Topics:

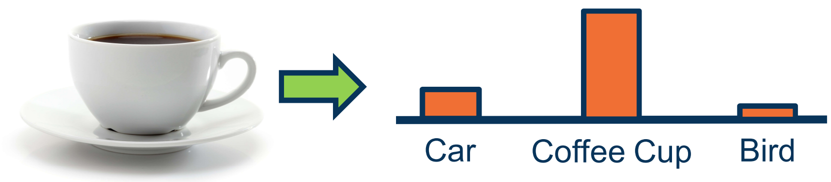
- Bias, Fairness, Calibration
- Structured Representations and recurrent neural networks

CS 4644-DL / 7643-A
ZSOLT KIRA

- **Assignment 3**
  - Due **March 14th 11:59pm EST.**
  - See https://piazza.com/class/ky0k0ha5vgy1mk?cid=176
    - (note: ignore logistics on that slide deck)

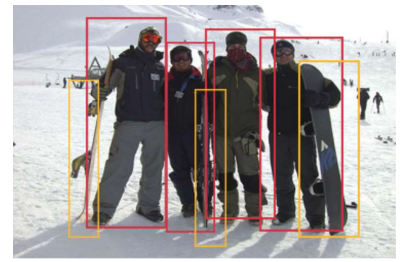
- **Projects**
  - Project proposal due **March 13th**

- **Meta Office Hours on Language Models Friday 3pm EST**



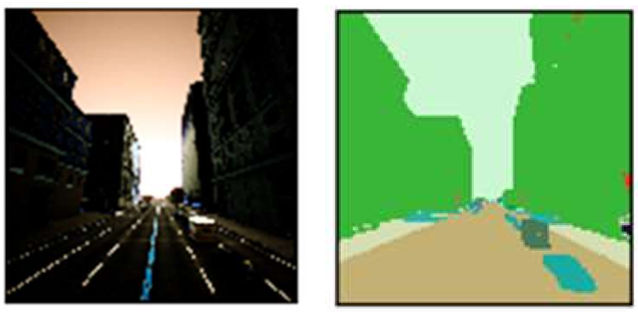
Classification

(Class distribution per image)



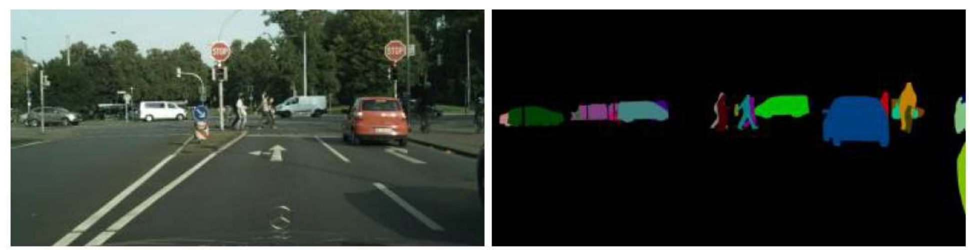
Object Detection

(List of bounding boxes with class distribution per box)



Semantic Segmentation

(Class distribution per pixel)



Instance Segmentation

(Class distribution per pixel with unique ID)

Computer Vision Tasks

Bias & Fairness

FACEBOOK AI Georgia Tech

ML and Fairness

- AI effects our lives in many ways
- Widespread algorithms with many small interactions
  - e.g. search, recommendations, social media
- Specialized algorithms with fewer but higher-stakes interactions
  - e.g. medicine, criminal justice, finance
- At this level of impact, algorithms can have unintended consequences
- Low classification error is not enough, need fairness

BUSINESS NEWS OCTOBER 10, 2018 / 3:12 AM / 6 MONTHS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin 8 MIN READ

SAN FRANCISCO (Reuters) - Amazon.com Inc's (AMZN.O) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

The team had been building computer programs since 2014 to review job applicants' resumes with the aim of mechanizing the search for top talent, five people familiar with the effort told Reuters.

Automation has been key to Amazon's e-commerce dominance, be it inside warehouses or driving pricing decisions. The company's experimental hiring tool used artificial intelligence to give job candidates scores ranging from one to five stars - much like

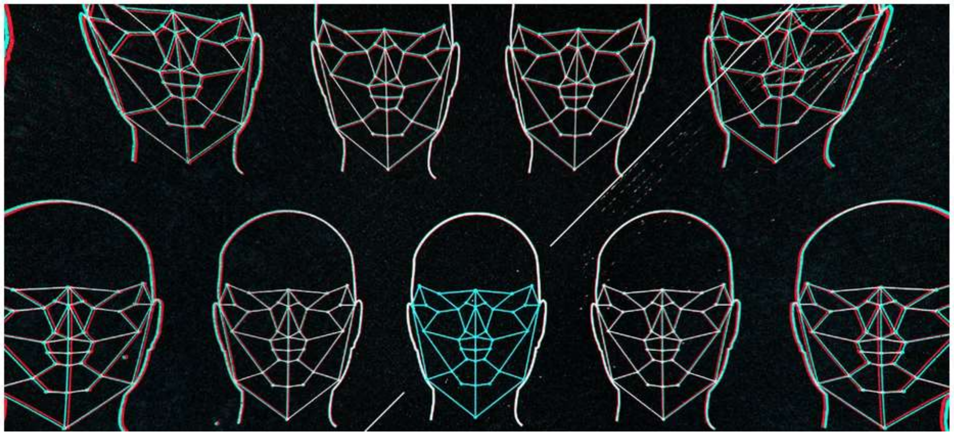
Gender and racial bias found in Amazon's facial recognition technology (again)

17

Research shows that Amazon's tech has a harder time identifying gender in darker-skinned and female faces

By James Vincent | Jan 25, 2019, 9:45am EST

SHARE



MOST READ

My Samsung Galaxy Fold screen broke after just a day

We finally know why the Instagram founders really quit

Command Line

Command Line delivers daily updates from the near-future.

ML and Fairness

- Fairness is morally and legally motivated
- Takes many forms
- Criminal justice: recidivism algorithms (COMPAS)
  - Predicting if a defendant should receive bail
  - Unbalanced false positive rates: more likely to wrongly deny a black person bail

Table 1: ProPublica Analysis of COMPAS Algorithm

	White	Black
Wrongly Labeled High-Risk	23.5%	44.9%
Wrongly Labeled Low-Risk	47.7%	28.0%

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Why Fairness is Hard

- Suppose we are a bank trying to fairly decide who should get a loan
  - i.e. Who is most likely to pay us back?
- Suppose we have two groups, A and B (the sensitive attribute)
  - This is where discrimination could occur
- The simplest approach is to remove the sensitive attribute from the data, so that our classier doesn't know the sensitive attribute

Table 2: To Loan or Not to Loan?

Age	Gender	Postal Code	Req Amt	A or B?	Pay
46	F	M5E	$300	A	1
24	M	M4C	$1000	B	1
33	M	M3H	$250	A	1
34	F	M9C	$2000	A	0
71	F	M3B	$200	A	0
28	M	M5W	$1500	B	0

Why Fairness is Hard

- However, if the sensitive attribute is correlated with the other attributes, this isn't good enough
- It is easy to predict race if you have lots of other information (e.g. home address, spending patterns)
- More advanced approaches are necessary

Table 3: To Loan or Not to Loan? (masked)

Age	Gender	Postal Code	Req Amt	A or B?	Pay
46	F	M5E	$300	?	1
24	M	M4C	$1000	?	1
33	M	M3H	$250	?	1
34	F	M9C	$2000	?	0
71	F	M3B	$200	?	0
28	M	M5W	$1500	?	0

Definitions of Fairness – Group Fairness

- So we've built our classier . . . how do we know if we're being fair?
- One metric is demographic parity | requiring that the same percentage of A and B receive loans
  - What if 80% of A is likely to repay, but only 60% of B is?
  - Then demographic parity is too strong
- Could require equal false positive/negative rates
  - When we make an error, the direction of that error is equally likely for both groups

$$P(loan|no\ repay, A) = P(loan|no\ repay, B)$$
$$P(no\ loan|would\ repay, A) = P(no\ loan|would\ repay, B)$$

- These are definitions of group fairness
- Treat different groups equally"

Definitions of Fairness – Individual Fairness

- Also can talk about individual fairness | “Treat similar examples similarly"
- Learn fair representations
  - Useful for classification, not for (unfair) discrimination
  - Related to domain adaptation
  - Generative modelling/adversarial approaches

(a) Unfair representations (b) Fair(er) representations

Figure 1: “The Variational Fair Autoencoder” (Louizos et al., 2016)

Conclusion

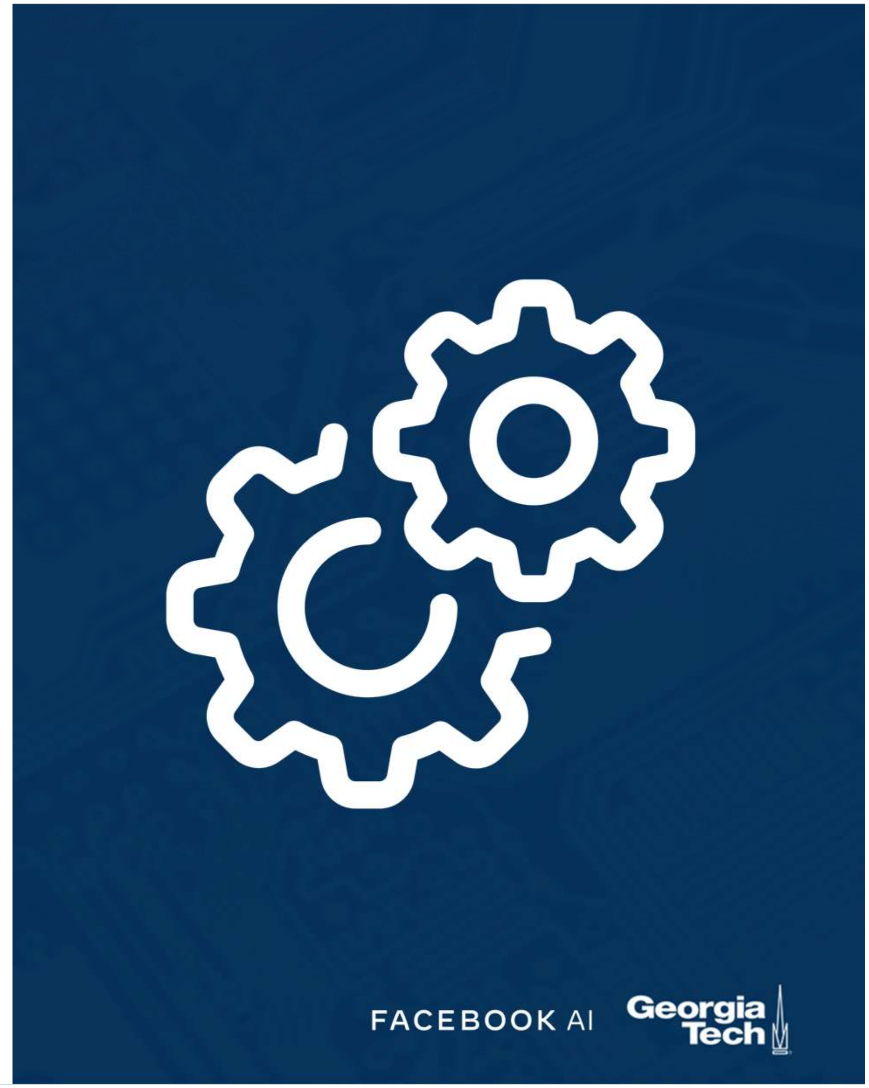
- This is an exciting field, quickly developing
- Central definitions still up in the air
- AI moves fast | lots of (currently unchecked) power
- Law/policy will one day catch up with technology
- Those who work with AI should be ready
  - **Think about implications of what you develop!**

