CS 480/680
Introduction to Machine Learning

Lecture 15
Recurrent Neural Networks

Kathryn Simone

07 November 2024

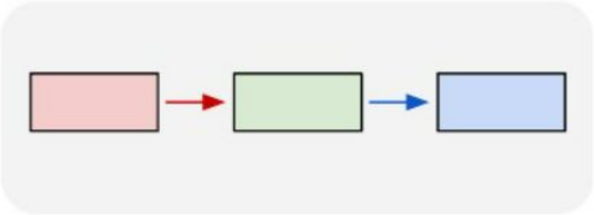
UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

We can approximate functions to map between fixed-size vector spaces

Classification: $f(\vec{x}) : \mathbb{R}^{\mathrm{d}} \to \{\pm 1\}$

Regression: $f(\vec{x}) : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$

Multivariate regression: $f(\vec{x}) : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}^{\mathrm{m}}$



could you tell me a story that involves time travel

Certainly! Here's a story that delves into the twists and turns of time travel:

In the bustling city of New York, 28-year-old Lena Walker worked as a curator at the Museum of Rare Artifacts. Her life was filled with routine: cataloging relics, arranging exhibits, and occasionally giving tours to curious tourists. But one rainy autumn evening, as she was closing the museum, Lena stumbled upon a peculiar object hidden behind a display case in the basement—a silver pocket watch, its surface engraved with an intricate spiral design and a Latin phrase: *Tempus Fugit*.

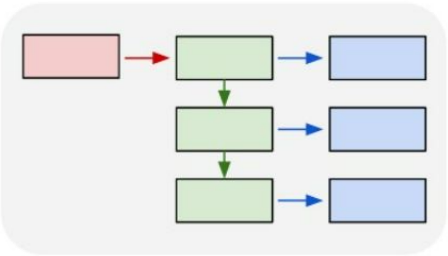
Curiosity got the best of her, and she gingerly wound the watch. To her surprise, it began to tick, though it felt oddly warm to the touch. Then, in a flash of light, the museum vanished around her, replaced by the quiet sounds of horses' hooves and distant bells.

How can I motivate the need to have models that can accept and/or generate arbitrary length sequences?

To motivate the need for models that can accept or generate arbitrary-length sequences, you can frame it around the kinds of data and tasks they are likely familiar with, showing why flexibility in input and output length is crucial. Here's a potential approach:

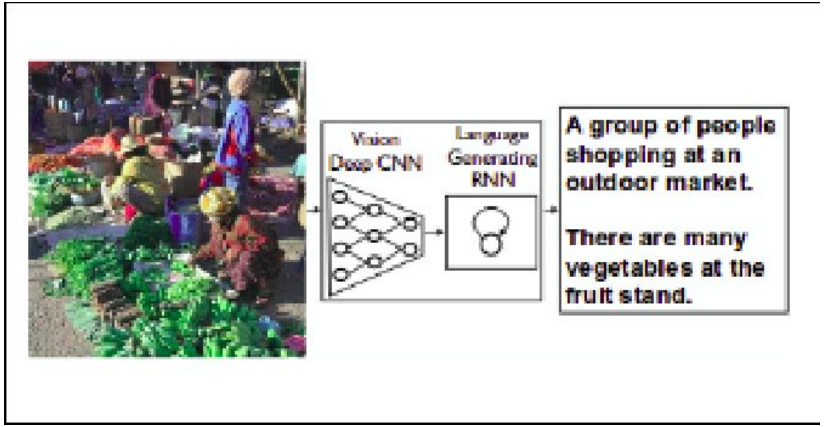
"Think about tasks you perform on a computer, like reading a news article, generating code snippets, or even interacting with Siri or Google Assistant. In all of these cases, there's no way to predict exactly how much input or output data is needed. If we want AI models to handle tasks like these, they need to be able to accept and generate sequences of any length.

Vec2Seq (One To Many)

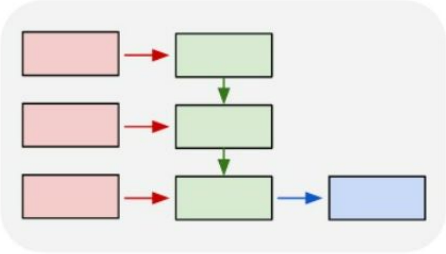


$$f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^{N_\infty C}$$

Application: Image captioning



Seq2Vec (Many to One)



$$f_{\theta} : \mathbb{R}^{TD} \to \mathbb{R}^{C}$$

Application: Sentiment analysis

Sentiment

This has to be one of the worst films of the 1990s. When my friends & I were watching this film (being the target audience it was aimed at) we just sat & watched the first half an hour with our jaws touching the floor at how bad it really was. The rest of the time, everyone else in the theater just started talking to each other, leaving or generally crying into their popcorn ...

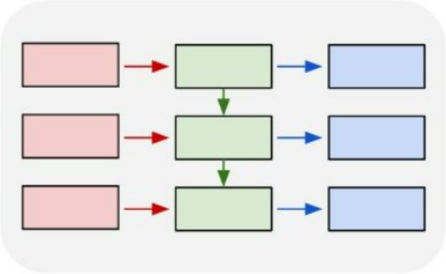
Negative

⟨START⟩ this film was just brilliant casting location scenery story direction everyone's really suited the part they played and you could just imagine being there robert ⟨UNK⟩ is an amazing actor and now the same being director ⟨UNK⟩ father came from the same scottish island as myself so i loved ...

