Machine Learning Department
School of Computer Science
Carnegie Mellon University

RNNs

+

PAC Learning

Matt Gormley & Henry Chai
Lecture 14
Oct. 13, 2021

Reminders

- **Homework 5: Neural Networks**
  - **Out: Mon, Oct. 11**
  - **Due: Thu, Oct. 21 at 11:59pm**
- **More exam viewings today! (Wed, Oct. 13)**
  - **12 – 1**
  - **3 – 5**
  - **Split across BH 235B & BH 255A based on where you took your exam.**
  - @1029 on Piazza

Q&A

Q: In Lecture 12, when you showed us the binary Cross Entropy objective function, was there a minus sign missing?

A: Oops! Yes. Since we want to *minimize* cross entropy, there should have been a minus sign out front!

	Forward	Backward
Quadratic	$J = \frac{1}{2}(y - y^*)^2$	$\frac{dJ}{dy} = y - y^*$
Cross Entropy	$J = -(y^* \log(y) + (1 - y^*) \log(1 - y))$	$\frac{dJ}{dy} = -(y^* \frac{1}{y} + (1 - y^*) \frac{1}{y - 1})$

SGD for CNNs

SGD for CNNs

Ex: Architecture: Given $\vec{x}, y^*$

$$J = \ell(y, y^*)$$
$$y = \text{softmax}(z^{(5)})$$
$$z^{(5)} = \text{linear}(z^{(4)}, W)$$
$$z^{(4)} = \text{relu}(z^{(3)})$$
$$z^{(3)} = \text{conv}(z^{(2)}, \beta)$$
$$z^{(2)} = \text{max-pool}(z^{(1)})$$
$$z^{(1)} = \text{conv}(\vec{x}, \alpha)$$

Parameters $\vec{\Theta} = [\alpha, \beta, W]$

SGD:

1. Init $\vec{\Theta}$
2. While not converged:

   Sample $i \in \{1, \ldots, N\}$

   Forward: $y = h_\Theta(\vec{x}^{(i)})$, $J_i(\Theta) = \ell(y, y^*)$

   Backward: $\nabla_{\vec{\Theta}} J_i(\Theta) = \ldots$

   Update: $\vec{\Theta} \leftarrow \vec{\Theta} - \lambda \nabla_{\vec{\Theta}} J_i(\Theta)$

LAYERS OF A CNN

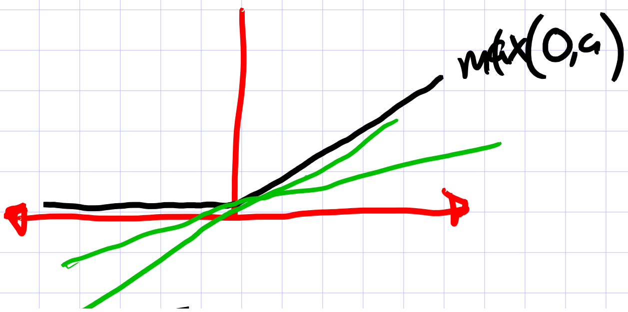
ReLU Layer

$\boxed{\text{ReLU Layer}}$ Input: $\vec{x} \in \mathbb{R}^K$ Output: $\vec{y} \in \mathbb{R}^K$

Forward:

$\vec{y} = \sigma(\vec{x})$ ← element-wise

$\sigma(a) = \max(0, a)$



Backward:

$$\frac{dJ}{dx_i} = \frac{dJ}{dy_i} \frac{dy_i}{dx_i}$$

where $\frac{dy_i}{dx_i} = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$ subderivative

Softmax Layer

Softmax Layer

Input: $\vec{x} \in \mathbb{R}^K$ Output: $\vec{y} \in \mathbb{R}^K$

Forward:

$$y_i = \frac{\exp(x_i)}{\sum_{k=1}^{K} \exp(x_k)}$$

Backward:

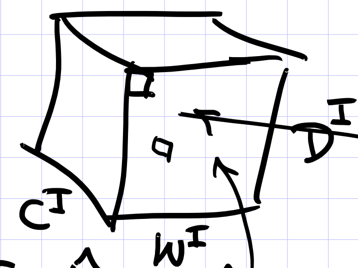
$$\frac{dJ}{dx_j} = \sum_{i=1}^{K} \frac{dJ}{dy_i} \frac{dy_i}{dx_j}$$

where $\frac{dy_i}{dx_j} = \begin{cases} y_i(1-y_i) & \text{if } i=j \\ -y_i y_j & \text{otherwise} \end{cases}$

Fully-Connected Layer

Fully Connected Layer (w/ tensor input)

- Suppose input is a 3D Tensor: $X =$



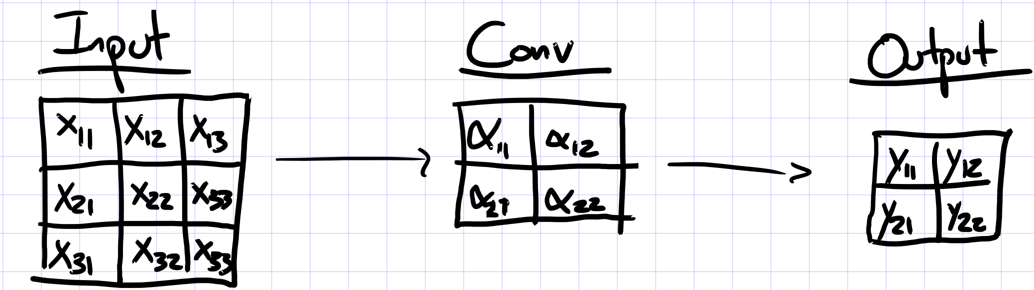
- stretch out into a long vector: $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_{(C\times H\times W)}]$
- then standard linear layer:

$$y = \alpha^T \hat{x} + \alpha_0 \quad \text{where } \alpha \in \mathbb{R}^{A\times B}$$

$$|\hat{x}| = A, \quad |y| = B$$

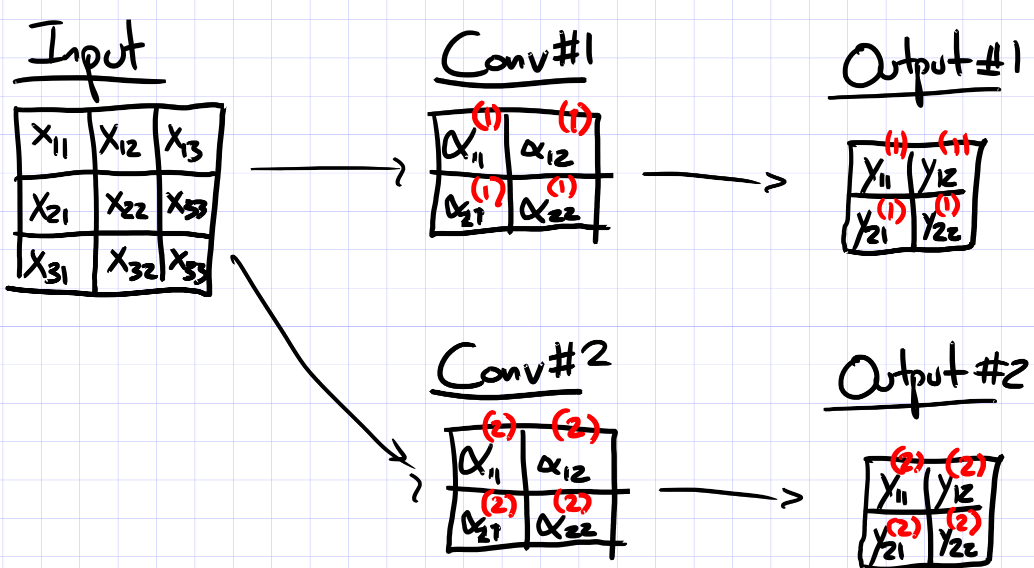
Convolutional Layer

Ex: 1 input channel, 1 output channel



$$y_{11} = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{21}x_{21} + \alpha_{22}x_{22} + \alpha_0$$
$$y_{12} = \alpha_{11}x_{12} + \alpha_{12}x_{13} + \alpha_{21}x_{22} + \alpha_{22}x_{23} + \alpha_0$$
$$y_{21} = \alpha_{11}x_{21} + \alpha_{12}x_{22} + \alpha_{21}x_{31} + \alpha_{22}x_{32} + \alpha_0$$
$$y_{22} = \alpha_{11}x_{22} + \alpha_{12}x_{23} + \alpha_{21}x_{32} + \alpha_{22}x_{33} + \alpha_0$$

Ex: 1 input channel, 2 output channels

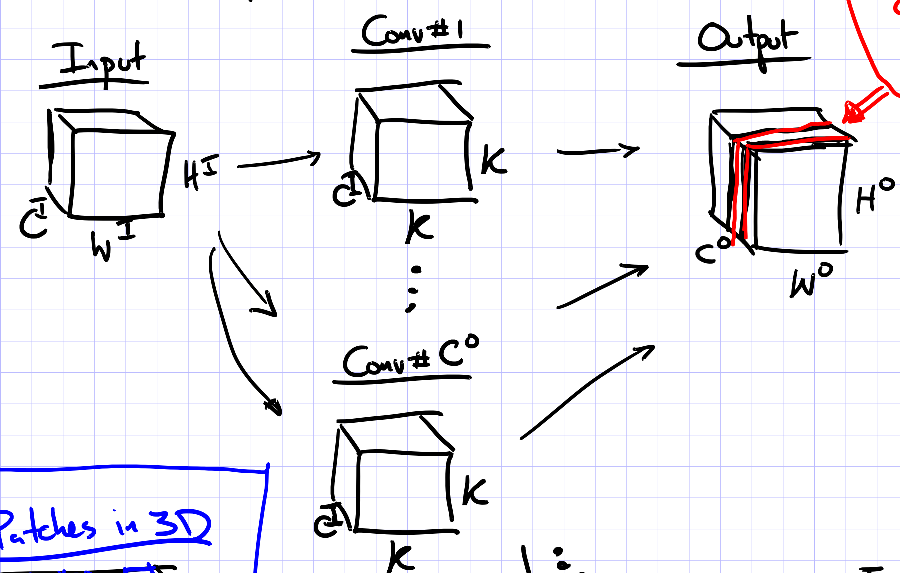


$$y_{11}^{(1)} = \alpha_{11}^{(1)}x_{11} + \alpha_{12}^{(1)}x_{12} + \alpha_{21}^{(1)}x_{21} + \alpha_{22}^{(1)}x_{22} + \alpha_0^{(1)}$$
$$y_{12}^{(1)} = \cdots$$
$$y_{21}^{(1)} = \cdots$$
$$y_{22}^{(1)} = \alpha_{11}^{(1)}x_{22} + \alpha_{12}^{(1)}x_{23} + \alpha_{21}^{(1)}x_{32} + \alpha_{22}^{(1)}x_{33} + \alpha_0^{(1)}$$

$$y_{11}^{(2)} = \alpha_{11}^{(2)}x_{11} + \alpha_{12}^{(2)}x_{12} + \alpha_{21}^{(2)}x_{21} + \alpha_{22}^{(2)}x_{22} + \alpha_0^{(2)}$$
$$y_{12}^{(2)} = \cdots$$
$$y_{21}^{(2)} = \cdots$$
$$y_{22}^{(2)} = \alpha_{11}^{(2)}x_{22} + \alpha_{12}^{(2)}x_{23} + \alpha_{21}^{(2)}x_{32} + \alpha_{22}^{(2)}x_{33} + \alpha_0^{(2)}$$

Convolutional Layer

Ex: C^I input channels, C^I output channels

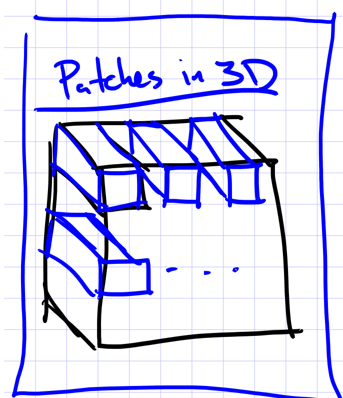


jth slice is output from jth convolution matrix

$$H^O = \lfloor (H^I + 2p - K)/s + 1 \rfloor$$

$$W^O = \lfloor (W^I + 2p - K)/s + 1 \rfloor$$

where p = # pixels of padding on input
K = size of conv. matrix
s = stride length



