Topics:

- Calibration (Fairness/Bias)
- Recurrent Neural Networks

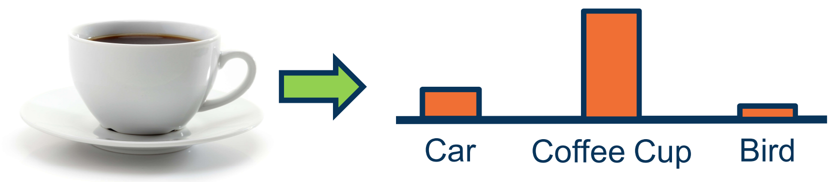
CS 4803-DL / 7643-A
ZSOLT KIRA

- **Assignment 3 out**
  - Due **March 14th 11:59pm EST.**

- **Projects**
  - Released assignments; please **reach out** to your groups to discuss team formation
    - Note: Some may have already found groups, etc. Note that it doesn't **have** to be 4 members so you can go with smaller. You can also converge on high-level topic and then reach out on piazza looking for members.
  - Rubric/description, project proposal instructions, FB projects released
  - Project proposal due **March 22nd**

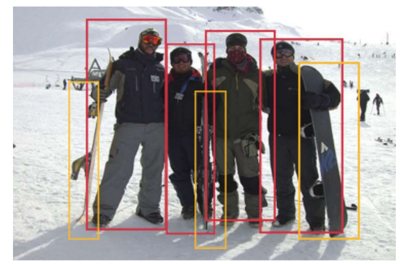
Here is an FAQ/guide for questions I've received in the past:

- **What is this about? I already have a team and didn't fill out catme (or get a catme assignment email); what do I do?** Nothing! Just submit the project proposal on time :) The catme assignments are only for those that DIDN'T have a team and filled out the survey.
- **I got assigned to a catme group but I already have a team:** Let the team members know ASAP so that they can plan accordingly!
- **All my catme teammates already found other teams, so now I have no team:** Try to reach out to existing teams that are looking for members. This can be on piazza (new posts or @5) or the project proposals on Canvas (they are all visible to everyone, and have a field indicating if they are looking for new members).
- **We have a team but are looking for additional members:** Post on piazza that you're looking (along with potential topics you're interested in). If you have a project planned already, post it on the Canvas project proposal assignment so that others can see, and indicate you're looking for additional members.
- **I didn't fill out catme but don't have a team:** See #3 above.
- **I requested removal from catme but still got assigned; do I have to join this new team?** No. Sorry about that, I received lots of these requests across many different communication channels, so may have missed some. See #2 above for what you should do.



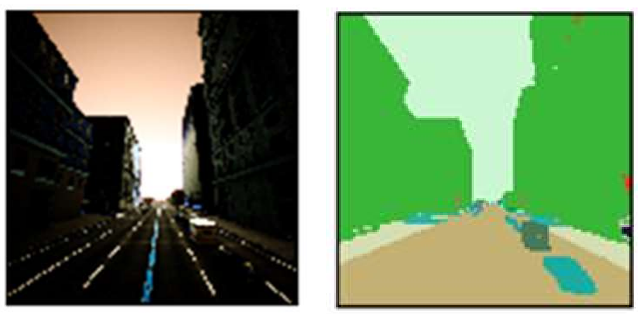
Classification

(Class distribution per image)



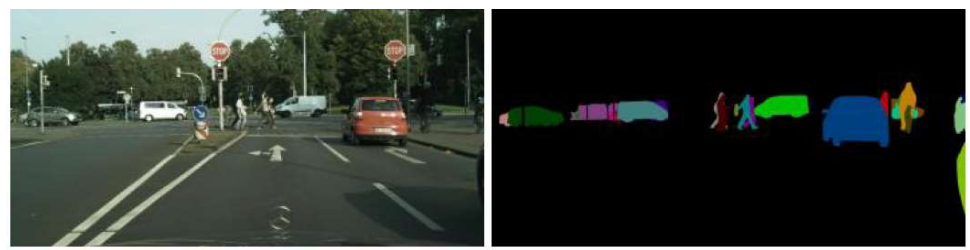
Object Detection

(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

Georgia Tech

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

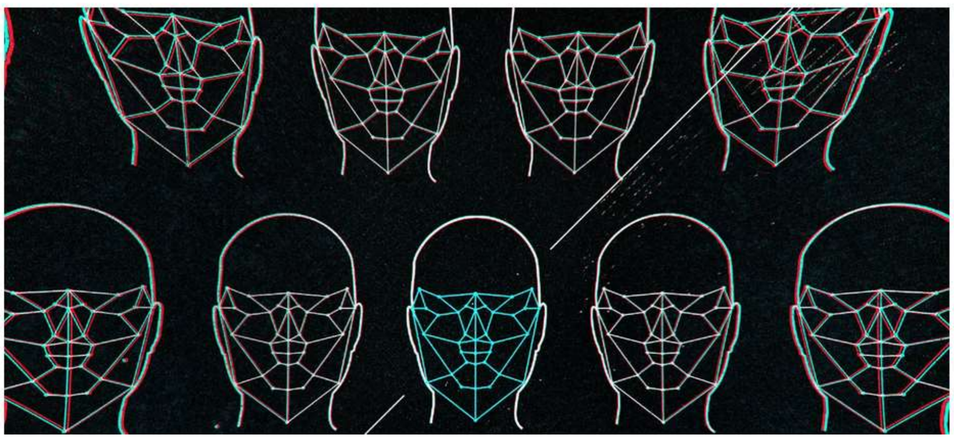
17

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

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 and the Fairness Impossibility Theorems

Calibration

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

FACEBOOK AI Georgia Tech

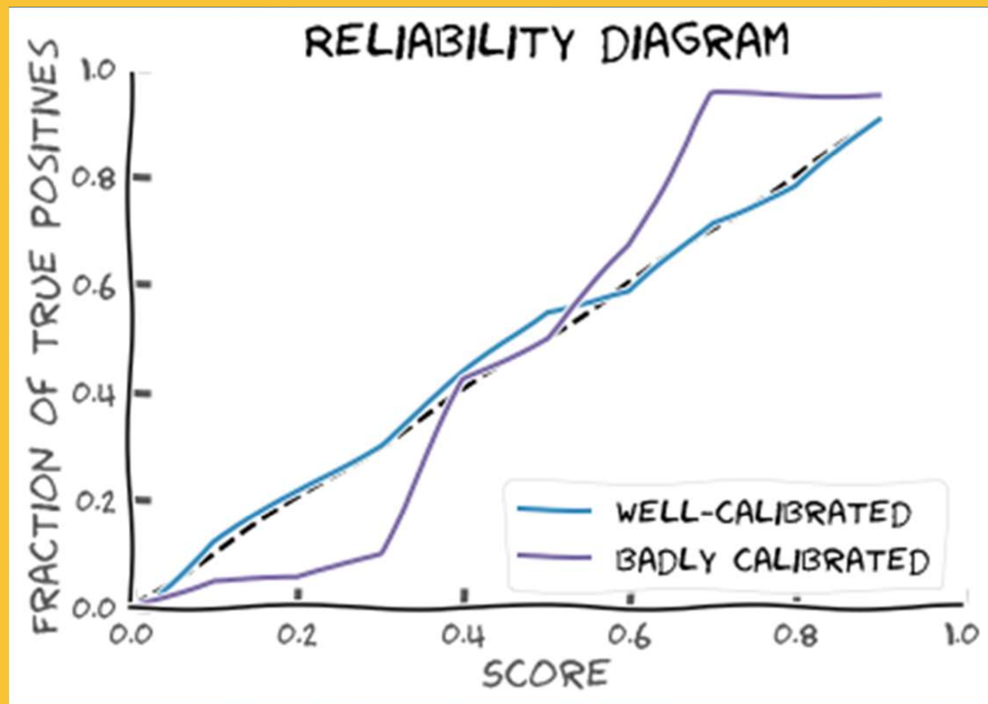
A classifier is **well-calibrated** if the probability of the observations with a given probability score of having a label is equal to the proportion of observations having that label

- **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 $\hat{Y}$ is the predicted label and $\hat{P}$ is the predicted probability (or score) for class Y