Positive

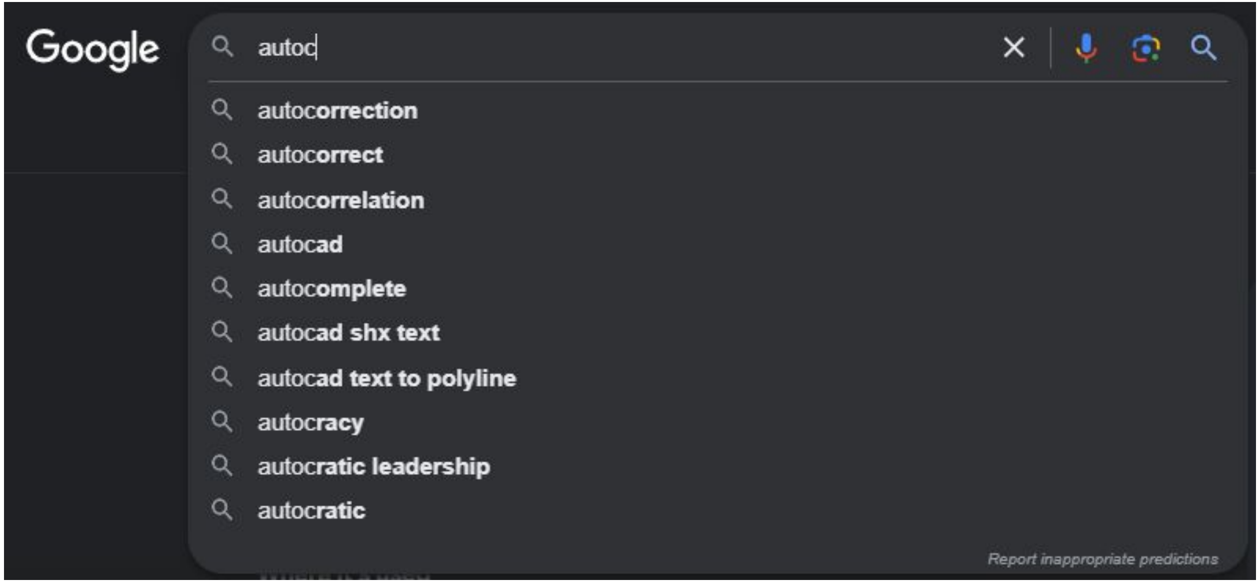
Right: Introduction to Statistical Learning Section 10.4

Seq2Seq (Many to Many)

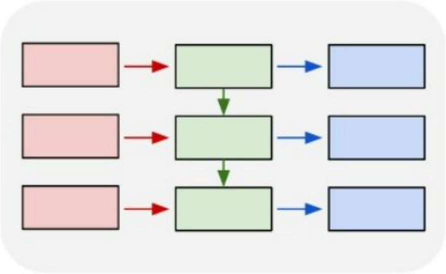


$$f_\theta : \mathbb{R}^{TD} \to \mathbb{R}^{T'C}$$

Application: Autocomplete



Seq2Seq (Many to Many)

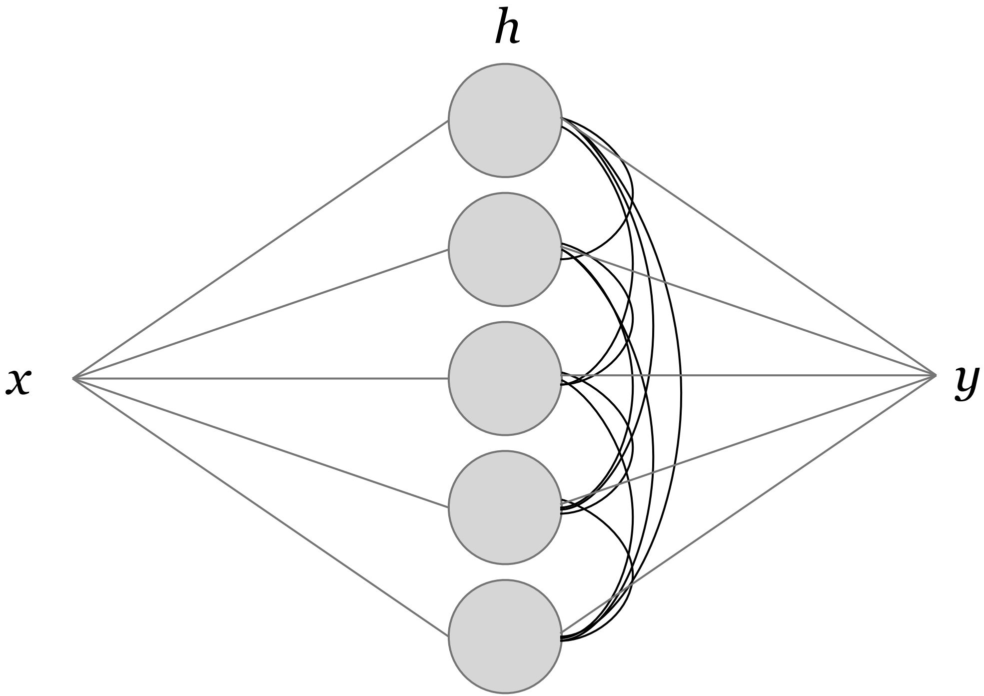


$$f_\theta : \mathbb{R}^{TD} \to \mathbb{R}^{T'C}$$

Application: Machine Translation

<| French → {Reprise de la session,
Je déclare reprise la session du Parlement européen qui avait été interrompue le vendredi 17 décembre dernier et je vous renouvelle tous mes vux en espérant que vous avez passé de bonnes vacances.,
Comme vous avez pu le constater, le grand "bogue de l'an 2000" ne s'est pas produit. En revanche, les citoyens d'un certain nombre de nos pays ont été victimes de catastrophes naturelles qui ont vraiment été terribles.},
English → {Resumption of the session, I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.,
Although, as you will have seen, the dreaded 'millennium bug' failed to materialise, still the people in a number of countries suffered a series of natural disasters that truly were dreadful.} |>

