



10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

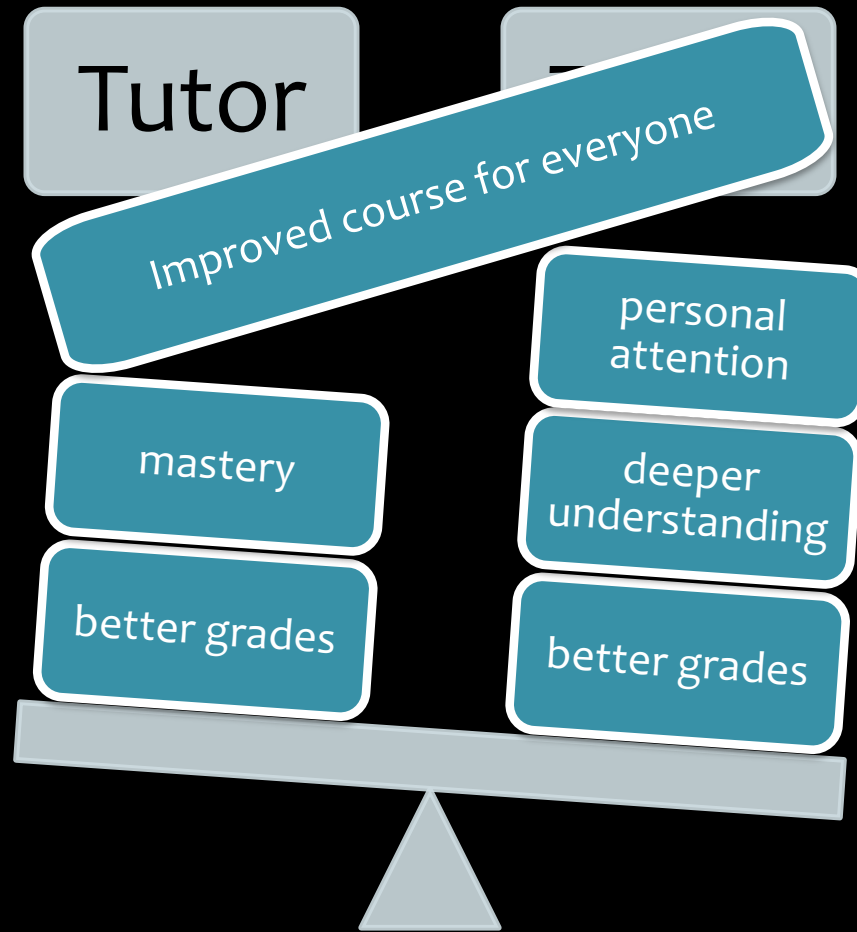
Deep Learning: RNNs & CNNs

Matt Gormley
Lecture 14
Oct. 12, 2022

Reminders

- **Post-Exam Followup:**
 - Exam Viewing
 - Exit Poll: Exam 1
 - Grade Summary 1
- **Homework 4: Logistic Regression**
 - Out: Tue, Oct 4
 - Due: Thu, Oct 13 at 11:59pm
- **Homework 5: Neural Networks**
 - Out: Thu, Oct 13
 - Due: Thu, Oct 27 at 11:59pm

Peer Tutoring

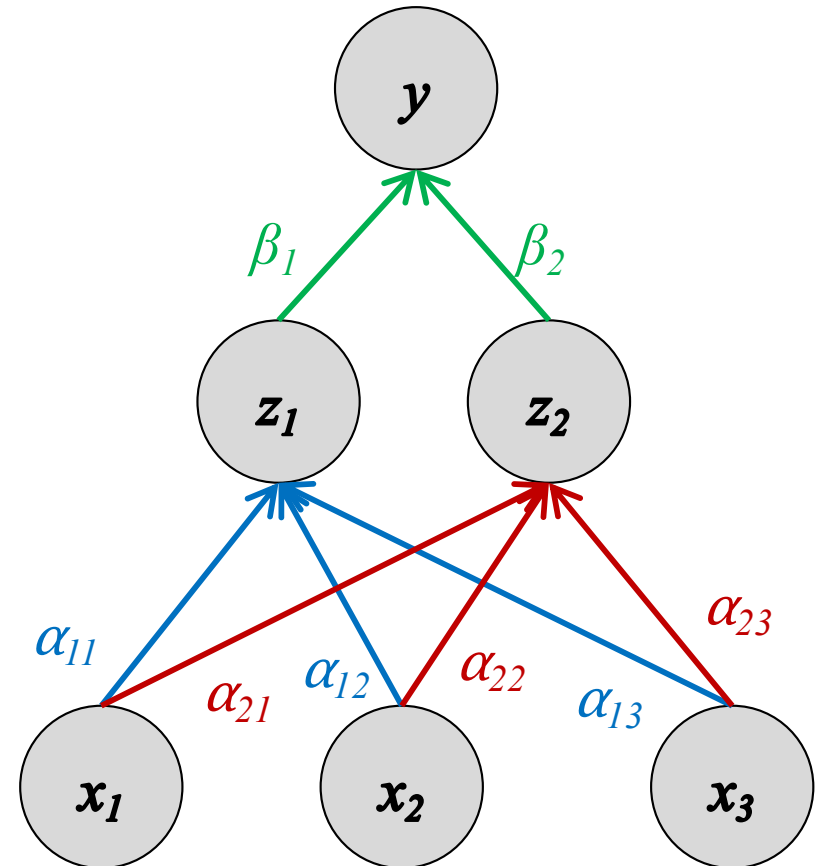


DRAWING A NEURAL NETWORK

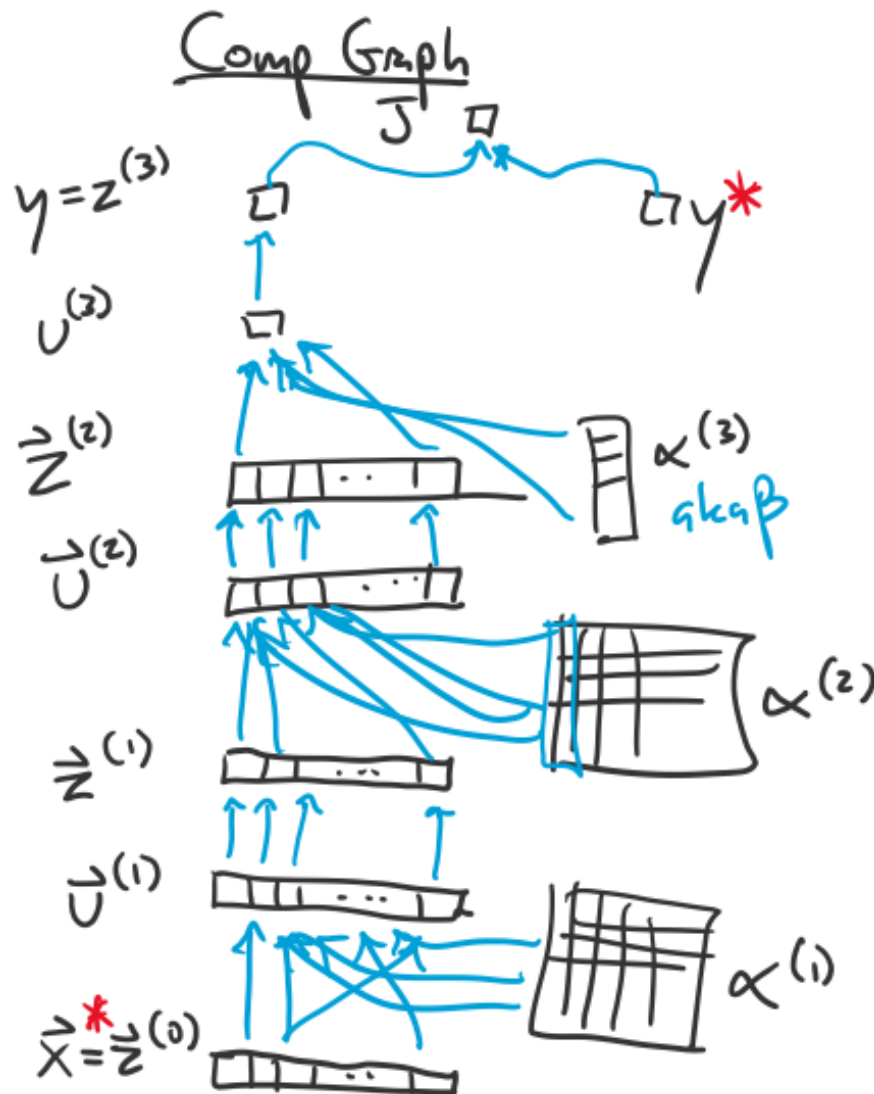
Ways of Drawing Neural Networks

Neural Network Diagram

- The diagram represents a neural network
- Nodes are **circles**
- One node per **hidden unit**
- Node is labeled with the **variable** corresponding to the hidden unit
- For a fully connected feed-forward neural network, a hidden unit is a nonlinear function of nodes in the previous layer
- *Edges are directed*
- Each **edge is labeled with its weight** (side note: we should be careful about ascribing how a matrix can be used to indicate the labels of the edges and pitfalls there)
- Other details:
 - Following standard convention, the **intercept term is NOT shown** as a node, but rather is assumed to be part of the non-linear function that yields a hidden unit. (i.e. its weight does NOT appear in the picture anywhere)
 - The diagram does **NOT include any nodes related to the loss computation**



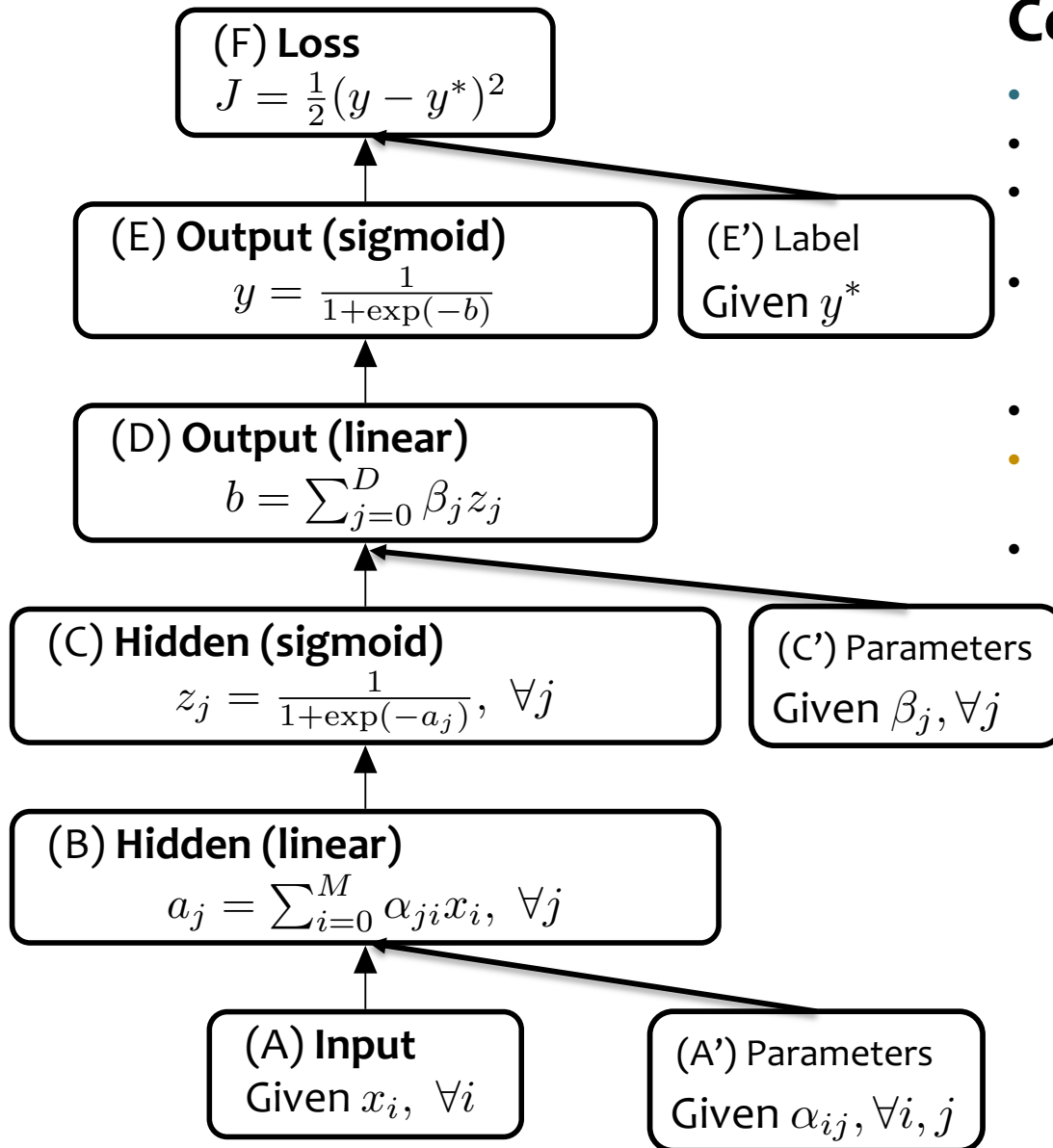
Ways of Drawing Neural Networks



Computation Graph

- The diagram represents an algorithm
- Nodes are **rectangles**
- One node per **intermediate variable in the algorithm**
- Node is labeled with the **function** that it computes (inside the box) and also the **variable name** (outside the box)
- Edges are directed
- Edges do not have labels (since they don't need them)
- For neural networks:
 - Each **intercept term should appear as a node** (if it's not folded in somewhere)
 - Each parameter should appear as a node
 - Each constant, e.g. a true label or a feature vector should appear in the graph
 - It's perfectly fine to include the loss

Ways of Drawing Neural Networks



Computation Graph

- The diagram represents an algorithm
- Nodes are **rectangles**
- One node per **intermediate variable in the algorithm**
- Node is labeled with the **function** that it computes (inside the box) and also the **variable** name (outside the box)
- Edges are directed
- Edges do not have labels (since they don't need them)
- For neural networks:
 - Each **intercept term** should appear as a node (if it's not folded in somewhere)
 - Each parameter should appear as a node
 - Each constant, e.g. a true label or a feature vector should appear in the graph
 - It's perfectly fine to include the loss

Ways of Drawing Neural Networks

Neural Network Diagram

- The diagram represents a neural network
- Nodes are **circles**
- One node per **hidden unit**
- Node is labeled with the **variable** corresponding to the hidden unit
- For a fully connected feed-forward neural network, a hidden unit is a nonlinear function of nodes in the previous layer
- *Edges are directed*
- Each **edge is labeled with its weight** (side note: we should be careful about ascribing how a matrix can be used to indicate the labels of the edges and pitfalls there)
- Other details:
 - Following standard convention, the **intercept term is NOT shown** as a node, but rather is assumed to be part of the non-linear function that yields a hidden unit. (i.e. its weight does NOT appear in the picture anywhere)
 - The diagram does **NOT include any nodes related to the loss computation**

Computation Graph

- The diagram represents an algorithm
- Nodes are **rectangles**
- One node per **intermediate variable in the algorithm**
- Node is labeled with the **function** that it computes (inside the box) and also the **variable** name (outside the box)
- *Edges are directed*
- **Edges do not have labels** (since they don't need them)
- For neural networks:
 - Each **intercept term should appear as a node** (if it's not folded in somewhere)
 - Each parameter should appear as a node
 - Each constant, e.g. a true label or a feature vector should appear in the graph
 - It's **perfectly fine to include the loss**

Important!

Some of these conventions are specific to 10-301/601. The literature abounds with variations on these conventions, but it's helpful to have some distinction nonetheless.

Summary

1. Neural Networks...

- provide a way of learning features
- are highly nonlinear prediction functions
- (can be) a highly parallel network of logistic regression classifiers
- discover useful hidden representations of the input

2. Backpropagation...

- provides an efficient way to compute gradients
- is a special case of reverse-mode automatic differentiation

Backprop Objectives

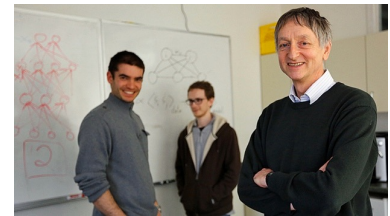
You should be able to...

- Differentiate between a neural network diagram and a computation graph
- Construct a computation graph for a function as specified by an algorithm
- Carry out the backpropagation on an arbitrary computation graph
- Construct a computation graph for a neural network, identifying all the given and intermediate quantities that are relevant
- Instantiate the backpropagation algorithm for a neural network
- Instantiate an optimization method (e.g. SGD) and a regularizer (e.g. L2) when the parameters of a model are comprised of several matrices corresponding to different layers of a neural network
- Apply the empirical risk minimization framework to learn a neural network
- Use the finite difference method to evaluate the gradient of a function
- Identify when the gradient of a function can be computed at all and when it can be computed efficiently
- Employ basic matrix calculus to compute vector/matrix/tensor derivatives.

DEEP LEARNING

Why is everyone talking about Deep Learning?

- Because a lot of money is invested in it...
 - DeepMind: Acquired by Google for **\$400 million**
 - Deep Learning startups command **millions of VC dollars**
 - Demand for deep learning engineers continually outpaces supply
- Because it made the **front page** of the New York Times



The New York Times

Why is everyone talking about Deep Learning?

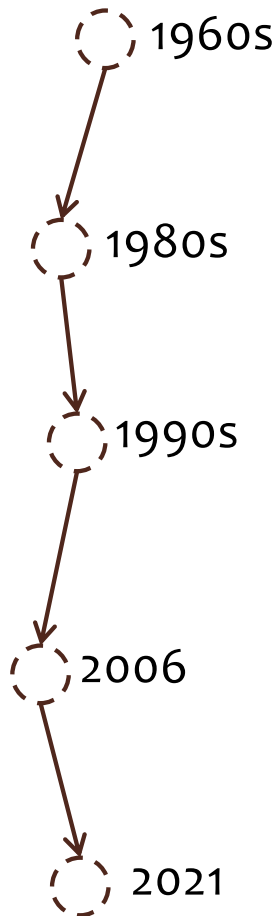
Deep learning:

- Has won numerous pattern recognition competitions
- Does so with minimal feature engineering

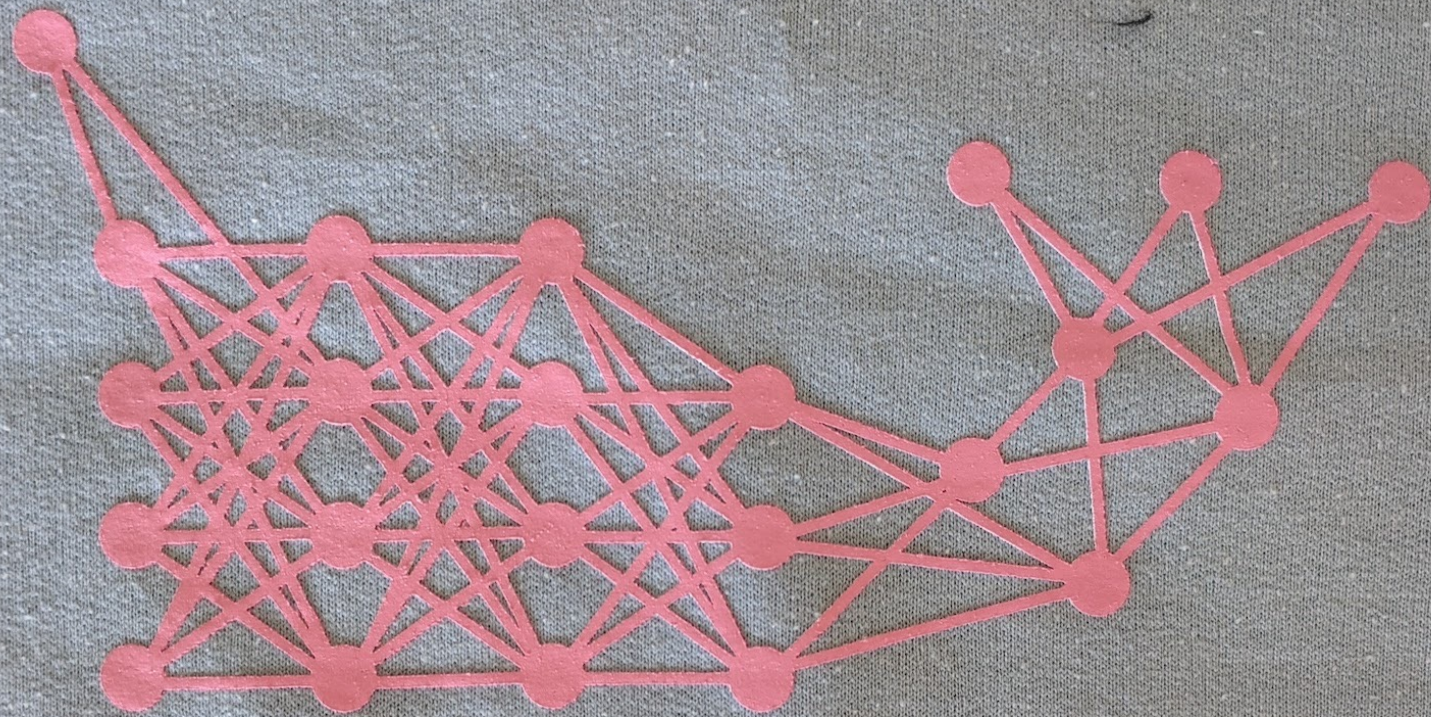
This wasn't always the case!