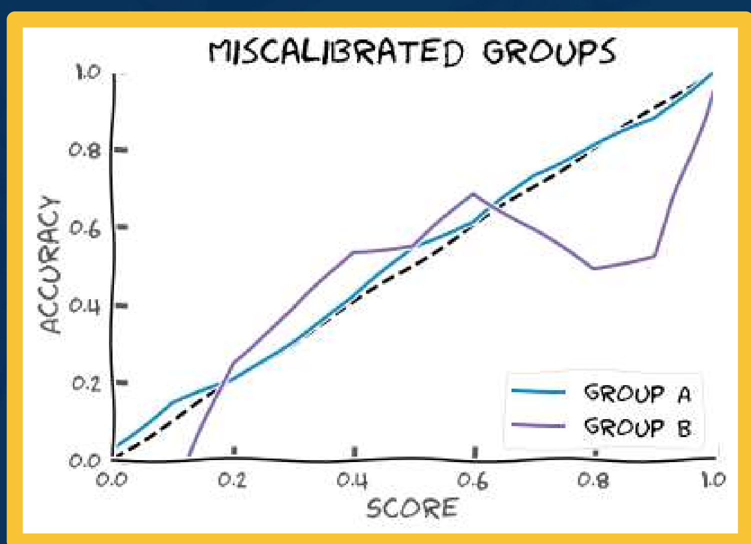
Calibration: Definition

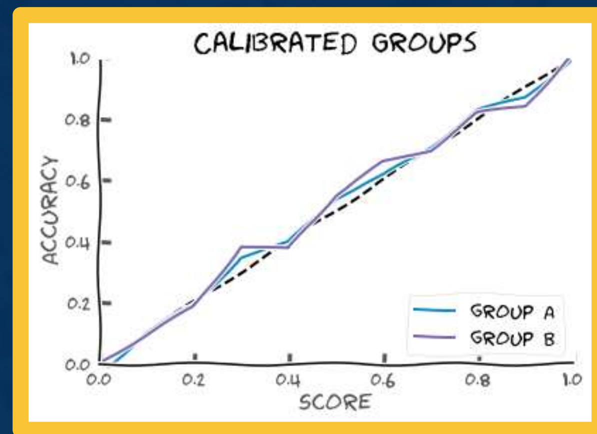


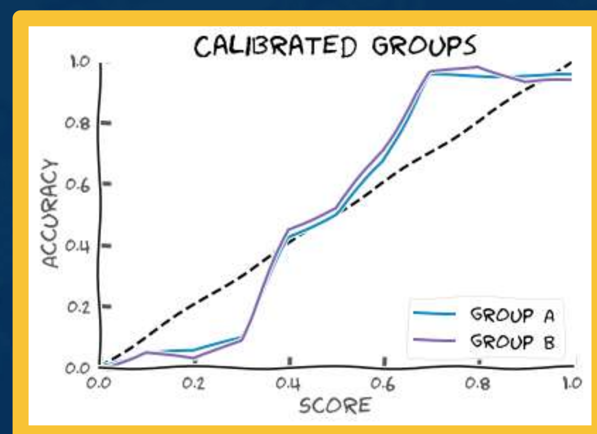
FACEBOOK AI Georgia Tech

Calibration: Definition

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







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

Platt scaling (binary classifier)

- Learn parameters $\boldsymbol{a}, \boldsymbol{b}$ so that the **calibrated probability** is $\hat{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{\sigma}(\boldsymbol{a}\boldsymbol{z}_{\boldsymbol{i}} + \boldsymbol{b})$)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 $\hat{\boldsymbol{q}}_{\boldsymbol{i}} = \max_{\boldsymbol{k}} \boldsymbol{\sigma}_{\boldsymbol{SoftMax}}(\boldsymbol{z}_{\boldsymbol{i}}/\boldsymbol{T})$

Calibration: Limitations

- Group based
- The Inherent Tradeoffs of Calibration

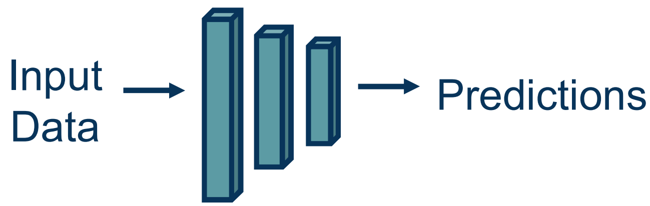
It is impossible for a classifier to achieve both equal calibration and error rates between groups, (if there is a difference in prevalence between the groups and the classifier is not perfect)

Kleinberg, Jon, Sendhil Mullainathan, and Manish Raghavan. "Inherent trade-offs in the fair determination of risk scores." arXiv preprint arXiv:1609.05807 (2016).

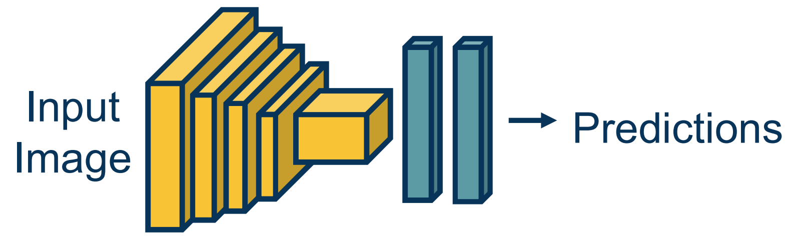
Chouldechova, Alexandra. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments." Big data 5, no. 2 (2017): 153-163.

