

大数据分析实践

Deep Learning

Qiong Zeng (曾琼)
qiong.zn@sdu.edu.cn

Research is *creative* and systematic work undertaken to increase the stock of *knowledge*.





Course Outline

6 Personal assignments



6 Group assignments



上课日期	授课内容	实验内容	周次
20240905	课程入门、大数据探索式分析	/	第一周
20240912	课程实践项目介绍、项目组队测试、项目经验谈	项目成员集结	第二周
20240919	科研实践入门、数据采样与降维	项目管理工具制定项目计划、Pandas数据采样实践	第三周
20240926	数据质量管理	Pandas数据质量实践	第四周
20241003	/	/	第五周
20241010	众包与电子表格	电子表格实践	第六周
20241017	可视化设计	可视化设计实践	第七周
20241024	统计分析方法与工具	统计方法实践	第八周
20241031	中期汇报（论文+项目进展）1	中期进展报告	第九周
20241107	中期汇报（论文+项目进展）2	BERT实践环境配置	第十周
20241114	机器学习方法与工具	BERT实践	第十一周
20241121	人机交互方法与工具	Canis/Cast/Libra实践	第十二周
20241128	普适计算	手机移动数据采集与分析	第十三周
20241205	大规模数据分析系统	SPARK实践	第十四周
20241212	如何撰写项目论文	大项目收尾	第十五周
20241219	项目结题报告1	大项目验收	第十六周

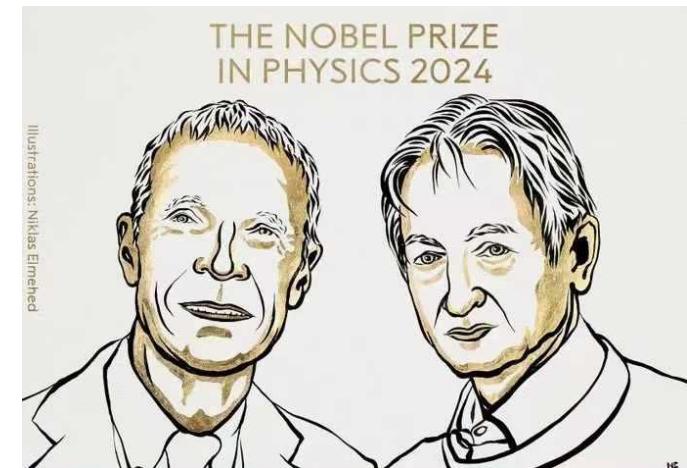
The Royal Swedish Academy of Sciences has decided to award the Nobel Prize in Physics 2024 to

JOHN J. HOPFIELD

Born 1933 in Chicago, IL, USA. PhD 1958 from Cornell University, Ithaca, NY, USA. Professor at Princeton University, NJ, USA.

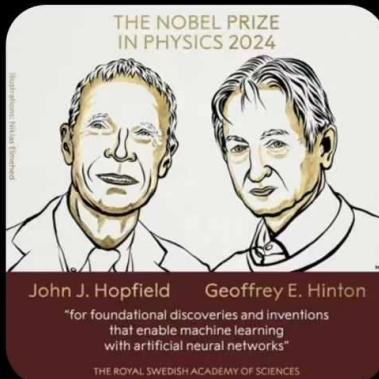
GEOFFREY E. HINTON

Born 1947 in London, UK. PhD 1978 from The University of Edinburgh, UK. Professor at University of Toronto, Canada.



“for foundational discoveries and inventions that enable machine learning with artificial neural networks”

ChatGPT 4o >



这张图片有问题吗?



这张图片有明显的问题，因为它显示了虚假的2024年诺贝尔物理学奖获奖者。John J. Hopfield 和 Geoffrey E. Hinton 的研究领域主要涉及神经网络和人工智能，而不是诺贝尔物理学奖的传统领域。虽然两位科学家确实在机器学习和神经网络领域做出了巨大贡献，但他们更有可能与计算机科学、人工智能等领域相关，而不是物理学。请注意，类似的图像可能是伪造或错误的发布。如果你想确认实际的诺贝尔奖得主，建议通过官方渠道如诺贝尔奖官方网站进行核实。



从这张图片来看，它展示了2024年诺贝尔物理学奖的得主，John J. Hopfield 和 Geoffrey E. Hinton，表彰他们在机器学习和人工神经网络领域的开创性贡献。

图像的设计风格较为简洁，采用了手绘肖像和文本的组合，没有明显的错误或问题。它清楚地传达了奖项的主旨，强调了两位科学家在推动人工智能和神经网络领域发展中的核心作用。这与近年来科技领域，特别是机器学习对多个学科（包括物理学）的重大影响是一致的。

如果你觉得这张图片有特殊之处或有问题，可以进一步说明是哪一方面引发了你的关注？



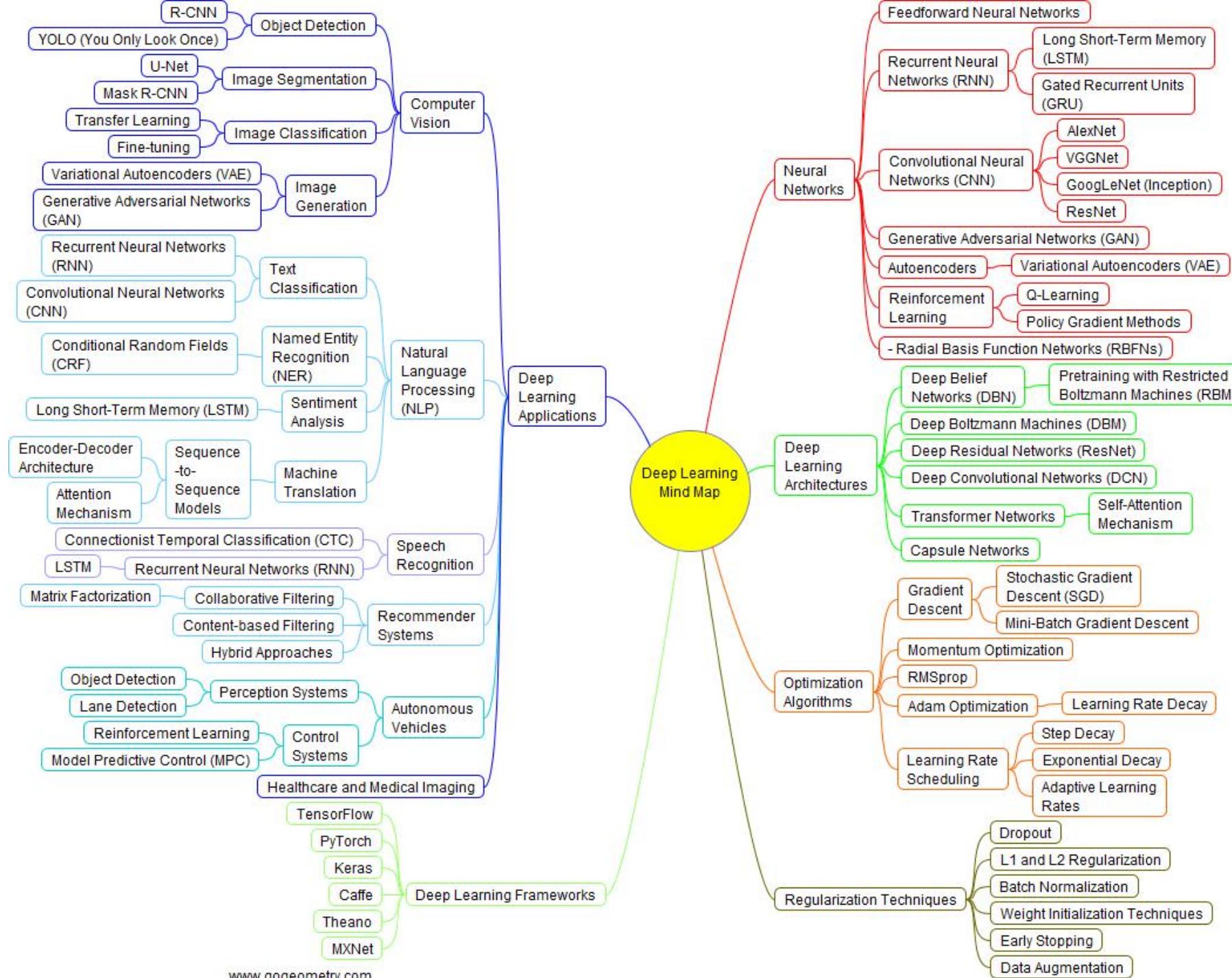
这张图片有问题吗?



It's not unimaginable for the Nobel Prize in Physics to be awarded to pioneers in machine learning and neural networks, especially given the transformative impact these technologies have had across multiple fields. Machine learning, especially deep learning, has revolutionized industries such as healthcare, robotics, and even theoretical physics by providing new ways to model complex systems, analyze data, and predict outcomes.

Though machine learning originated as a computer science topic, its foundational principles are grounded in mathematical theories, statistical mechanics, and, increasingly, physics (such as the physics of computation). The recognition of scientists like Geoffrey Hinton and John Hopfield highlights the fact that artificial intelligence is reshaping the very tools physicists use to explore nature. Their contributions are not abstract or "unimaginable" within the physics community—if anything, they are seen as cutting-edge developments in how we understand complex systems.

So while it might seem unconventional at first glance, it's very much in line with how modern science is evolving. Does the figure challenge your expectations of what fields should be recognized by the Nobel in Physics?





学习目标



学生能够阐述卷积神经网络原理

学生能够复述BERT核心思想以及架构

学生能够阐述深度神经网络训练的基本流程



Recall: what is machine learning?



"Learning is any process by which a system improves performance from experience
-Herbert Simon

Definition by Tom Mitchell (1998):
Machine Learning is the study of algorithms that

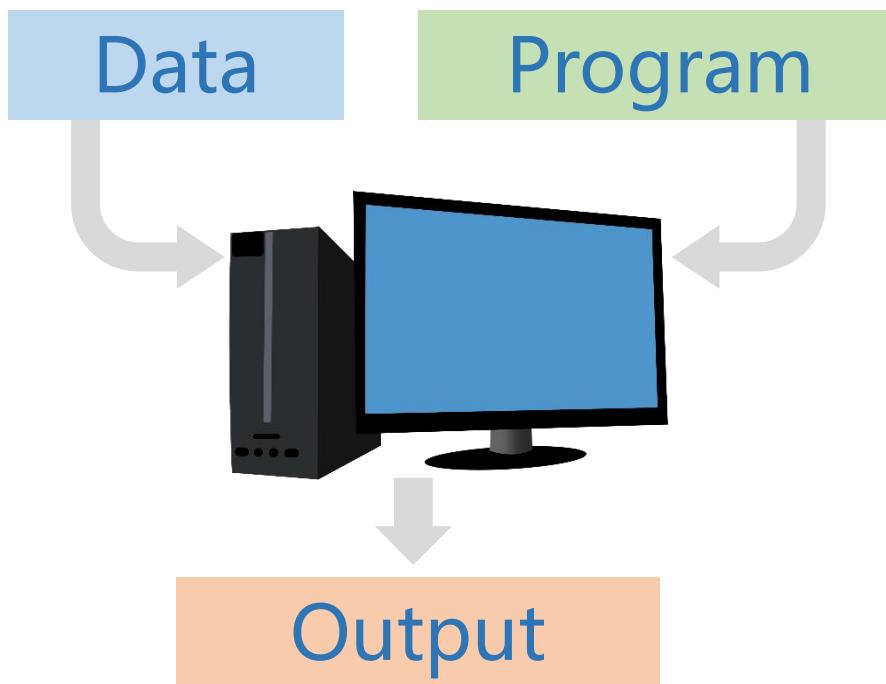
- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by $\langle P, T, E \rangle$

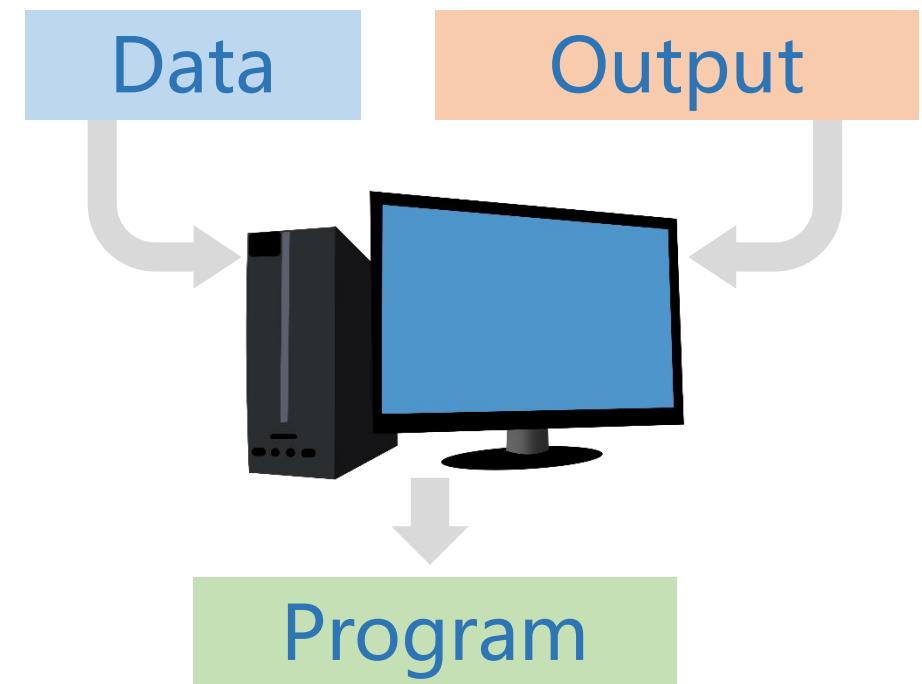


Recall: what is machine learning?