Calibration

FACEBOOK AI Georgia Tech

Calibration

- Definition
- Measuring Calibration
- Calibrating models
- Limitations of Calibration



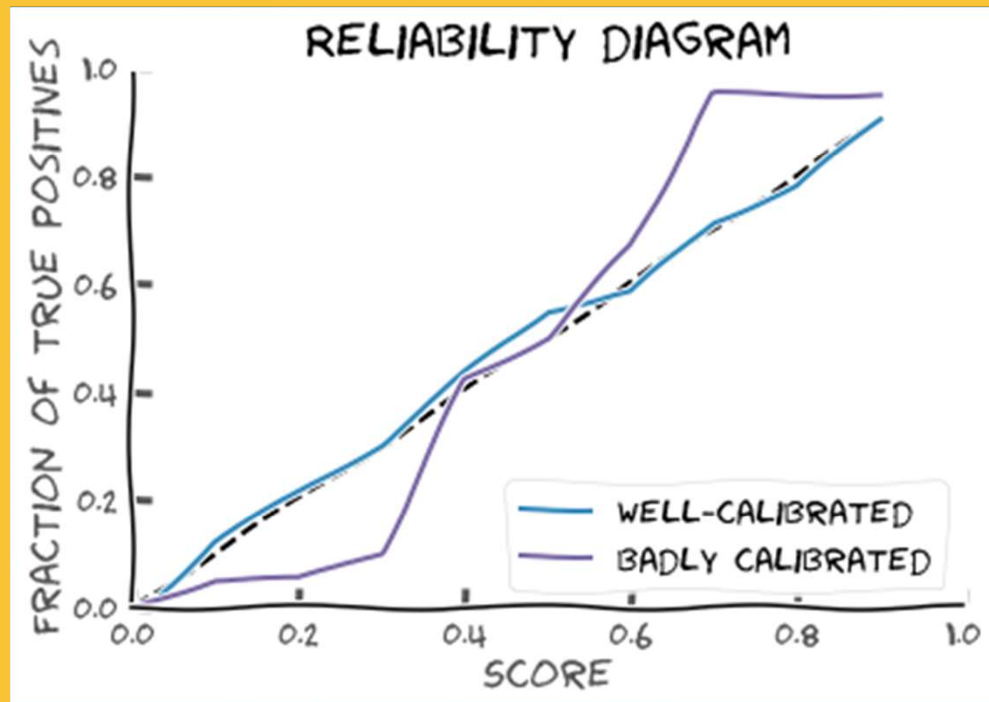
A classifier is **well-calibrated** if the probability of the observations with a given probability score of having a label is equal to the proportion of observations having that label

- **Example:** if a binary classifier gives a score of 0.8 to 100 observations, then 80 of them should be in the positive class

$$\forall p \in [0,1], P(\hat{Y} = Y | \hat{P} = p) = p$$

where $\widehat{\boldsymbol{Y}}$ is the predicted label and $\widehat{\boldsymbol{P}}$ is the predicted probability (or score) for class Y

Calibration: Definition

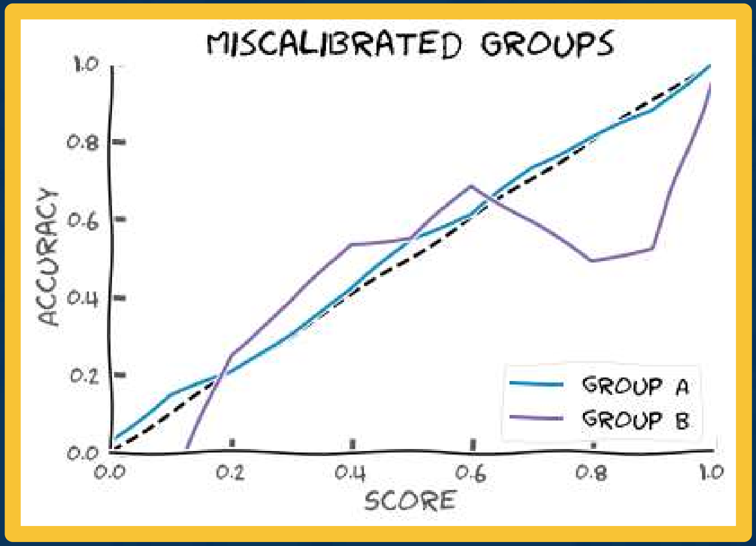


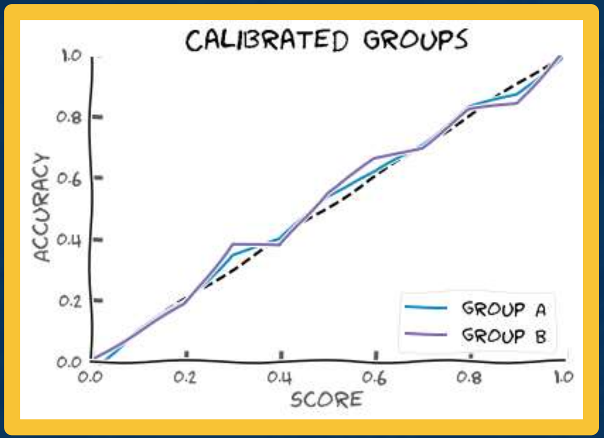
FACEBOOK AI

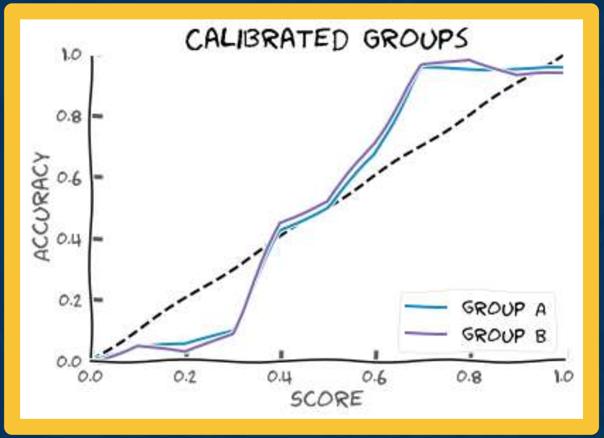
Georgia Tech

Calibration: Definition

Group Calibration: the scores for subgroups of interest are calibrated (or at least, equally mis-calibrated)







FACEBOOK AI Georgia Tech

- Some models (e.g Logistic Regression) tend to have **well-calibrated predictions**
- Some DL models (e.g. ResNet) tend to be **overconfident** (*https://arxiv.org/pdf/1706.04599.pdf*)
- **Logistic calibration/Platt scaling**

Post-processing approach requiring an **additional validation dataset**

Platt scaling (binary classifier)

- Learn parameters $\boldsymbol{a}, \boldsymbol{b}$ so that the **calibrated probability** is $\widehat{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{\sigma}(\boldsymbol{a}\boldsymbol{z}_{\boldsymbol{i}} + \boldsymbol{b})$)where $\boldsymbol{z}_{\boldsymbol{i}}$ is the network's logit output)

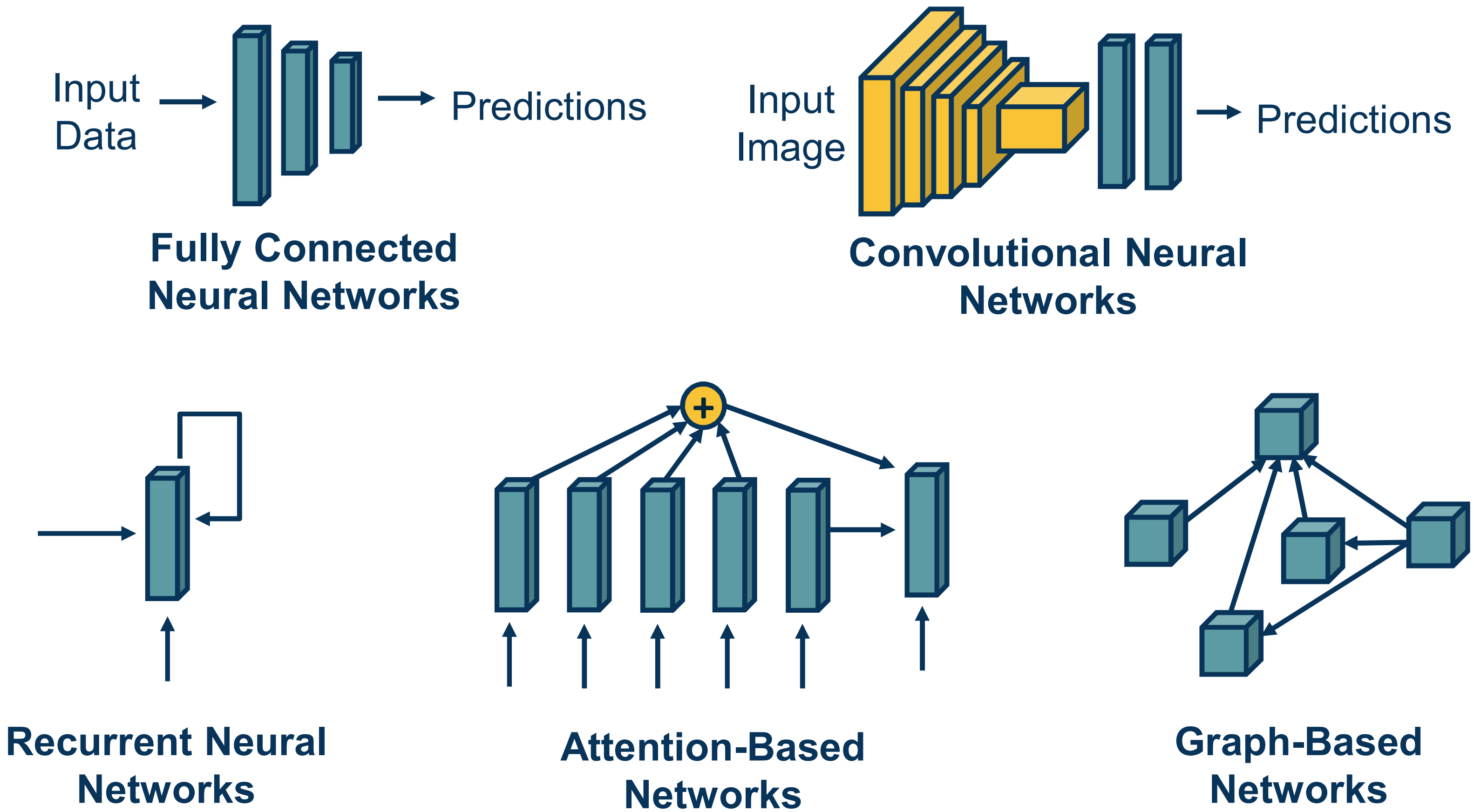
Temperature scaling extends this to multi-class classification

- Learn a temperature $\boldsymbol{T}$, and produce calibrated probabilities $\widehat{\boldsymbol{q}}_{\boldsymbol{i}} = \max_{\boldsymbol{k}} \boldsymbol{\sigma}_{\boldsymbol{SoftMax}}(\boldsymbol{z}_{\boldsymbol{i}}/\boldsymbol{T})$