Module 3 Introduction

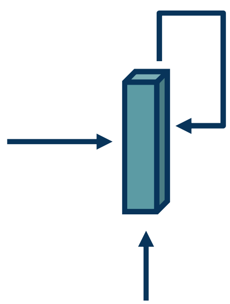
The Space of Architectures



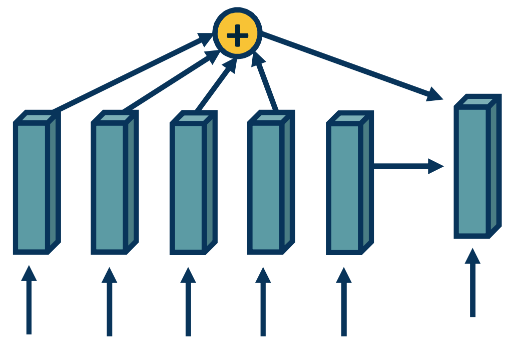
Fully Connected Neural Networks



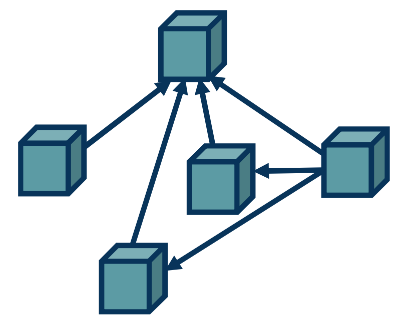
Convolutional Neural Networks



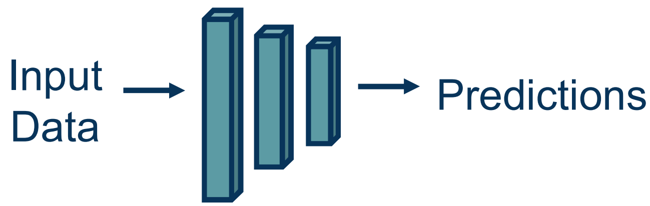
Recurrent Neural Networks



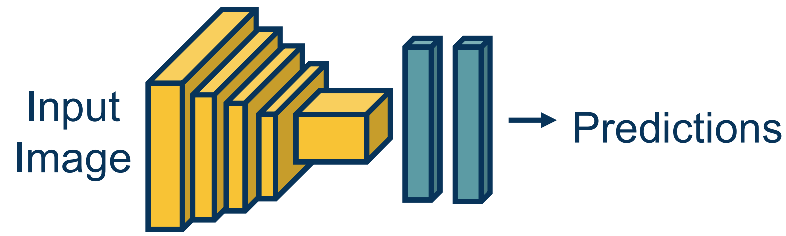
Attention-Based Networks



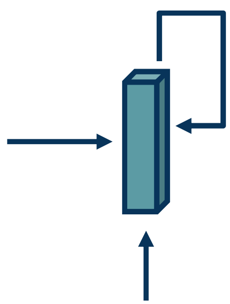
Graph-Based Networks



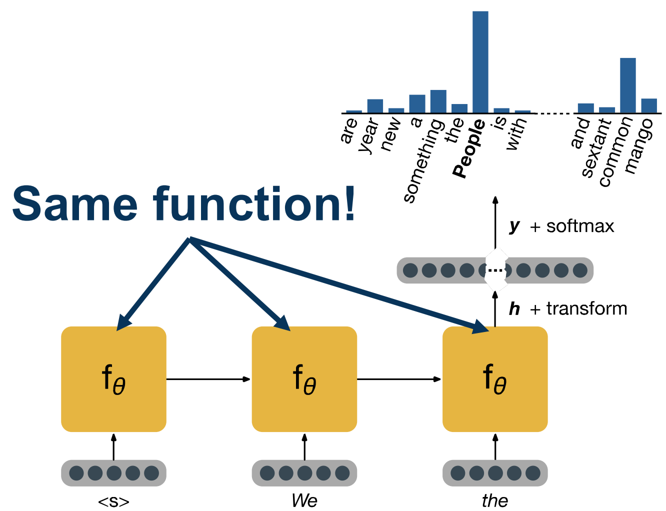
Fully Connected Neural Networks



Convolutional Neural Networks

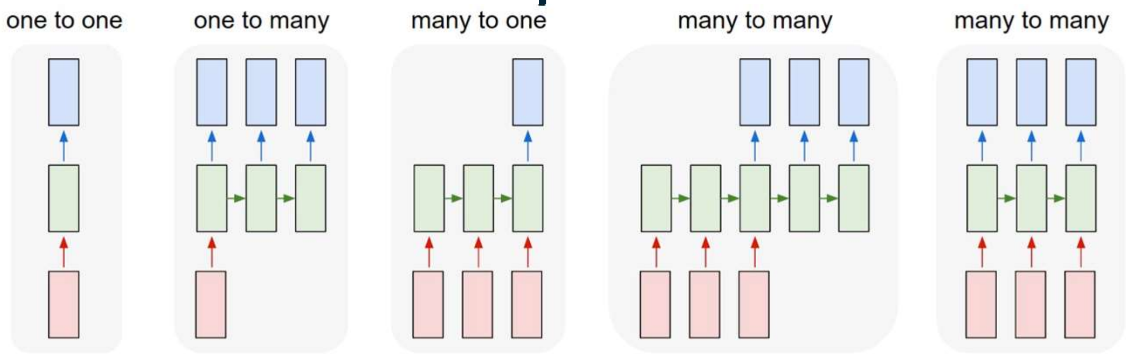


Recurrent Neural Networks



Recurrent Neural Networks

New Topic: RNNs



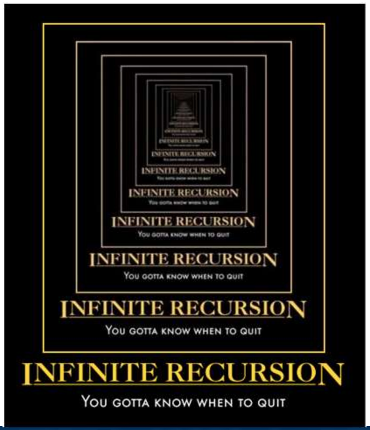
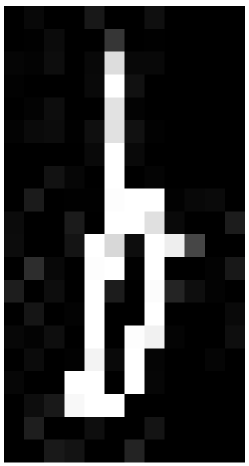


Image Credit: Andrei Karpathy

Why model sequences?



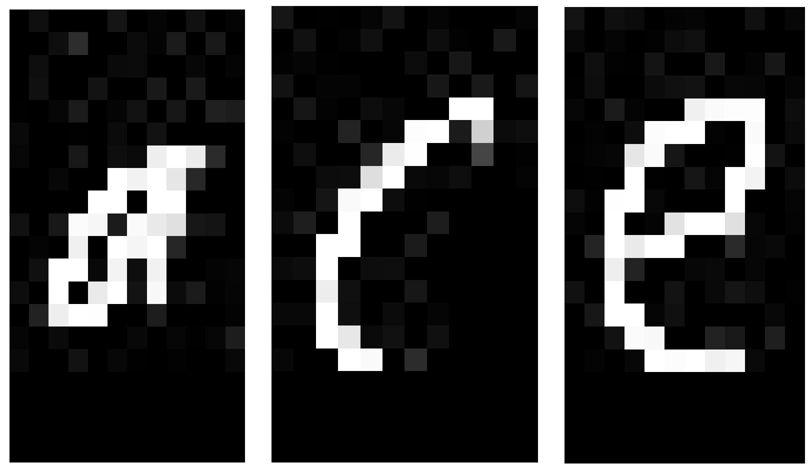


Figure Credit: Carlos Guestrin

Why model sequences?

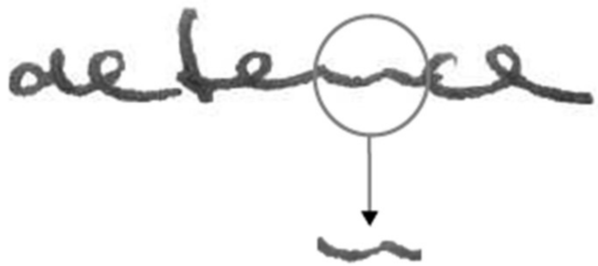
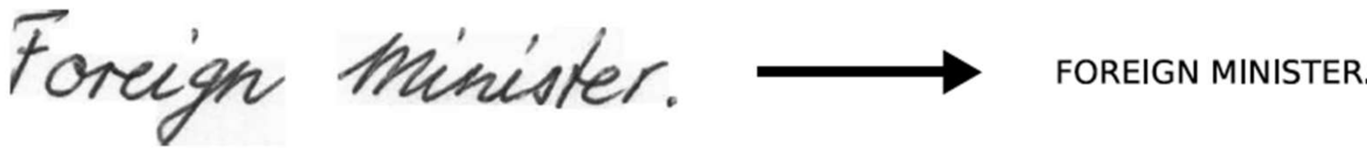


Image Credit: Alex Graves

Sequences are everywhere…



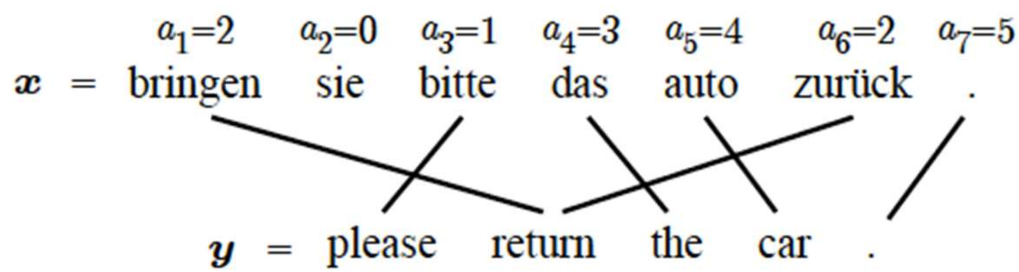
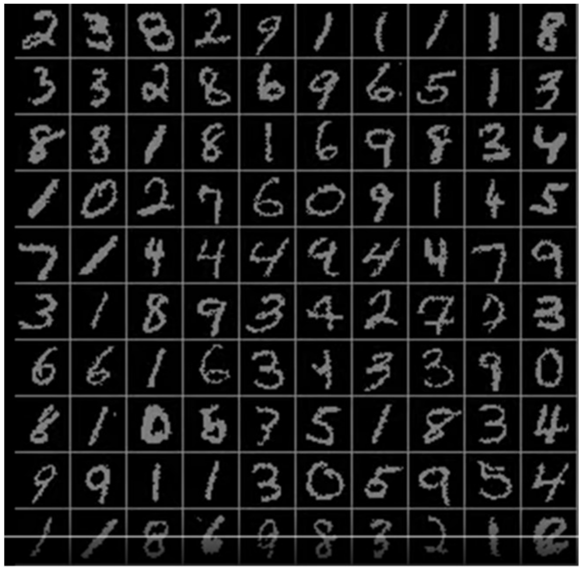


Image Credit: Alex Graves and Kevin Gimpel

Even where you might not expect a sequence…

Classify images by taking a series of "glimpses"



Ba, Mnih, and Kavukcuoglu, "Multiple Object Recognition with Visual Attention", ICLR 2015.
Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015
Figure copyright Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra, 2015. Reproduced with permission.

