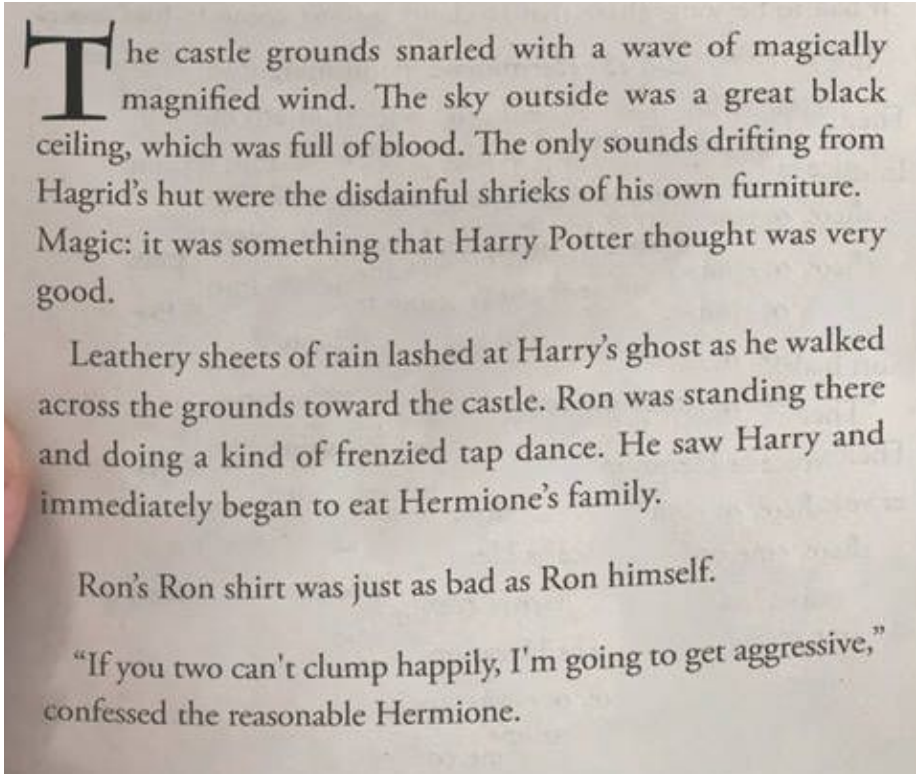
Please enjoy.



The castle grounds snarled with a wave of magically magnified wind. The sky outside was a great black ceiling, which was full of blood. The only sounds drifting from Hagrid's hut were the disdainful shrieks of his own furniture. Magic: it was something that Harry Potter thought was very good.

Leathery sheets of rain lashed at Harry's ghost as he walked across the grounds toward the castle. Ron was standing there and doing a kind of frenzied tap dance. He saw Harry and immediately began to eat Hermione's family.

Ron's Ron shirt was just as bad as Ron himself.

"If you two can't clump happily, I'm going to get aggressive," confessed the reasonable Hermione.

CIS 522: Generative RNNs

Feb 27, Guest lecture by the one and only Jeff

Penn Engineering

This is a lecture at the intersection of many topics.

By the end, I hope you may have some insight on:

- How to ~~abuse~~ design RNNs to solve specific problem domains
- What voodoo magic to invoke in order to convince skeptical people that your models are awesome
- How to leverage really good data science into USD $MM

I have a pet peeve.

I have a pet peeve.
"CNNs are for CV; RNNs are for NLP."

Claim: architectures should be chosen based on how their inductive bias / prior fits your problem domain.

Recall from module 11: a more systematic, rigorous way to consider how to model sequence data (including text and language problems).

There are 4 canonical ways to deal with variable length.

1. Truncation: chop off the tail.
2. Bagging: turn the vector into a count-dict.
3. Convolution: convolute over a fixed-length filter and aggregate the results.
4. Recurrence: make your forward pass variable-length.

Each of the 4 methods has its own prior.

1. Truncation: chop off the tail.
    a. Use when you have a ranking of the most important features (similar to PCA).
    b. Ex: for each person in a group, we have a variable-length ranked list of their favorite books.
2. Bagging: turn the vector into a count-dict.
    a. Use when your prior is that syntax (i.e. ordering) does not matter.
    b. Ex: for each baseball player in a game, we have a sequence of strikes / hits.
3. Convolution: convolute over a fixed-length filter and aggregate the results.
    a. Use when your prior is that only short-term syntax matters.
    b. Ex: *n*-gram models
4. Recurrence: make your forward pass variable-length.
    a. Use when your prior is that both short-term and long-term ordering matters.
    b. Ex: most general-purpose NLP architectures

This begs a very reasonable follow-up question.

What exactly is the point of a data scientist?

What is the marginal benefit of hiring a data scientist vs. having your ETL data engineer guy just use AutoML on everything?

The original purpose of CIS 700-004.

- Learn to identify which problems are well-solved by DL solutions.
- **Learn how inductive biases improve performance and how to induce good ones.**
- **Learn how to create and present high-quality findings from DL models.**

The original purpose of CIS 700-004.

- Learn to identify which problems are well-solved by DL solutions.

- **Learn how inductive biases improve performance and how to induce good ones.**
  - **Typically reduce the number of trainable parameters.**
  - **Makes your pipeline less data-hungry.**

- **Learn how to create and present high-quality findings from DL models.**

The original purpose of CIS 700-004.

- Learn to identify which problems are well-solved by DL solutions.

- **Learn how inductive biases improve performance and how to induce good ones.**
  - **Typically reduce the number of trainable parameters.**
  - **Makes your pipeline less data-hungry.**

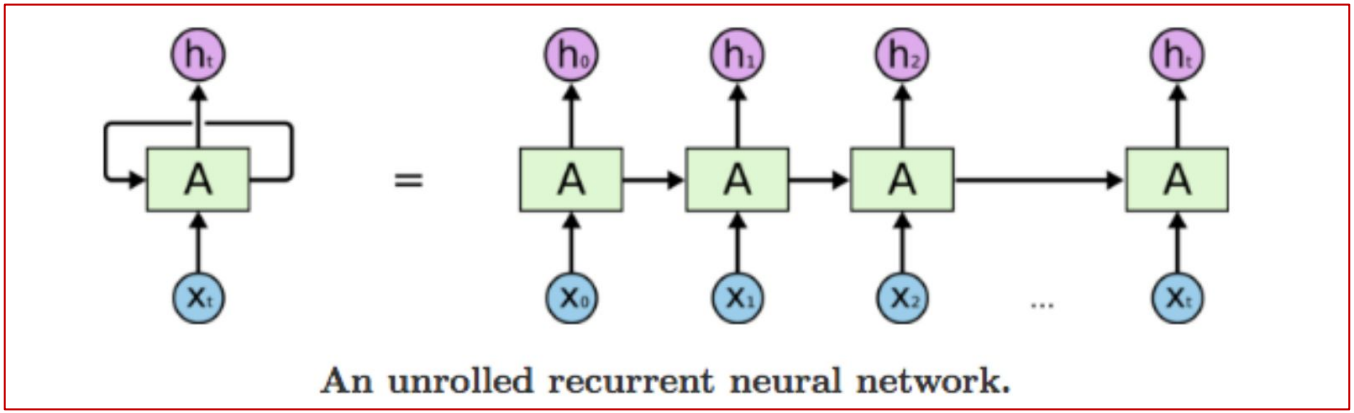
- **Learn how to create and present high-quality findings from DL models.**
  - Understand how to phrase the impact of your models to technical and non-technical audiences.
  - **Learn to interpret your models and model outputs.**

Roadmap for today's lecture

- Define a generative RNN.
- Identify what problem domains its "shape" most conveniently solves.
- Interpret the outputs of generative models.
  - We'll visit Andrej Karpathy's glorious, glorious Ode to the RNN.
- Analyze 2 case studies of applying inductive biases to generative RNNs.
  - LyreBird's SampleRNN
  - The BALM framework

What is a generative RNN?