Forward:

$$y_{ij}^{(k)} = \alpha_0^{(k)} + \sum_{c=1}^{C^I} \sum_{q=1}^{K} \sum_{r=1}^{K} \alpha_{qr}^{(k)} x_{mn}^{(c)} \quad \text{where } \begin{array}{l} m = s(i-1)+q \\ n = s(j-1)+r \end{array}$$

Backward:

$$\frac{dJ}{d\alpha_0^{(k)}} = \sum_i \sum_j \frac{dJ}{dy_{ij}^{(k)}} \frac{dy_{ij}^{(k)}}{d\alpha_0^{(k)}}$$

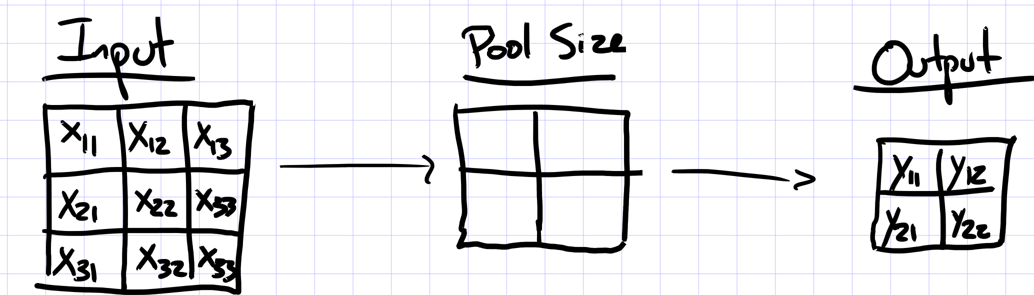
$$\frac{dJ}{d\alpha_{qr}^{(k)}} = \sum_i \sum_j \frac{dJ}{dy_{ij}^{(k)}} \frac{dy_{ij}^{(k)}}{d\alpha_{qr}^{(k)}}$$

$$\frac{dJ}{dx_{mn}^{(c)}} = \sum_i \sum_j \sum_k \frac{dJ}{dy_{ij}^{(k)}} \frac{dy_{ij}^{(k)}}{dx_{mn}^{(c)}}$$

just some calculus

Max-Pooling Layer

Ex: 1 input channel, 1 output channel, stride of 1



$$y_{11} = \max(x_{11}, x_{12}, x_{21}, x_{22})$$

$$y_{12} = \max(x_{12}, x_{13}, x_{22}, x_{23})$$

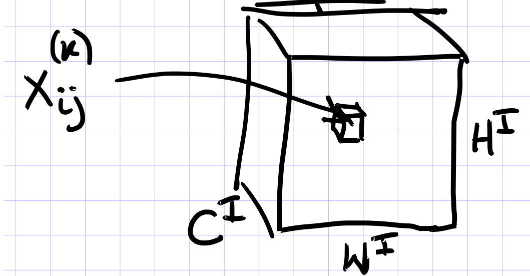
$$y_{21} = \max(x_{21}, x_{22}, x_{31}, x_{32})$$

$$y_{22} = \max(x_{22}, x_{23}, x_{32}, x_{33})$$

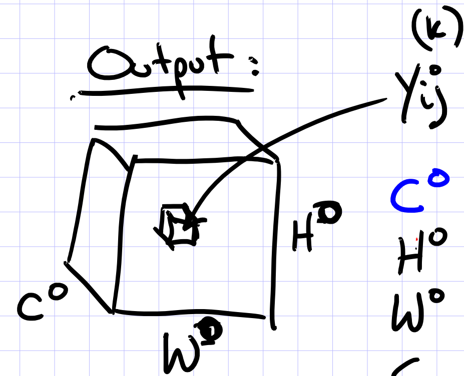
Max-Pooling Layer

Max-Pooling Layer

Input :



Output :



$C^O = C^I$

$\left.\begin{array}{l} H^O = \ldots \\ W^O = \ldots \end{array}\right\}$ same as conv. layer

Same hyperparams too.

Forward :

$$y_{ij}^{(k)} = \max_{\substack{q \in \{1,\ldots,K\} \\ r \in \{1,\ldots,K\}}} x_{mn}^{(k)} \quad \text{where} \quad \begin{array}{l} m = s(i-1)+q \\ n = s(j-1)+r \end{array}$$

Backward :

$$\frac{dJ}{dx_{mn}^{(k)}} = \sum_i \sum_j \frac{dJ}{dy_{ij}^{(k)}} \frac{dy_{ij}^{(k)}}{dx_{mn}^{(k)}}$$

Subderivatives

+ Max() is not differentiable, but subdifferentiable.
+ There are a set of derivatives and we can just choose one for SGD.

$y = \max(a, b)$

$$\Rightarrow \frac{dJ}{da} = \frac{dJ}{dy}\frac{dy}{da} \quad \text{where} \quad \frac{dy}{da} = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{otherwise} \end{cases}$$

Convolutional Neural Network (CNN)

- Typical layers include:
  - Convolutional layer
  - Max-pooling layer
  - Fully-connected (Linear) layer
  - ReLU layer (or some other nonlinear activation function)
  - Softmax
- These can be arranged into arbitrarily deep topologies

Architecture #1: LeNet-5

PROC. OF THE IEEE, NOVEMBER 1998 7

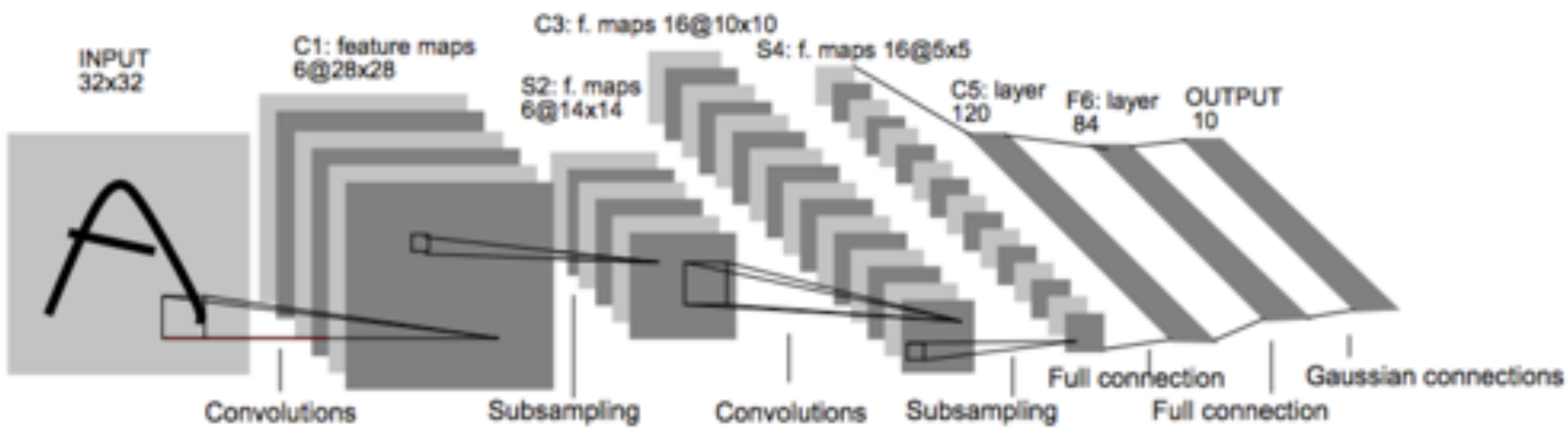


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

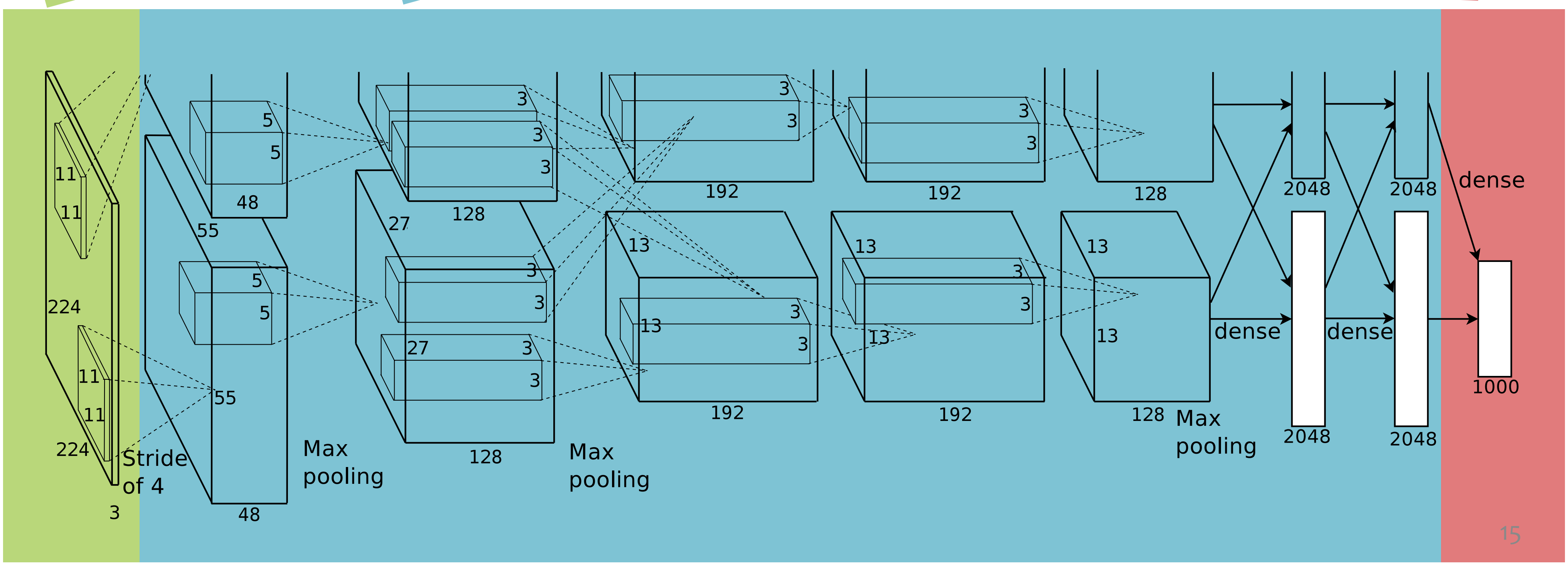
Architecture #2: AlexNet

CNN for Image Classification
(Krizhevsky, Sutskever & Hinton, 2012)
15.3% error on ImageNet LSVRC-2012 contest

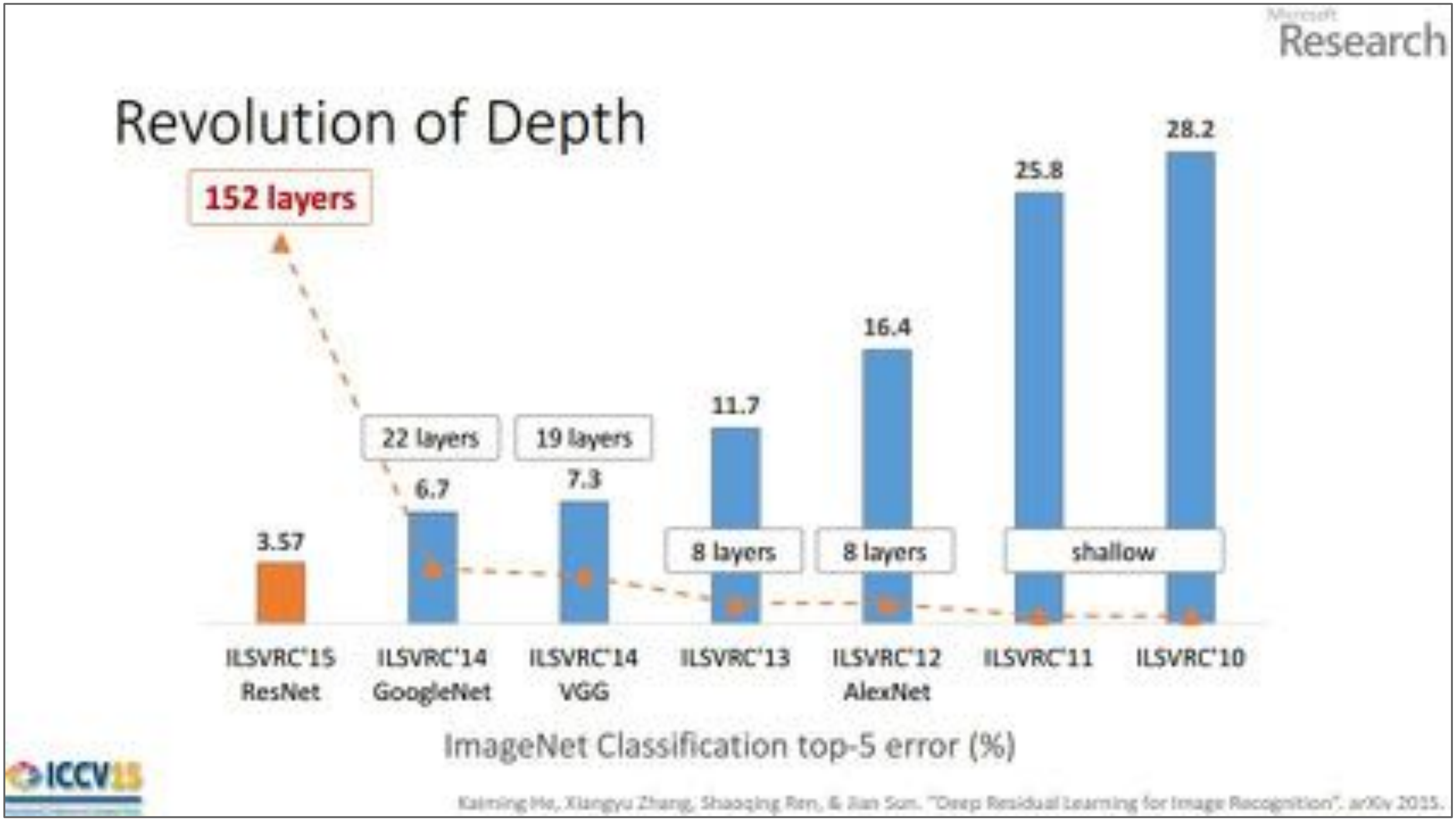
Input image (pixels)

