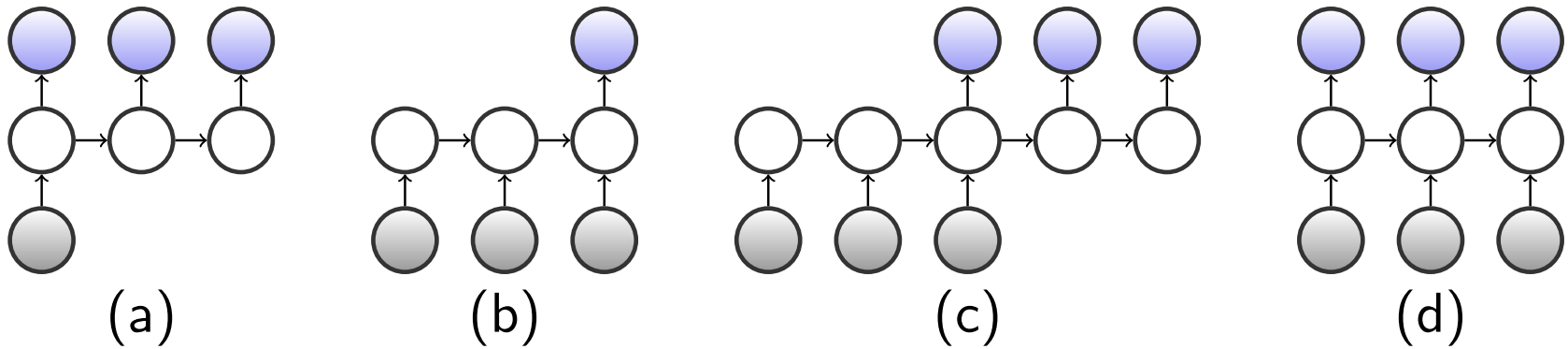
Recurrent Neural Networks (cont.)

Nan Ye

School of Mathematics and Physics
The University of Queensland

Recall: Sequence Modelling



RNNs are good for various sequence modelling problems, including

- **(a)** One to many, e.g. image captioning
- **(b)** Many to one, e.g. video classification
- **(c)** Many to many, e.g. machine translation
- **(d)** Many to many, e.g. video frame classification

STAT3007 is suitable for ...

How will you complete STAT3007 course description?

> **This course provides a comprehensive and rigorous coverage of deep learning from both applied and theoretical perspectives. This course is suitable for**

STAT3007 is suitable for ...

A sample completion by AI

This course provides a comprehensive and rigorous coverage of deep learning from both applied and theoretical perspectives. This course is suitable for a broad spectrum of researchers and engineers, including researchers, practitioners, and developers with a strong theoretical background.

Faculty:
Vladas Griskevicius
Vlatko Vedral
Simon Heng
Hadi Hariri
Yakoub Toubani

We will cover the fundamentals of deep learning from a theoretical and practical perspective in a single, integrated course.

Using Python and Scipy, we will implement two different deep learning models: One for classifying images and another for language processing.