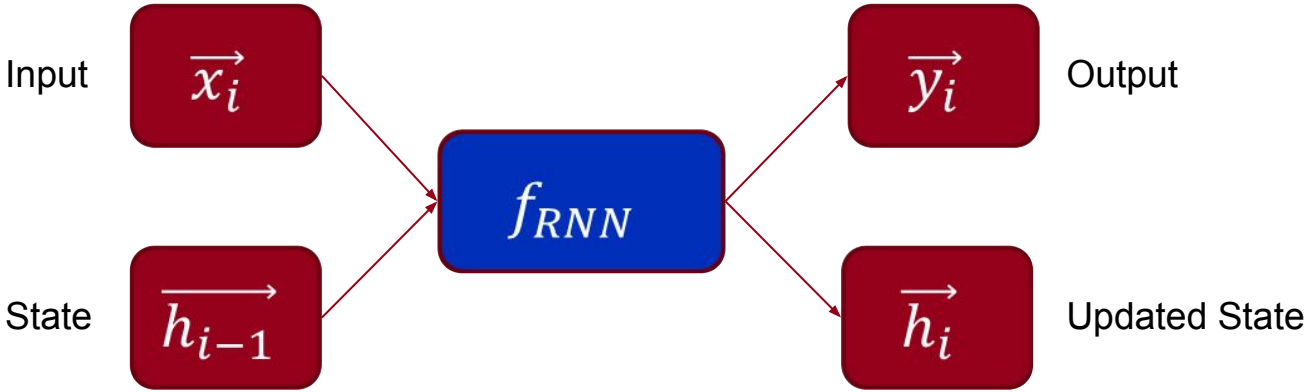
This picture does not tell much of a story.



An unrolled recurrent neural network.

What is A even passing from timestep to timestep?!

Recall the API of an RNN.



RNNs come in 3 shapes.

- "Encoders": variable-length input to fixed-length output
  - (n → k)

- "Taggers": variable-length input and output
  - (n → n)

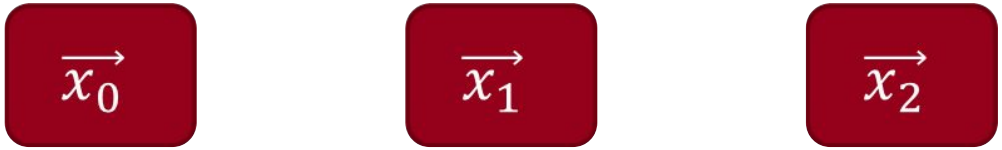
- "Decoders / Generative RNNs": fixed-length input, variable-length output
  - (k → n)

Note that the degenerate case (fixed-length input and output) is just a feedforward network.

Encoders

Encoders: variable-length input

Input



(Here's the shape.)

Label

Encoders: variable-length input

Input

Jeffrey

rocks

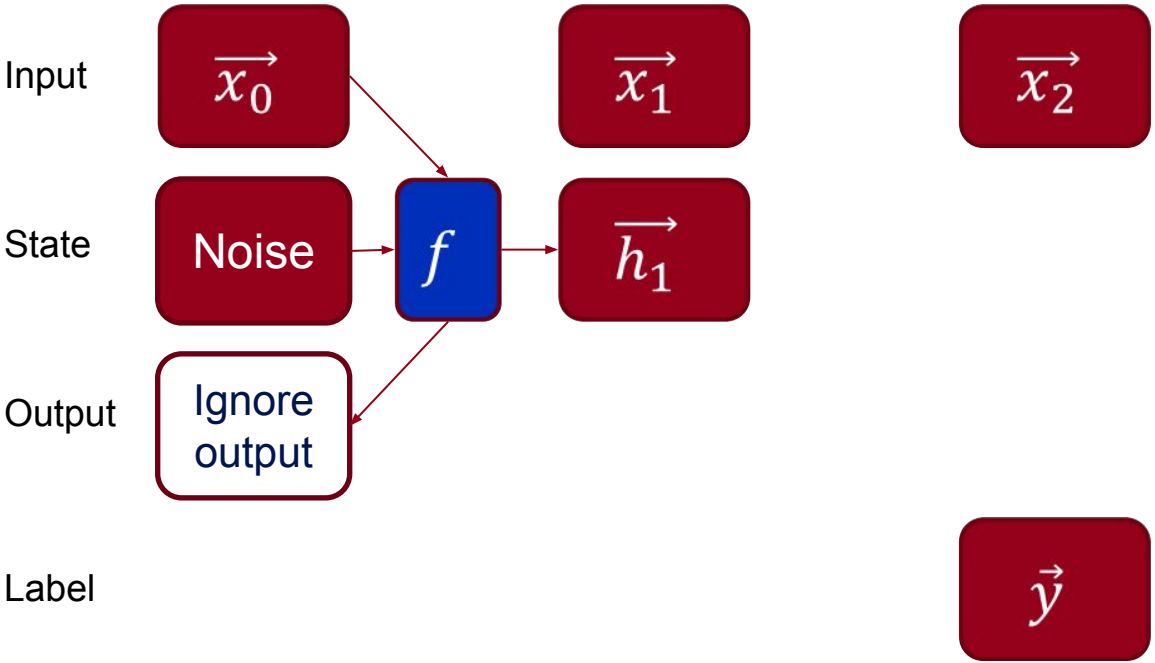
<end>

(Here's a sentiment analysis example.)