Recurrent Neural Networks are a *diverse* family of networks specialized for processing sequential data



Deep Learning Section 10.1

Key questions

I. **How can we formulate the sequence generation problem?**

II. **How do we estimate the parameters?**

III. **Are there other ways to build the network?**

Key questions

I. **How can we formulate the sequence generation problem?**

II. **How do we estimate the parameters?**

III. **Are there other ways to build the network?**

Conditional generative model for sequence generation

$$p(y_{1:T}|x)$$

$$= \sum_{h_{1:T}} p(y_{1:T}, h_{1:T}|x)$$

$$= \sum_{h_{1:T}} \prod_{t=1}^{T} p(y_t|h_t) p(h_t|h_{t-1}, y_{t-1}, x)$$

given $p(h_1|h_0, y_0, x) = p(h_1|x)$,
initial distribution over hidden states

$y_{1:T}$: Sequence of targets from time step 1 to T

x : input data

$h_{1:T}$: Sequence of hidden states

Marginalize the joint probability:

$$p_X(x_i) = \sum_j p(x_i, y_j)$$

Chain rule of probability

$$p(X_1, X_2, \ldots, X_n)$$
$$= p(X_1)\, p(X_2|X_1)\, p(X_3|X_1, X_2) \cdots$$
$$\cdots p(X_n|X_1, X_2, \ldots, X_{n-1})$$

Probabilistic Machine Learning Section 15.2

Sequence generation requires maintaining a context

Consider generation of the sequence

HELLO

Given the limited alphabet

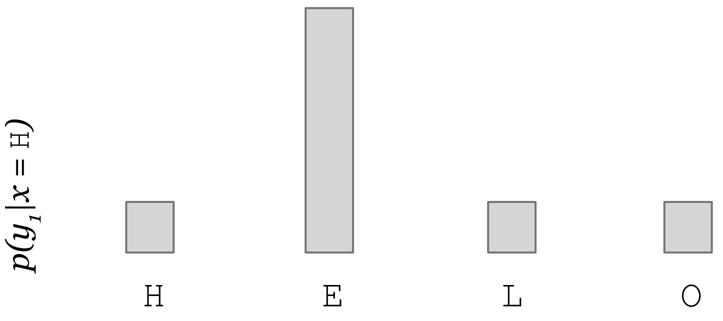
H E L O

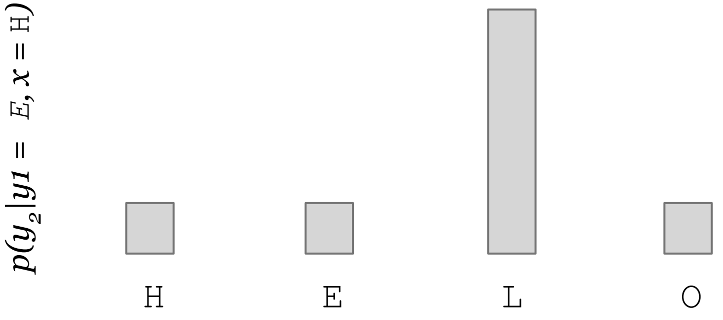
Context:

$x = \mathrm{H}$ | $x = \mathrm{H}$

$y_1 = \mathrm{E}$ | $y1 = E, x = \mathrm{H}$

$y_2 = \mathrm{L}$





Character-level RNN implementation

Initial State:

h_1

Sample:

$p(y_1|h_1) = g(h_1)$

$\tilde{y}_1 \sim p(y_1|h_1)$

Update State:

$h_2 = f(h_1, y_1, x)$

$p(y_2|h_2) = g(h_2)$

$\tilde{y}_2 \sim p(y_2|h_2)$

Where:

$$f(h_t, y_t, x) = \tanh\left(W_{xh}[x; y_t] + W_{hh}h_{t-1} + b_h\right)$$

$$g(h_t) = \mathrm{softmax}(W_{hy}h_t + b_y)$$

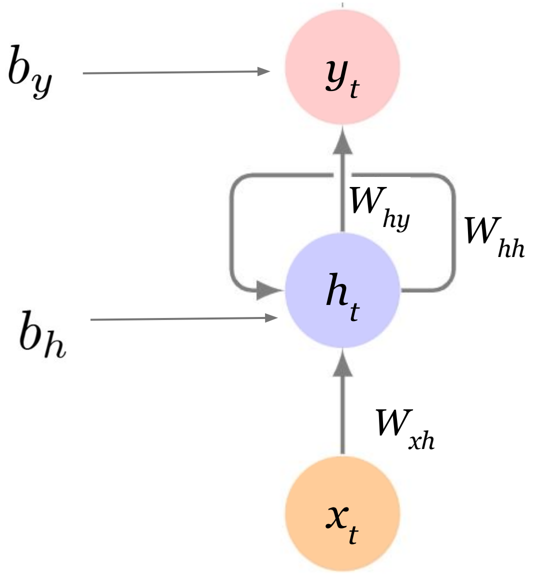
Character-Level RNN Implementation: Architecture