- Five convolutional layers (w/max-pooling)
- Three fully connected layers

1000-way softmax



CNNs for Image Recognition

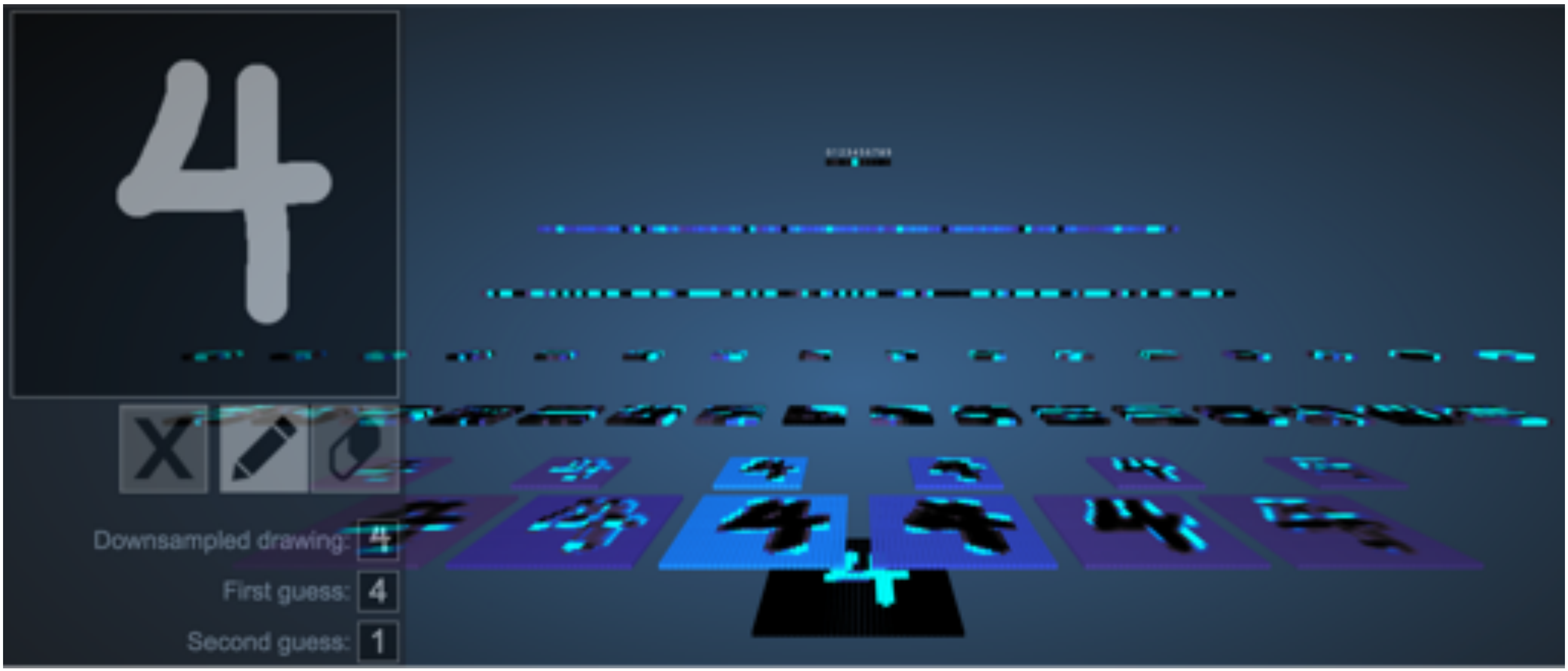


Slide from Kaiming He

CNN VISUALIZATIONS

3D Visualization of CNN

http://scs.ryerson.ca/~aharley/vis/conv/



Convolution of a Color Image

- Color images consist of 3 floats per pixel for RGB (red, green blue) color values
- Convolution must also be 3-dimensional

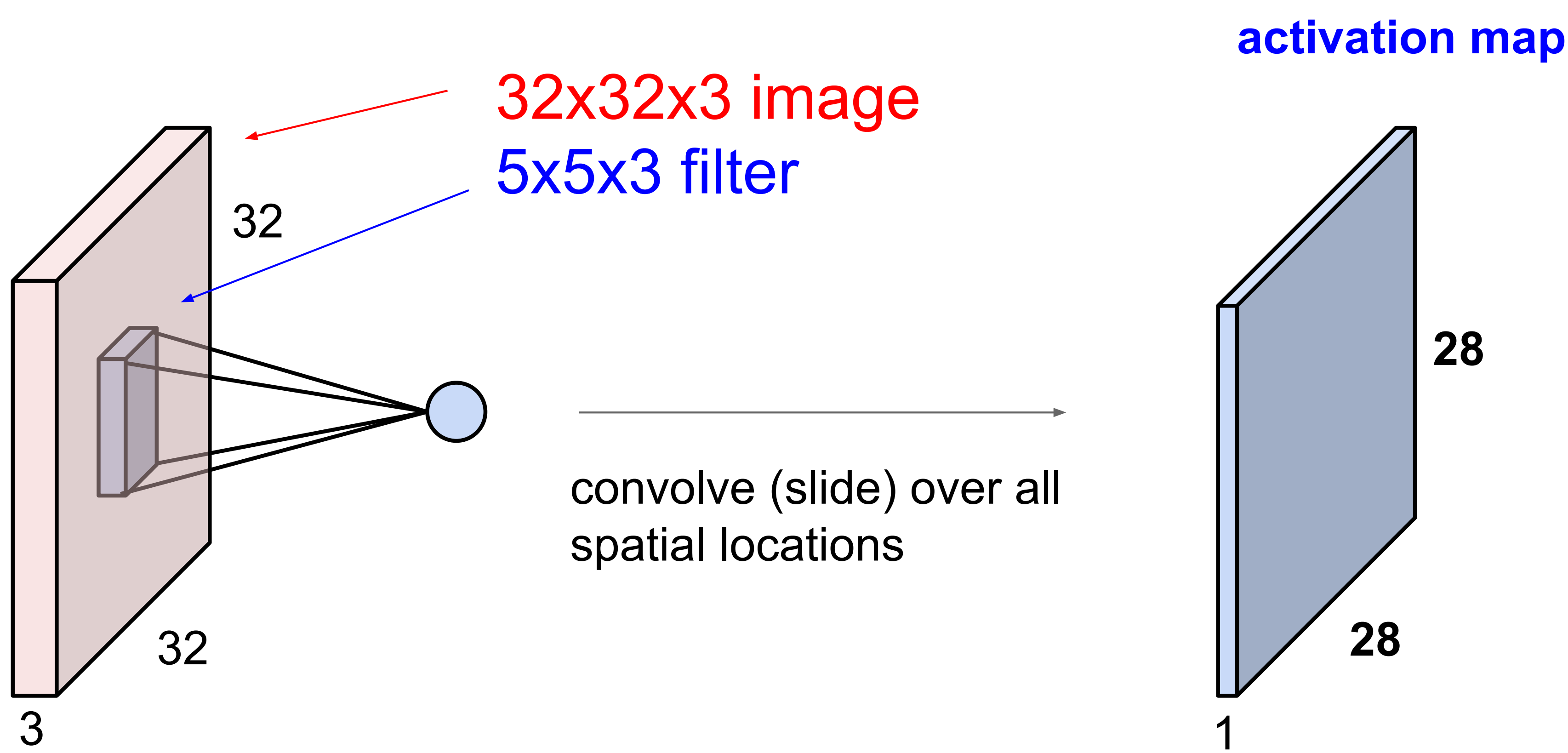


Figure from Fei-Fei Li & Andrej Karpathy & Justin Johnson (CS231N)

Animation of 3D Convolution

http://cs231n.github.io/convolutional-networks/

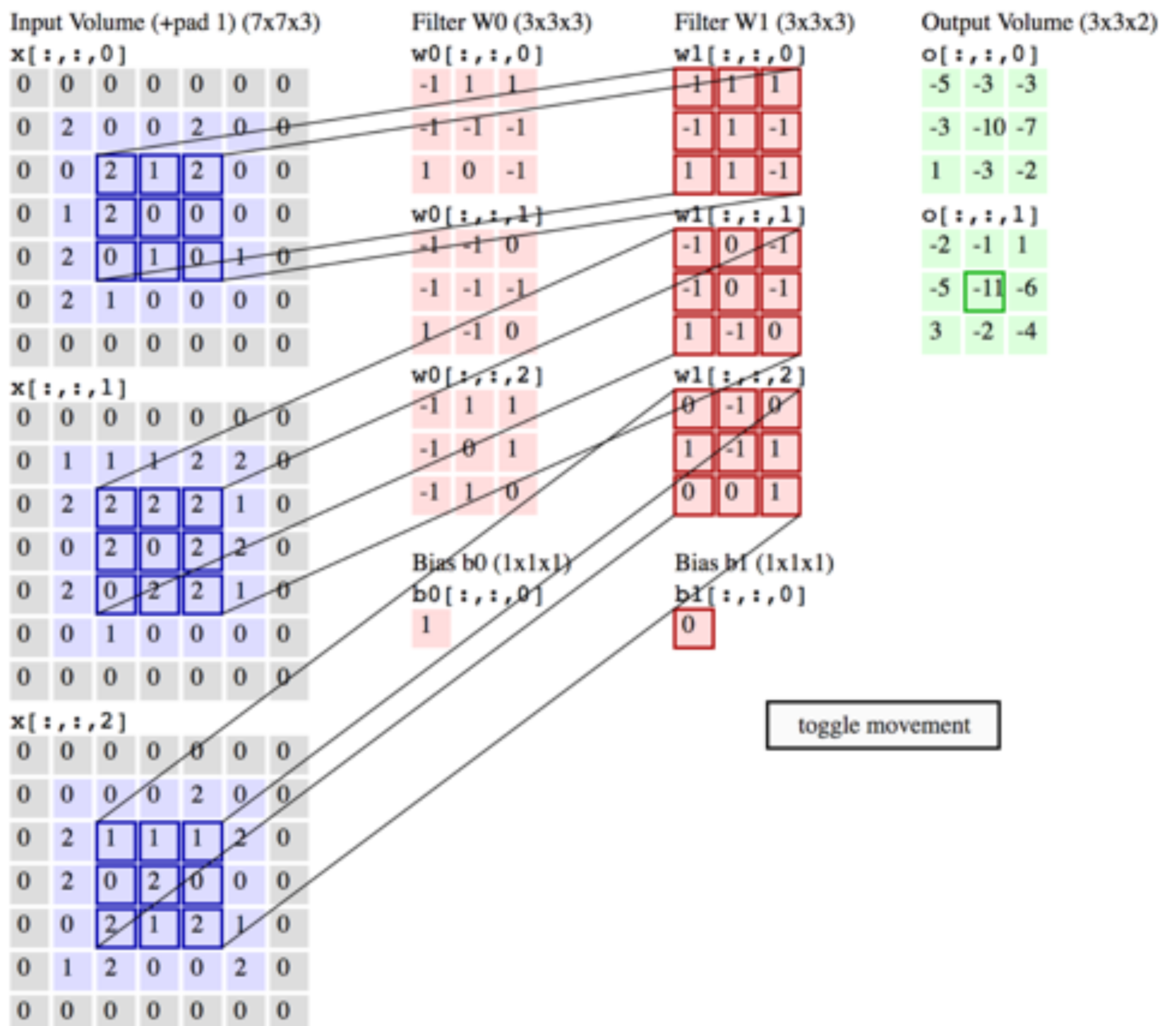


Figure from Fei-Fei Li & Andrej Karpathy & Justin Johnson (CS231N)