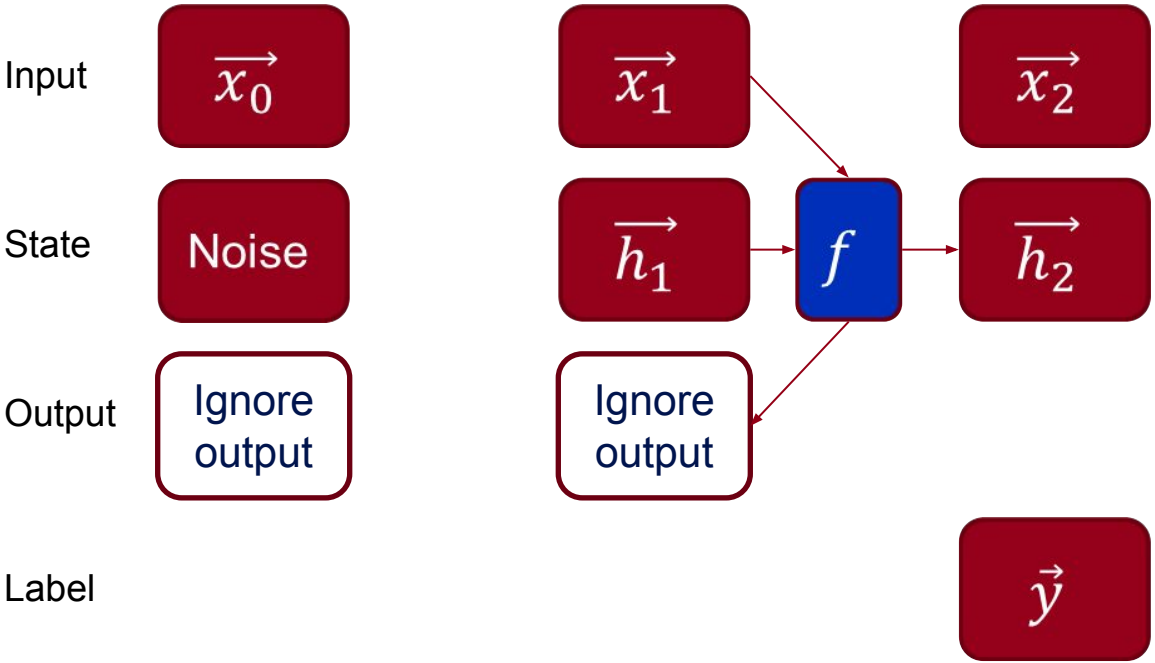
Label

0.95

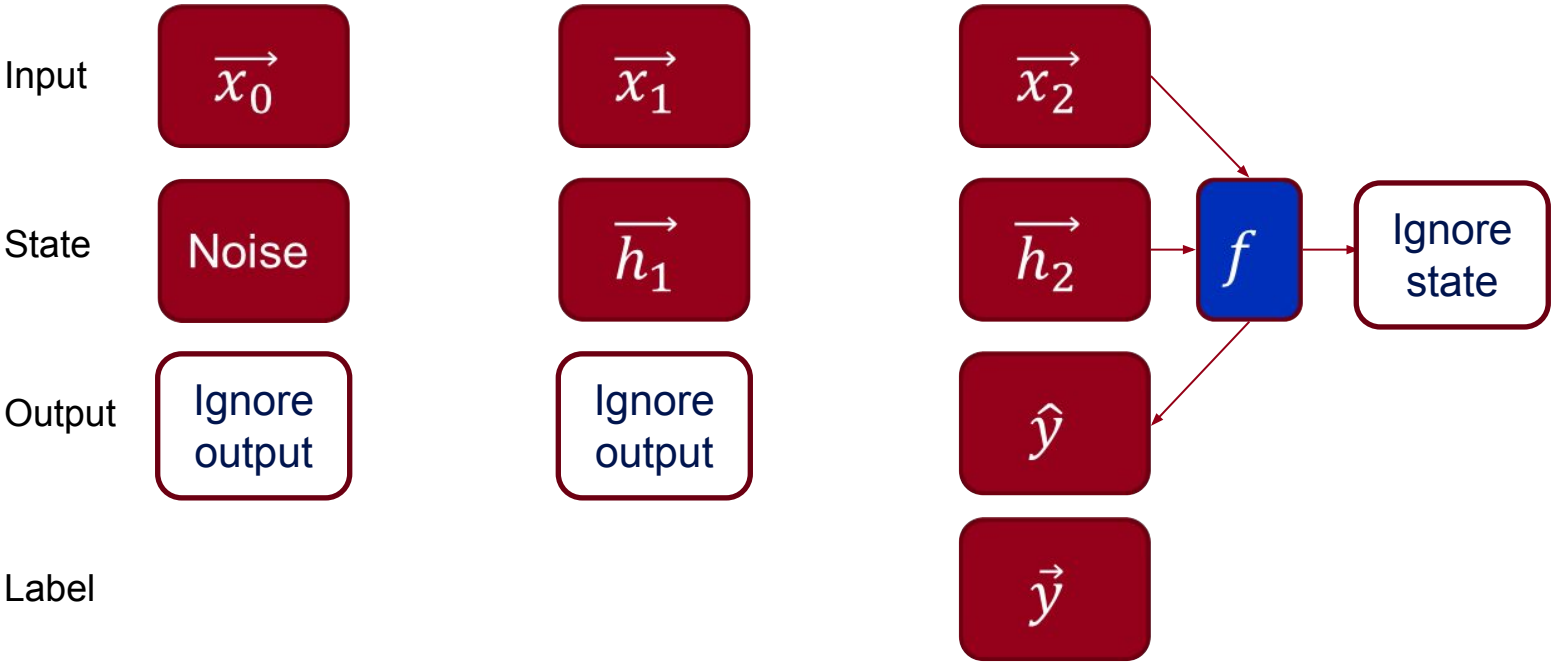
Encoders: variable-length input



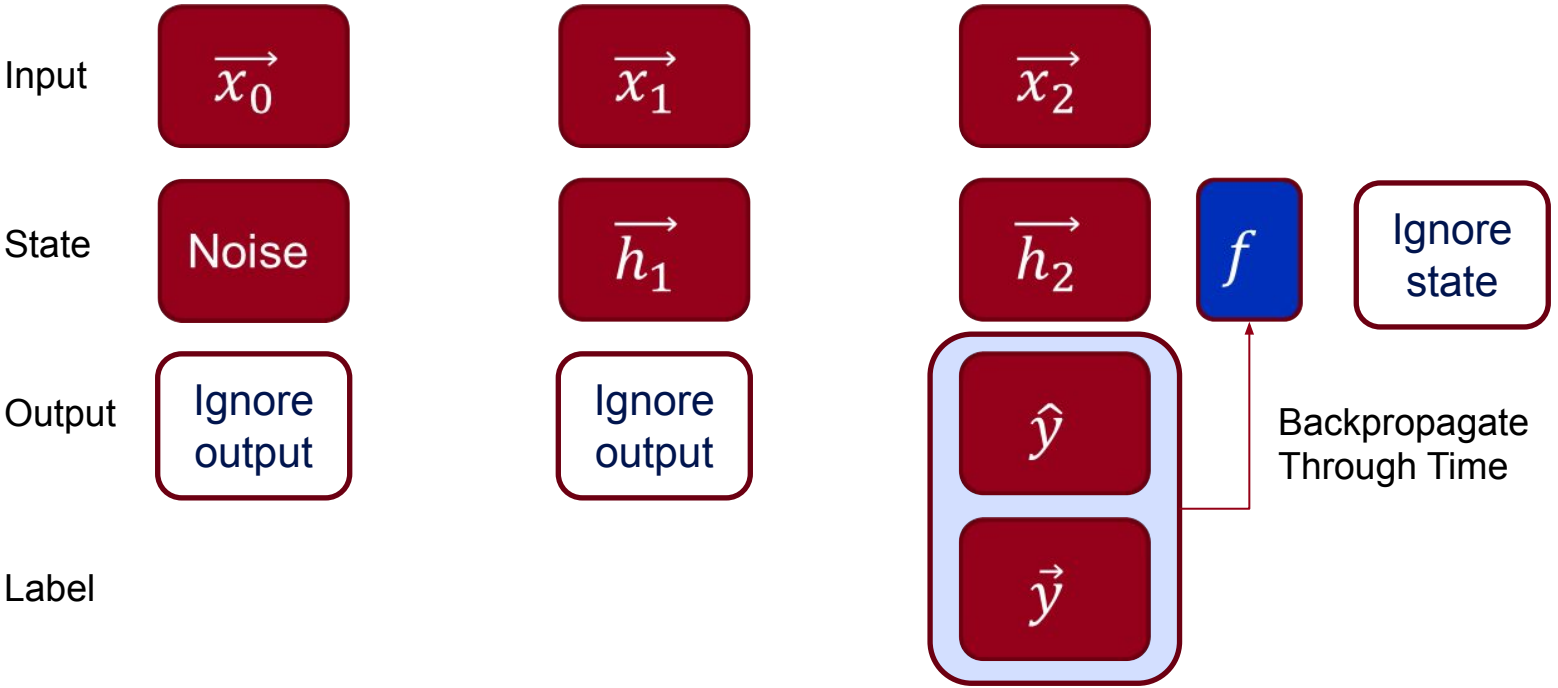
Encoders: variable-length input



Encoders: variable-length input



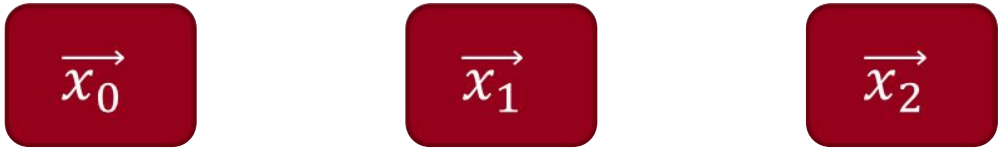
Encoders: variable-length input



Taggers

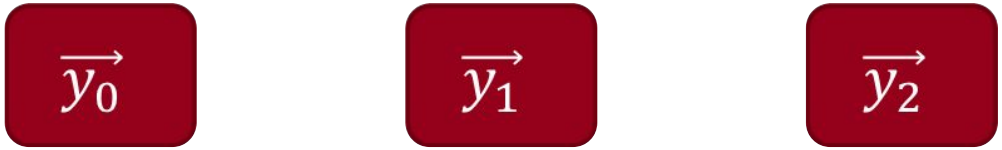
Taggers: variable-length input and output

Input



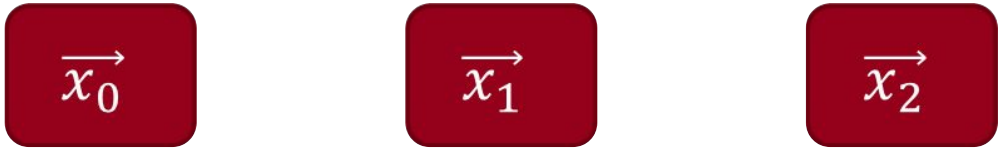
(Here's the shape.)