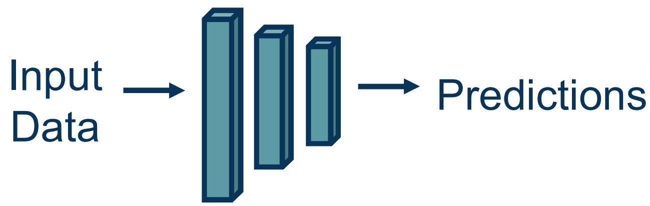
Calibration: Limitations

- Group based
- The Inherent Tradeoffs of Calibration

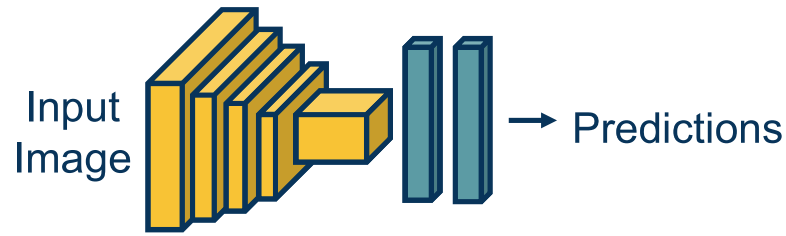
Module 3 Introduction



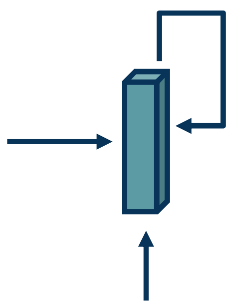
The Space of Architectures



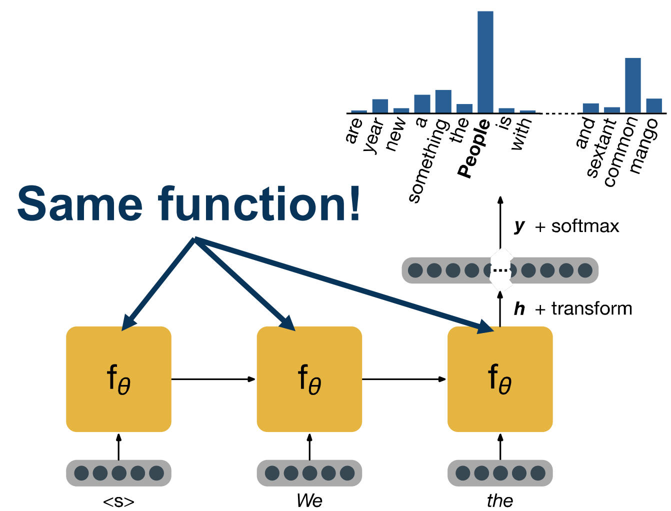
Fully Connected Neural Networks



Convolutional Neural Networks

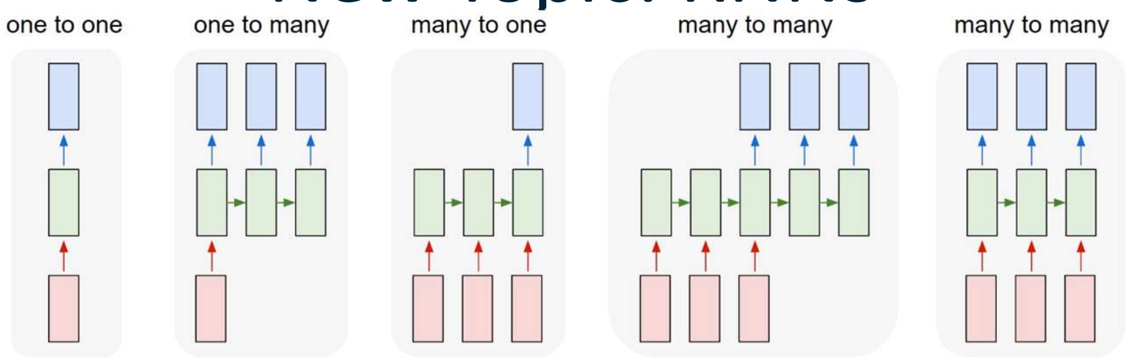


Recurrent Neural Networks



Recurrent Neural Networks

New Topic: RNNs



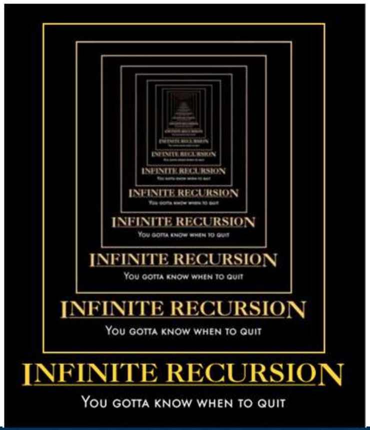
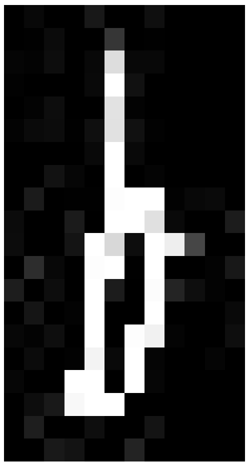


Image Credit: Andrej Karpathy

Why model sequences?



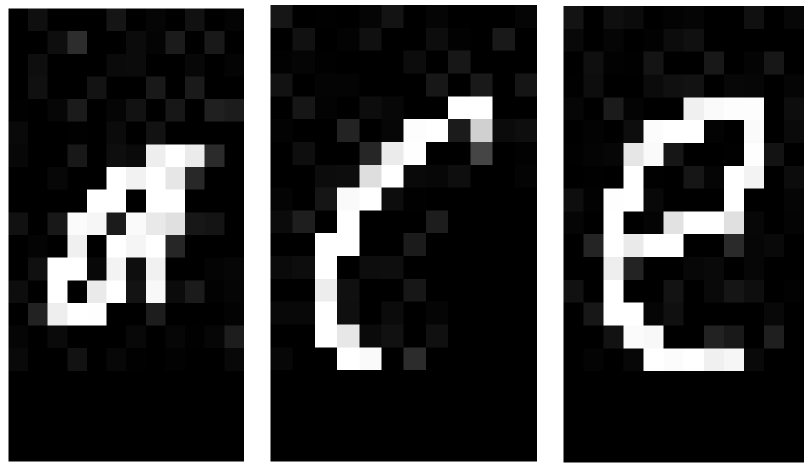


Figure Credit: Carlos Guestrin

Sequences are everywhere…

FOREIGN MINISTER.

THE SOUND OF

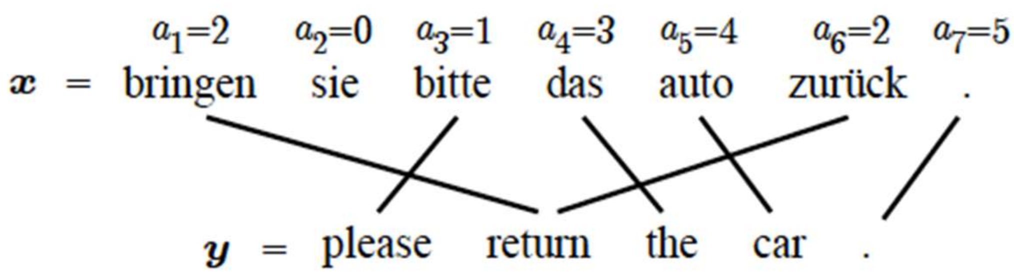
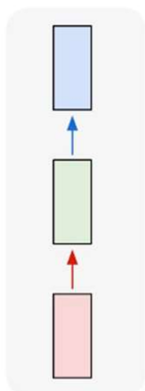


Image Credit: Alex Graves and Kevin Gimpel

Sequences in Input or Output?

- It's a spectrum…

one to one



Input: No sequence

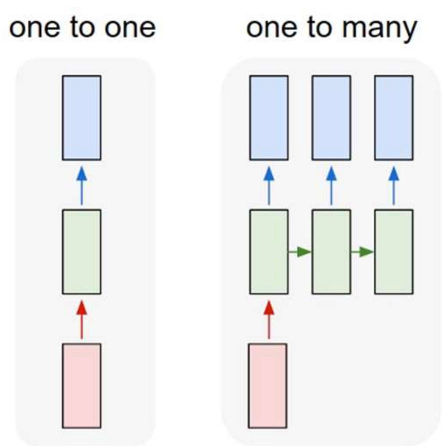
Output: No sequence

Example: "standard" classification / regression problems

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

Output: No sequence

Example: "standard" classification / regression problems

Input: No sequence

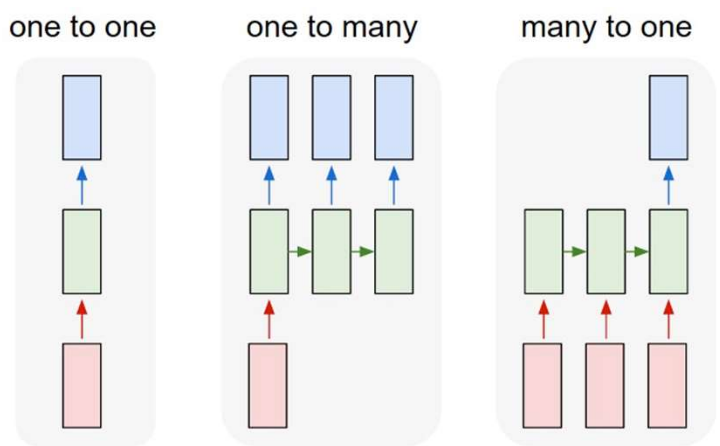
Output: Sequence

Example: Im2Caption

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

Output: No sequence

Example: "standard" classification / regression problems

Input: No sequence

Output: Sequence

Example: Im2Caption

Input: Sequence

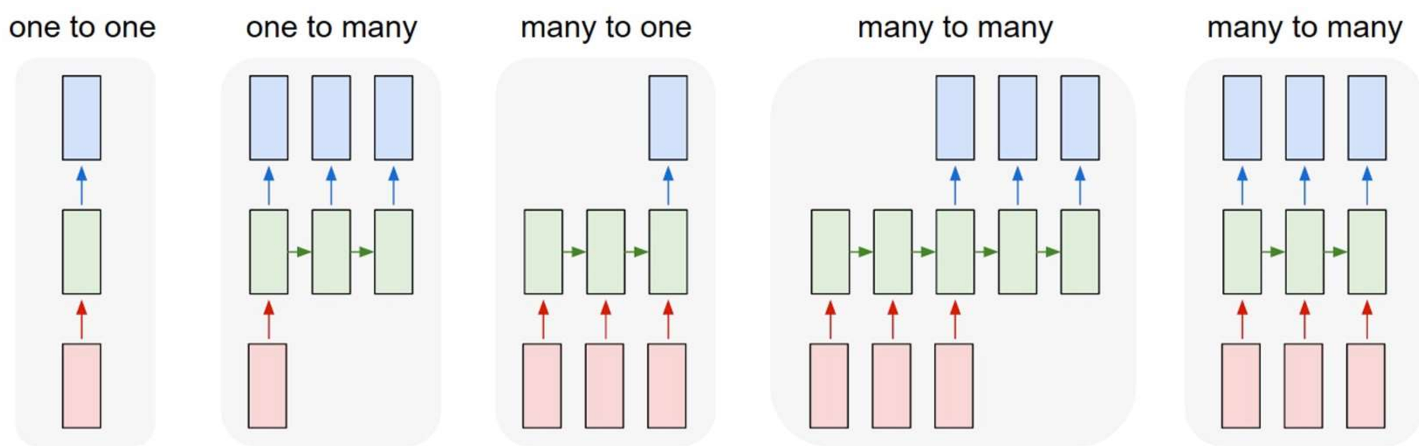
Output: No sequence

Example: sentence classification, multiple-choice question answering

Image Credit: Andrej Karpathy

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

Output: No sequence

Example: "standard" classification / regression problems

Input: No sequence

Output: Sequence

Example: Im2Caption

Input: Sequence

Output: No sequence

Example: sentence classification, multiple-choice question answering

Input: Sequence

Output: Sequence

Example: machine translation, video classification, video captioning, open-ended question answering

Image Credit: Andrej Karpathy

What's wrong with MLPs?

