



10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

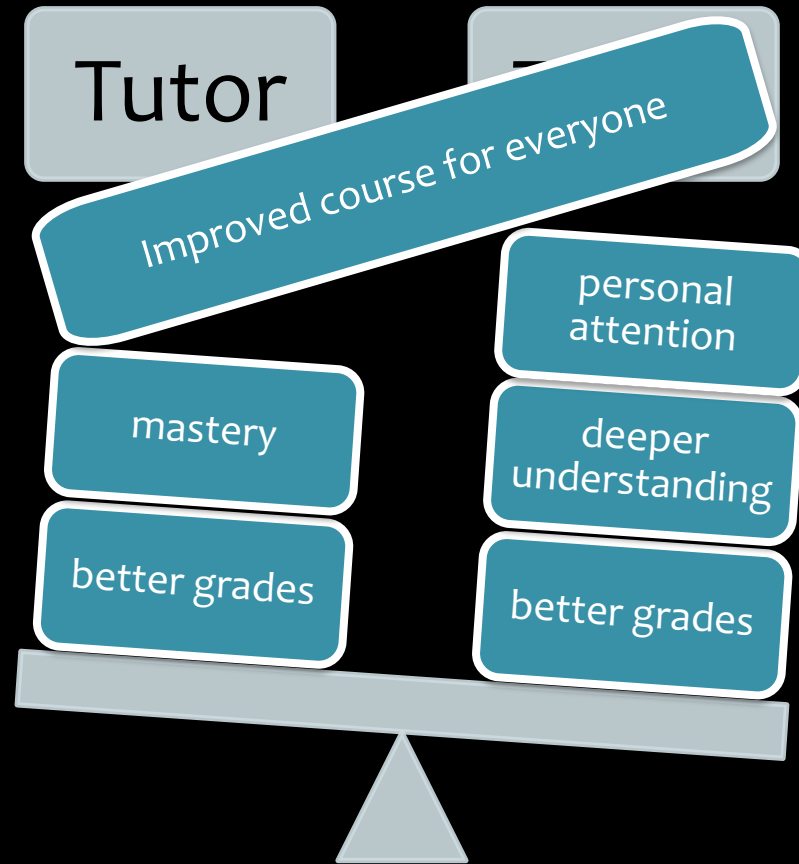
Foundations: RNNs & CNNs

Matt Gormley
Lecture 17
Oct. 30, 2023

Reminders

- **Homework 6: Learning Theory & Generative Models**
 - **Out: Fri, Oct 27**
 - **Due: Fri, Nov 3 at 11:59pm**

Peer Tutoring



DISCRIMINATIVE AND GENERATIVE CLASSIFIERS

Generative vs. Discriminative

- **Generative Classifiers:**

- Example: Naïve Bayes
- Define a joint model of the observations \mathbf{x} and the labels y : $p(\mathbf{x}, y)$
- Learning maximizes (joint) likelihood
- Use Bayes' Rule to classify based on the posterior:

$$p(y|\mathbf{x}) = p(\mathbf{x}|y)p(y)/p(\mathbf{x})$$

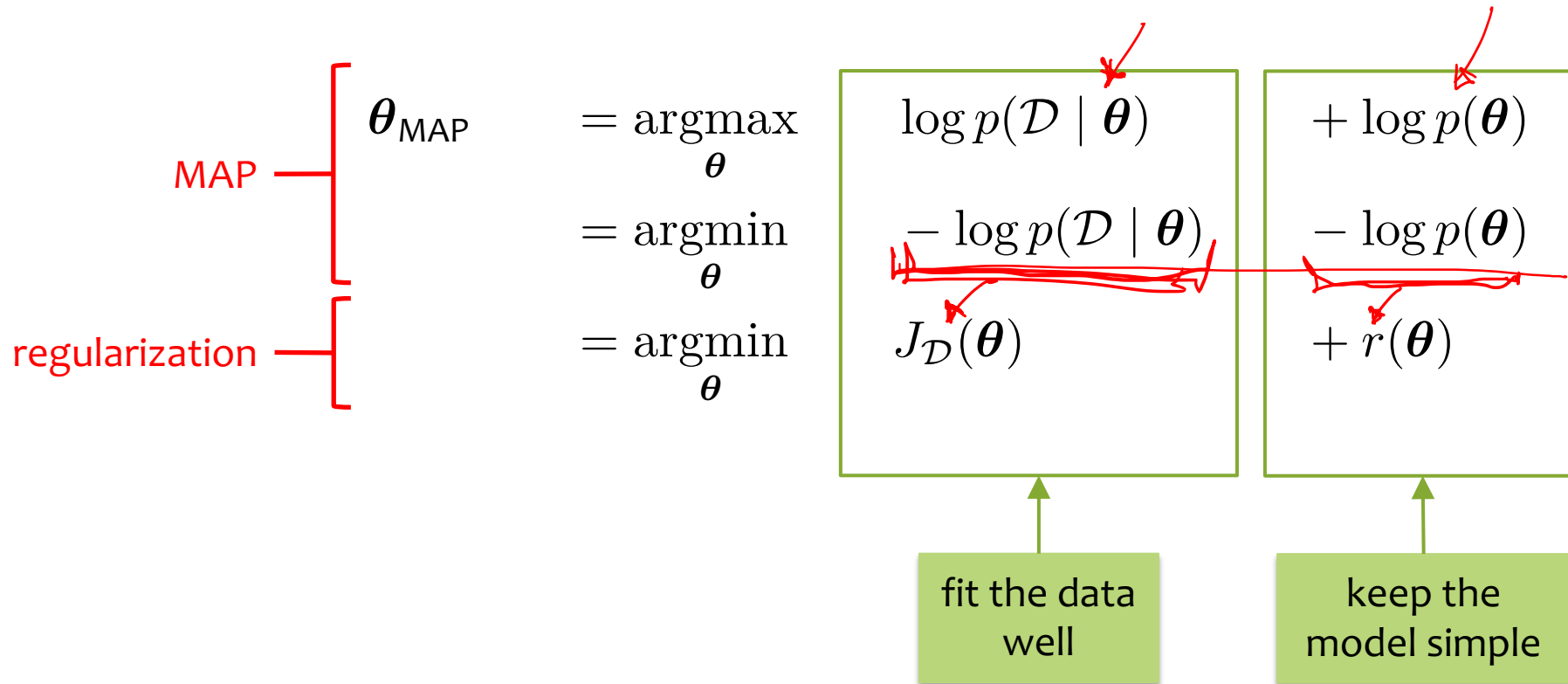
- **Discriminative Classifiers:**

- Example: Logistic Regression
- Directly model the conditional: $p(y|\mathbf{x})$
- Learning maximizes conditional likelihood

Generative vs. Discriminative

	Gen.	Disc.
MLE	$\prod_i p(\mathbf{x}^{(i)}, y^{(i)} \theta)$	$\prod_i p(y^{(i)} \mathbf{x}^{(i)}, \theta)$
MAP	$p(\theta) \prod_i p(\mathbf{x}^{(i)}, y^{(i)} \theta)$	$p(\theta) \prod_i p(y^{(i)} \mathbf{x}^{(i)}, \theta)$

MAP Estimation and Regularization



Example: L2 regularization is equivalent to a Gaussian prior

Generative vs. Discriminative

Finite Sample Analysis (Ng & Jordan, 2001)

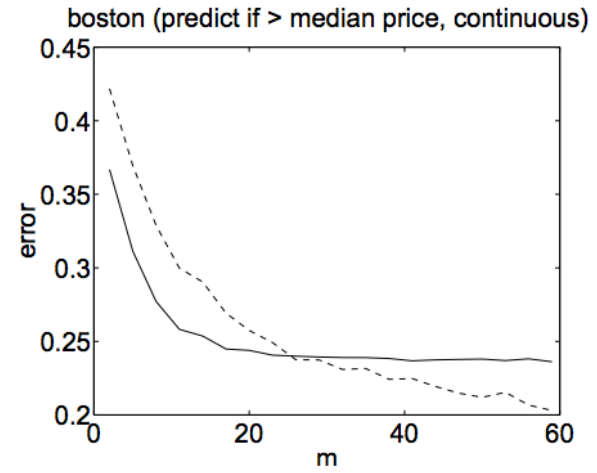
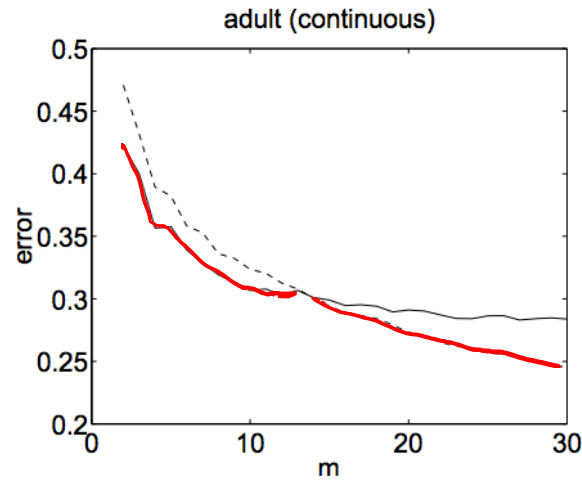
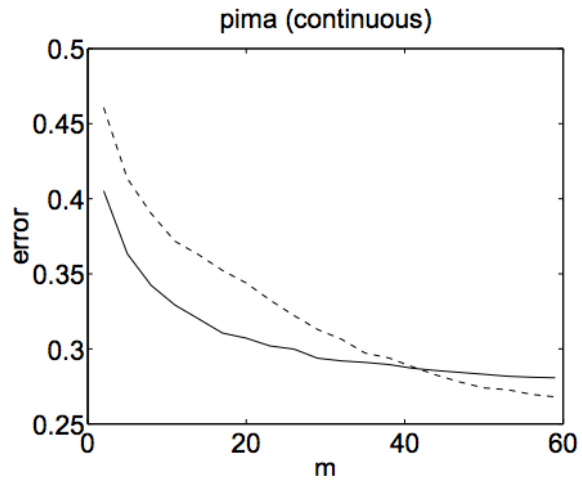
[Assume that we are learning from a finite training dataset]

Naïve Bayes and logistic regression form a *generative-discriminative* model pair:

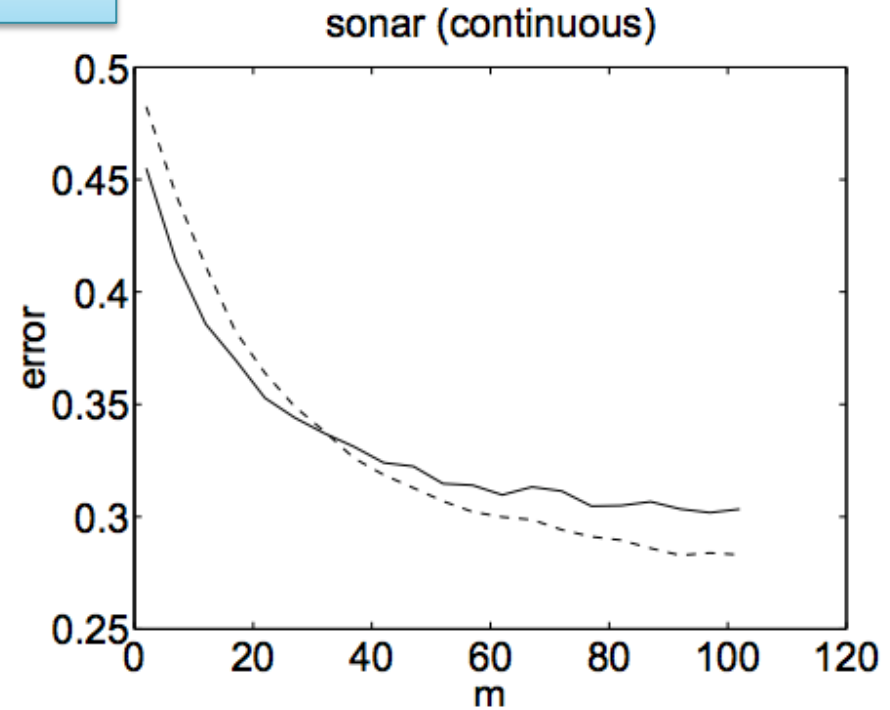
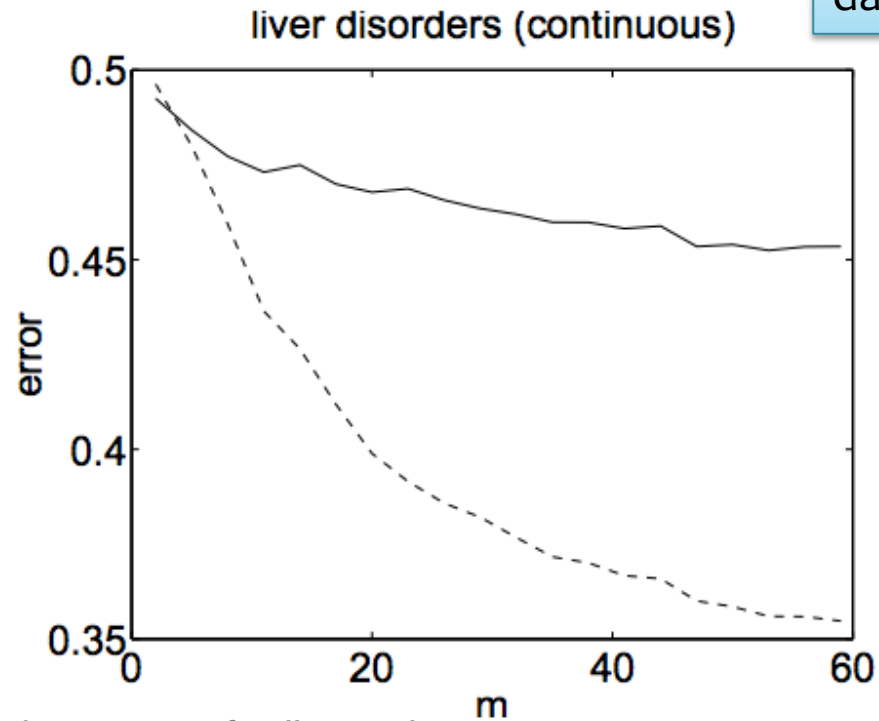
If model assumptions are correct: as the amount of training data increases, Gaussian Naïve Bayes and logistic regression approach the same (linear) decision boundary!

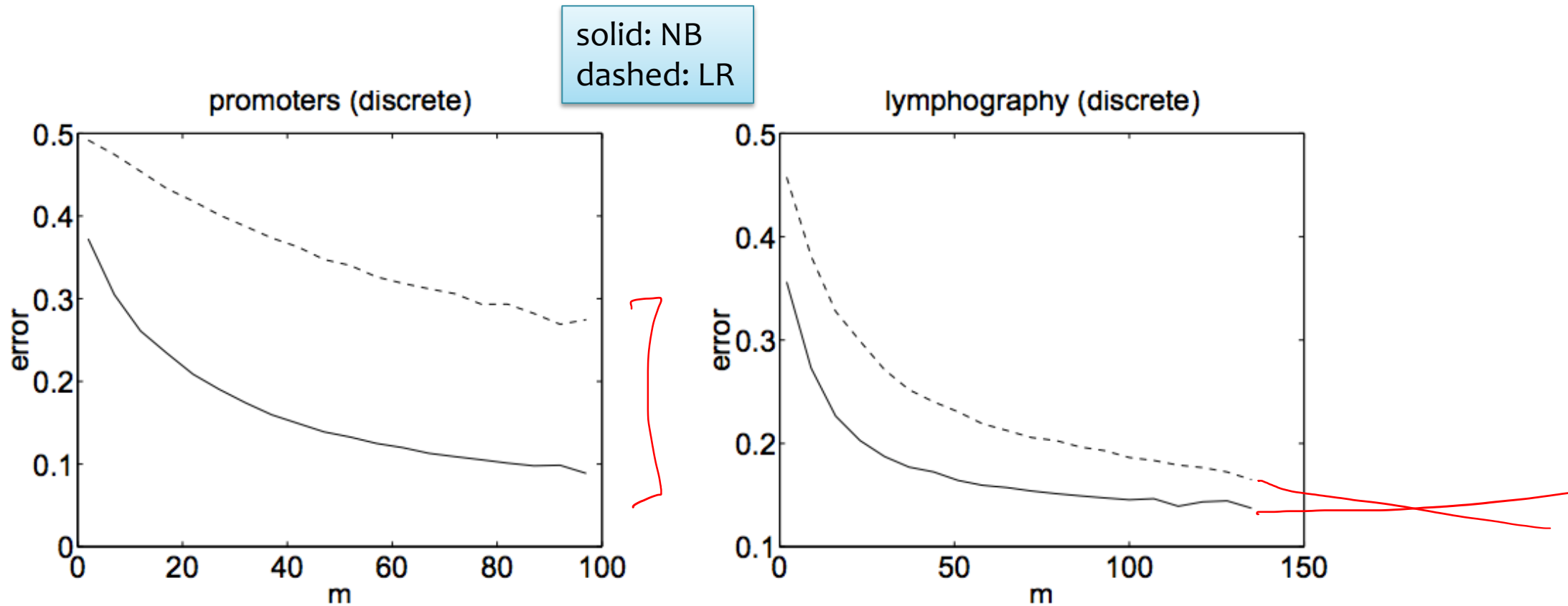
Furthermore, Gaussian Naïve Bayes is a more efficient learner (requires fewer samples) than Logistic Regression

If model assumptions are incorrect: Logistic Regression has lower asymptotic error and does better than Gaussian Naïve Bayes



solid: NB
dashed: LR





Naïve Bayes makes stronger assumptions about the data but needs fewer examples to estimate the parameters

“On Discriminative vs Generative Classifiers:” Andrew Ng and Michael Jordan, NIPS 2001.

Naïve Bayes vs. Logistic Reg.

Features

Naïve Bayes:

Features x are assumed to be conditionally independent given y . (i.e. Naïve Bayes Assumption)

Logistic Regression:

No assumptions are made about the form of the features x . They can be dependent and correlated in any fashion.

Naïve Bayes vs. Logistic Reg.

Learning (Parameter Estimation)

Naïve Bayes:

Parameters are decoupled → Closed form solution for MLE

Logistic Regression:

Parameters are coupled → No closed form solution – must use iterative optimization techniques instead

Naïve Bayes vs. Logistic Reg.

Learning (MAP Estimation of Parameters)

Bernoulli Naïve Bayes:

Parameters are probabilities \rightarrow Beta prior (usually) pushes probabilities away from zero / one extremes

Logistic Regression:

Parameters are not probabilities \rightarrow Gaussian prior encourages parameters to be close to zero

(effectively pushes the probabilities away from zero / one extremes)

Naïve Bayes vs. Logistic Regression

Question:

Q1

You just started working at a new company that manufactures comically large pennies. Your manager asks you to build a binary classifier that takes an image of a penny (on the factory assembly line) and predicts whether or not it has a defect.

What follow-up questions would you pose to your manager in order to decide between using a Naïve Bayes classifier and a Logistic Regression classifier?

Answer:

- are the defects correlated?
↳ or their feature rep?
- how many training pennies?
- how likely is a defect?
- how efficient does my need to be?

THE BIG PICTURE

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

Classification and Regression: The Big Picture

Recipe for Machine Learning

1. Given data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
2. (a) Choose a decision function $h_{\theta}(\mathbf{x}) = \dots$ (parameterized by θ)
(b) Choose an objective function $J_{\mathcal{D}}(\theta) = \dots$ (relies on data)
3. Learn by choosing parameters that optimize the objective $J_{\mathcal{D}}(\theta)$

$$\hat{\theta} \approx \underset{\theta}{\operatorname{argmin}} J_{\mathcal{D}}(\theta)$$

4. Predict on new test example \mathbf{x}_{new} using $h_{\theta}(\cdot)$

$$\hat{y} = h_{\theta}(\mathbf{x}_{\text{new}})$$

Optimization Method

- Gradient Descent: $\theta \rightarrow \theta - \gamma \nabla_{\theta} J(\theta)$
- SGD: $\theta \rightarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$
for $i \sim \text{Uniform}(1, \dots, N)$
where $J(\theta) = \frac{1}{N} \sum_{i=1}^N J^{(i)}(\theta)$
- mini-batch SGD
- closed form
 1. compute partial derivatives
 2. set equal to zero and solve

Decision Functions

- Perceptron: $h_{\theta}(\mathbf{x}) = \text{sign}(\theta^T \mathbf{x})$
- Linear Regression: $h_{\theta}(\mathbf{x}) = \theta^T \mathbf{x}$
- Discriminative Models: $h_{\theta}(\mathbf{x}) = \underset{y}{\operatorname{argmax}} p_{\theta}(y | \mathbf{x})$
 - Logistic Regression: $p_{\theta}(y = 1 | \mathbf{x}) = \sigma(\theta^T \mathbf{x})$
 - Neural Net (classification):
 $p_{\theta}(y = 1 | \mathbf{x}) = \sigma((\mathbf{W}^{(2)})^T \sigma((\mathbf{W}^{(1)})^T \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$
- Generative Models: $h_{\theta}(\mathbf{x}) = \underset{y}{\operatorname{argmax}} p_{\theta}(\mathbf{x}, y)$
 - Naive Bayes: $p_{\theta}(\mathbf{x}, y) = p_{\theta}(y) \prod_{m=1}^M p_{\theta}(x_m | y)$

Objective Function

- MLE: $J(\theta) = - \sum_{i=1}^N \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$
- MCLE: $J(\theta) = - \sum_{i=1}^N \log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)})$
- L2 Regularized: $J'(\theta) = J(\theta) + \lambda \|\theta\|_2^2$
(same as Gaussian prior $p(\theta)$ over parameters)
- L1 Regularized: $J'(\theta) = J(\theta) + \lambda \|\theta\|_1$
(same as Laplace prior $p(\theta)$ over parameters)

Backpropagation and Deep Learning

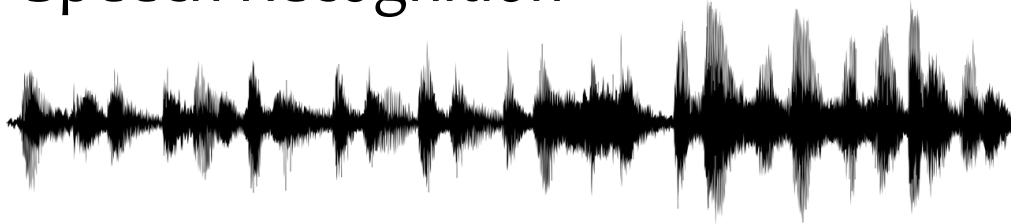
Convolutional neural networks (CNNs) and **recurrent neural networks (RNNs)** are simply fancy computation graphs (aka. hypotheses or decision functions).