Traditional Programming



Machine Learning



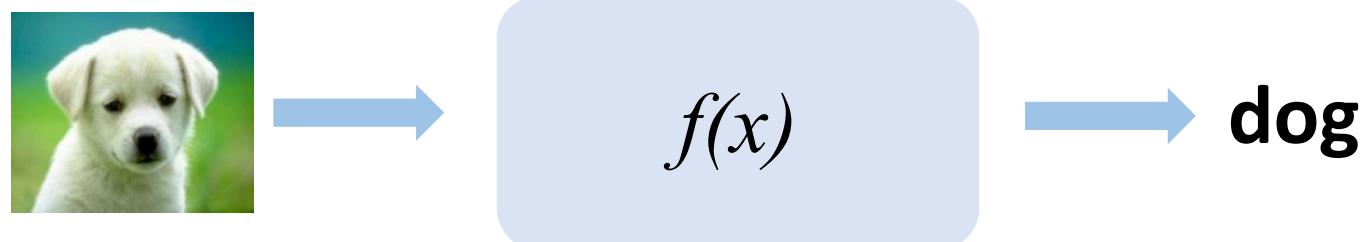


Recall: what is machine learning?

Composition of non-linear transformation of the data

Goal: learn useful representations directly from data

Supervised Learning:



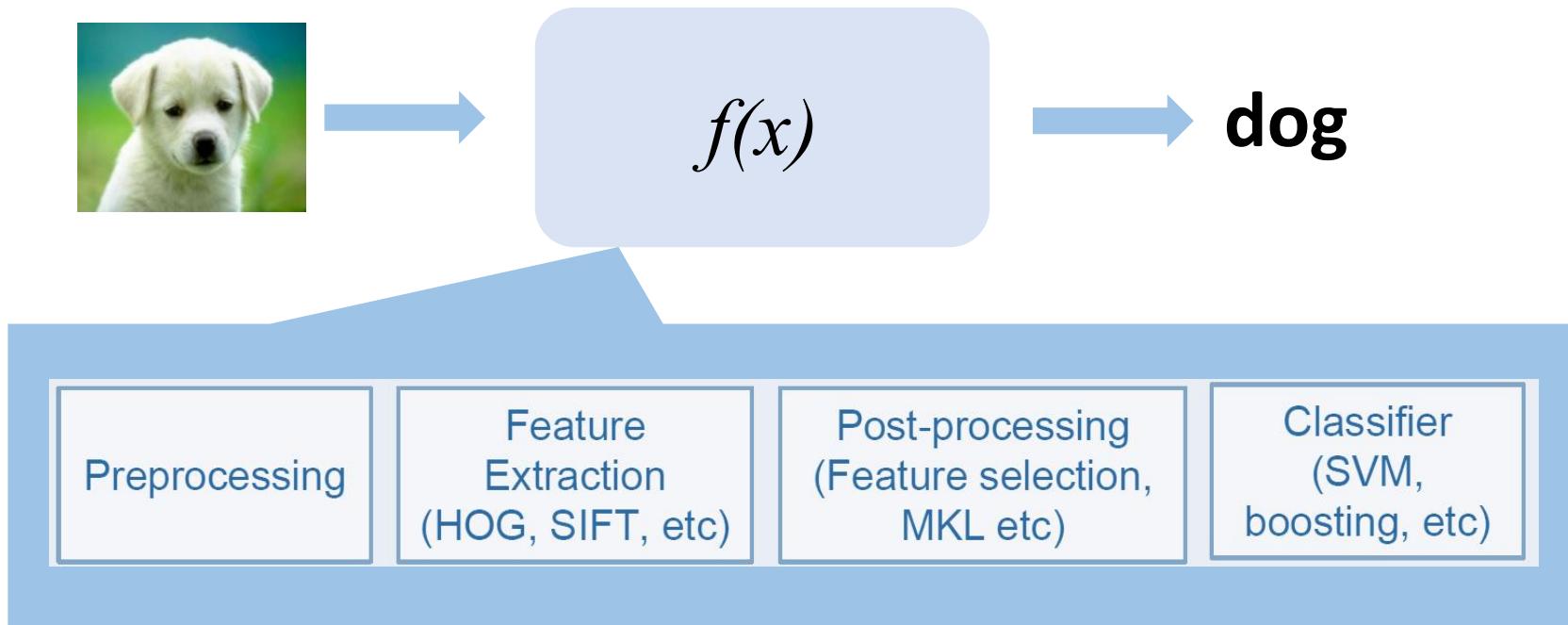


Recall: what is machine learning?

Composition of non-linear transformation of the data

Goal: learn useful representations directly from data

Supervised Learning:



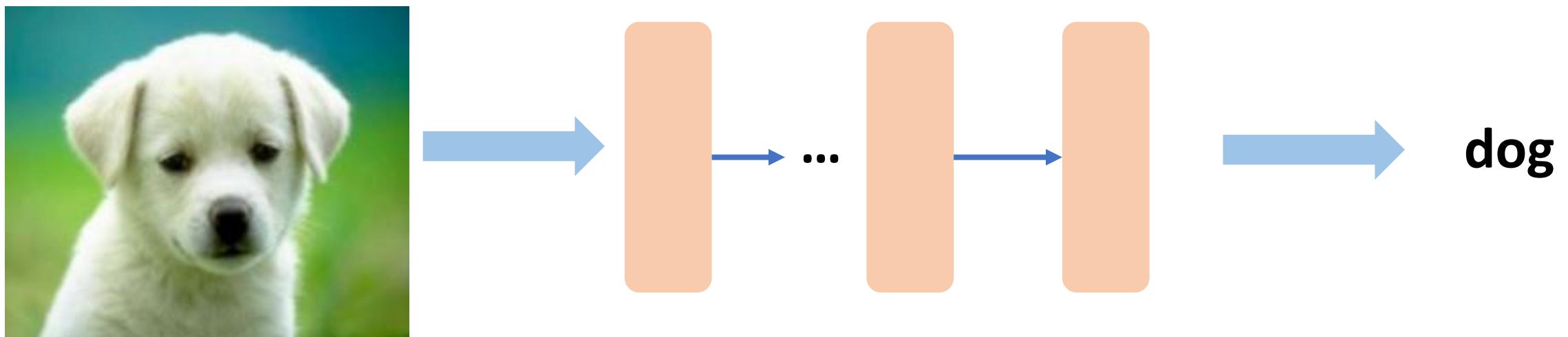


Neural Networks

Composition of non-linear transformation of the data

Goal: learn useful representations directly from data

Supervised Learning:



*Each box is a simple nonlinear function, and has trainable parameters.
Their composition makes a highly non-linear system.*



Neural Networks



Neurons that fire together, wire together

——Donald Hebb



深度学习模型受到了**神经系统层次化结构特性**的启发，以“层层递进、逐层抽象”机制完成特定任务。

1949年心理学家唐纳德·赫布提出了有名的赫布理论(Hebbian theory)：

神经元之间持续重复经验刺激可导致突触传递效能增加(Neurons that fire together, wire together)，即学习和记忆的本质是大脑结构在后期经验刺激下改变神经元之间强弱联结的权重的结果。



Neural Networks

1943年，神经科学家瓦伦·麦卡洛克 (Warren McCulloch) 和逻辑学家沃尔特·皮茨 (Walter Pitts) 合作提出了“**McCulloch–Pitts (MCP) neuron**”的思想。

历史意义：我们在科学史上第一次知道了我们是怎么知道的 (for the first time in the history of science, we know how we know)"

神经网络基本单元
MCP神经元

1949年，心理学家唐纳德·赫布 (Donald Hebb) 提出了赫布理论 (Hebbian theory)：神经元之间持续**重复经验刺激**可导致突触传递效能增加(Neurons that fire together, wire together)"

历史意义：神经元之间突触的强弱变化是学习与记忆的生理学基础

神经元因何链接
赫布理论

20世纪50年代弗朗克·罗森布拉特 (Frank Rosenblatt) 提出“**感知机** (perceptron) ”模型，其是一个仅包含输入层和输出层的两层神经网络，可完成两类问题的线性分类。

历史意义：感知机模型被誉为“Electronic 'Brain' Teaches Itself (能自学的电脑) ”

神经元链接成 “网”
感知机



Neural Networks

1974年Werbos在博士论文中提出**误差后向传播**算法，后由鲁梅尔哈特(Rumelhart)和欣顿(Hinton)等人(Rumelhart et al., 1986)完善，解决了多层感知机中参数优化这一难题。

历史意义：以数据驱动方式来根据输出误差自动优化神经网络参数成为可能

神经元之间刺激可层层递进学习
误差后向传播

2006年，欣顿在Science等期刊上发表论文，首次提出“深度信念网络(deep belief network)”模型(Hinton et al., 2006)，在相关分类任务中可取得性能超过传统浅层学习模型(如支持向量机)的结果。

历史意义：有别于分段式学习，端到端(end to end)的深度学习框架建立

逐层抽象、层层递进
深度学习

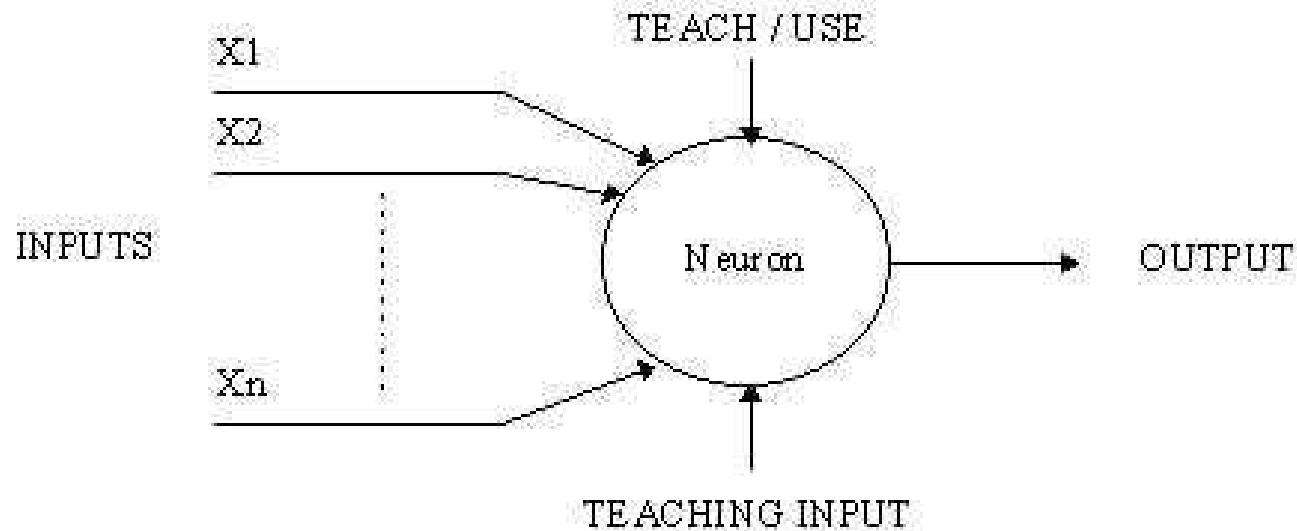
- RNN、LSTM
- CNN
- Word2vec
- BERT、Transformer、LLM

历史意义：从造人和造脑到更通用人工智能

引发AI第三次崛起
语言大模型



Neural Networks: a simple neuron

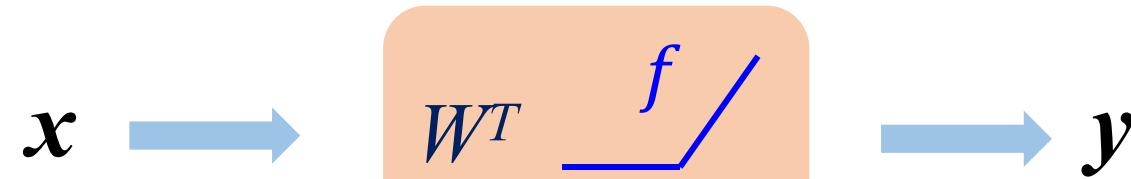


- An artificial neuron is a device with many inputs and one output.
- The neuron has two modes of operation;
- The training mode and
- The using mode.



Neural Networks: 1 layer example

A simple single layer neural network consists of a linear combination of input through a nonlinear function:



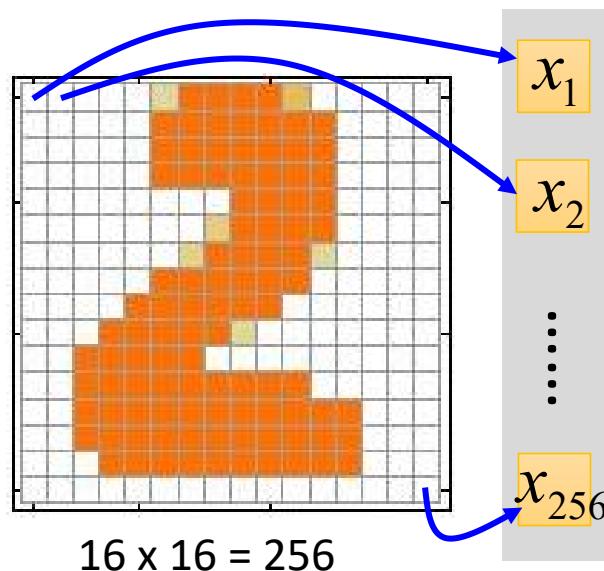
$$z = \mathbf{W}x + b$$

$$a = f(z)$$

- W is the weight parameter to be learned
- b is the bias parameter
- x is the output of the previous layer
- f is a simple nonlinear function, called activate function. Popular choice is $\max(x, 0)$, called ReLU (rectified linear unit).