Since 1980s: Form of models hasn't changed much, but lots of new tricks...

- More hidden units
- Better (online) optimization
- New nonlinear functions (ReLUs)
- Faster computers (CPUs and GPUs)



FIRST EXAMPLE OF A DEEP NETWORK

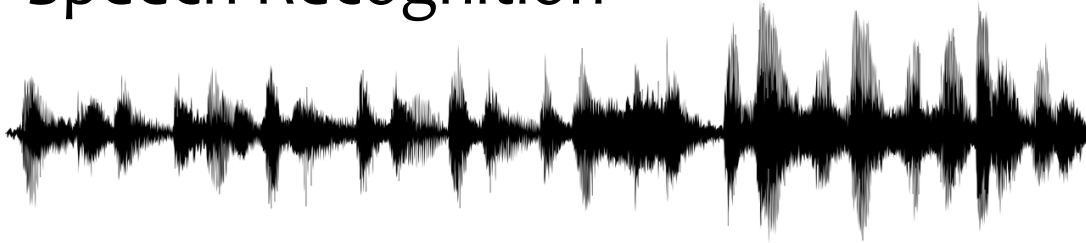


10-601 course staff

BACKGROUND: HUMAN LANGUAGE TECHNOLOGIES

Human Language Technologies

Speech Recognition



Machine Translation

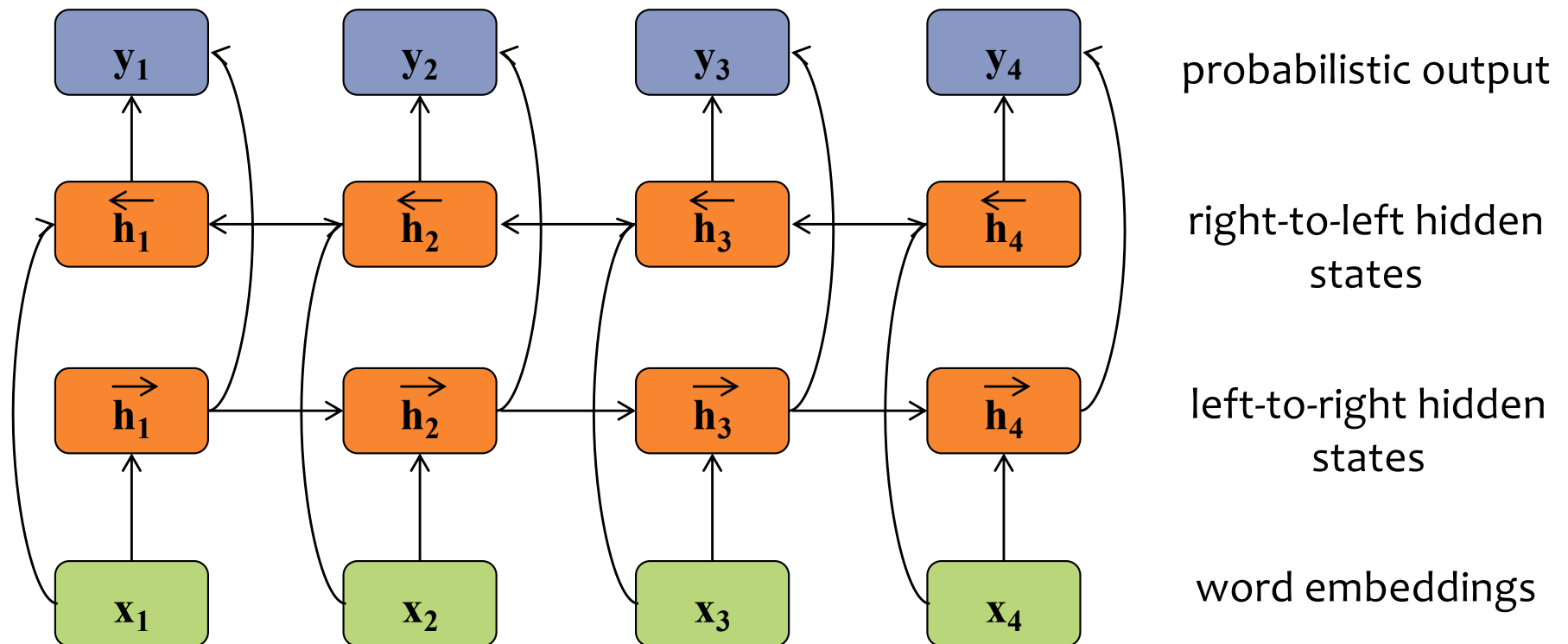
기계 번역은 특히 영어와 한국어와 같은 언어 쌍의 경우 매우 어렵습니다.

Summarization

Lorem ipsum dolor sit amet,
 coi
 elu
 lab Lorem ipsum dolor sit amet,
 coi
 nib
 nib elu Lorem ipsum dolor sit amet,
 vol nib coi
 Po nib elu Lorem ipsum dolor sit amet,
 Qu vol lab coi
 dia Po nib elu Lorem ipsum dolor sit amet,
 sol Qu vol lab consectetur adipiscing elit, sed do
 eui dia Po nib eiusmod tempor incididunt ut
 eu egr Qu vol labore et dolore magna aliqua. Id
 qu eui dia Po nibh tortor id aliquet lectus proin
 ut. eu sol Qu nibh nisi. Odio ut enim blandit
 lac qui egr dia volutpat maecenas volutpat.
 pei ut. eui sol Porta nibh venenatis cras sed.
 viv ut. eu sol Quam id leo in vitae. Aliquam id
 lac qui eui diam maecenas ultricies mi. Et
 ac. pei ut. eu sollicitudin ac orci phasellus
 viv lac qu egestas. Diam in arcu cursus
 ac. pei ut. euismod quis viverra. Vitae auctor
 viv lac eu augue ut lectus arcu. Semper
 ac. pei quis lectus nulla at volutpat diam
 viv ut. Sed arcu non odio euismod
 lacinia. Velit euismod in
 ac. pellentesque massa. Augue lacus
 viverra vitae congue eu consequat
 ac. Tincidunt id ali.

Bidirectional RNN

RNNs are a now commonplace backbone in deep learning approaches to natural language processing



BACKGROUND: COMPUTER VISION

Example: Image Classification

- ImageNet LSVRC-2011 contest:
 - **Dataset:** 1.2 million labeled images, 1000 classes
 - **Task:** Given a new image, label it with the correct class
 - **Multiclass** classification problem
- Examples from <http://image-net.org/>

Bird

Warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings

2126
pictures

92.85%
Popularity
Percentile



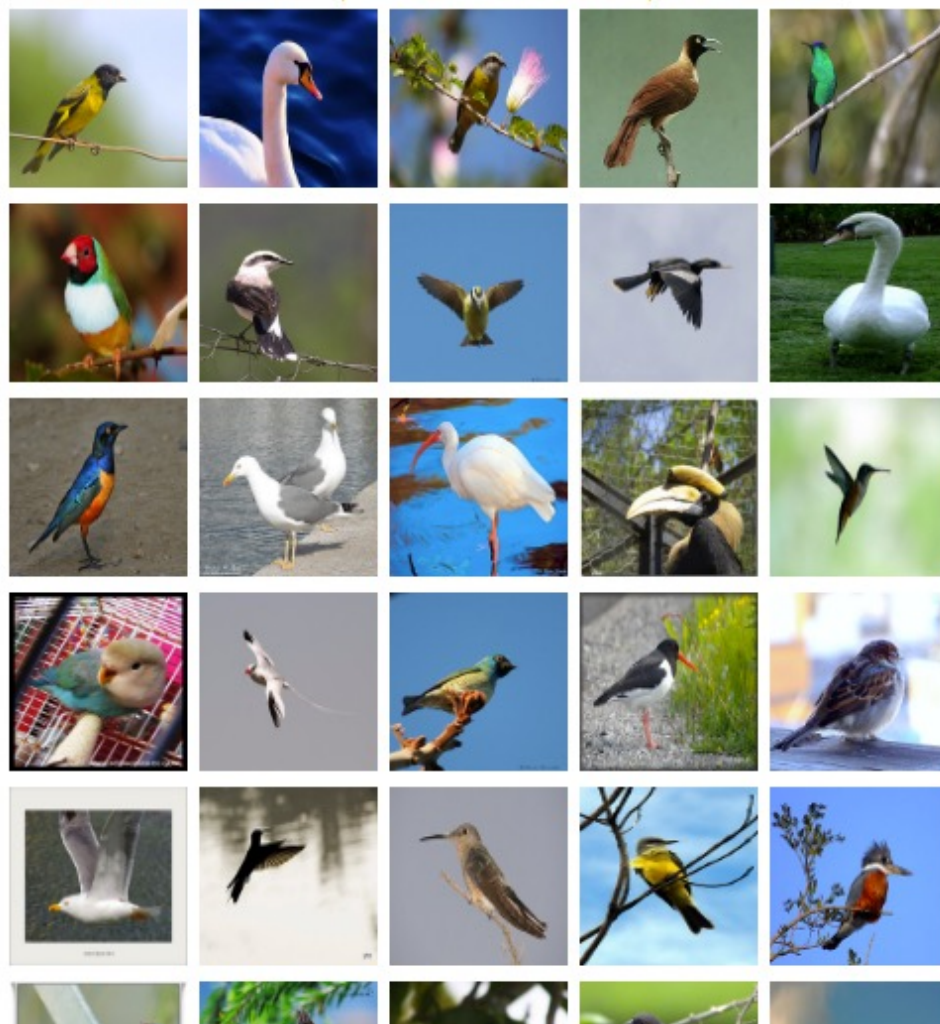
Wordnet
IDs

- marine animal, marine creature, sea animal, sea creature (1)
- scavenger (1)
- biped (0)
- predator, predatory animal (1)
- larva (49)
- acrodont (0)
- feeder (0)
- stunt (0)
- chordate (3087)**
 - tunicate, urochordate, urochord (6)
 - cephalochordate (1)
 - vertebrate, craniate (3077)**
 - mammal, mammalian (1169)
 - bird (871)**
 - dickeybird, dickey-bird, dickybird, dicky-bird (0)
 - cock (1)
 - hen (0)
 - nester (0)
 - night bird (1)
 - bird of passage (0)
 - protoavis (0)
 - archaeopteryx, archeopteryx, Archaeopteryx lithographi Sinornis (0)
 - lbero-mesornis (0)
 - archaeornis (0)
 - ratite, ratite bird, flightless bird (10)
 - carinate, carinate bird, flying bird (0)
 - passerine, passeriform bird (279)
 - nonpasserine bird (0)
 - bird of prey, raptor, raptorial bird (80)
 - gallinaceous bird, gallinacean (114)

Treemap Visualization

Images of the Synset

Downloads



German iris, *Iris kochii*

Iris of northern Italy having deep blue-purple flowers; similar to but smaller than *Iris germanica*

469 pictures

49.6% Popularity Percentile



Wordnet IDs

Treemap Visualization

Images of the Synset

Downloads



- halophyte (0)
- succulent (39)
- cultivar (0)
- cultivated plant (0)
- weed (54)
- evergreen, evergreen plant (0)
- deciduous plant (0)
- vine (272)
- creeper (0)
- woody plant, ligneous plant (1868)
- geophyte (0)
- desert plant, xerophyte, xerophytic plant, xerophile, xerophilic
- mesophyte, mesophytic plant (0)
- aquatic plant, water plant, hydrophyte, hydrophytic plant (11)
- tuberous plant (0)
- bulbous plant (179)
 - iridaceous plant (27)
 - iris, flag, fleur-de-lis, sword lily (19)
 - bearded iris (4)
 - Florentine iris, orris, *Iris germanica florentina*, Iris
 - German iris, *Iris germanica* (0)
 - German iris, *Iris kochii* (0)
 - Dalmatian iris, *Iris pallida* (0)
 - beardless iris (4)
 - bulbous iris (0)
 - dwarf iris, *Iris cristata* (0)
 - stinking iris, gladdon, gladdon iris, stinking gladwyn, Persian iris, *Iris persica* (0)
 - yellow iris, yellow flag, yellow water flag, *Iris pseudo*
 - dwarf iris, vernal iris, *Iris verna* (0)
 - blue flag, *Iris versicolor* (0)

Court, courtyard

An area wholly or partly surrounded by walls or buildings; "the house was built around an inner court"

165 pictures

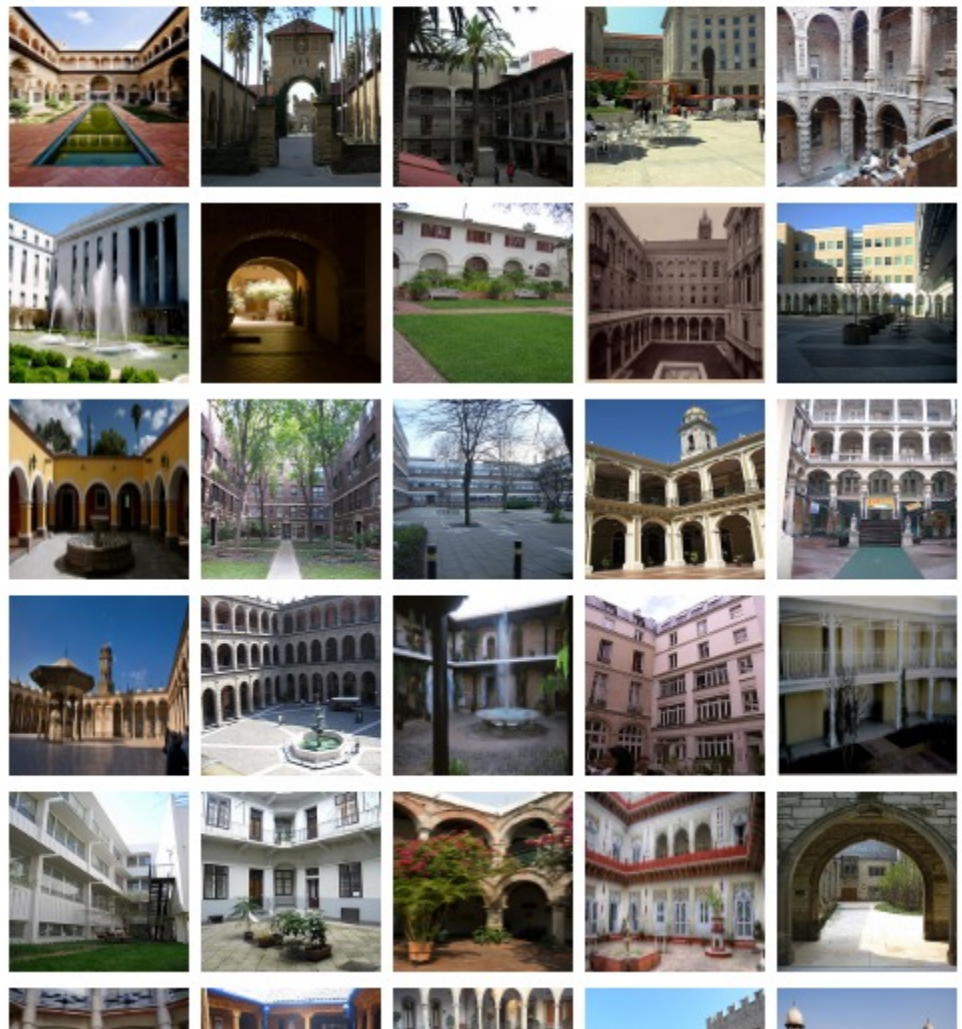
92.61% Popularity Percentile

Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
 - plant, flora, plant life (4486)
 - geological formation, formation (175)
 - natural object (1112)
 - sport, athletics (176)
 - artifact, artefact (10504)
 - instrumentality, instrumentation (5494)
 - structure, construction (1405)
 - airdock, hangar, repair shed (0)
 - altar (1)
 - arcade, colonnade (1)
 - arch (31)
 - area (344)
 - aisle (0)
 - auditorium (1)
 - baggage claim (0)
 - box (1)
 - breakfast area, breakfast nook (0)
 - bullpen (0)
 - chancel, sanctuary, bema (0)
 - choir (0)
 - corner, nook (2)
 - court, courtyard (6)
 - atrium (0)
 - bailey (0)
 - cloister (0)
 - food court (0)
 - forecourt (0)
 - narvis (0)

Treemap Visualization Images of the Synset Downloads

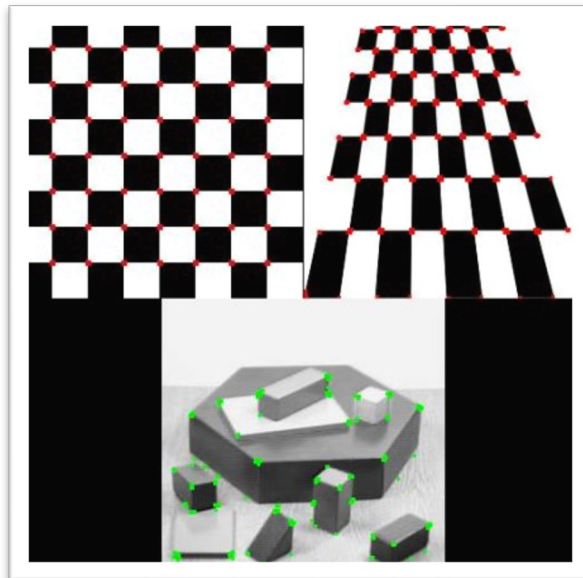


Feature Engineering for CV

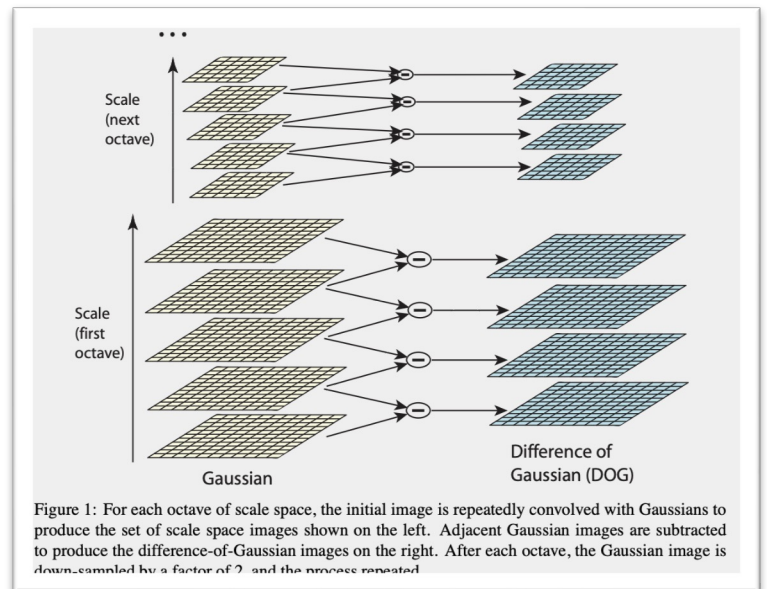
Edge detection (Canny)



Corner Detection (Harris)



Scale Invariant Feature Transform (SIFT)



Example: Image Classification

CNN for Image Classification

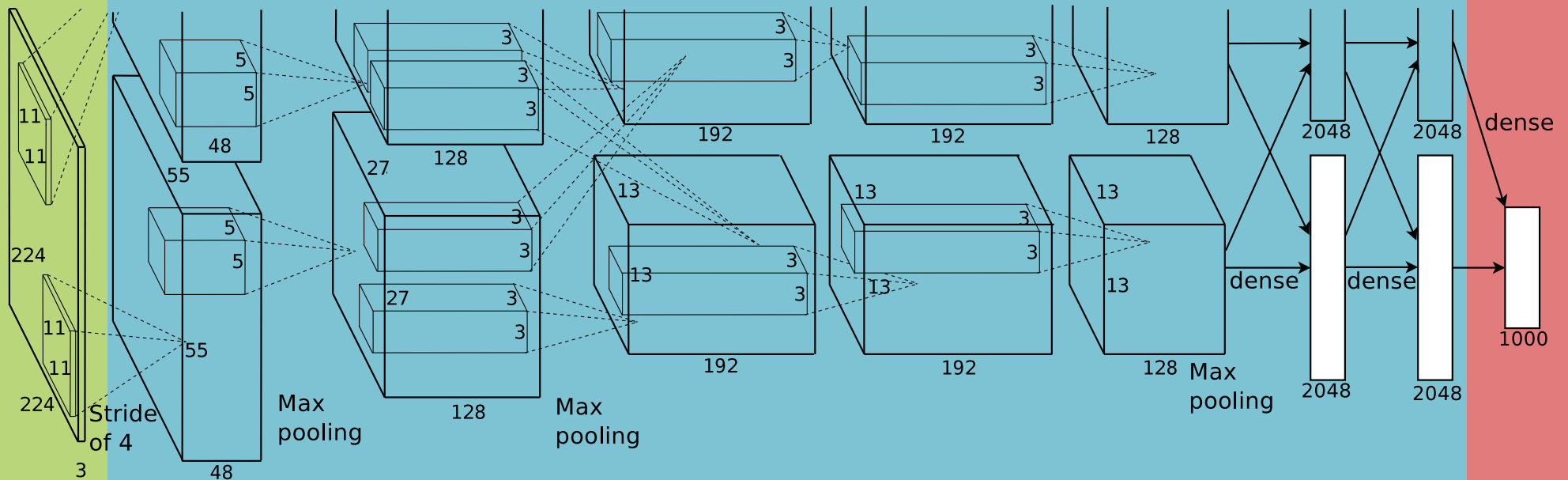
(Krizhevsky, Sutskever & Hinton, 2012)