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

BACKGROUND: HUMAN LANGUAGE TECHNOLOGIES

Human Language Technologies

Speech Recognition



Machine Translation

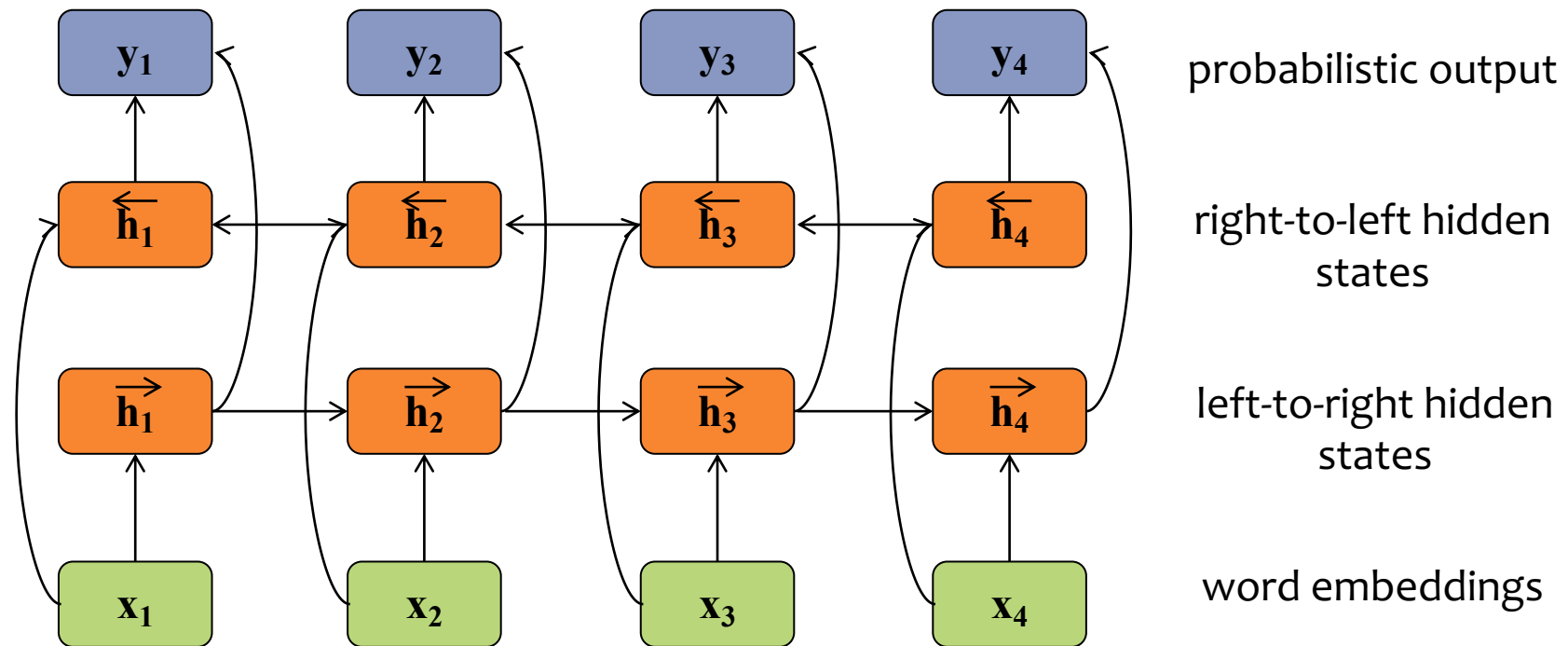
기계 번역은 특히 영어와 한국어와 같은 언어 쌍의 경우 매우 어렵습니다.

Summarization

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

Bidirectional RNN

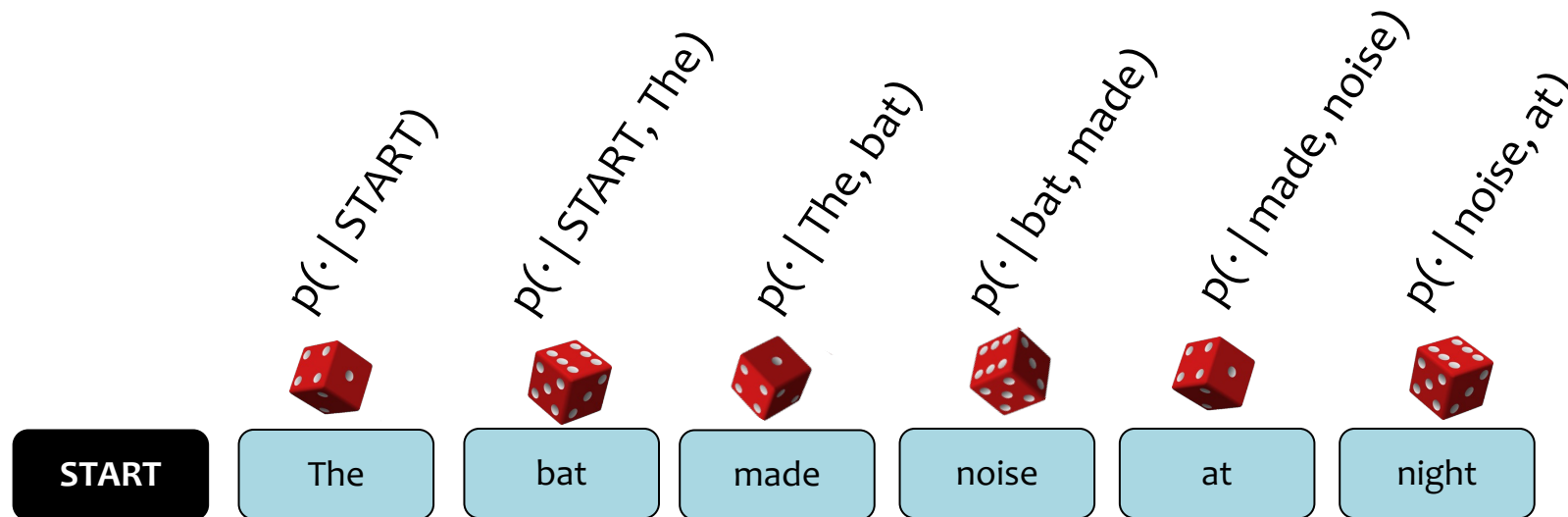
RNNs are a now commonplace backbone in deep learning approaches to natural language processing



BACKGROUND: N-GRAM LANGUAGE MODELS

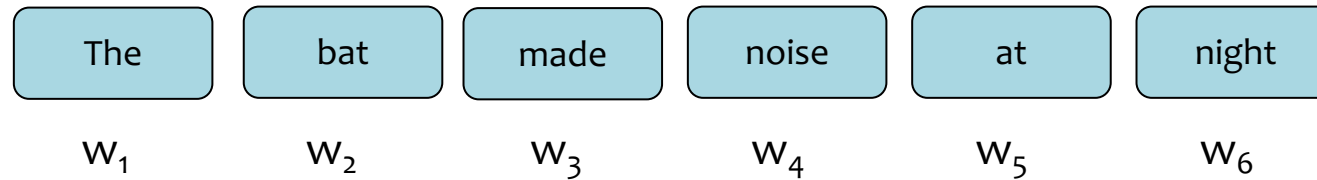
n-Gram Language Model

- 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

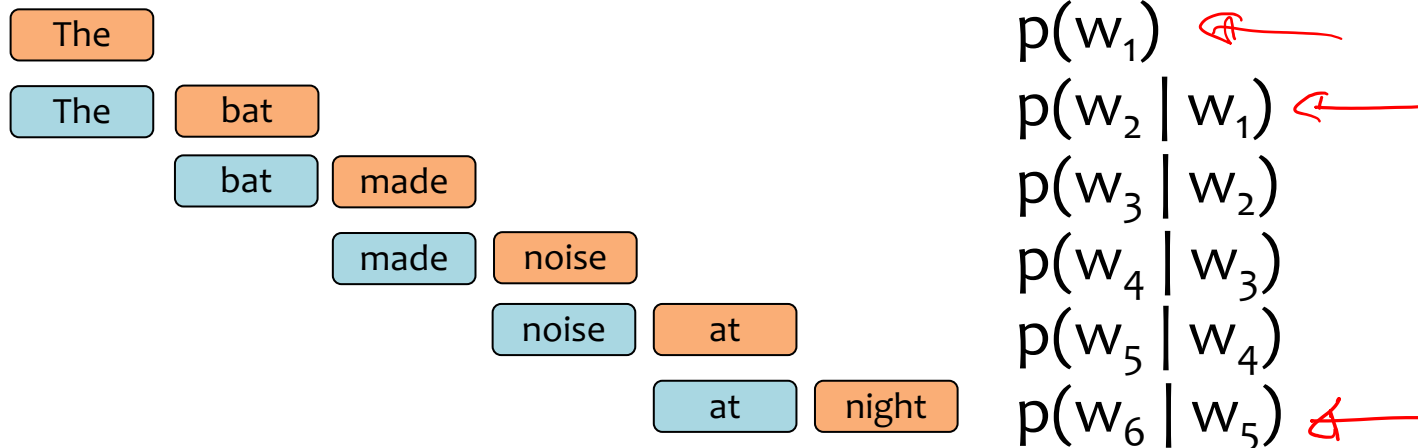
Question: How can we **define** a probability distribution over a sequence of length T?



n-Gram Model (n=2)

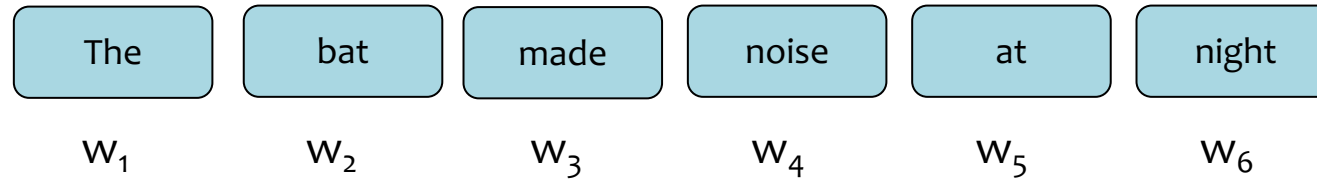
$$p(w_1, w_2, \dots, w_T) = \prod_{t=1}^T p(w_t | w_{t-1})$$

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



n-Gram Language Model

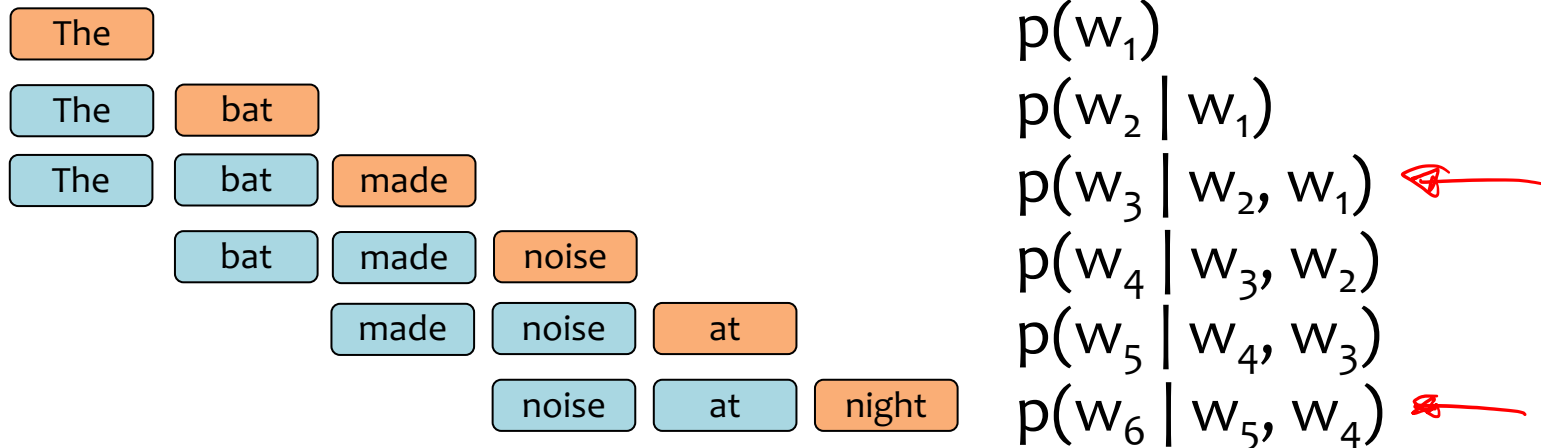
Question: How can we **define** a probability distribution over a sequence of length T?



n-Gram Model (n=3)

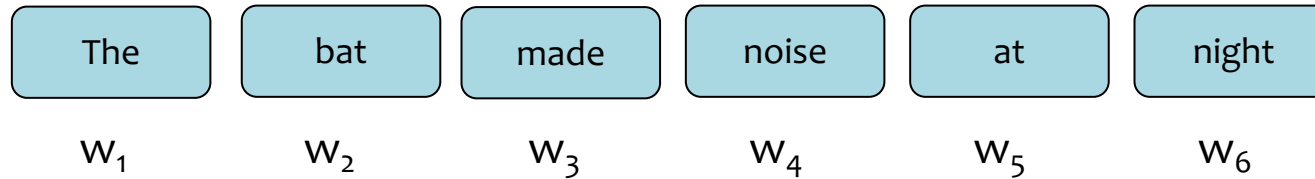
$$p(w_1, w_2, \dots, w_T) = \prod_{t=1}^T p(w_t | w_{t-1}, w_{t-2})$$

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



n-Gram Language Model

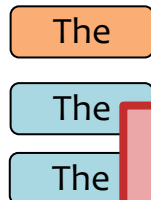
Question: How can we **define** a probability distribution over a sequence of length T?



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?

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


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?...

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



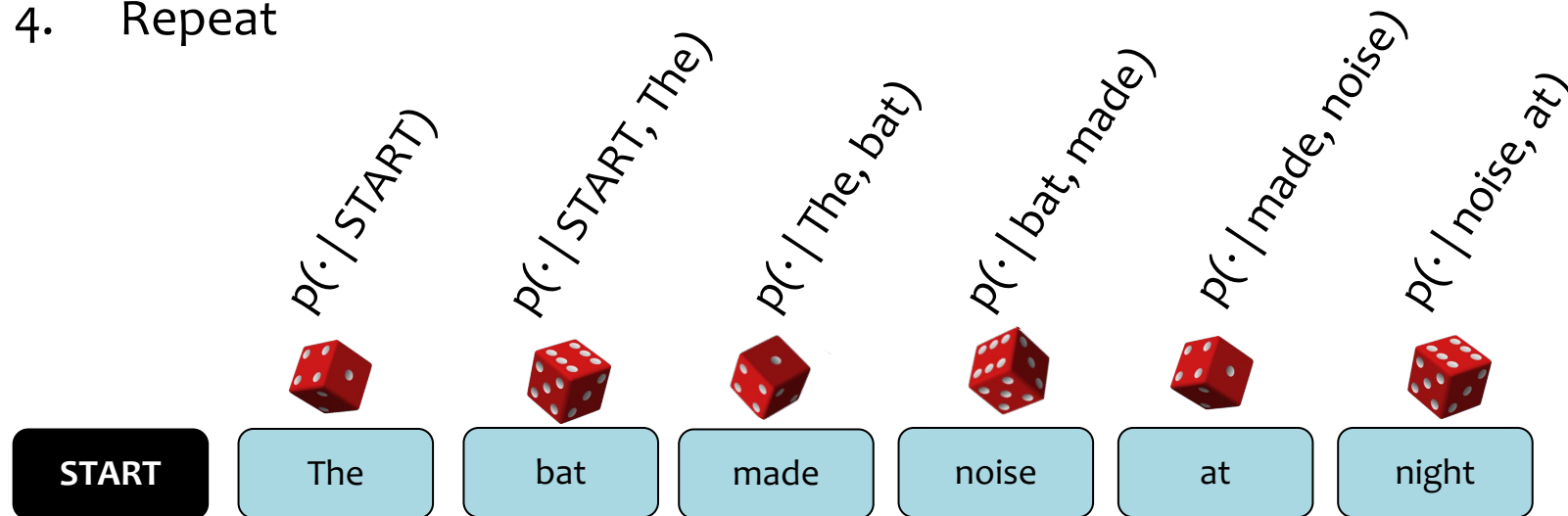
w_t	$p(\cdot \mid \cdot, \cdot)$
corn	4/11 ←
grass	3/11 ↗
hay	2/11
if	1/11
which	1/11

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 $p(w_t | w_{t-2}, w_{t-1})$
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 $p(w_t | w_{t-2}, w_{t-1})$
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, deny. 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: $\mathbf{h} = (h_1, h_2, \dots, h_T), h_i \in \mathcal{R}^J$

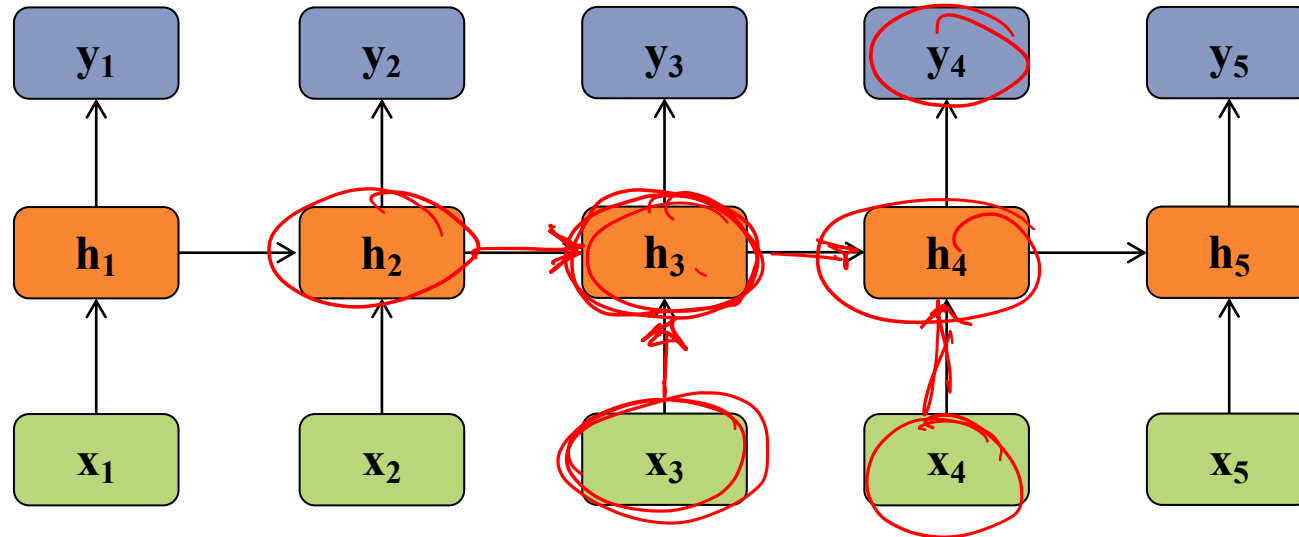
outputs: $\mathbf{y} = (y_1, y_2, \dots, y_T), y_i \in \mathcal{R}^K$

nonlinearity: \mathcal{H}

Definition of the RNN:

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

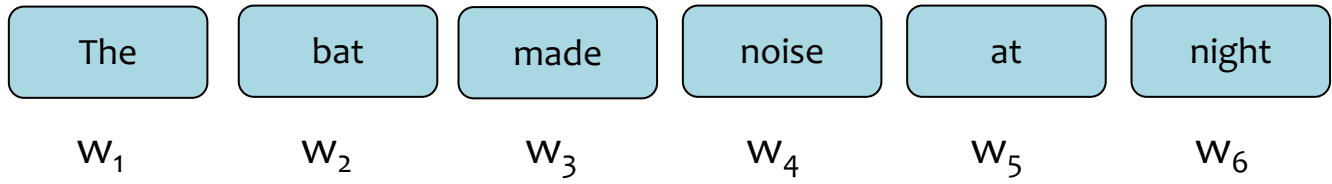
$$y_t = W_{hy}h_t + b_y$$



Recall...

The Chain Rule of Probability

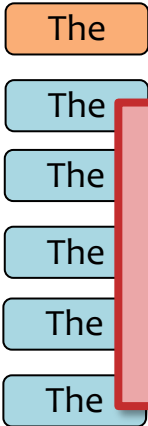
Question: How can we **define** a probability distribution over a sequence of length T?