Even where you might not expect a sequence…

- Output ordering = sequence

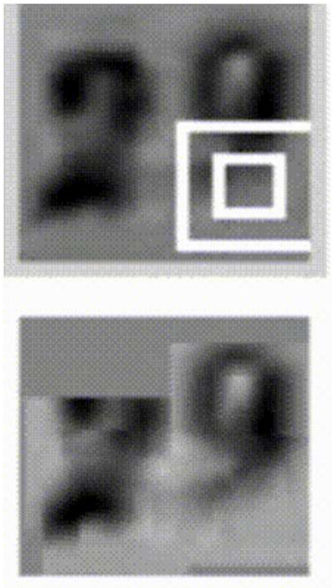
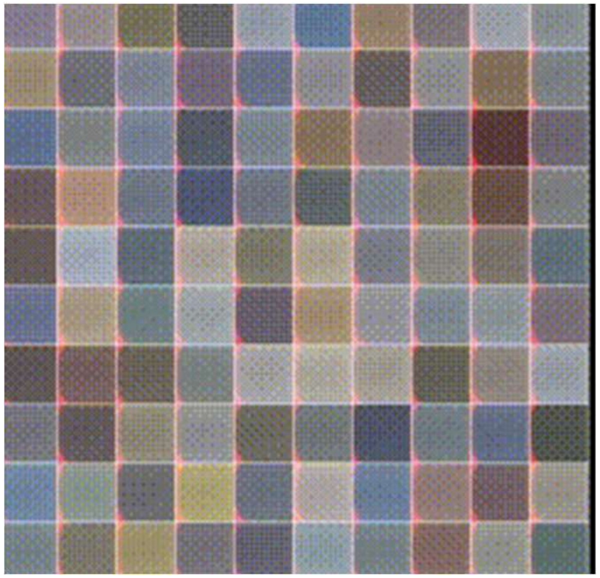
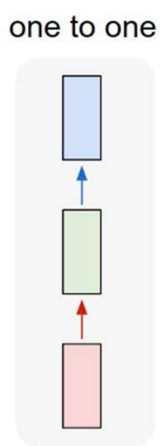



Image Credit: Ba et al.; Gregor et al

Sequences in Input or Output?

- It's a spectrum…



Input: No sequence

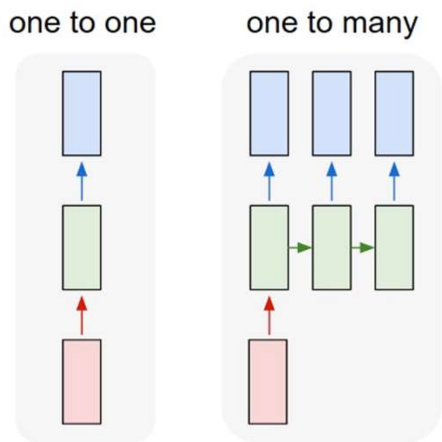
Output: No sequence

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

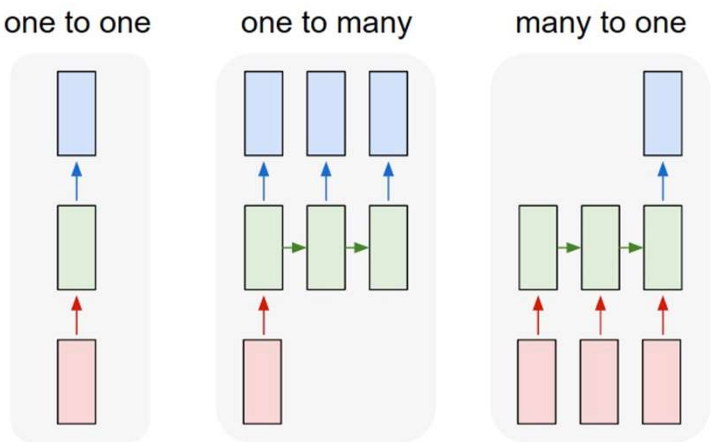
Output: Sequence

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

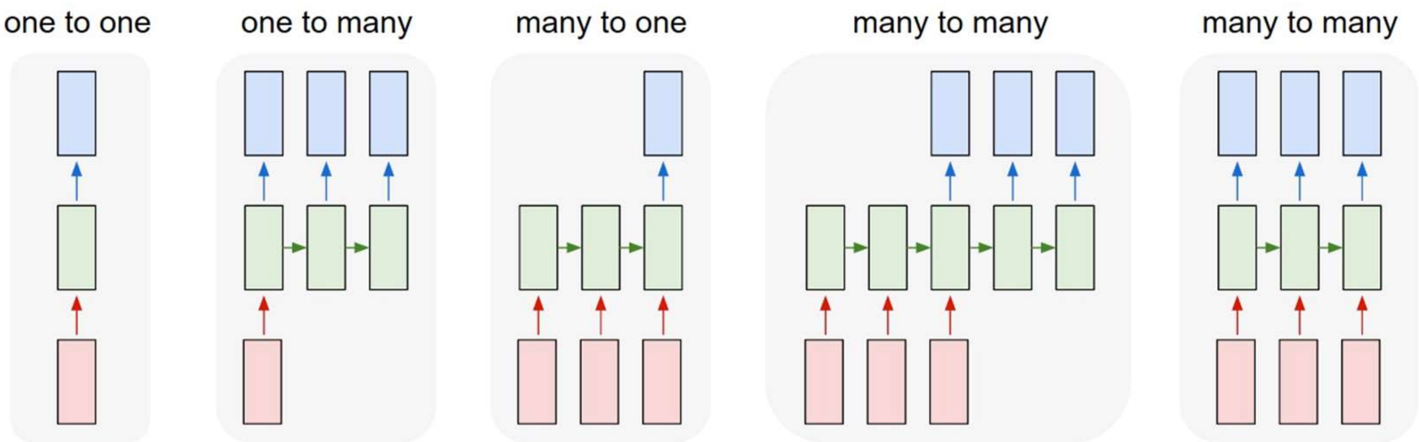
Output: No sequence

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

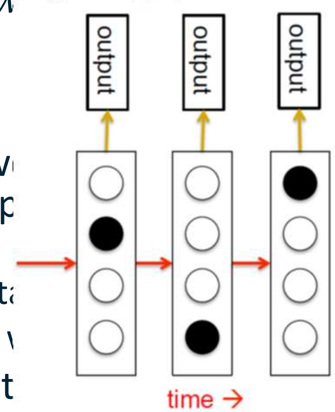
Output: Sequence

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

Image Credit: Andrej Karpathy

(Non-Deep) Ways to deal with sequence labelling

- **Autoregressive models**
  - Predict the next term in a sequence from a fixed number of previous terms using delay taps.
  - 1st-order Autoregressive model, AR(1): $y_t = w_0 + w_1 y_{t-1} + \epsilon_t$
  - 2nd-order Autoregressive model, AR(2): $y_t = w_0 + w_1 y_{t-1} + w$
  - And so on.
- **Hidden Markov Model, HMM**
  - HMMs have a discrete one-of-N hidden state. Transitions betw astic and controlled by a transition probability matrix. Also, the outp tate are also stochastic, and are controlled by emission probabilities.
    - We can not be sure which state produced a given output. So, the st
    - It is easy to represent a probability distribution across the N states
  - To predict the next output we need to infer the probability dist n states. HMMs have efficient algorithms for inference and learning.



What's wrong with MLPs?

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

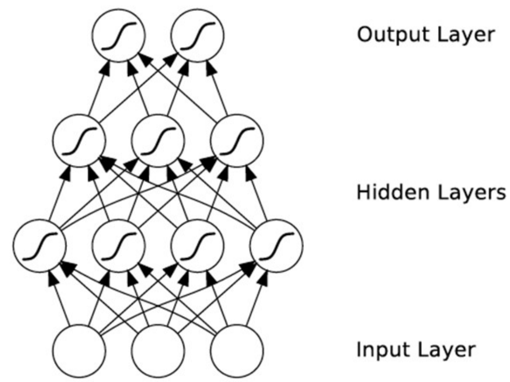
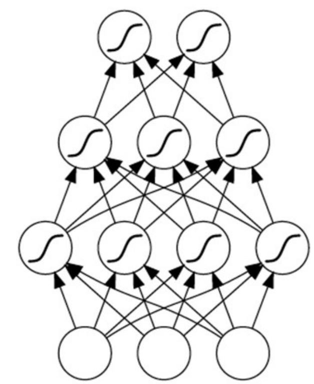


Image Credit: Alex Graves, book

What’s wrong with MLPs?