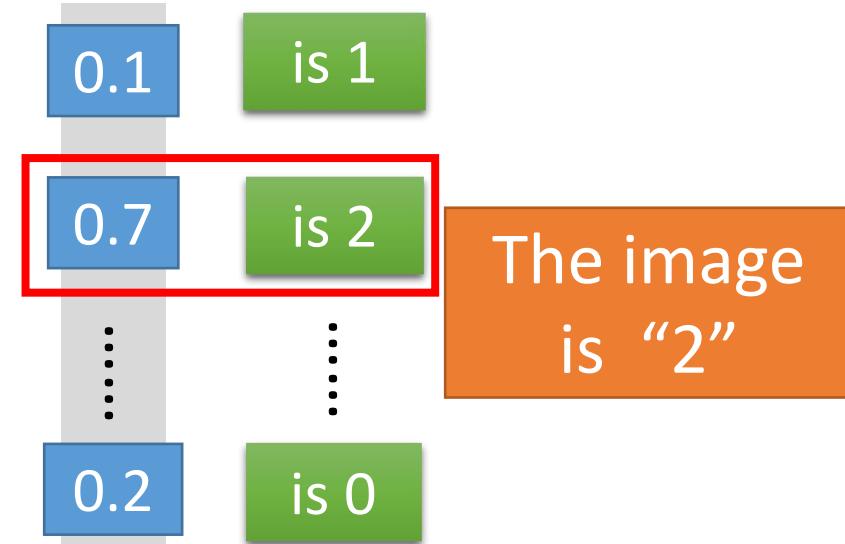
Handwriting Digit Recognition

Input



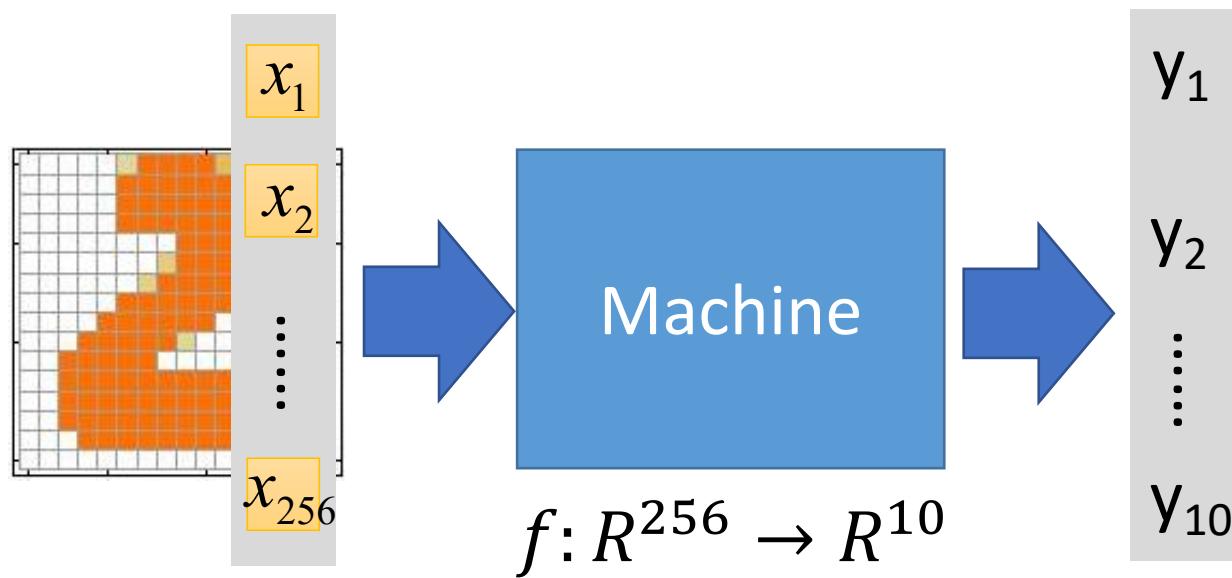
Ink → 1
No ink → 0

Output



Each dimension represents the confidence of a digit.

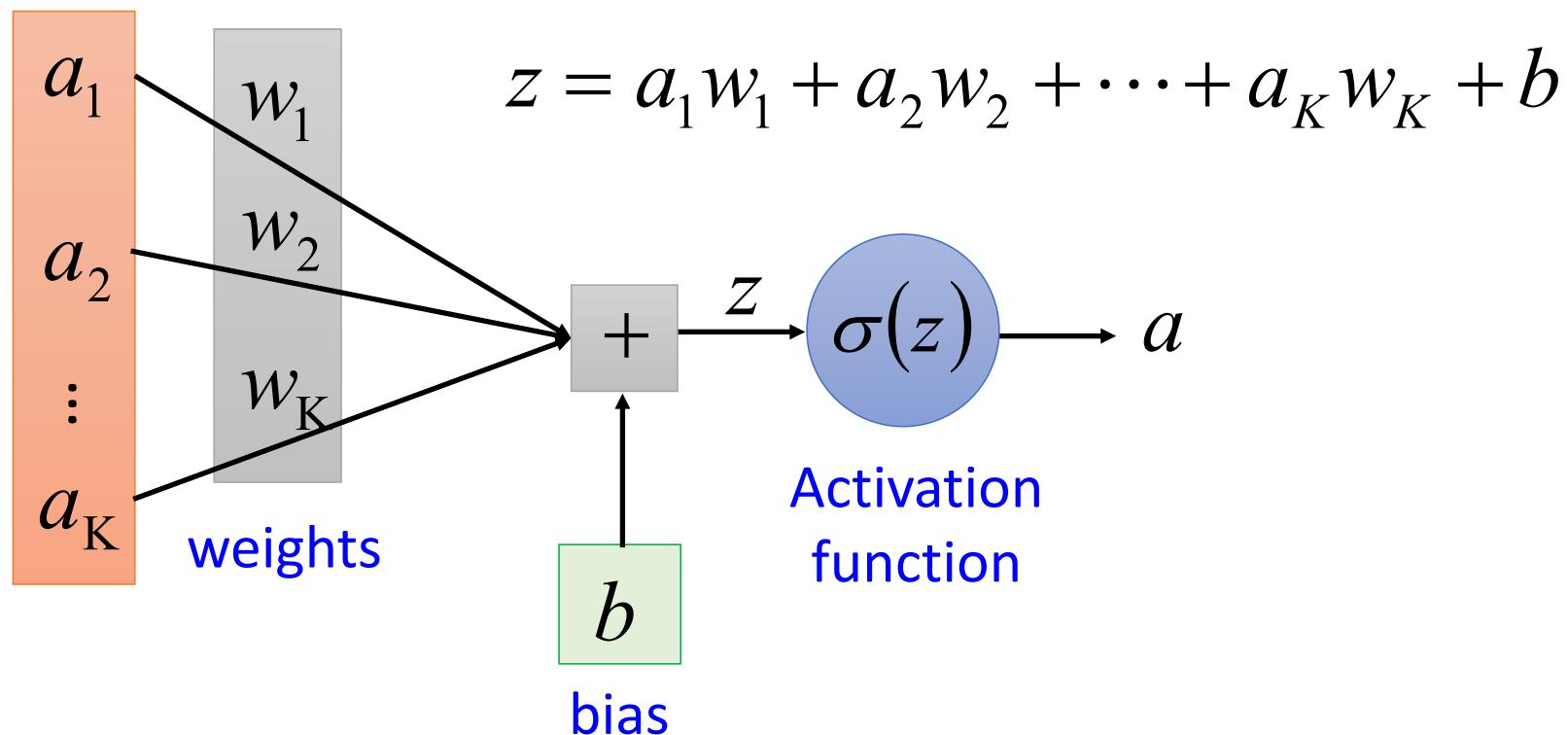
Handwriting Digit Recognition



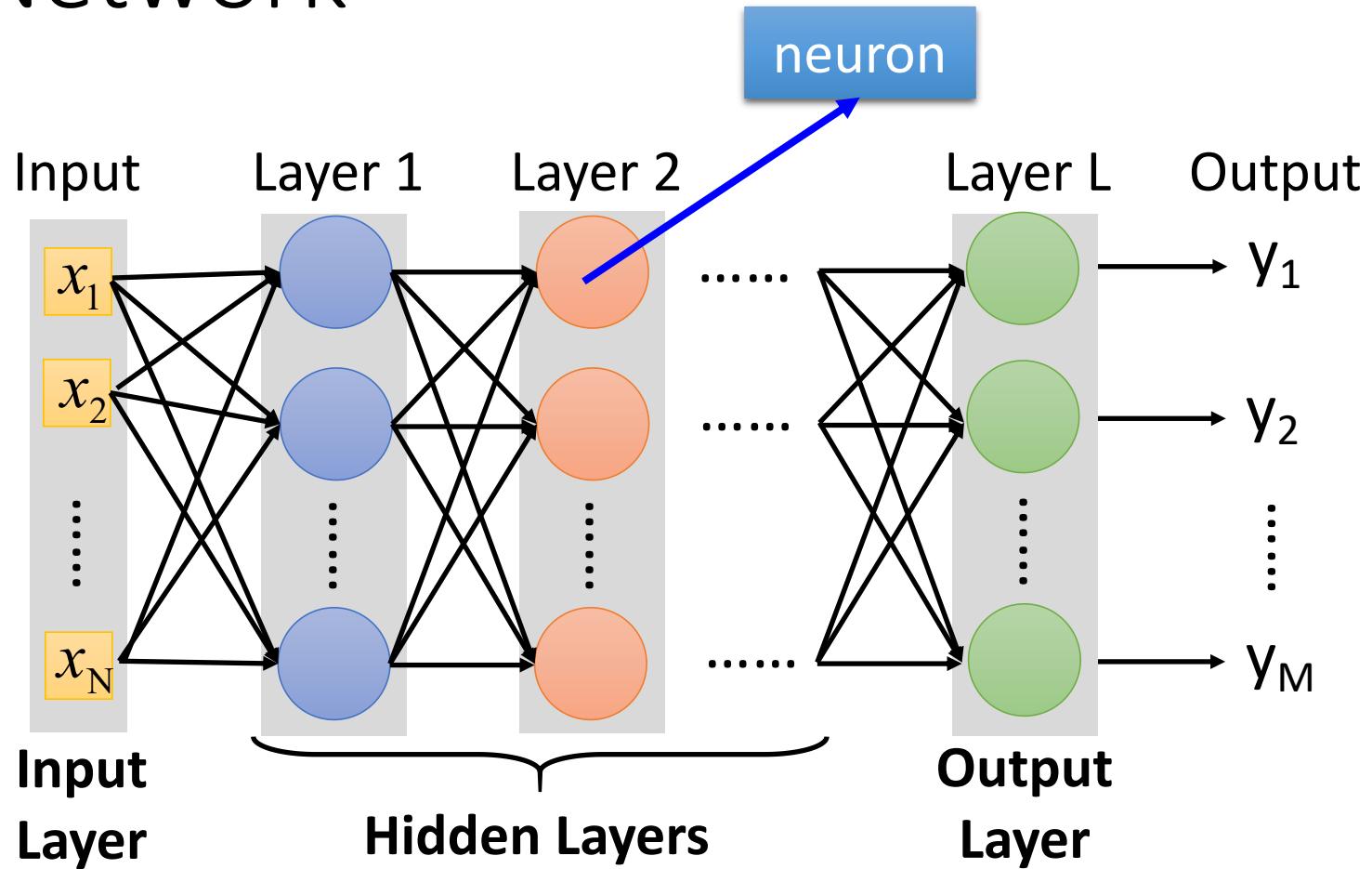
In deep learning, the function f is represented by neural network