Label



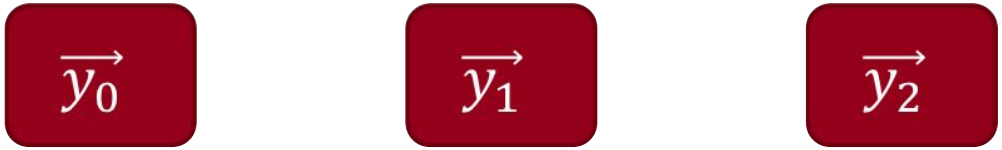
Taggers: variable-length input and output

Input

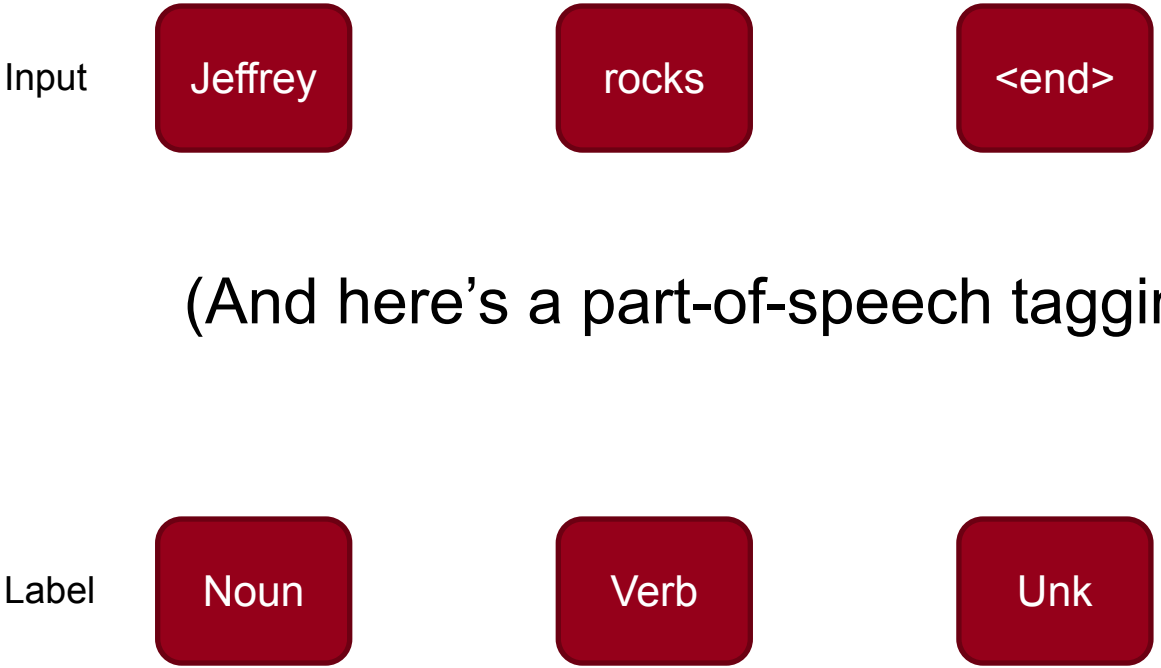


Note that the input and output usually have the same length.

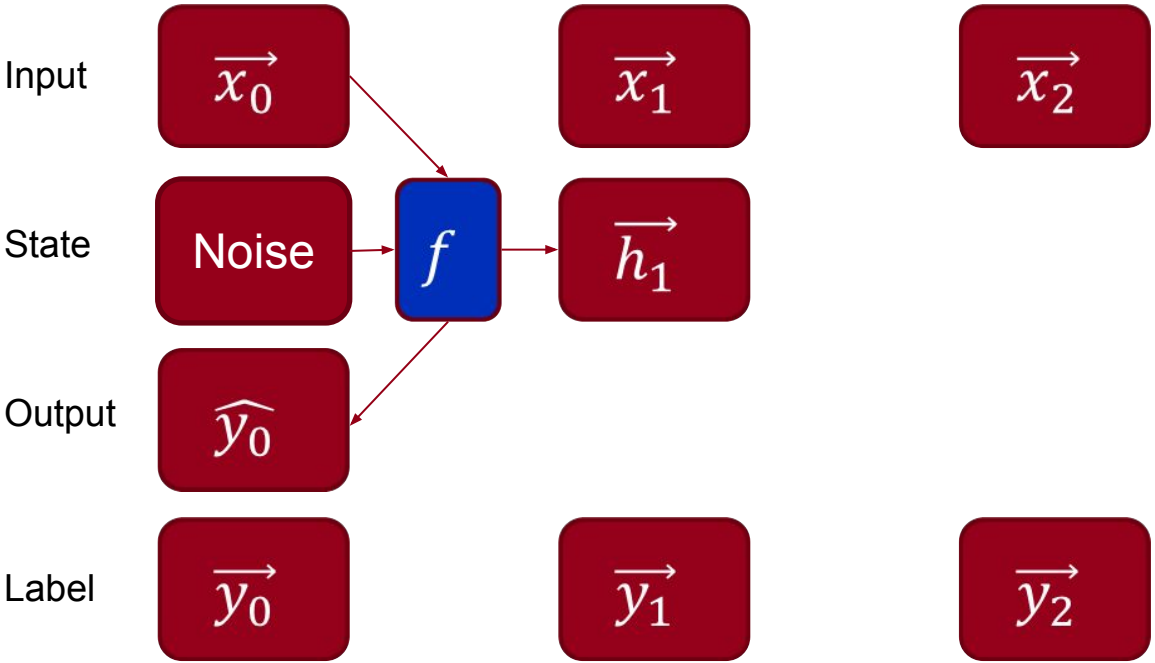
Label



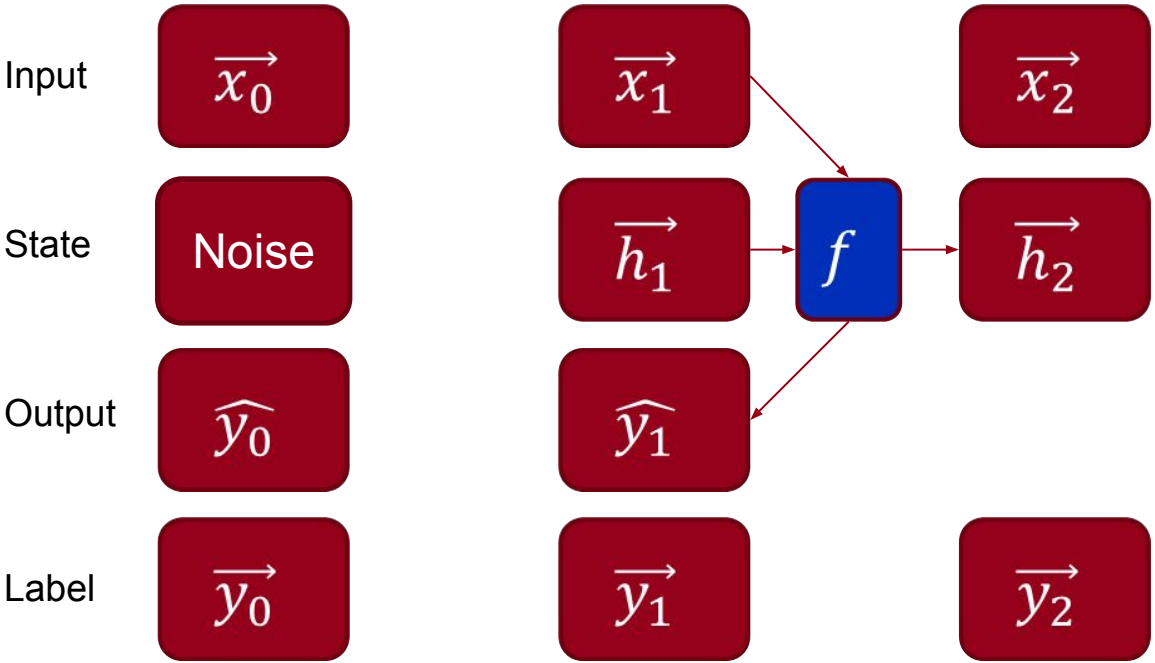
Taggers: variable-length input and output