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

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

$$p(w_1) \cdot p(w_2 | w_1) \cdot p(w_3 | w_2, w_1) \cdot \dots$$



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

$$p(w_6 | w_5, w_4, w_3, w_2, w_1)$$

RNN Language Model

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

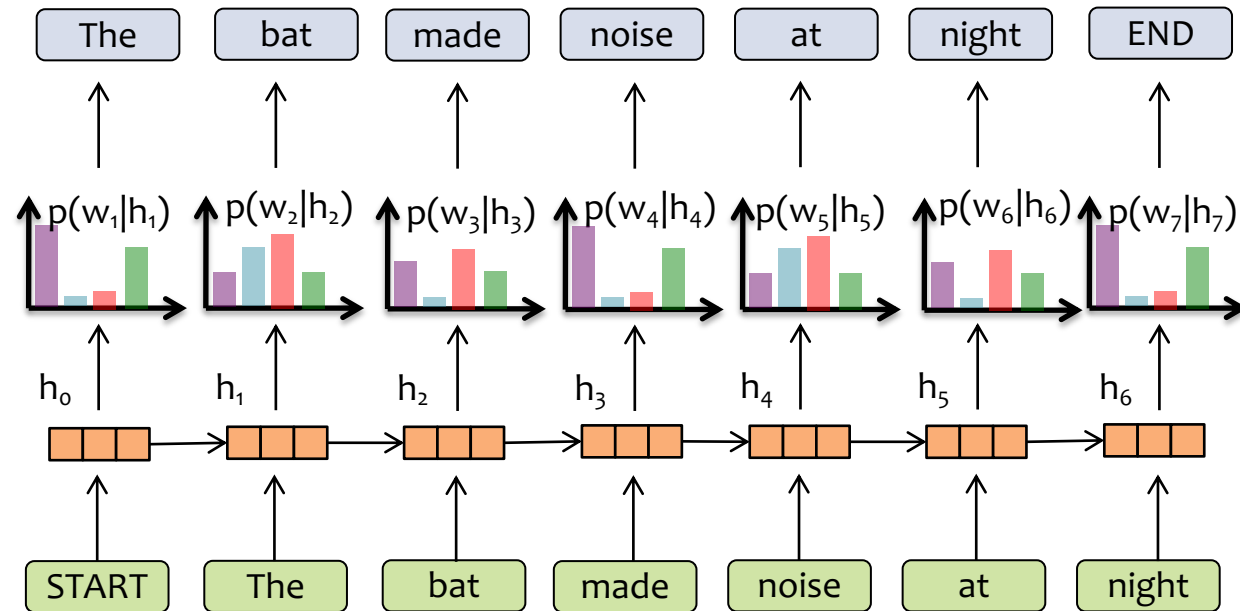
$$p(w_1, w_2, w_3, \dots, w_6) =$$

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

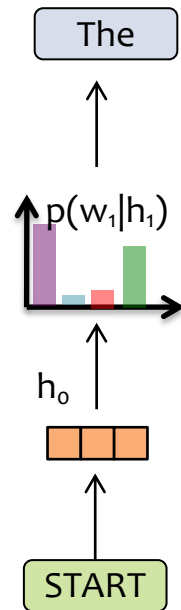
RNN Language Model



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 $\mathbf{h}_t = f_{\theta}(w_{t-1}, \dots, w_1)$

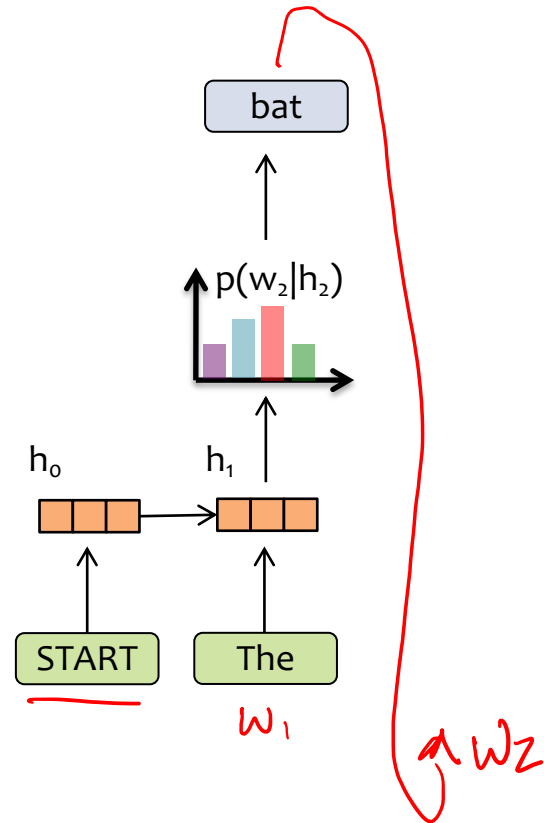
RNN Language Model



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 $\mathbf{h}_t = f_{\theta}(w_{t-1}, \dots, w_1)$

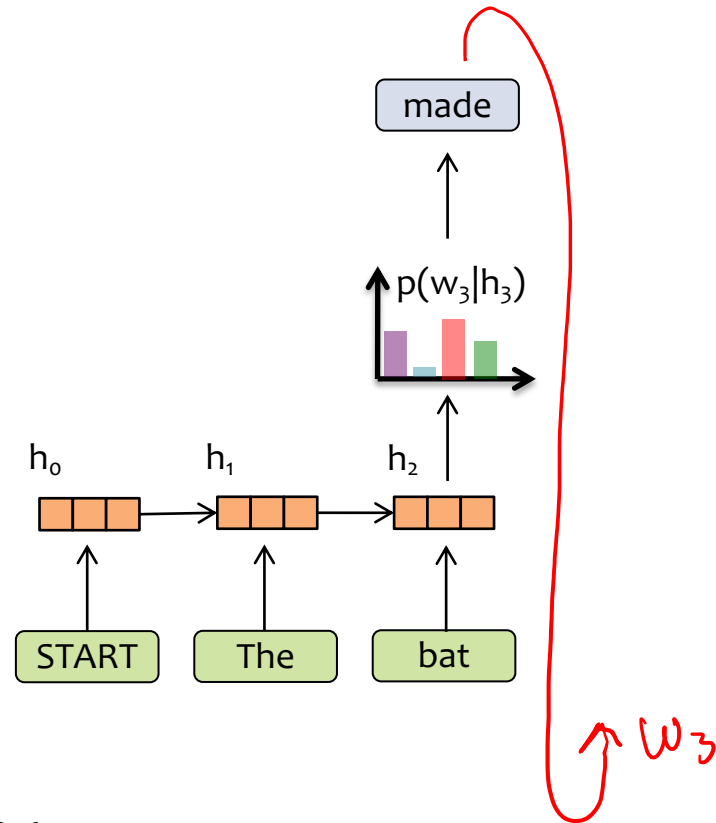
RNN Language Model



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 $\mathbf{h}_t = f_{\theta}(w_{t-1}, \dots, w_1)$

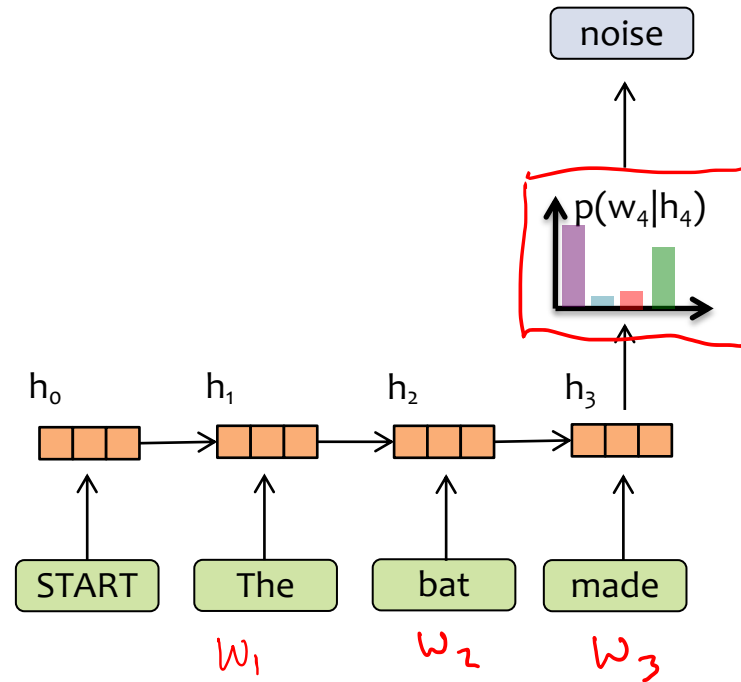
RNN Language Model



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 $\mathbf{h}_t = f_{\theta}(w_{t-1}, \dots, w_1)$

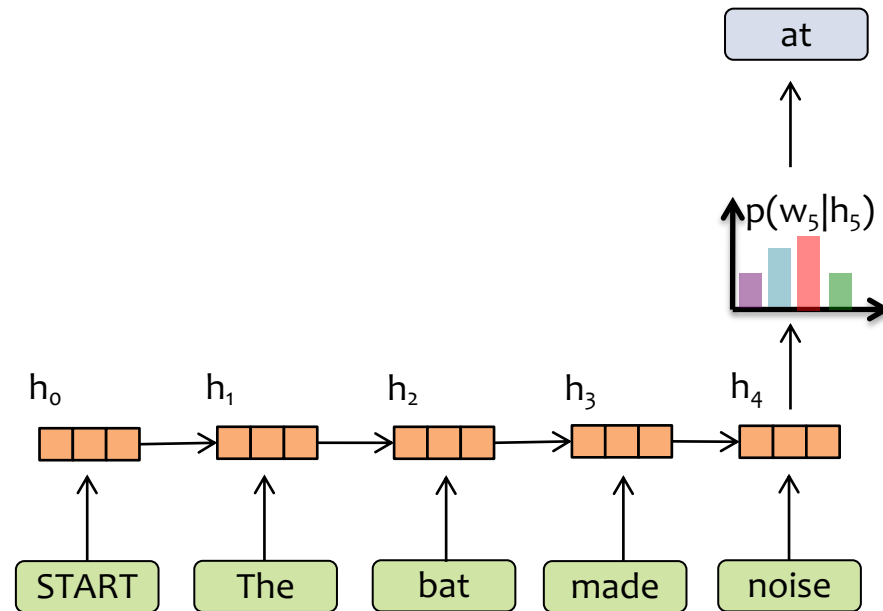
RNN Language Model



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

Q2

Question: How can we create a distribution $p(w_t|h_t)$ from h_t ?

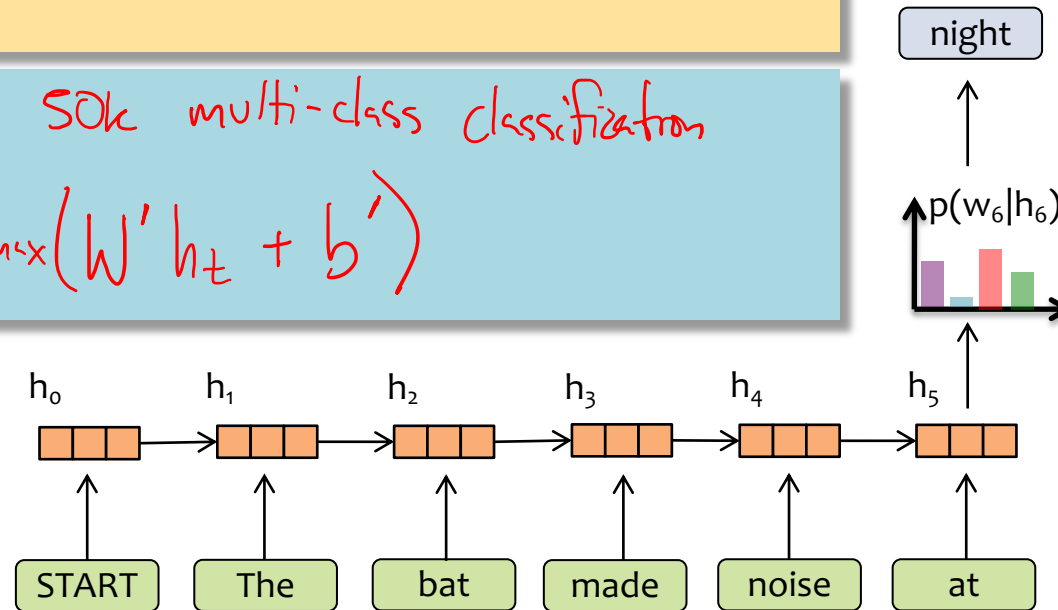
Answer: Treat as 50k multi-class classification

$$p(w_t|h_t) = \text{softmax}(W'h_t + b')$$

$$h_t \in \mathbb{R}^J$$

$$W' \in \mathbb{R}^{50k \times J}$$

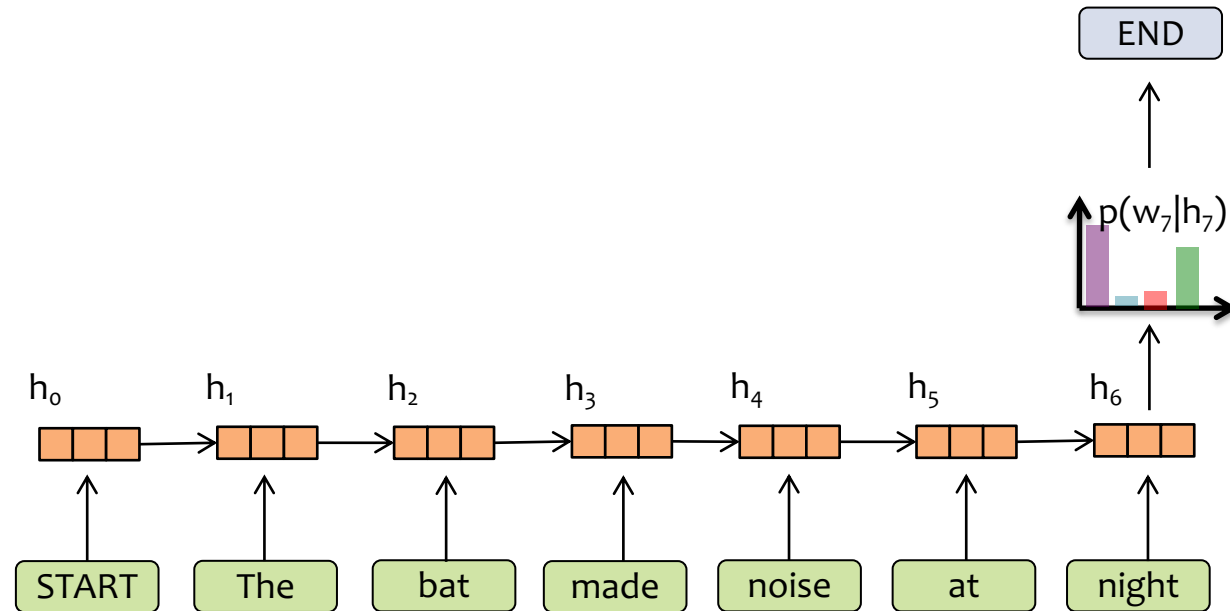
$$b \in \mathbb{R}^{50k}$$



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 $\mathbf{h}_t = f_{\theta}(w_{t-1}, \dots, w_1)$

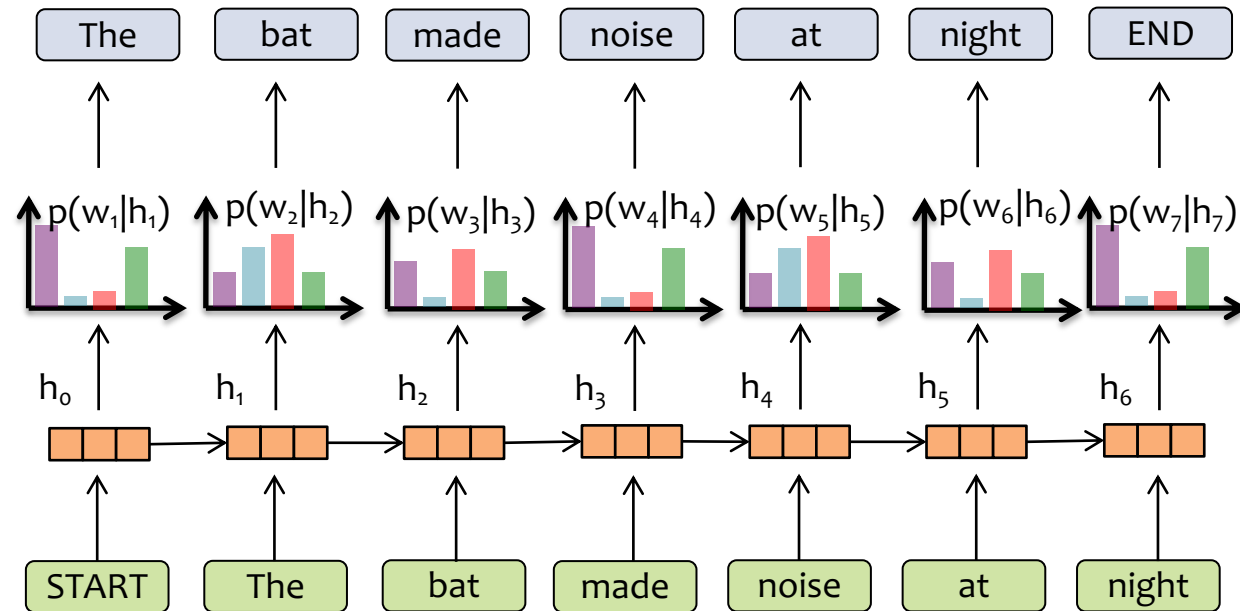
RNN Language Model



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



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

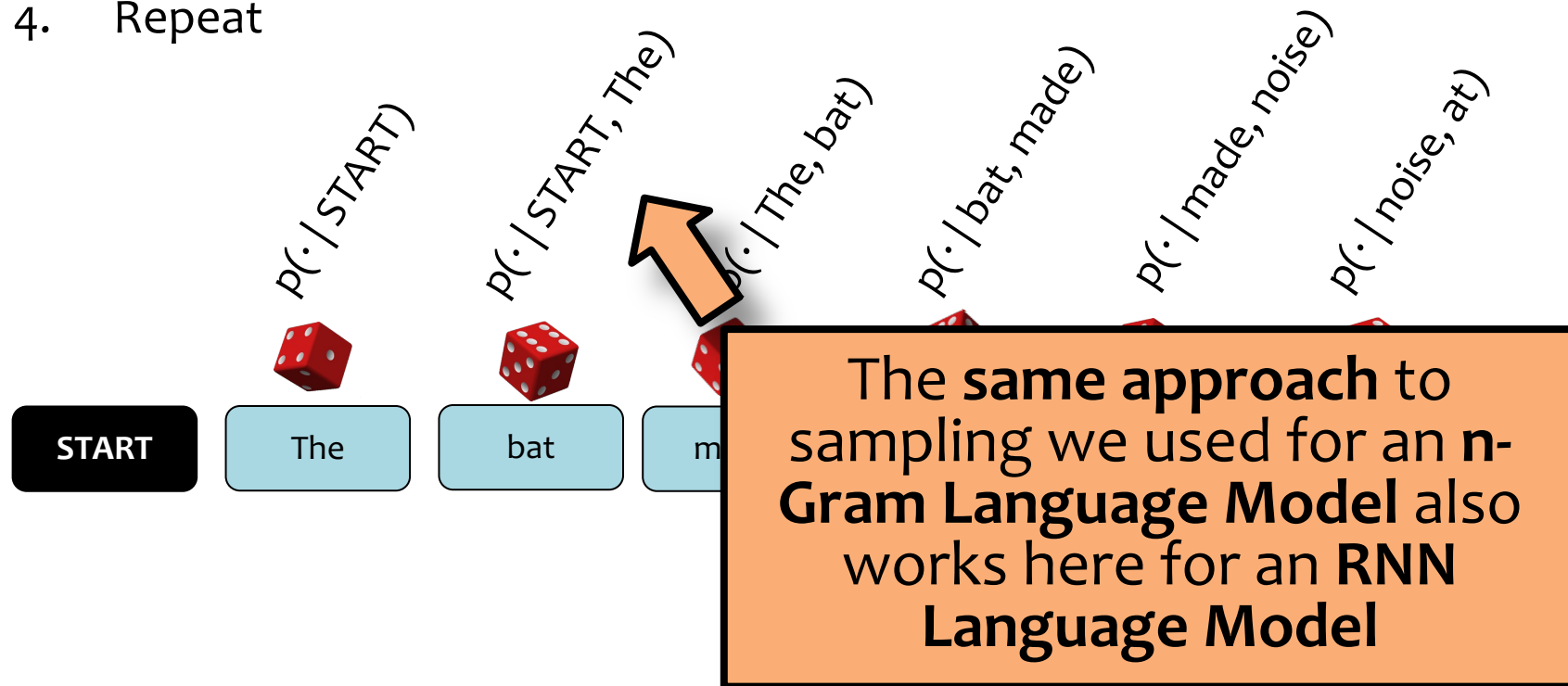
The bat night

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 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 m
there My power to give thee but so much
service in the noble bondman here, Would
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.

??

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
tender; and, for your love, I would be
as I must, for my own honour, if he
fore, out of 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 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 and by thy m
there My power to give thee but so much
service in the noble bondman here, Would
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.

??

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
tender; and, for your love, I would be
as I must, for my own honour, if he
fore, out of 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 is a thing of his own search and
altogether against my will.