Element of Neural Network

Neuron $f: R^K \rightarrow R$

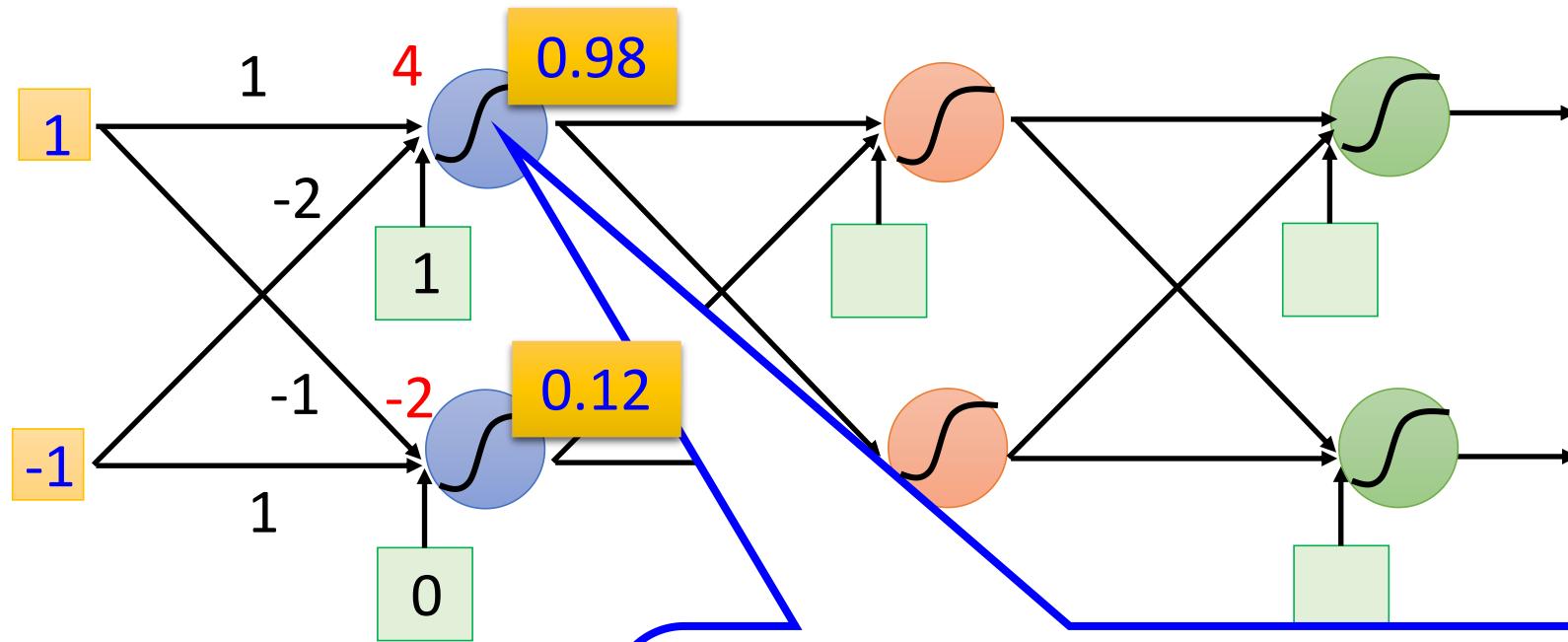


Neural Network



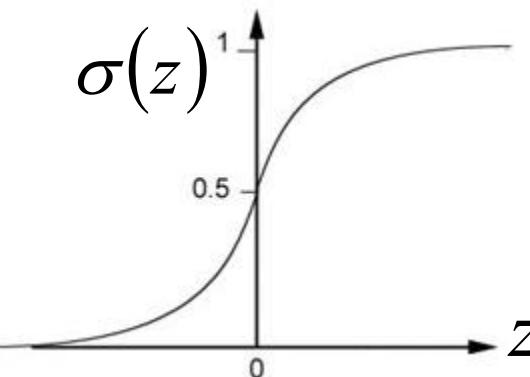
Deep means many hidden layers

Example of Neural Network

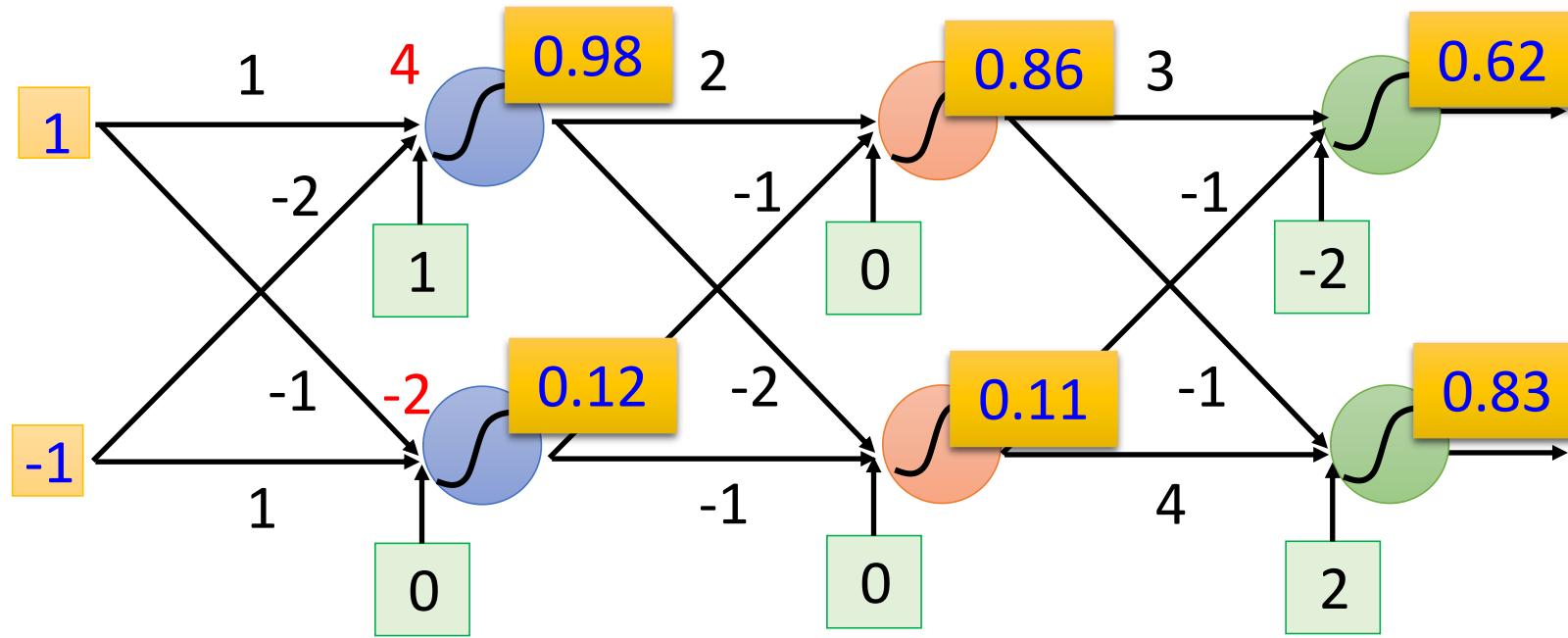


Sigmoid Function

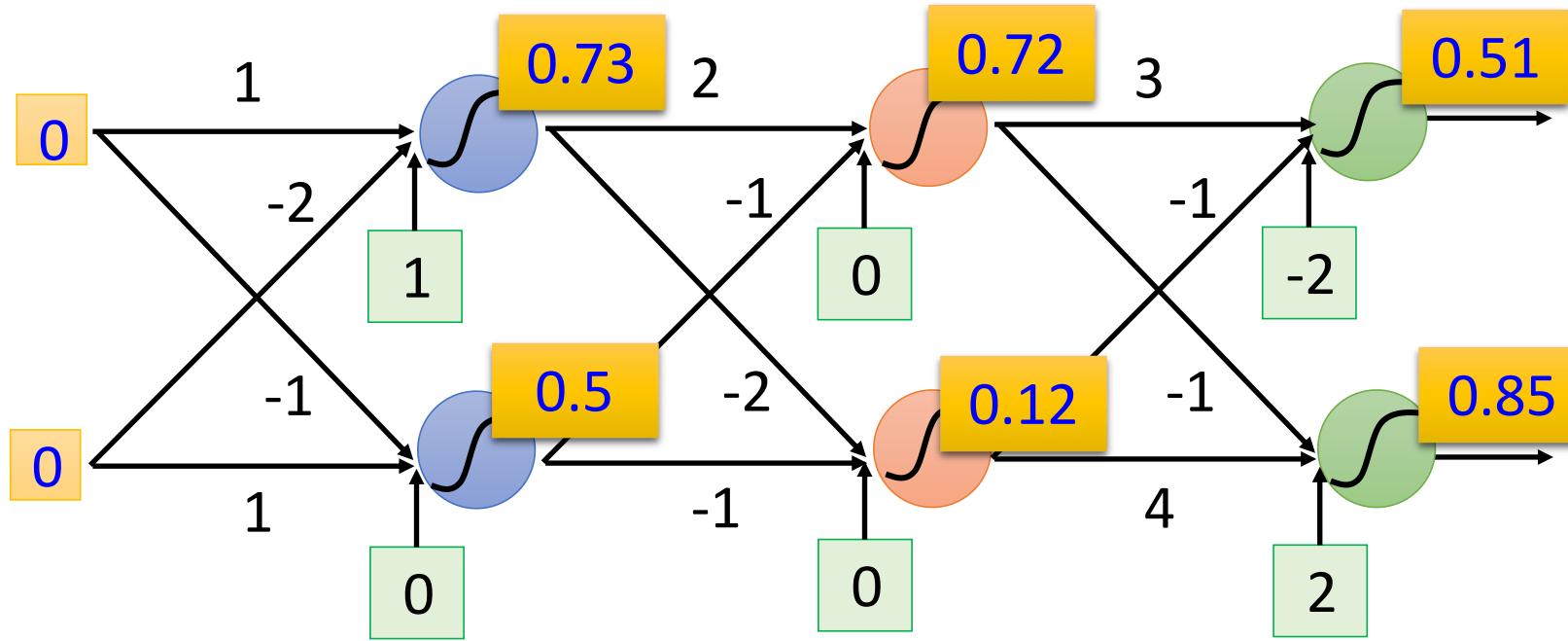
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Example of Neural Network



Example of Neural Network

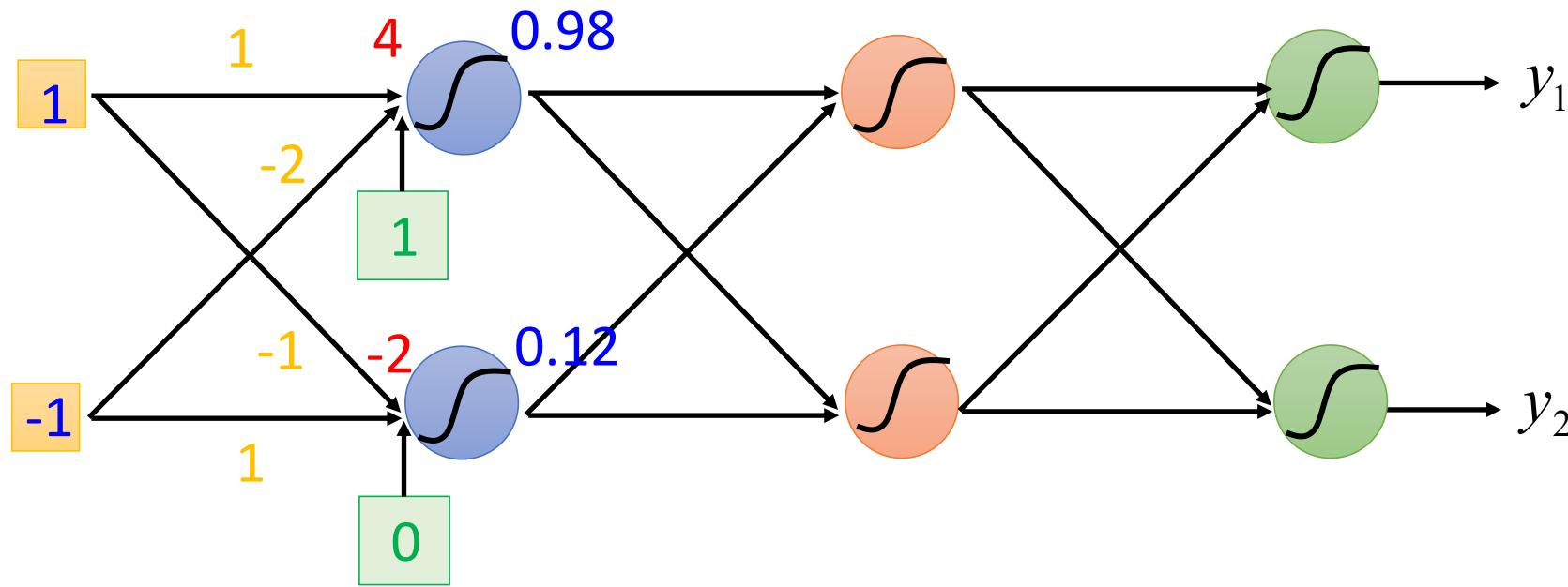


$$f: R^2 \rightarrow R^2$$

$$f \left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) = \begin{bmatrix} 0.62 \\ 0.83 \end{bmatrix} \quad f \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0.51 \\ 0.85 \end{bmatrix}$$

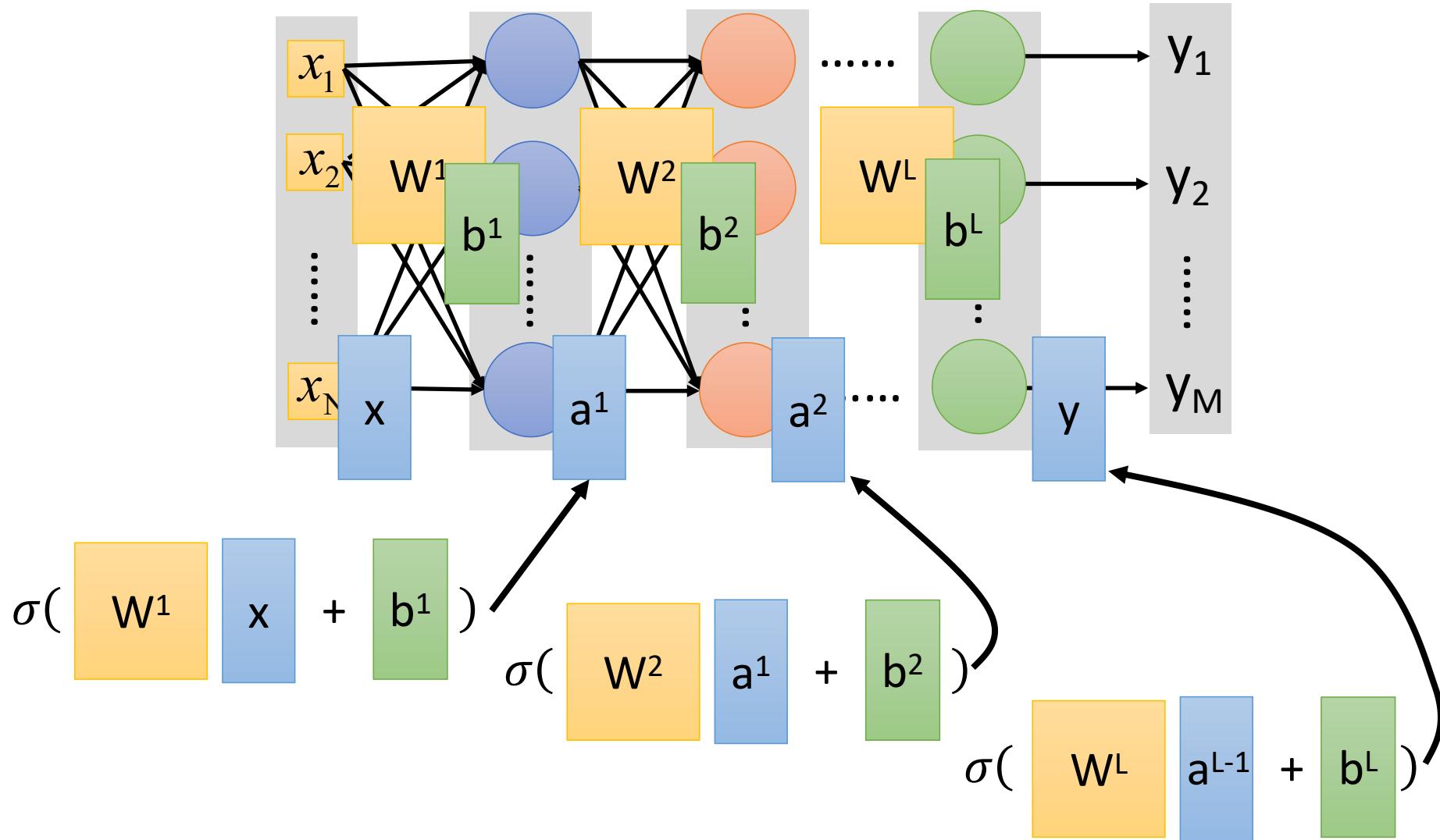
Different parameters define different function

Matrix Operation

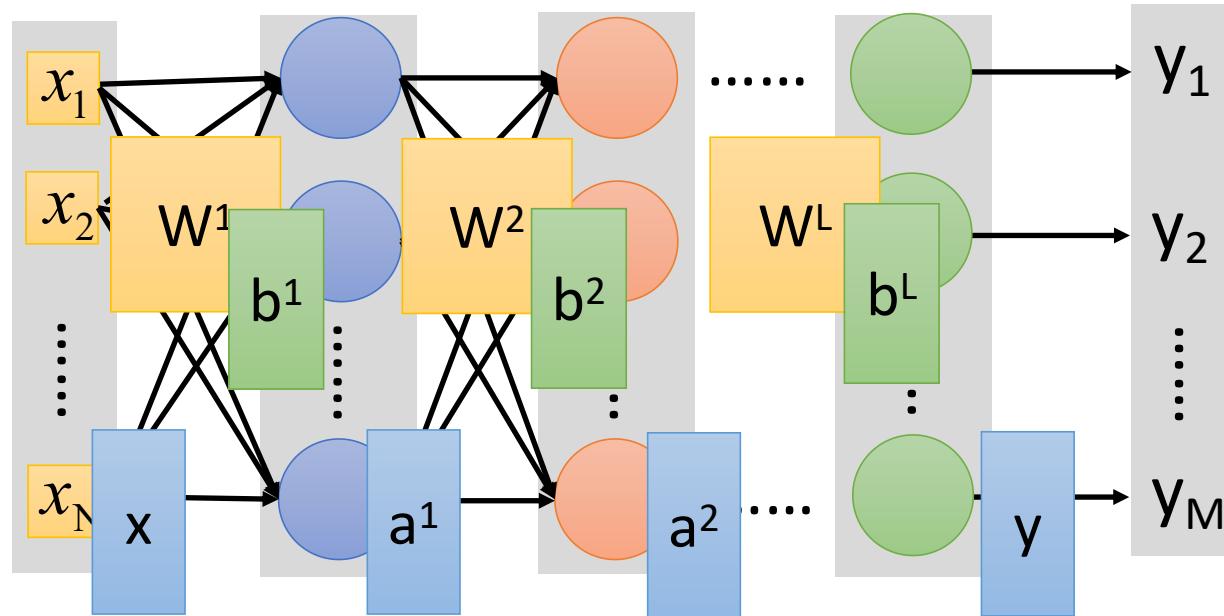


$$\sigma \left(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

Neural Network



Neural Network



$$y = f(x)$$

$$= \sigma(W^L \dots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \dots + b^L)$$

Softmax

- Softmax layer as the output layer

Ordinary Layer

$$z_1 \rightarrow \sigma \rightarrow y_1 = \sigma(z_1)$$

$$z_2 \rightarrow \sigma \rightarrow y_2 = \sigma(z_2)$$

$$z_3 \rightarrow \sigma \rightarrow y_3 = \sigma(z_3)$$

In general, the output of network can be any value.