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

$$p_t = \frac{\exp(y_t)}{\sum_i \exp(y_{t,i})}$$

$$\tilde{y}_{t+1} \sim \text{Categorical}(p_t)$$

$$L = -\sum_{t=1}^{T} \log(p_t[\text{target}])$$



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

Key questions

I. How can we formulate the sequence generation problem?

II. How do we estimate the parameters?

III. Are there other ways to build the network?

Unrolling an RNN

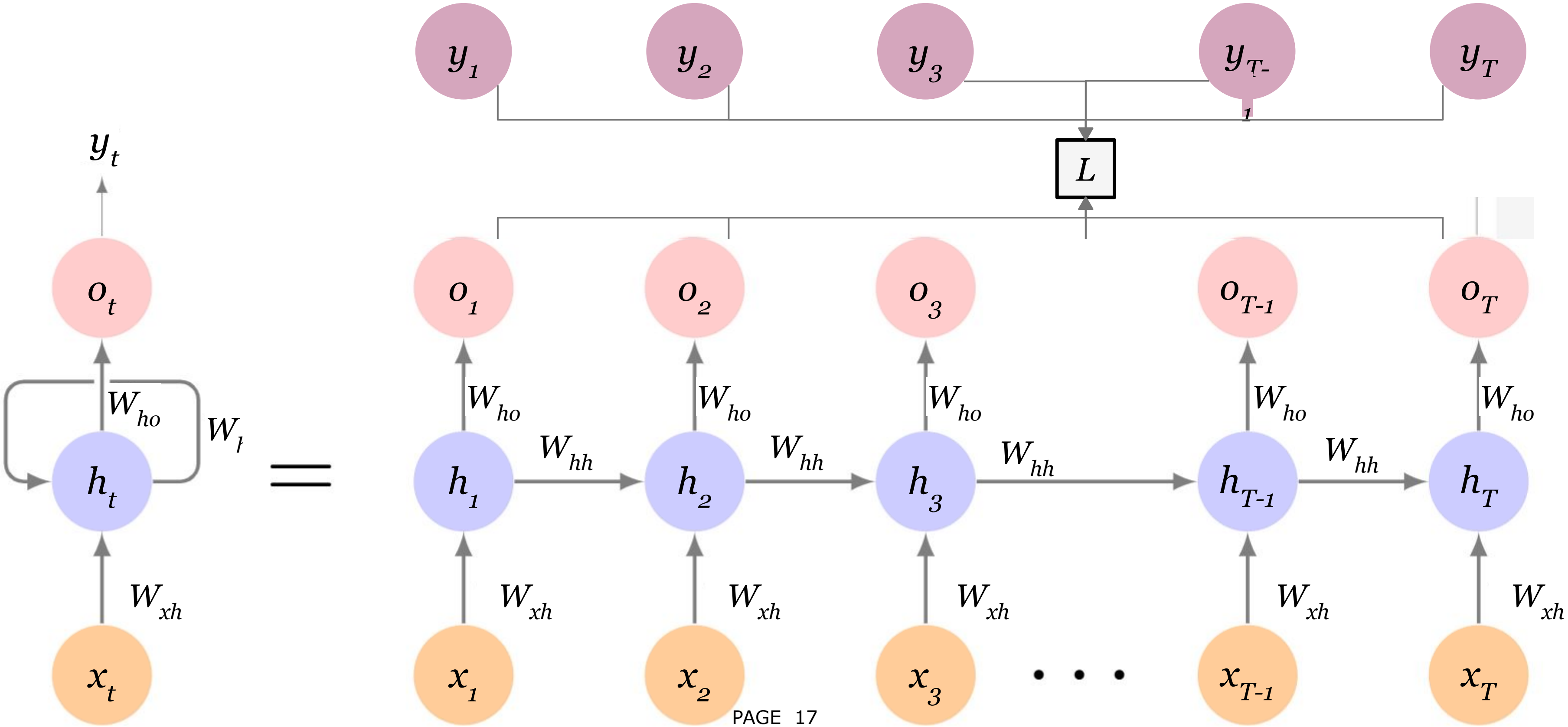


Figure Adapted from Introduction to Statistical Learning 10.5

Maximum Likelihood Estimation for an RNN

$$\theta^* = \arg\max_\theta p(y_{1:T}|x_{1:T}, \theta)$$

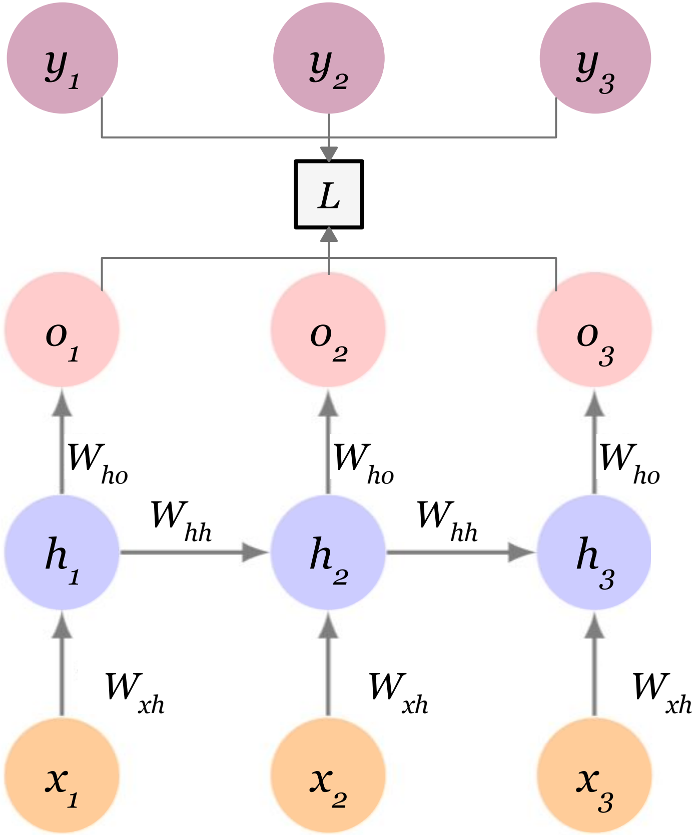
y_t true target label at step t

$$h_t = f(x_t, h_{t-1}, w_h),$$
$$o_t = g(h_t, w_o)$$

h_t hidden state

o_t output

$$\frac{\partial L}{\partial w_h} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial l(y_t, o_t)}{\partial w_h}$$
$$= \frac{1}{T}\sum_{t=1}^{T}\frac{\partial l(y_t, o_t)}{\partial o_t}\frac{\partial g(h_t, w_o)}{\partial h_t}\frac{\partial h_t}{\partial w_h}$$