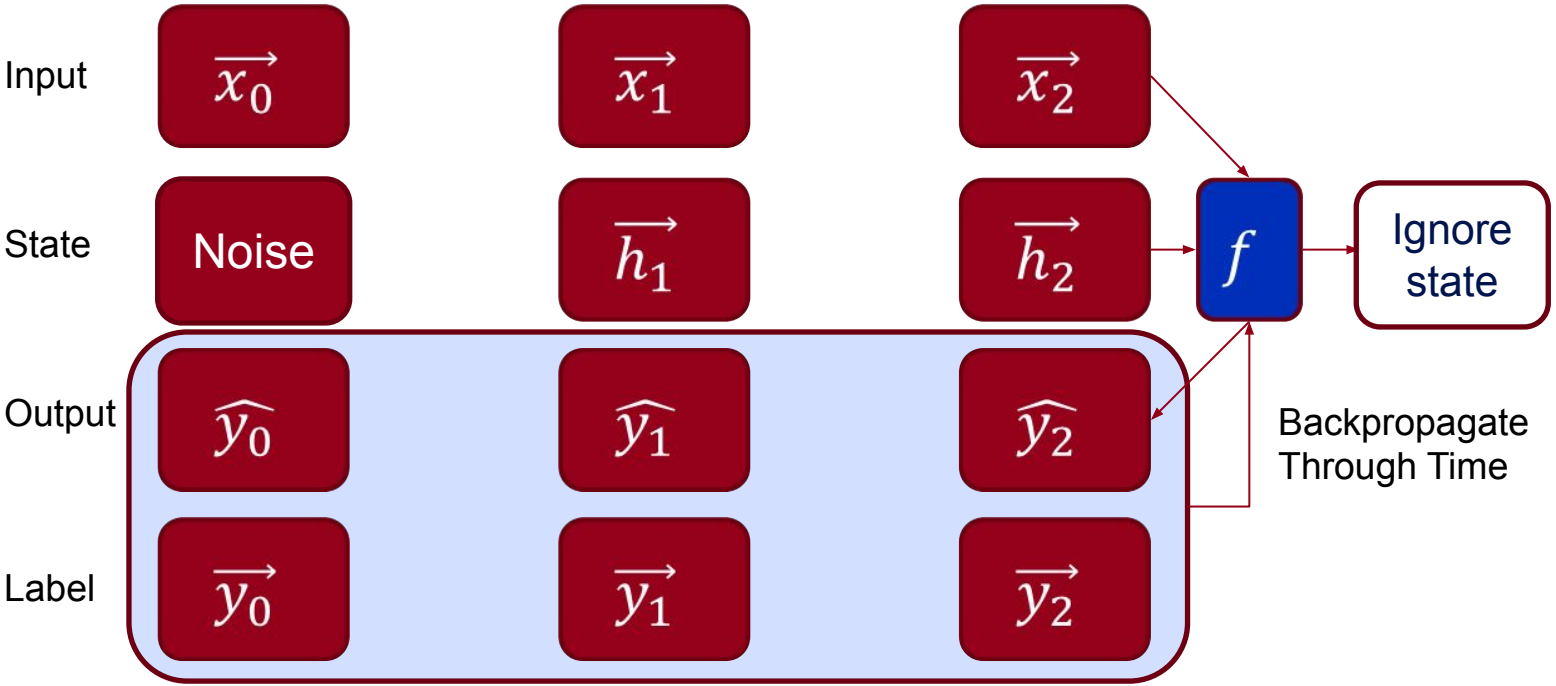
Taggers: variable-length input and output



Taggers: variable-length input and output



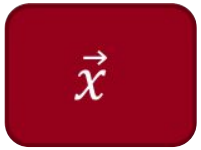
Taggers: variable-length input and output



Decoders
AKA Generative RNNs

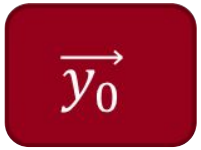
Decoders: variable-length output

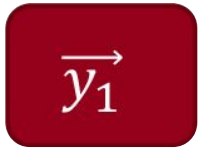
Input



(Here's the shape.)

Label





Decoders: variable-length output

Input

Mystery vector

(Mysterious example: stay tuned)