May not be easy to interpret

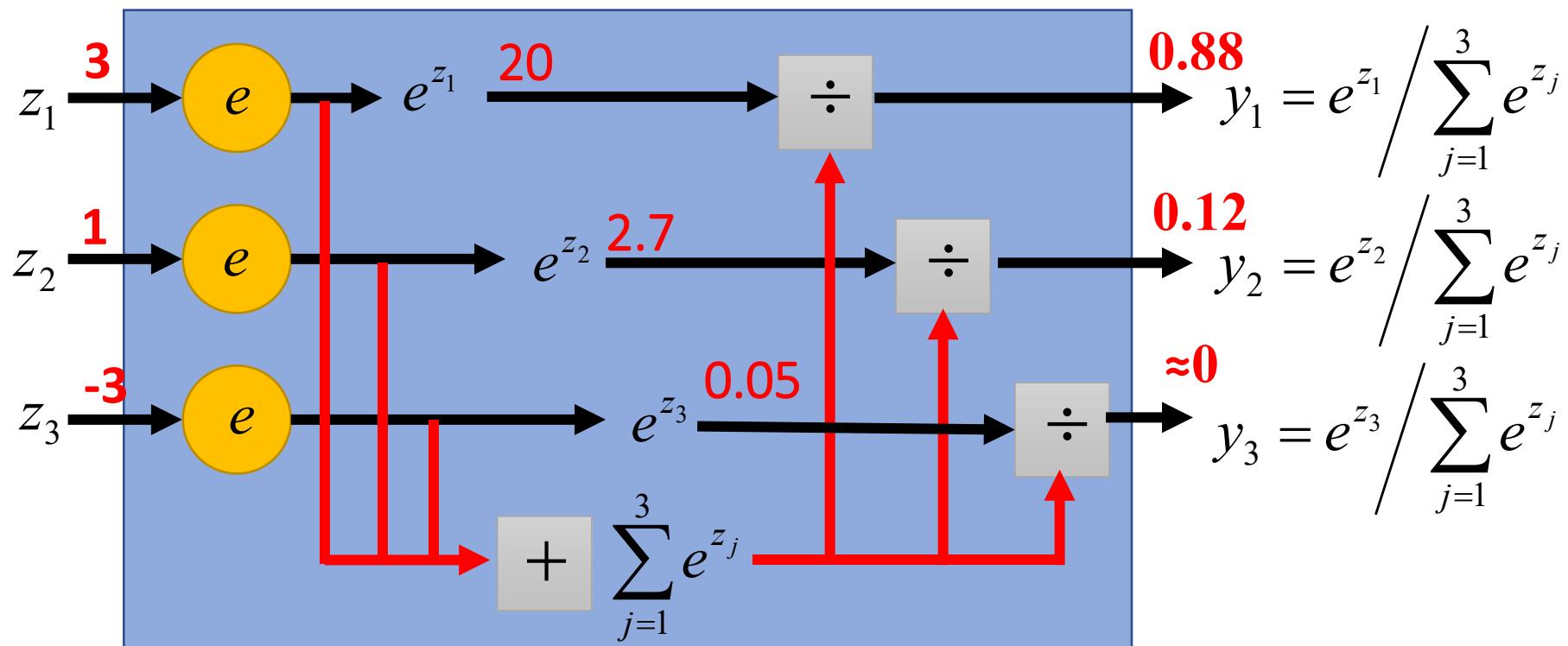
Softmax

- Softmax layer as the output layer

Probability:

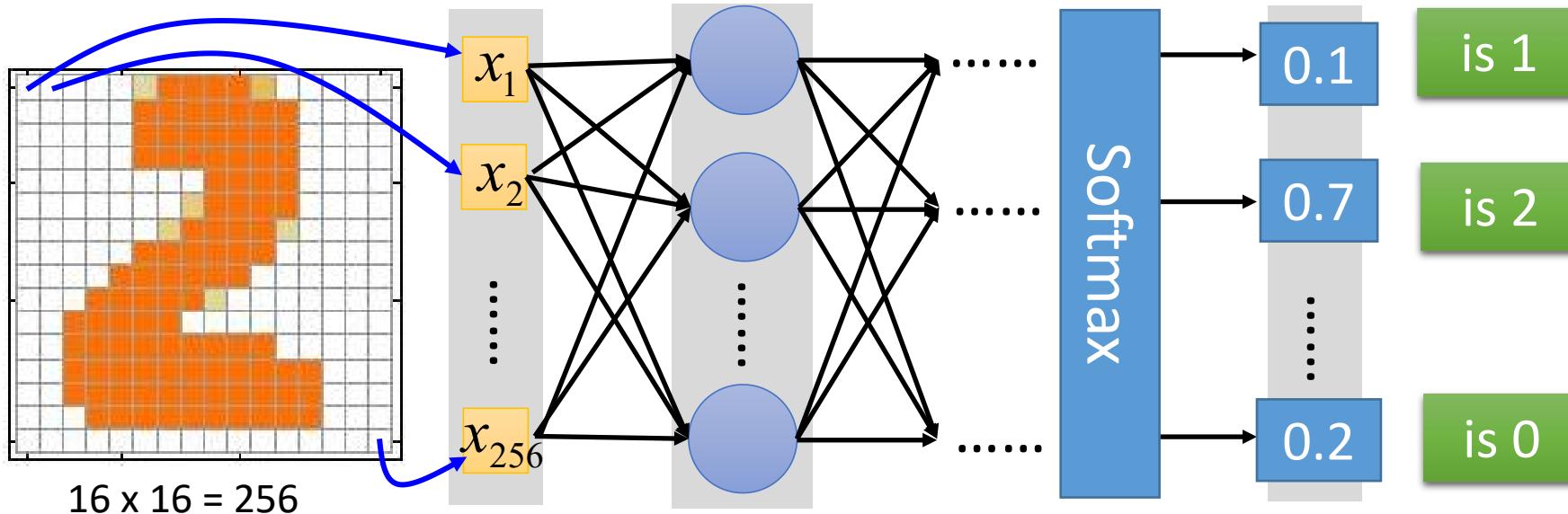
- $1 > y_i > 0$
- $\sum_i y_i = 1$

Softmax Layer



How to set network parameters

$$\theta = \{W^1, b^1, W^2, b^2, \dots, W^L, b^L\}$$



Ink $\rightarrow 1$

No ink $\rightarrow 0$

Set the network parameters θ such that

Input: How to let the neural network achieve this m value

Input: $\rightarrow y_2$ has the maximum value

Training Data

- Preparing training data: images and their labels



“5”



“0”



“4”



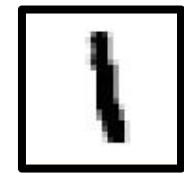
“1”



“9”



“2”



“1”

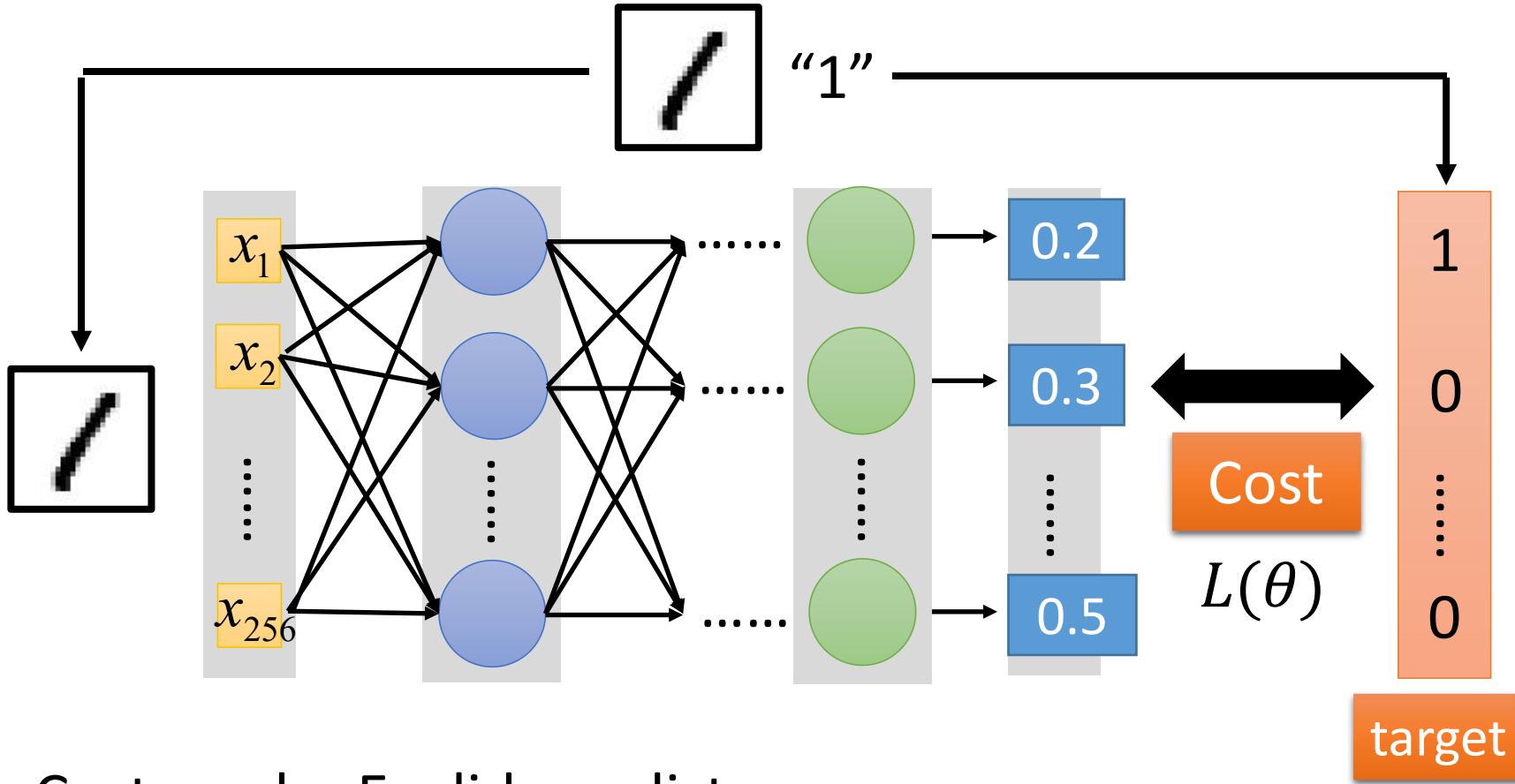


“3”

Using the training data to find
the network parameters.

Cost

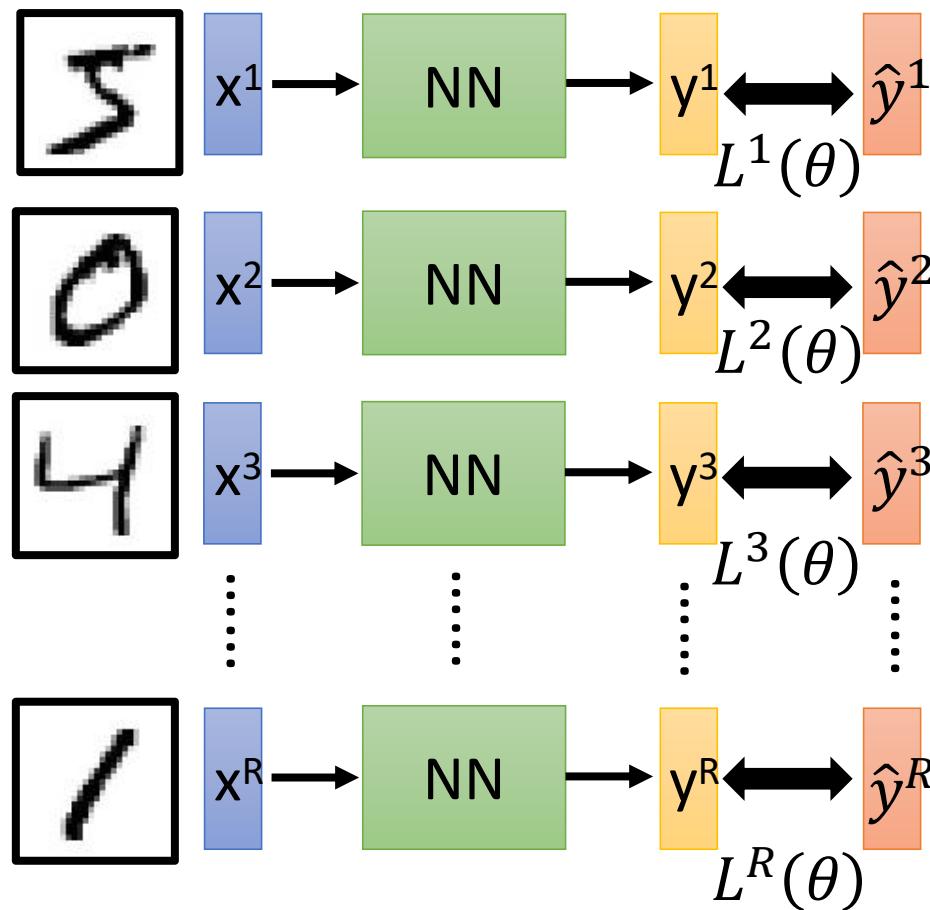
Given a set of network parameters θ , each example has a cost value.