15.3% error on ImageNet LSVRC-2012 contest

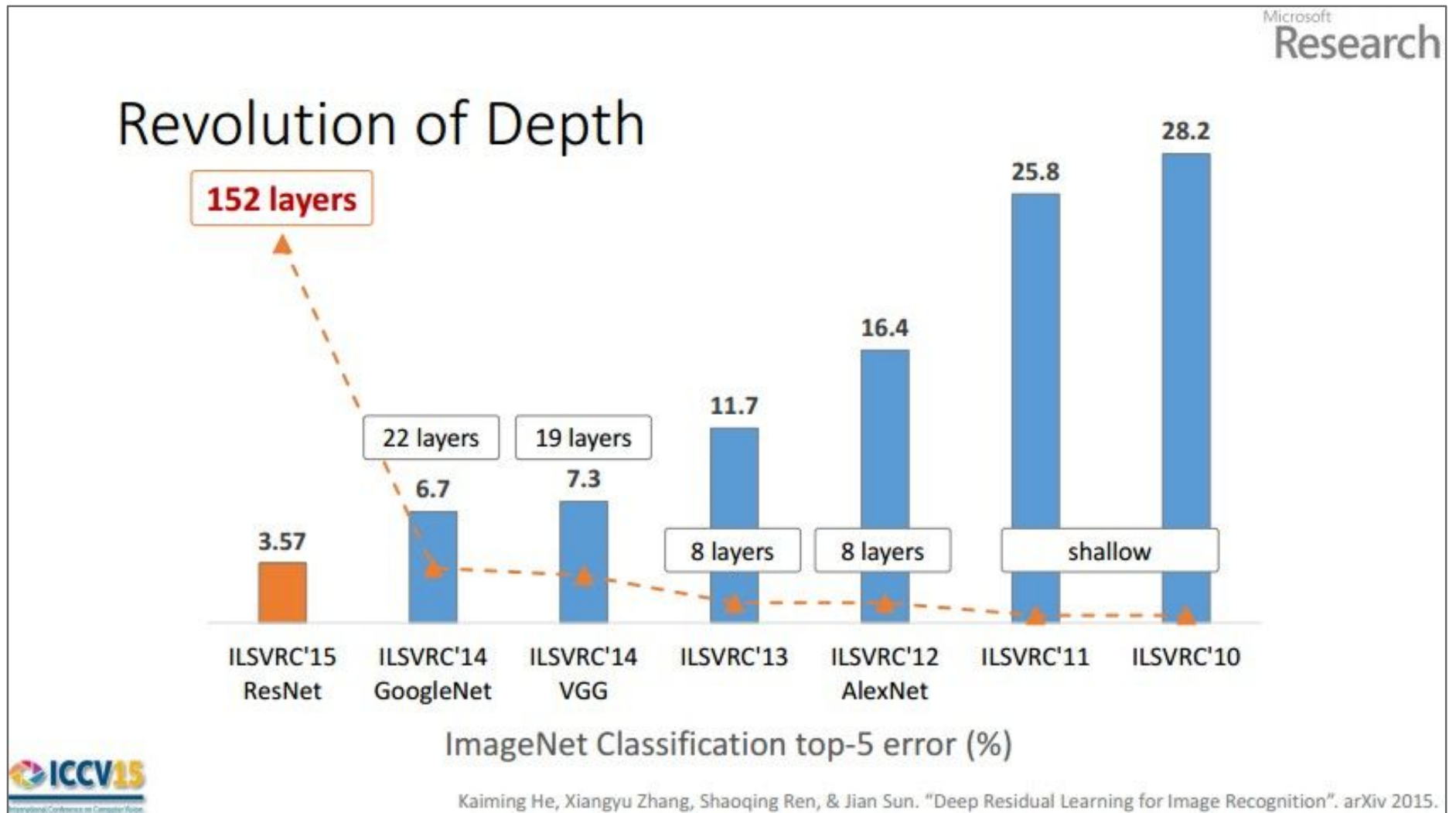
Input image (pixels)

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

1000-way softmax



CNNs for Image Recognition



Backpropagation and Deep Learning

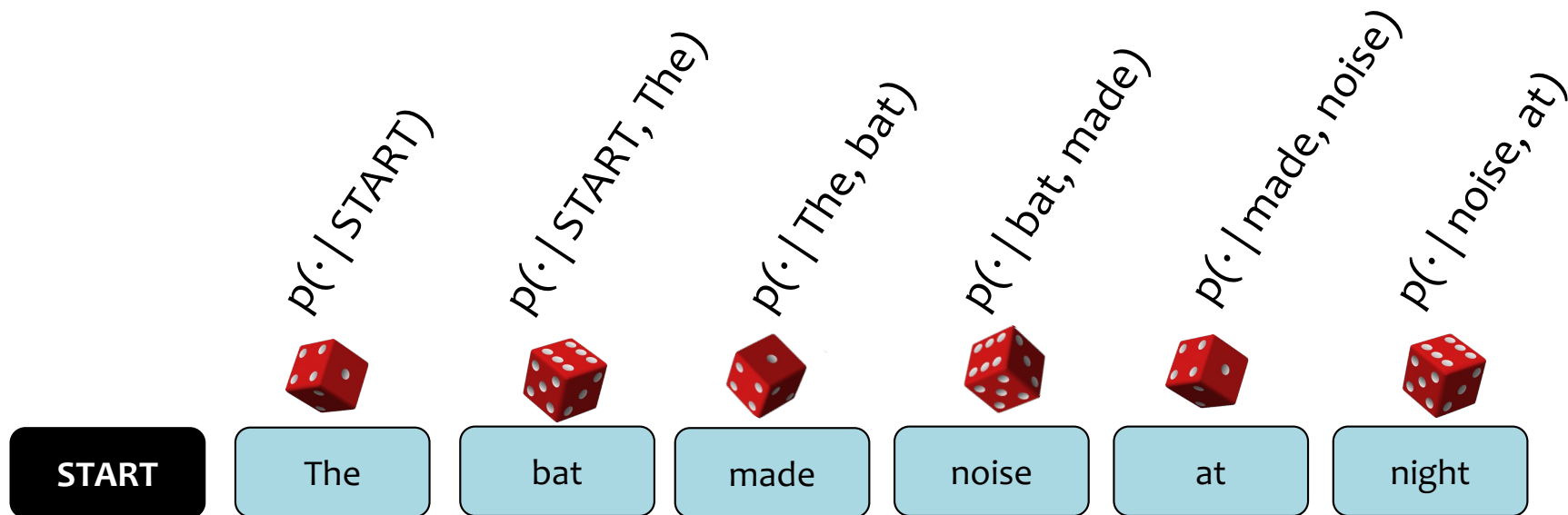
Convolutional neural networks (CNNs) and **recurrent neural networks (RNNs)** are simply fancy computation graphs (aka. hypotheses or decision functions).

Our recipe also applies to these models and (again) relies on the **backpropagation algorithm** to compute the necessary gradients.

BACKGROUND: N-GRAM LANGUAGE MODELS

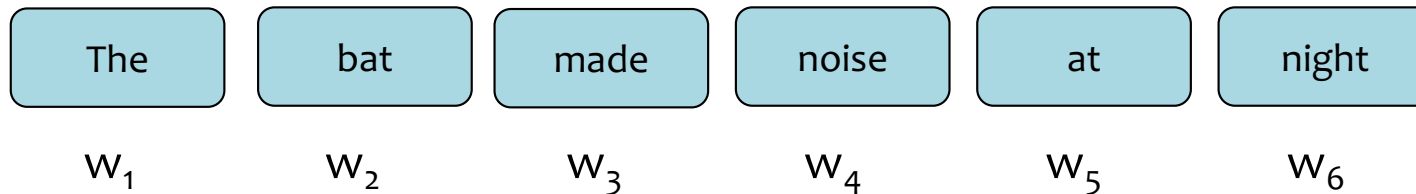
n-Gram Language Model

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



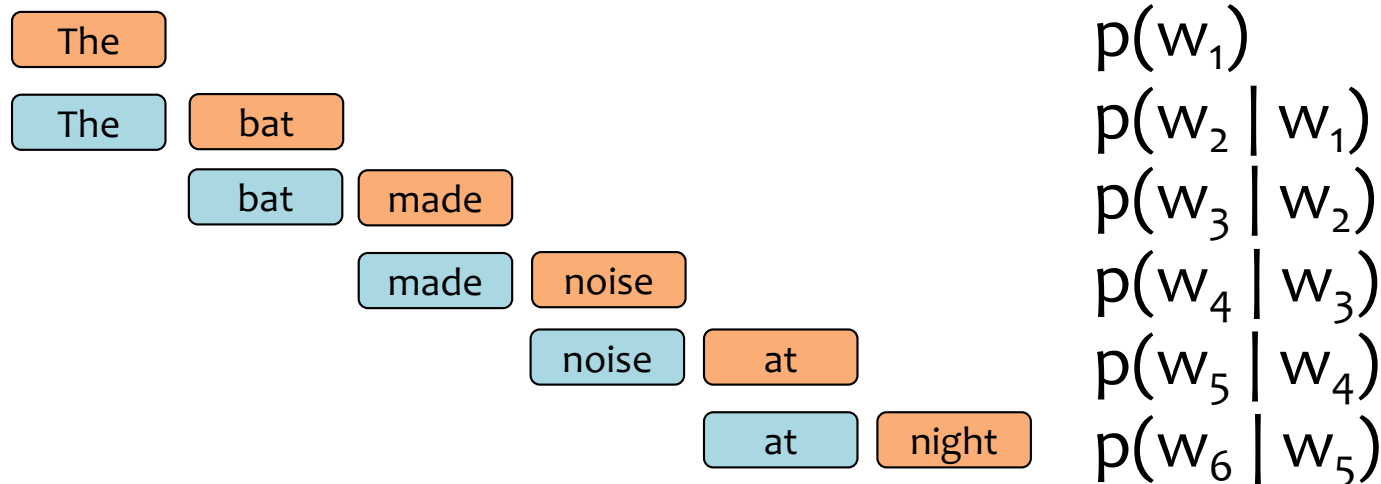
n-Gram Language Model

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



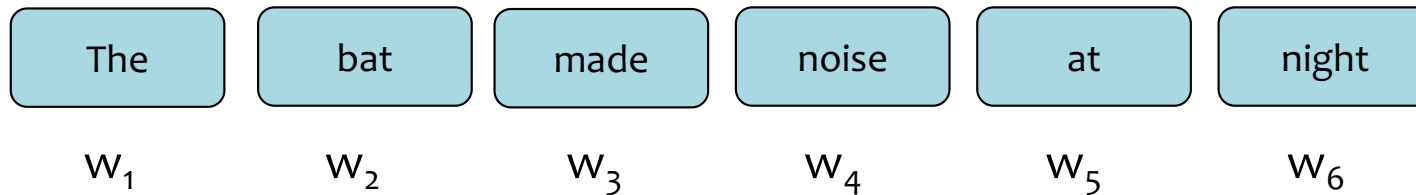
n-Gram Model (n=2)
$$p(w_1, w_2, \dots, w_T) = \prod_{t=1}^T p(w_t | w_{t-1})$$

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



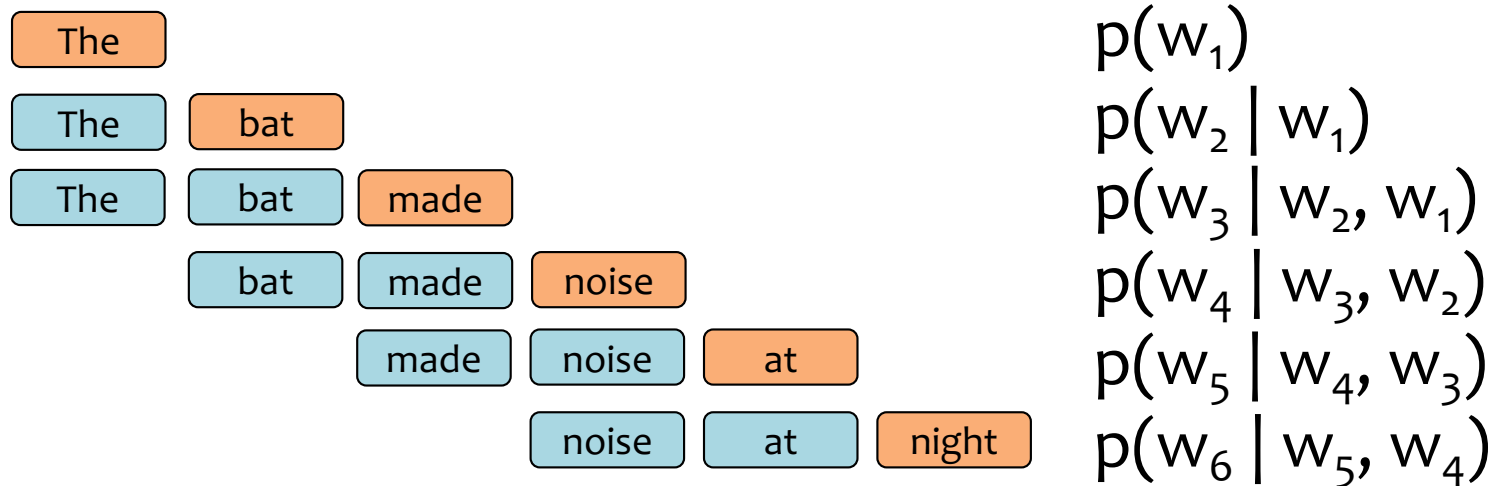
n-Gram Language Model

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



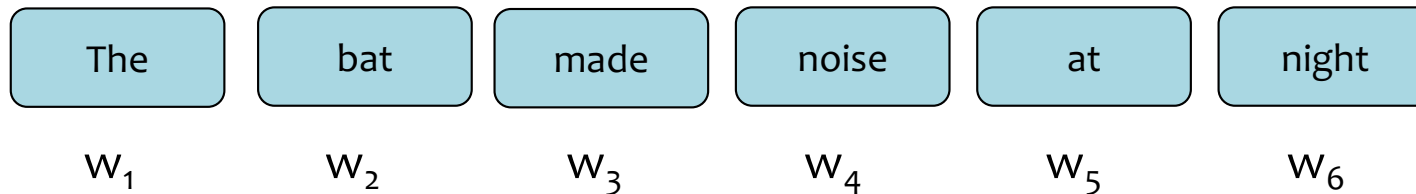
n-Gram Model (n=3)
$$p(w_1, w_2, \dots, w_T) = \prod_{t=1}^T p(w_t | w_{t-1}, w_{t-2})$$

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



n-Gram Language Model

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



n-Gram Model (n=3)

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

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

$$p(w_1)$$

$$p(w_2 | w_1)$$

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

Question: How do we **learn** the probabilities for the n-Gram Model?

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



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

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



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

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



w_t	$p(\cdot \cdot, \cdot)$
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**?...

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


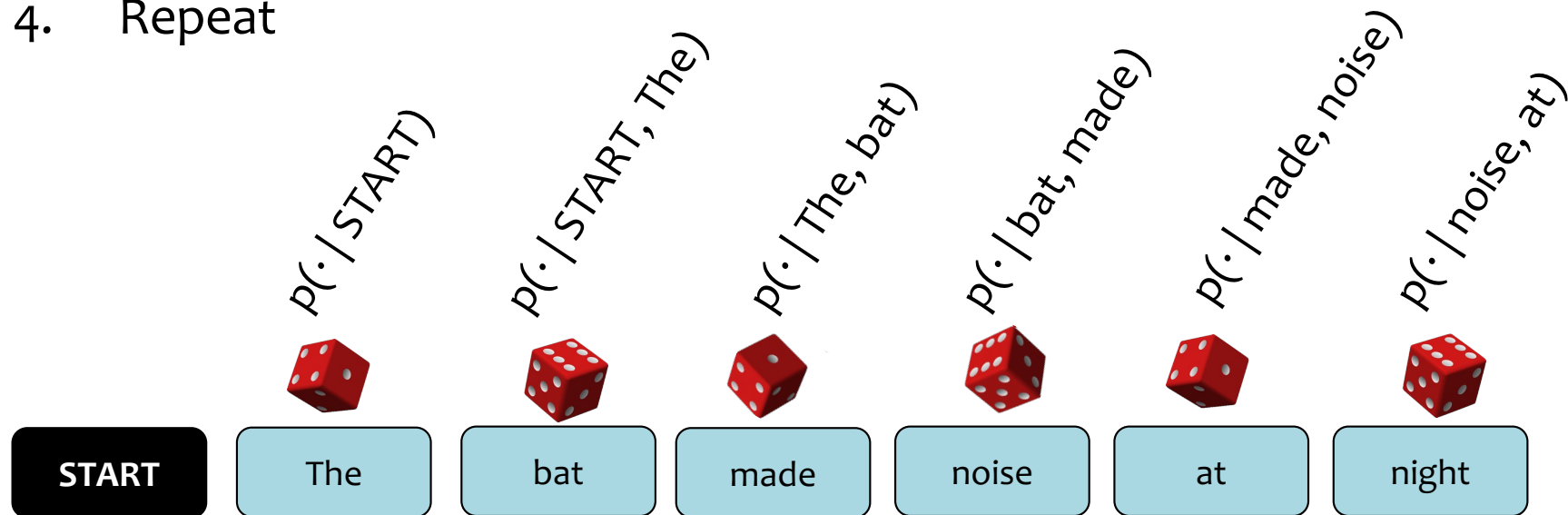
w_t	$p(\cdot \mid \cdot, \cdot)$
corn	4/11
grass	3/11
hay	2/11
if	1/11
which	1/11

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 $p(w_t | w_{t-2}, w_{t-1})$
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 (Shakespeare)

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.

5-Gram Model

Approacheth, denay. dungy
Thither! Julius think: grant,--0
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: $\mathbf{h} = (h_1, h_2, \dots, h_T), h_i \in \mathcal{R}^J$

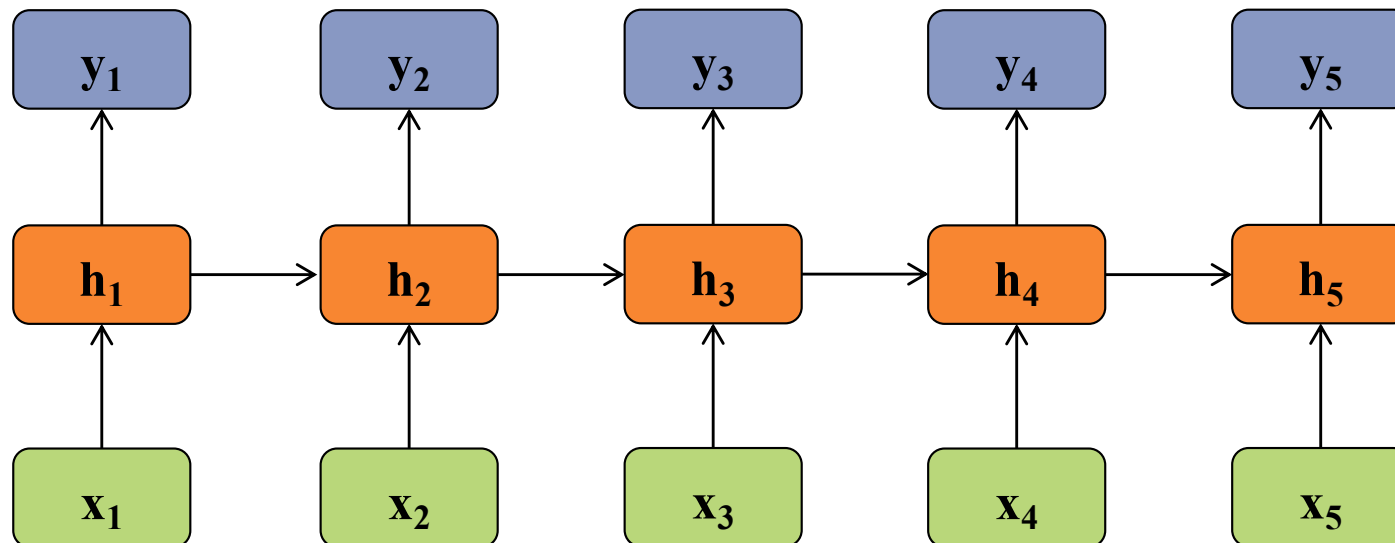
outputs: $\mathbf{y} = (y_1, y_2, \dots, y_T), y_i \in \mathcal{R}^K$

nonlinearity: \mathcal{H}

Definition of the RNN:

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

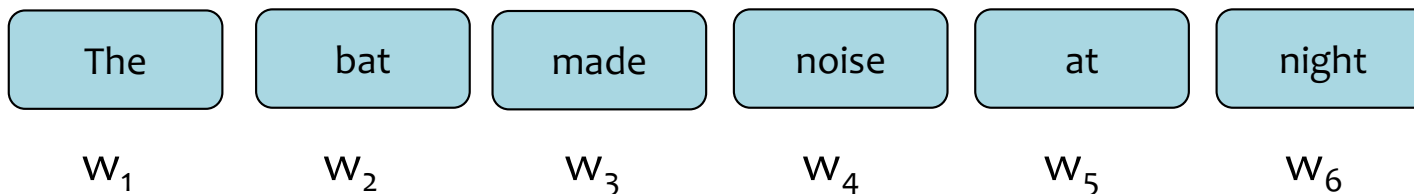
$$y_t = W_{hy}h_t + b_y$$



Recall...