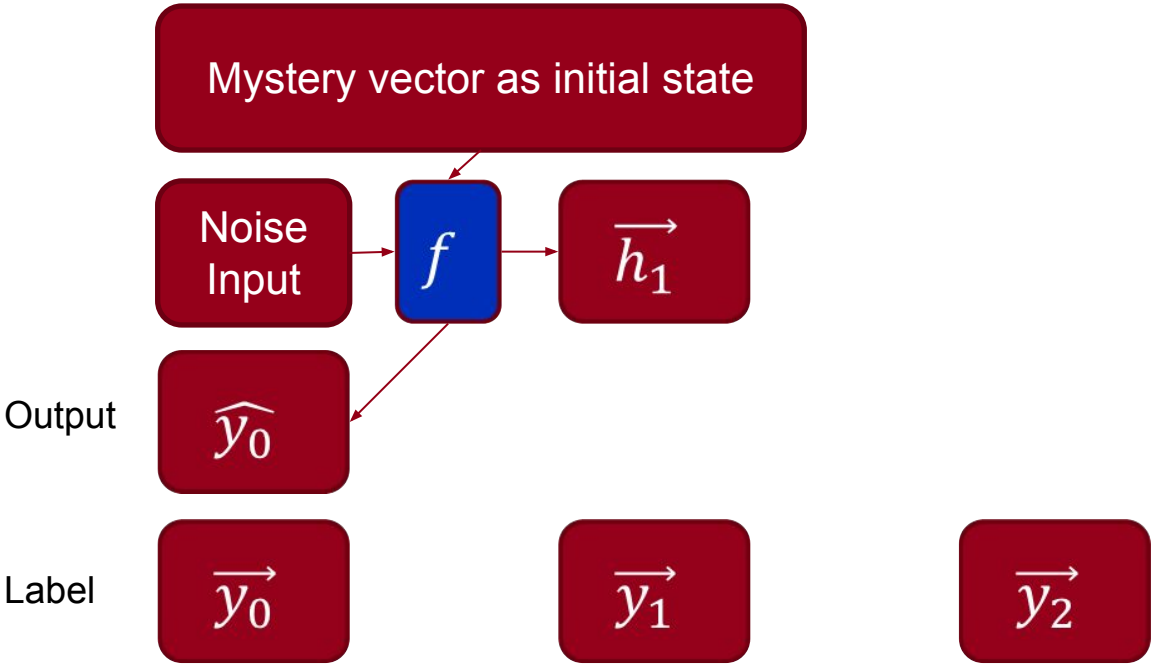
Label

Jeffrey

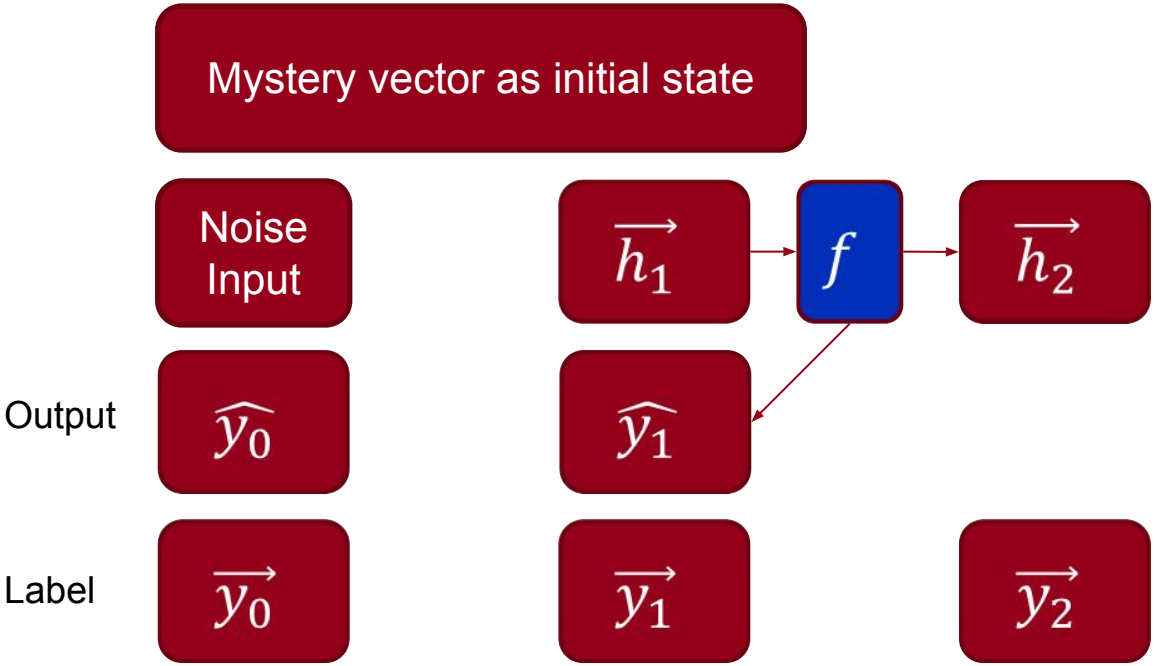
rocks

<end>

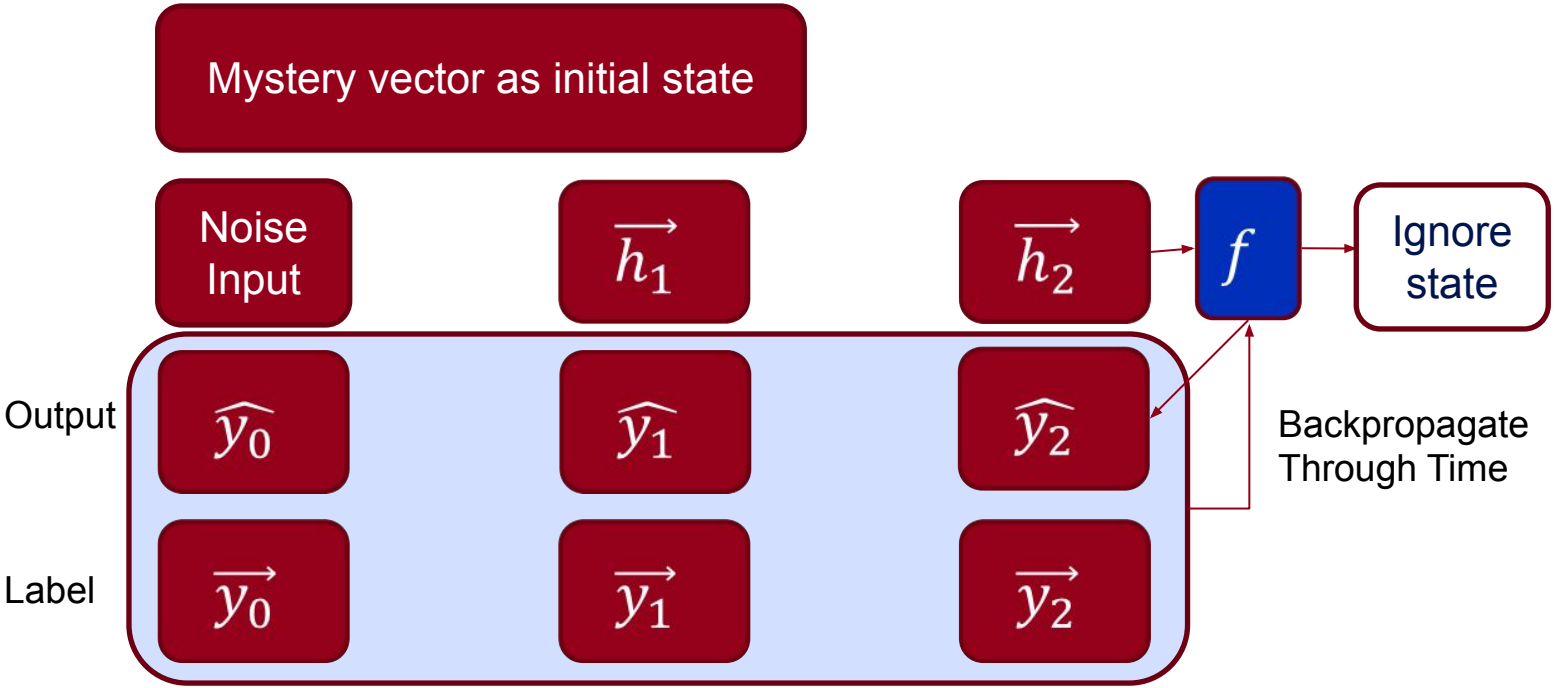
Decoders: variable-length output



Decoders: variable-length output



Decoders: variable-length output



So RNNs come in 3 shapes. Who cares?

- **Feedforward networks**: fixed-length I/O
- **Encoders**: variable-length input, fixed-length output
- **Taggers**: variable-length input and output (if they have the same length)
- **Generative RNNs**: fixed-length input, variable-length output

These are the 4 lego blocks that we have to work with. I claim that you can solve any problem domain with just these four shapes.

Intuition check

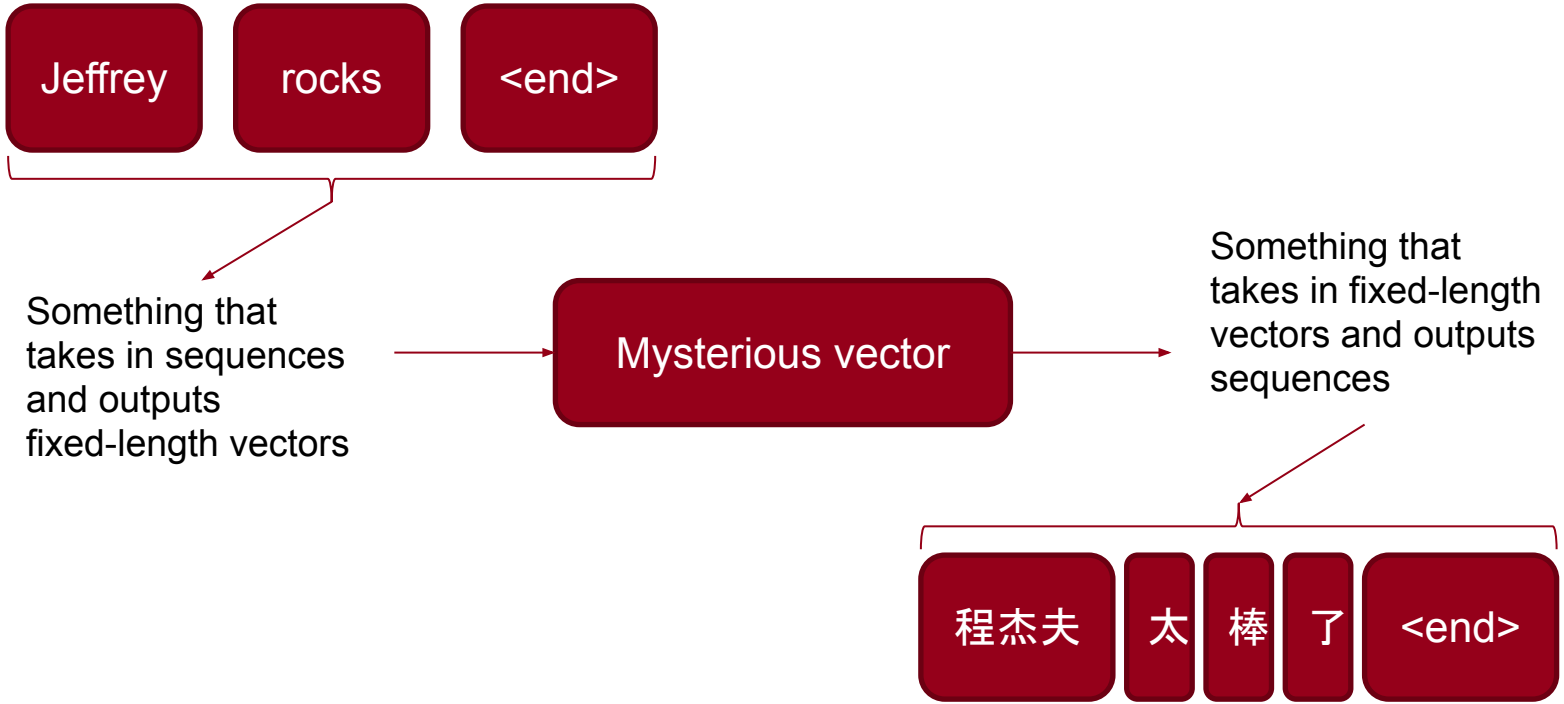
Hold up, we said that we can solve any problem shape with just these four building blocks.