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

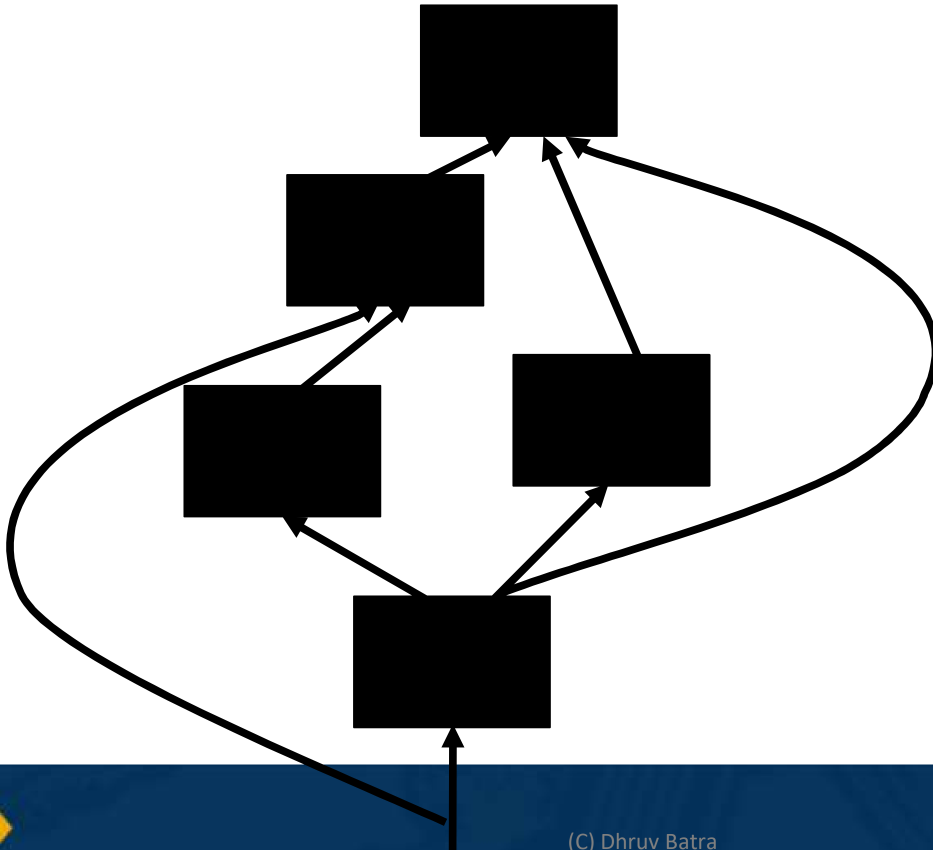
- Problem 2: Pure feed-forward processing
  - No “memory”, no feedback



2 Key Ideas

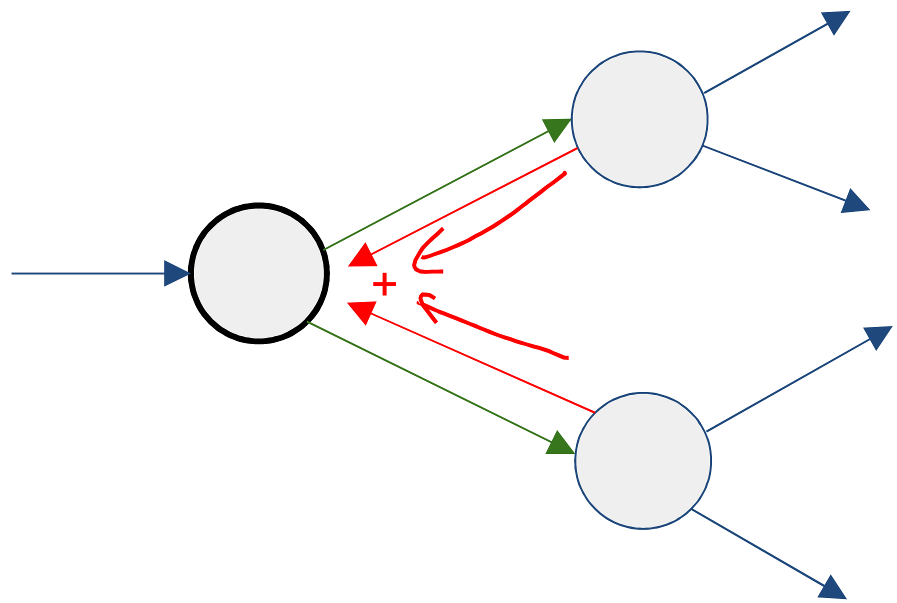
- Parameter Sharing
  - in computation graphs = adding gradients

Computational Graph



Slide Credit: Marc'Aurelio Ranzato

Gradients add at branches



2 Key Ideas

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

How do we model sequences?

- No input

$$s_t = f_\theta(s_{t-1})$$

Image Credit: Bengio, Goodfellow, Courville

How do we model sequences?

- No input

$$s_t = f_\theta(s_{t-1})$$

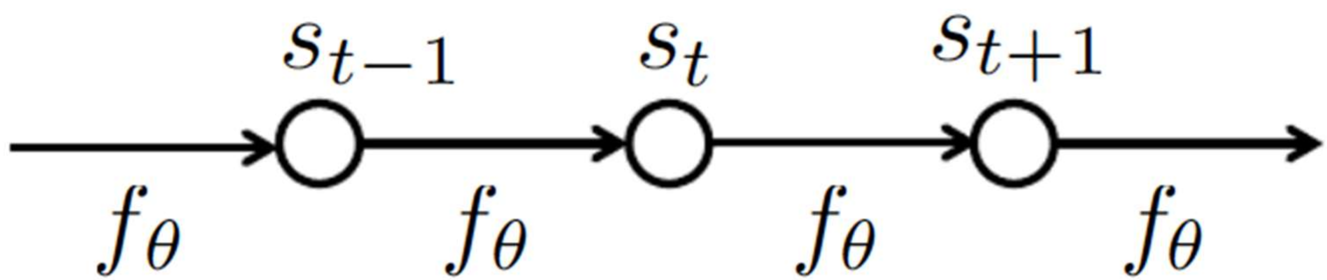


Image Credit: Bengio, Goodfellow, Courville

How do we model sequences?

- With inputs

$$s_t = f_\theta(s_{t-1}, x_t)$$

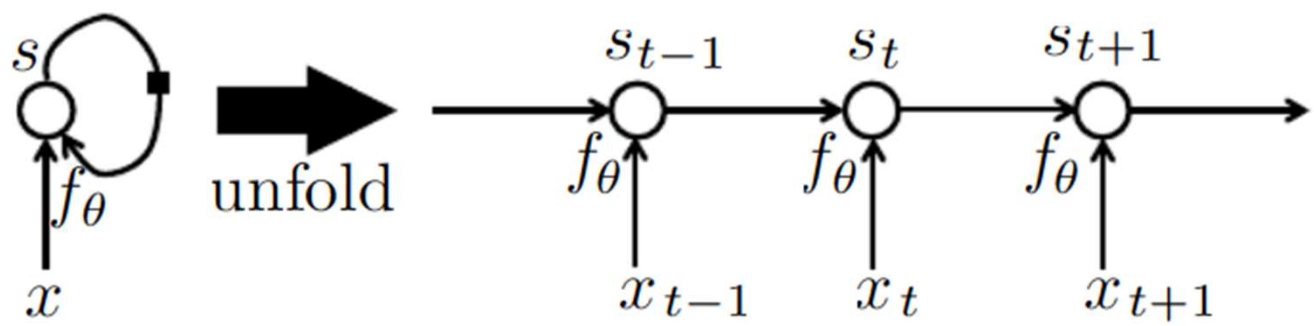


Image Credit: Bengio, Goodfellow, Courville

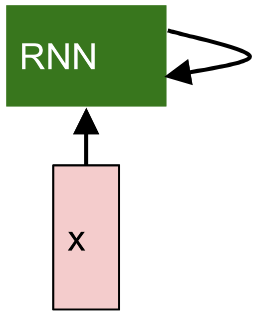
2 Key Ideas

- Parameter Sharing
  - in computation graphs = adding gradients
- “Unrolling”
  - in computation graphs with parameter sharing
- Parameter sharing + Unrolling
  - Allows modeling arbitrary sequence lengths!
  - Keeps numbers of parameters in check