The Chain Rule of Probability

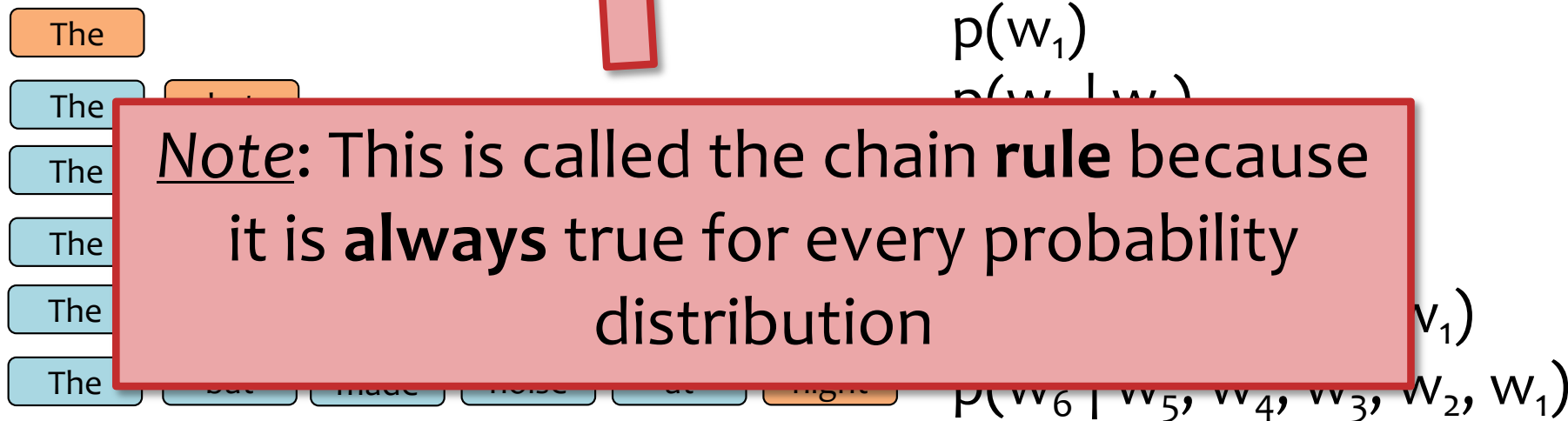
Question: How can we **define** a probability distribution over a sequence of length T?



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

$$p(w_1, w_2, w_3, \dots, w_6) = p(w_1) p(w_2 | w_1) p(w_3 | w_2, w_1) \dots$$

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



RNN Language Model

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

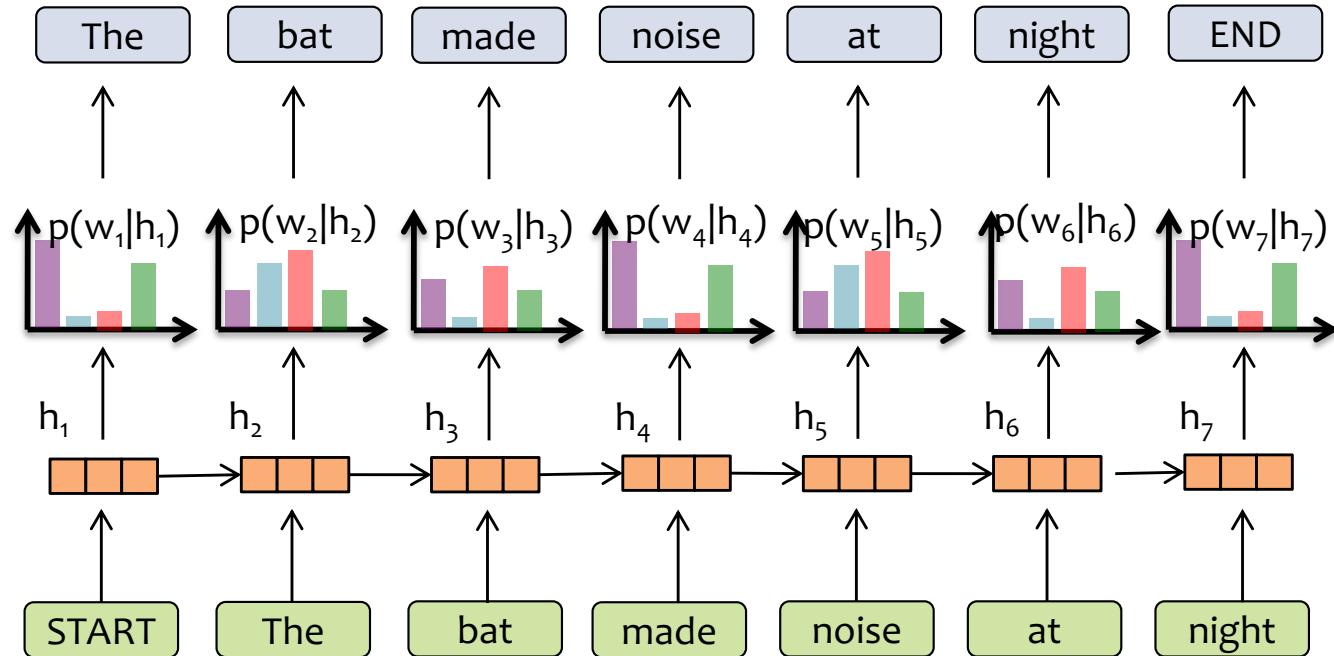
$$p(w_1, w_2, w_3, \dots, w_6) =$$

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

Key Idea:

- (1) convert all previous words to a **fixed length vector**
- (2) define distribution $p(w_t | f_{\theta}(w_{t-1}, \dots, w_1))$ that conditions on the vector

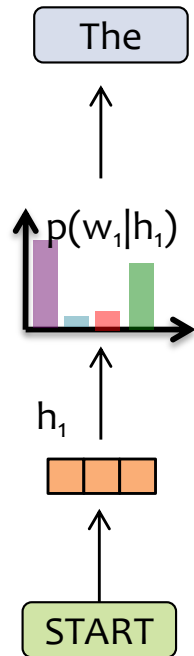
RNN Language Model



Key Idea:

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

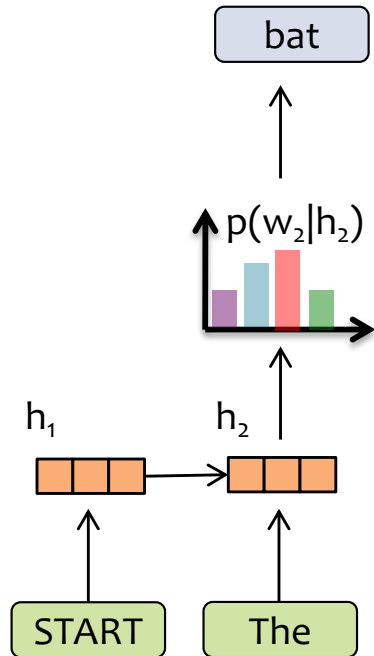
RNN Language Model



Key Idea:

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

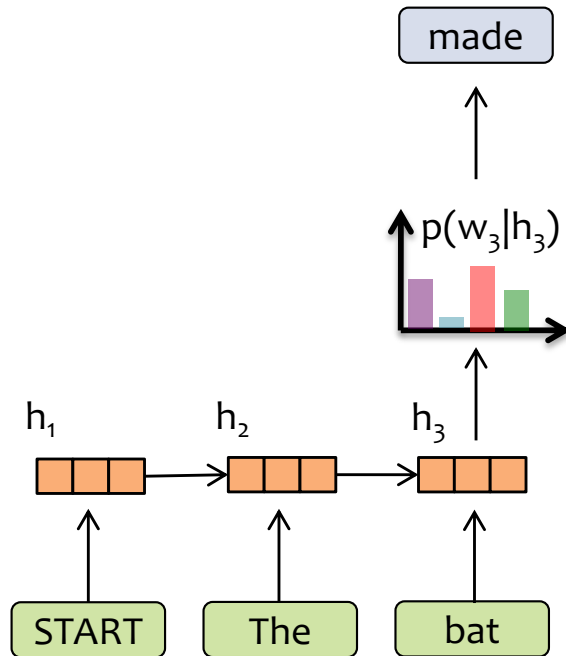
RNN Language Model



Key Idea:

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

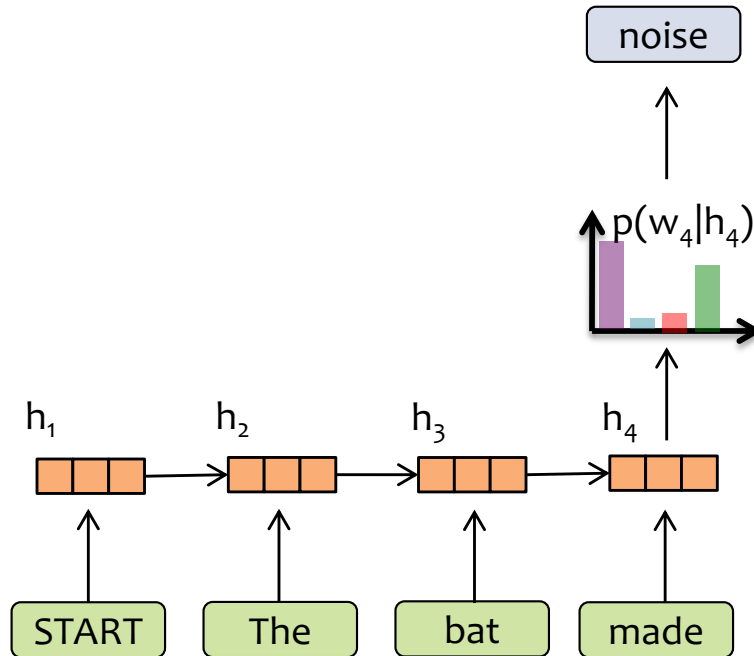
RNN Language Model



Key Idea:

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

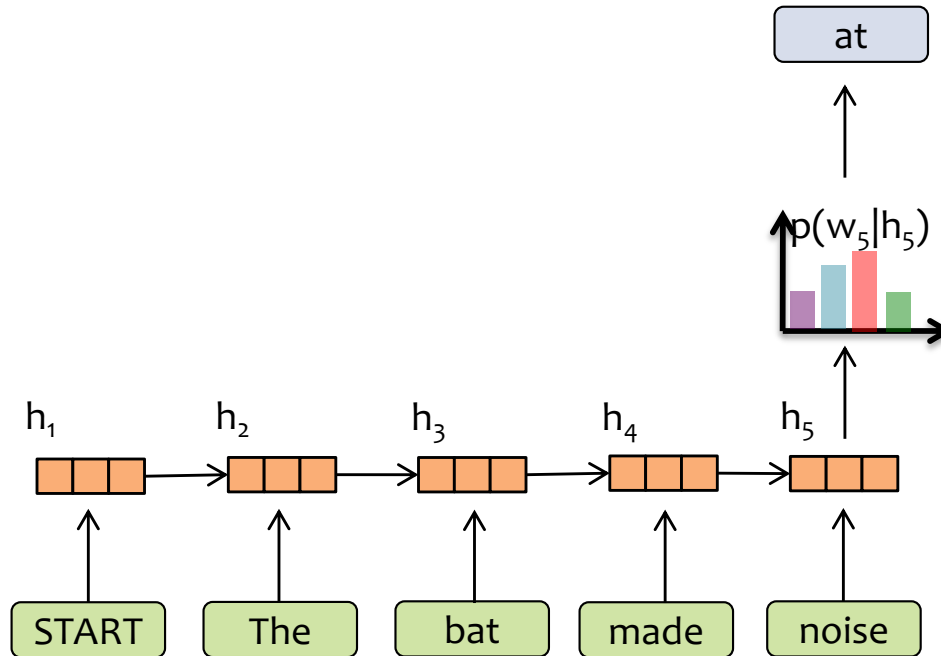
RNN Language Model



Key Idea:

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

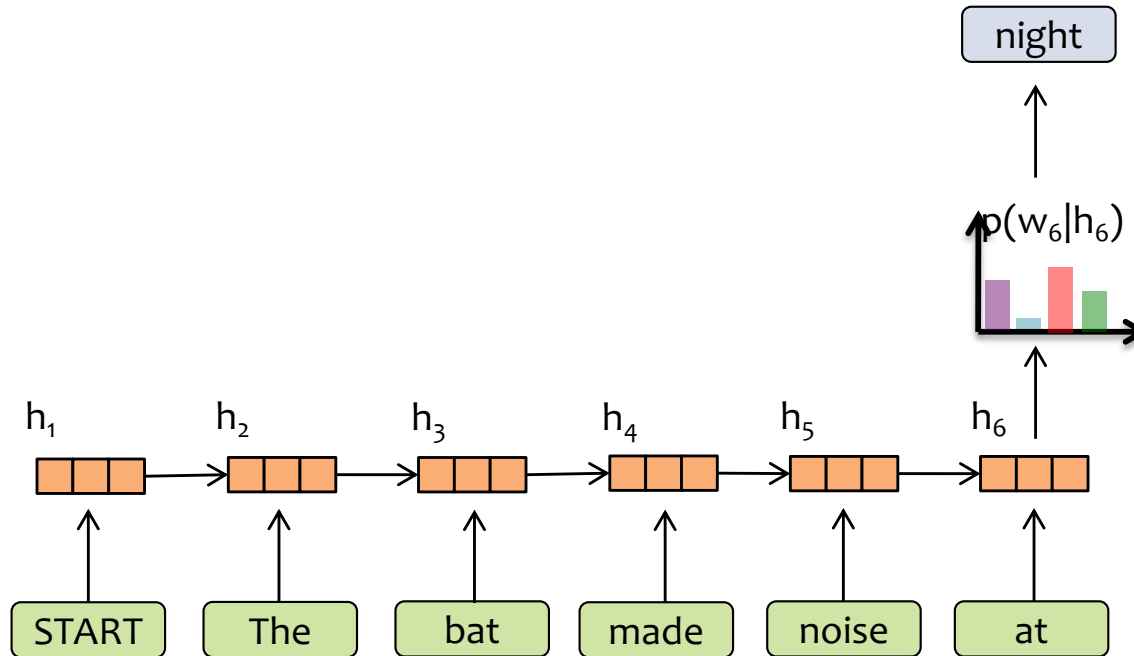
RNN Language Model



Key Idea:

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

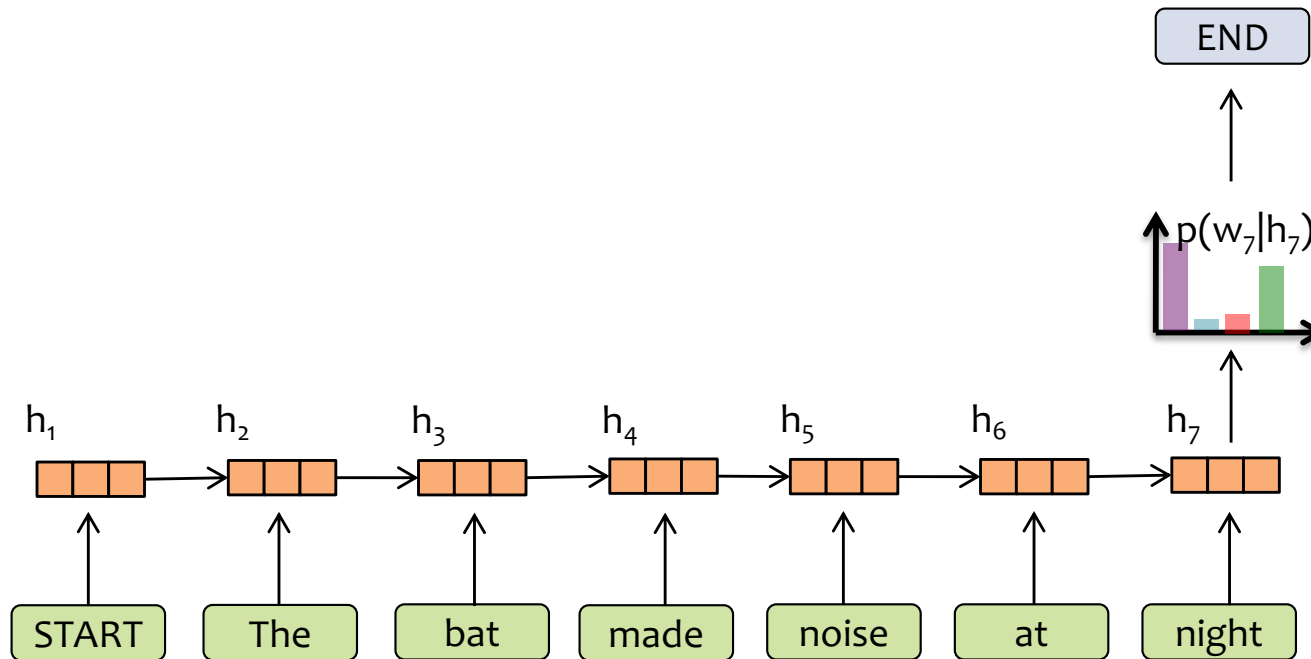
RNN Language Model



Key Idea:

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

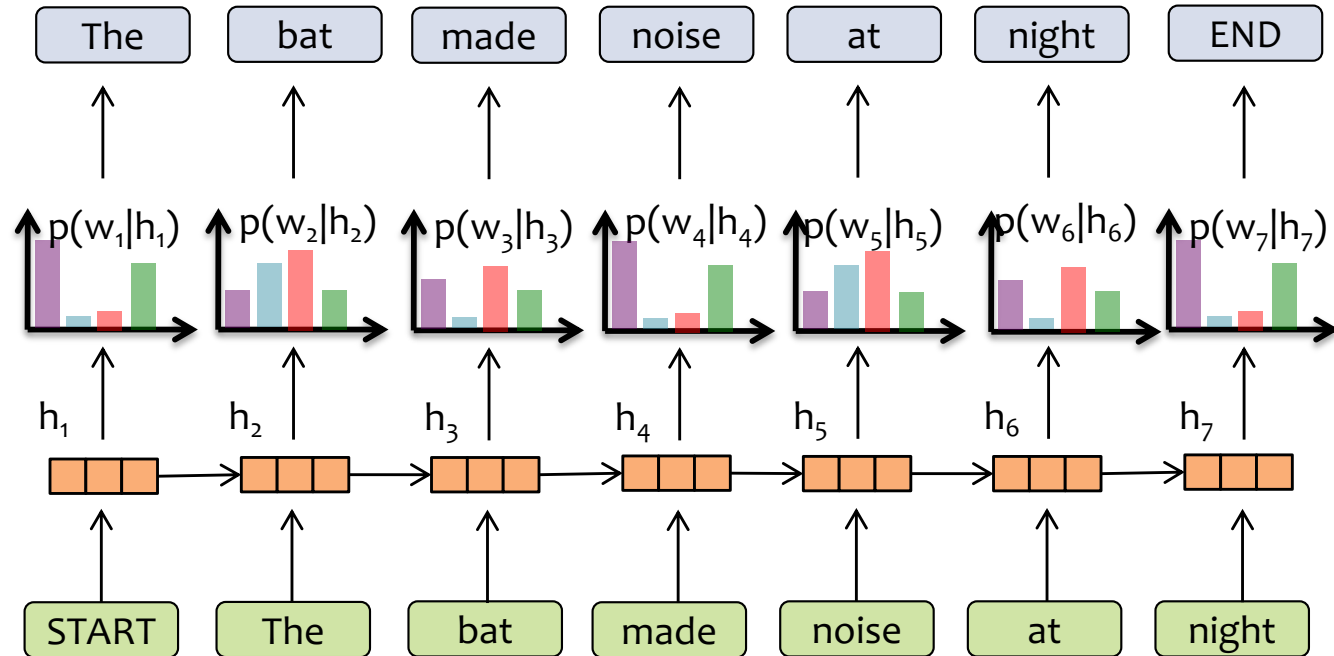
RNN Language Model



Key Idea:

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

RNN Language Model



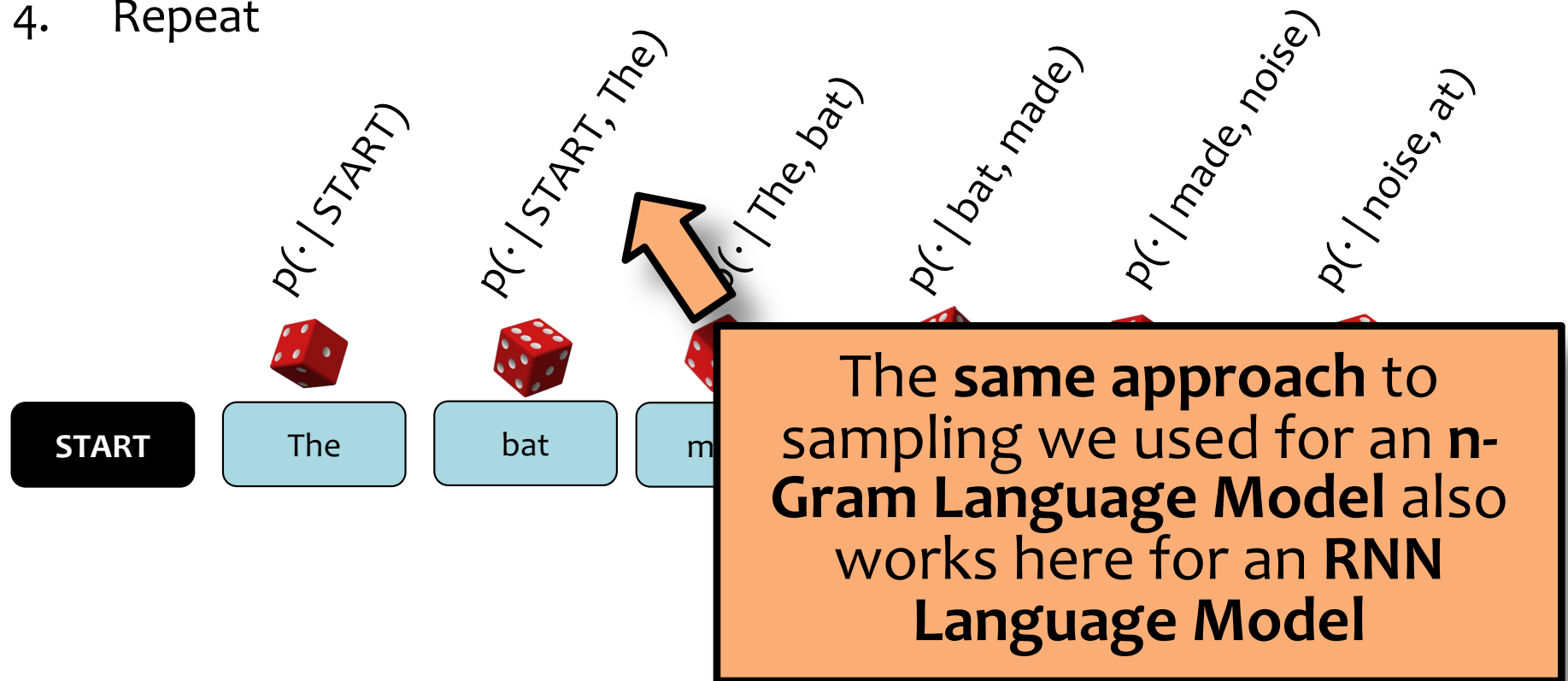
$$p(w_1, w_2, w_3, \dots, w_T) = p(w_1 | h_1) p(w_2 | h_2) \dots p(w_T | h_T)$$

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 $p(w_t | w_{t-2}, w_{t-1})$
3. Roll that die and generate whichever word w_t lands face up
4. Repeat



Sampling from an RNN-LM

??

VIOLA: Why, Salisbury must find his flesh and thought That which I am not ap, 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 power 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, I should be loath to foil him, as I honour, if he come in: My love to you, I came hither to acquaint you with that either you might stay him from his intent 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.

Which is the real Shakespeare?!

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 apt, 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.

Sampling from an RNN-LM

RNN-LM Sample

VIOLA: Why, Salisbury must find his flesh and thought That which I am not apt, 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.

Sampling from an RNN-LM

??

VIOLA: Why, Salisbury must find his flesh and thought That which I am not apt, 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 power so much as hell: Some service in bondman here, Would show his

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.

??

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 self; he is but young and tender; and, I should be loath to foil him, as I should lose my honour, if he come in: My love to you, I came hither to acquaint you withal, that either you might stay him from his intended 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.

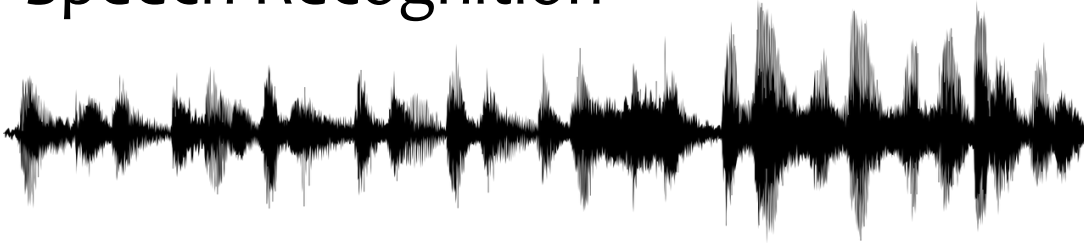
TOUCHSTONE: For my part, I had rather bear with you than bear you; yet I should bear no cross if I did bear you, for I think you have no money in your purse.

Which is the real Shakespeare?!

SEQUENCE TO SEQUENCE MODELS

Sequence to Sequence Model

Speech Recognition



Machine Translation

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

Summarization

```

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Id nibh tortor id aliquet lectus proin nibh nisi. Odio ut enim blandit volutpat maecenas volutpat. Porta nibh venenatis cras sed. Quam id leo in vitae. Aliquam id diam maecenas ultricies mi. Et sollicitudin ac orci phasellus egestas. Diam in arcu cursus euismod quis viverra. Vitae auctor eu augue ut lectus arcu. Semper quis lectus nulla at volutpat diam ut. Sed arcu non odio euismod lacinia. Velit euismod in pellentesque massa. Augue lacus viverra vitae congue eu consequat ac. Tincidunt id ali.
```


Sequence to Sequence Model

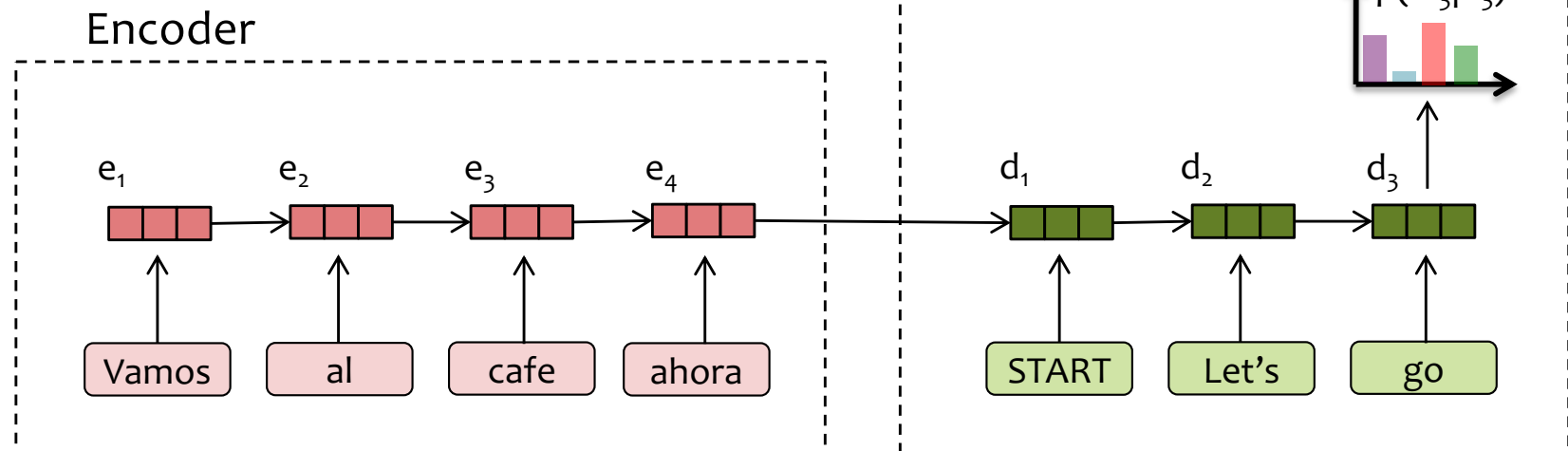
Now suppose you want generate a sequence conditioned on another input

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



Dynamic Programming

Question:

Have you studied dynamic programming in a previous course?

- A. Yes
- B. No

Answer:

Question:

What is the difference between memoization and tabulation, when applied to a recursive function $f(x)$?

- A. **memoization** computes a function recursively without storing intermediate results, whereas **tabulation** stores intermediate results
- B. **memoization** stores function values as they are encountered top-down, whereas **tabulation** stores function values as they are encountered bottom-up
- C. **memoization** stores only the output of a tertiary function $g(x)$, whereas **tabulation** stores the outputs of $f(x)$ directly
- D. **memoization** typically increases computational complexity of an algorithm while decreasing space complexity, whereas **tabulation** typically decreases computational complexity and increases space complexity
- E. **memoization** memorizes a function, whereas **tabulation** has a programmer generate code for the function on-the-fly (i.e. I answered “Yes” to previous question)

Answer:

BACKGROUND: COMPUTER VISION

Example: Image Classification

- ImageNet LSVRC-2011 contest:
 - **Dataset:** 1.2 million labeled images, 1000 classes
 - **Task:** Given a new image, label it with the correct class
 - **Multiclass** classification problem
- Examples from <http://image-net.org/>

Bird

Warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings

2126
pictures

92.85%
Popularity
Percentile

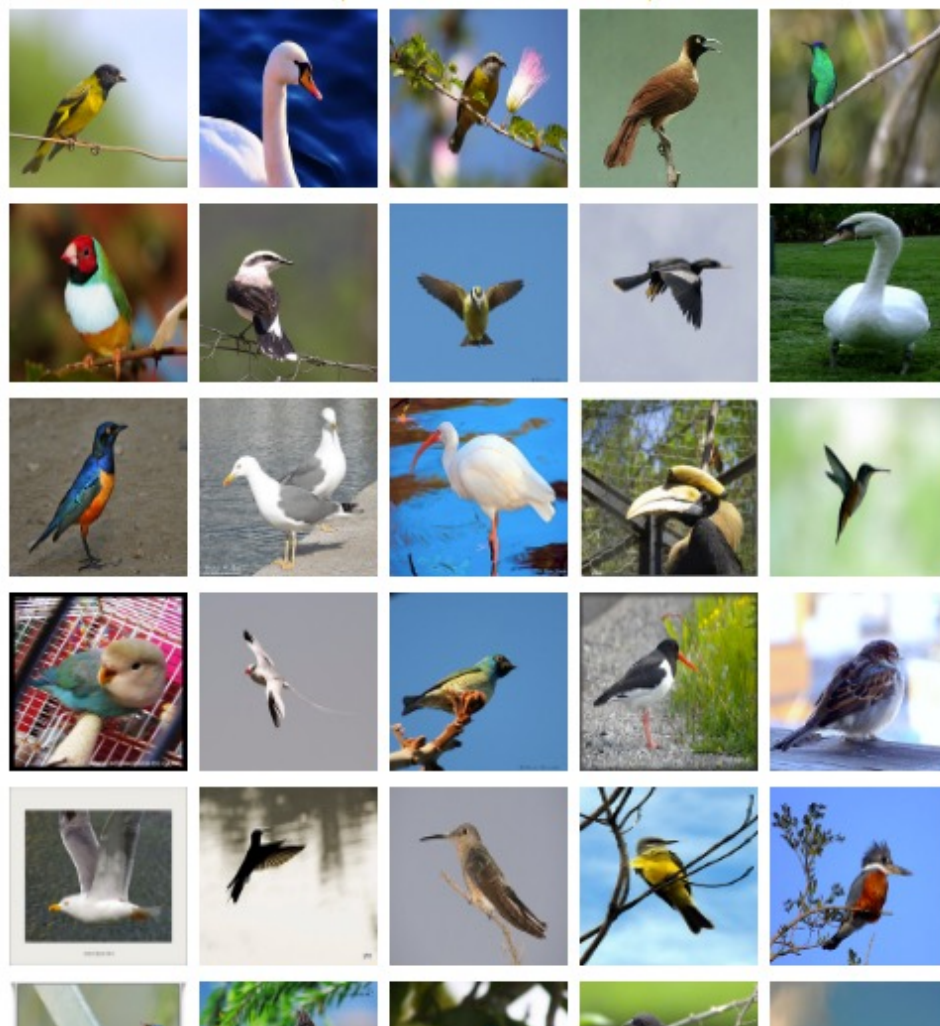


- marine animal, marine creature, sea animal, sea creature (1)
- scavenger (1)
- biped (0)
- predator, predatory animal (1)
- larva (49)
- acrodont (0)
- feeder (0)
- stunt (0)
- chordate (3087)**
 - tunicate, urochordate, urochord (6)
 - cephalochordate (1)
 - vertebrate, craniate (3077)**
 - mammal, mammalian (1169)
 - bird (871)**
 - dickeybird, dickey-bird, dickybird, dicky-bird (0)
 - cock (1)
 - hen (0)
 - nester (0)
 - night bird (1)
 - bird of passage (0)
 - protoavis (0)
 - archaeopteryx, archeopteryx, Archaeopteryx lithographi Sinornis (0)
 - lbero-mesornis (0)
 - archaeornis (0)
 - ratite, ratite bird, flightless bird (10)
 - carinate, carinate bird, flying bird (0)
 - passerine, passeriform bird (279)
 - nonpasserine bird (0)
 - bird of prey, raptor, raptorial bird (80)
 - gallinaceous bird, gallinacean (114)

Treemap Visualization

Images of the Synset

Downloads



German iris, *Iris kochii*

Iris of northern Italy having deep blue-purple flowers; similar to but smaller than *Iris germanica*

469 pictures

49.6% Popularity Percentile



Wordnet IDs

Treemap Visualization

Images of the Synset

Downloads



- halophyte (0)
- succulent (39)
- cultivar (0)
- cultivated plant (0)
- weed (54)
- evergreen, evergreen plant (0)
- deciduous plant (0)
- vine (272)
- creeper (0)
- woody plant, ligneous plant (1868)
- geophyte (0)
- desert plant, xerophyte, xerophytic plant, xerophile, xerophilic
- mesophyte, mesophytic plant (0)
- aquatic plant, water plant, hydrophyte, hydrophytic plant (11)
- tuberous plant (0)
- bulbous plant (179)
 - iridaceous plant (27)
 - iris, flag, fleur-de-lis, sword lily (19)
 - bearded iris (4)
 - Florentine iris, orris, *Iris germanica florentina*, *Iris germanica* (0)
 - German iris, *Iris germanica* (0)
 - German iris, *Iris kochii* (0)
 - Dalmatian iris, *Iris pallida* (0)
 - beardless iris (4)
 - bulbous iris (0)
 - dwarf iris, *Iris cristata* (0)
 - stinking iris, gladdon, gladdon iris, stinking gladwyn, Persian iris, *Iris persica* (0)
 - yellow iris, yellow flag, yellow water flag, *Iris pseudo*
 - dwarf iris, vernal iris, *Iris verna* (0)
 - blue flag, *Iris versicolor* (0)

Court, courtyard

An area wholly or partly surrounded by walls or buildings; "the house was built around an inner court"

165 pictures

92.61% Popularity Percentile

Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
 - plant, flora, plant life (4486)
 - geological formation, formation (175)
 - natural object (1112)
 - sport, athletics (176)
 - artifact, artefact (10504)
 - instrumentality, instrumentation (5494)
 - structure, construction (1405)
 - airdock, hangar, repair shed (0)
 - altar (1)
 - arcade, colonnade (1)
 - arch (31)
 - area (344)
 - aisle (0)
 - auditorium (1)
 - baggage claim (0)
 - box (1)
 - breakfast area, breakfast nook (0)
 - bullpen (0)
 - chancel, sanctuary, bema (0)
 - choir (0)
 - corner, nook (2)
 - court, courtyard (6)
 - atrium (0)
 - bailey (0)
 - cloister (0)
 - food court (0)
 - forecourt (0)
 - narvis (0)

Treemap Visualization

Images of the Synset

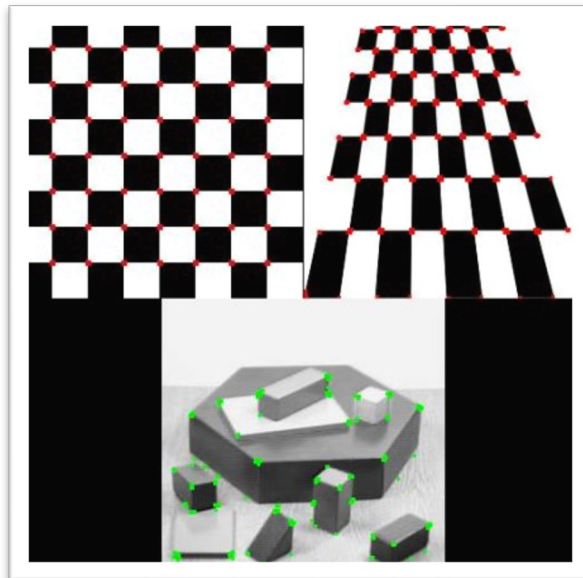
Downloads

Feature Engineering for CV

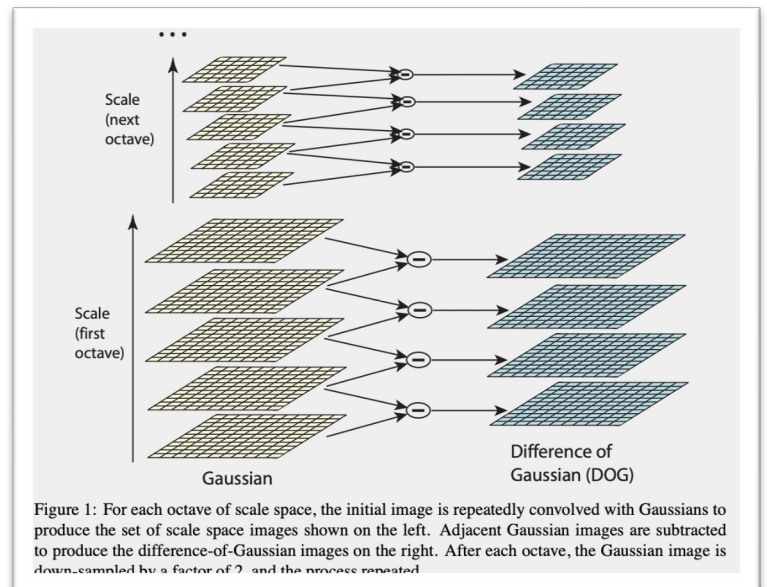
Edge detection (Canny)



Corner Detection (Harris)



Scale Invariant Feature Transform (SIFT)



Example: Image Classification

CNN for Image Classification

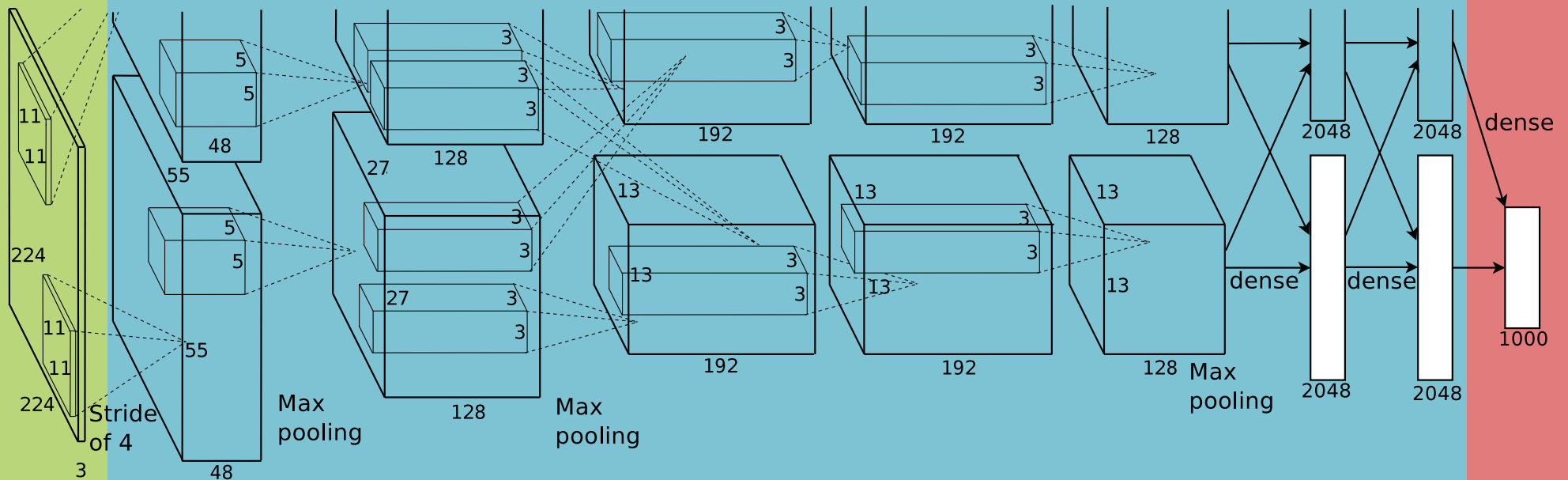
(Krizhevsky, Sutskever & Hinton, 2012)