Popular problems like machine translation have sequence inputs and outputs that **need not have the same length**.

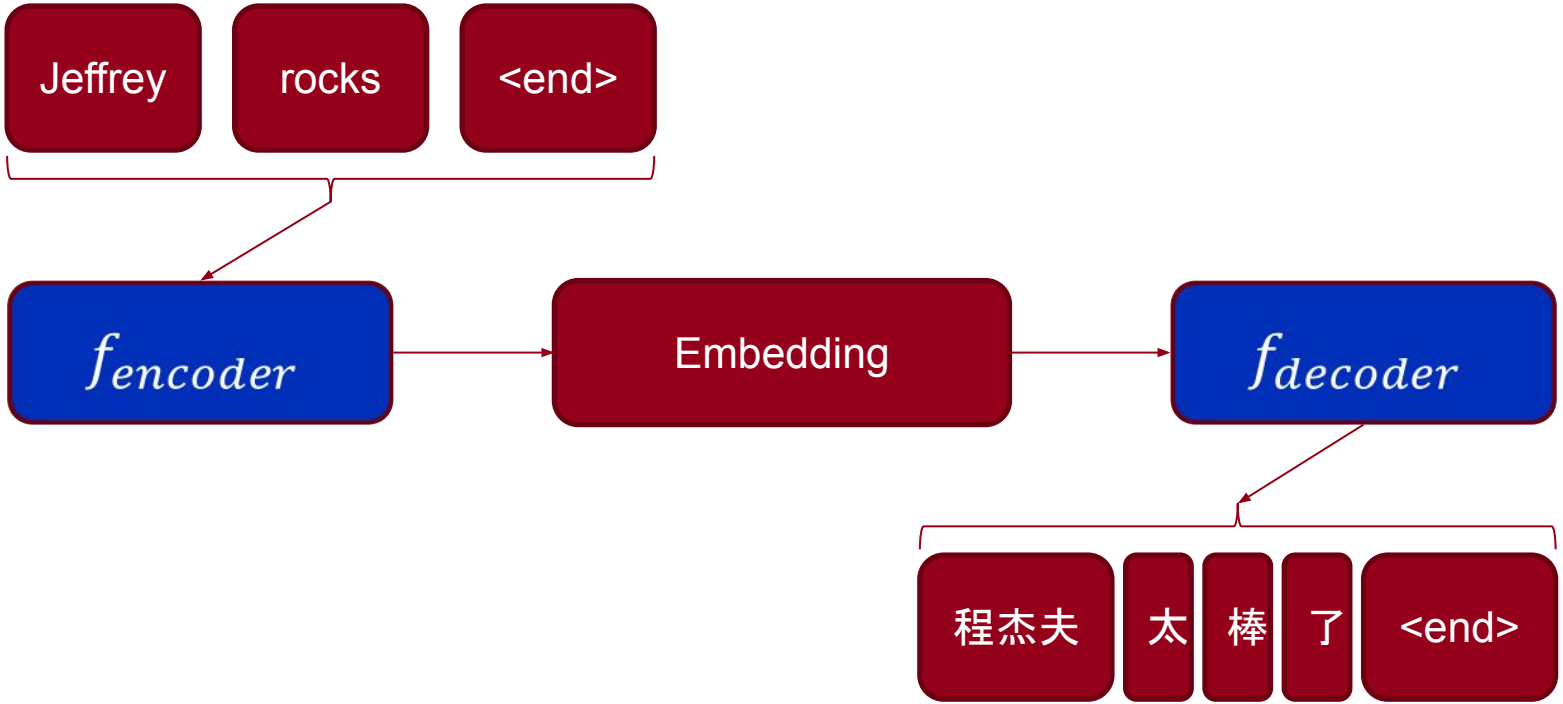
What gives?

The seq2seq framework

The seq2seq framework



The seq2seq framework



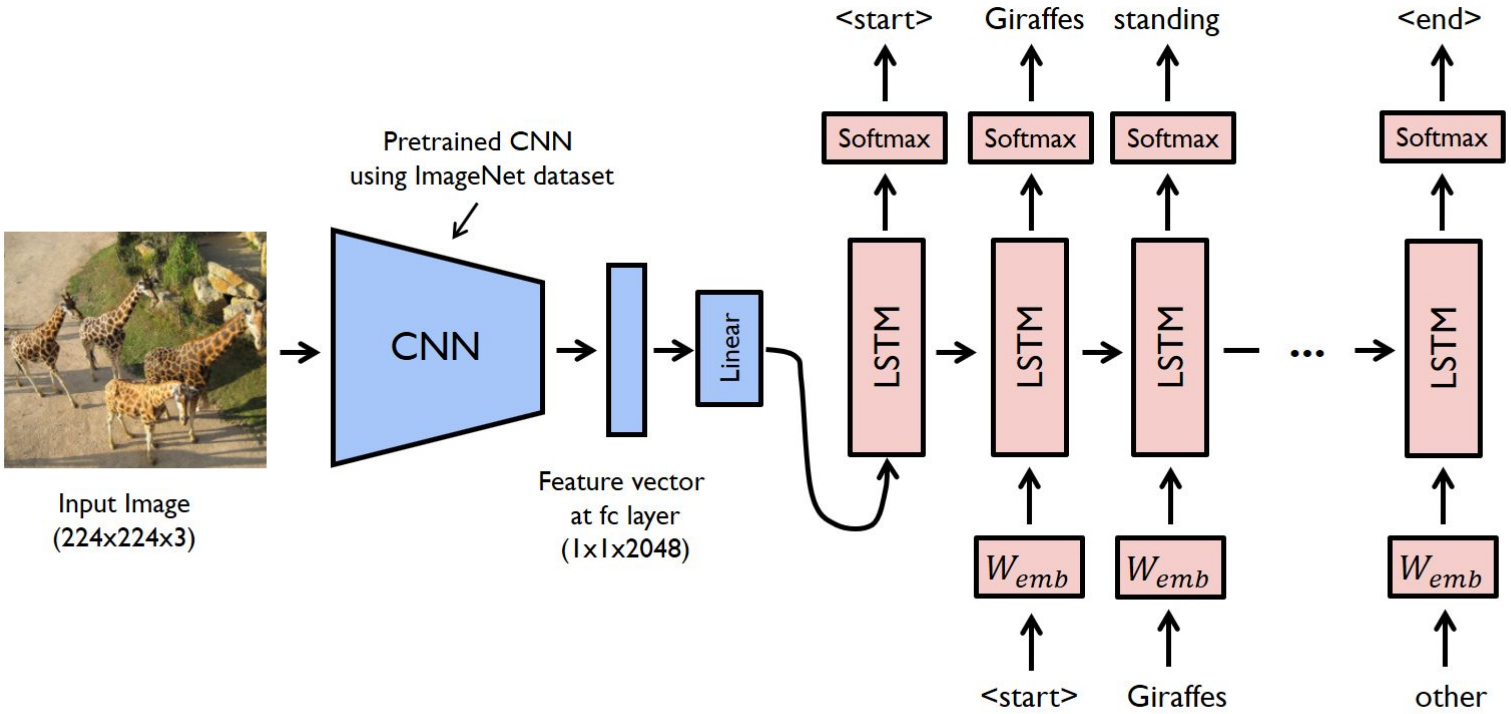
Some informal definitions

- Embedding: a vector in $\mathbb{R}^k$ (constant k) that describes an example from the problem domain.
  - E.g. the unsupervised GloVe algorithm gives us **word embeddings**.
  - Geoffrey Hinton likes to call embeddings in NLP as **thought vectors**.
- Latent space: a vector space $\mathbb{R}^k$ where the embeddings live.
  - Typically has the following features: k is small (low-dimensional), vector similarity implies semantic similarity.
- Autoencoder: a pipeline consisting of an encoder/decoder pair that takes in examples (x, x) and learns a latent space for these examples.
- Seq2seq: a pipeline consisting of an encoder/pair that takes in sequence pairs $(x, y) \in X \times Y$ and learns a latent space in-between the domains X, Y.