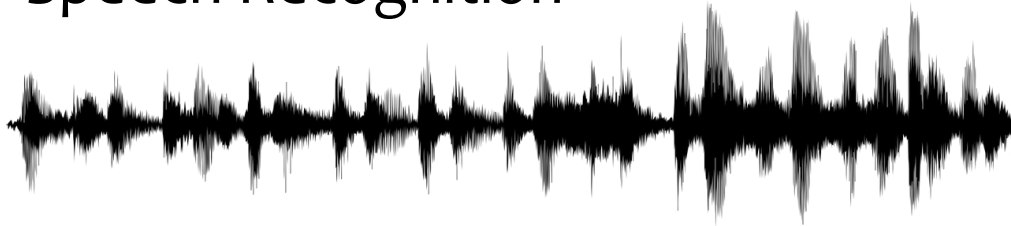
TOUCHSTONE: For my part, I had rather bear with you
than bear you; yet I should bear no cross if I did bear you,
for I think you have no money in your purse.

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

Sequence to Sequence Model

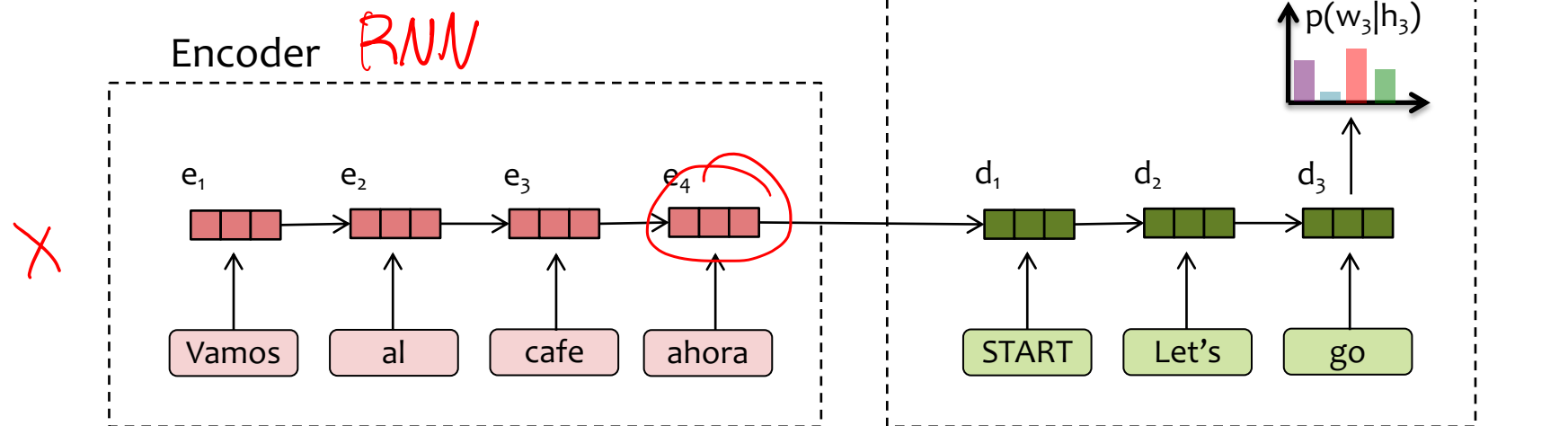
Now suppose you want generate a sequence conditioned on another input

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



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

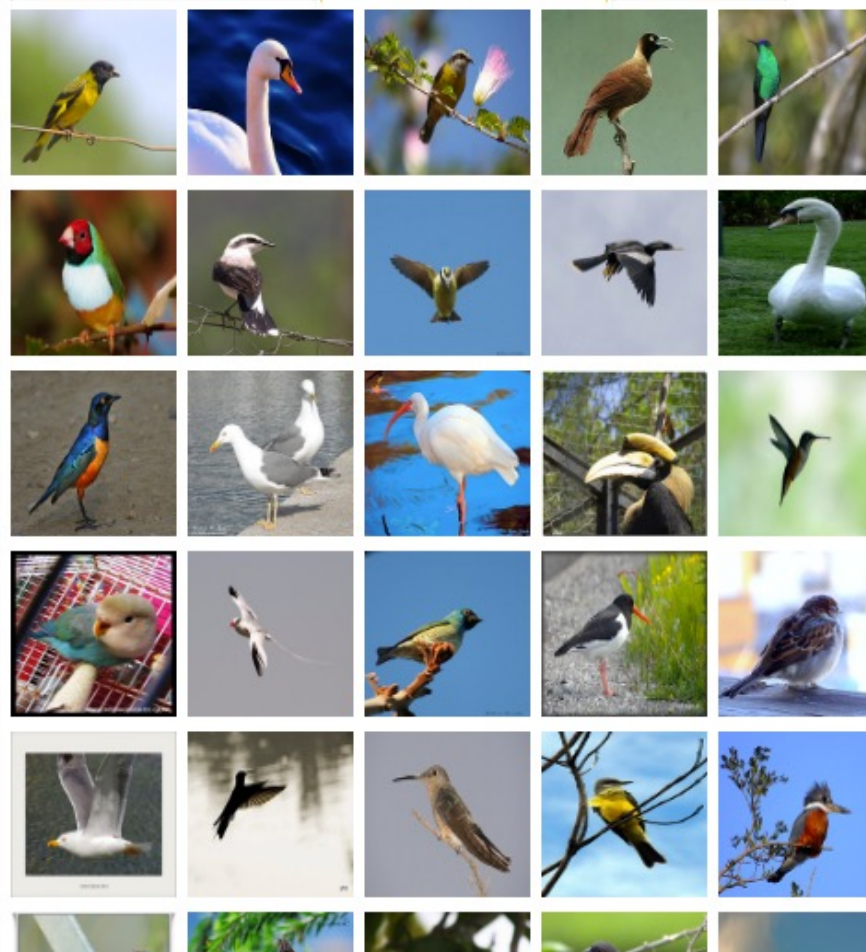


- marine animal, marine creature, sea animal, sea creature (1)
- scavenger (1)
- biped (0)
- predator, predatory animal (1)
- larva (49)
- acrodont (0)
- feeder (0)
- stunt (0)
- chordate (3087)**
 - tunicate, urochordate, urochord (6)
 - cephalochordate (1)
 - vertebrate, craniate (3077)
 - mammal, 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)
 - Ibero-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



- ... 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)

Treemap Visualization

Images of the Synset

Downloads



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



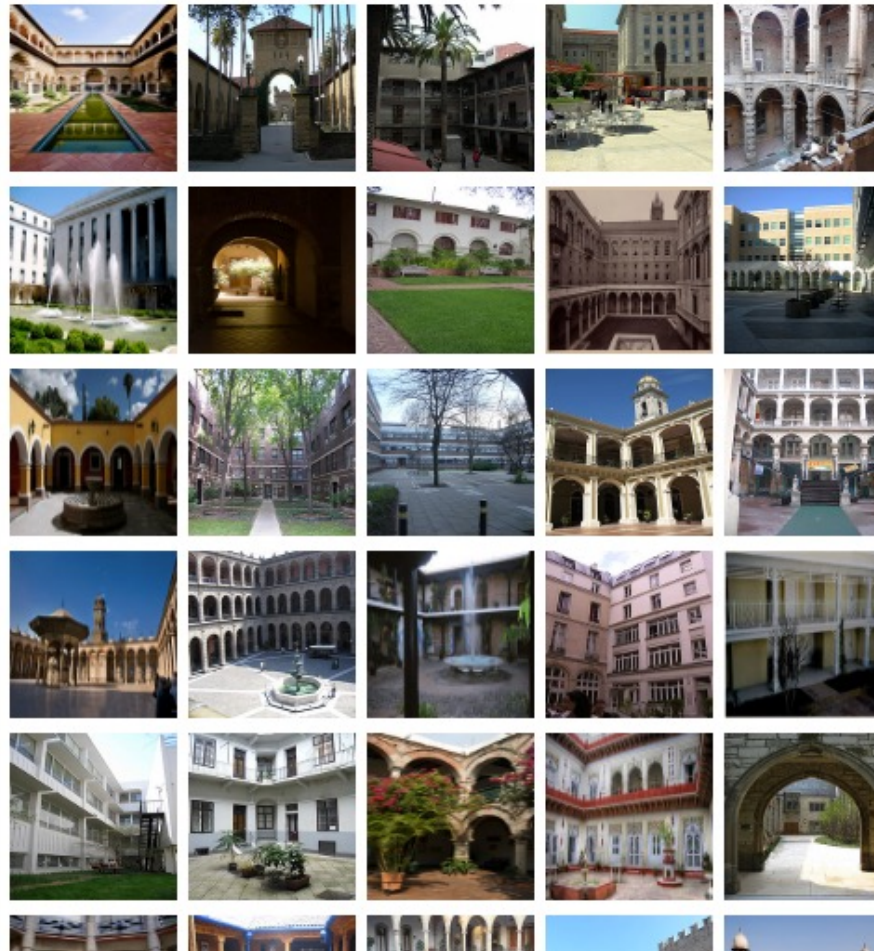
Numbers in brackets: (the number of synsets in the subtree).

- 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

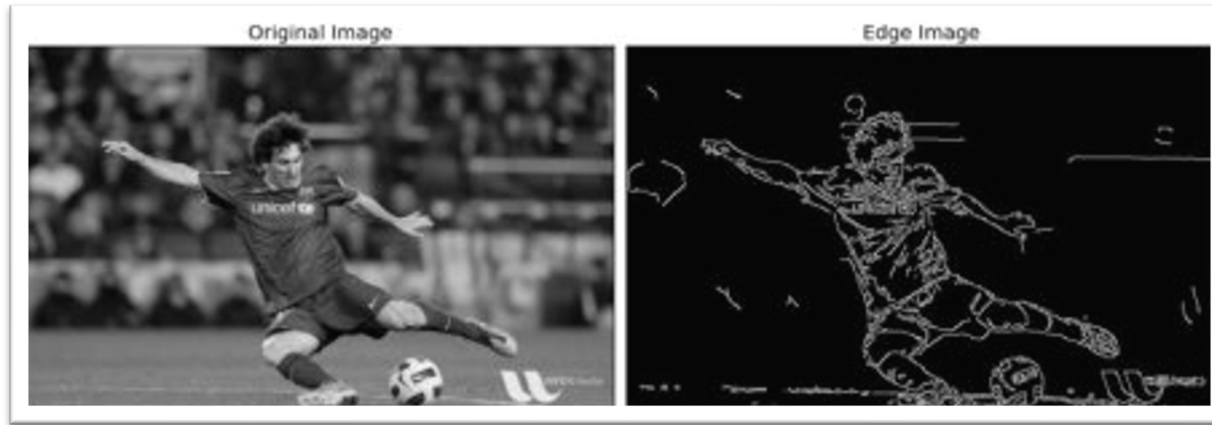
Images of the Synset

Downloads

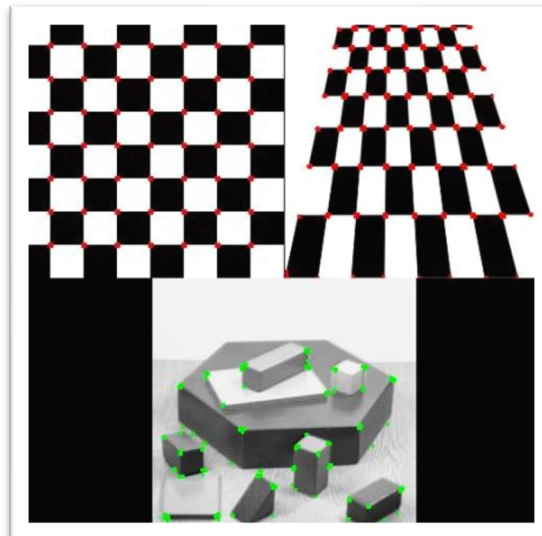


Feature Engineering for CV

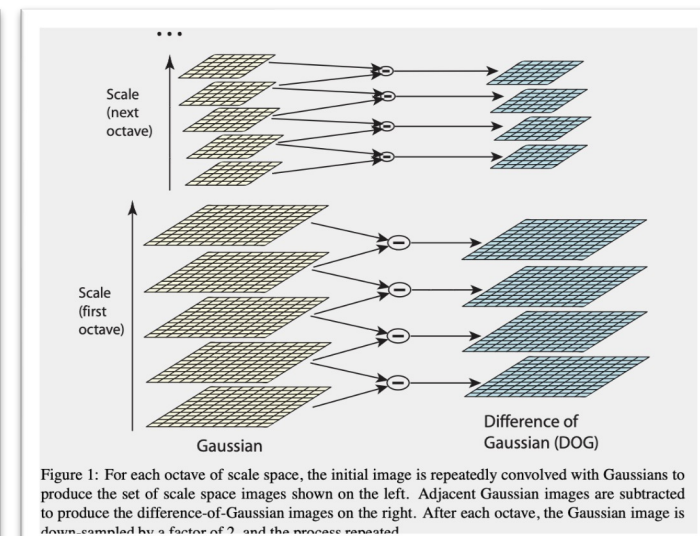
Edge detection (Canny)



Corner Detection (Harris)



Scale Invariant Feature Transform (SIFT)



Example: Image Classification

CNN for Image Classification

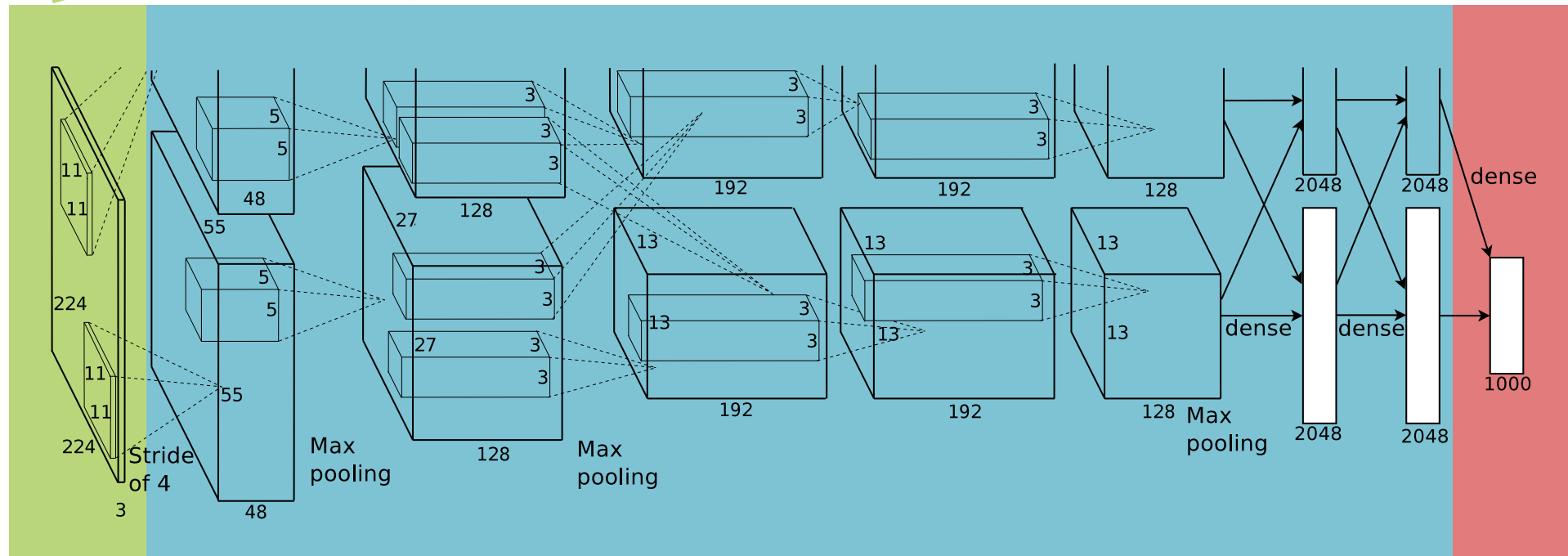
(Krizhevsky, Sutskever & Hinton, 2012)

15.3% error on ImageNet LSVRC-2012 contest

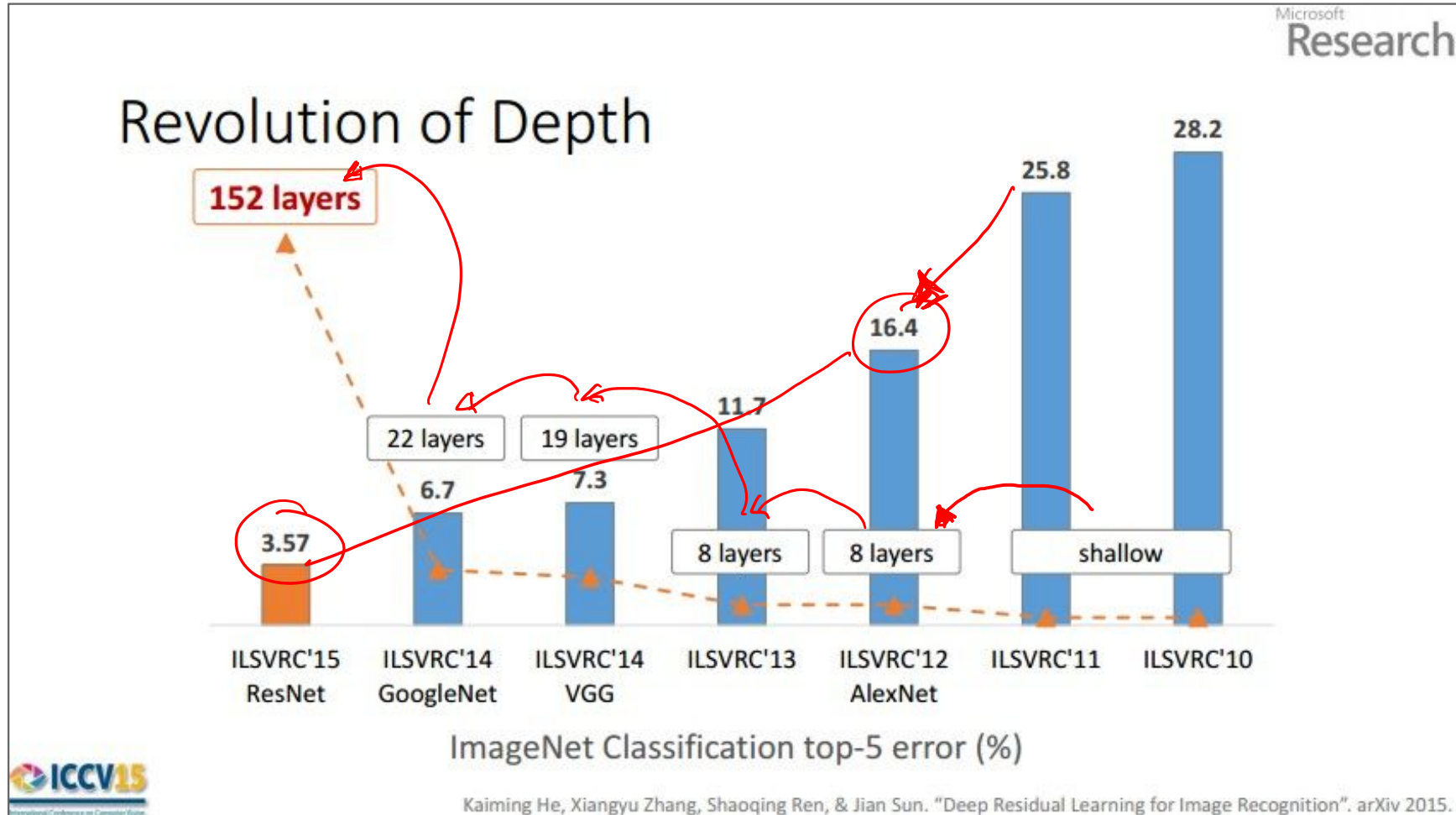
Input image (pixels)

- 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).

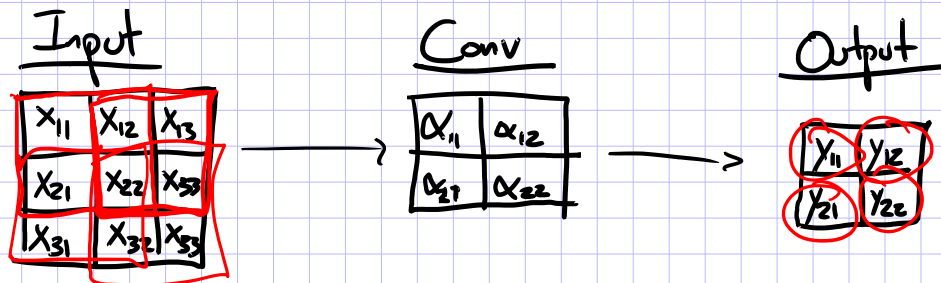
Our recipe also applies to these models and (again) relies on the **backpropagation algorithm** to compute the necessary gradients.

CONVOLUTION

What's a convolution?

