

Class 5: RNNs and LSTMs for sequence labeling, LSTMs for language modeling

Learning Objectives

R1, R3, R5, L1, L2, L3, L4

Mini-lecture

How can RNNs be applied to sequence labeling tasks? What are the shortcomings of RNN language models? How does an LSTM improve on an RNN?

Learning activity: LSTM

Students calculate the forward step by hand for a tiny LSTM. Students then plan and pair-program to implement the forward step for an LSTM in Python.

Readings

J&M Chapter 9.4 to end

Class 6: Wrapping up

Learning Objectives

L5

Mini-lecture

Summarize the segment, compare HMM with RNN/LSTM on language modeling/sequence tagging and look ahead.

In-class activity

Segment Quiz

Design for class #4

Learning objectives

After this class students should be able to:

- Give an intuitive explanation of how an RNN works.
- Compare the advantages of an RNN over an n-gram model or HMM.
- Identify potential applications of RNNs.
- Explain in detail how a recurrent neural network works.
 - Work out the equations associated with the forward step of an RNN.
 - Work out the equations associated with backpropagation for a neural network.

Assessments

Students will be graded for participation, not correctness in the in-class activities.

Reading materials

In preparation for this class, students should read J&M chapter 9 up to and including 9.2.

Class schedule

Time	Activity
00:00	Announcements
00:05	Opening remarks: Motivation for RNNS, review of RNN structure
00:15	Activity 1
00:25	Suggest moving to activity 2
00:40	Suggest moving to activity 3
00:65	Closing remarks: discuss solutions, implicaitons, preview LSTMs

Mini-lecture

The following are some drawbacks to using n-gram language models.

- Sequences not encountered in training get minimum probability.
- Curse of dimensionality - huge number of possible sequences of words must be accounted for
- Rely on sequence matching - no linguistic basis
- Limited view to n sized window

RNNs present an improvement over n-gram language models.

- Take into account a large number of previous inputs
- Deal well with sequences not previously encountered

Note that there are two kinds of problems above: the ability of the model to deal with unseen sequences and the ability of the model to memorize previous inputs. While RNN is able to solve both of these problems, a neural version of n-gram language model can solve the unseen sequences problem as well through using word embeddings.

In-class activities

Please refer to our handout for details. There are mainly four parts.

- Compute the forward step for a toy RNN.
 - Builds on feed-forward networks from classification segment.
 - Prepares for class 5 activity on LSTMs.
- Design an RNN to behave like a bigram model.
 - Builds on n-gram models from class 1.
- Derive the formula to find the partial derivative of the loss function with respect to a parameter of an RNN.
 - Builds on feed-forward networks from classification.
 - Builds on forward-backward algorithm for HMMs in class 3.
- Find the partial derivative of the output of an RNN with respect to a parameter of an RNN.

- Builds on feed-forward networks from classification.

Reflection

CS280

The good

We felt that the small group work was important for the design process. Rotating and changing the makeup of the groups brought together people with different experience and expertise in a helpful way. In class discussions were also very helpful, especially the guest speakers at the beginning to frame the concept of backward design.

We liked that the class built up each segment incrementally from a high-level view to the finer details.

Individual group meetings and group work sessions were helpful for in-person collaboration and for unifying the vision for what we were building across all groups.

Could be done better

Sometimes the discussions seemed to get stuck. We think training students on cooperative discussion and listening might help move things along better. A longer class time might be appropriate.

One shortcoming of the group rotations was that new groups sometimes did not have the materials from previous groups working on the same segment immediately available. Perhaps a Github repository for each segment would have been helpful, as well as a dedicated Slack channel for each group, so previous discussions would be available to new group members. Inter-group collaboration would also have been easier if each group had full access to the other groups' work.

Students seemed timid about collaborating publicly through Slack. Perhaps this could have been fixed by setting expectations and norms about Slack earlier on in the class.