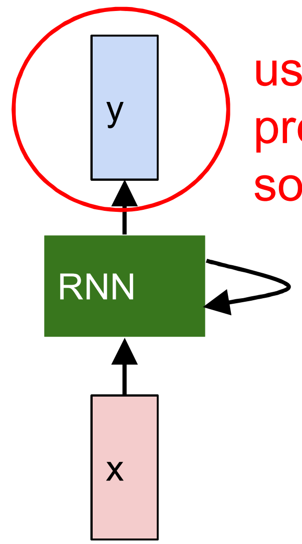
New Words

- 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



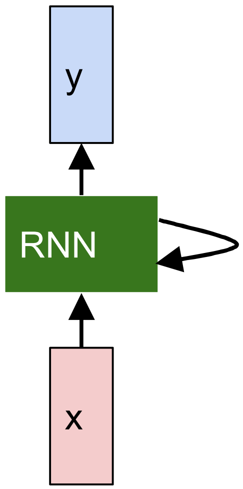
Recurrent Neural Network



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

Recurrent Neural Network

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

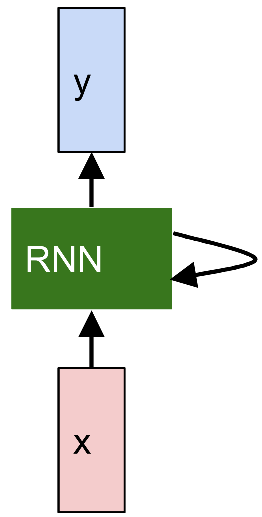
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

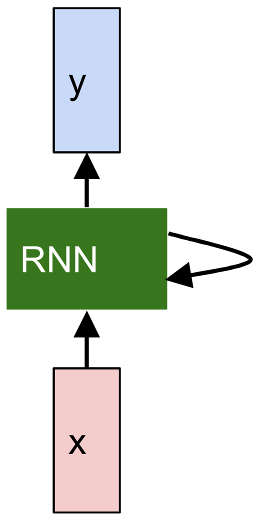


Recurrent Neural Network

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

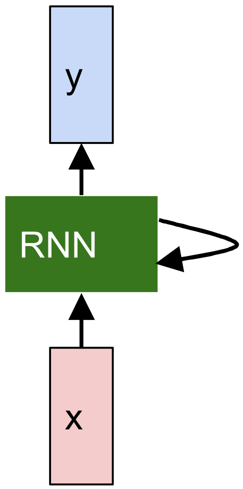
$$h_t = f_W(h_{t-1}, x_t)$$

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



(Vanilla) Recurrent Neural Network

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

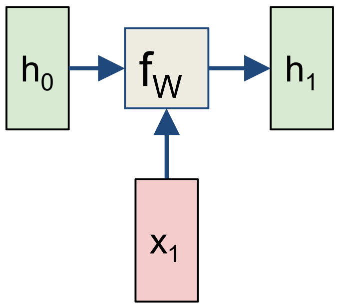


$$y_t = W_{hy} h_t + b_y$$

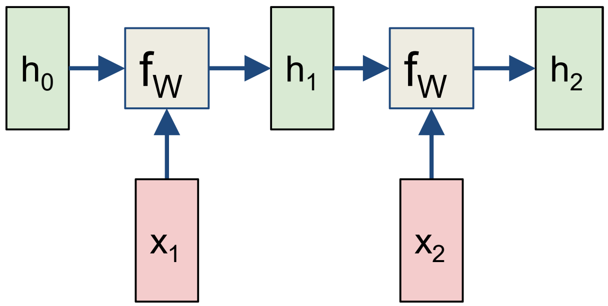
$$\begin{aligned} h_t &= \tanh(W_{hh} h_{t-1} + W_{xh} x_t) \\ &= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \end{aligned}$$