15.3% error on ImageNet LSVRC-2012 contest

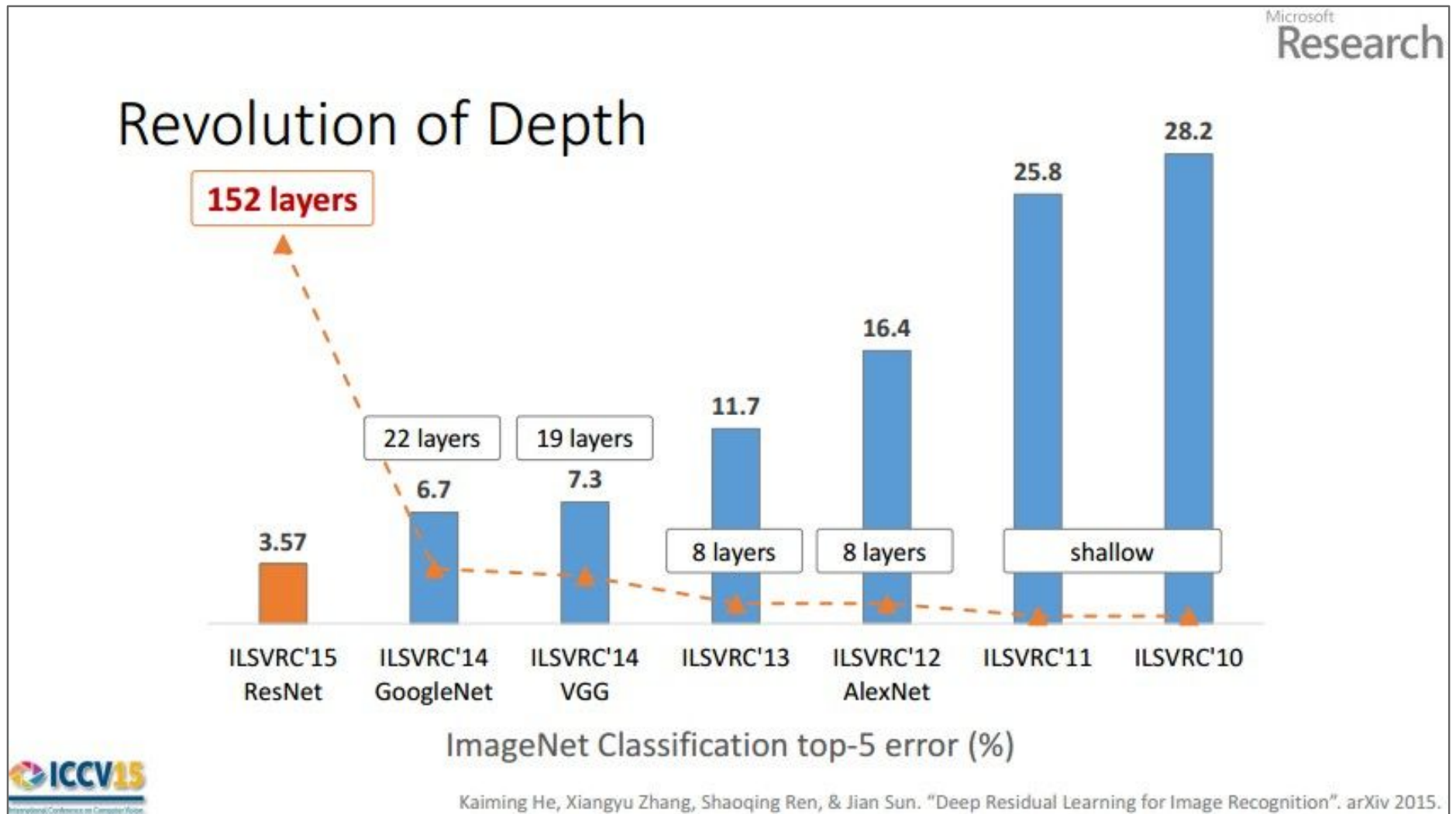
Input image (pixels)

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

1000-way softmax



CNNs for Image Recognition



Backpropagation and Deep Learning

Convolutional neural networks (CNNs) and **recurrent neural networks (RNNs)** are simply fancy computation graphs (aka. hypotheses or decision functions).

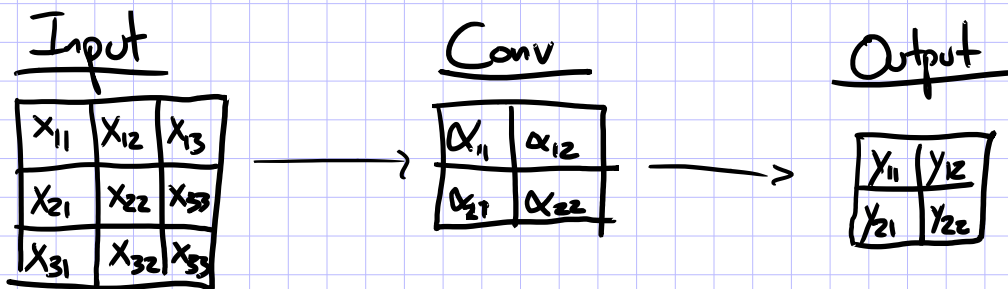
Our recipe also applies to these models and (again) relies on the **backpropagation algorithm** to compute the necessary gradients.

CONVOLUTION

What's a convolution?

- Basic idea:
 - Pick a 3×3 matrix F of weights
 - Slide this over an image and compute the “inner product” (similarity) of F and the corresponding field of the image, and replace the pixel in the center of the field with the output of the inner product operation
- Key point:
 - Different convolutions extract different types of low-level “features” from an image
 - All that we need to vary to generate these different features is the weights of F

Ex: 1 input channel, 1 output channel



$$y_{11} = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{21}x_{21} + \alpha_{22}x_{22} + \alpha_0$$

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

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

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

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

0	0	0
0	1	1
0	1	0

Convolved Image

1	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	0	0	0	0

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

0	0	0
0	1	1
0	1	0

Convolved Image

3	2	2	3	1
2	0	2	1	0
2	2	1	0	0
3	1	0	0	0
1	0	0	0	0

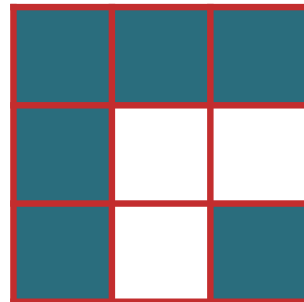
Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution



Convolved Image

3	2	2	3	1
2	0	2	1	0
2	2	1	0	0
3	1	0	0	0
1	0	0	0	0

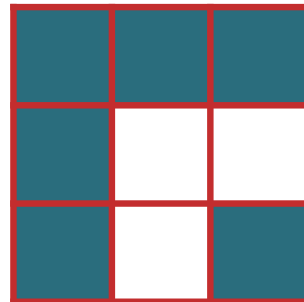
Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution



Convolved Image

3	2	2	3	1
2	0	2	1	0
2	2	1	0	0
3	1	0	0	0
1	0	0	0	0

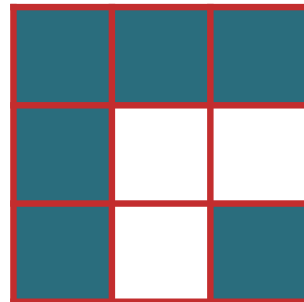
Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution



Convolved Image

3	2	2	3	1
2	0	2	1	0
2	2	1	0	0
3	1	0	0	0
1	0	0	0	0

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

			0	0	0	0
	1	1	1	1	1	0
	1		0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3				

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0				0	0	0
0		1	1	1	1	0
0		0		1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3	2			

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0				0	0
0	1		1	1	1	0
0	1		0		0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3	2	2		

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	1	1	1	0
0	1	1	1	1	1	0
0	1	0	1	1	1	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

1	1	1
1	0	0
1	0	1

Convolved Image

3	2	2	3	
	1	1		1
	1	0	1	1
		1	1	1
	1	1	1	1

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0			
0	1	1	1		1	0
0	1	0	0		0	
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3	2	2	3	1

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
			1	1	1	0
	1	0	0	1	0	0
	1		1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3	2	2	3	1
2				

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0				1	1	0
0			0	0	1	0
0			0		0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution

Convolved Image

3	2	2	3	1
2	0			

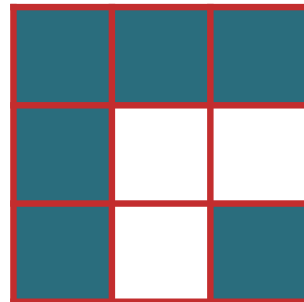
Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Convolution



Convolved Image

3	2	2	3	1
2	0	2	1	0
2	2	1	0	0
3	1	0	0	0
1	0	0	0	0

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Identity
Convolution

0	0	0
0	1	0
0	0	0

Convolved Image

1	1	1	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	0	0	0	0

Background: Image Processing

A **convolution matrix** is used in image processing for tasks such as edge detection, blurring, sharpening, etc.

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Blurring
Convolution

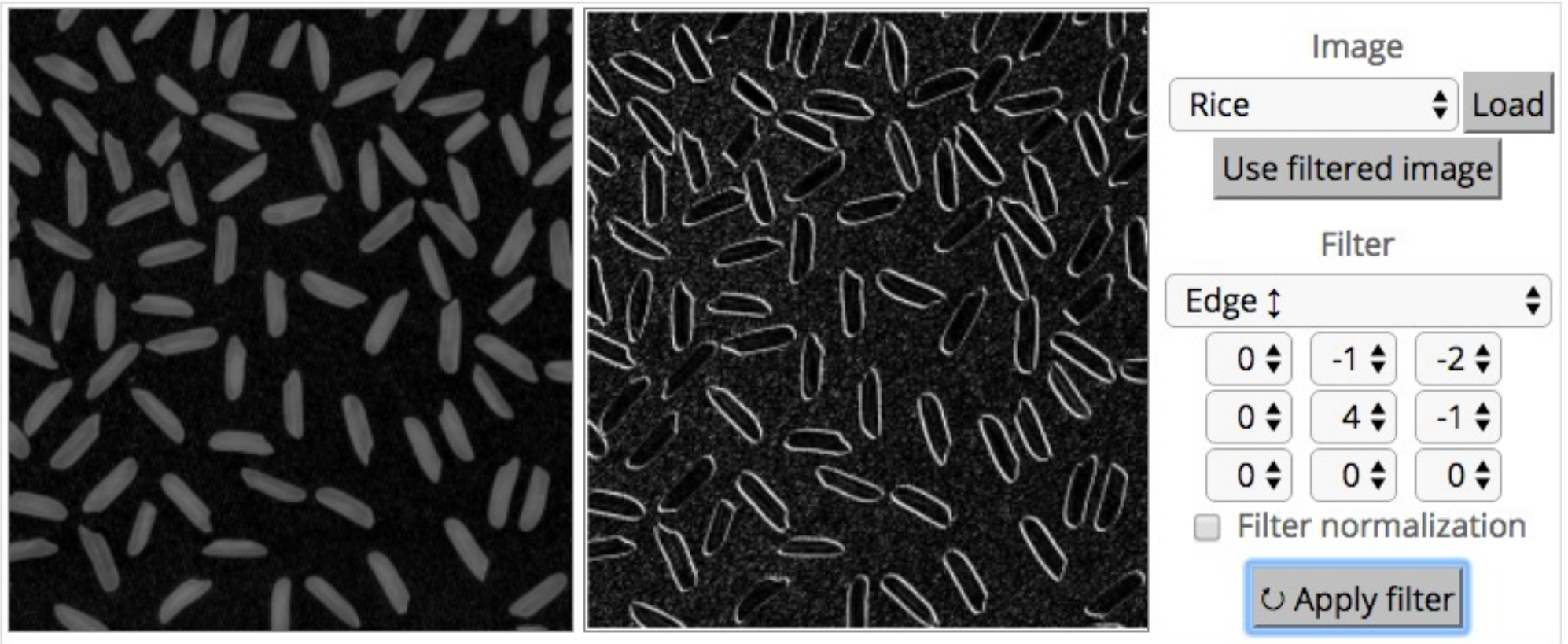
.1	.1	.1
.1	.2	.1
.1	.1	.1

Convolved Image

.4	.5	.5	.5	.4
.4	.2	.3	.6	.3
.5	.4	.4	.2	.1
.5	.6	.2	.1	0
.4	.3	.1	0	0

What's a convolution?

<http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo>



Image

Rice

Filter

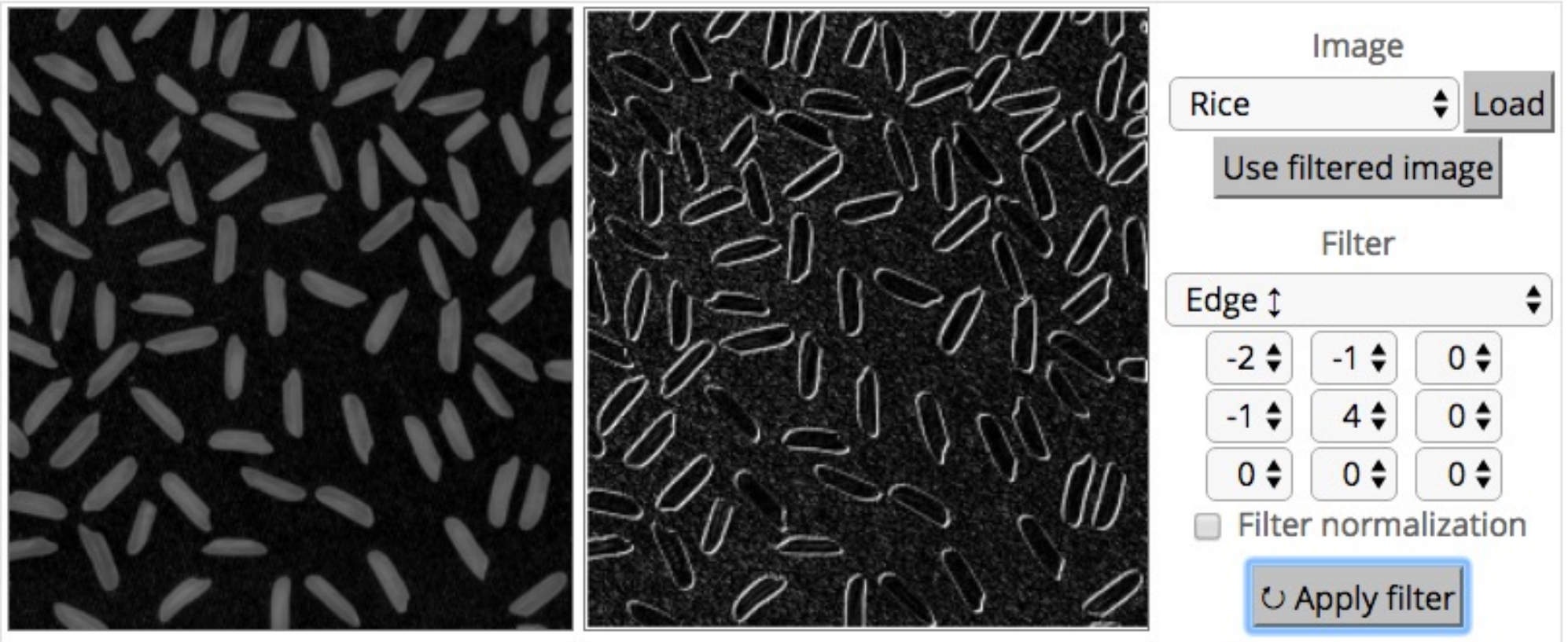
Edge

0 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>	-2 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	4 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>

Filter normalization

What's a convolution?

<http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo>



Image

Rice

Filter

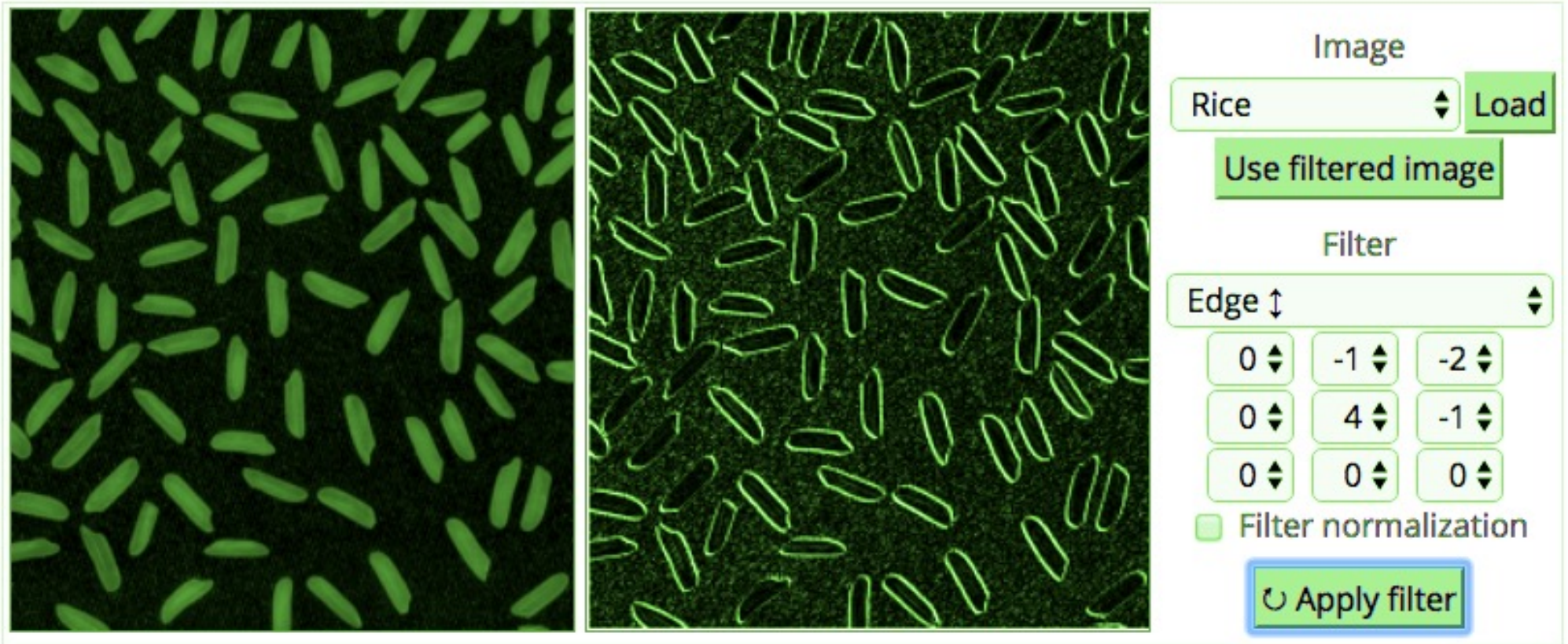
Edge

-2 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>
-1 <input type="button" value="↓"/>	4 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>

Filter normalization

What's a convolution?

<http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo>



Image

Rice

Filter

Edge

0 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>	-2 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	4 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>

Filter normalization

What's a convolution?

<http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo>

Image

Rice

Filter

Edge

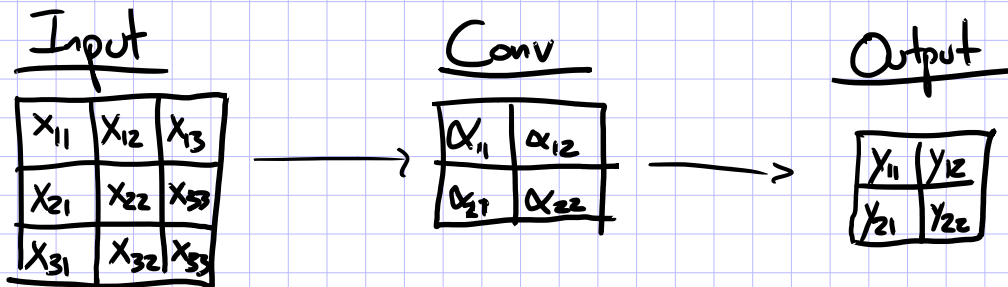
-2 <input type="button" value="↓"/>	-1 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>
-1 <input type="button" value="↓"/>	4 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>
0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>	0 <input type="button" value="↓"/>

Filter normalization

What's a convolution?