- Problem 1: Can't model sequences
  - Fixed-sized Inputs & Outputs
  - No temporal structure

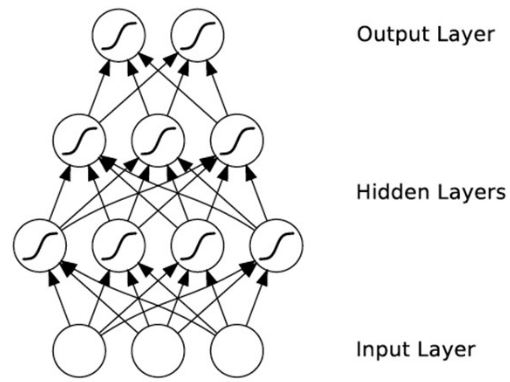


Image Credit: Alex Graves, book

What's wrong with MLPs?

- Problem 1: Can't model sequences
  - Fixed-sized Inputs & Outputs
  - No temporal structure

- Problem 2: Pure feed-forward processing
  - No "memory", no feedback

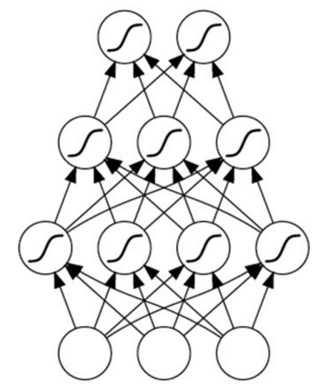


Image Credit: Alex Graves, book

3 Key Ideas

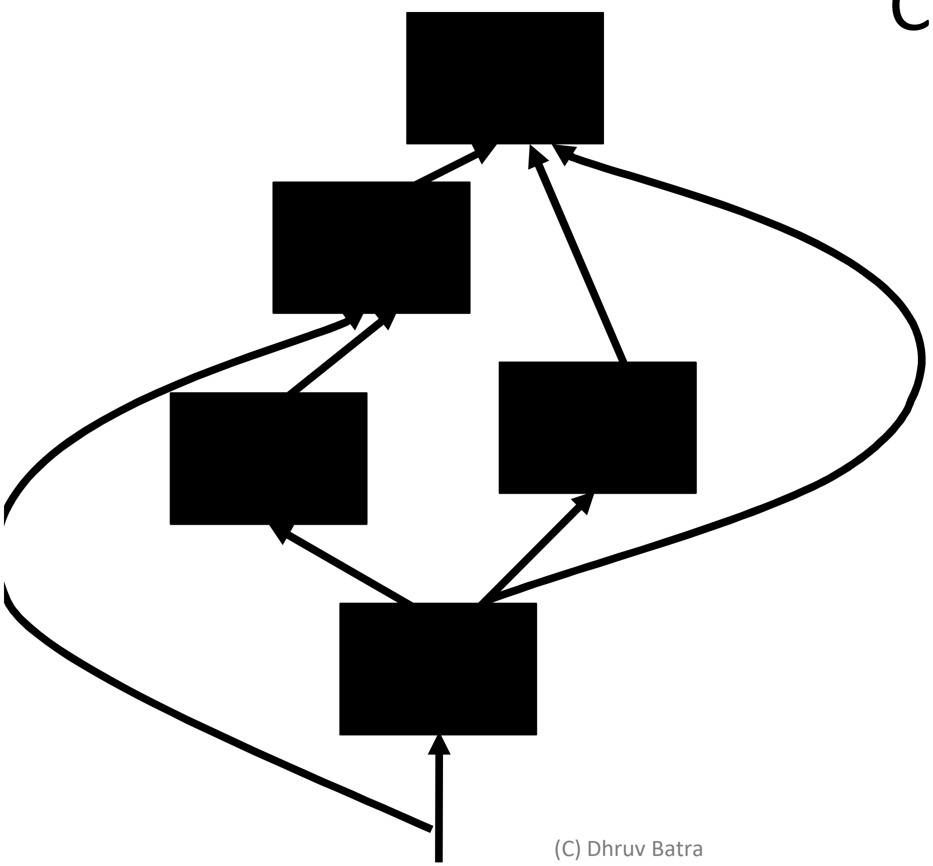
- The notion of memory (state)
  - We want to propagate information across the sequence
  - We will do this with *state*, represented by a vector (embedding/representation)
  - Just as a CNN represents an image with the final hidden vector/bmedding before the final classifier

3 Key Ideas

- The notion of memory (state)

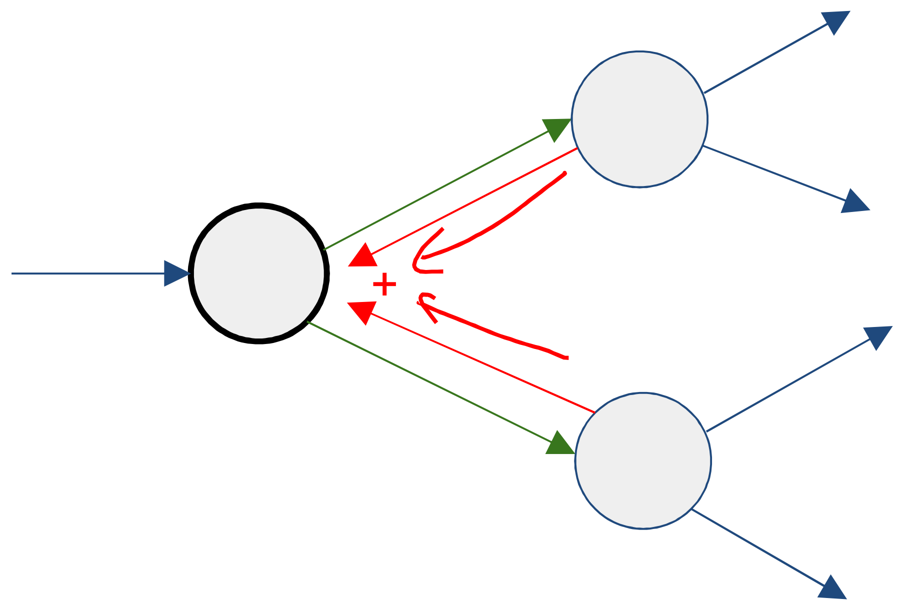
- Parameter Sharing
  – in computation graphs = adding gradients

Computational Graph



Slide Credit: Marc'Aurelio Ranzato

Gradients add at branches



3 Key Ideas

- The notion of memory (state)
- Parameter Sharing
  - in computation graphs = adding gradients
- "Unrolling"
  - in computation graphs with parameter sharing

New Words

- Recurrent Neural Networks (RNNs)

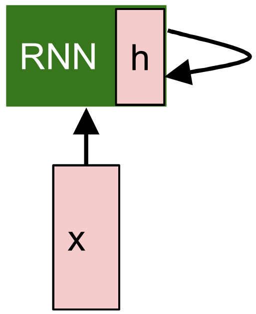
- Recursive Neural Networks
  - General family; think graphs instead of chains

- Types:
  - "Vanilla" RNNs (Elman Networks)
  - Long Short Term Memory (LSTMs)
  - Gated Recurrent Units (GRUs)
  - …

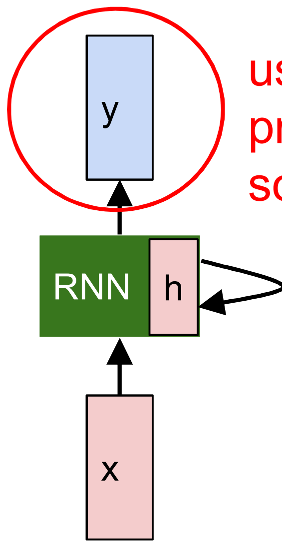
- Algorithms
  - BackProp Through Time (BPTT)
  - BackProp Through Structure (BPTS)

Recurrent Neural Network

- Idea: Input is a **sequence** and we will process it sequentially though a neural network module with *state*
- For each timestep (element of sequence):



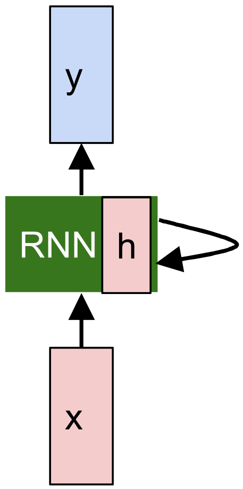
Recurrent Neural Network



usually want to predict a vector at some time steps

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



$$y_t = W_{hy}h_t + b_y$$

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

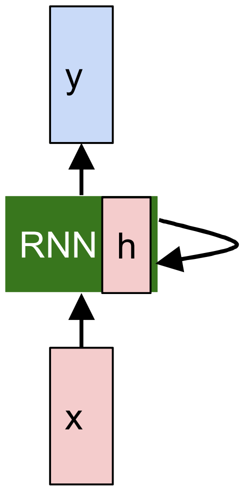
$$\downarrow$$

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

Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



$$y_t = W_{hy} h_t + b_y$$

$$\begin{aligned} h_t &= \tanh(W_{hh} h_{t-1} + W_{xh} x_t) \\ &= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \end{aligned}$$

Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

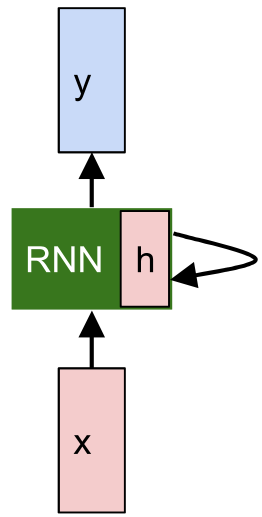
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

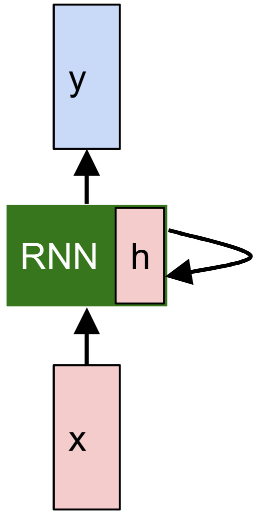


Recurrent Neural Network

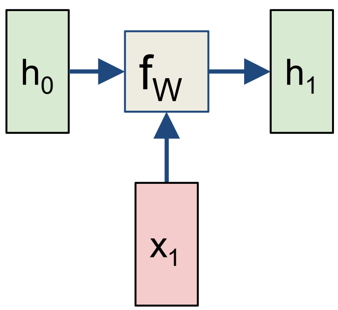
We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

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

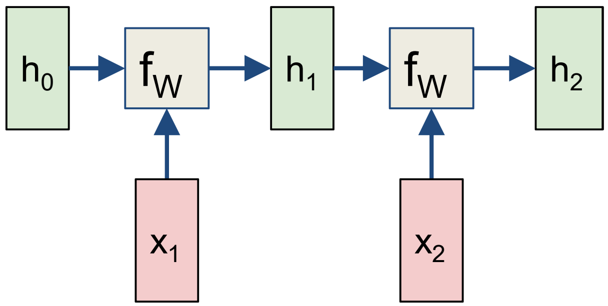
Notice: the same function and the same set of parameters are used at every time step.