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

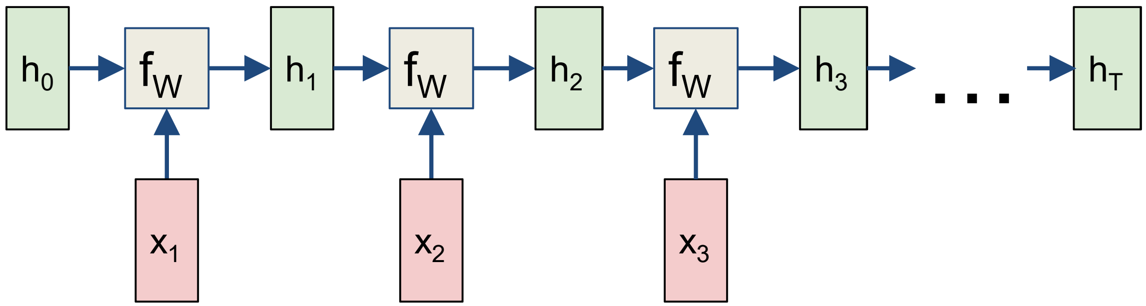
RNN: Computational Graph



RNN: Computational Graph

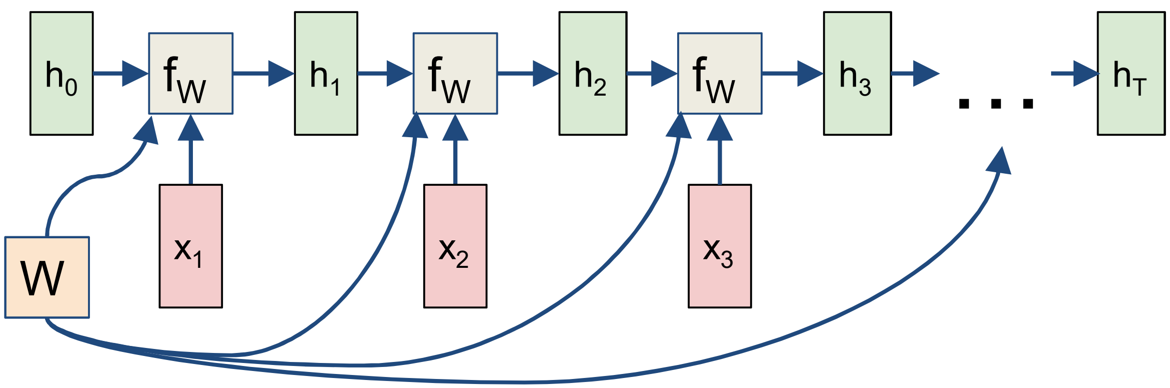


RNN: Computational Graph

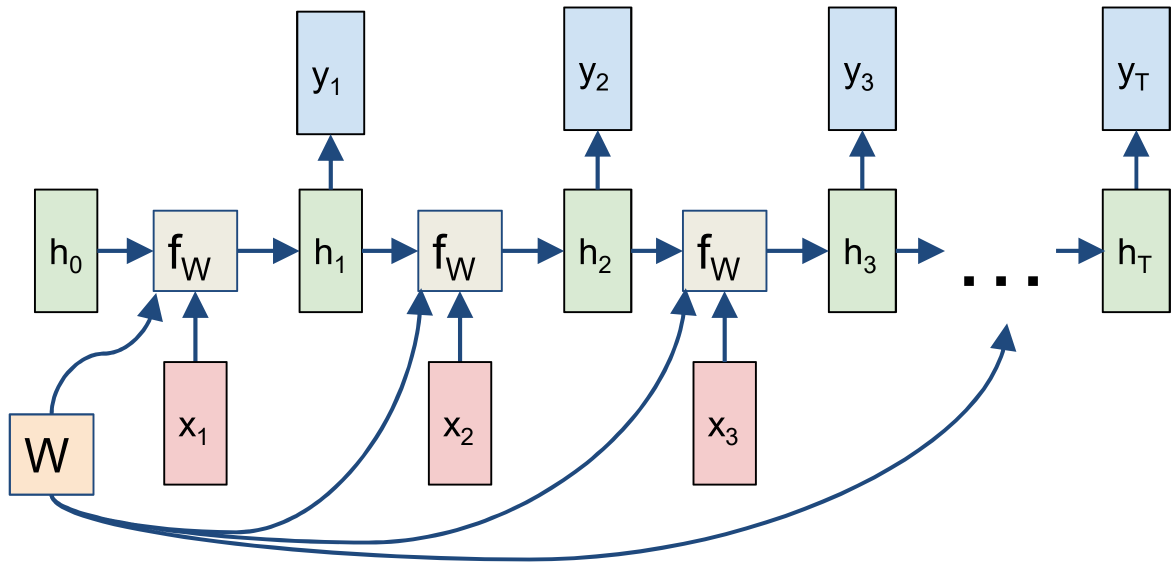


RNN: Computational Graph

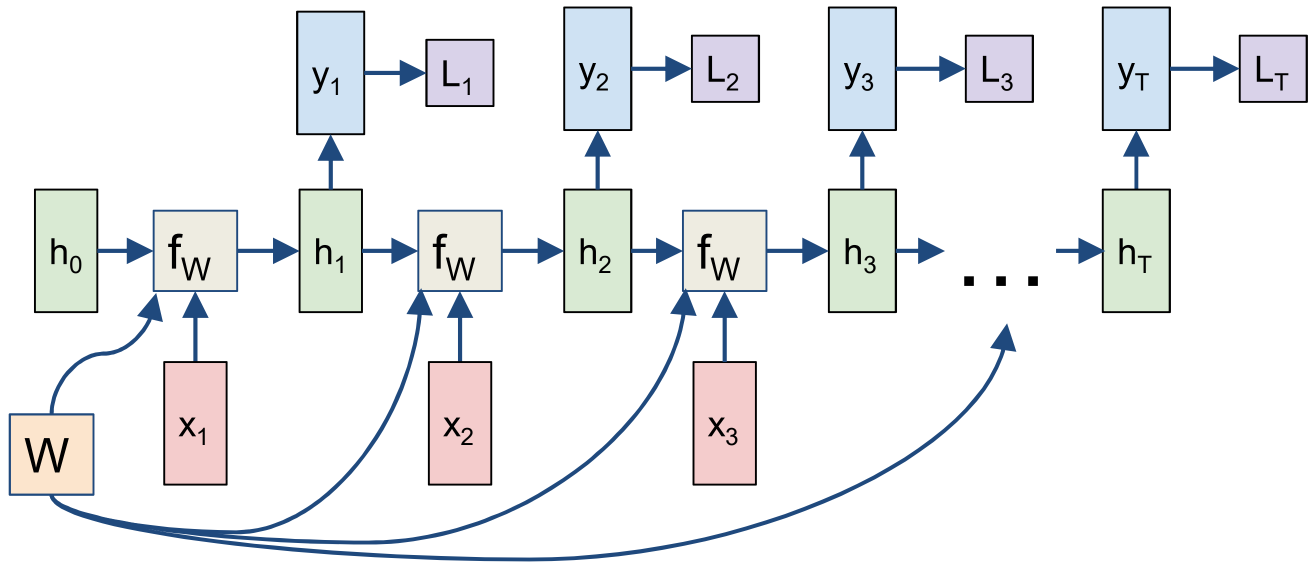
Re-use the same weight matrix at every time-step