MNIST Digit Recognition with CNNs (in your browser)

https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html

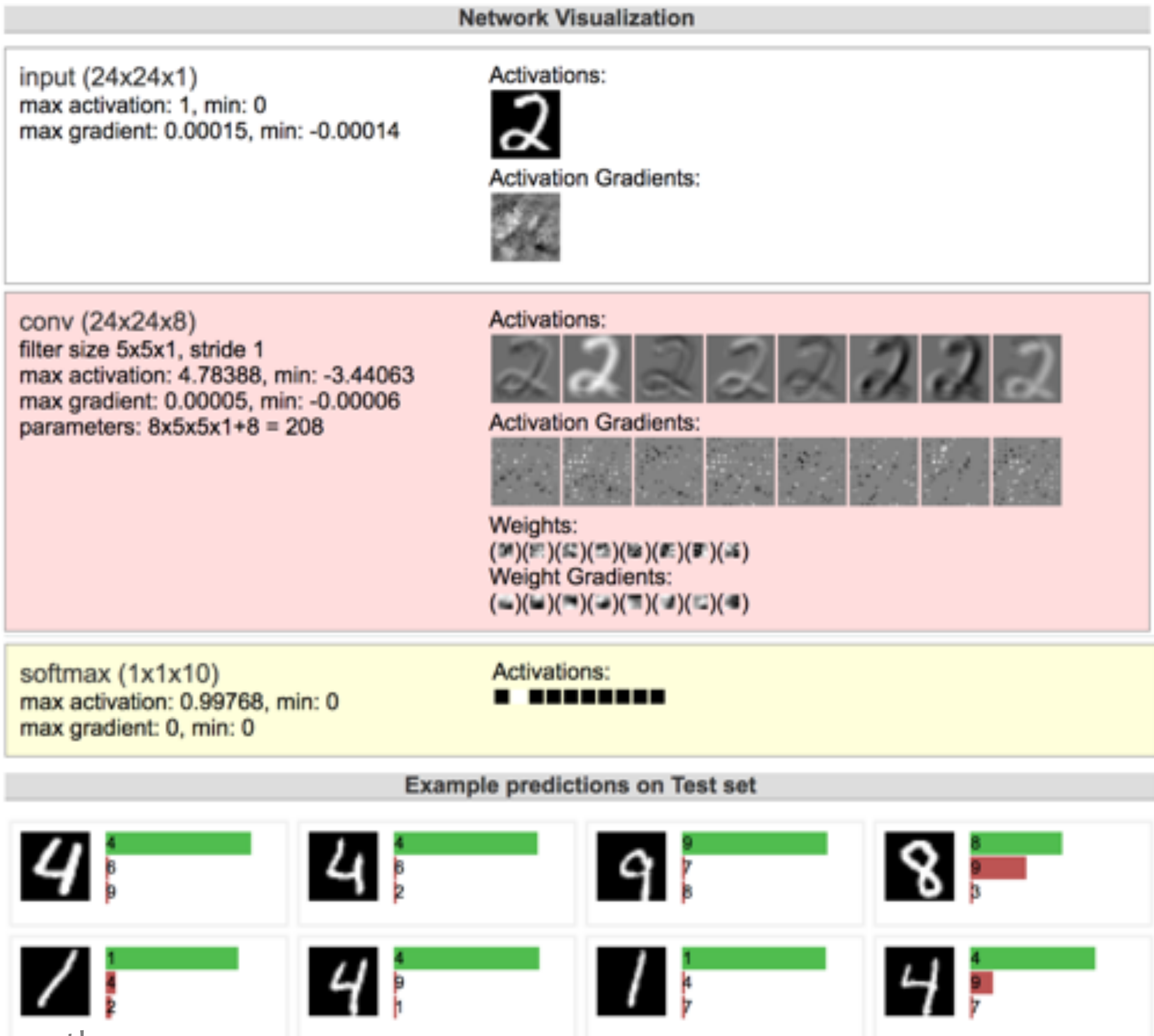


Figure from Andrej Karpathy

CNN Summary

CNNs

- Are used for all aspects of **computer vision**, and have won numerous pattern recognition competitions
- Able learn **interpretable features** at different levels of abstraction
- Typically, consist of **convolution** layers, **pooling** layers, **nonlinearities**, and **fully connected** layers

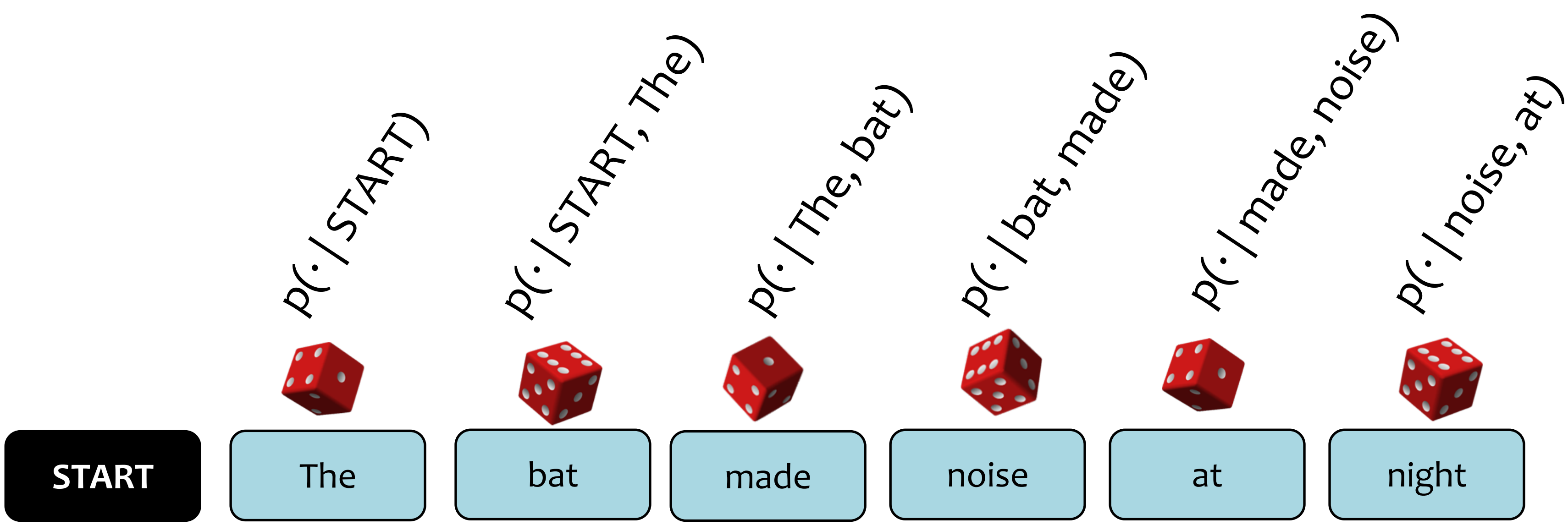
Other Resources:

- Readings on course website
- Andrej Karpathy, CS231n Notes http://cs231n.github.io/convolutional-networks/

BACKGROUND:
N-GRAM LANGUAGE MODELS

n-Gram Language Model

- *Goal*: Generate realistic looking sentences in a human language
- *Key Idea*: condition on the last n-1 words to sample the n^{th} word



n-Gram Language Model

<u>Question</u>: How can we **define** a probability distribution over a sequence of length T?

The	bat	made	noise	at	night
w_1	w_2	w_3	w_4	w_5	w_6

n-Gram Model (n=2)

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_{t-1})$$

$$p(w_1, w_2, w_3, \ldots, w_6) =$$

The	$p(w_1)$
The bat	$p(w_2 \mid w_1)$
bat made	$p(w_3 \mid w_2)$
made noise	$p(w_4 \mid w_3)$
noise at	$p(w_5 \mid w_4)$
at night	$p(w_6 \mid w_5)$

n-Gram Language Model

Question: How can we **define** a probability distribution over a sequence of length T?

The	bat	made	noise	at	night
w_1	w_2	w_3	w_4	w_5	w_6

n-Gram Model (n=3)

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_{t-1}, w_{t-2})$$

$p(w_1, w_2, w_3, \ldots, w_6) =$

The						$p(w_1)$
The	bat					$p(w_2 \mid w_1)$
The	bat	made				$p(w_3 \mid w_2, w_1)$
	bat	made	noise			$p(w_4 \mid w_3, w_2)$
		made	noise	at		$p(w_5 \mid w_4, w_3)$
			noise	at	night	$p(w_6 \mid w_5, w_4)$

n-Gram Language Model

Question: How can we **define** a probability distribution over a sequence of length T?

The	bat	made	noise	at	night
w_1	w_2	w_3	w_4	w_5	w_6

n-Gram Model (n=3)

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_{t-1}, w_{t-2})$$

$p(w_1, w_2, w_3, \ldots, w_6) =$

The $\quad p(w_1)$

The

The

Note: This is called a **model** because we made some **assumptions** about how many previous words to condition on (i.e. only n-1 words)

Learning an n-Gram Model

<u>*Question*</u>: How do we **learn** the probabilities for the n-Gram Model?

$p(w_t \mid w_{t-2} = \text{The}, w_{t-1} = \text{bat})$

w_t	$p(\cdot \mid \cdot, \cdot)$
ate	0.015
…	
flies	0.046
…	
zebra	0.000

$p(w_t \mid w_{t-2} = \text{made}, w_{t-1} = \text{noise})$

w_t	$p(\cdot \mid \cdot, \cdot)$
at	0.020
…	
pollution	0.030
…	
zebra	0.000

$p(w_t \mid w_{t-2} = \text{cows}, w_{t-1} = \text{eat})$

w_t	$p(\cdot \mid \cdot, \cdot)$
corn	0.420
…	
grass	0.510
…	
zebra	0.000

Learning an n-Gram Model

Question: How do we **learn** the probabilities for the n-Gram Model?
Answer: From data! Just **count** n-gram frequencies

… the **cows eat** grass…
… our **cows eat hay** daily…
… factory-farm **cows eat** corn…
… on an organic farm, **cows eat hay** and…
… do your **cows eat** grass or corn?...
… what do **cows eat if** they have…
… **cows eat** corn when there is no…
… which **cows eat which** foods depends…
… if **cows eat** grass…
… when **cows eat** corn their stomachs…
… should we let **cows eat** corn?...

$p(w_t \mid w_{t-2} = \text{cows}, w_{t-1} = \text{eat})$

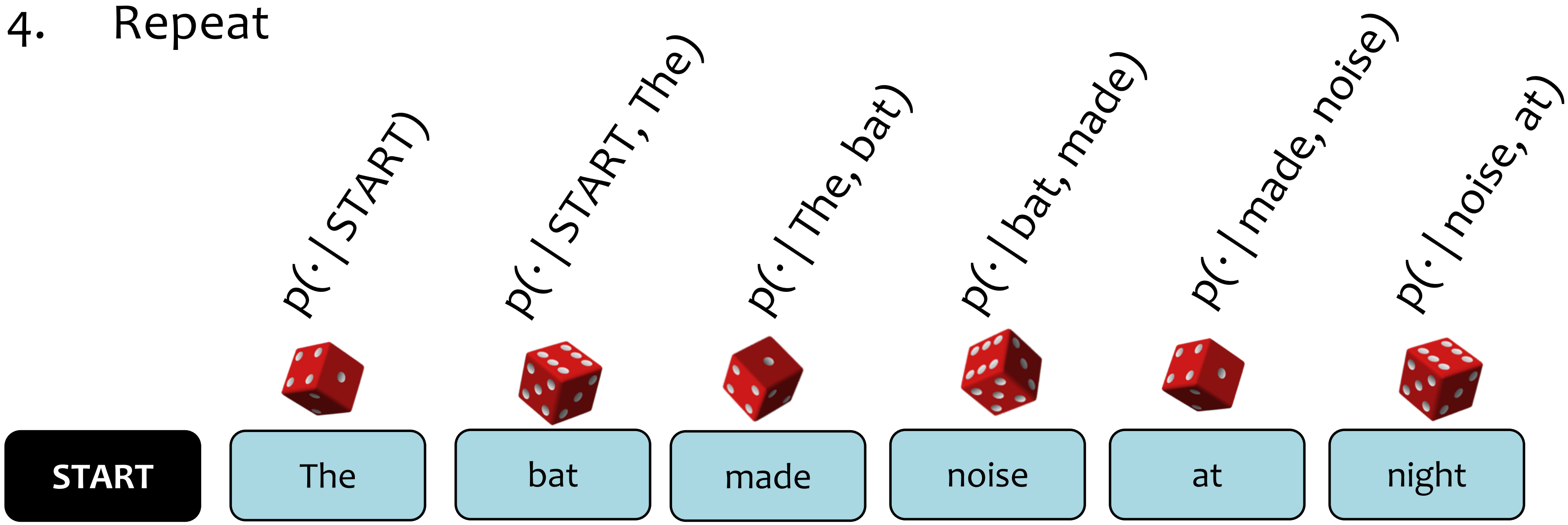
w_t	$p(\cdot \mid \cdot, \cdot)$
corn	4/11
grass	3/11
hay	2/11
if	1/11
which	1/11

Sampling from a Language Model

Question: How do we sample from a Language Model?