- Basic idea:
 - Pick a 3×3 matrix F of weights
 - Slide this over an image and compute the “inner product” (similarity) of F and the corresponding field of the image, and replace the pixel in the center of the field with the output of the inner product operation
- Key point:
 - Different convolutions extract different types of low-level “features” from an image
 - All that we need to vary to generate these different features is the weights of F

Ex: 1 input channel, 1 output channel



$$y_{11} = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{21}x_{21} + \alpha_{22}x_{22} + \alpha_0$$

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

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

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

DOWNSAMPLING

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3		

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3		

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3	1	

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3	1	0

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3	1	0
1		

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3	1	0
1	0	

Downsampling

- Suppose we use a convolution with stride 2
- Only 9 patches visited in input, so only 9 pixels in output

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

1	1
1	1

Convolved Image

3	3	1
3	1	0
1	0	0

Downsampling by Averaging

- Downsampling by averaging is a special case of convolution where the weights are fixed to a uniform distribution
- The example below uses a stride of 2

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Convolution

$1/4$	$1/4$
$1/4$	$1/4$

Convolved Image

$3/4$	$3/4$	$1/4$
$3/4$	$1/4$	0
$1/4$	0	0

Max-Pooling

- Max-pooling is another form of downsampling
- Instead of averaging, we take the max value within the same range as the equivalently-sized convolution
- The example below uses a stride of 2

Input Image

1	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

Max-pooling

$x_{i,j}$	$x_{i,j+1}$
$x_{i+1,j}$	$x_{i+1,j+1}$

$$y_{ij} = \max(x_{ij}, x_{i,j+1}, x_{i+1,j}, x_{i+1,j+1})$$

Max-Pooled Image

1	1	1
1	1	0
1	0	0

CONVOLUTIONAL NEURAL NETS

Background

A Recipe for Machine Learning

1. Given training data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

2. Choose each of these:

– Decision function

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

– Loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}_i) \in \mathbb{R}$$

3. Define goal:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

4. Train with SGD:

(take small steps
opposite the gradient)

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

1. • Convolutional Neural Networks (CNNs) provide another form of **decision function**
• Let's see what they look like...

2. CHOOSE EACH OF THESE:

- Decision function

$$\hat{y} = f_{\theta}(x_i)$$

- Loss function

$$\ell(\hat{y}, y_i) \in \mathbb{R}$$

4. Train with SGD:

(Take small steps opposite the gradient)

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t \nabla \ell(f_{\theta}(x_i), y_i)$$

Convolutional Layer

CNN key idea:
Treat convolution matrix as parameters and learn them!

Input Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0



Learned
Convolution

θ_{11}	θ_{12}	θ_{13}
θ_{21}	θ_{22}	θ_{23}
θ_{31}	θ_{32}	θ_{33}

Convolved Image

.4	.5	.5	.5	.4
.4	.2	.3	.6	.3
.5	.4	.4	.2	.1
.5	.6	.2	.1	0
.4	.3	.1	0	0

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

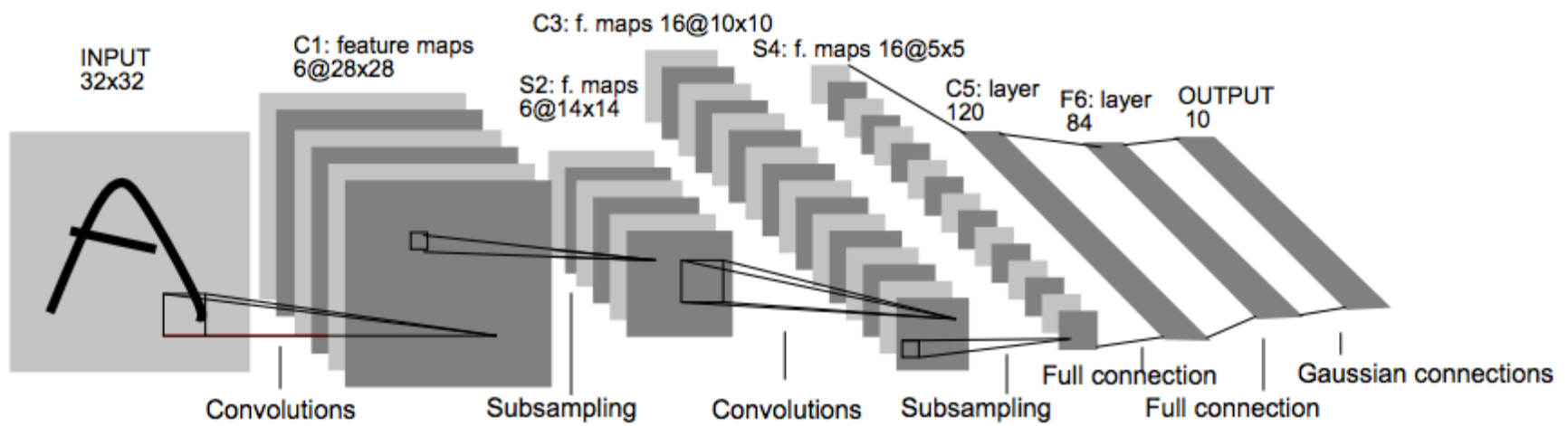


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

TRAINING CNNs

Background

A Recipe for Machine Learning

1. Given training data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

2. Choose each of these:

– Decision function

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

– Loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}_i) \in \mathbb{R}$$

3. Define goal:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

4. Train with SGD:

(take small steps
opposite the gradient)

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

Background

A Recipe for Machine Learning

1. Given training data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

2. Choose each of the following:

– Decision function

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

– Loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}_i) \in \mathbb{R}$$

3. Define goal:

- Q: Now that we have the CNN as a decision function, how do we compute the gradient?
- A: Backpropagation of course!

(opposite the gradient)


$$\boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

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 $i \in \{1, \dots, N\}$

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

Backward: $\nabla_{\vec{\theta}} J_i(\theta) = \dots$

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

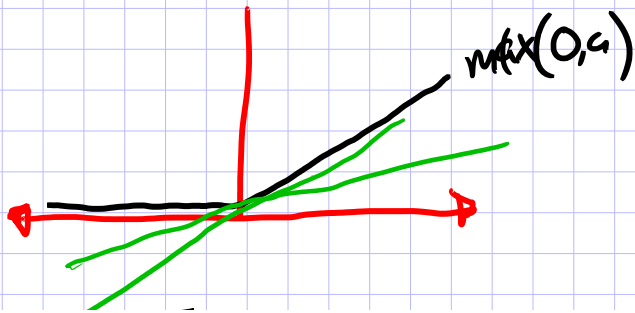
LAYERS OF A CNN

ReLU Layer

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:
 $\frac{dJ}{dx_i} = \frac{dJ}{dy_i} \frac{dy_i}{dx_i}$ *subderivative*

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

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:

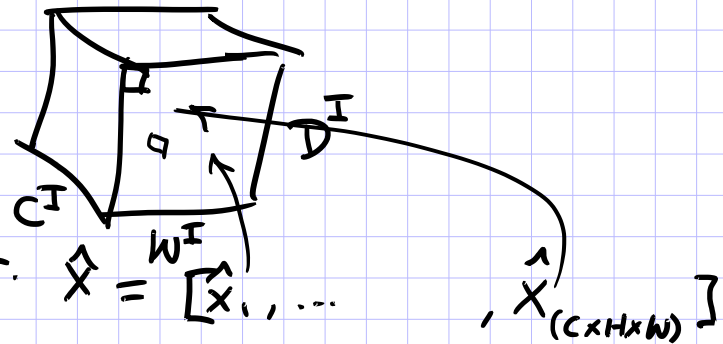
$$\frac{dJ}{dx_j} = \sum_{i=1}^k \frac{dJ}{dy_i} \frac{dy_i}{dx_j}$$

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

Fully-Connected Layer

Fully Connected Layer (w/ tensor input)

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



- stretch out into a long vector. $\hat{X} = [x_1, \dots, x_{(C \times H \times W)}]$

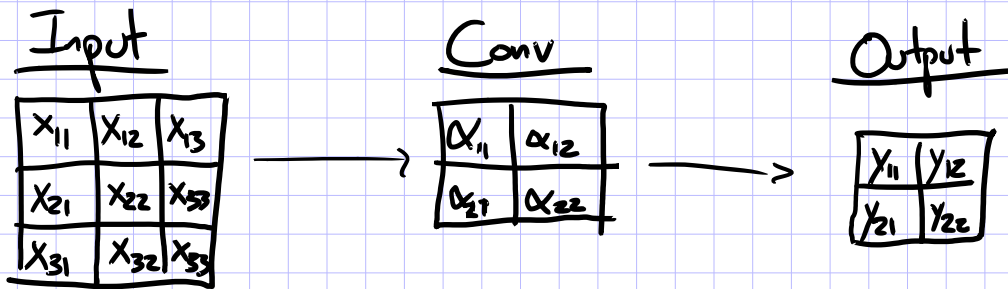
- then standard linear layer:

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

$|\hat{X}| = A, |y| = B$

Convolutional Layer

Ex: 1 input channel, 1 output channel



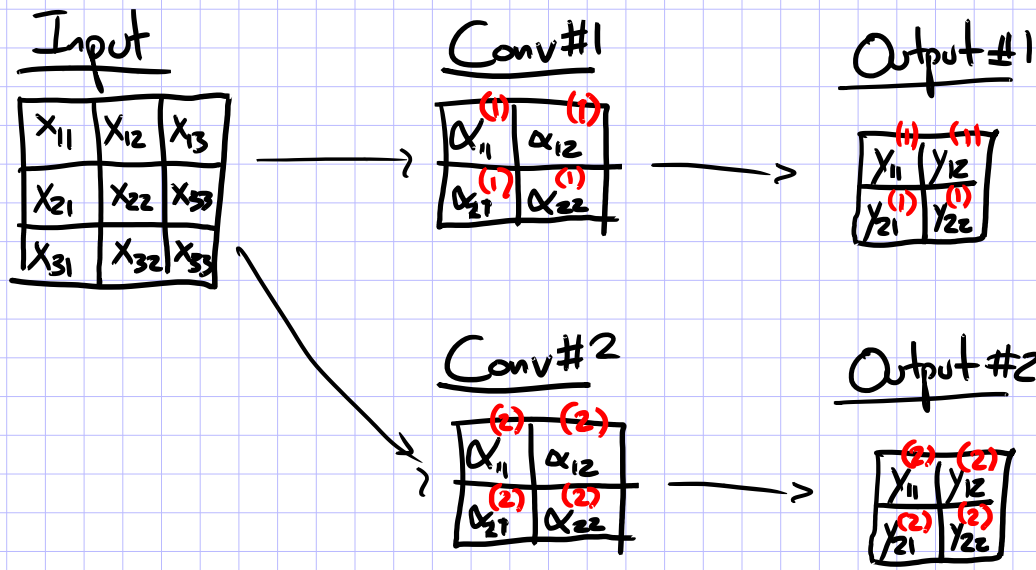
$$y_{11} = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{21}x_{21} + \alpha_{22}x_{22} + \alpha_0$$

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

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

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

Ex: 1 input channel, 2 output channels



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

$$y_{12}^{(1)} = \dots$$

$$y_{21}^{(1)} = \dots$$

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

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

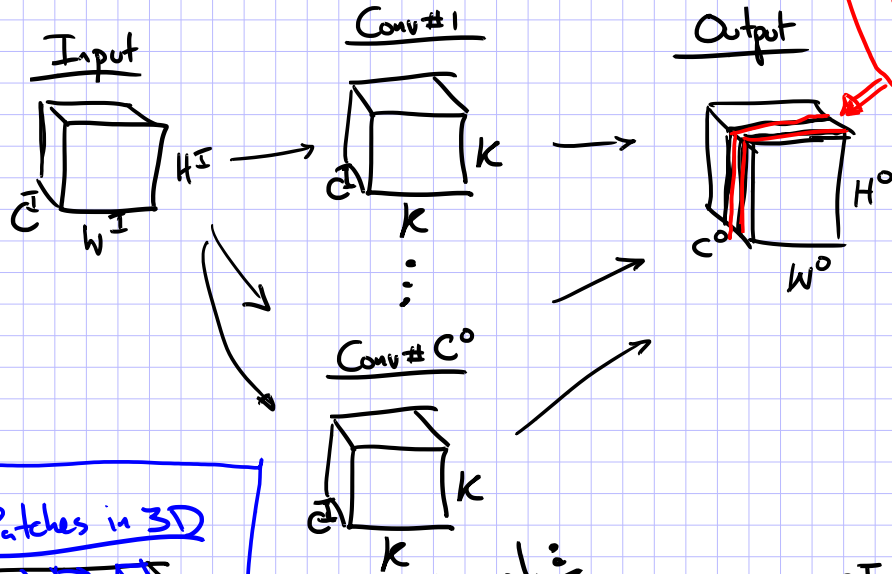
$$y_{12}^{(2)} = \dots$$

$$y_{21}^{(2)} = \dots$$

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

Convolutional Layer

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

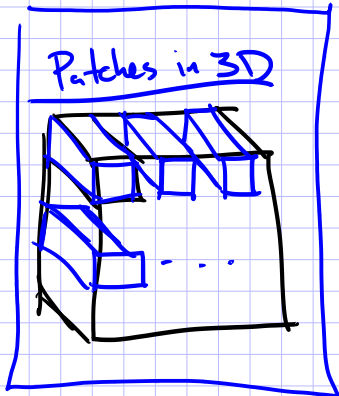


i-th slice is output from j-th convolution matrix

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

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

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



Forward:

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

Backward:

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

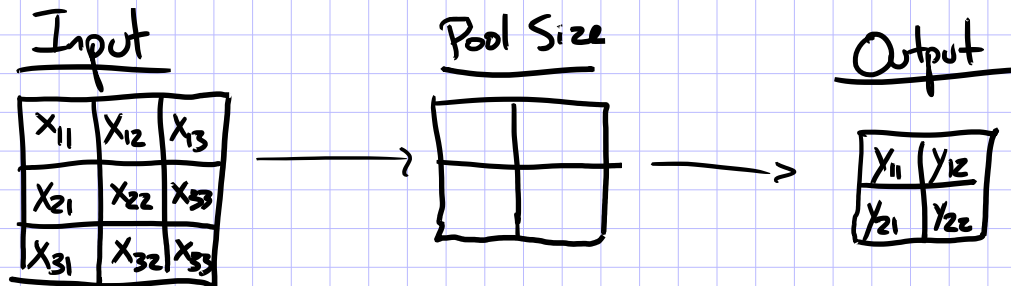
$$\frac{dJ}{d\alpha_{qr}^{(c)}} = \sum_i \sum_j \frac{dJ}{dy_{ij}^{(k)}} \frac{dy_{ij}^{(k)}}{d\alpha_{qr}^{(c)}}$$

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

just some calculus

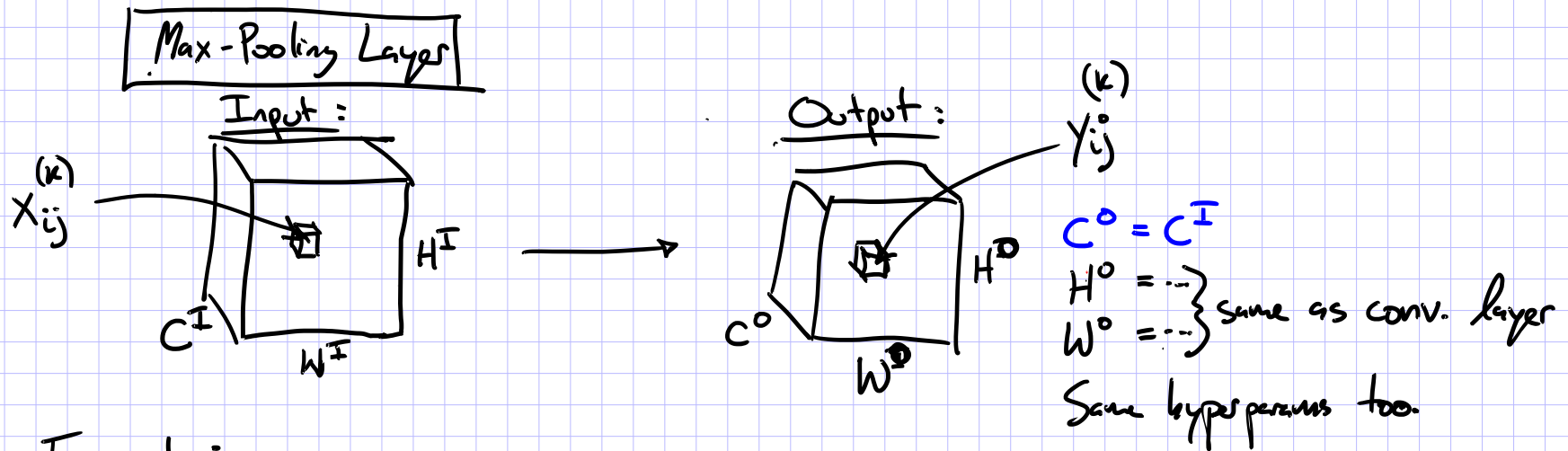
Max-Pooling Layer

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



Forward:

$$Y_{ij}^{(k)} = \max_{\substack{q \in \{1, \dots, k\} \\ r \in \{1, \dots, k\}}} X_{mn}^{(k)} \text{ where } \begin{cases} m = s(i-1) + q \\ n = s(j-1) + r \end{cases}$$

Backward:

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

Subderivatives

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

$$y = \max(a, b)$$

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

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

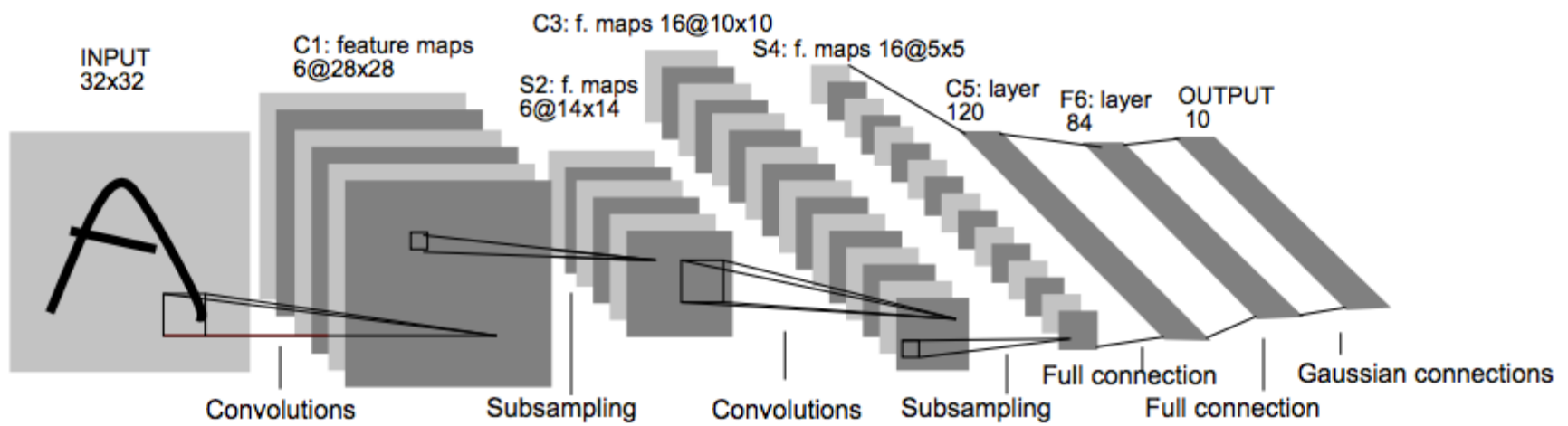


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Architecture #2: AlexNet

CNN for Image Classification

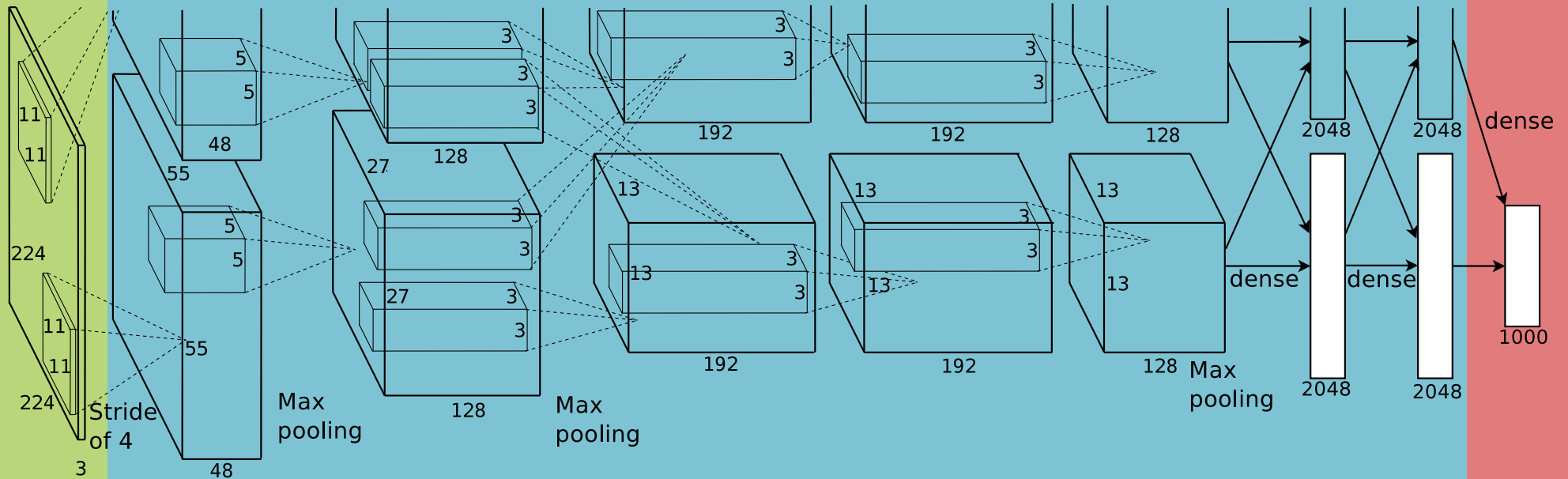
(Krizhevsky, Sutskever & Hinton, 2012)