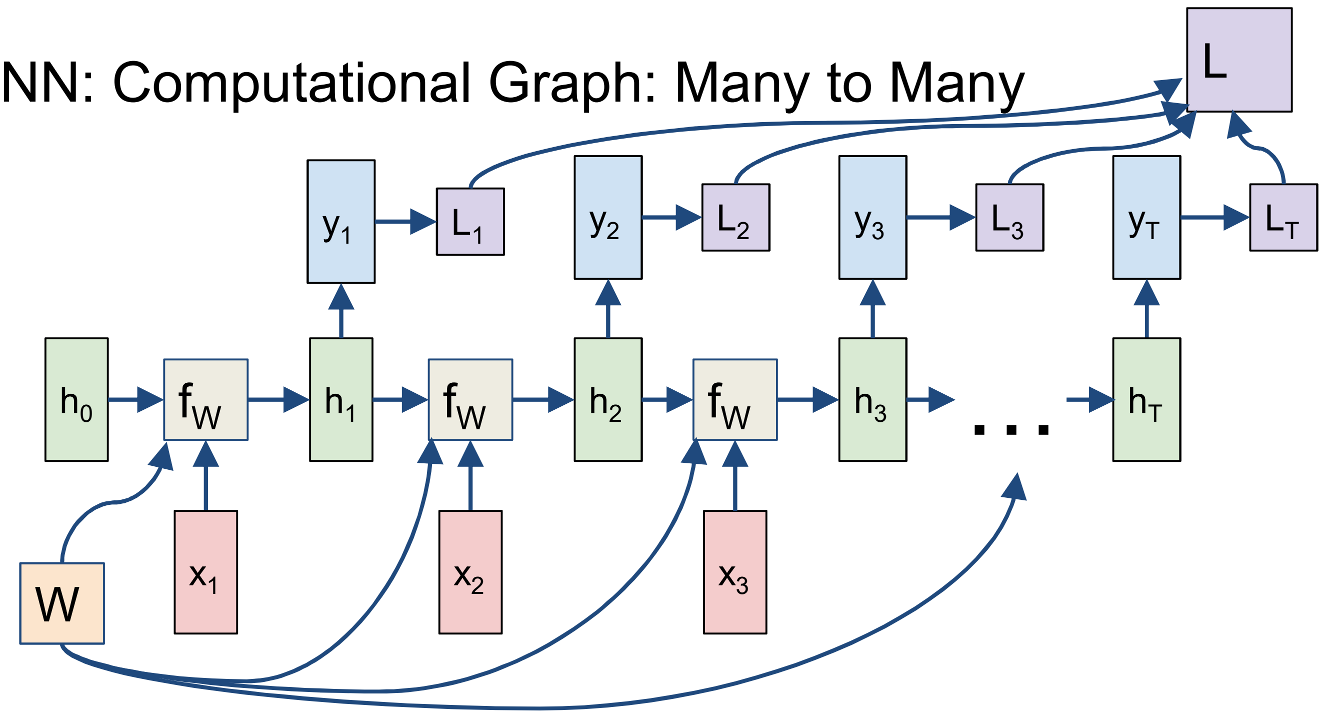
RNN: Computational Graph: Many to Many



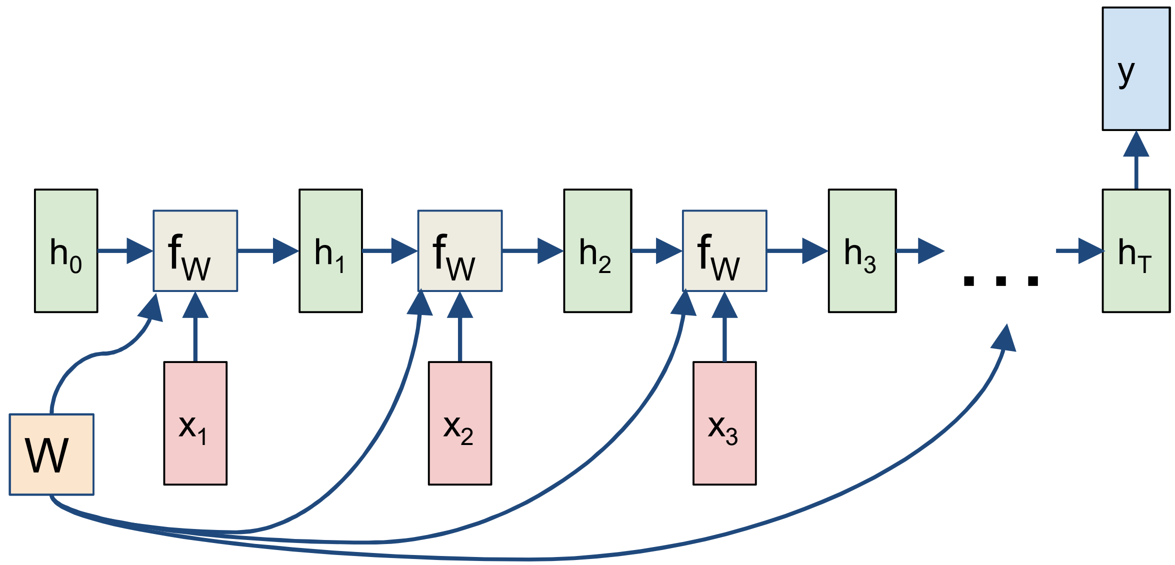
RNN: Computational Graph: Many to Many



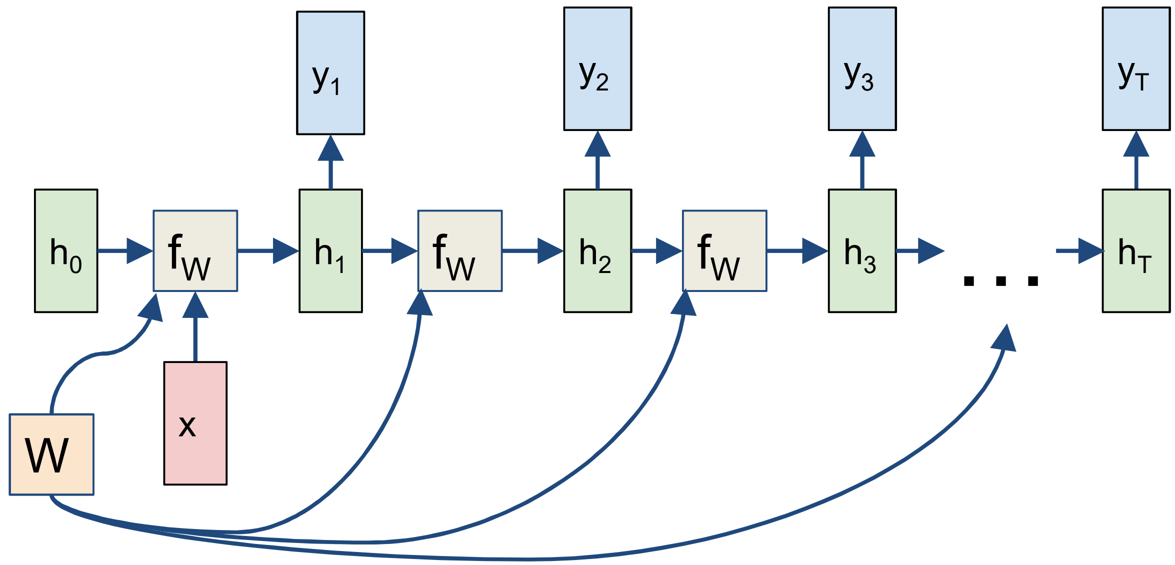
RNN: Computational Graph: Many to Many



RNN: Computational Graph: Many to One

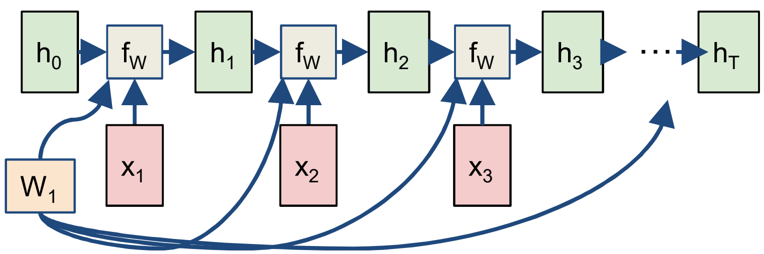


RNN: Computational Graph: One to Many



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

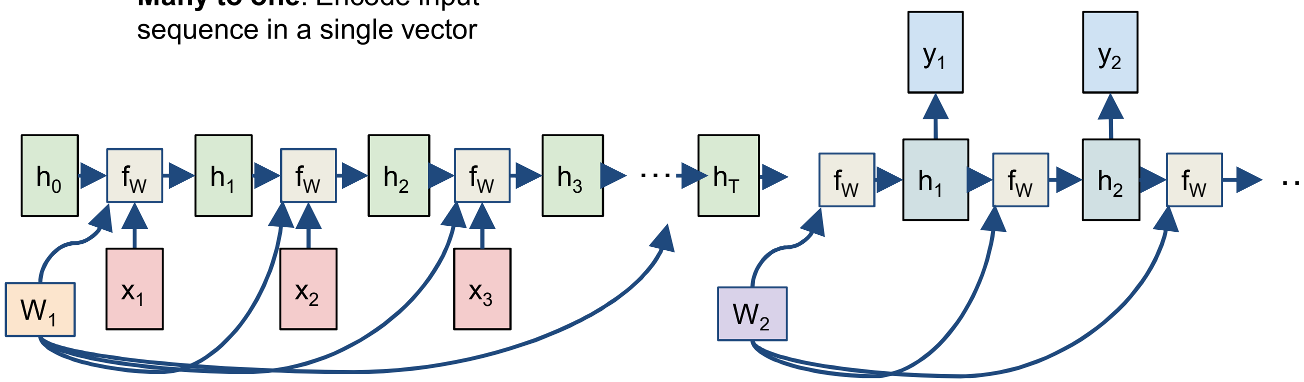
Many to one: Encode input sequence in a single vector



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

One to many: Produce output sequence from single input vector

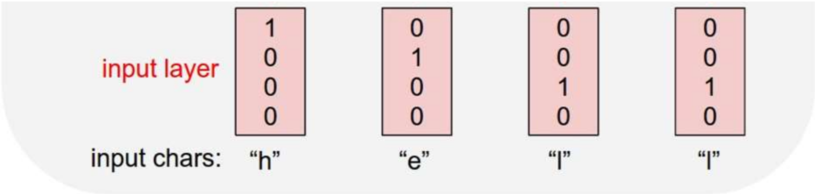
Many to one: Encode input sequence in a single vector



Example: Character-level Language Model

Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**

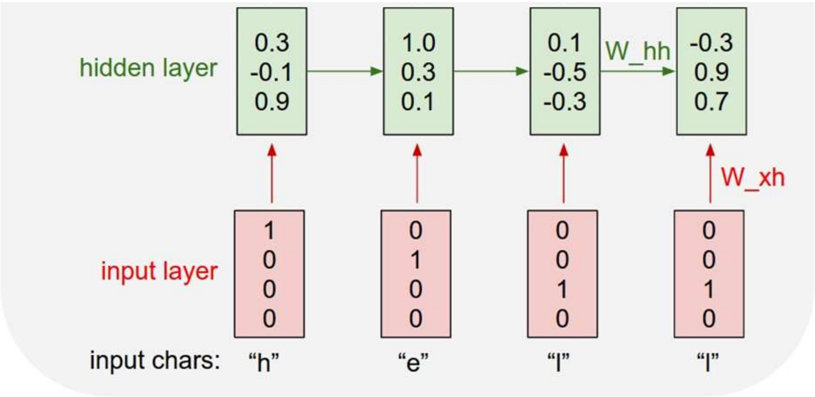


Example: Character-level Language Model

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

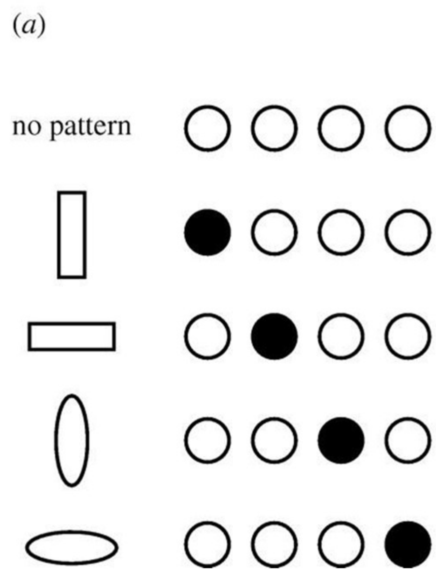
Vocabulary: [h,e,l,o]

Example training sequence: **"hello"**



Distributed Representations Toy Example

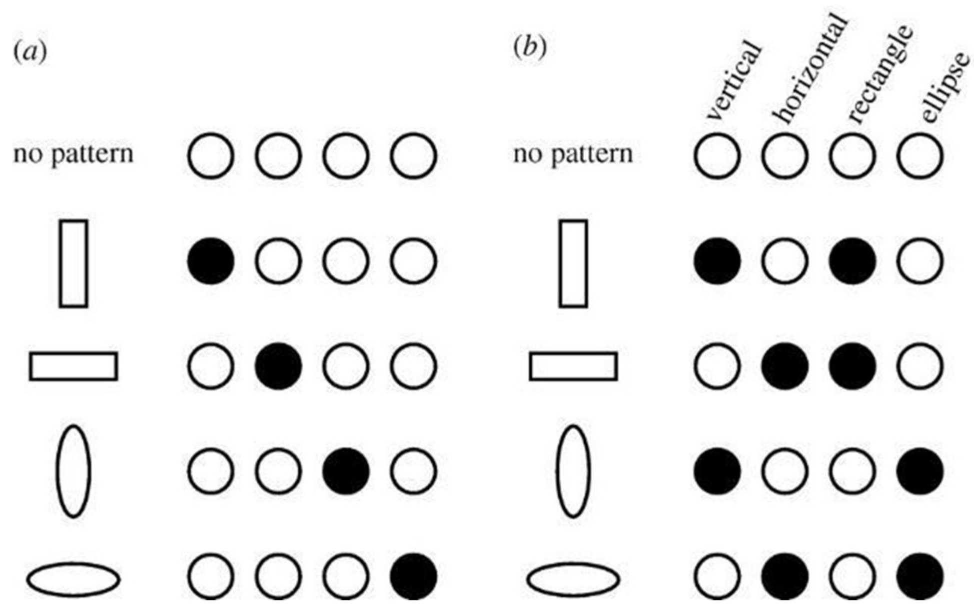
- Local vs Distributed



Slide Credit: Moontae Lee

Distributed Representations Toy Example

- Can we interpret each dimension?



Power of distributed representations!