- Basic idea:
 - Pick a 3x3 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



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

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

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

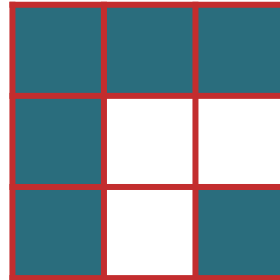
Background: Image Processing

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

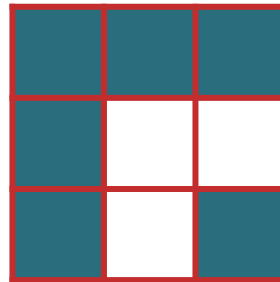
Background: Image Processing

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
	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	0	0
0	0	1	1	1	1	0
0	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

0	0	0
0	1	1
0	1	0

Convolved Image

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

Convolved Image

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

0	0	0
0	0	0
0	0	0

Convolved Image

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

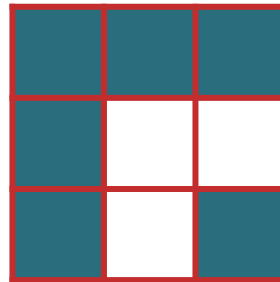
Background: Image Processing

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

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

Convolution Examples

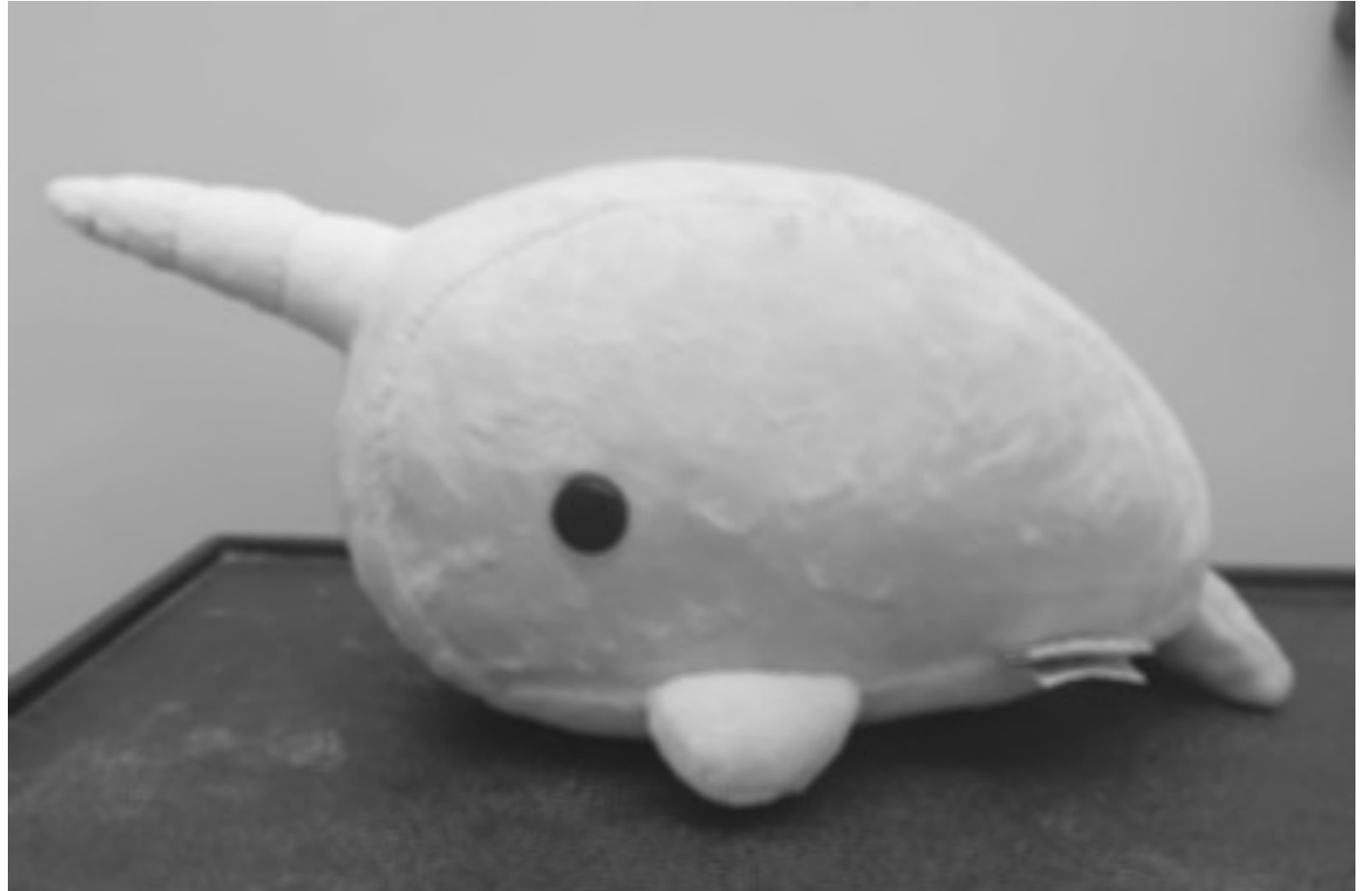
Original
Image



Convolution Examples

Smoothing
Convolution

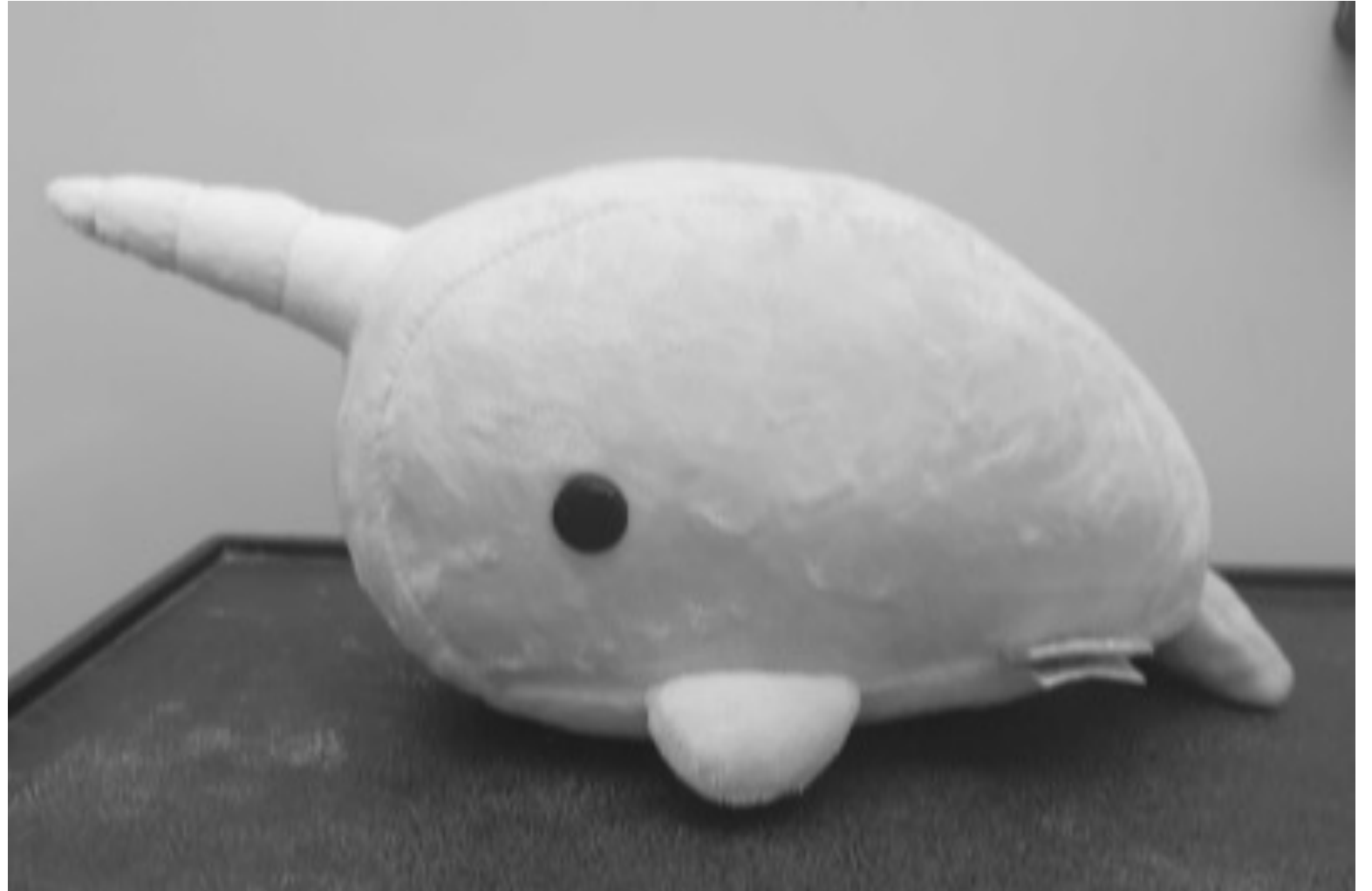
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$



Convolution Examples

Gaussian
Blur

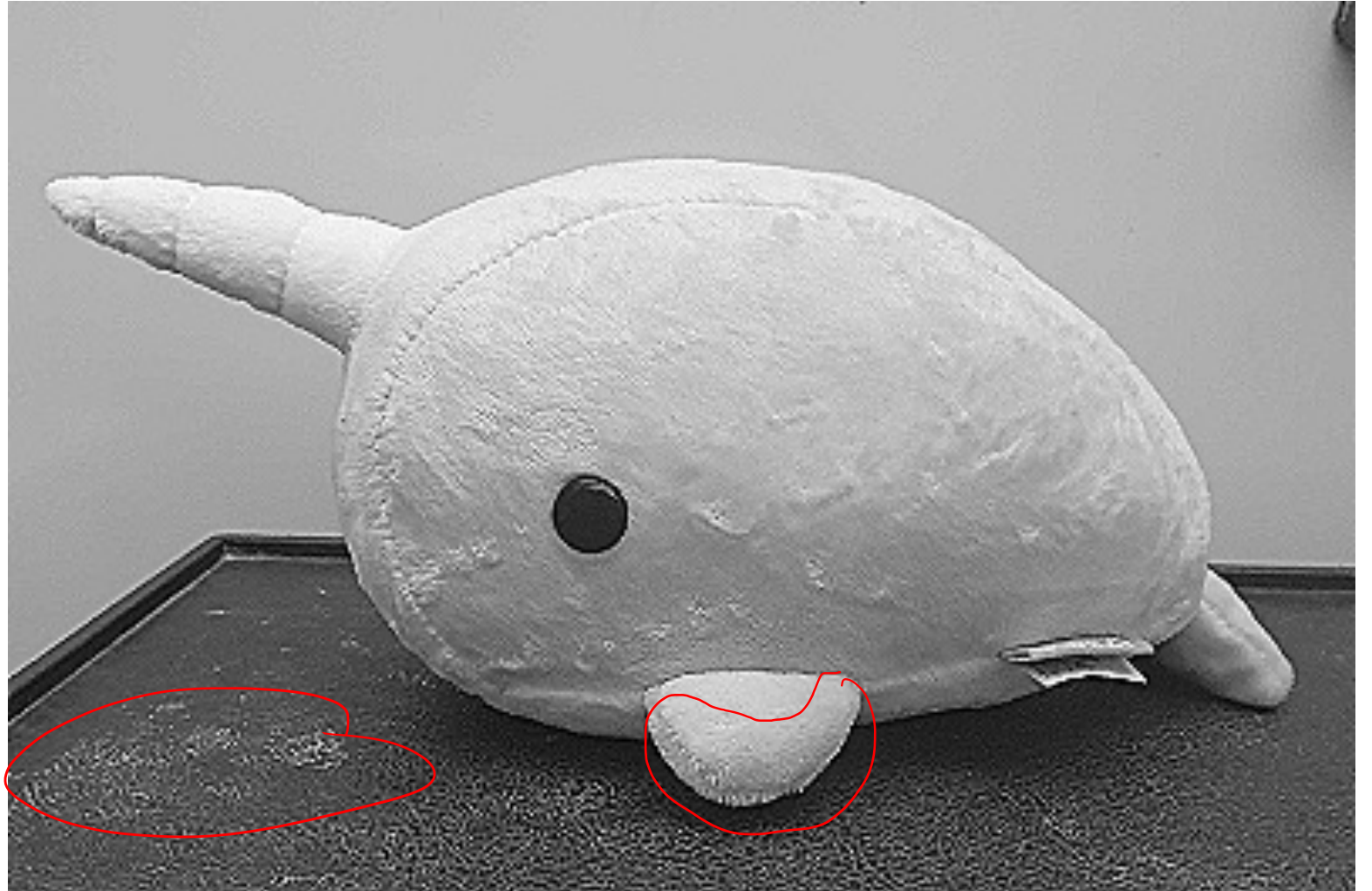
.01	.04	.06	.04	.01
.04	.19	.25	.19	.04
.06	.25	.37	.25	.06
.04	.19	.25	.19	.04
.01	.04	.06	.04	.01



Convolution Examples

Sharpening
Kernel

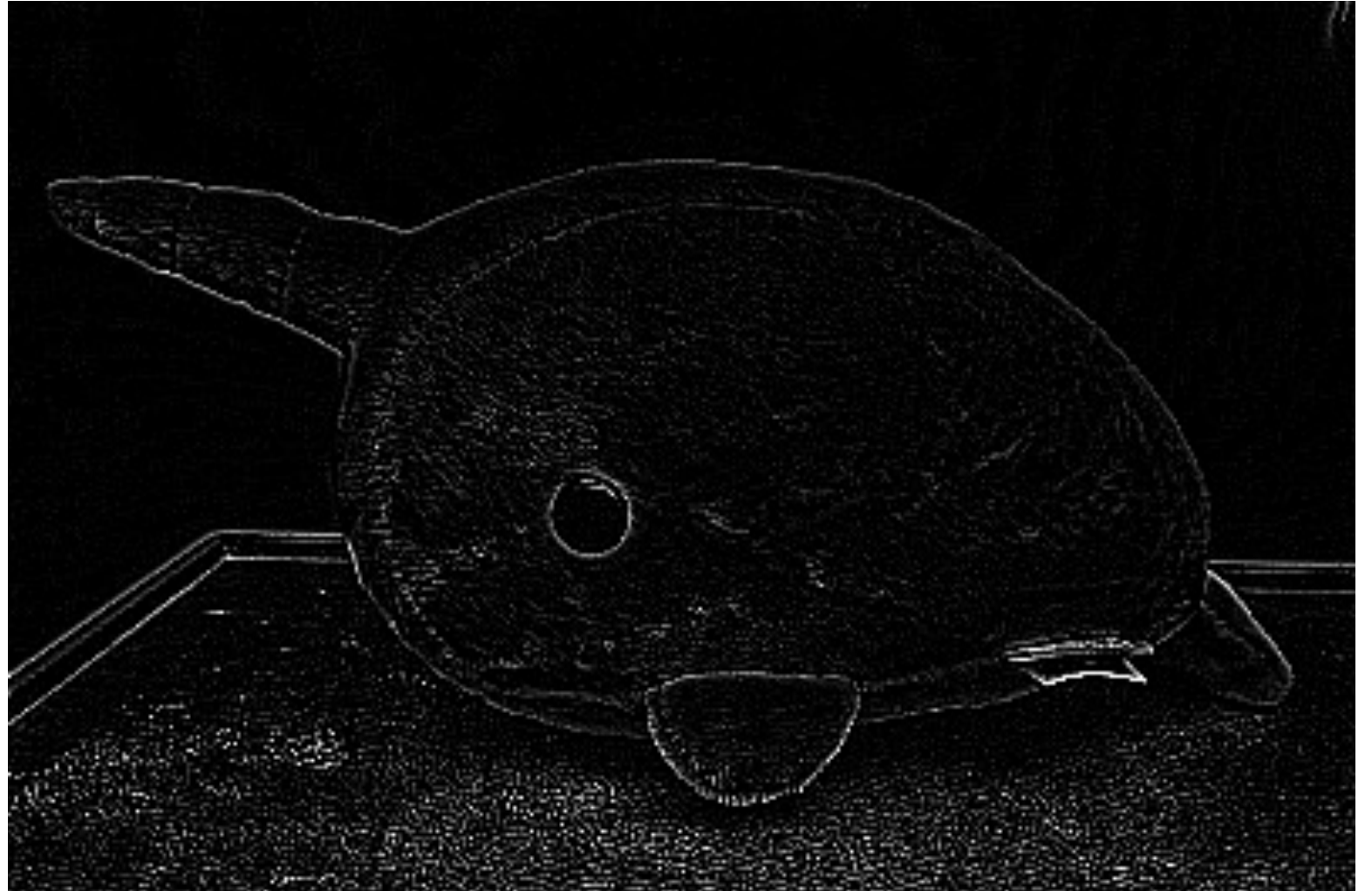
0	-1	0
-1	5	-1
0	-1	0



Convolution Examples

Edge
Detector

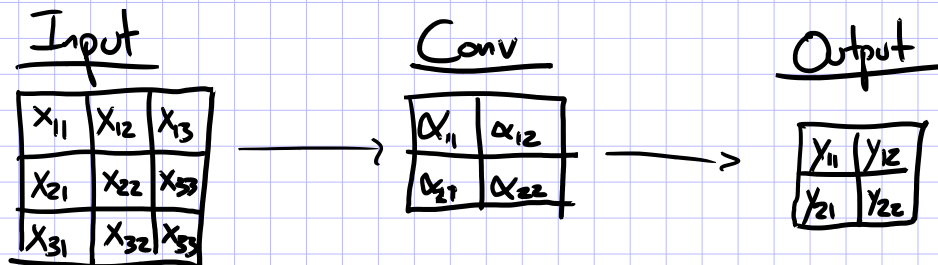
-1	-1	-1
-1	8	-1
-1	-1	-1



What's a convolution?

- Basic idea:
 - Pick a 3x3 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	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 with a stride > 1 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

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

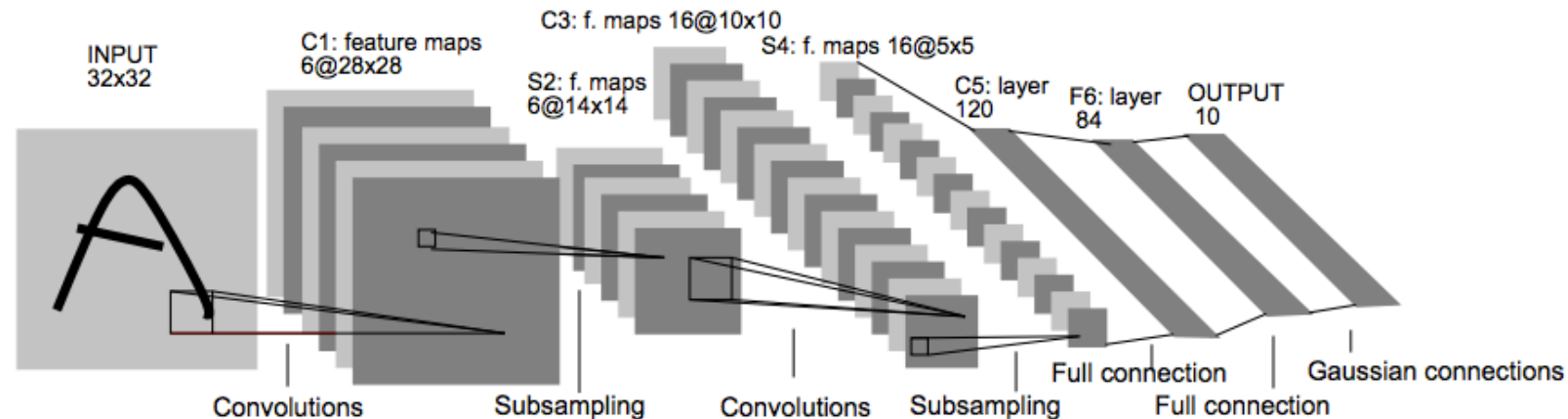


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

TRAINING CNNs

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. Given training data:

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

2. Choose each of the

– Decision function

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

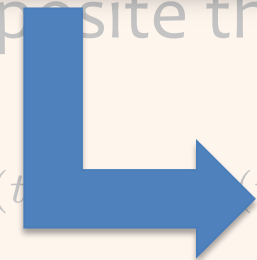
– Loss function

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

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)


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

SGD for CNNs

SGD for CNNs

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

$\vec{z}^{(5)} = \text{softmax}(\vec{z}^{(4)})$
 $\vec{z}^{(4)} = \text{linear}(\vec{z}^{(3)}, W)$
 $\vec{z}^{(3)} = \text{relu}(\vec{z}^{(2)})$
 $\vec{z}^{(2)} = \text{conv}(\vec{z}^{(1)}, \beta)$
 $\vec{z}^{(1)} = \text{conv}(\vec{x}, \alpha)$

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

SGD:

- ① Init $\vec{\theta}$
- ② While not converged:

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

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

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

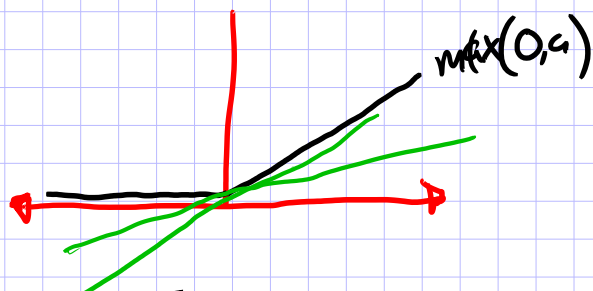
ReLU Layer

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

Forward:

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

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



Backward:

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

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

$$\frac{dy_i}{dx_j} = 0 \quad \forall i \neq j$$

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:

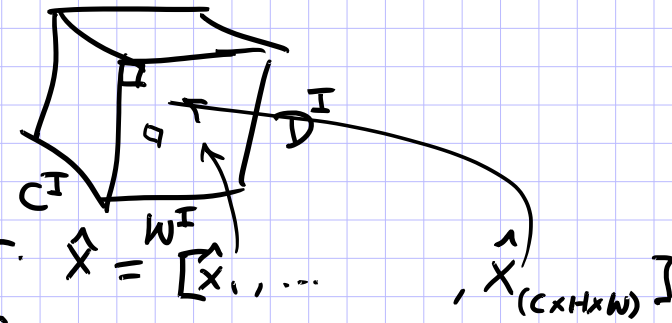
$$\frac{dJ}{dx_j} = \sum_{i=1}^k \frac{dJ}{dy_i} \frac{dy_i}{dx_j}$$