15.3% error on ImageNet LSVRC-2012 contest

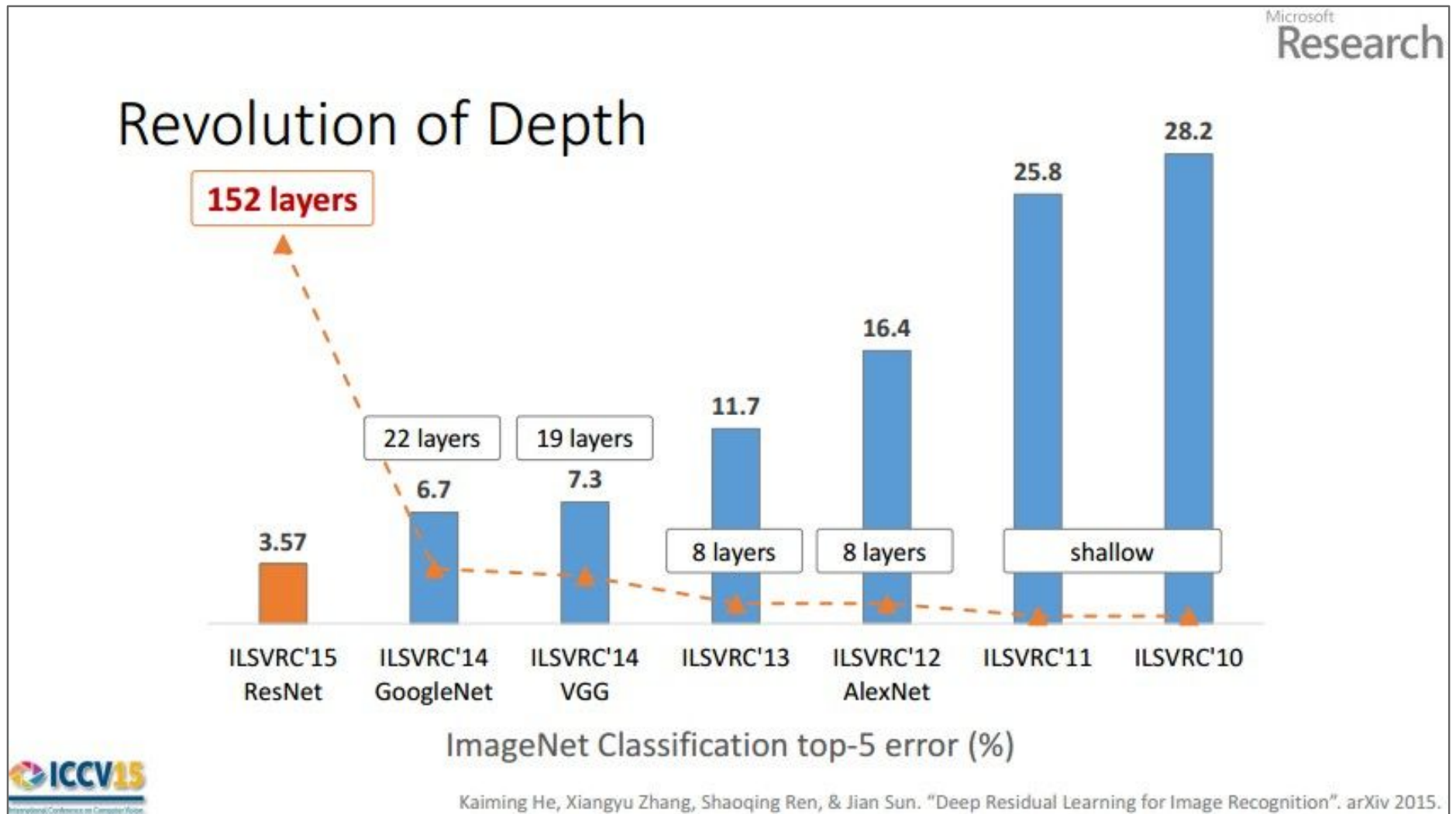
Input image (pixels)

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

1000-way softmax



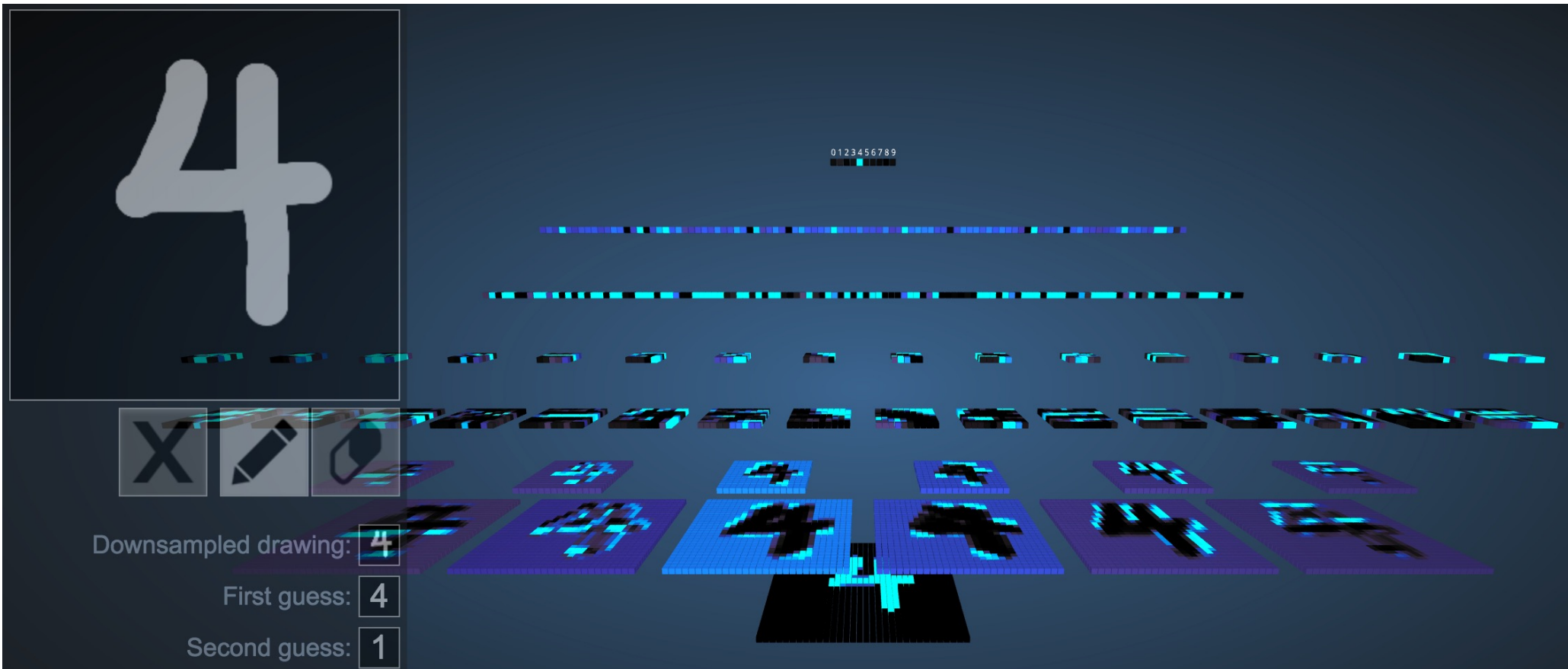
CNNs for Image Recognition



CNN VISUALIZATIONS

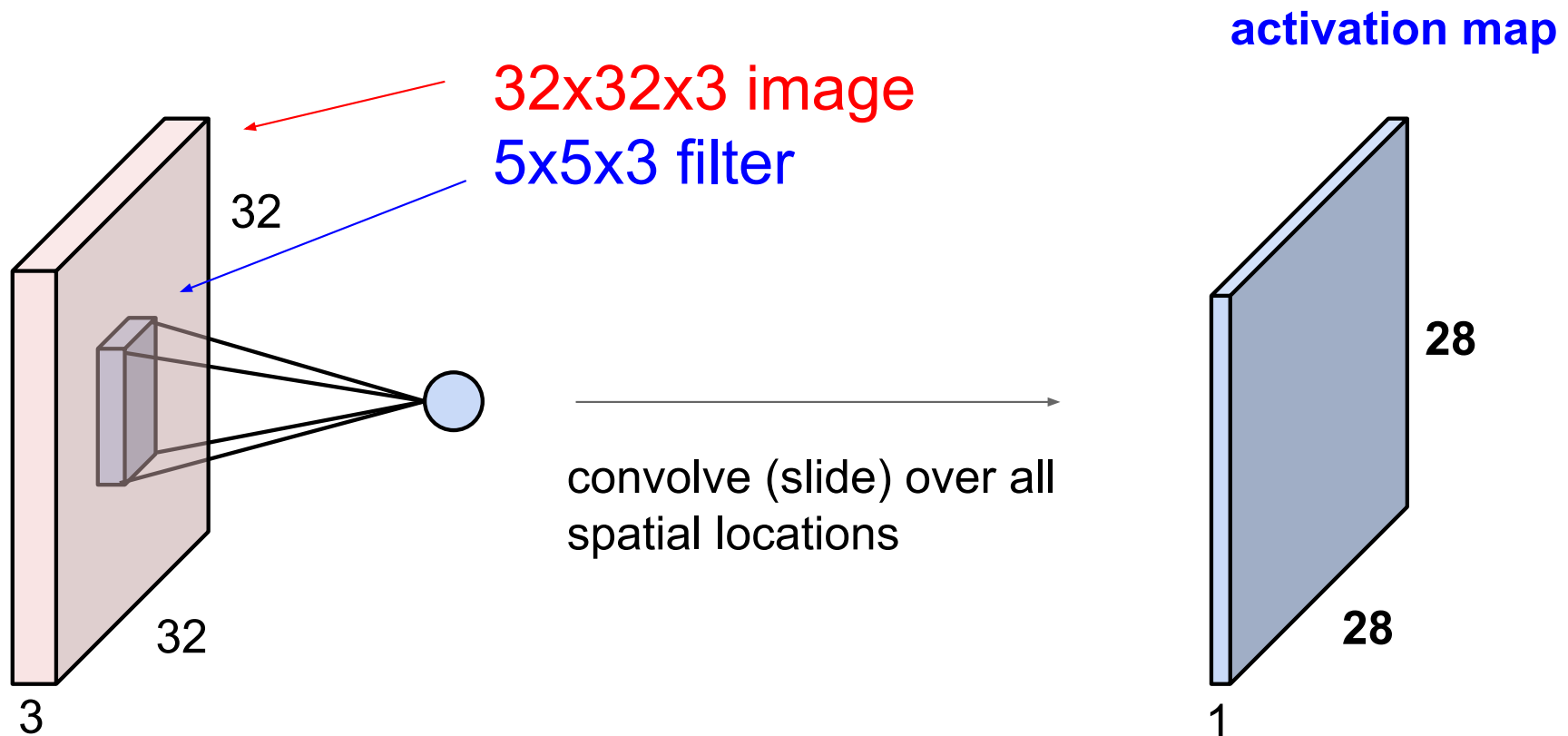
3D Visualization of CNN

<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



Animation of 3D Convolution

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

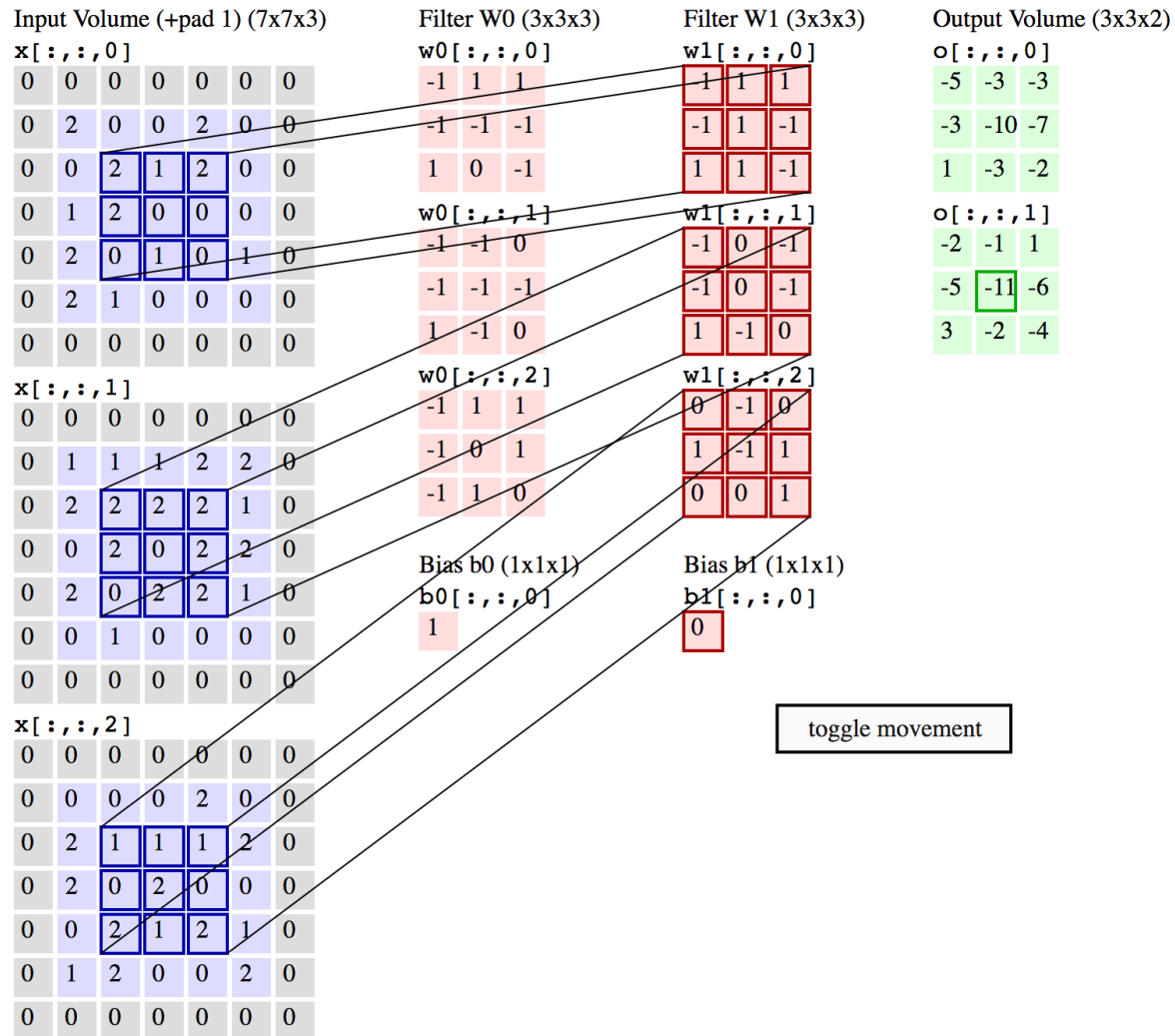


Figure from Fei-Fei Li & Andrej Karpathy & Justin Johnson (CS231N)

MNIST Digit Recognition with CNNs (in your browser)

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

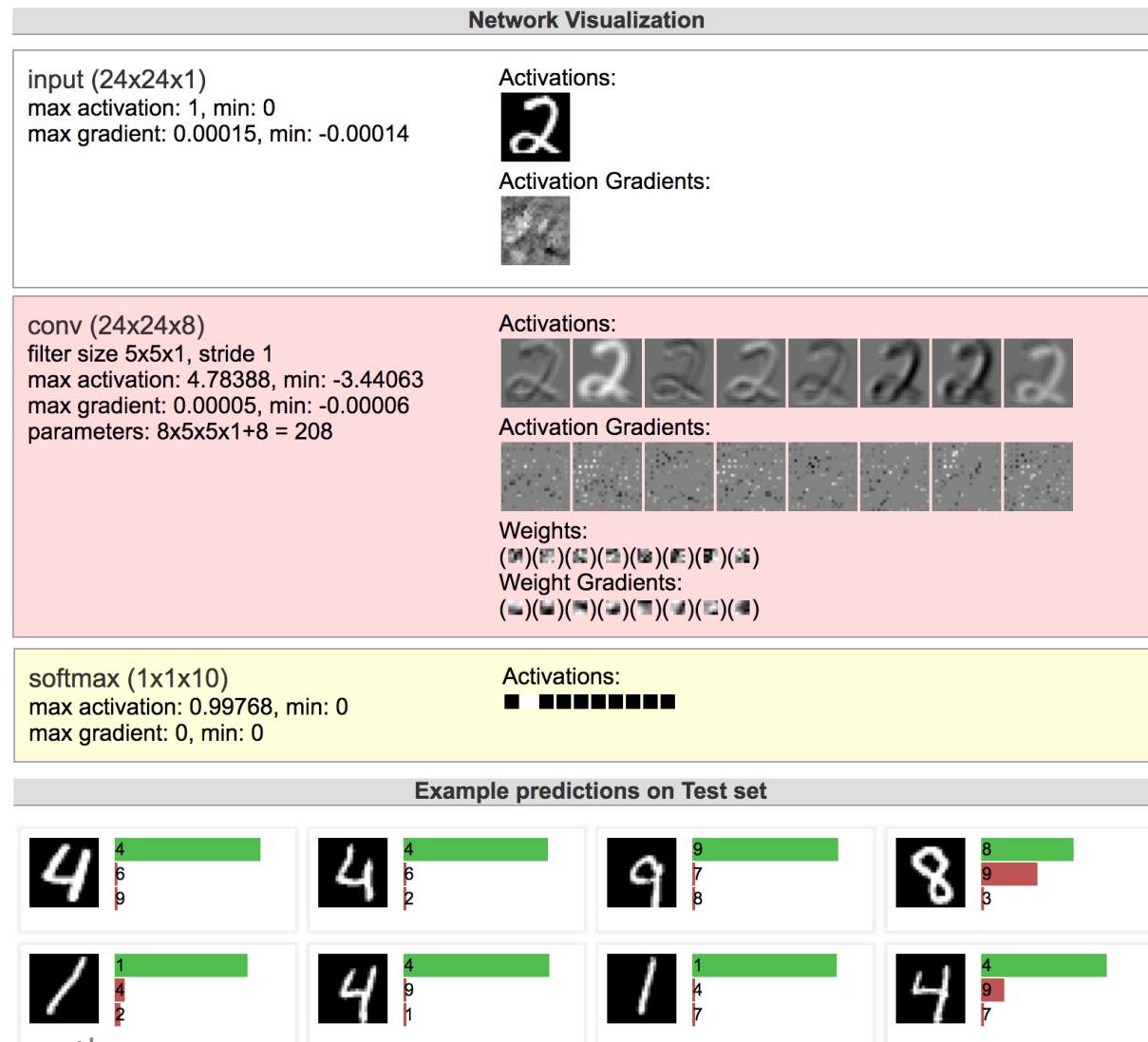


Figure from Andrej Karpathy

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/>

Deep Learning Objectives

You should be able to...

- Implement the common layers found in Convolutional Neural Networks (CNNs) such as linear layers, convolution layers, max-pooling layers, and rectified linear units (ReLU)
- Explain how the shared parameters of a convolutional layer could learn to detect spatial patterns in an image
- Describe the backpropagation algorithm for a CNN
- Identify the parameter sharing used in a basic recurrent neural network, e.g. an Elman network
- Apply a recurrent neural network to model sequence data
- Differentiate between an RNN and an RNN-LM

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Theoretical Foundations:

What principles guide learning?

- probabilistic
- information theoretic
- evolutionary search
- ML as optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. Data prep
2. Model selection
3. Training (optimization / search)
4. Hyperparameter tuning on validation data
5. (Blind) Assessment on test data

Big Ideas in ML:

Which are the ideas driving development of the field?

- inductive bias
- generalization / overfitting
- bias-variance decomposition
- generative vs. discriminative
- deep nets, graphical models
- PAC learning
- distant rewards

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search