Answer:

1. Treat each probability distribution like a (50k-sided) weighted die
2. Pick the die corresponding to $p(w_t \mid w_{t-2}, w_{t-1})$
3. Roll that die and generate whichever word w_t lands face up
4. Repeat



Sampling from a Language Model

Question: How do we sample from a Language Model?
Answer:

1. Treat each probability distribution like a (50k-sided) weighted die
2. Pick the die corresponding to $p(w_t \mid w_{t-2}, w_{t-1})$
3. Roll that die and generate whichever word w_t lands face up
4. Repeat

Training Data (Shakespeaere)	**5-Gram Model**
I tell you, friends, most charitable care ave the patricians of you. For your wants, Your suffering in this dearth, you may as well Strike at the heaven with your staves as lift them Against the Roman state, whose course will on The way it takes, cracking ten thousand curbs Of more strong link asunder than can ever Appear in your impediment. For the dearth, The gods, not the patricians, make it, and Your knees to them, not arms, must help.	`Approacheth, denay. dungy Thither! Julius think: grant,--O Yead linens, sheep's Ancient, Agreed: Petrarch plaguy Resolved pear! observingly honourest adulteries wherever scabbard guess; affirmation--his monsieur; died. jealousy, chequins me. Daphne building. weakness: sun- rise, cannot stays carry't, unpurposed. prophet-like drink; back-return 'gainst surmise Bridget ships? wane; interim? She's striving wet;`

RECURRENT NEURAL NETWORK (RNN) LANGUAGE MODELS

Recurrent Neural Networks (RNNs)

inputs: $\mathbf{x} = (x_1, x_2, \dots, x_T), x_i \in \mathcal{R}^I$

hidden units: $\mathbf{h} = (h_1, h_2, \dots, h_T), h_i \in \mathcal{R}^J$

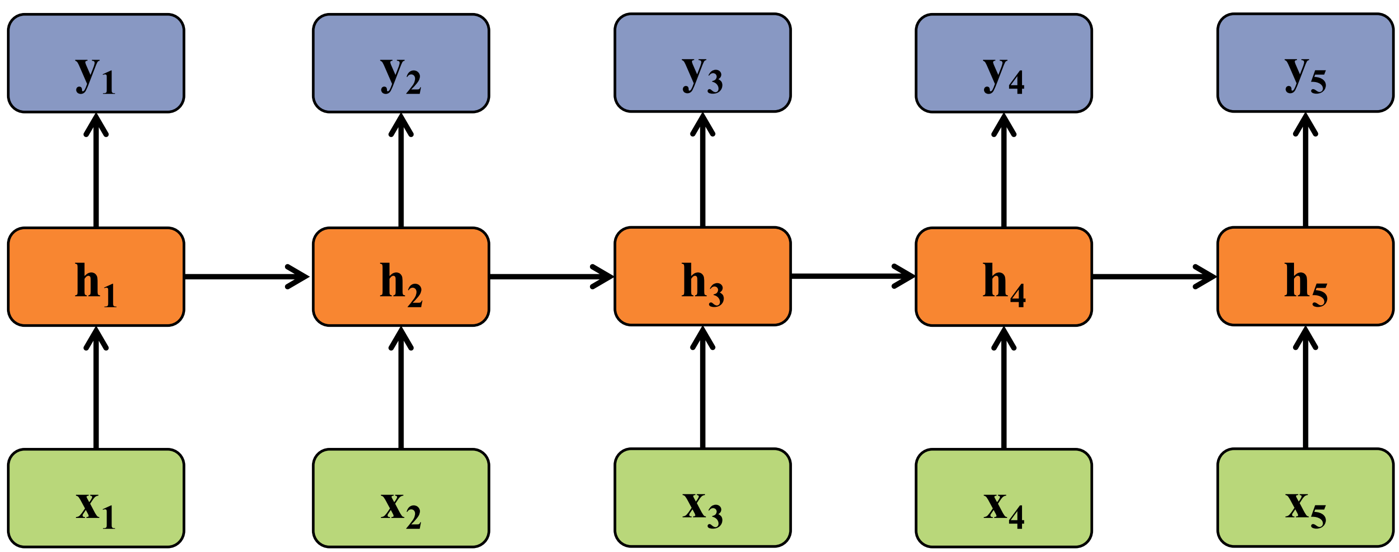
outputs: $\mathbf{y} = (y_1, y_2, \dots, y_T), y_i \in \mathcal{R}^K$

nonlinearity: $\mathcal{H}$

Definition of the RNN:

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right)$$

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



The Chain Rule of Probability

Recall…

Question: How can we **define** a probability distribution over a sequence of length T?

The	bat	made	noise	at	night
w_1	w_2	w_3	w_4	w_5	w_6

Chain rule of probability: $$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_{t-1}, \ldots, w_1)$$

$p(w_1, w_2, w_3, \ldots, w_6) =$

The $\quad p(w_1)$

The

The

The

The

The $\quad p(w_6 \mid w_5, w_4, w_3, w_2, w_1)$

Note: This is called the chain **rule** because it is **always** true for every probability distribution

RNN Language Model

RNN Language Model: $p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid f_{\boldsymbol{\theta}}(w_{t-1}, \ldots, w_1))$

$$p(w_1, w_2, w_3, \ldots, w_6) =$$

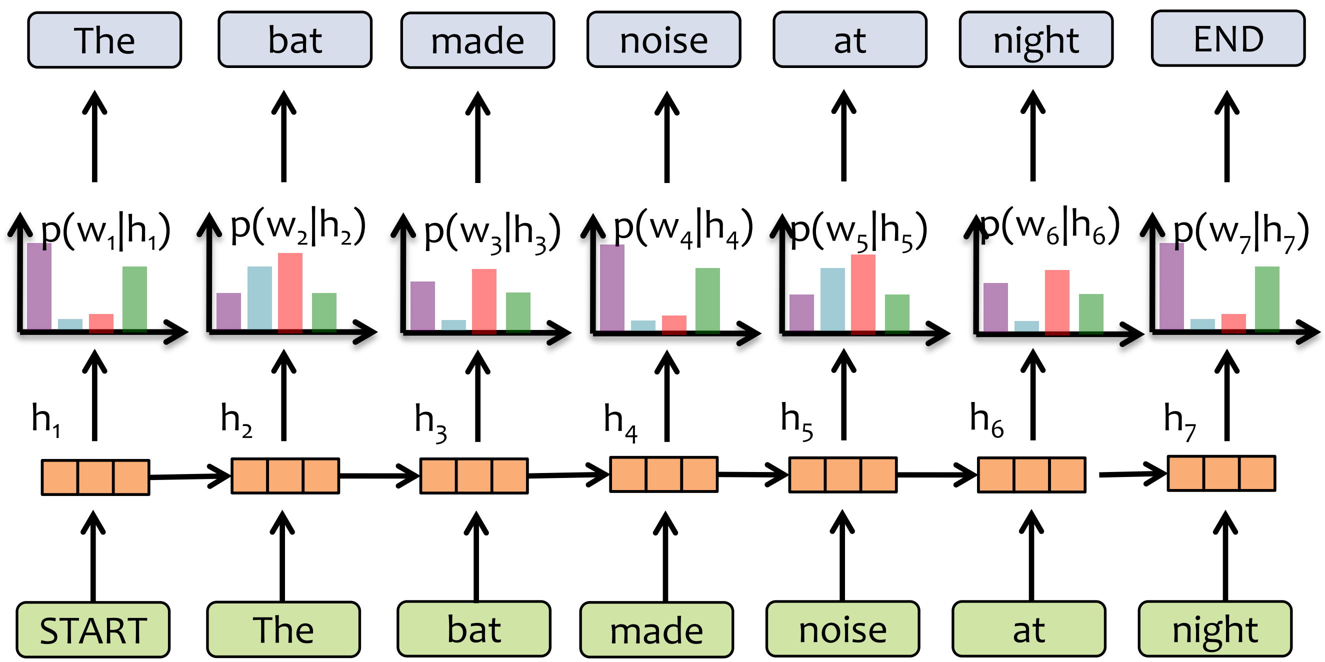
The						$p(w_1)$
The	bat					$p(w_2 \mid f_\theta(w_1))$
The	bat	made				$p(w_3 \mid f_\theta(w_2, w_1))$
The	bat	made	noise			$p(w_4 \mid f_\theta(w_3, w_2, w_1))$
The	bat	made	noise	at		$p(w_5 \mid f_\theta(w_4, w_3, w_2, w_1))$
The	bat	made	noise	at	night	$p(w_6 \mid f_\theta(w_5, w_4, w_3, w_2, w_1))$