Cost can be Euclidean distance or cross entropy of the network output and target

Total Cost

For all training data ...



Total Cost:

$$C(\theta) = \sum_{r=1}^R L^r(\theta)$$

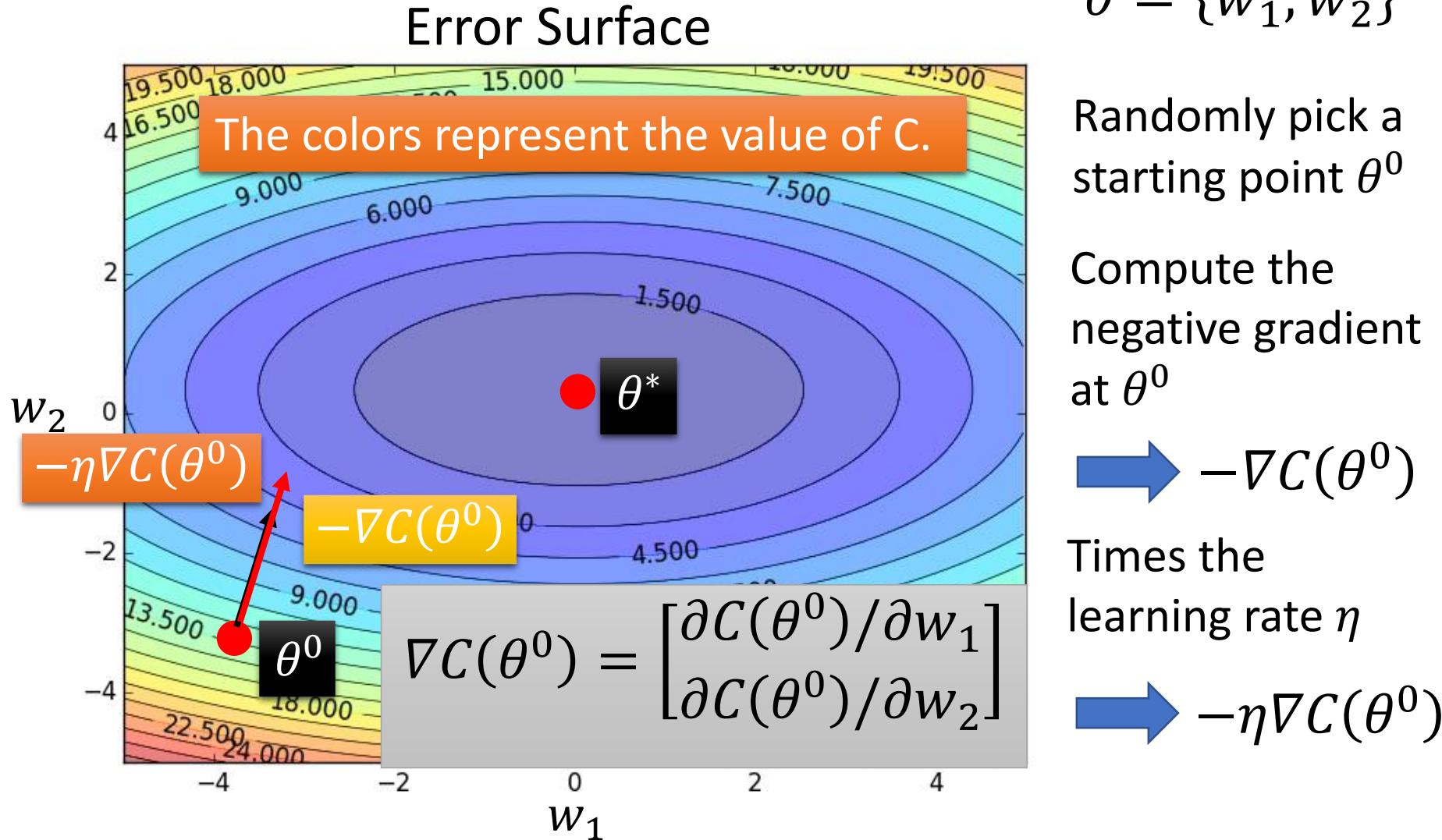
How bad the network parameters θ is on this task

Find the network parameters θ^* that minimize this value

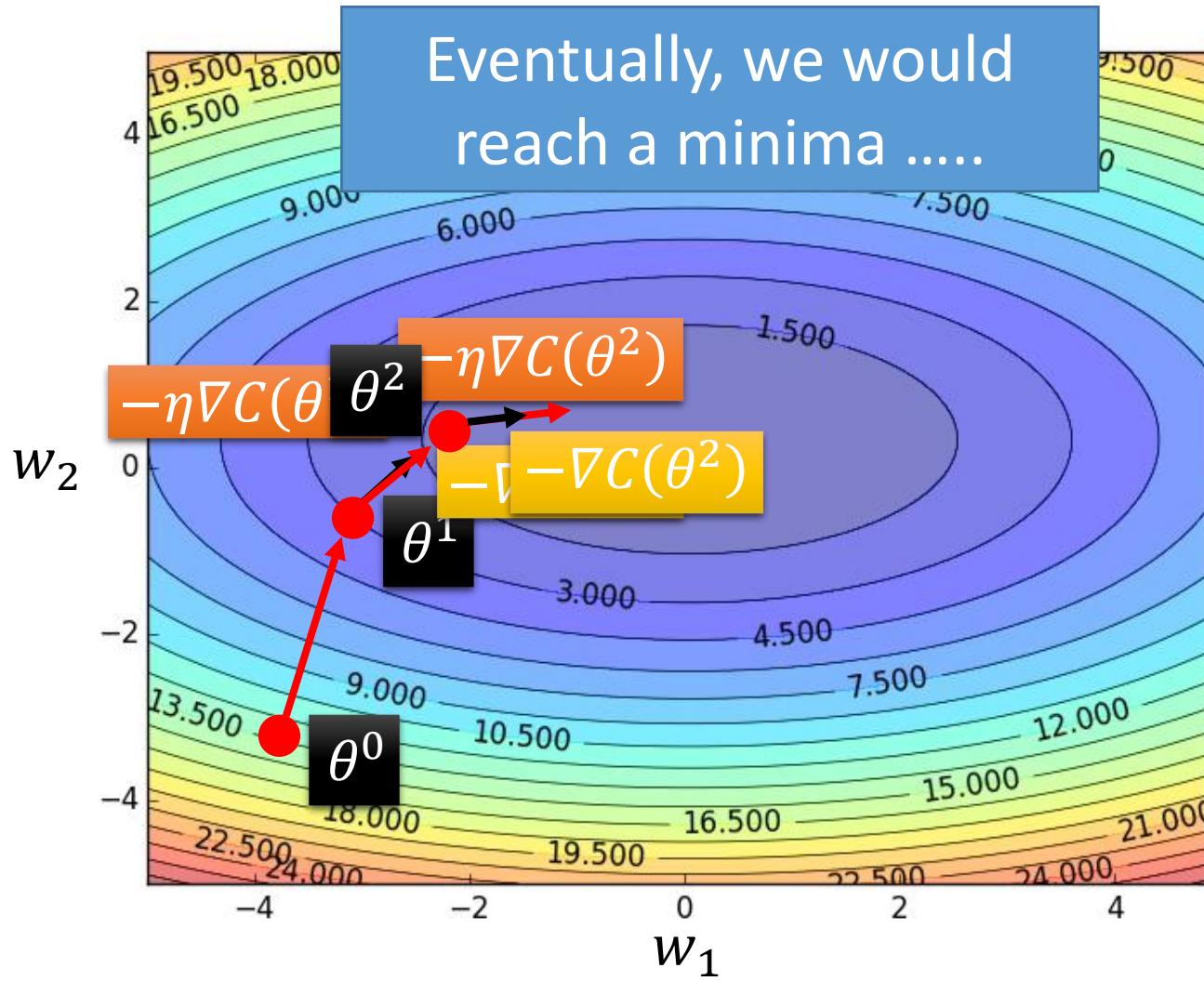
Gradient Descent

Assume there are only two parameters w_1 and w_2 in a network.