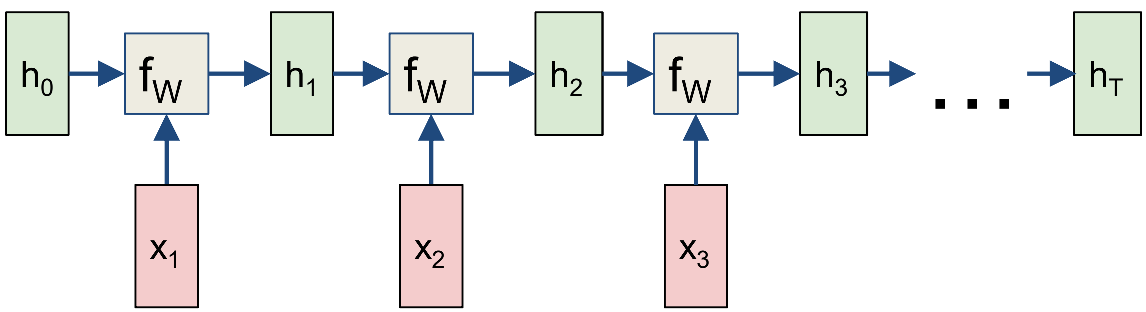
RNN: Computational Graph



RNN: Computational Graph

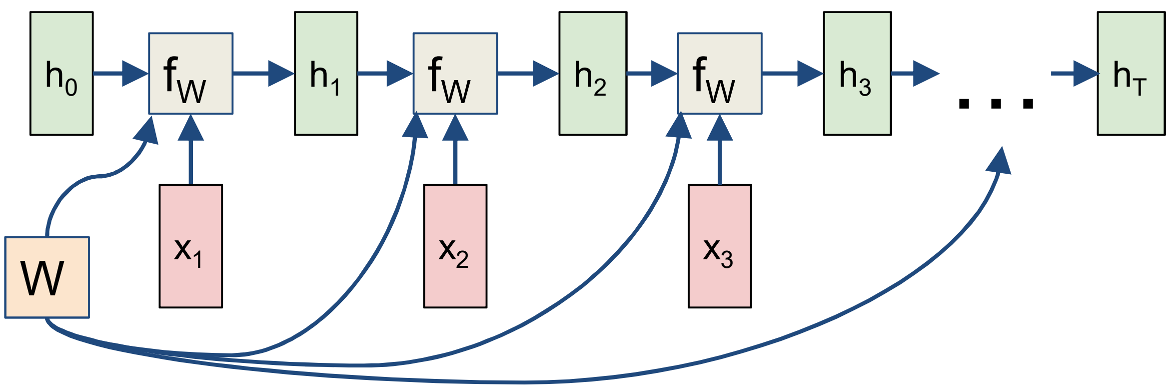


RNN: Computational Graph

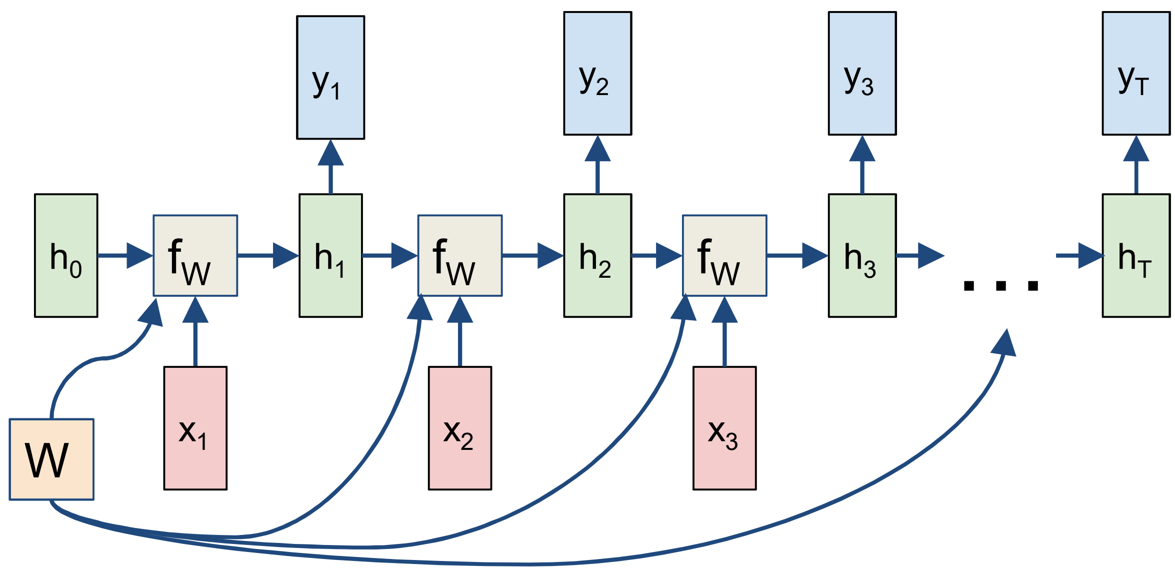


RNN: Computational Graph

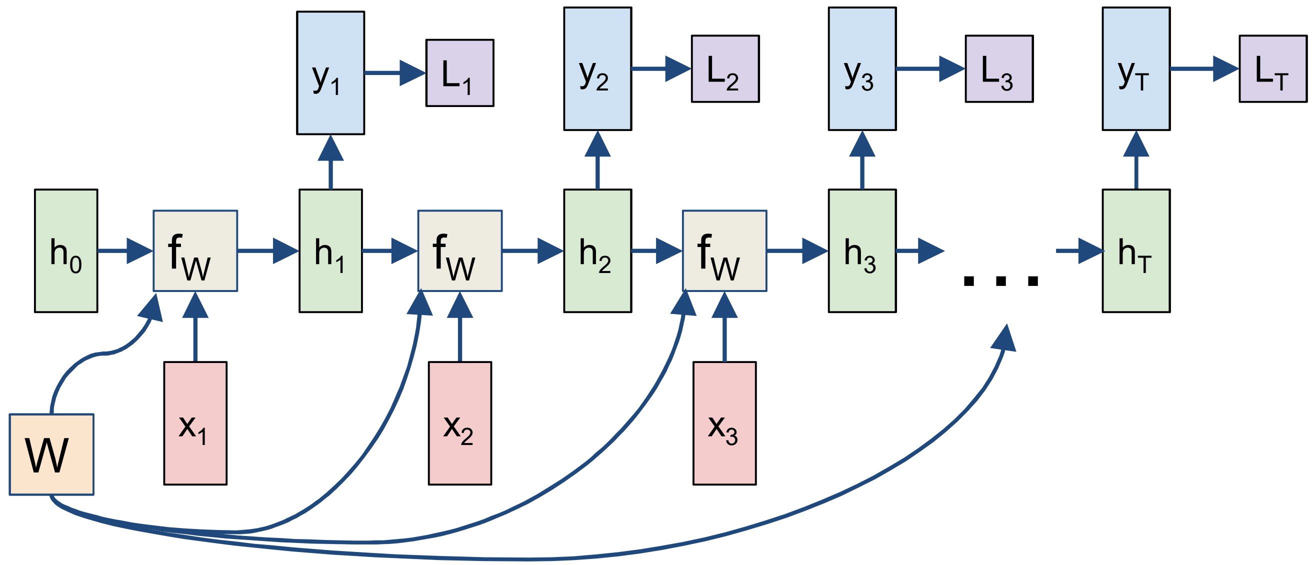
Re-use the same weight matrix at every time-step



