Announcements

- A4, A5 grading underway
- Final project check-ins due **November 23**
- Final projects due **December 9**

Recap

- Dataset artifacts / spurious correlations
  - Single-word correlations in NLI: hypothesis contains *not* -> contradiction
  - Answer type bias in QA: *where* -> return any reasonable location
- Various debiasing techniques:
  - Understand what examples are contributing to the bias
  - Reweighting training data to remove those examples
  - Data augmentation (not discussed)

Today

- Interpreting neural networks: what does this mean and why should we care?
- Local explanations: erasure techniques
- Gradient-based methods
- Evaluating explanations

Interpreting Neural Networks

Interpreting Neural Networks

- Neural models have complex behavior. How can we understand them?
- QA: why did the model prefer *Stewart* over *Devin Funchess*?

QID: 1f4b668a0343453b9d4bf3edc86daf63
Question: who caught a 16-yard pass on this drive ?
Answer: devin funchess

Start Distribution

```
there would be no more scoring in the third quarter , but early in the
fourth , the broncos drove to the panthers 41-yard line . on the next play
, ealy knocked the ball out of manning 's hand as he was winding up for a
pass , and then recovered it for carolina on the 50-yard line . a 16-yard
reception by devin funchess and a 12-yard run by stewart then set up gano
's 39-yard field goal , cutting the panthers deficit to one score at
16â€"10 . the next three drives of the game would end in punts .
```

Interpreting Neural Networks

- Neural models have complex behavior. How can we understand them?
- Sentiment:

	DAN	Ground Truth
this movie was not good	negative	negative
this movie was good	positive	positive
this movie was bad	negative	negative
the movie was not bad	negative	positive

  - Left side: predictions model makes on individual words
  - Tells us how these words combine
- **How do we know why a neural network model made the prediction it made?**

Iyyer et al. (2015)

Why explanations?

- **Trust:** if we see that models are behaving in human-like ways and making human-like mistakes, we might be more likely to trust them and deploy them
- **Causality:** if our classifier predicts class *y* because of input feature *x*, does that tell us that *x* causes *y*? Not necessarily, but it might be helpful to know
- **Informativeness:** more information may be useful (e.g., predicting a disease diagnosis isn't that useful without knowing more about the patient's situation)
- **Fairness:** ensure that predictions are non-discriminatory

Lipton (2016)

Why explanations?

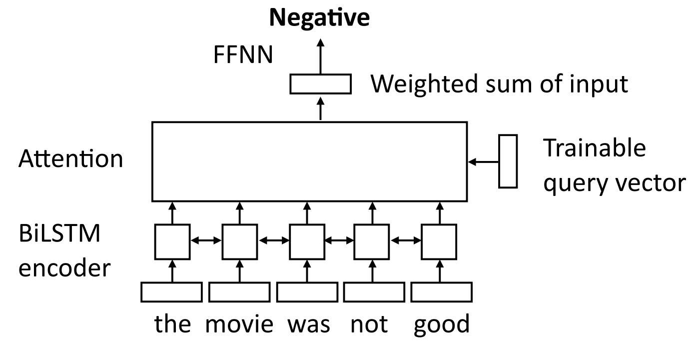
- Some models are naturally **transparent**: we can understand why they do what they do (e.g., a decision tree with <10 nodes)
- Explanations of more complex models
  - **Local explanations:** highlight what led to this classification decision. (Counterfactual: if these features were different, the model would've predicted a different class) — focus of this lecture
  - **Text explanations:** describe the model's behavior in language
  - **Model probing:** auxiliary tasks, challenge sets, adversarial examples to understand more about how our model works

Lipton (2016); Belinkov and Glass (2018)

Local Explanations

(which parts of the input were responsible for the model's prediction on this particular data point?)