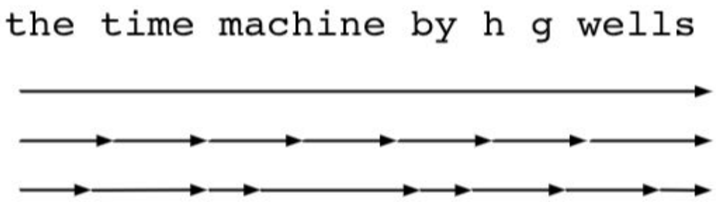
Dive into Deep Learning Section 9.7

Gradients and backpropagation through time (BPTT)

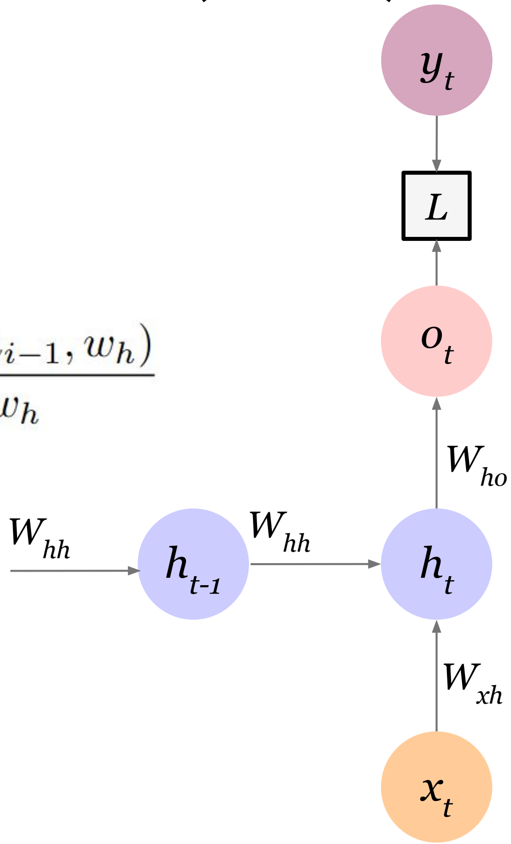
$$h_t = f(x_t, h_{t-1}, w_h),$$

$$\frac{\partial h_t}{\partial w_h} = \frac{\partial f(x_t, h_{t-1}, w_h)}{\partial w_h} + \frac{\partial f(x_t, h_{t-1}, w_h)}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial w_h}$$

$$\frac{\partial h_t}{\partial w_h} = \frac{\partial f(x_t, h_{t-1}, w_h)}{\partial w_h} + \sum_{i=1}^{t-1} \left(\prod_{j=i+1}^{t} \frac{\partial f(x_j, h_{j-1}, w_h)}{\partial h_{j-1}} \right) \frac{\partial f(x_i, h_{i-1}, w_h)}{\partial w_h}$$