$$\theta = \{w_1, w_2\}$$



Gradient Descent



Randomly pick a starting point θ^0

Compute the negative gradient at θ^0

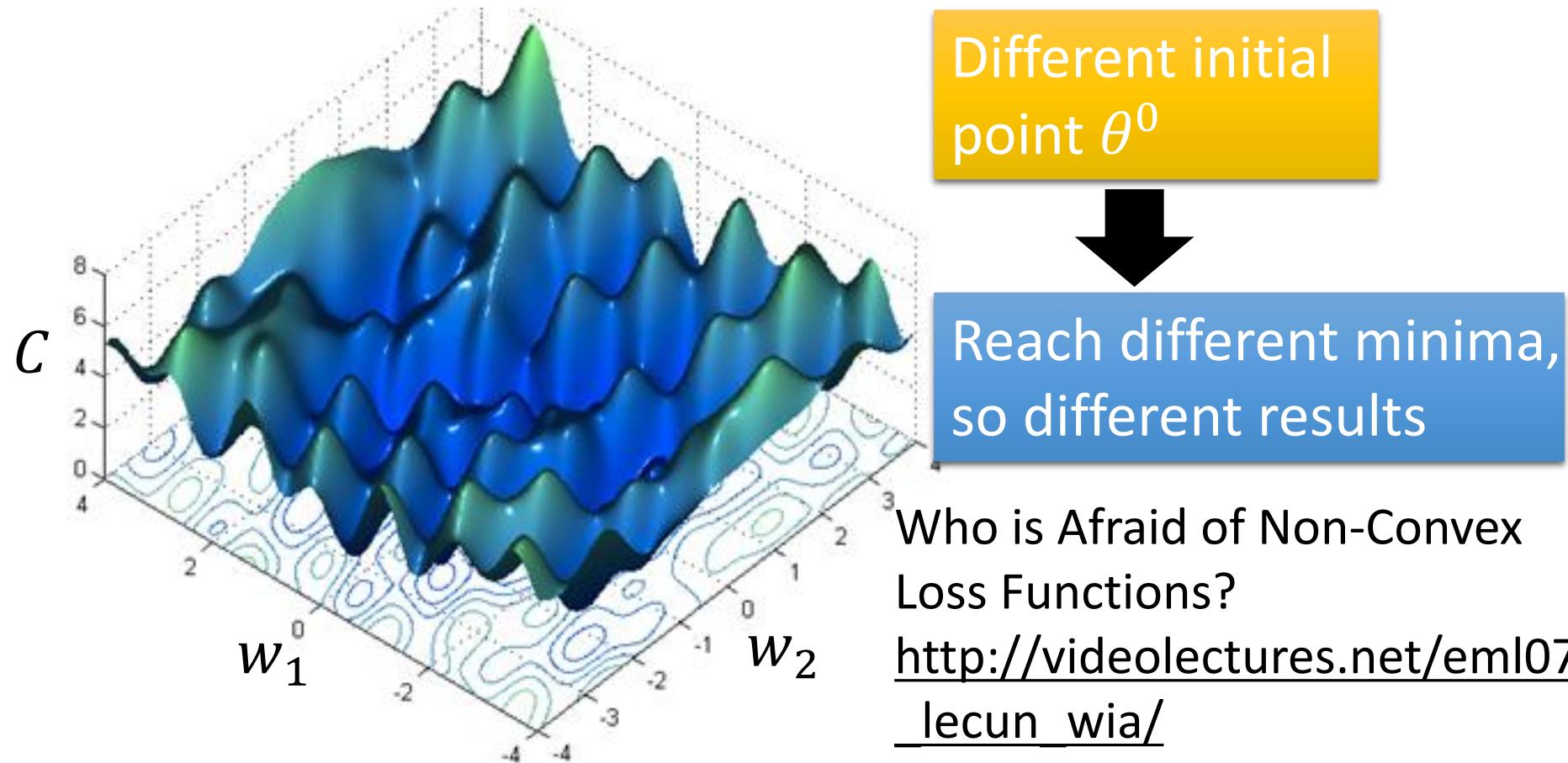
→ $-\nabla C(\theta^0)$

Times the learning rate η

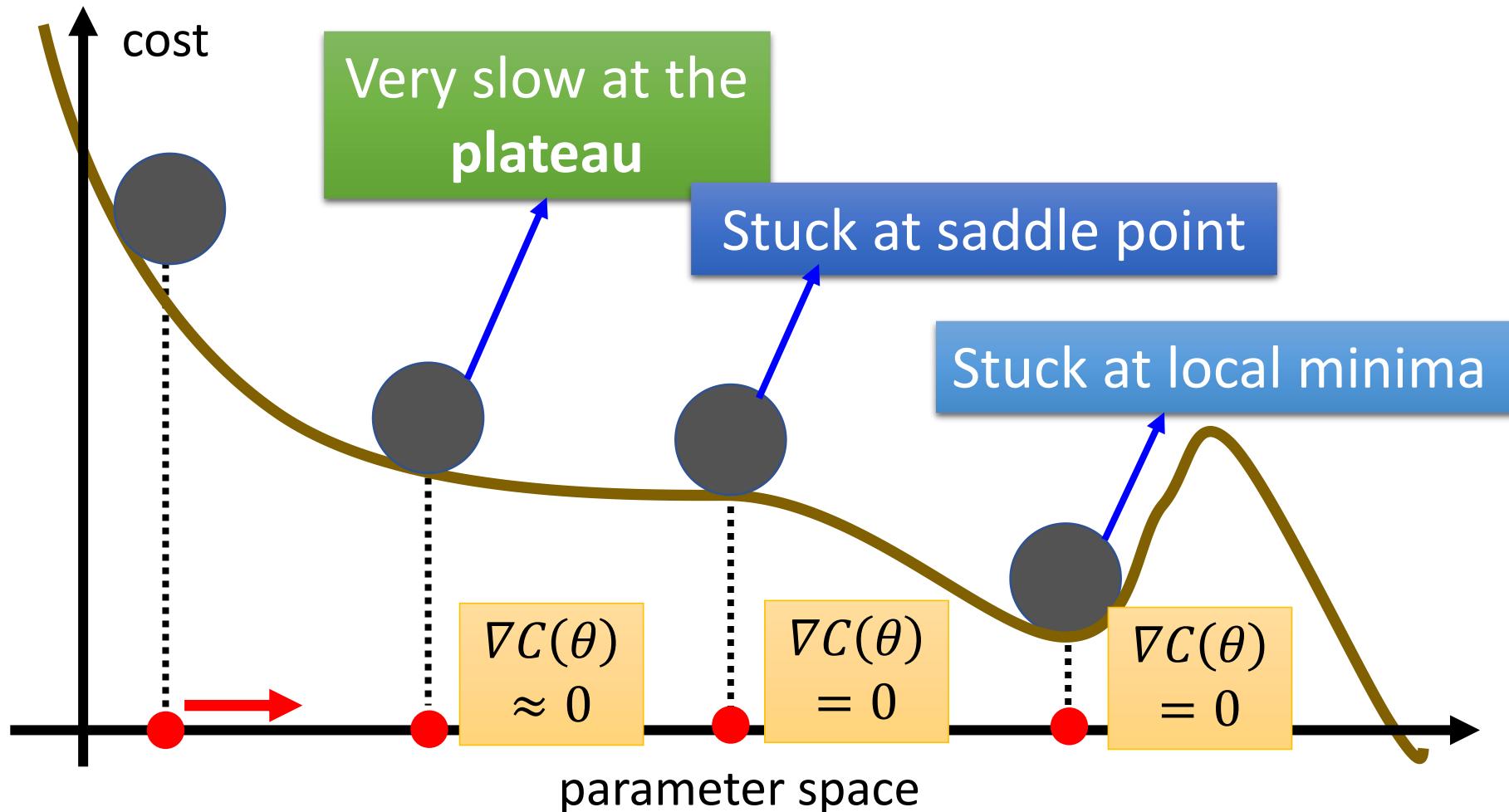
→ $-\eta \nabla C(\theta^0)$

Local Minima

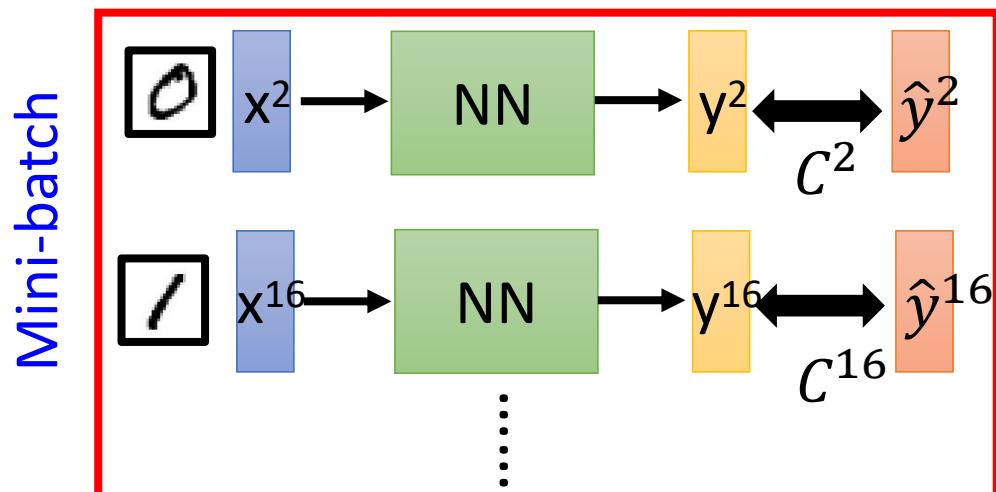
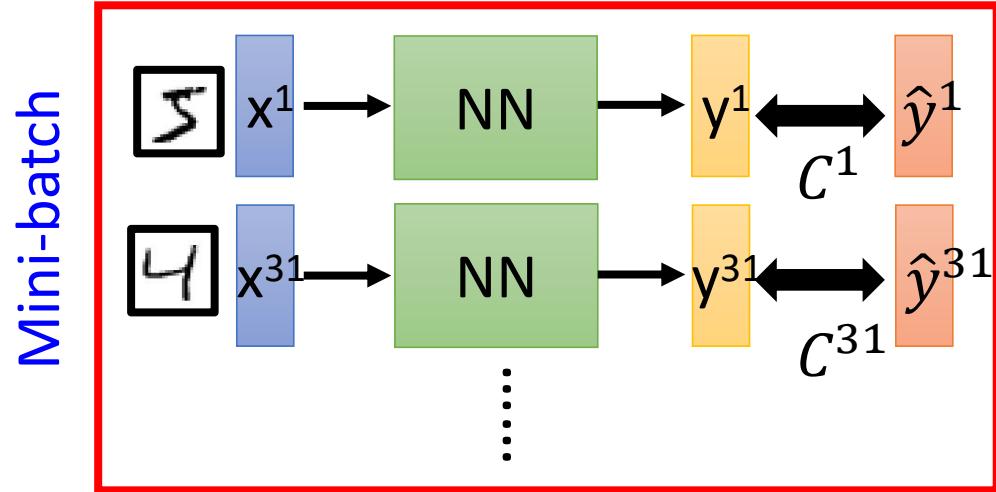
- Gradient descent never guarantee global minima



Besides local minima



Mini-batch



➤ Randomly initialize θ^0

➤ Pick the 1st batch

$$C = C^1 + C^{31} + \dots$$

$$\theta^1 \leftarrow \theta^0 - \eta \nabla C(\theta^0)$$

➤ Pick the 2nd batch

$$C = C^2 + C^{16} + \dots$$

$$\theta^2 \leftarrow \theta^1 - \eta \nabla C(\theta^1)$$

⋮

➤ Until all mini-batches have been picked

one epoch

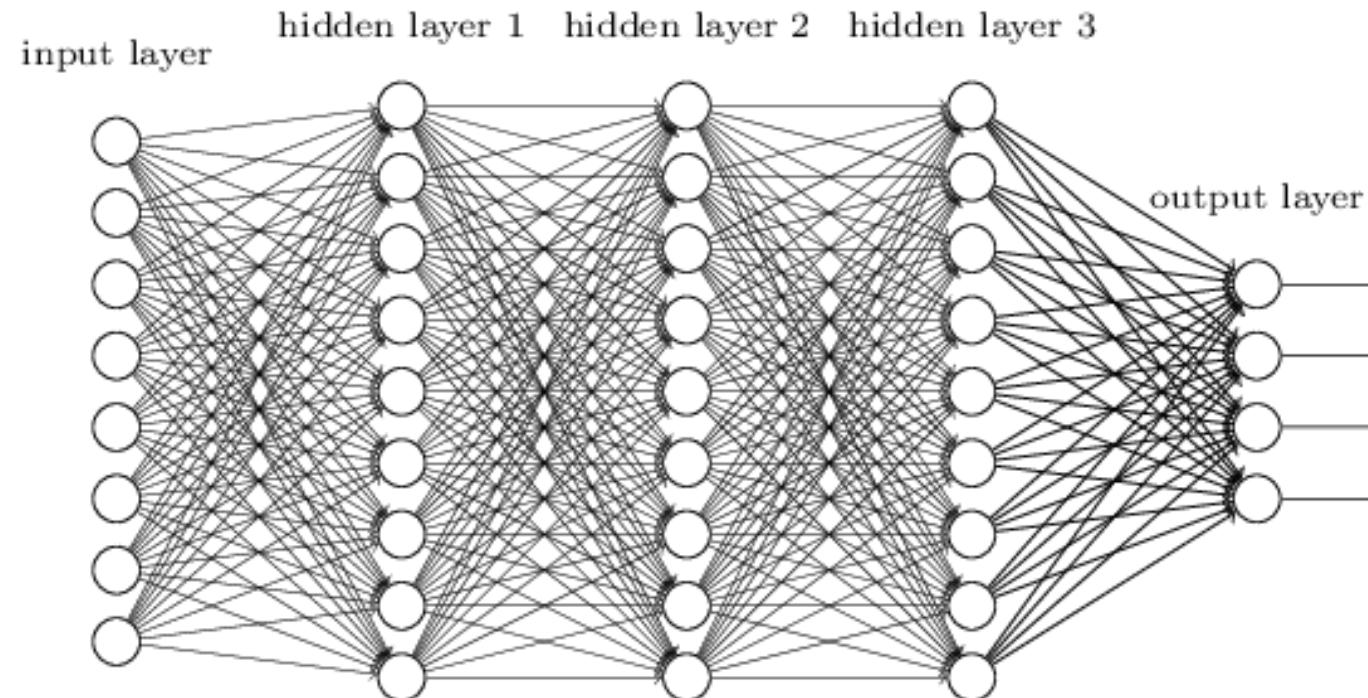
Repeat the above process

Training: Backprop algorithm

- The Backprop algorithm searches for weight values that minimize the total error of the network over the set of training examples (training set).
- Backprop consists of the repeated application of the following two passes:
 - **Forward pass:** in this step the network is activated on one example and the error of (each neuron of) the output layer is computed.
 - **Backward pass:** in this step the network error is used for updating the weights. Starting at the output layer, the error is propagated backwards through the network, layer by layer. This is done by recursively computing the local gradient of each neuron.

Smaller Network: CNN

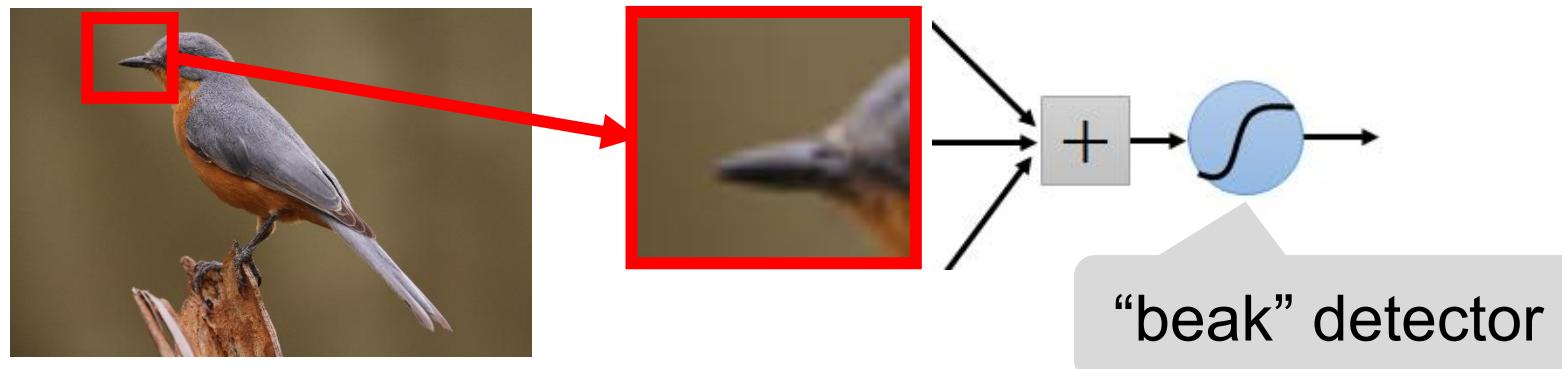
- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can some of these be shared?



Smaller Network: CNN

Observation 1: some patterns are much smaller than the whole image

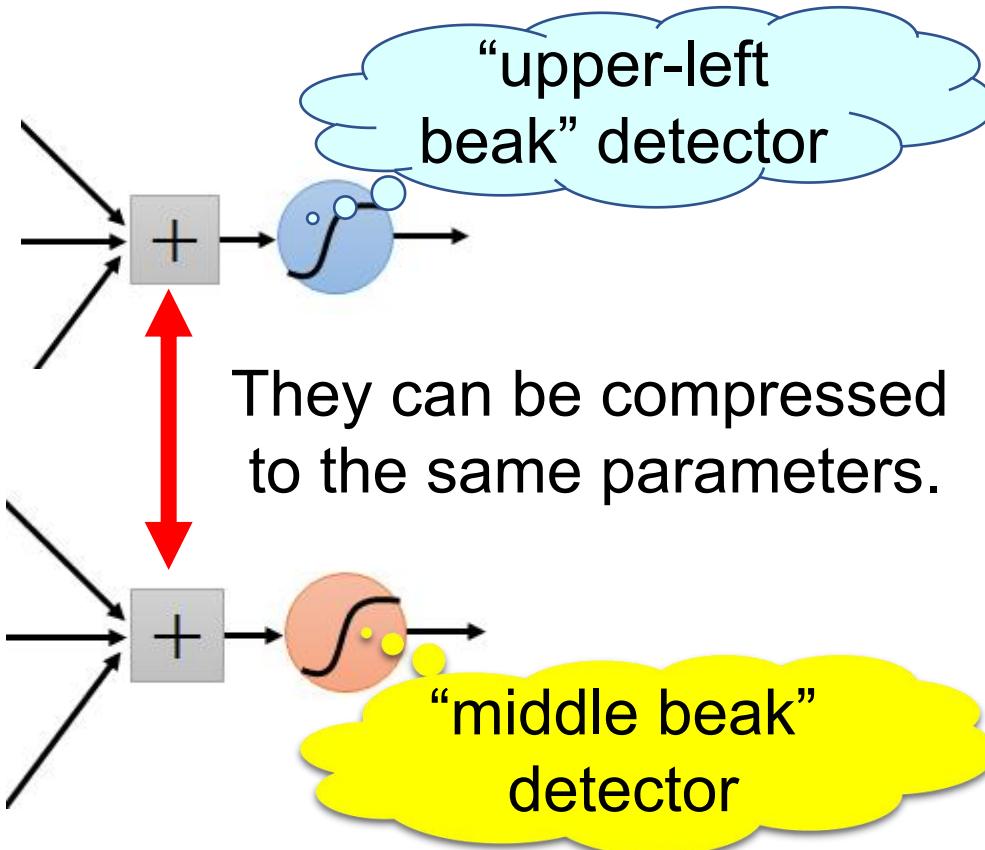
- A neuron does not have to see the whole image to discover the pattern
- Can represent a small region with fewer parameters



Smaller Network: CNN

Observation 2: same pattern appears in different places.
They can be compressed!