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. $\hat{x} = [x_1, \dots, x_n]$

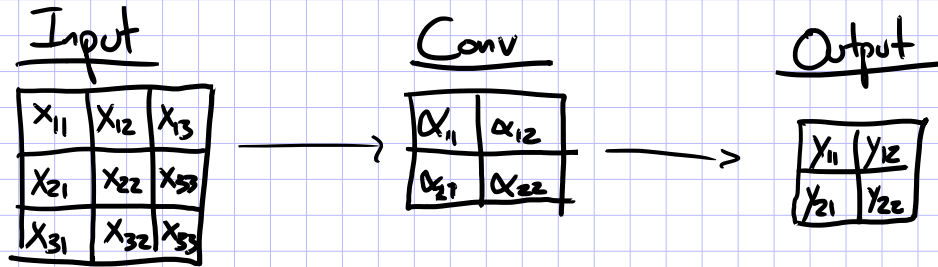
- then standard linear layer:

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

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

Convolutional Layer

Ex: 1 input channel, 1 output channel



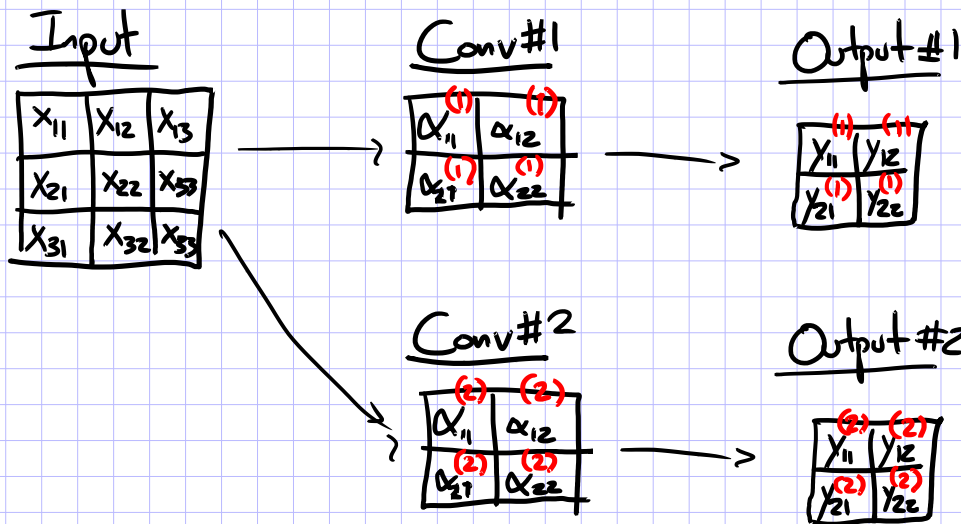
$$Y_{11} = \alpha_{11} X_{11} + \alpha_{12} X_{12} + \alpha_{21} X_{21} + \alpha_{22} X_{22} + \alpha_0$$

$$Y_{12} = \alpha_{11} X_{12} + \alpha_{12} X_{13} + \alpha_{21} X_{22} + \alpha_{22} X_{23} + \alpha_0$$

$$Y_{21} = \alpha_{11} X_{21} + \alpha_{12} X_{22} + \alpha_{21} X_{31} + \alpha_{22} X_{32} + \alpha_0$$

$$Y_{22} = \alpha_{11} X_{22} + \alpha_{12} X_{23} + \alpha_{21} X_{32} + \alpha_{22} X_{33} + \alpha_0$$

Ex: 1 input channel, 2 output channels



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

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

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

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

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

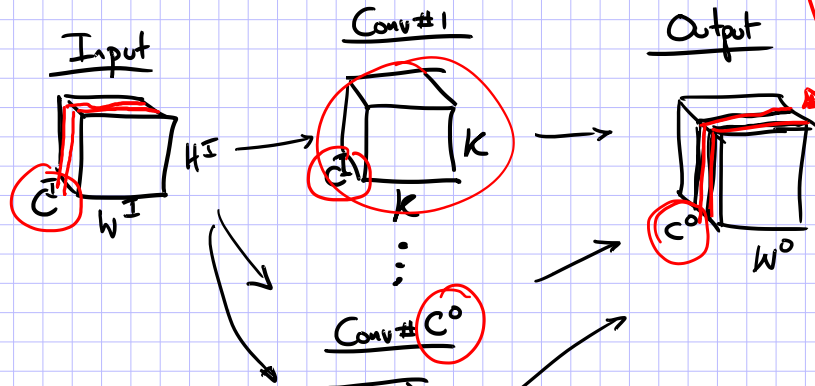
$$Y_{12}^{(2)} = \dots$$

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

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

Convolutional Layer

Ex: $C^I=3$ input channels, C^O output channels

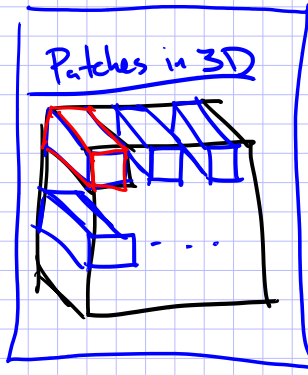


i -th slice is
 j -th output from
 j -th convolution matrix

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

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

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



Forward:

$$y_{ij}^{(k)} = \alpha_0^{(k)} + \sum_{c=1}^{C^I} \sum_{q=1}^K \sum_{r=1}^K \alpha_{qr}^{(k)} x_{mn}^{(c)}$$

where $m = s(i-1) + q$
 $n = s(j-1) + r$

Backward:

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

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

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

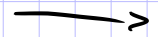
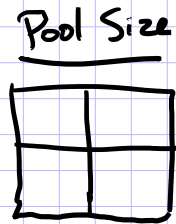
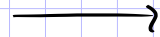
just some
calculus

Max-Pooling Layer

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

Input

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

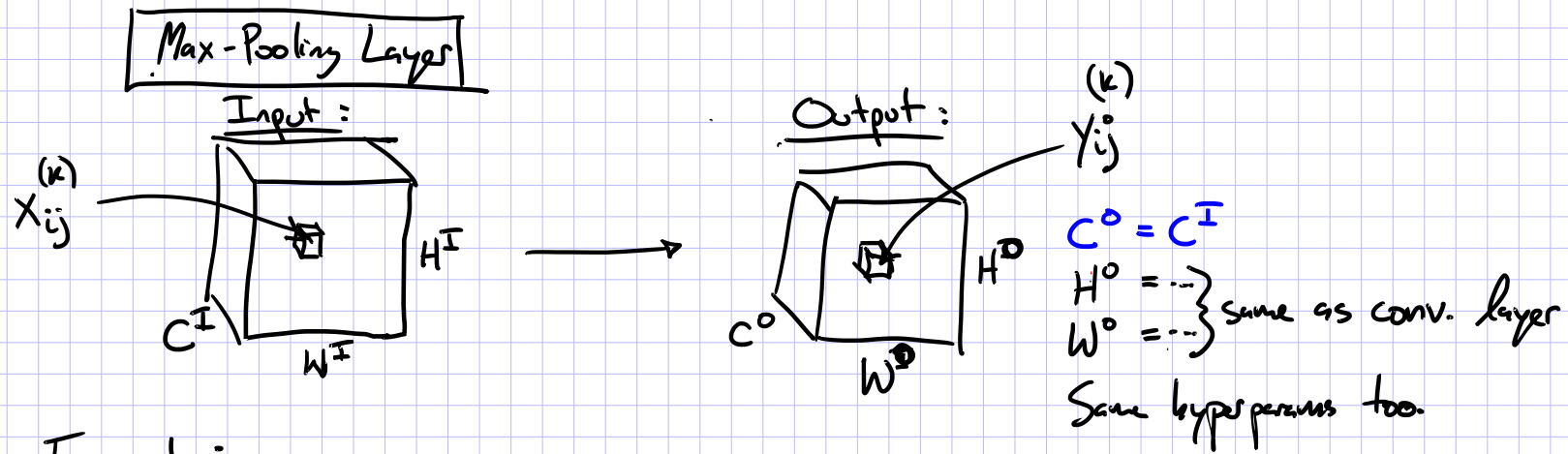


Output

y_{11}	y_{12}
y_{21}	y_{22}

$$y_{11} = \max(x_{11}, x_{12}, x_{21}, x_{22})$$
$$y_{12} = \max(x_{12}, x_{13}, x_{22}, x_{23})$$
$$y_{21} = \max(x_{21}, x_{22}, x_{31}, x_{32})$$
$$y_{22} = \max(x_{22}, x_{23}, x_{32}, x_{33})$$

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

- + $\max()$ 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

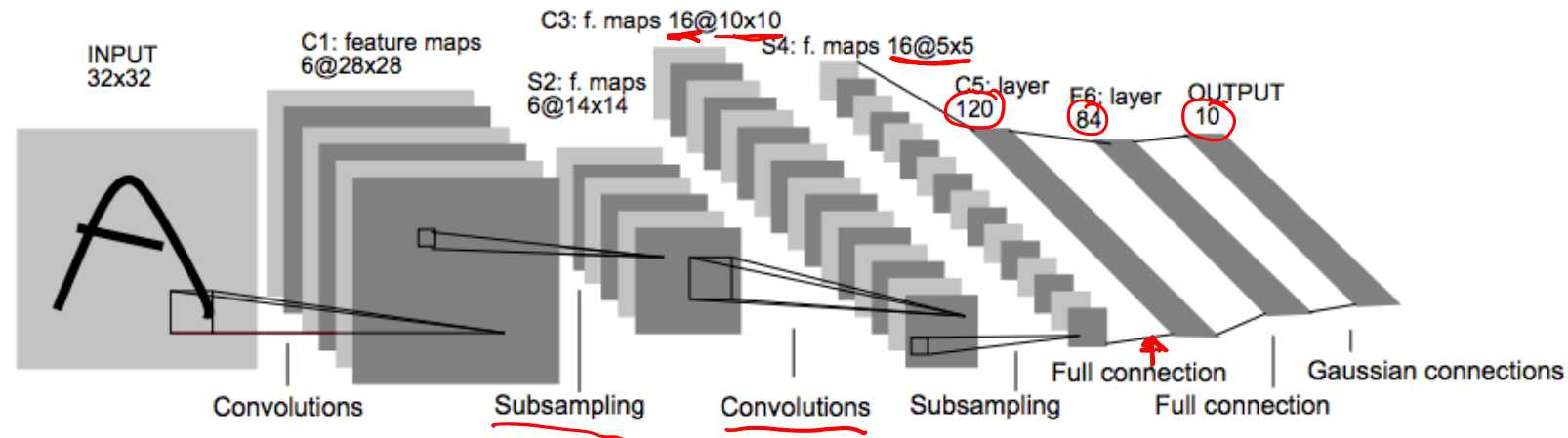


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Architecture #2: AlexNet

CNN for Image Classification

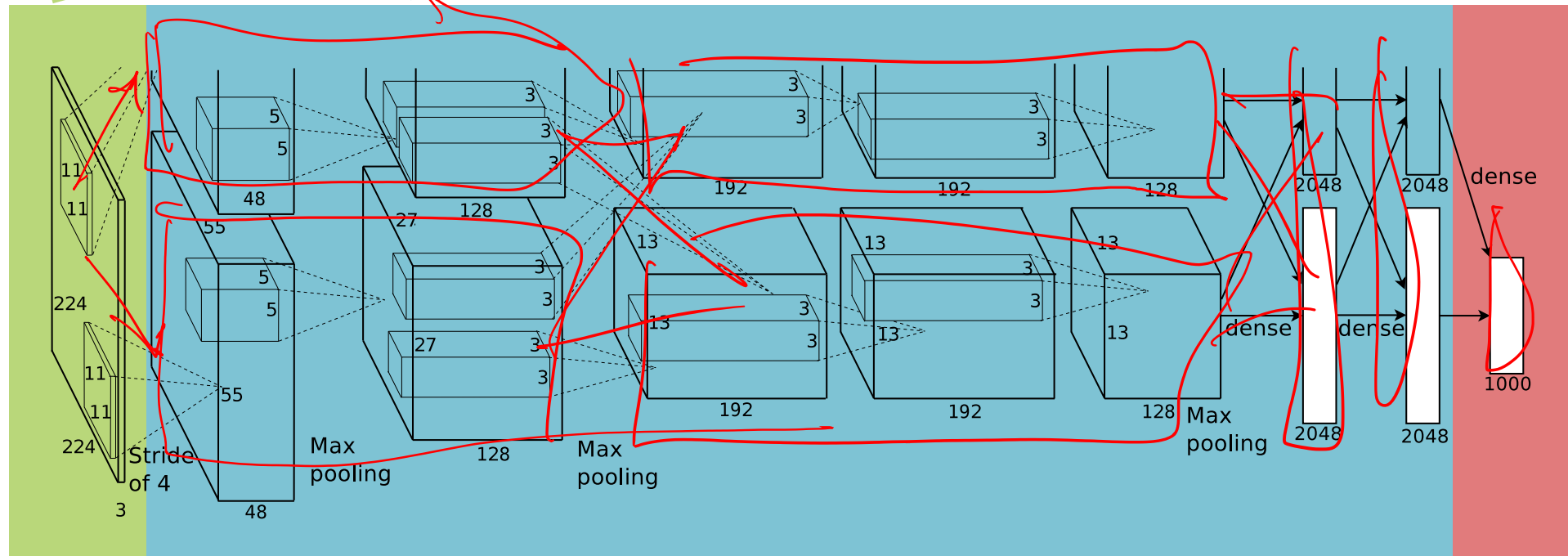
(Krizhevsky, Sutskever & Hinton, 2012)

15.3% error on ImageNet LSVRC-2012 contest

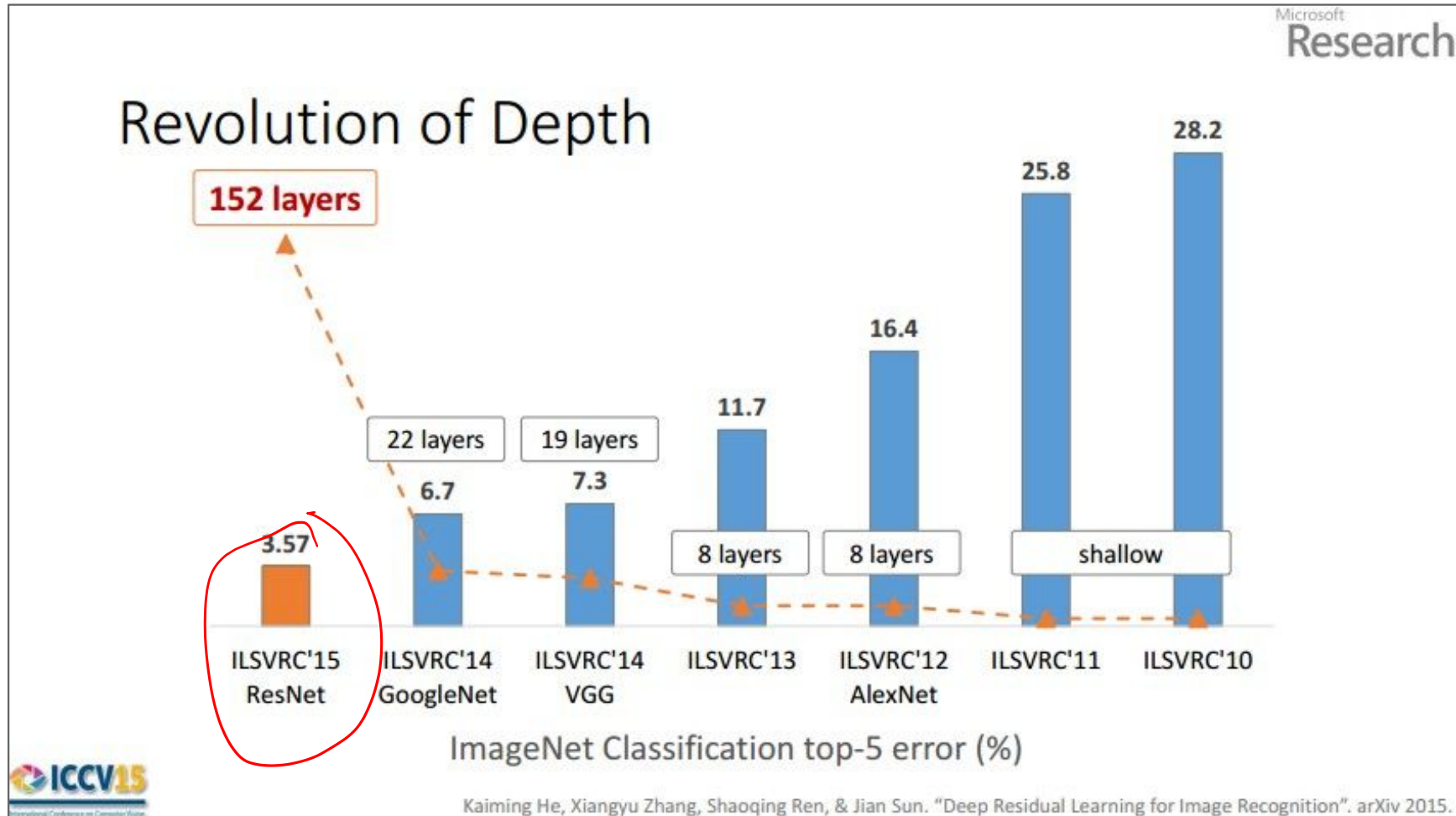
Input image (pixels)