Key Idea:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector

RNN Language Model

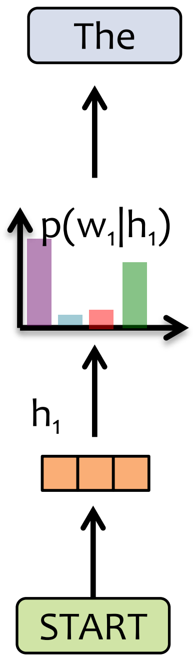


<u>Key Idea</u>:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

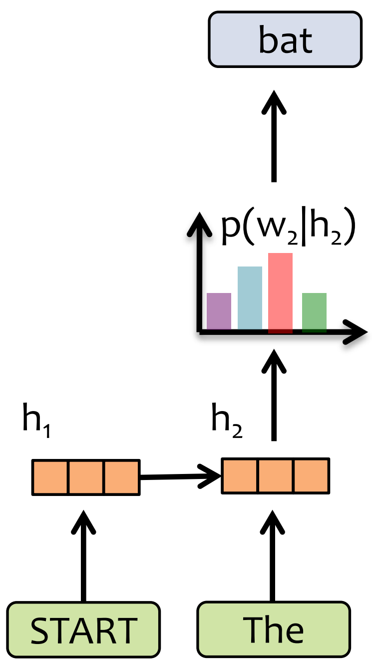


Key Idea:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

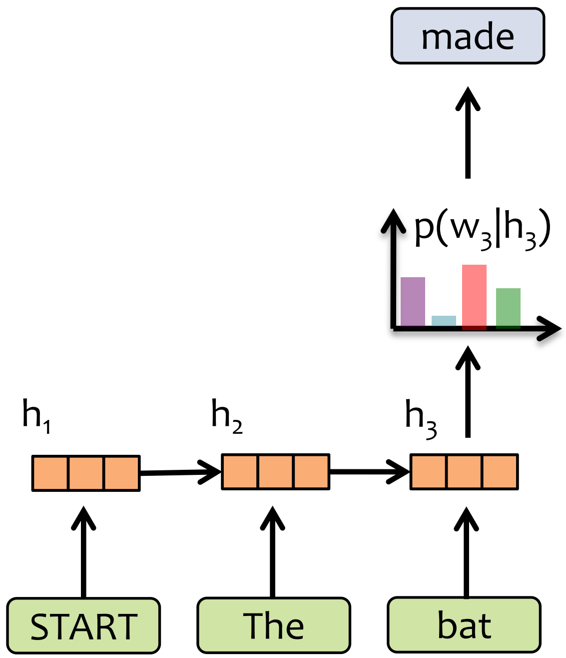


Key Idea:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

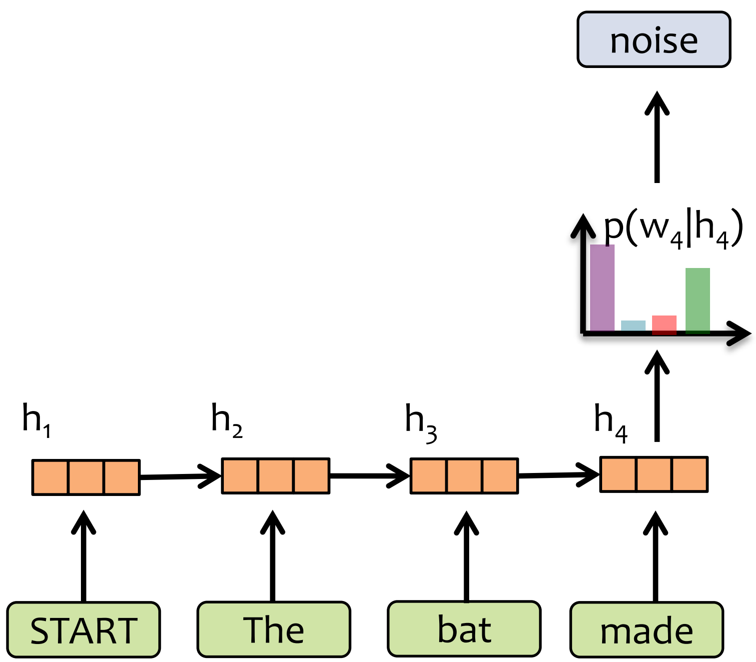


Key Idea:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

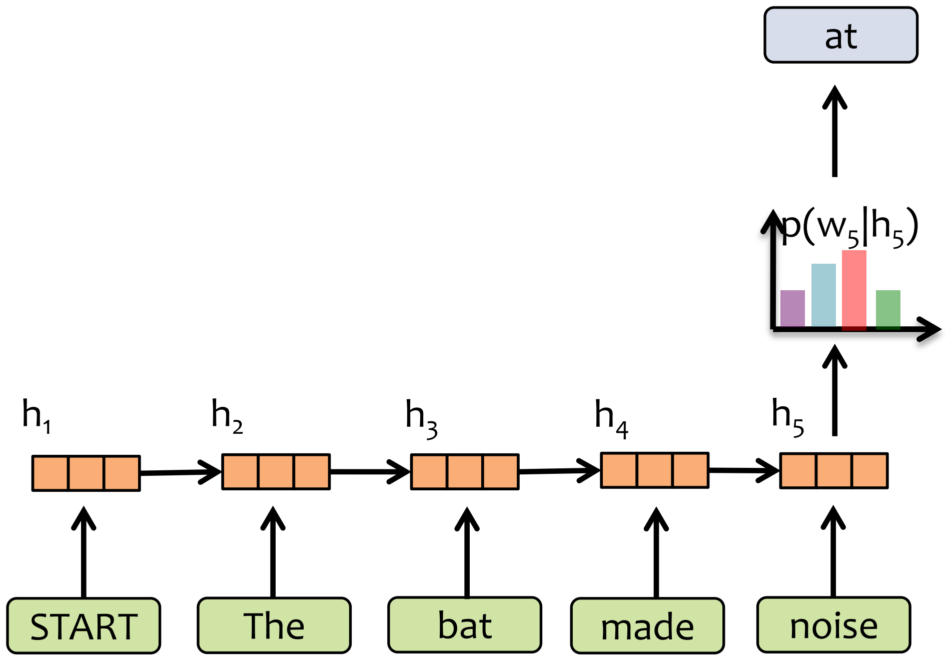


<u>Key Idea</u>:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

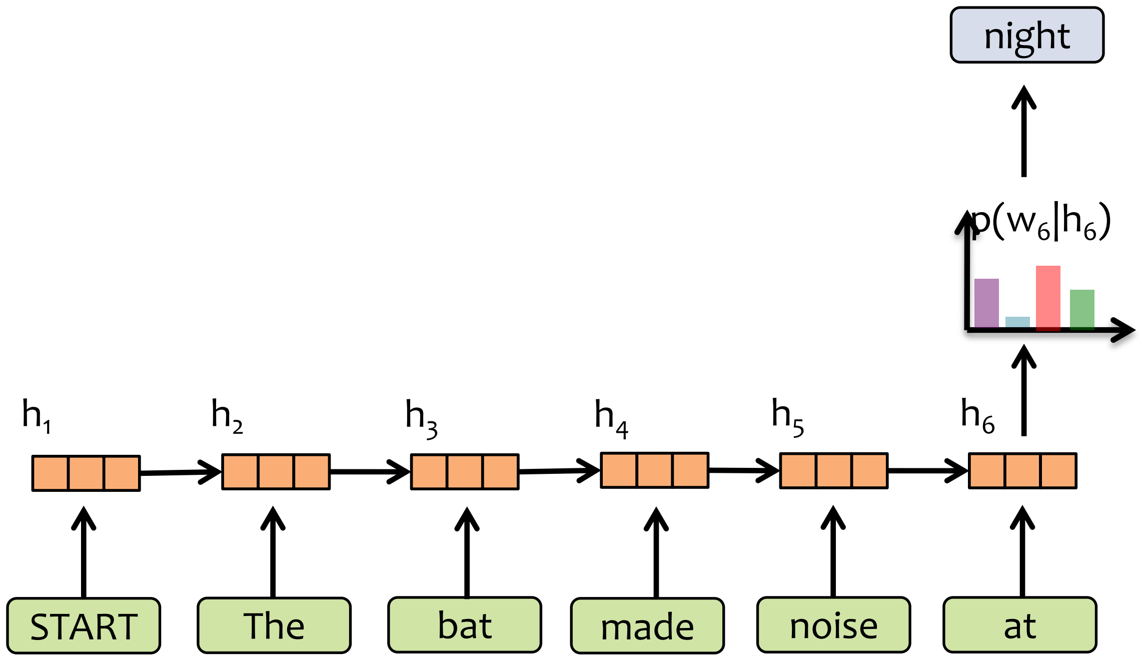


Key Idea:

(1) convert all previous words to a **fixed length vector**

(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model

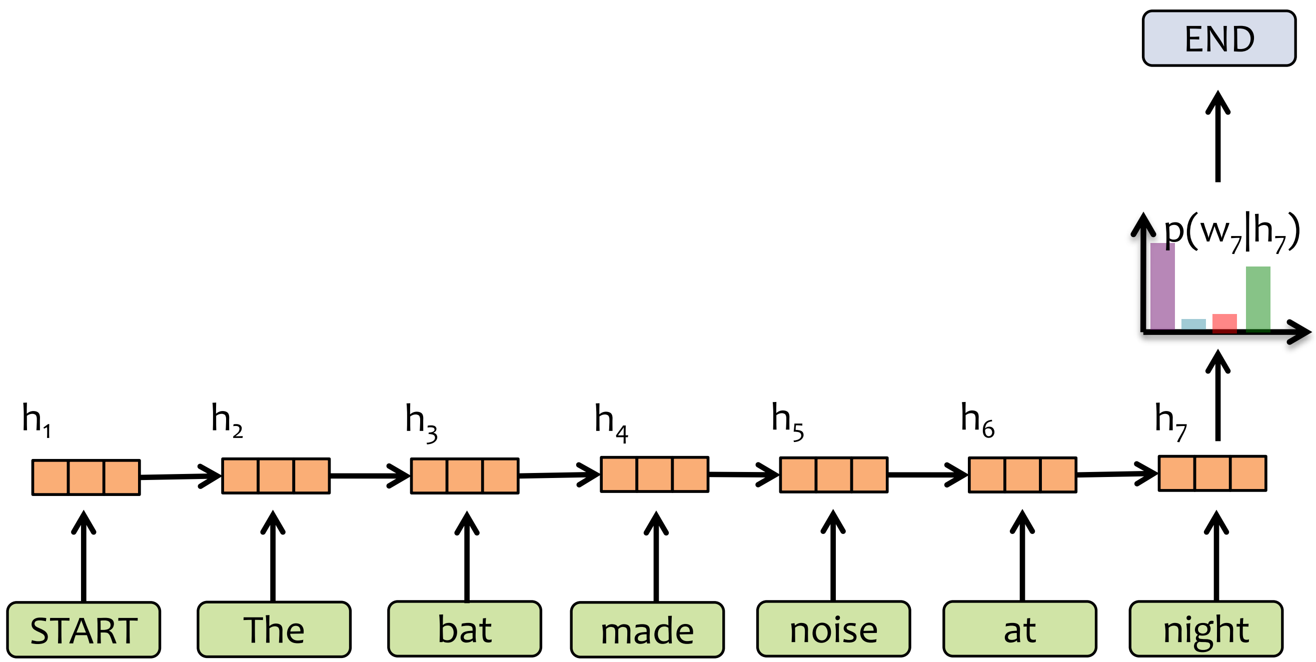


Key Idea:

(1) convert all previous words to a **fixed length vector**

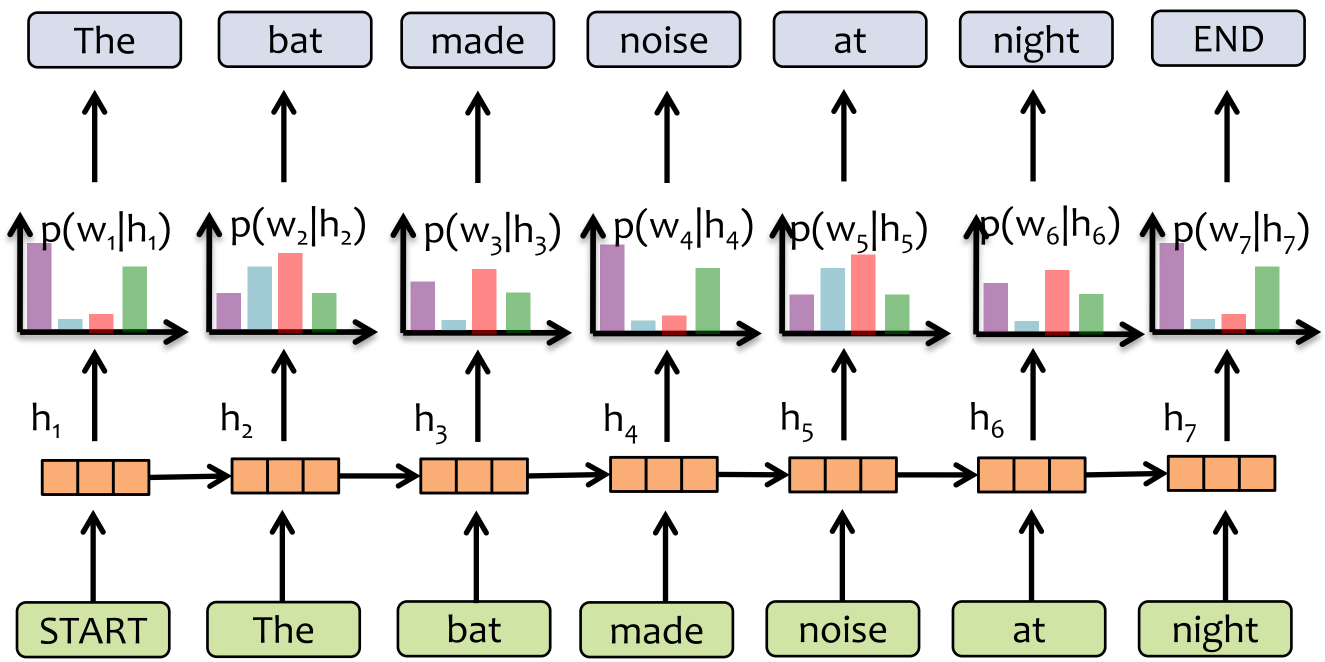
(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model



Key Idea:
(1) convert all previous words to a **fixed length vector**
(2) define distribution $p(w_t \mid f_\theta(w_{t-1}, \ldots, w_1))$ that conditions on the vector $\mathbf{h}_t = f_\theta(w_{t-1}, \ldots, w_1)$