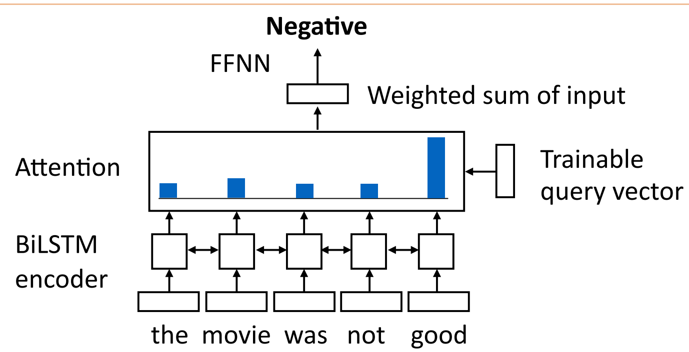
Sentiment Analysis with Attention



- Similar to a DAN model, but (1) extra BiLSTM layer; (2) attention layer instead of just a sum

Jain and Wallace (2019)

Attention Analysis



- Attention places most mass on *good* — did the model ignore *not*?
- What if we removed *not* from the input?

Jain and Wallace (2019)

Local Explanations

- An explanation could help us answer counterfactual questions: if the input were ***x***' instead of ***x***, what would the output be?

	Model
that movie was not great , in fact it was terrible !	—
that movie was not ____ , in fact it was terrible !	—
that movie was ____ great , in fact it was ____ !	+

- Attention can't necessarily help us answer this!

Erasure Method

- Delete each word one by and one and see how prediction prob changes

that movie was not great , in fact it was terrible ! — prob = 0.97

___ movie was not great , in fact it was terrible ! — prob = 0.97

that ____ was not great , in fact it was terrible ! — prob = 0.98

that movie ____not great, in fact it was terrible ! — prob = 0.97

that movie was ___ great, in fact it was terrible ! — prob = 0.8

that movie was not ____, in fact it was terrible ! — prob = 0.99

Erasure Method

- Output: highlights of the input based on how strongly each word affects the output

that movie was not great , in fact it was terrible !

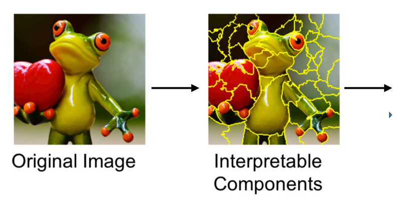
  - *not* contributed to predicting the negative class (removing it made it less negative), great contributed to predicting the positive class (removing it made it more negative)
- Will this work well?
  - Inputs are now unnatural, model may behave in "weird" ways
  - Saturation: if there are two features that each contribute to negative predictions, removing each one individually may not do much

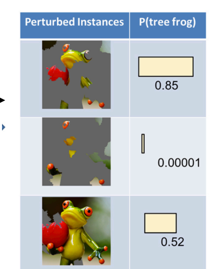
LIME

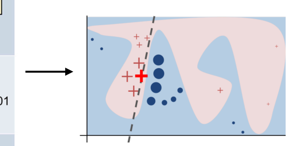
- Locally-interpretable, model-agnostic explanations (LIME)
- Similar to erasure method, but we're going to delete collections of things at once
  - Can lead to more realistic input (although people often just delete words with it)
  - More scalable to complex settings

Ribeiro et al. (2016)

LIME



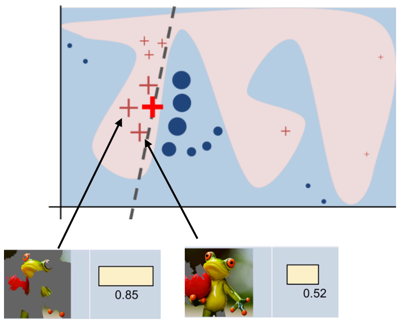




- Break input into components (for text: could use words, phrases, sentences, …)
- Check predictions on subsets of those
- Now we have model predictions on perturbed examples

https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime

LIME (cont'd)



- This is what the model is doing on perturbed examples of the input
- Now we train a classifier to predict **the model's behavior** based on **what subset of the input it sees**
- The weights of that classifier tell us which parts of the input are important

LIME (cont'd)

- This secondary classifier's **weights** now give us highlights on the input

The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%
The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%
The movie is ~~mediocre~~, ~~maybe~~ even ~~bad~~.	**Positive** 74.5%
The ~~movie~~ is mediocre, maybe even ~~bad~~.	**Negative** 97.9%

The movie is mediocre, maybe even bad.