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

1000-way softmax

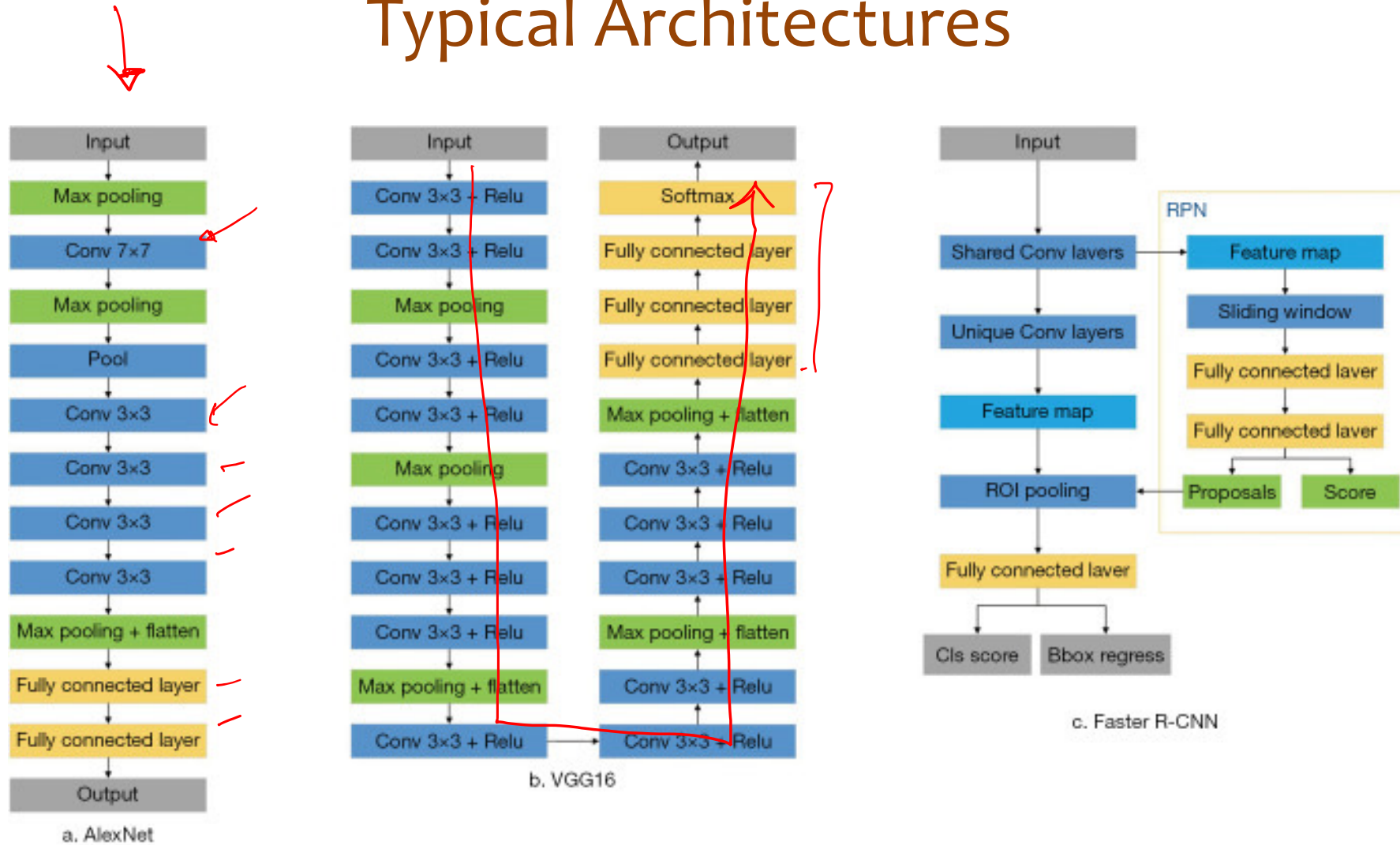


CNNs for Image Recognition



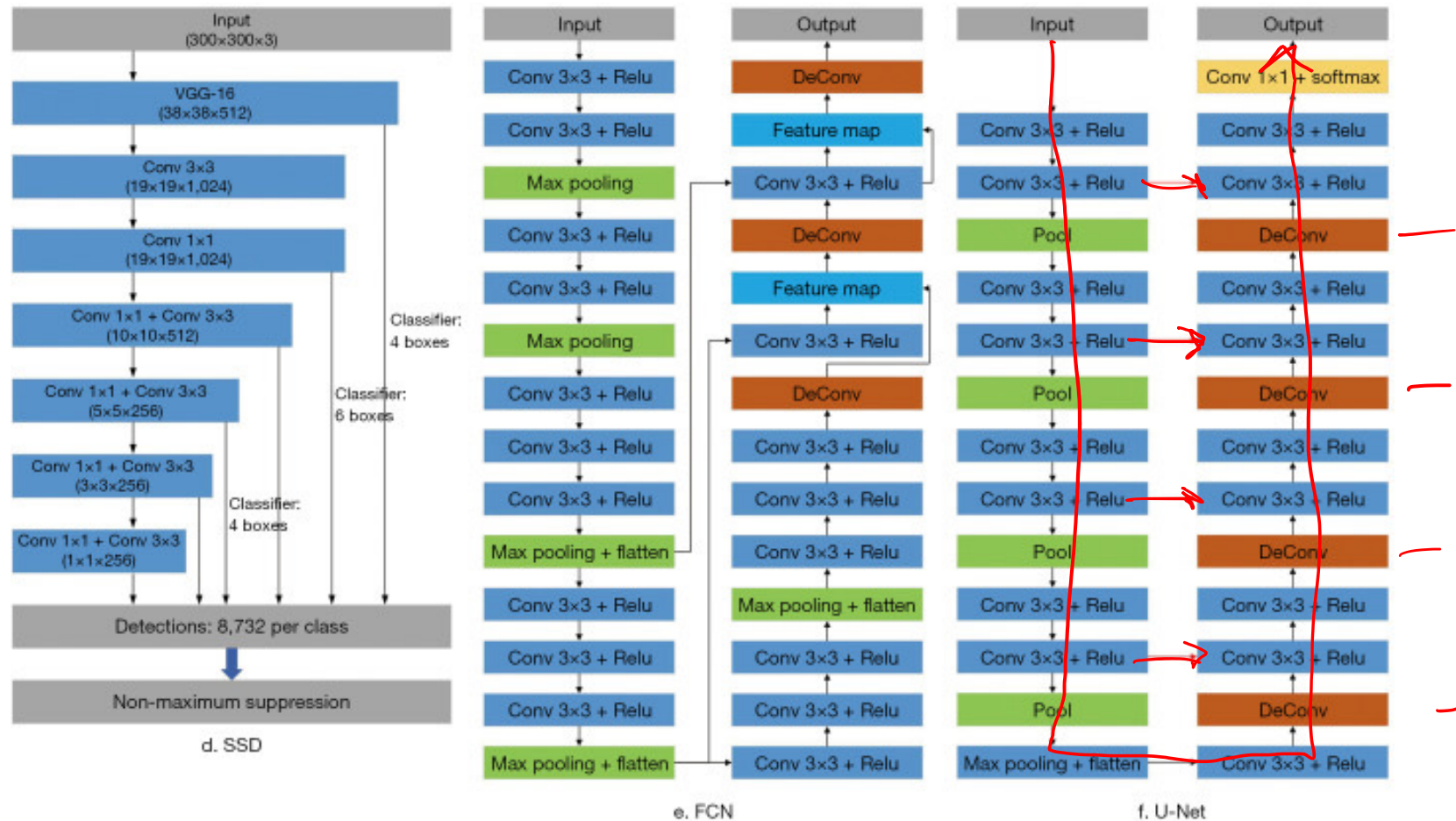
Convolutional Neural Network (CNN)

Typical Architectures



Convolutional Neural Network (CNN)

Typical Architectures



Convolutional Neural Network (CNN)

Typical Architectures

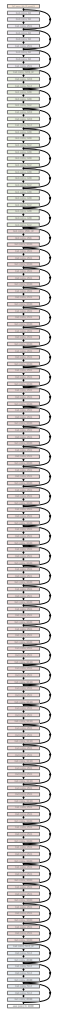
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



ResNet, 152 layers
(ILSVRC 2015)



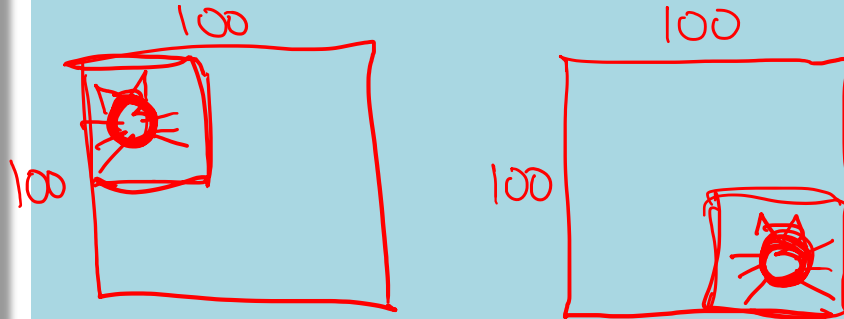
Microsoft
Research

In-Class Poll

Question: Q3: skip

Why do many layers used in computer vision *not have* location specific parameters?

Answer:



Convolutional Layer

For a convolutional layer, how do we pick the kernel size (aka. the size of the convolution)?

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

2x2
Convolution

θ_{11}	θ_{12}
θ_{21}	θ_{22}

3x3
Convolution

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

4x4
Convolution

θ_{11}	θ_{12}	θ_{13}	θ_{14}
θ_{21}	θ_{22}	θ_{23}	θ_{24}
θ_{31}	θ_{32}	θ_{33}	θ_{34}
θ_{41}	θ_{42}	θ_{43}	θ_{44}

- A small kernel can only see a very small part of the image, but is fast to compute
- A large kernel can see more of the image, but at the expense of speed

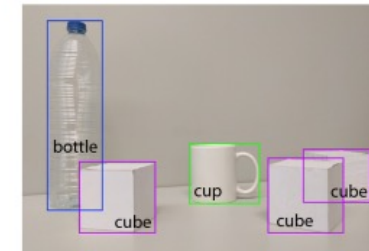
COMPUTER VISION

Common Tasks in Computer Vision

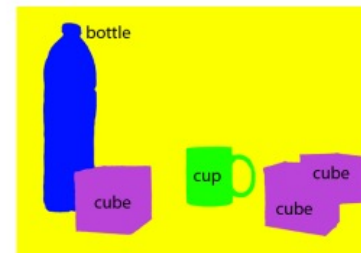
1. Image Classification
2. Image Classification + Localization
3. Human Pose Estimation
4. Semantic Segmentation
5. Object Detection
6. Instance Segmentation
7. Image Captioning



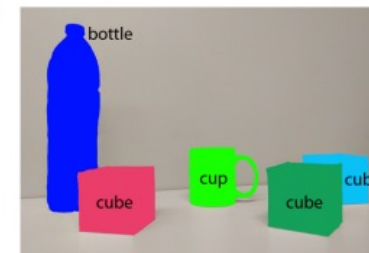
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) Instance segmentation

Image Classification

- Given an image, predict a single label
- A multi-class classification problem

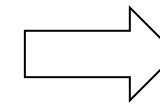
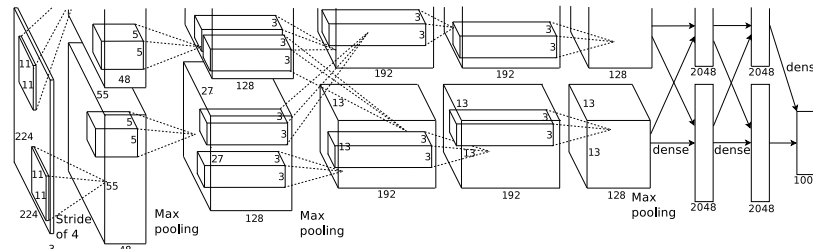
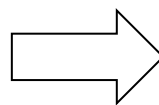
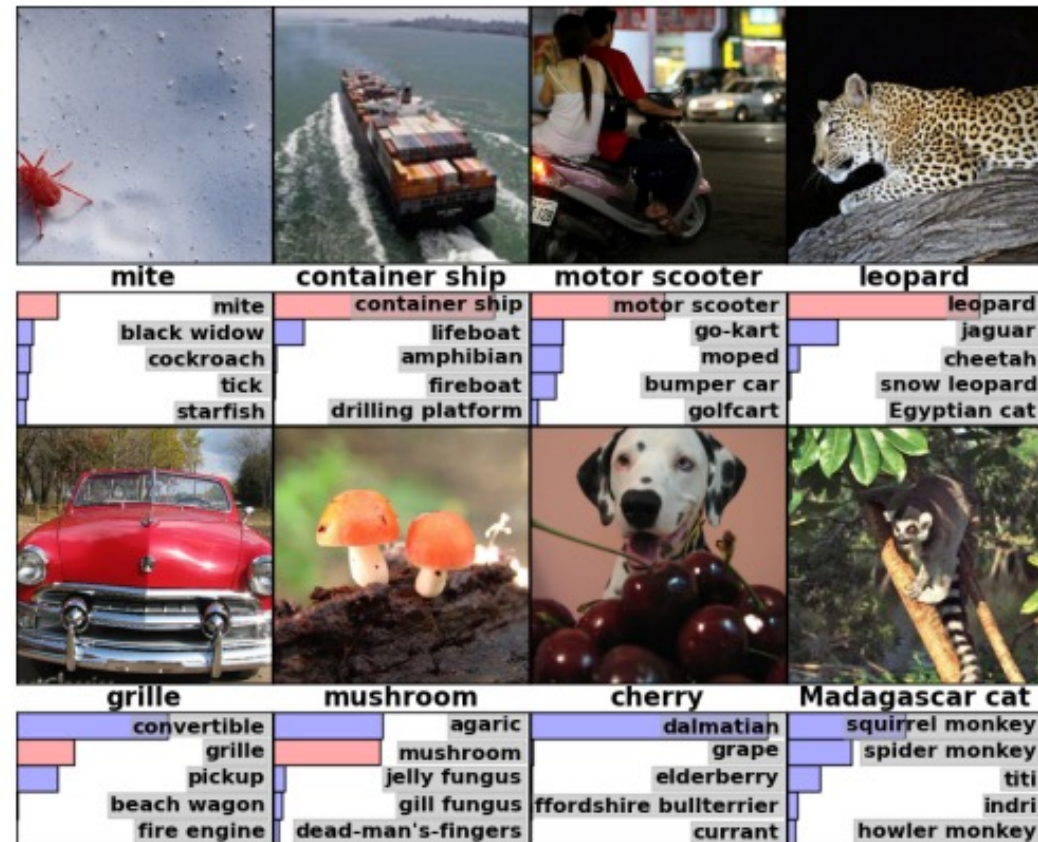
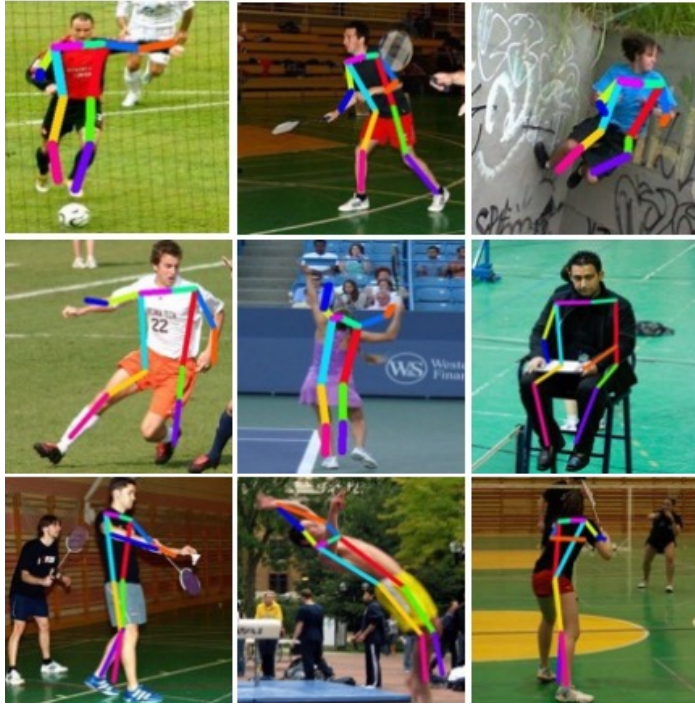


Image Classification + Localization

- Given an image, predict a single label and a bounding box for the object
- Bounding box is represented as (x, y, h, w) , position (x,y) and height/width (h,w)



Human Pose Estimation



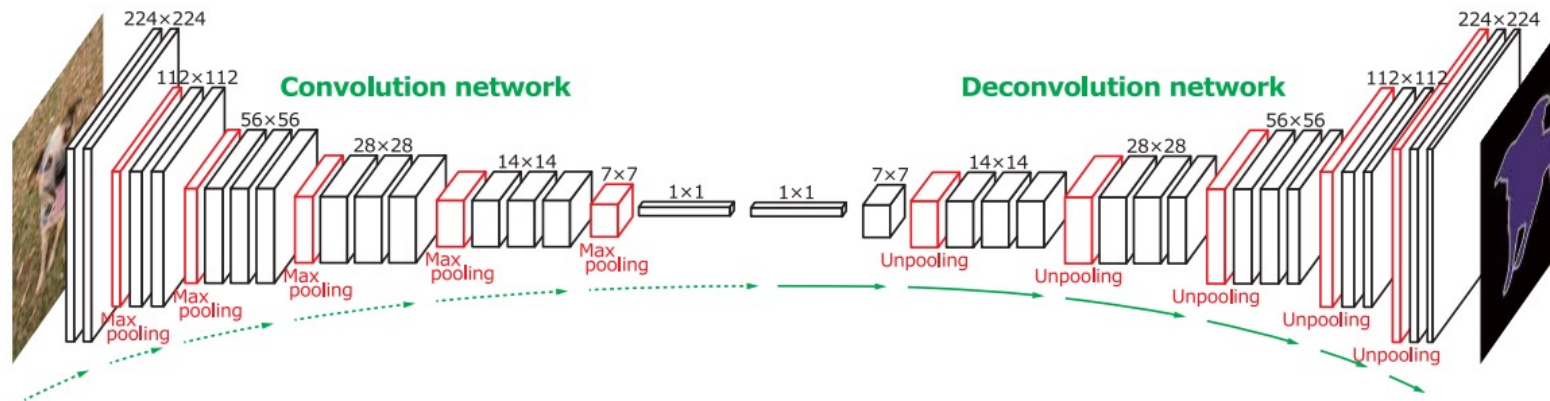
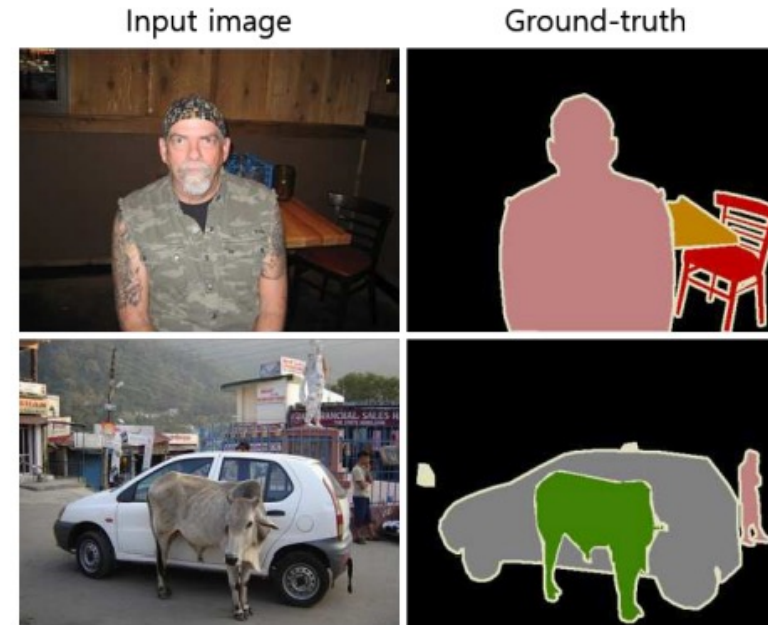
- Given an image of a human, predict the position of several keypoints (left hand, right hand, left elbow, ..., right foot)
- This is a multiple regression problem, where each keypoint has a corresponding position (x_i, y_i)



Figure from https://openaccess.thecvf.com/content_cvpr_2014/papers/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.pdf

Semantic Segmentation

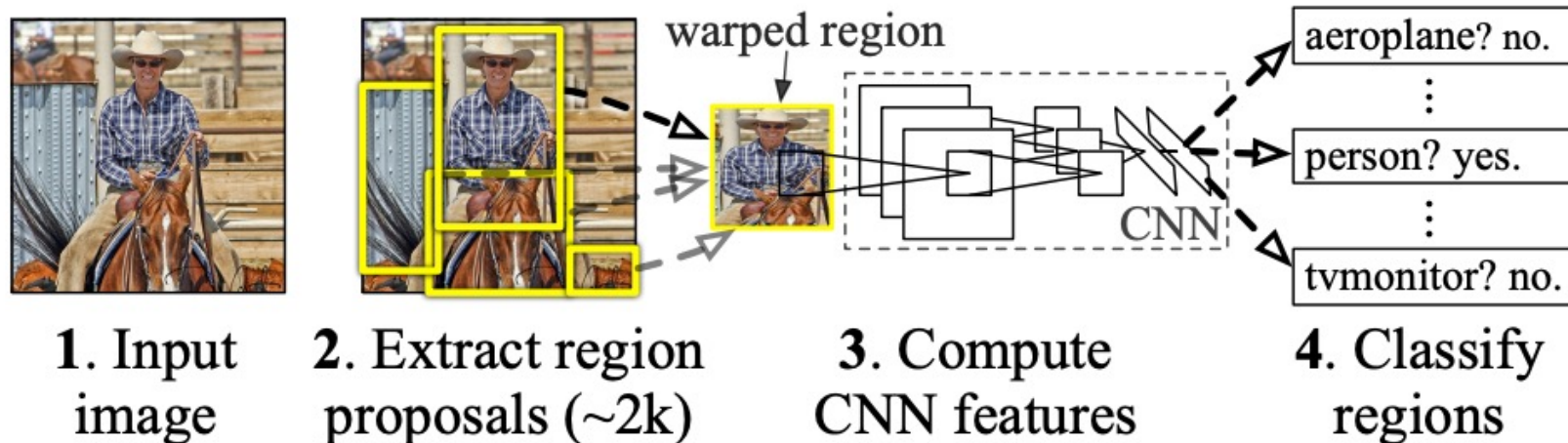
- Given an image, predict a label for every pixel in the image
- Not merely a classification problem, because there are strong correlations between pixel-specific labels



Object Detection

- Given an image, for each object predict a bounding box and a label (x,y,w,h,l)
- Example: R-CNN
 - $(x=110, y=13, w=50, h=72, l=person)$
 - $(x=90, y=55, w=81, h=87, l=horse)$
 - $(x=421, y=533, w=24, h=30, l=chair)$
 - $(x=2, y=25, w=51, h=121, l=gate)$

R-CNN: *Regions with CNN features*



Instance Segmentation

- Predict per-pixel labels as in semantic segmentation, but differentiate between different instances of the same label
- *Example:* if there are two people in the image, one person should be labeled **person-1** and one should be labeled **person-2**

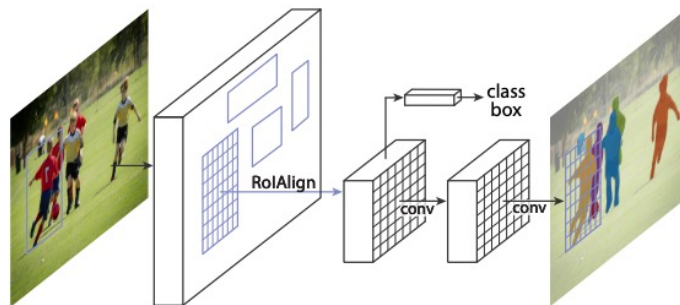
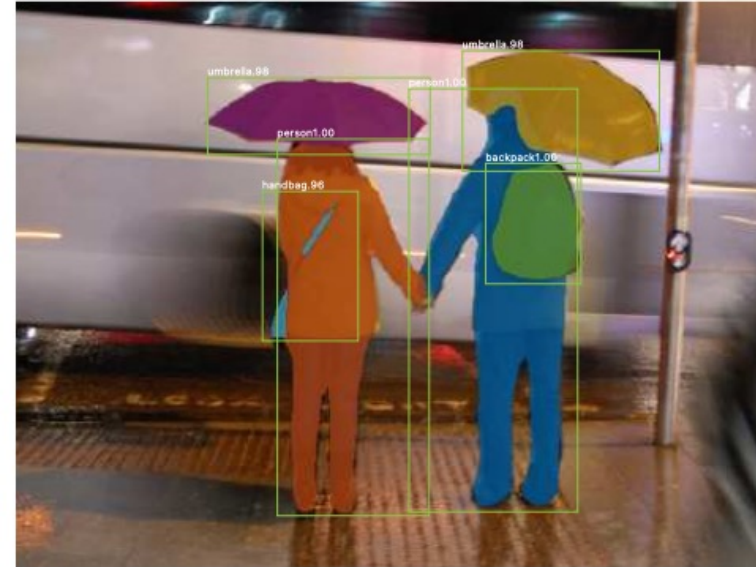
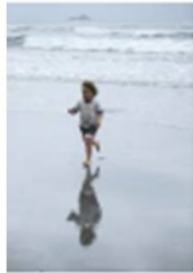


Figure 1. The **Mask R-CNN** framework for instance segmentation.

Image Captioning



Ground Truth Caption: A little boy runs away from the approaching waves of the ocean.

Generated Caption: A young boy is running on the beach.