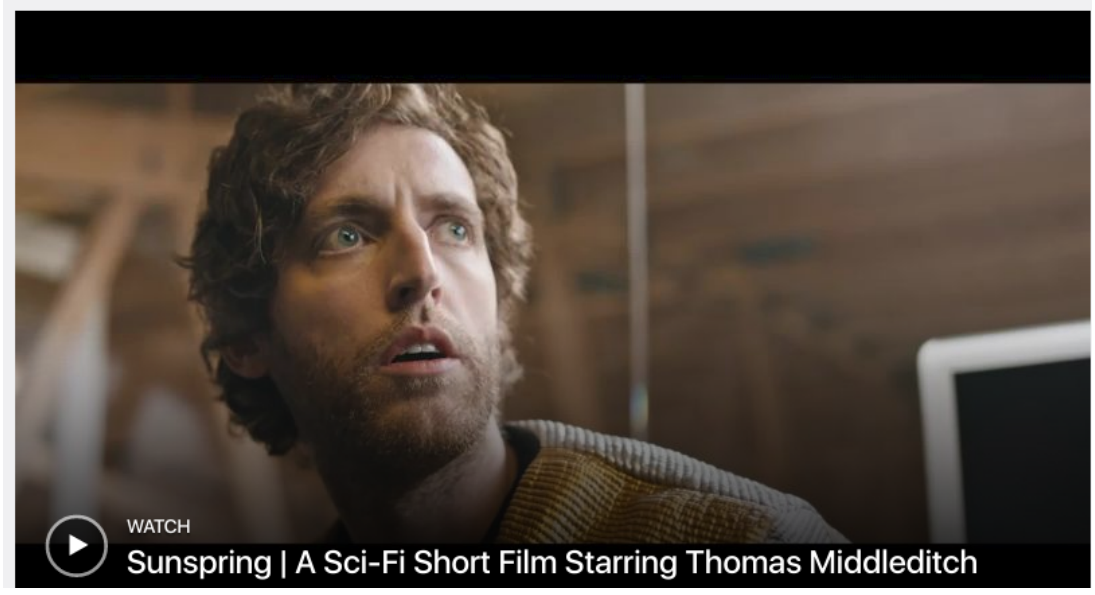
https://app.inferkit.com/demo

STAT3007 is suitable for ...

Description on the course profile

This course provides a comprehensive and rigorous coverage of deep learning from both applied and theoretical perspectives. This course is suitable for both students who want to build data-driven enabling applications with deep learning, and students who want to develop a solid foundation for doing research in deep learning in particular, and machine learning or artificial intelligence more broadly.

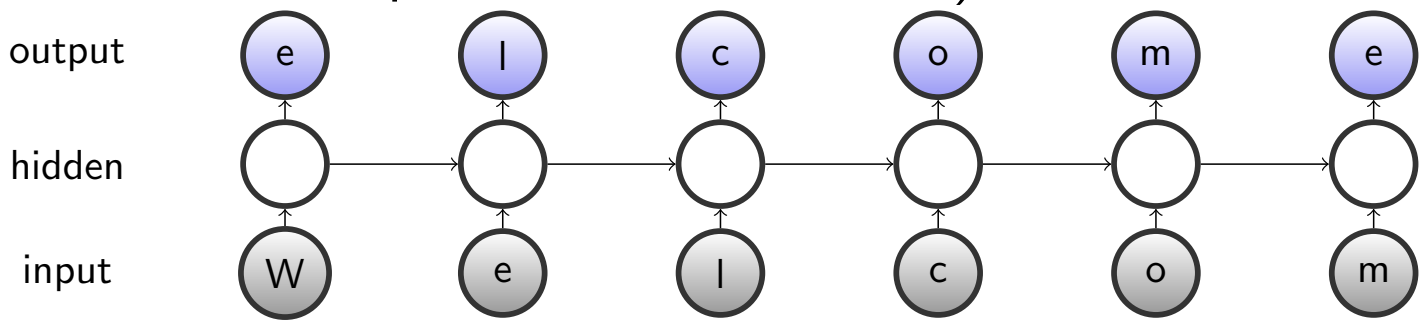
Sci-Fi Film Written by LSTM



Sunspring

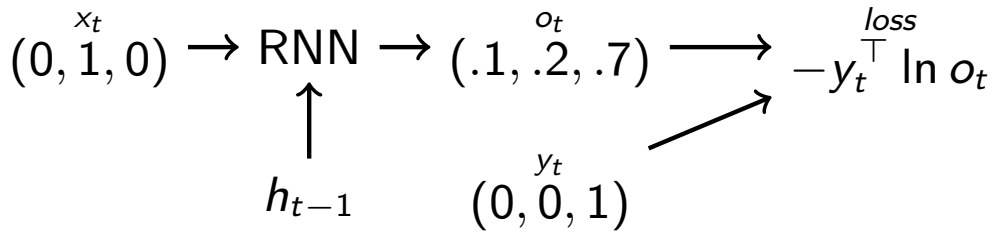
Character-Level Language Models

- We can train an RNN to generate natural language text.
- We only need to have natural language text as the training data.
- The model is trained to predict the next character (i.e., the target at each time step is the next character)



- This is at the boundary of supervised learning and unsupervised learning
  - No separate teaching signal required.
  - Supervised learning technique is used.