Wallace, Gardner, Singh
Interpretability Tutorial at EMNLP 2020

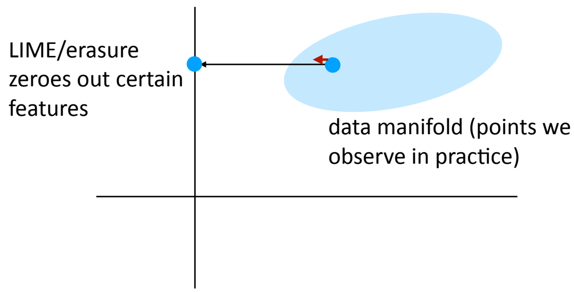
Problems with LIME

- Lots of moving parts here: what perturbations to use? what model to train? etc.
- Expensive to call the model all these times
- Linear assumption about interactions may not be reliable

Gradient-based Methods

Problems with LIME

- Problem: fully removing pieces of the input may cause it to be very unnatural



- Alternative approach: look at what this perturbation does locally right around the data point using gradients

Gradient-based Methods

score = weights * features
(or an NN, or whatever)

Learning a model

Compute derivative of score with respect to weights: how can changing weights improve score of correct class?

Gradient-based Explanations

Compute derivative of score with respect to ***features***: how can changing ***features*** improve score of correct class?

Problems with LIME

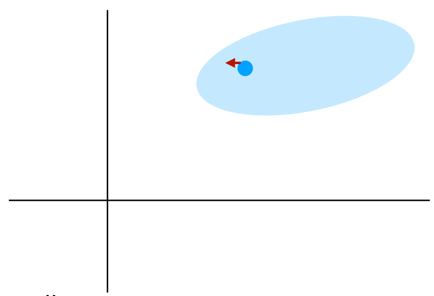
- Originally used for images

S_c = score of class c

I_0 = current image

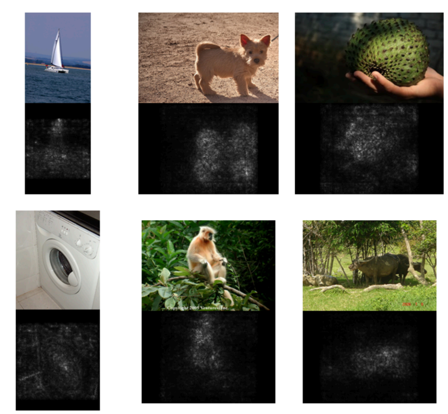
$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

- Higher gradient magnitude = small change in pixels leads to large change in prediction



Simonyan et al. (2013)

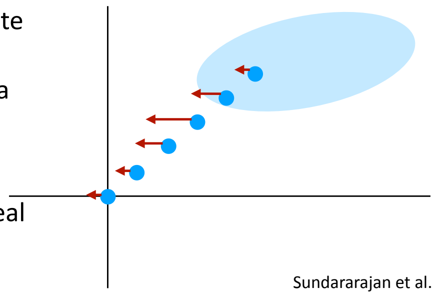
Problems with LIME



Simonyan et al. (2013)

Problems with LIME

- Suppose you have prediction = A OR B for features A and B. Changing either feature doesn't change the prediction, but changing both would. Gradient-based method says neither is important
- Integrated gradients: compute gradients along a path from the origin to the current data point, aggregate these to learn feature importance
- Intermediate points can reveal new info about features



Sundararajan et al. (2017)

Evaluating Explanations

Faithfulness vs. Plausibility

- Suppose our model is a bag-of-words model with the following:

  the = -1, movie = -1, good = +3, bad =0

  the movie was good   prediction score=+1

  the movie was bad   prediction score=-2

- Suppose explanation returned by LIME is:

  the movie was good

  the movie was bad

- Is this a "correct" explanation?

Faithfulness vs. Plausibility

- *Plausible* explanation: matches what a human would do

  the movie was good   the movie was bad

  - Maybe useful to explain a task to a human, but it's not what the model is really doing!