RNN Language Model



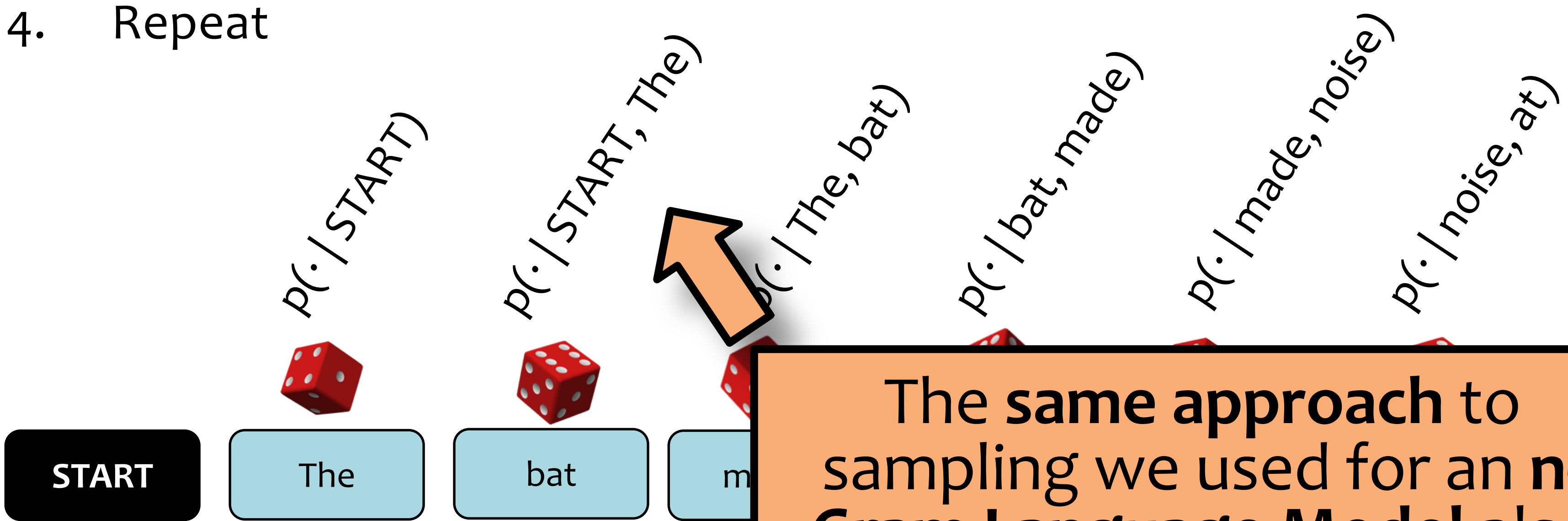
$$p(w_1, w_2, w_3, \ldots, w_T) = p(w_1 \mid h_1)\, p(w_2 \mid h_2) \ldots\, p(w_2 \mid h_T)$$

Sampling from a Language Model

Question: How do we sample from a Language Model?

Answer:

1. Treat each probability distribution like a (50k-sided) weighted die
2. Pick the die corresponding to $p(w_t \mid w_{t-2}, w_{t-1})$
3. Roll that die and generate whichever word w_t lands face up
4. Repeat



The **same approach** to sampling we used for an **n-Gram Language Model** also works here for an **RNN Language Model**

Sampling from an RNN-LM

??

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 a master's ready there My powe so much as hell: Some service i bondman here, Would show hi

KING LEAR: O, if you we 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.

??

CHARLES: Marry, do I, sir; and I came to acquaint you with a matter. I am given, sir, secretly to understand that your younger brother Orlando hath a disposition to come in disguised against me to try a fall. To-morrow, sir, I wrestle for my credit; and he that escapes me without some broken limb shall acquit him is but young and tender; and, uld be loath to foil him, as I honour, if he come in: ny love to you, I came hither to acquaint you wi that either you might stay him from his int ent or brook such disgrace well as he sh n into, in that it is a thing of his own search and altogether against my will.

TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

Which is the real Shakespeare?!

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

Sampling from an RNN-LM

Shakespeare's As You Like 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.

RNN-LM Sample

CHARLES: Marry, do I, sir; and I came to acquaint you with a matter. I am given, sir, secretly to understand that your younger brother Orlando hath a disposition to come in disguised against me to try a fall. To-morrow, sir, I wrestle for my credit; and he that escapes me without some broken limb shall acquit him well. Your brother is but young and tender; and, for your love, I would be loath to foil him, as I must, for my own honour, if he come in: therefore, out of my love to you, I came hither to acquaint you withal, that either you might stay him from his intendment or brook such disgrace well as he shall run into, in that it is a thing of his own search and altogether against my will.

TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

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

Sampling from an RNN-LM

RNN-LM Sample

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.

Shakespeare's As You Like It

CHARLES: Marry, do I, sir; and I came to acquaint you with a matter. I am given, sir, secretly to understand that your younger brother Orlando hath a disposition to come in disguised against me to try a fall. To-morrow, sir, I wrestle for my credit; and he that escapes me without some broken limb shall acquit him well. Your brother is but young and tender; and, for your love, I would be loath to foil him, as I must, for my own honour, if he come in: therefore, out of my love to you, I came hither to acquaint you withal, that either you might stay him from his intendment or brook such disgrace well as he shall run into, in that it is a thing of his own search and altogether against my will.

TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

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

Sampling from an RNN-LM

??

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 a master's ready there My powe so much as hell: Some service i bondman here, Would show hi

KING LEAR: O, if you we 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.

??

CHARLES: Marry, do I, sir; and I came to acquaint you with a matter. I am given, sir, secretly to understand that your younger brother Orlando hath a disposition to come in disguised against me to try a fall. To-morrow, sir, I wrestle for my credit; and he that escapes me without some broken limb shall acquit him is but young and tender; and, uld be loath to foil him, as I honour, if he come in: ny love to you, I came hither to acquaint you wi that either you might stay him from his int ent or brook such disgrace well as he sh un into, in that it is a thing of his own search and altogether against my will.

TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

Which is the real Shakespeare?!

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

SEQUENCE TO SEQUENCE MODELS

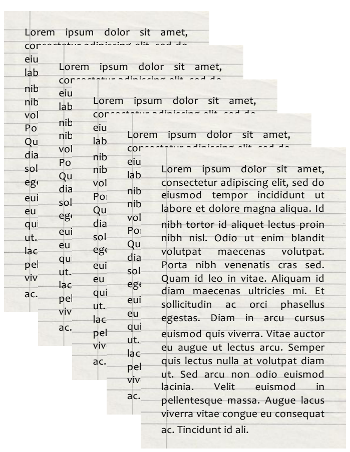
Sequence to Sequence Model

Speech Recognition

Machine Translation

기계 번역은 특히 영어와 한국어와 같은 언어 쌍의 경우 매우 어렵습니다.

Summarization



Sequence to Sequence Model

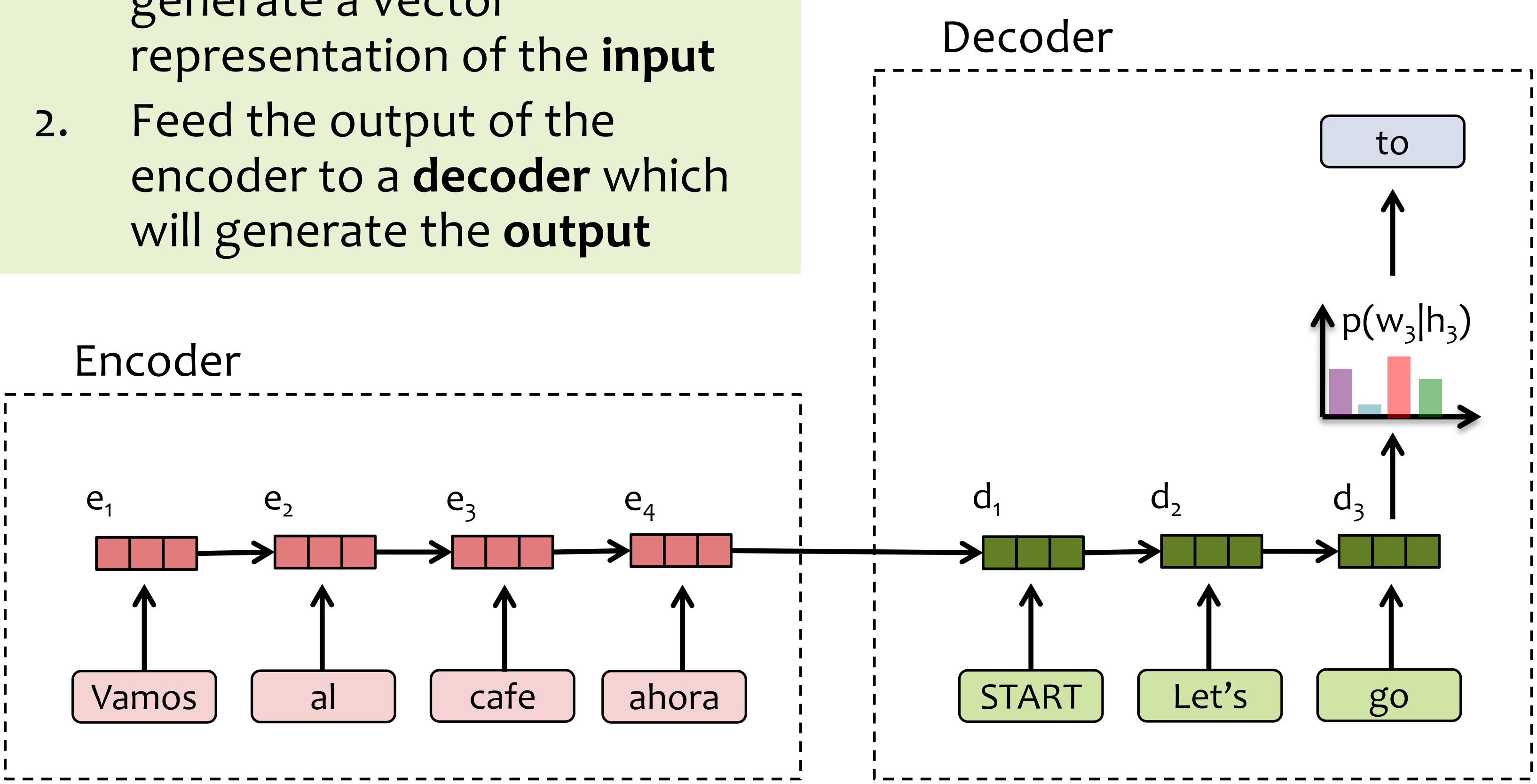
Now suppose you want generate a sequence conditioned on another input

Key Idea:

1. Use an **encoder** model to generate a vector representation of the **input**
2. Feed the output of the encoder to a **decoder** which will generate the **output**

Applications:

- translation:
  Spanish → English
- summarization:
  article → summary
- speech recognition:
  speech signal → transcription



LEARNING THEORY

PAC(-MAN) Learning

For some hypothesis $h \in \mathcal{H}$:

1. True Error

$$R(h)$$

2. Training Error

$$\hat{R}(h)$$

Question 2:
What is the expected number of PAC-MAN levels Matt will complete before a **Game-Over**?