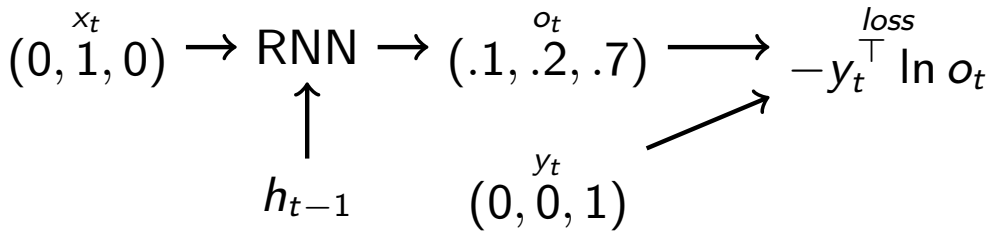
Details



Input and output representation

- We need numerical representations for the input and output.
- Assume that there are v characters.
- Input: $x_t \in \mathbf{R}^v$ is the one-hot encoding for the t-th input character
  - Specifically, the one-hot encoding of the i-th character is a vector of 0's except the i-th entry is 1 (or i-th standard unit vector in $\mathbf{R}^v$).
  - e.g. a $= (1, 0, 0, \ldots)$, b $= (0, 1, 0, \ldots)$, $\ldots$
- RNN output: $o_t \in \mathbf{R}^v$ is a v-dimensional probability vector representing the distribution over the next character.
- True output: $y_t \in \mathbf{R}^v$ is the one-hot encoding of the true output character.

Details



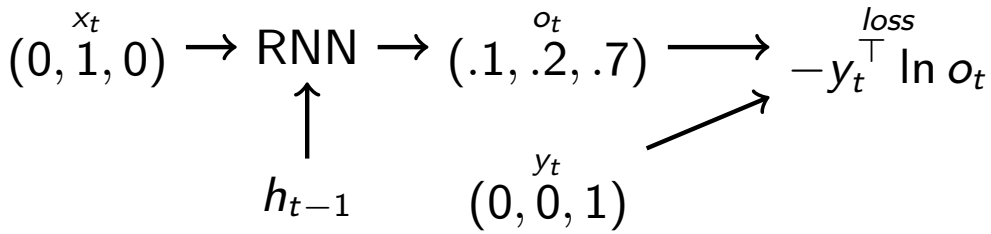
Network architecture

- We can use different RNNs. For example, we can use a vanilla RNN

$$h_t = f_W(h_{t-1}, x_t),$$
$$o_t = g_W(h_t).$$

  We can also use LSTM or GRU.
- The input layer has v neurons.
- The output layer has v neurons, fully connected to the hidden units in the last hidden layer.

Details



Training

- Train the RNN to minimize the cross-entropy loss.
  - i.e. maximize the likelihood
- In the above example, $y_t^\top \ln o_t = -0 \ln .1 - 0 \ln .2 - 1 \ln .7$.

Prediction

- For prediction, input some characters and update the hidden state.
- At each time step, we predict the most likely output character, and then feed it to the RNN as the next input character.

100 iterations

tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e plia tklrgd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng

500 iterations

we counter. He stutn co des. His stanted out one ofler that concossions and was to gearang reay Jotrets and with fre colt otf paitt thin wall. Which das stimn

1200 iterations

"Kite vouch!" he repeated by her
door. "But I would be done and quarts, feeling, then, son is people...."

2000 iterations

"Why do what that day," replied Natasha, and wishing to himself the fact the princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

Leo Tolstoy's War and Peace

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

SEED: Jobs

> *Good afternoon. God bless you.*
>
> *The United States will step up to the cost of a new challenges of the American people that will share the fact that we created the problem. They were attacked and so that they have to say that all the task of the final days of war that I will not be able to get this done. The promise of the men and women who were still going to take out the fact that the American people have fought to make sure that they have to be able to protect our part. It was a chance to stand together to completely look for the commitment to borrow from the American people. And the fact is the men and women in uniform and the millions of our country with the law system that we should be a strong stretcks of the forces that we can afford to increase our spirit of the American people and the leadership of our country who are on the Internet of American lives.*
>
> *Thank you very much. God bless you, and God bless the United States of America.*