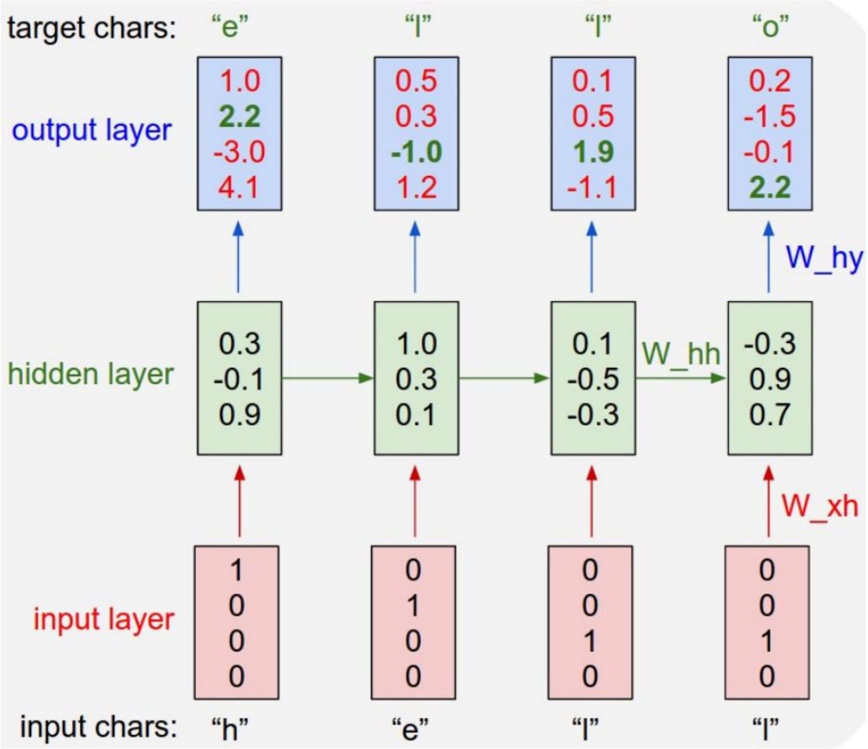


Full BPTT
Truncated BPTT
Randomized Truncated BPTT



Dive into Deep Learning Section 9.7

Character-Level RNN



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

Truncated BPTT for character prediction

```
Training Data:
 A glaring limitation of Vanilla Neural Networks (and also Convolutional Networks) is that their API is too constrained: the
y accept a fixed-sized vector as input (e.g. an image) and produce a fixed-sized vector as output (e.g. probabilities of dif
ferent classes).
```

```
Iteration 0 (loss: 89.587973) Output:
 ----
 r:Af(pv-)enfkrCroIg-rnmVnIprya-:PyClrgsfzVxa(ddPvh-abx()xrIpgaCnIAiyuAm.)AeAVp.PIt:zkihVbakNyupxn.yz.otVwIdarozdIi:dsmckCn)
i.gVVywrh(fprPnzcoyolaheAVPAwabtzC:bbxitVIsPv(:.nmrhkn(p:Iyaxavrhnhd.m(krsAuAPuph:CVPnVeaVNevtrwt(Clv(r(I)x(gIy:P:.trChkd:VV
hsz Ngtu-z:afuI
```

```
Iteration 500 (loss: 79.722071) Output:
 ----
  glarothctibinod tifnmg szecto vatiutioe. k nto: xidiois Ned ogut izez-ct prcr opnorkso anibinepro a)t atine. a ranle)-ed a
s aned asstiof af ti(e.: imimnased drsi ed) fsd ved fsd ed pnar ansehedied olenizecPece ne.lut itd. bapr aize) pnVnuft alkol
utnifltif ticed
```

```
Iteration 2500 (loss: 13.077288) Output:
 ----
  glaring limitation of Vanilla Noural Networks (and also Convolutional Networks) is that their API is too constrained: they
accept thetrababilities of differe Vand also Conalls itg. pAe too ce Convolutional Networks) is that their AhInsieutput (e.
g. probabilitie
```

```
Iteration 10000 (loss: 0.072970) Output:
 ----
  glaring limitation of Vanilla Neural Networks (and also Convolutional Networks) is that their API is too constrained: they
accept a fixed-sized vector as input (e.g. an image) and produce a fixed-sized vector as output (e.g. probabilities of diffe
red vector as
```


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

Truncated BPTT struggles with long-range dependencies

```
Training Data:
 If training vanilla neural nets is optimization over functions, training recurrent nets is optimization over programs.
```

```
Iteration 100 (loss: 76.656102) Output:
 ----
 eztrtioio.ngnviifunitpfpm tItntitenpennucifptioif piaiounet vpzst ziniI refcnsocn,snlu is f tinInvtinaopt rutasinutsn
```

```
Iteration 1000 (loss: 40.152319) Output:
 ----
 f training vanilla neural ner optimizationtimization over functions, training recurrent nets is optimizations, traini
```

```
Iteration 10000 (loss: 0.070277) Output:
 ----
 f training vanilla neural nets is optimization over functions, training recurrent nets is optimization over functions
```

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

Key questions

I. How can we formulate the sequence generation problem?

II. How do we estimate the parameters?

III. **Are there other ways to build the network?**

Long Short-Term Memory (LSTM)
(Hochreiter and Schmidhuber 1997)

Forget Gate:

$$F_t = \sigma(W_f \cdot [H_{t-1}, X_t] + b_f)$$

Input Gate:

$$I_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, X_t] + b_c)$$

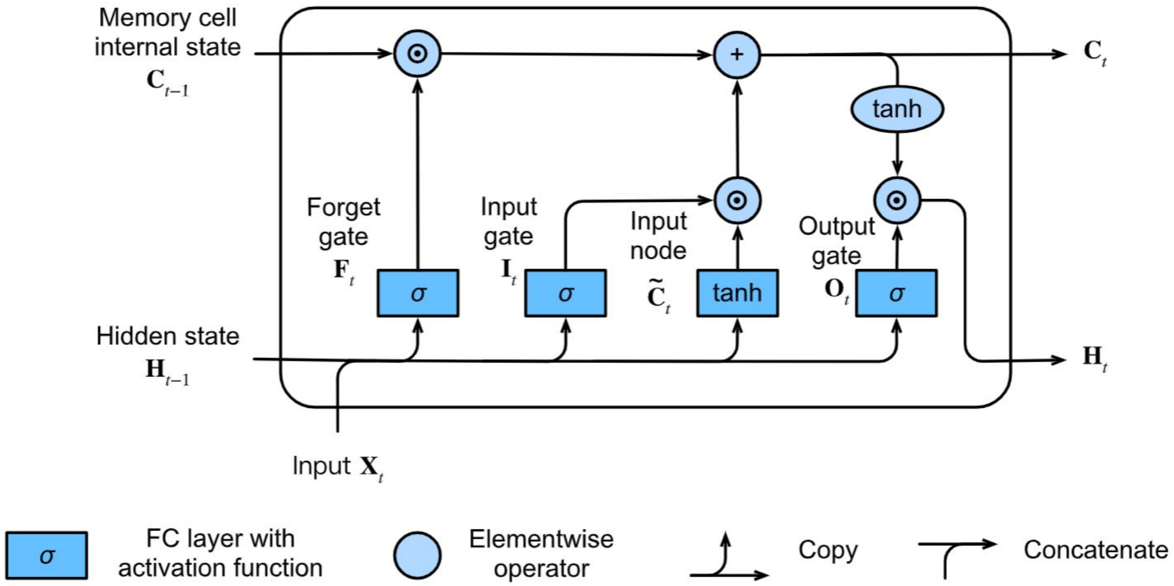
Update Cell State:

$$C_t = F_t \times C_{t-1} + I_t * \tilde{C}_t$$

Output Gate:

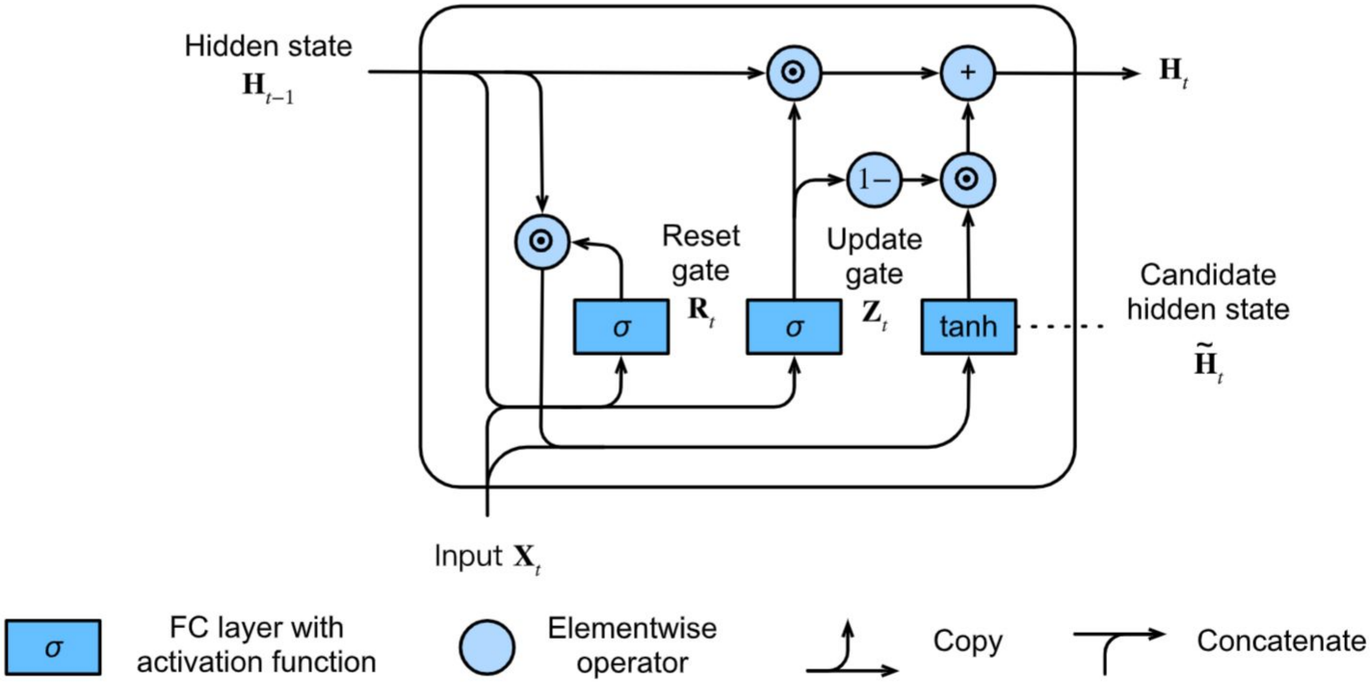
$$O_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o)$$

$$H_t = O_t \times \tanh(C_t)$$



Dive into Deep Learning Section 10.1

Gated Recurrent Unit (GRU)



Dive into Deep Learning Section 10.2

Reservoir Computers

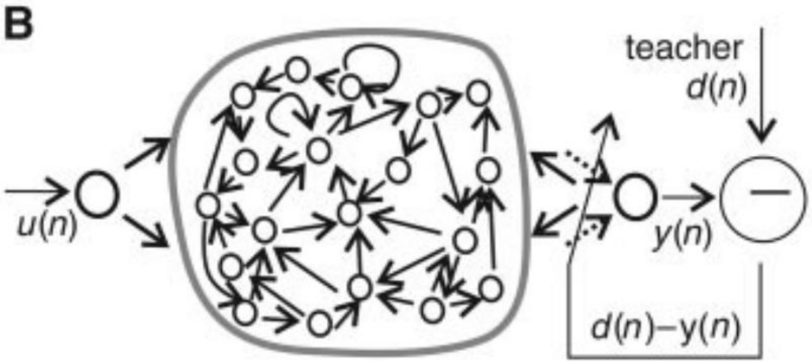
Ensemble of recurrently-connected neurons

- Sparse, random connections
- Generates random ***temporal features*** of the input
- Weights must be selected to ensure stability

Learn a function to perform classification or regression on those features

Two closely-related methods:

- Echo State Network, Liquid State Machine



Right: *Jaeger and Haas, 2004*

Approximating a pure delay with Legendre basis set

$F(s) = e^{-\theta s}$ Laplace representation of a pure delay

$$\theta \dot{m}(t) = \mathbf{A}m(t) + \mathbf{B}u(t)$$

$u(t) \in \mathbb{R}$: input signal

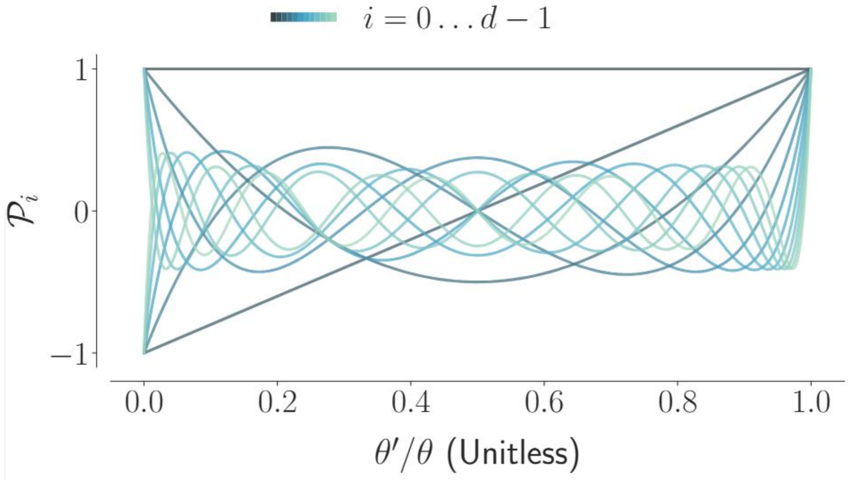
$\theta \in \mathbb{R}_{>0}$: window length