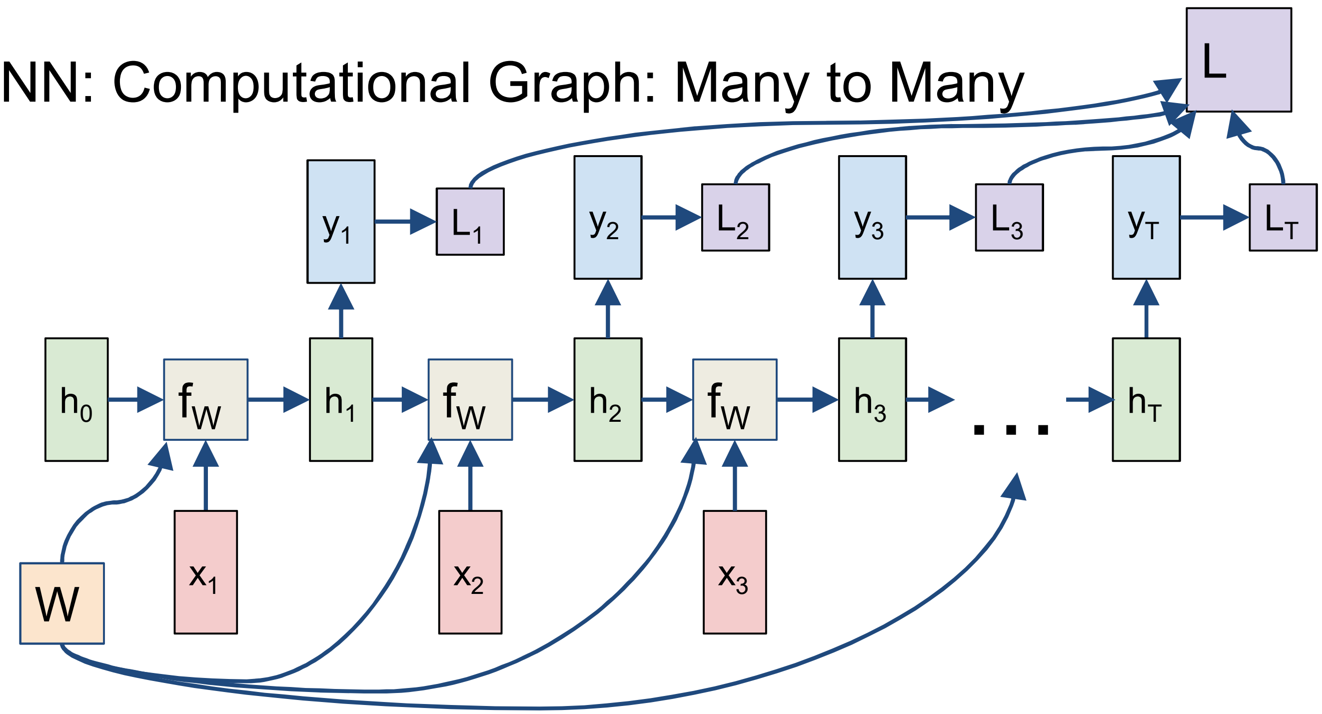
RNN: Computational Graph: Many to Many



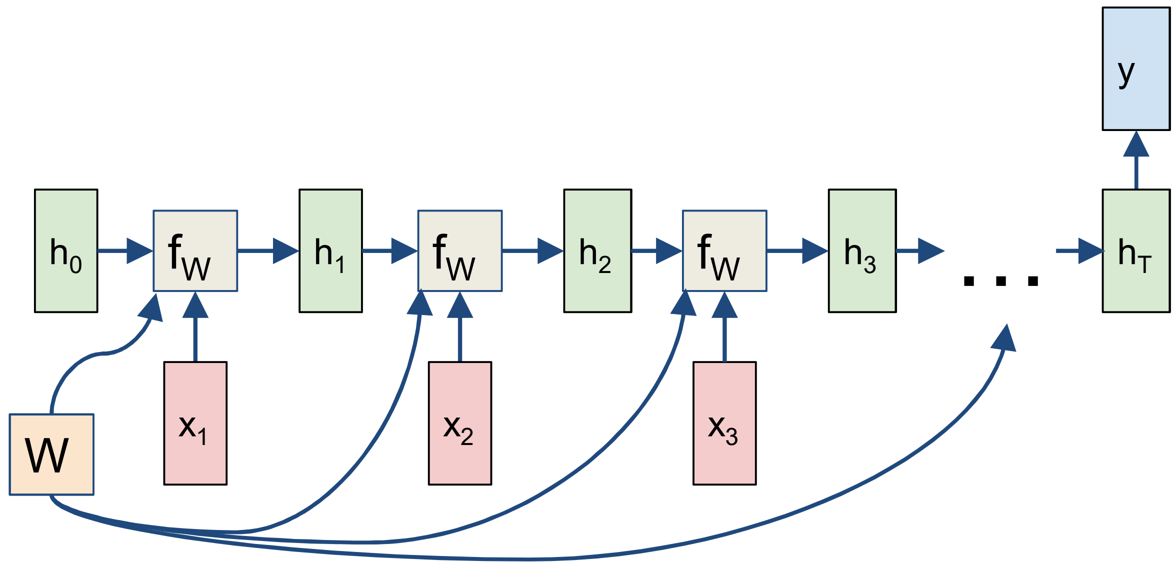
RNN: Computational Graph: Many to Many



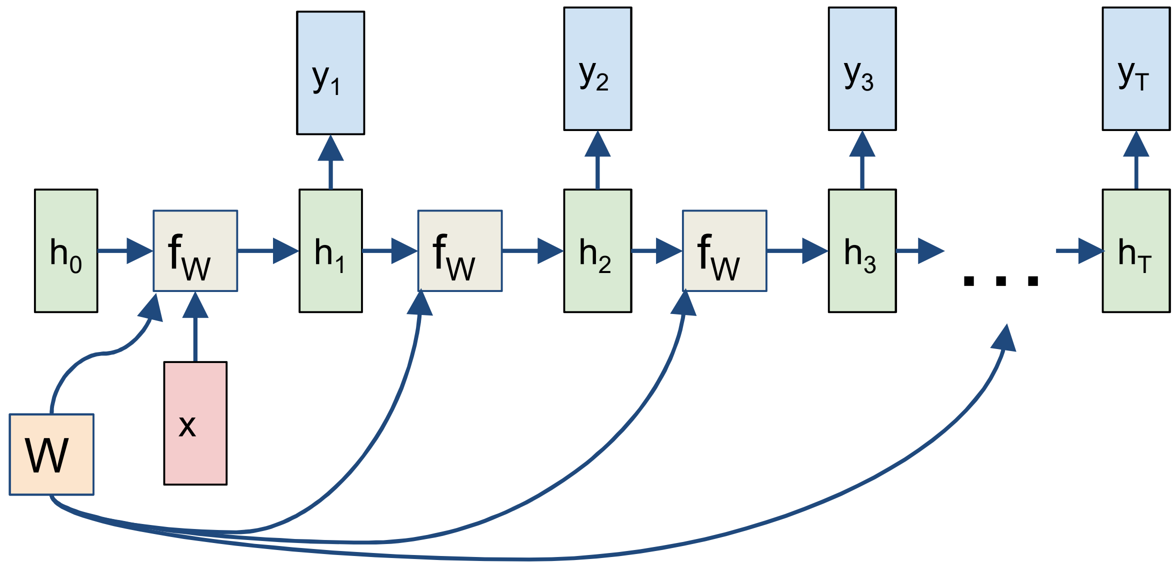
RNN: Computational Graph: Many to Many



RNN: Computational Graph: Many to One

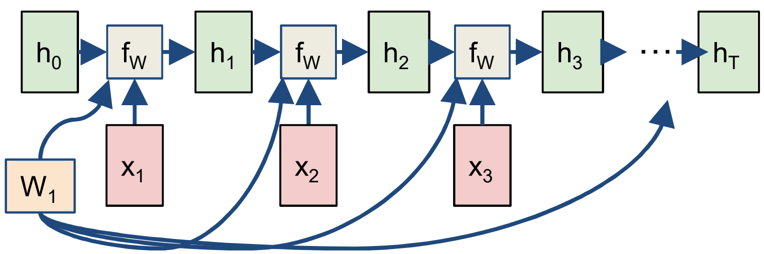


RNN: Computational Graph: One to Many

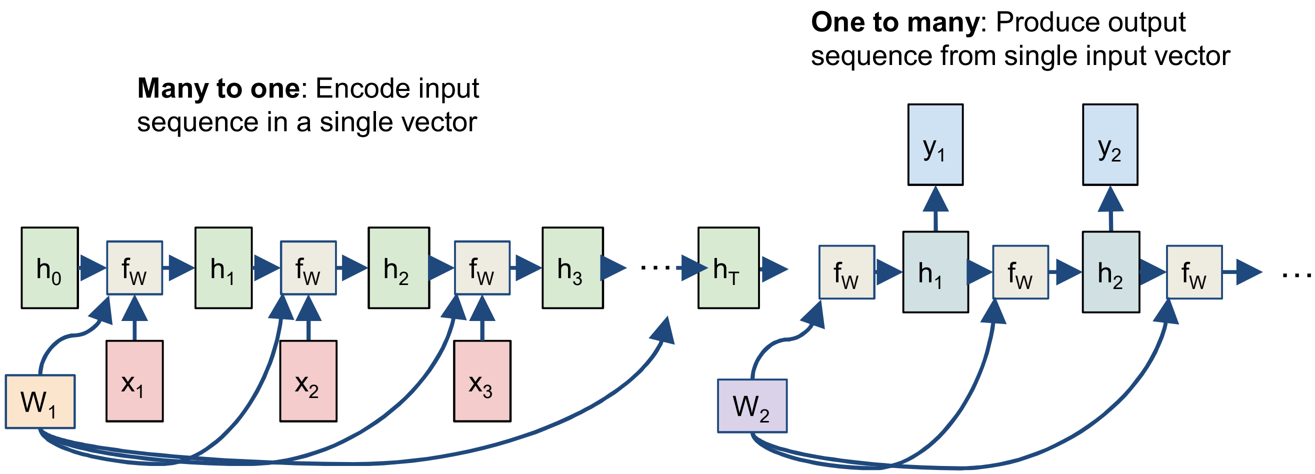


Sequence to Sequence: Many-to-one + one-to-many

Many to one: Encode input sequence in a single vector



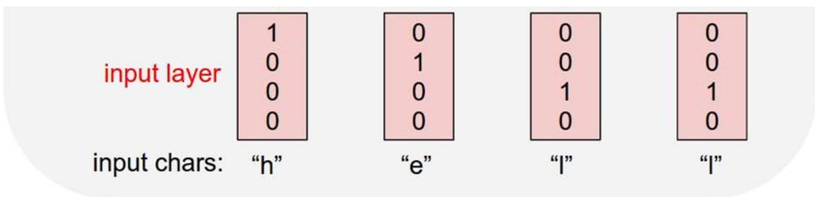
Sequence to Sequence: Many-to-one + one-to-many



Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

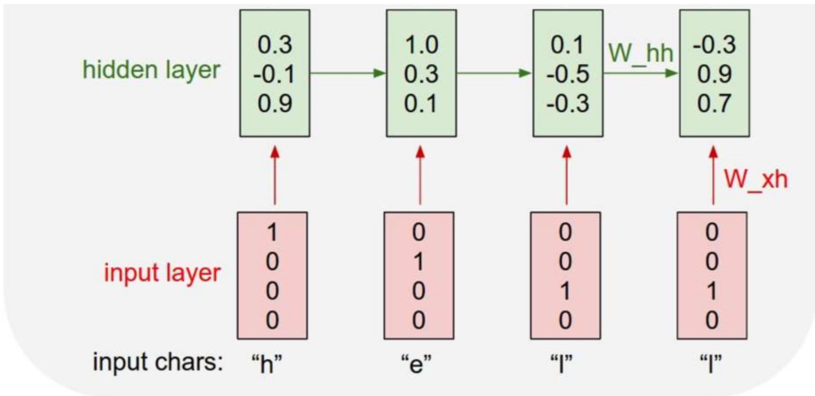


Example: Character-level Language Model

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

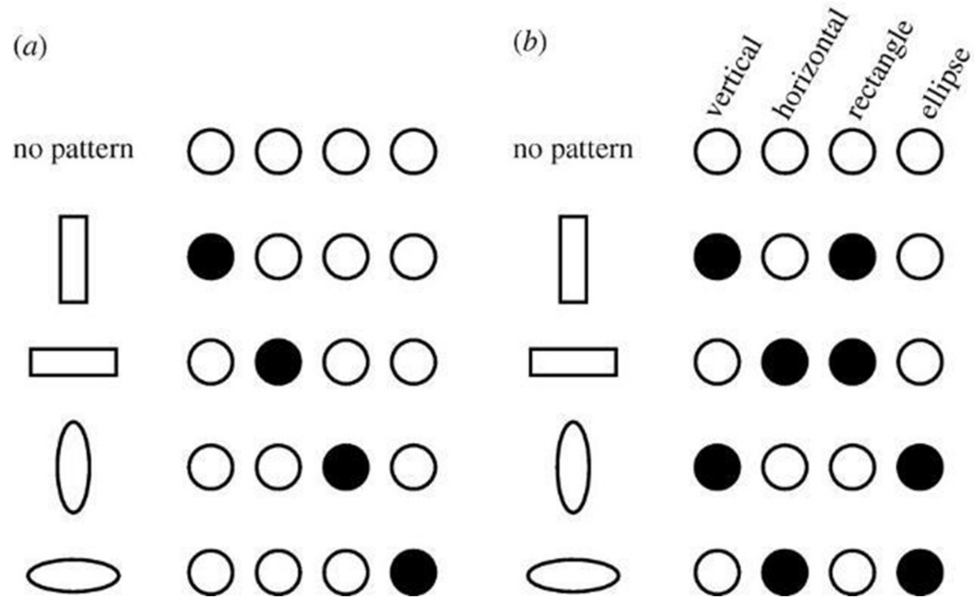
Vocabulary: [h,e,l,o]

Example training sequence: **"hello"**



Distributed Representations Toy Example

- Can we interpret each dimension?



Power of distributed representations!

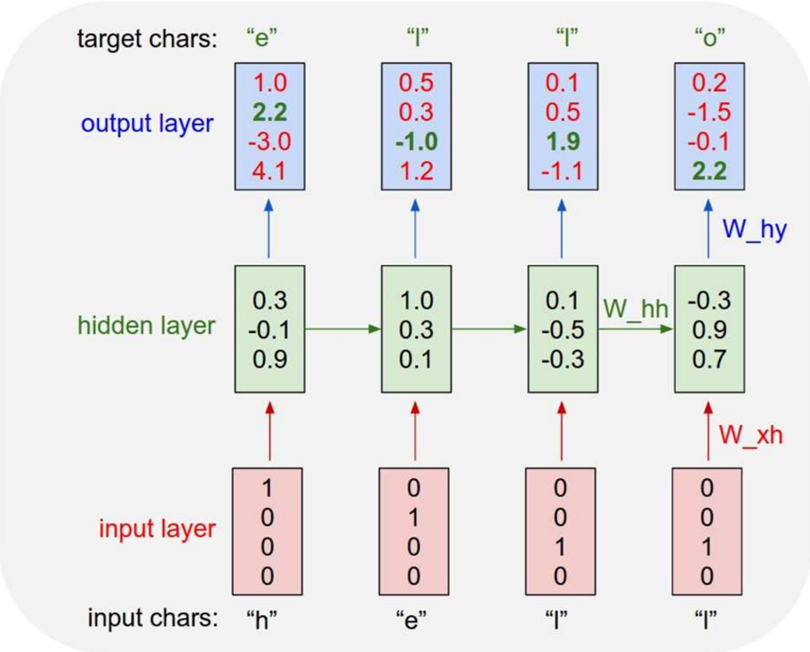
Local ● ● ○ ● = VR + HR + HE = ?