A. 1-10
B. 11-20
C. 21-30

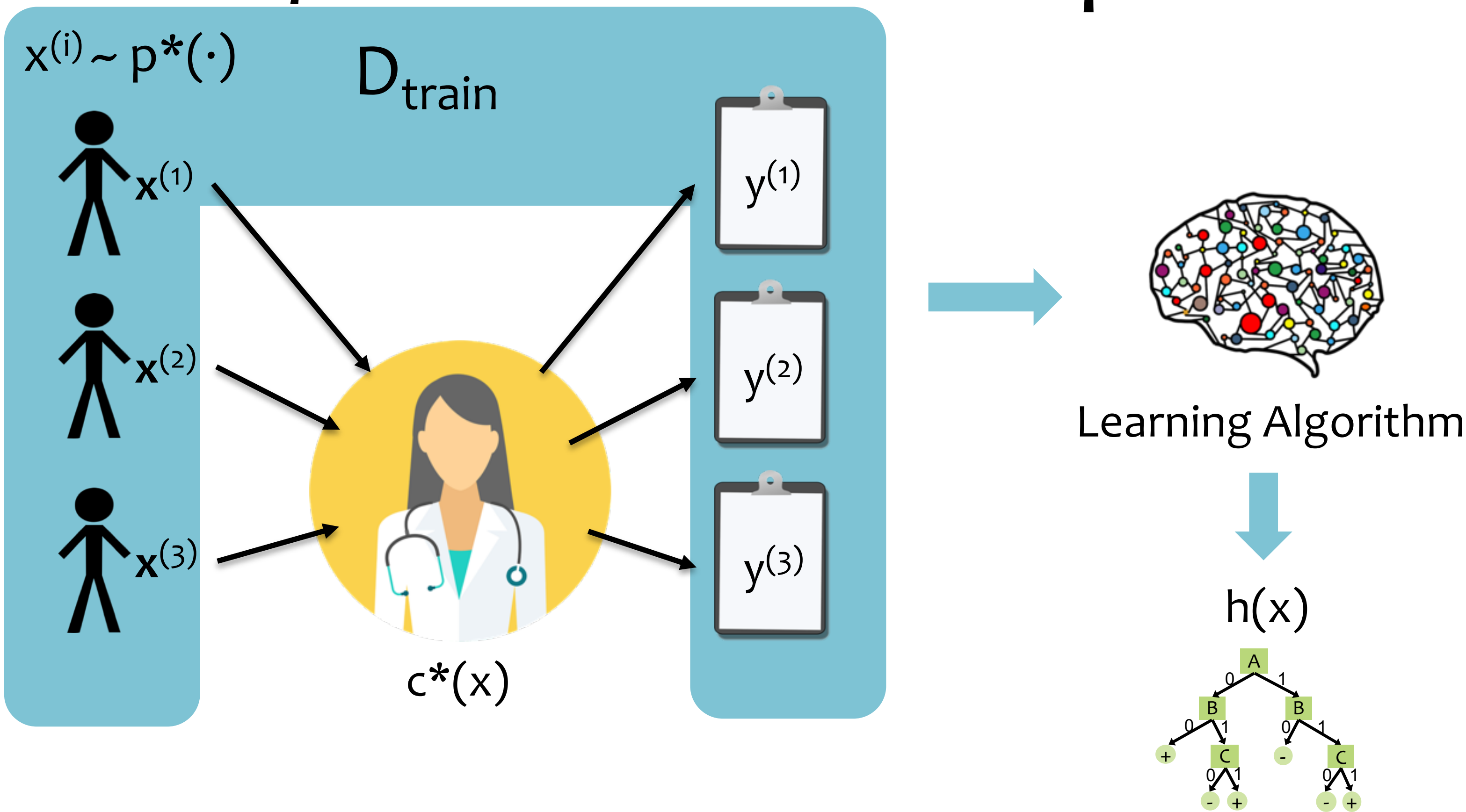
Questions for today (and next lecture)

1. Given a classifier with **zero training error**, what can we say about **true error** (aka. generalization error)?
   (Sample Complexity, Realizable Case)

2. Given a classifier with **low training error**, what can we say about **true error** (aka. generalization error)?
   (Sample Complexity, Agnostic Case)

3. Is there a **theoretical justification for regularization** to avoid overfitting?
   (Structural Risk Minimization)

PAC/SLT Model for Supervised ML



PAC/SLT Model for Supervised ML

- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Distribution over instances, $p^*(\cdot)$
  - Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$ (the doctor's brain)
  - Set, $\mathcal{H}$, of candidate hypothesis functions, $h : \mathcal{X} \rightarrow \mathcal{Y}$ (all possible decision trees)
- **Learner is given** N training examples $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$ where $x^{(i)} \sim p^*(\cdot)$ and $y^{(i)} = c^*(\mathbf{x}^{(i)})$ (history of patients and their diagnoses)
- **Learner produces** a hypothesis function, $\hat{y} = h(x)$, that best approximates unknown target function $y = c^*(x)$ on the training data

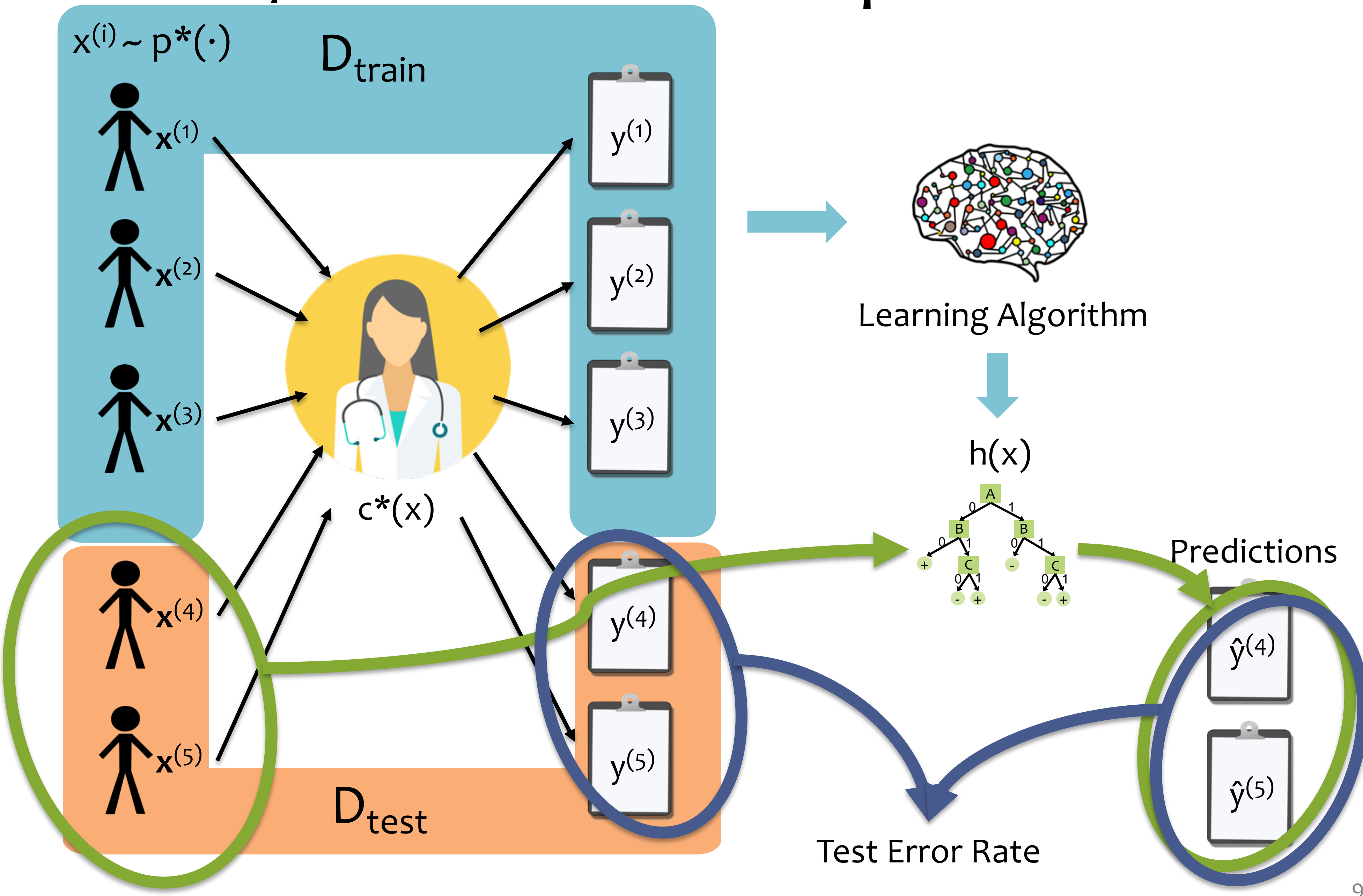
PAC/SLT Model for Supervised ML

- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Distribution over instances, $p^*(\cdot)$
  - Exists an unknown target function, $c^* : \mathcal{X} \to \mathcal{Y}$
    (the doctor's brain
  - Set, $\mathcal{H}$, of candida
    (all possible decisi
- **Learner is given** N t
  $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)},$
  where $x^{(i)} \sim p^*(\cdot)$ an
  (history of patients
- **Learner produces** a
  best approximates
  the training data

Two important settings we'll consider:

1. **Classification:** the possible outputs are **discrete**
2. **Regression:** the possible outputs are **real-valued**

PAC/SLT Model for Supervised ML



Two Types of Error

1. True Error (aka. **expected risk**)

$$R(h) = P_{\mathbf{x} \sim p^*(\mathbf{x})}(c^*(\mathbf{x}) \neq h(\mathbf{x}))$$

This quantity is always **unknown**

2. Train Error (aka. **empirical risk**)

$$\begin{aligned}\hat{R}(h) &= P_{\mathbf{x} \sim \mathcal{S}}(c^*(\mathbf{x}) \neq h(\mathbf{x})) \\ &= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(c^*(\mathbf{x}^{(i)}) \neq h(\mathbf{x}^{(i)})) \\ &= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq h(\mathbf{x}^{(i)}))\end{aligned}$$

We can **measure** this on the training data

where $\mathcal{S} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})\}_{i=1}^{N}$ is the training data set, and $\mathbf{x} \sim \mathcal{S}$ denotes that $\mathbf{x}$ is sampled from the empirical distribution.

PAC / SLT Model

We've also referred to this as the "Function Approximation View"

1. Generate instances from *unknown* distribution p^*

$$\mathbf{x}^{(i)} \sim p^*(\mathbf{x}), \forall i \tag{1}$$

2. Oracle labels each instance with *unknown* function c^*

$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \forall i \tag{2}$$

3. Learning algorithm chooses hypothesis $h \in \mathcal{H}$ with low(est) training error, $\hat{R}(h)$

$$\hat{h} = \operatorname*{argmin}_h \hat{R}(h) \tag{3}$$

4. Goal: Choose an h with low generalization error $R(h)$

Three Hypotheses of Interest

The **true function** c^* is the one we are trying to learn and that labeled the training data:

$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \forall i \tag{1}$$

The **expected risk minimizer** has lowest true error:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\, R(h)$$

Question:
True or False:
h* and c* are always equal.

The **empirical risk minimizer** has lowest training error:

$$\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\, \hat{R}(h) \tag{3}$$

Question 1

A

B

C

PAC LEARNING

Probably Approximately Correct (PAC) Learning

Whiteboard:

- PAC Criterion
- Meaning of "Probably Approximately Correct"
- Def: PAC Learner
- Sample Complexity
- Consistent Learner
- Realizable vs. Agnostic Cases
- Finite vs. Infinite Hypothesis Spaces

SAMPLE COMPLEXITY RESULTS

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

We'll start with the finite case…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$		
Infinite $\|\mathcal{H}\|$		

Probably Approximately Correct (PAC) Learning

Whiteboard:

- Theorem 1: Realizable Case, Finite |H|
- Proof of Theorem 1

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	
Infinite $\|\mathcal{H}\|$		

Example: Conjunctions

Question:

Suppose H = class of conjunctions over **x** in $\{0,1\}^M$

Example hypotheses:

$h(\mathbf{x}) = x_1 (1-x_3) x_5$

$h(\mathbf{x}) = x_1 (1-x_2) x_4 (1-x_5)$

If M = 10, ε = 0.1, δ = 0.01, how many examples suffice according to Theorem 1?

Answer:

A. 10*(2*ln(10)+ln(100)) ≈ 92
B. 10*(3*ln(10)+ln(100)) ≈ 116
C. 10*(10*ln(2)+ln(100)) ≈ 116
D. 10*(10*ln(3)+ln(100)) ≈ 156
E. 100*(2*ln(10)+ln(10)) ≈ 691
F. 100*(3*ln(10)+ln(10)) ≈ 922
G. 100*(10*ln(2)+ln(10)) ≈ 924
H. 100*(10*ln(3)+ln(10)) ≈ 1329

Thm. 1 $N \geq \frac{1}{\epsilon}\left[\log(|\mathcal{H}|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.

Question 2

- A
- B
- C
- D
- E
- F
- G
- H

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\|\mathcal{H}\|) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.
Infinite $\|\mathcal{H}\|$		

1. Bound is **inversely linear in epsilon** (e.g. halving the error requires double the examples)
2. Bound is **only logarithmic in |H|** (e.g. quadrupling the hypothesis space only requires double the examples)

1. Bound is **inversely quadratic in epsilon** (e.g. halving the error requires 4x the examples)
2. Bound is **only logarithmic in |H|** (i.e. same as Realizable case)

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient s… probability $(1-\delta)$ all $h \in \mathcal{H}$ with… have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ …fficient so that …or all $h \in \mathcal{H}$ we … $\leq \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

We need a new definition of "complexity" for a Hypothesis space for these results (see *VC Dimension*)

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\|\mathcal{H}\|) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.
Infinite $\|\mathcal{H}\|$	**Thm. 3** $N = O(\frac{1}{\epsilon}\left[\mathrm{VC}(\mathcal{H})\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 4** $N = O(\frac{1}{\epsilon^2}\left[\mathrm{VC}(\mathcal{H}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.