$$\mathbf{A} = [a_{ij}] \in \mathbb{R}^{d\times d}$$

$$a_{ij} = (2i+1)\begin{cases} -1 & i < j \\ (-1)^{i-j+1} & i \geq j \end{cases}$$

$$\mathbf{B} = [b_i] \in \mathbb{R}^{d\times 1}$$

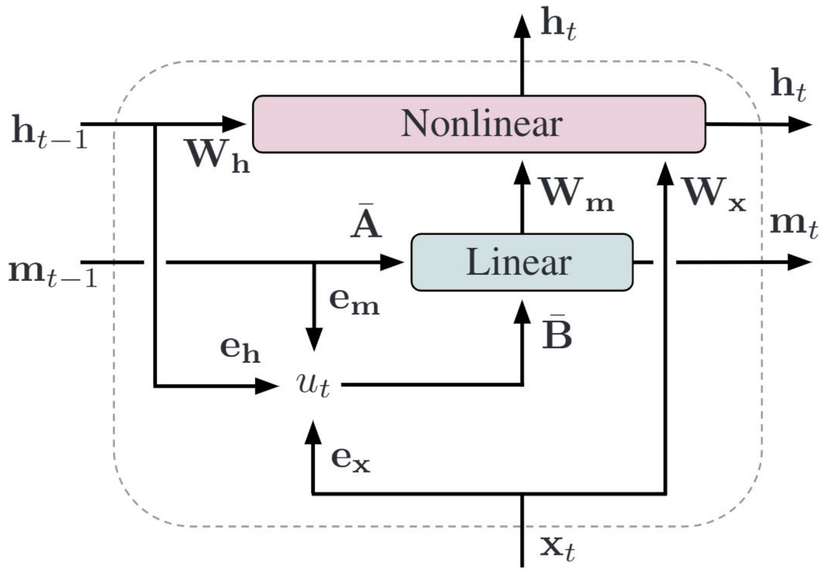
$$b_i = (2i+1)(-1)^i, \quad i, j \in [0, d-1]$$

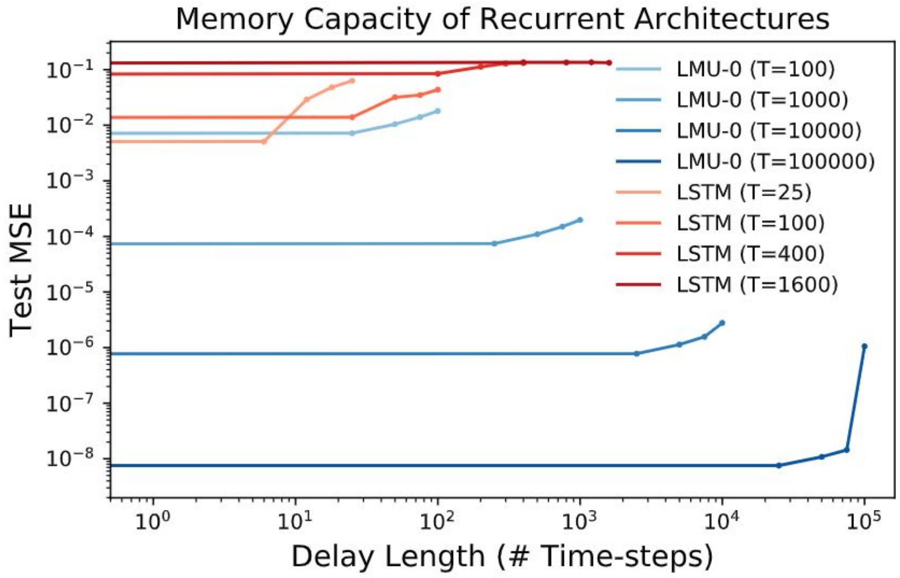


$$u(t-\theta') \approx \sum_{i=0}^{d-1} P_i\left(\frac{\theta'}{\theta}\right) m_i(t)$$

$$P_i(r) = (-1)^i \sum_{j=0}^{i} \binom{i}{j}\binom{i+j}{j}(-r)^j$$

Legendre Memory Unit
(Voelker, 2019)





Now that we're at the end of the lecture, you should be able to…

- ★ Define **backpropagation through time (BPTT)** and explain its relation to building an RNN.
- ★ Implement a **character-level RNN** for sequence generation.
- ★ Recall three other approaches to building a recurrent neural network that alleviate or avoid the issues associated with backpropagation through time.
- ★ Label the **components of an LSTM or GRU memory cell**, and explain their role with reference to the **hidden and/or internal cell states**.

Errata and Changes

- On slide 14, in a previous version of the slides the equation for the hidden state at timestep t depended on the hidden state at timestep t. This should have been h_{t-1}. This has been fixed (03/12/2024)
- Graphics appearing on slides 14-19 have been replaced with qualitatively similar ones to those used in the lecture, to aid in the explanation provided in the recitation.