Ground Truth Caption: A brunette girl wearing sunglasses and a yellow shirt.

Generated Caption: A woman in a black shirt and sunglasses smiles.

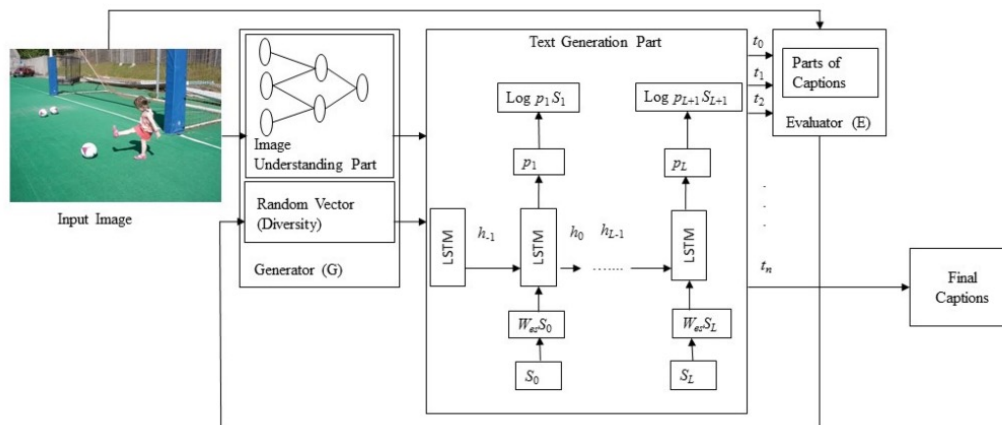


Fig. 3. A block diagram of other deep-learning-based captioning.

- Take an image as input, and generate a sentence describing it as output (i.e. the caption)
- Typical methods include a deep CNN/transformer and a RNN-like language model
- (The task of *Dense Captioning* is to generate one caption per bounding box)

Image Captioning

Table 1. An Overview of the Deep-Learning-Based Approaches for Image Captioning

Reference	Image Encoder	Language Model	Category
Kiros et al. 2014 [69]	AlexNet	LBL	MS, SL, WS, EDA
Kiros et al. 2014 [70]	AlexNet, VGGNet	1. LSTM 2. SC-NLM	MS, SL, WS, EDA
Mao et al. 2014 [95]	AlexNet	RNN	MS, SL, WS
Karpathy et al. 2014 [66]	AlexNet	DTR	MS, SL, WS, EDA
Mao et al. 2015 [94]	AlexNet, VGGNet	RNN	MS, SL, WS
Chen et al. 2015 [23]	VGGNet	RNN	VS, SL, WS, EDA
Fang et al. 2015 [33]	AlexNet, VGGNet	MELM	VS, SL, WS, CA
Jia et al. 2015 [59]	VGGNet	LSTM	VS, SL, WS, EDA
Karpathy et al. 2015 [65]	VGGNet	RNN	MS, SL, WS, EDA
Vinyals et al. 2015 [142]	GoogLeNet	LSTM	VS, SL, WS, EDA
Xu et al. 2015 [152]	AlexNet	LSTM	VS, SL, WS, EDA, AB
Jin et al. 2015 [61]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Wu et al. 2016 [151]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Sugano et al. 2016 [129]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Mathews et al. 2016 [97]	GoogLeNet	LSTM	VS, SL, WS, EDA, SC
Wang et al. 2016 [144]	AlexNet, VGGNet	LSTM	VS, SL, WS, EDA
Johnson et al. 2016 [62]	VGGNet	LSTM	VS, SL, DC, EDA
Mao et al. 2016 [92]	VGGNet	LSTM	VS, SL, WS, EDA
Wang et al. 2016 [146]	VGGNet	LSTM	VS, SL, WS, CA
Tran et al. 2016 [135]	ResNet	MELM	VS, SL, WS, CA
Ma et al. 2016 [90]	AlexNet	LSTM	VS, SL, WS, CA
You et al. 2016 [156]	GoogLeNet	RNN	VS, SL, WS, EDA, SCB
Yang et al. 2016 [153]	VGGNet	LSTM	VS, SL, DC, EDA
Anne et al. 2016 [6]	VGGNet	LSTM	VS, SL, WS, CA, NOB
Yao et al. 2017 [155]	GoogLeNet	LSTM	VS, SL, WS, EDA, SCB
Lu et al. 2017 [88]	ResNet	LSTM	VS, SL, WS, EDA, AB
Chen et al. 2017 [21]	VGGNet, ResNet	LSTM	VS, SL, WS, EDA, AB
Gan et al. 2017 [41]	ResNet	LSTM	VS, SL, WS, CA, SCB
Pedersoli et al. 2017 [112]	VGGNet	RNN	VS, SL, WS, EDA, AB
Ren et al. 2017 [119]	VGGNet	LSTM	VS, ODL, WS, EDA
Park et al. 2017 [111]	ResNet	LSTM	VS, SL, WS, EDA, AB
Wang et al. 2017 [148]	ResNet	LSTM	VS, SL, WS, EDA
Tavakoli et al. 2017 [134]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Liu et al. 2017 [84]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Gan et al. 2017 [39]	ResNet	LSTM	VS, SL, WS, EDA, SC
Dai et al. 2017 [26]	VGGNet	LSTM	VS, ODL, WS, EDA
Shetty et al. 2017 [126]	GoogLeNet	LSTM	VS, ODL, WS, EDA
Liu et al. 2017 [85]	Inception-V3	LSTM	VS, ODL, WS, EDA
Gu et al. 2017 [51]	VGGNet	1. Language CNN 2. LSTM	VS, SL, WS, EDA
Yao et al. 2017 [154]	VGGNet	LSTM	VS, SL, WS, CA, NOB

(Continued)

- Take an image as input, and generate a sentence describing it as output (i.e. the caption)
- Typical methods include a deep CNN/transformer and a RNN-like language model
- (The task of *Dense Captioning* is to generate one caption per bounding box)

Medical Image Analysis

Notice that **most** of these tasks are structured prediction problems, not merely classification

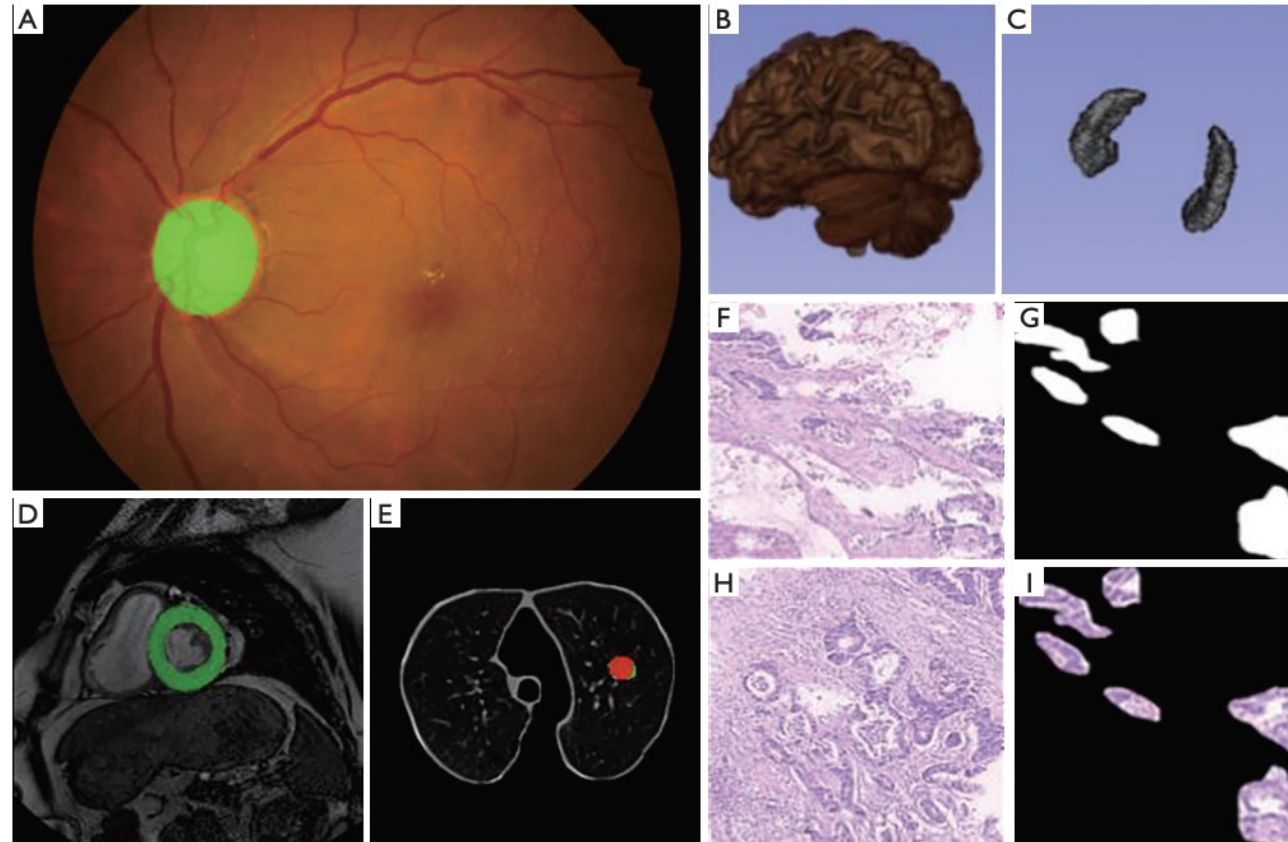


Figure 2 Deep learning application in medical image analysis. (A) Fundus detection; (B,C) hippocampus segmentation; (D) left ventricular segmentation; (E) pulmonary nodule classification; (F,G,H,I) gastric cancer pathology segmentation. The staining method is H&E, and the magnification is $\times 40$.

CNN VISUALIZATIONS

Visualization of CNN

https://adamharley.com/nn_vis/cnn/2d.html

The screenshot displays a CNN visualization interface. At the top left, a drawing area shows a handwritten digit '4' on a dark background. Below it are drawing tools (erase, pencil, lasso) and a small grid of the digit. The main visualization area shows the digit being processed through several layers, with the output of each layer shown as a grid of small images. The layers are: Input layer, Convolution layer 1, Downsampling layer 1, Convolution layer 2, Downsampling layer 2, Fully-connected layer 1, Fully-connected layer 2, and Output layer. A 'Layer visibility' panel on the right allows toggling each layer's visibility. The output layer shows the final classification result, which is '4'. The interface also displays '60 FPS (1-61)', a score '0123456789', and a 'Downsampled drawing' section with 'First guess: 7' and 'Second guess: 8'.

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

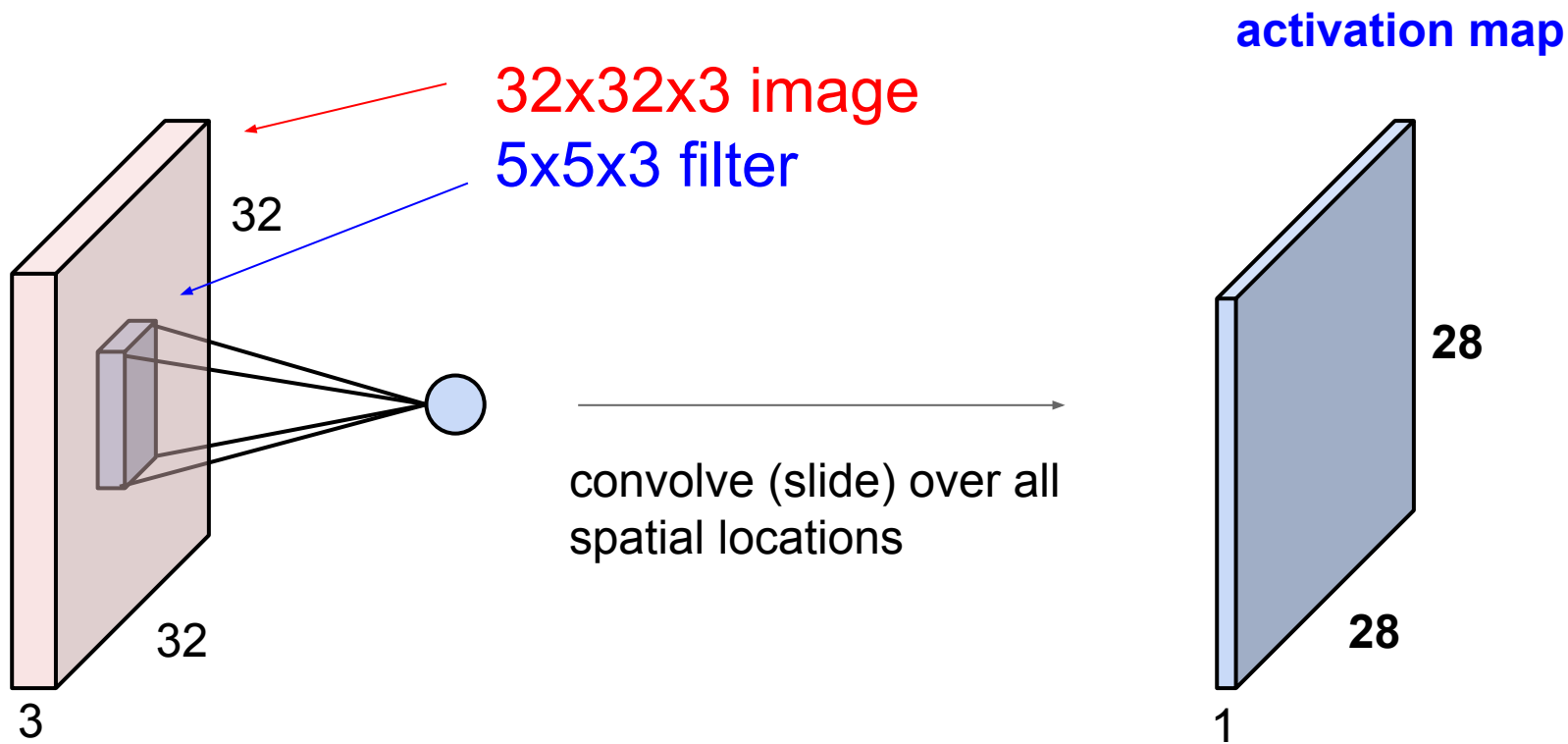


Figure from Fei-Fei Li & Andrej Karpathy & Justin Johnson (CS231N)

Animation of 3D Convolution

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

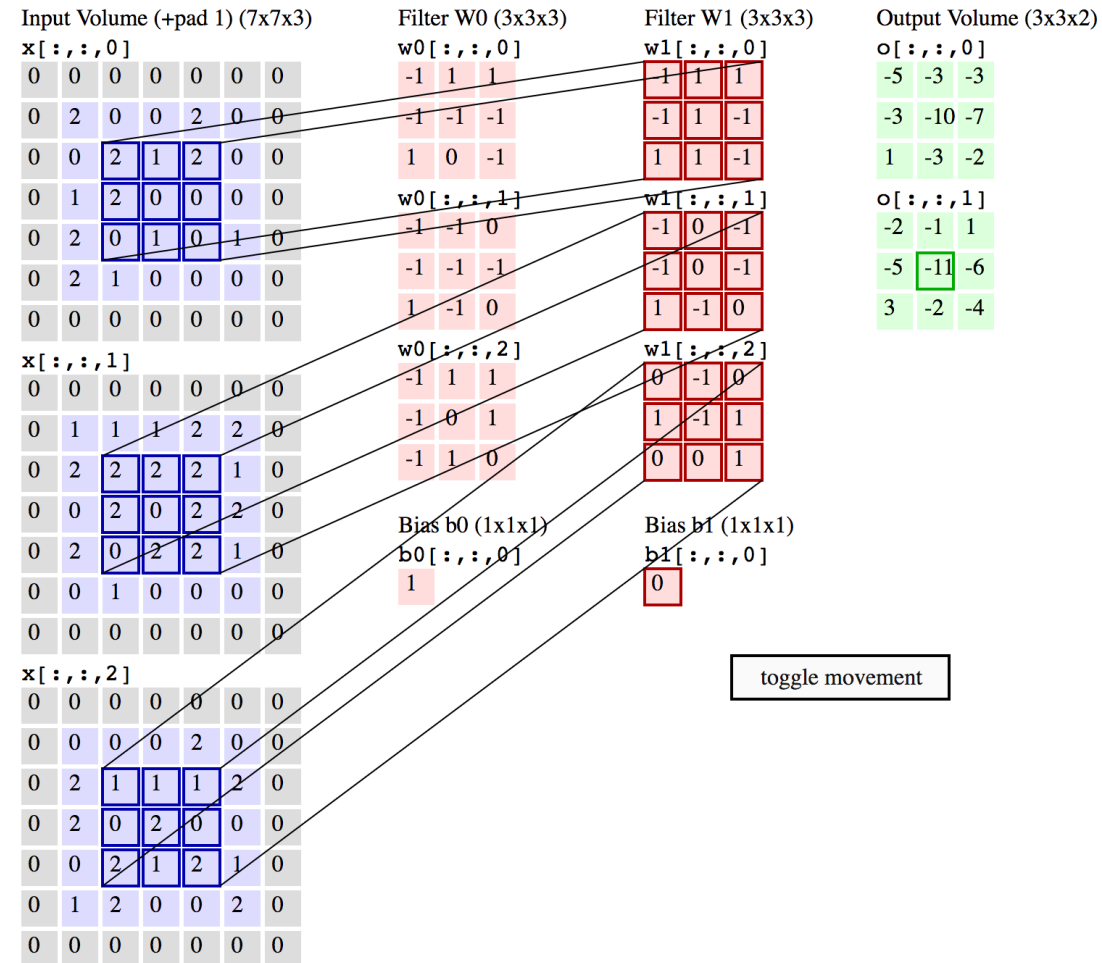


Figure from Fei-Fei Li & Andrej Karpathy & Justin Johnson (CS231N)

MNIST Digit Recognition with CNNs (in your browser)

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

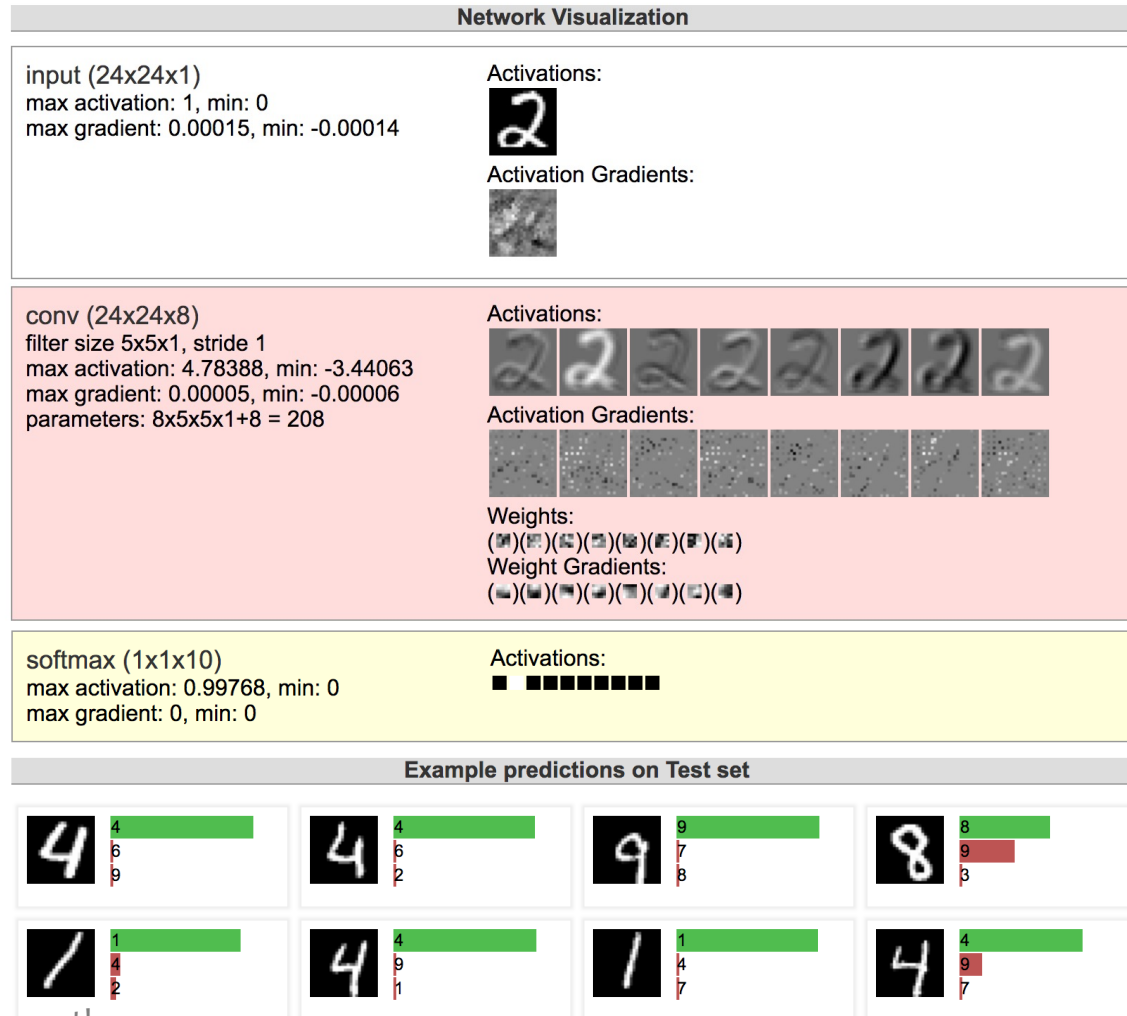


Figure from Andrej Karpathy

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

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