Obama speech

http://tinyurl.com/nutq8e7

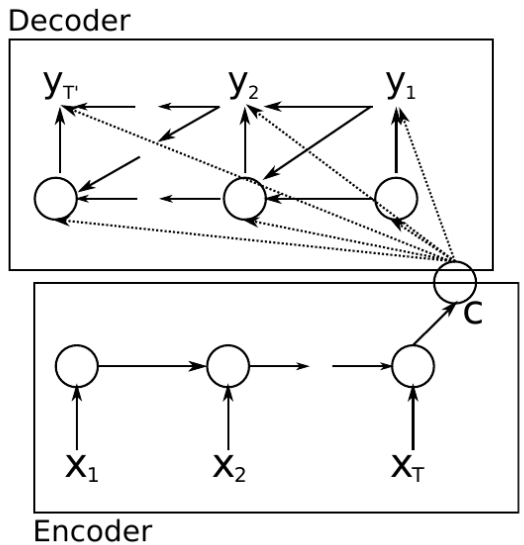
```
MMMMM----- Recipe via Meal-Master (tm) v8.05

      Title: CARAMEL CORN GARLIC BEEF
 Categories: Soups, Desserts
      Yield: 10 Servings

      2 tb Parmesan cheese, ground
    1/4 ts Ground cloves
           -- diced
      1 ts Cayenne pepper

  Cook it with the batter. Set aside to cool. Remove the peanut oil in a small saucepan and
  pour into the margarine until they are soft. Stir in a
  a mixer (dough). Add the chestnuts, beaten egg whites, oil, and salt
  and brown sugar and sugar; stir onto the boqtly brown it.

  The recipe from an oiled by fried and can. Beans, by Judil Cookbook, Source: Pintore,
  October, by Chocolates, Breammons of Jozen, Empt.com
```

Cooking recipe

https://gist.github.com/nylki/1efbaa36635956d35bcc

Machine Translation

- We can perform machine translation using a two-RNN architecture
  - The encoder RNN sequentially reads each word from the source sentence, and produces the final hidden state as a context vector c summarizing what has been seen
  - The decoder RNN produces a translation by sequentially predicting the next workd based on previous word, previous hidden state and c



Cho, Merrienboer, Gulcehre, Bougares, Schwenk, and Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014

Details

A probabilistic model

- Encoder processes each x_t in a source sentence $(x_1, \ldots, x_n)$ and updates its hidden state h^e_{t-1} sequentially using

$$h^e_t = f^e(h^e_{t-1}, x_t).$$

  The context vector c is the final hidden state h^e_n.

- Decoder defines a distribution on the translations given c via a generative process that sequentially generates $y_1, y_2, \ldots$

$$h^d_t = f^d(h^d_{t-1}, y_{t-1}, c),$$
$$y_t \sim g(\cdot \mid h^d_t, y_{t-1}, c),$$

  where g is a distribution on words in the target language.

- The generative process stops when a special end-of-sentence token is encountered.
- Thus the encoder and the decoder together specifies a distribution $p_\theta(\mathbf{y} \mid \mathbf{x})$ of a translation $\mathbf{x}$ given a source sentence $\mathbf{x}$, where θ are the parameters of the encoder and the decoder.

Training

- Given source-target sentence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$, we can train the encoder-decoder to maximize the log-likelihood