Distributed ● ● ○ ● = V + H + E $\approx$ ◯

Example: Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**

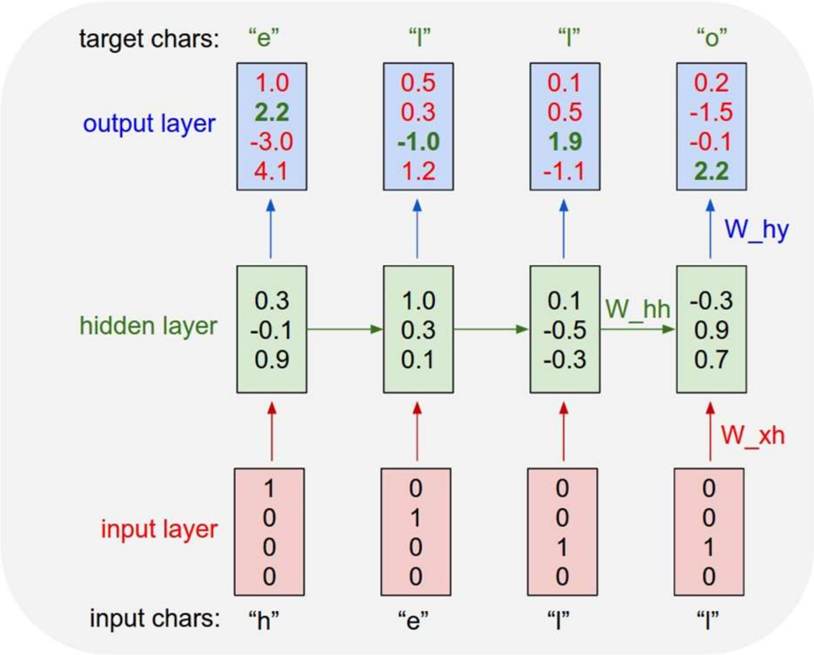


Training Time: MLE / “Teacher Forcing”

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

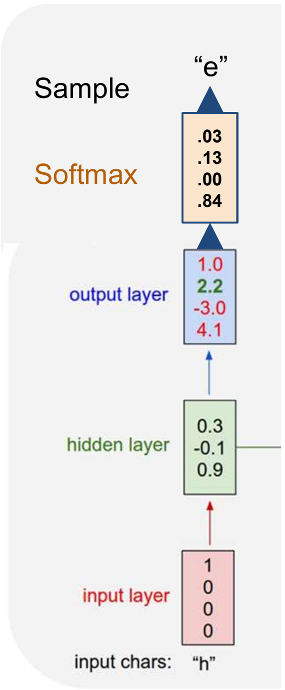


Test Time: Sample / Argmax / Beam Search

Example:
Character-level
Language Model
Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model

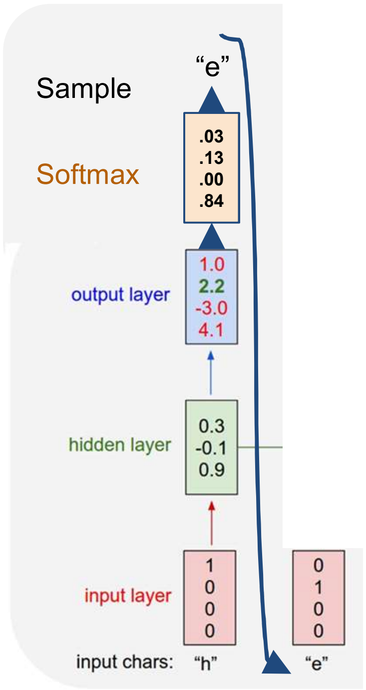


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

At test-time sample characters one at a time, feed back to model

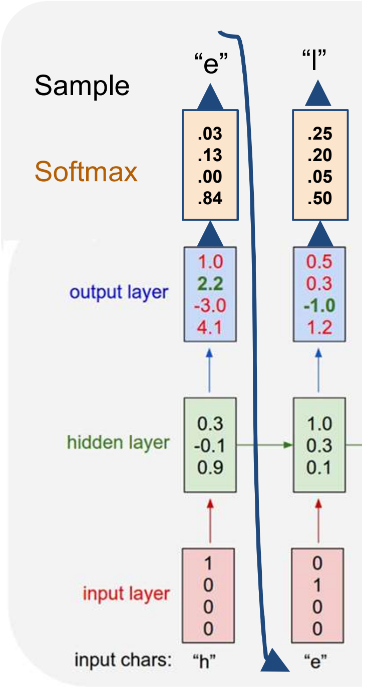


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

At test-time sample characters one at a time, feed back to model

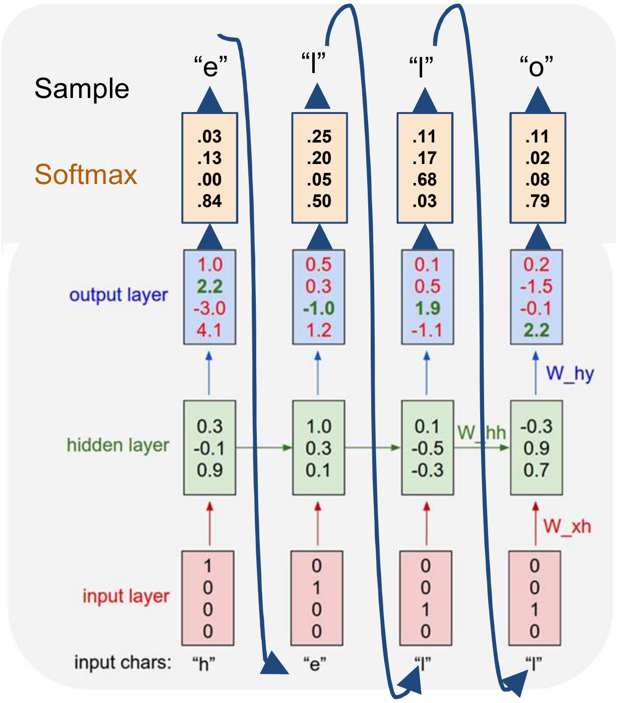


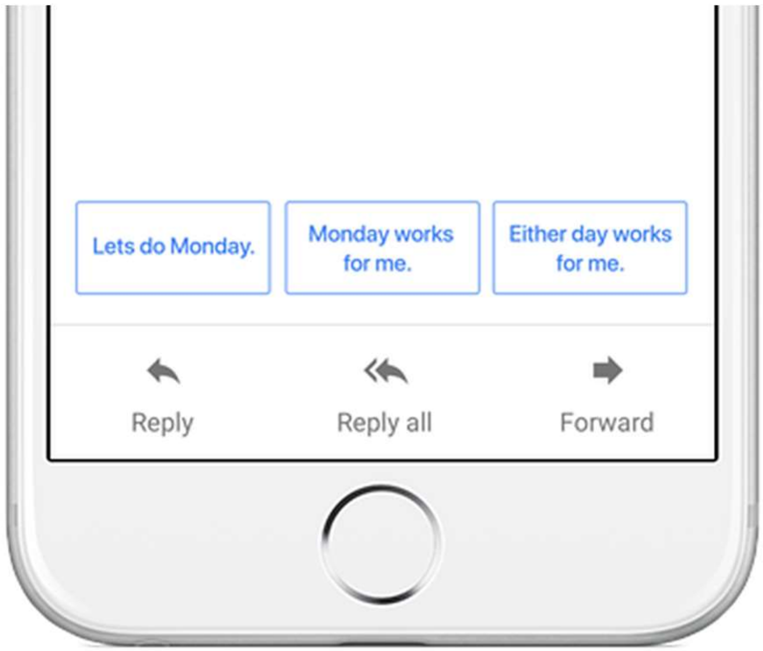
Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

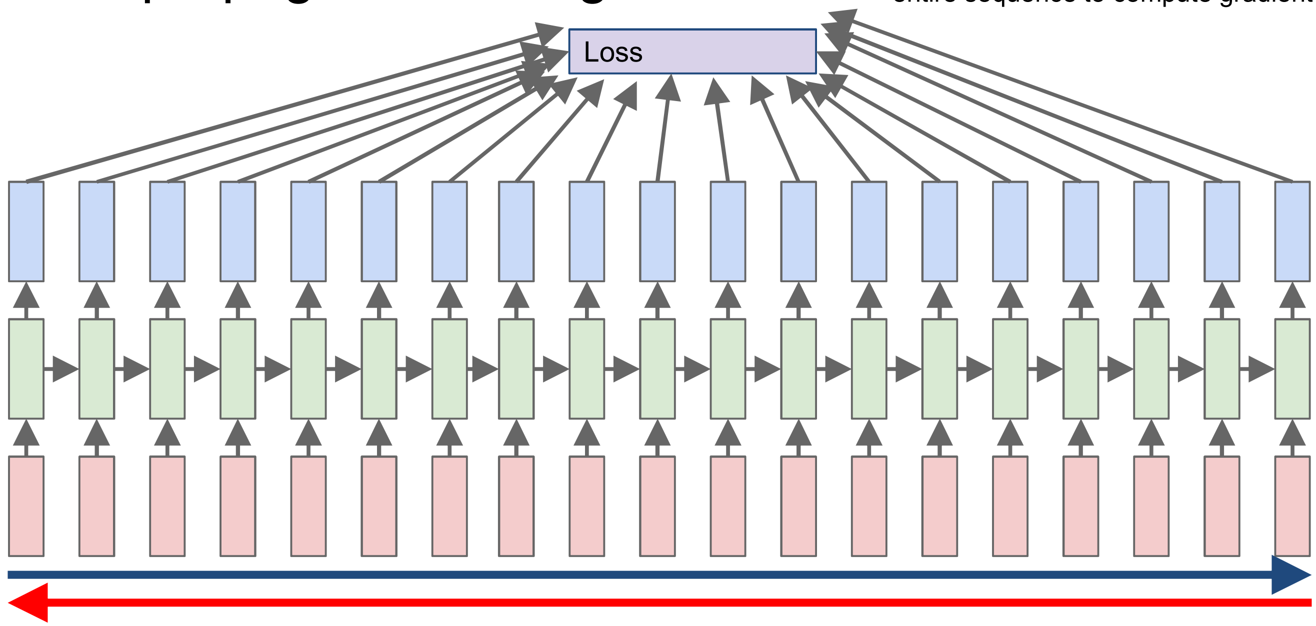
At test-time sample characters one at a time, feed back to model



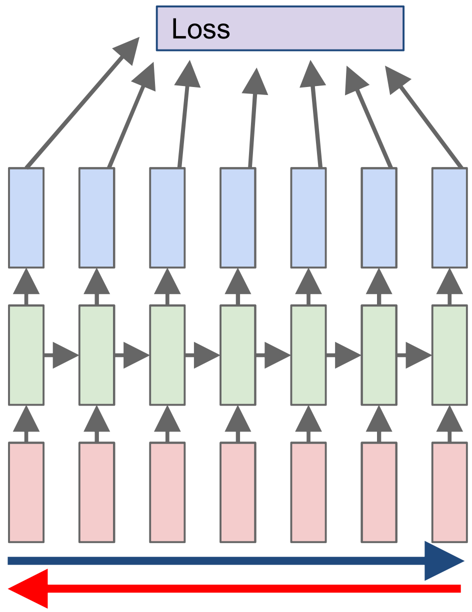

Lets do Monday.
Monday works for me.
Either day works for me.
Reply
Reply all
Forward

Backpropagation through time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient

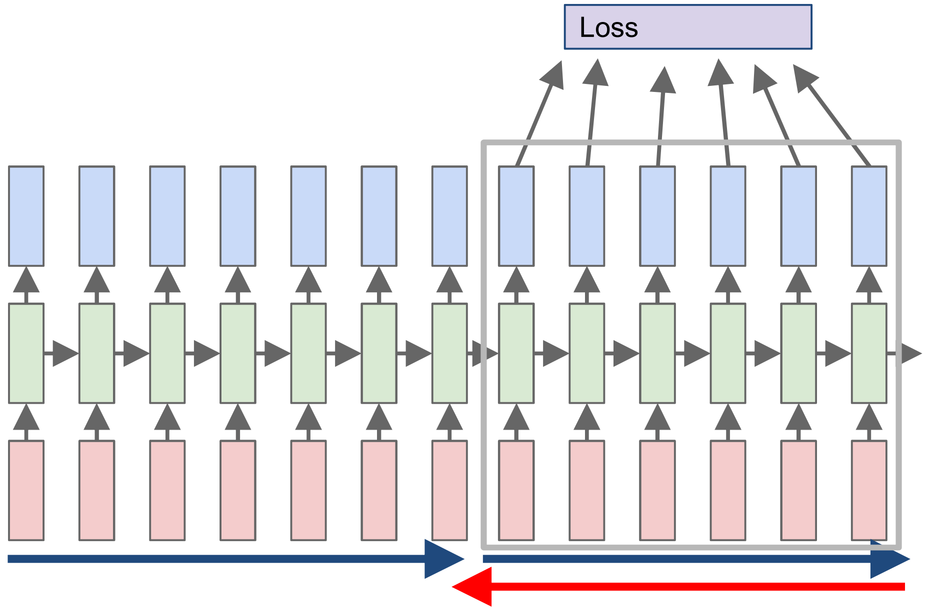


Truncated Backpropagation through time



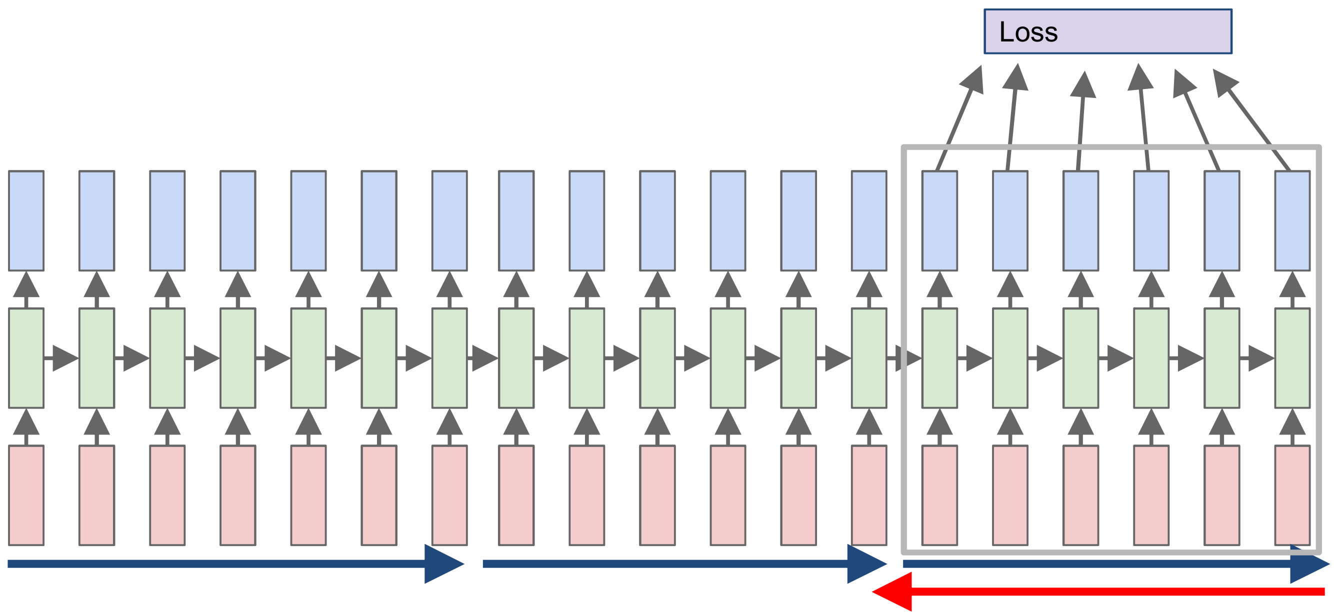
Run forward and backward through chunks of the sequence instead of whole sequence

Truncated Backpropagation through time



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Truncated Backpropagation through time

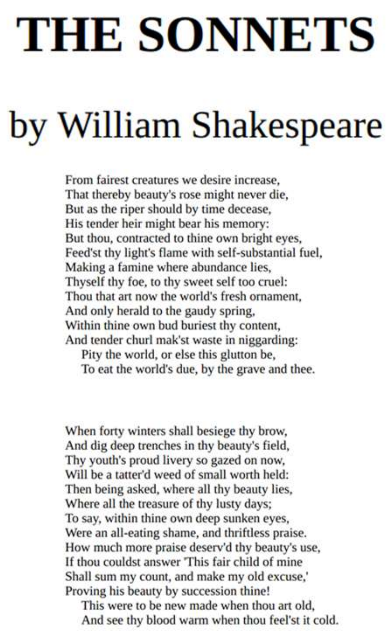


min-char-rnn.py gist: 112 lines of Python

```python
"""
Minimal character-level Vanilla RNN model. Written by Andrej Karpathy (@karpathy)
BSD License
"""
import numpy as np

# data I/O
data = open('input.txt', 'r').read() # should be simple plain text file
chars = list(set(data))
data_size, vocab_size = len(data), len(chars)
print 'data has %d characters, %d unique.' % (data_size, vocab_size)
char_to_ix = { ch:i for i,ch in enumerate(chars) }
ix_to_char = { i:ch for i,ch in enumerate(chars) }

# hyperparameters
hidden_size = 100 # size of hidden layer of neurons
seq_length = 25 # number of steps to unroll the RNN for
learning_rate = 1e-1

# model parameters
Wxh = np.random.randn(hidden_size, vocab_size)*0.01 # input to hidden
Whh = np.random.randn(hidden_size, hidden_size)*0.01 # hidden to hidden
Why = np.random.randn(vocab_size, hidden_size)*0.01 # hidden to output
bh = np.zeros((hidden_size, 1)) # hidden bias
by = np.zeros((vocab_size, 1)) # output bias

def lossFun(inputs, targets, hprev):
  """
  inputs,targets are both list of integers.
  hprev is Hx1 array of initial hidden state
  returns the loss, gradients on model parameters, and last hidden state
  """
  xs, hs, ys, ps = {}, {}, {}, {}
  hs[-1] = np.copy(hprev)
  loss = 0
  # forward pass
  for t in xrange(len(inputs)):
    xs[t] = np.zeros((vocab_size,1)) # encode in 1-of-k representation
    xs[t][inputs[t]] = 1
    hs[t] = np.tanh(np.dot(Wxh, xs[t]) + np.dot(Whh, hs[t-1]) + bh) # hidden state
    ys[t] = np.dot(Why, hs[t]) + by # unnormalized log probabilities for next chars
    ps[t] = np.exp(ys[t]) / np.sum(np.exp(ys[t])) # probabilities for next chars
    loss += -np.log(ps[t][targets[t],0]) # softmax (cross-entropy loss)
  # backward pass: compute gradients going backwards
  dWxh, dWhh, dWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
  dbh, dby = np.zeros_like(bh), np.zeros_like(by)
  dhnext = np.zeros_like(hs[0])
  for t in reversed(xrange(len(inputs))):
    dy = np.copy(ps[t])
    dy[targets[t]] -= 1 # backprop into y
    dWhy += np.dot(dy, hs[t].T)
    dby += dy
    dh = np.dot(Why.T, dy) + dhnext # backprop into h
    dhraw = (1 - hs[t] * hs[t]) * dh # backprop through tanh nonlinearity
    dbh += dhraw
    dWxh += np.dot(dhraw, xs[t].T)
    dWhh += np.dot(dhraw, hs[t-1].T)
    dhnext = np.dot(Whh.T, dhraw)
  for dparam in [dWxh, dWhh, dWhy, dbh, dby]:
    np.clip(dparam, -5, 5, out=dparam) # clip to mitigate exploding gradients
  return loss, dWxh, dWhh, dWhy, dbh, dby, hs[len(inputs)-1]
```

```python
def sample(h, seed_ix, n):
  """
  sample a sequence of integers from the model
  h is memory state, seed_ix is seed letter for first time step
  """
  x = np.zeros((vocab_size, 1))
  x[seed_ix] = 1
  ixes = []
  for t in xrange(n):
    h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h) + bh)
    y = np.dot(Why, h) + by
    p = np.exp(y) / np.sum(np.exp(y))
    ix = np.random.choice(range(vocab_size), p=p.ravel())
    x = np.zeros((vocab_size, 1))
    x[ix] = 1
    ixes.append(ix)
  return ixes

n, p = 0, 0
mWxh, mWhh, mWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
mbh, mby = np.zeros_like(bh), np.zeros_like(by) # memory variables for Adagrad
smooth_loss = -np.log(1.0/vocab_size)*seq_length # loss at iteration 0
while True:
  # prepare inputs (we're sweeping from left to right in steps seq_length long)
  if p+seq_length+1 >= len(data) or n == 0:
    hprev = np.zeros((hidden_size,1)) # reset RNN memory
    p = 0 # go from start of data
  inputs = [char_to_ix[ch] for ch in data[p:p+seq_length]]
  targets = [char_to_ix[ch] for ch in data[p+1:p+seq_length+1]]

  # sample from the model now and then
  if n % 100 == 0:
    sample_ix = sample(hprev, inputs[0], 200)
    txt = ''.join(ix_to_char[ix] for ix in sample_ix)
    print '----\n %s \n----' % (txt, )

  # forward seq_length characters through the net and fetch gradient
  loss, dWxh, dWhh, dWhy, dbh, dby, hprev = lossFun(inputs, targets, hprev)
  smooth_loss = smooth_loss * 0.999 + loss * 0.001
  if n % 100 == 0: print 'iter %d, loss: %f' % (n, smooth_loss) # print progress

  # perform parameter update with Adagrad
  for param, dparam, mem in zip([Wxh, Whh, Why, bh, by],
                                [dWxh, dWhh, dWhy, dbh, dby],
                                [mWxh, mWhh, mWhy, mbh, mby]):
    mem += dparam * dparam
    param += -learning_rate * dparam / np.sqrt(mem + 1e-8) # adagrad update

  p += seq_length # move data pointer
  n += 1 # iteration counter
```

(https://gist.github.com/karpathy/d4dee566867f8291f086)

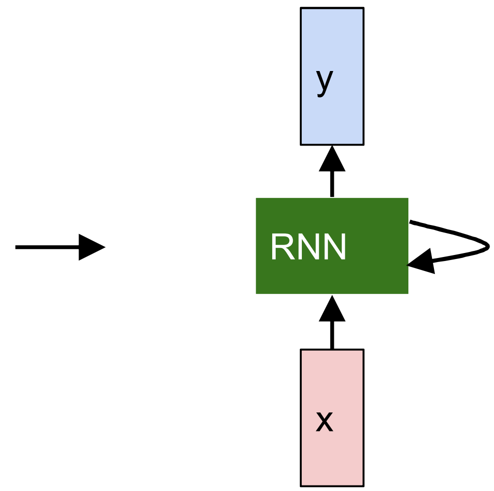


THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.



at first:

```
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
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

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

```
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.
```

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```