PixelRNN

Figure 1. Image completions sampled from a PixelRNN.

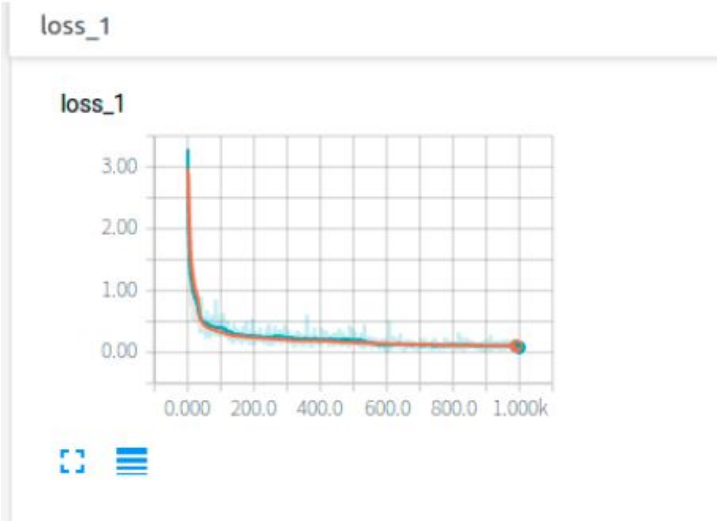
Image captioning



Why should we care about interpretability?

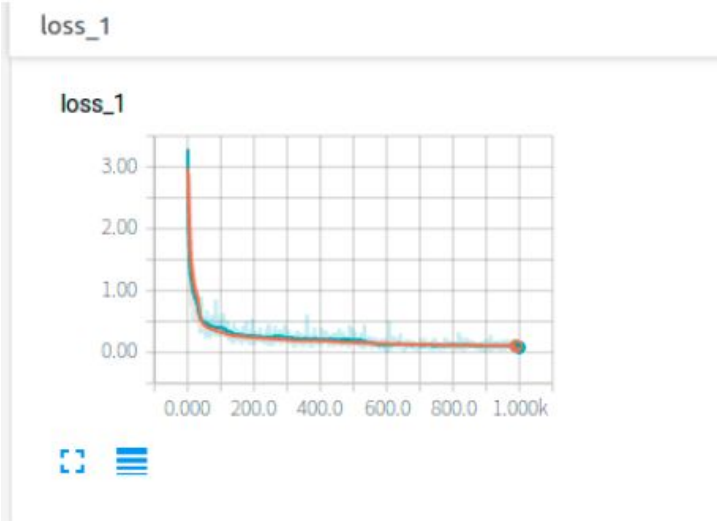
Example 1

- Suppose a model converges to a loss within floating-point error; is it a good model?



Example 1

- Suppose a model converges to a loss within floating-point error; is it a good model?
  - I accidentally repeated the same batch of my training data every pass.
  - Test accuracy was 0%.



Example 2

- Suppose a model that only converges to a loss of 10^{10}; is it a good model?

Example 2

- Suppose a model that only converges to a loss of 10^{10}; is it a good model?
  - I was predicting a time series for a manufacturing company's monthly demand. The demand itself is measured in billions of USD, and the loss function is MSE.
  - This is literally magical levels of performance.

Example 3

- Suppose a NLU (natural language understanding) model achieves 87% accuracy, and I've unit-tested every part of the pipeline, having complete faith that I've done everything correctly. Is it a good model?

Example 3

- Suppose a NLU (natural language understanding) model achieves 87% accuracy, and I've unit-tested every part of the pipeline, having complete faith that I've done everything correctly. Is it a good model?
- I later ran an out-of-the-box CRF that achieved 89%.

Example 4

- Suppose I gave a generative model the following completely original and fictitious prompt:
  - *In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
- And it wrote the following:

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.
Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.
Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.
While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."
Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America…

Why interpretability matters

- The process verifies the basic correctness of implementation.
- Humans responsible for model can better deal with non-stationarity of data.
- Interpretability builds trust downstream (to *non-technical* users of the models).
- Interpretability informs future design / decisions around ML (e.g. data collection)

BLEU (2002)

- "bilingual evaluation understudy"
- "the closer a machine translation is to a professional human translation, the better it is"

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

The Ode to the RNN

Andrej Karpathy

Andrej Karpathy is a researcher at Tesla. [1, 2] He specializes in artificial intelligence and deep learning. [3, 4]

Andrej Karpathy was a computer science PhD student at Stanford University in California, focusing on natural language processing and recurrent neural networks, which can model languages. [5] He has mostly worked in academia, but he joined Tesla's artificial intelligence group OpenAI last September as a research scientist. [6] Most of Karpathy's research focuses on image recognition and understanding. [7] His Reddit username, badmephisto, is also the username for his YouTube channel dedicated to solving Rubik's cubes. [7]

Andrej Karpathy

From Wikipedia, the free encyclopedia

Andrej Karpathy is the director of artificial intelligence and Autopilot Vision at Tesla.[1][2][3] He specializes in deep learning[4] and image recognition and understanding.[5]

Andrej Karpathy was born in Slovakia and moved with his family to Toronto when he was 15.[6] He completed his Computer Science / Physics undergraduate degree at University of Toronto in 2009[7] and completed his Master's degree at University of British Columbia in 2011,[7] where he worked on physically-simulated figures. He graduated with PhD from Stanford University in 2015 under the supervision of Dr. Fei-Fei Li, focusing on intersection of natural language processing and computer vision, and deep learning models suited for this task.[8] He joined the artificial intelligence group OpenAI as a research scientist in September 2016[9] and became Tesla's director of artificial intelligence in June 2017.[5]

He has a YouTube channel dedicated to speedcubing.[5][10]

Temperature

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$\hat{y}'_i = \frac{e^{z_i/t}}{\sum_j e^{z_j/t}}$$

Review: language models

- Language models assign probabilities to sequences of words.

Review: language models

- Language models assign probabilities to sequences of words.

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\dots P(w_n|w_1^{n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

Previously: the n-gram assumption

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n-1})$$

The Unreasonable Effectiveness of RNNs

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Karpathy's RNNs generate remarkable outputs.

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariciely in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S,} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,*} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,\acute{e}tale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Karpathy analyzes the *neuron-level* behavior.

Cell that turns on inside quotes:

```
"You mean to imply that I have nothing to eat out of.... On the
contrary, I can supply you with everything even if you want to give
dinner parties," warmly replied Chichagov, who tried by every word he
spoke to prove his own rectitude and therefore imagined Kutuzov to be
animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating
smile: "I meant merely to say what I said."
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
        siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

Case Studies

The SampleRNN

- https://arxiv.org/pdf/1612.07837.pdf

Why is the SampleRNN good data science?

- Identified the correct shape.
  - used a generative RNN in order to produce sequences.
- Used the correct inductive bias.
  - Understanding the waveform structure allowed the team to overcome the vanishing gradient problem.
- Interpreted the models.
  - Looked at comparative losses to benchmarks and SOTA
  - Visually compared the short-term and long-term waveform outputs to true audio.
  - Actually listened to the output recordings and had real people rate them.
- https://youtu.be/VnFC-s2nOtI?t=167

Bilingual is at Least Monolingual

- https://arxiv.org/abs/1909.01146

"“Drop your RNN and LSTM, they are no good!”

"Use attention. Attention really is all you need!" -- Eugenio Culurciello