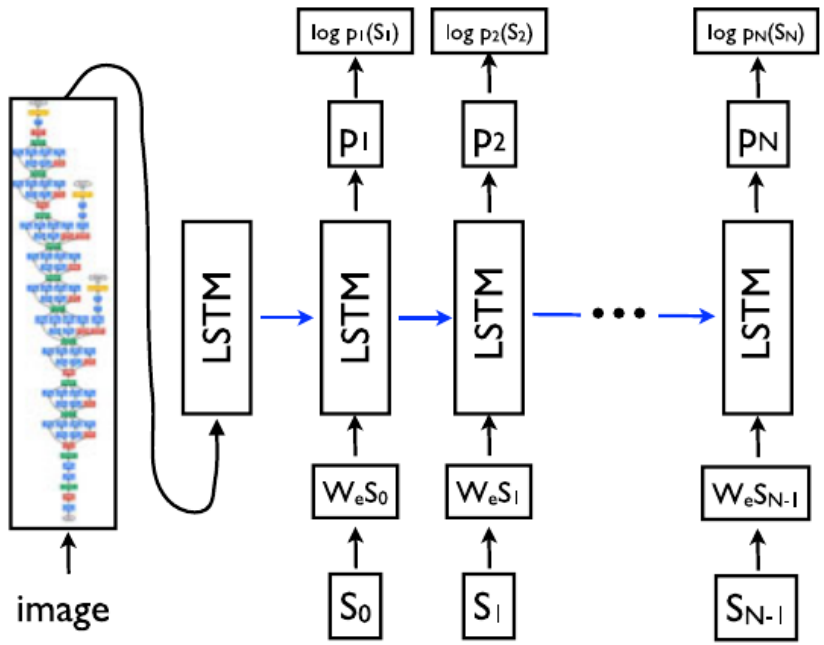
$$\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} \ln p_\theta(\mathbf{y}_i \mid \mathbf{x}_i).$$

Prediction

- Given a source sentence $\mathbf{x}$, we can use the generative process for $p_\theta(\mathbf{y} \mid \mathbf{x})$ to generate a random translation.
- A better strategy is to predict the most likely translation, but this is computationally hard.
- An approximation is to greedily predict the most likely word at each time step.

Image Captioning

- Image captioner generates a caption for a given image.
- This can be treated as a one-to-many sequence modelling problem.
- An RNN architecture

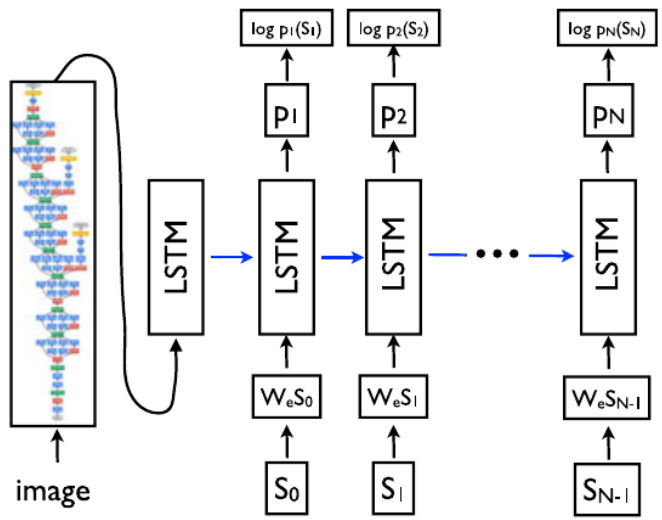


Vinyals, Toshev, Bengio, and Erhan, Show and tell: Lessons learned from the 2015 mscoco image captioning challenge, 2016

Details

Training

- Training data: images and their captions
- Architecture
  - First input vector is a CNN feature vector for the input image
  - Subsequent input vectors are *word embeddings* (real vectors, same dimension as image feature vector) of words in the caption.
  - Output at each step is a probability distribution for the next word.



;

- Given an image I and a caption $S_{1:T}$, the likelihood on this example has the form

$$p_\theta(S_{1:T} \mid I) = \prod_{t=1}^{T} p_\theta(S_t \mid I, S_{1:t-1}),$$

  where θ is the set of model parameters, and S_0 is a special start-of-sequence token $\langle SOS \rangle$.
- Training maximizes the model likelihood on the entire training set.

Prediction

- This is similar to the RNN language model.
  - Input the image feature vector and the word embedding for $\langle SOS \rangle$.
  - Predict the most likely S_1, then feed it to the RNN to predict S_2, and so on.
- The greedy prediction algorithm usually do not find the most likely sequence.
- Beam search can be used to find higher probability captions
  - Keep the top few captions generated so far at each time step, find their best extensions, prune the set of extensions.
  - At the last time step, output the caption with the largest probability among all the remaining candidates.

What You Need to Know...

- Several RNN applications
  - Language modelling: character-level RNN
  - Machine translation: the encoder-decoder architecture
  - Image captioning