- *Faithful* explanation: actually reflects the behavior of the model

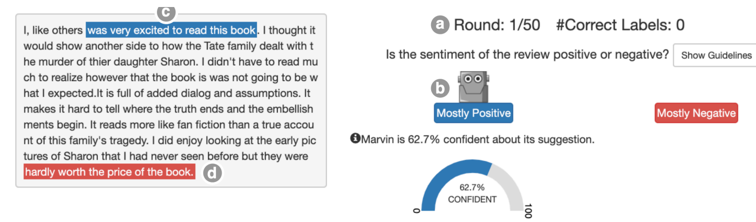
  the movie was good   the movie was bad

  - We usually prefer faithful explanations; non-faithful explanations are actually deceiving us about what our models are doing!
  - Rudin: *Stop Explaining Black Box Models for High-Stakes Decisions and Use Interpretable Models Instead*

Evaluating Explanations

- Nguyen (2018): delete words from the input and see how quickly the model flips its prediction?
  - Downside: not a “real” use case
- Hase and Bansal (2020): counterfactual simulatability: user should be able to predict what the model would do in another situation
  - Hard to evaluate

Evaluating Explanations



- Human is trying to label the sentiment. The AI provides its prediction to try to help. Does the human-AI team beat human/AI on their own?
- AI provides both an explanation for its prediction (blue) and also a possible counterargument (red)
- Do these explanations help the human? Slightly, but **AI is still better**
- No positive results on “human-AI teaming” with explanations

Bansal et al. (2020)

What to Expect from Explanations?

Ye et al. (2021)

- What do we really want from explanations?
  - Explanations should describe model behavior with respect to counterfactuals (Miller, 2019; Jacovi and Goldberg, 2021)

The movie is not that bad.

The movie is not ___ ___.

- What about realistic counterfactuals? Since dropping tokens isn’t always meaningful

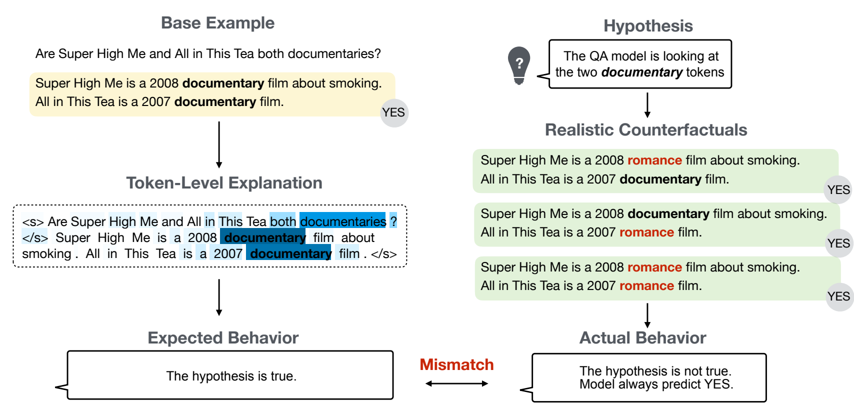
The movie is not actually bad.

- We are going to evaluate explanations based on whether they can tell us useful things about model behavior

A Multi-hop QA Example

Ye et al. (2021)

- We formulate a hypothesis about the model’s behavior, and test it using counterfactuals



Ongoing Conversation

- Lots of ongoing research:
  - How do we interpret explanations?
  - How do *users* interpret our explanations?
  - How should *automated systems* make use of explanations?
- Still a growing area

Packages

- AllenNLP Interpret: https://allennlp.org/interpret
- Captum (Facebook): https://captum.ai/
- LIT (Google): https://ai.googleblog.com/2020/11/the-language-interpretability-tool-lit.html
- Various pros and cons to the different frameworks

Takeaways

- Many other ways to do explanation:
  - Probing tasks: do vectors capture information about part-of-speech tags?
  - Diagnostic test sets ("unit tests" for models)
  - Building models that are explicitly interpretable (decision trees)

Wallace, Gardner, Singh
Interpretability Tutorial at EMNLP 2020