What about training a lot of such “small” detectors and each detector must “move around”.



Smaller Network: CNN

Observation 3: subsampling the pixels will not change the object

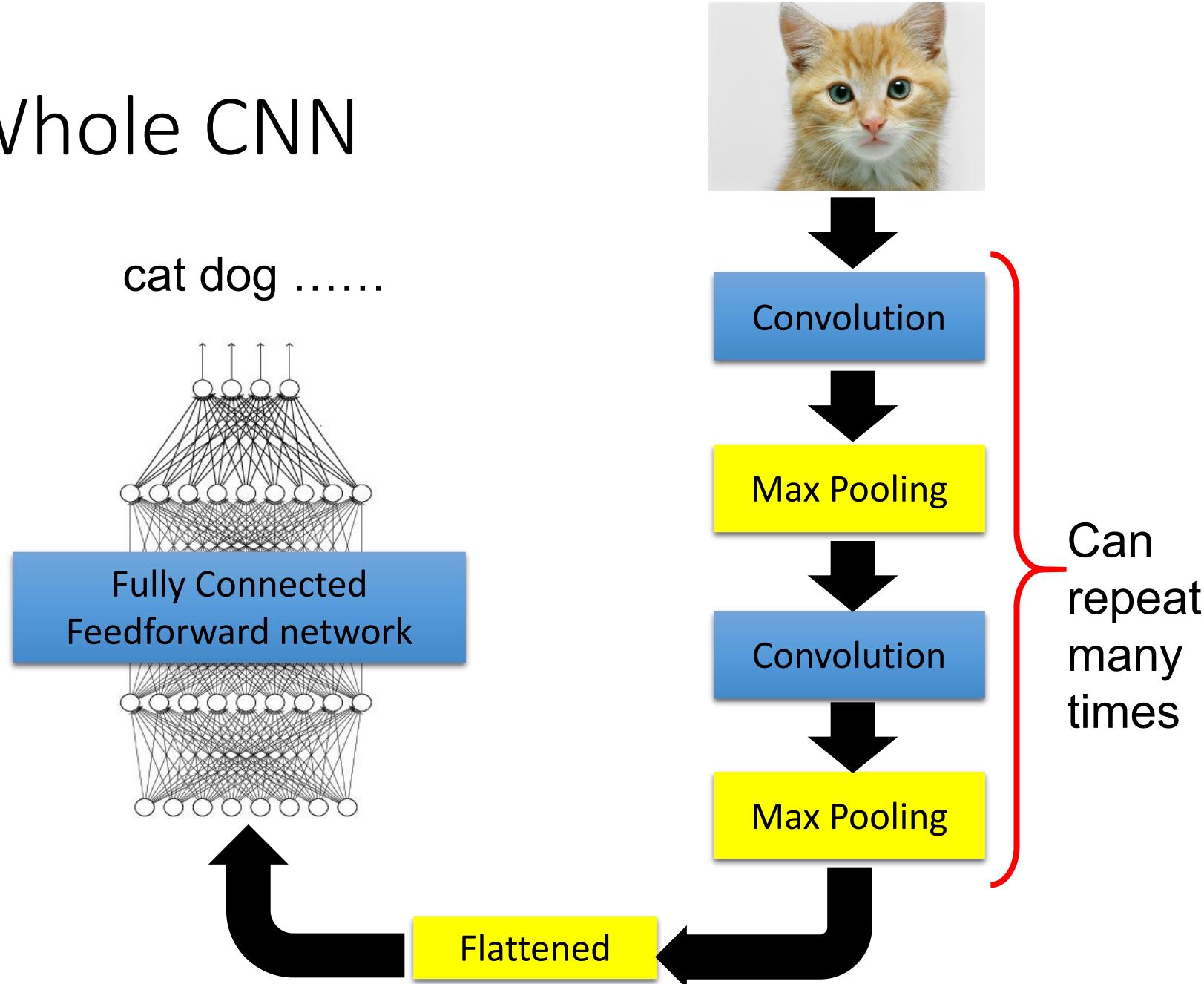
- We can subsample the pixels to be smaller



subsampling



The Whole CNN

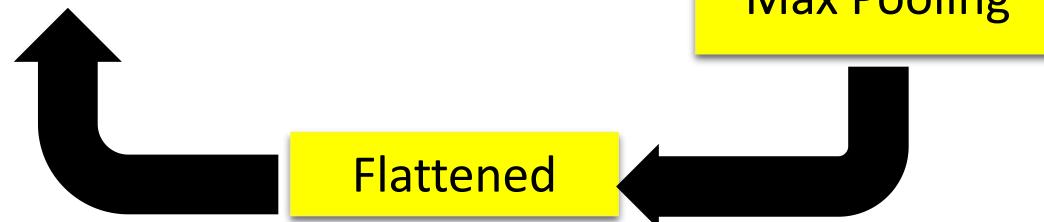


The Whole CNN

Observation 1: some patterns are much smaller than the whole image

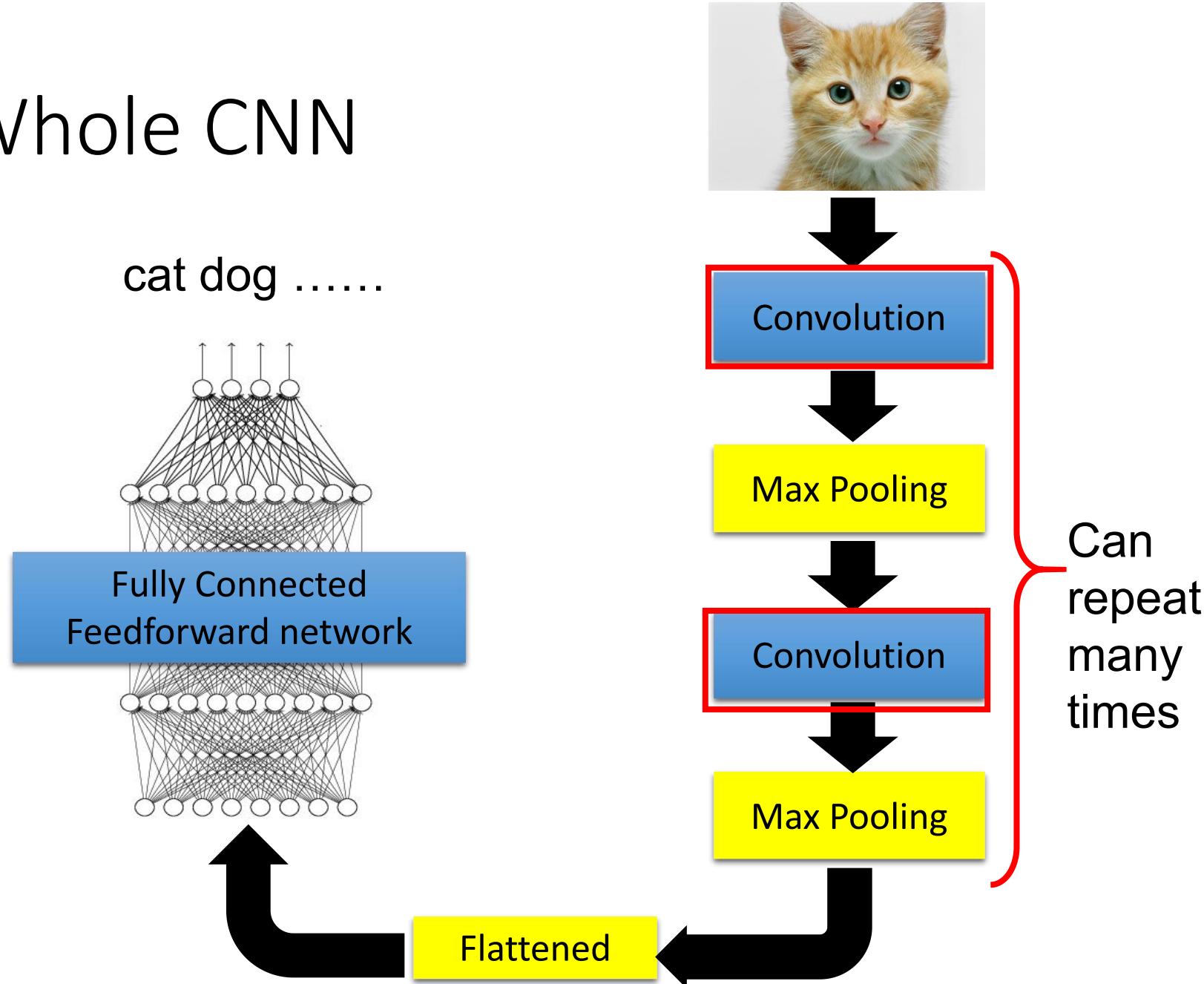
Observation 2: same pattern appears in different places.

Observation 3: subsampling the pixels will not change the object



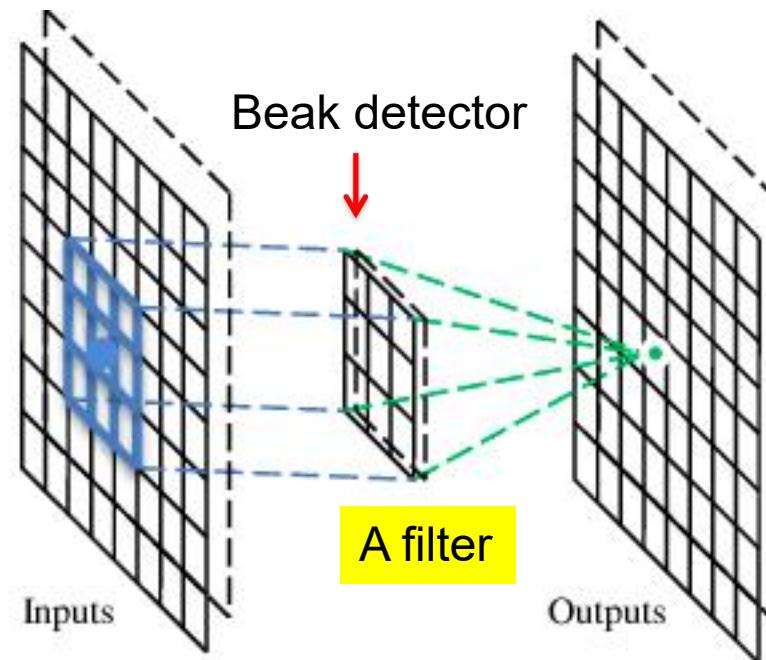
Can
repeat
many
times

The Whole CNN



A Convolutional Layer

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.



Convolution

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

6 x 6 image

These are the network parameters to be learned.

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

:

Each filter detects a small pattern (3 x 3).

Convolution

stride=1

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

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1



Convolution

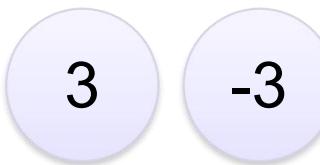
If stride=2

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

6 x 6 image

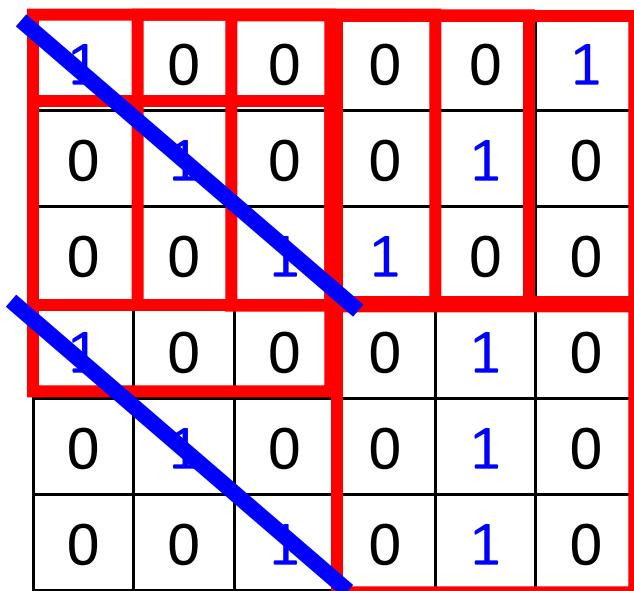
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

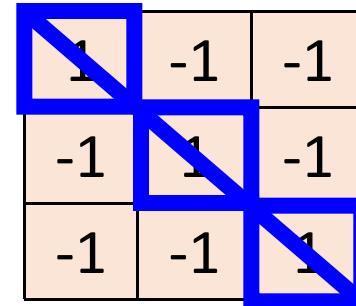


Convolution

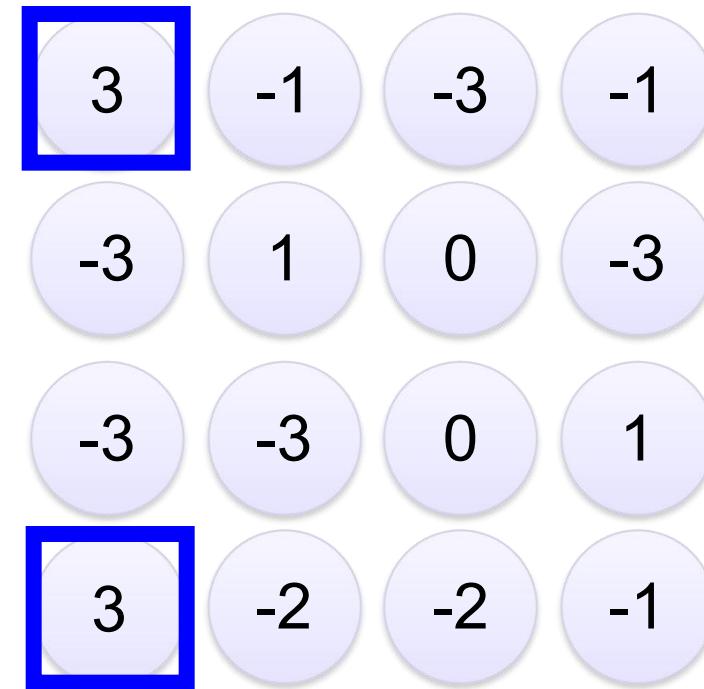
stride=1



6 x 6 image



Filter 1



Convolution

stride=1

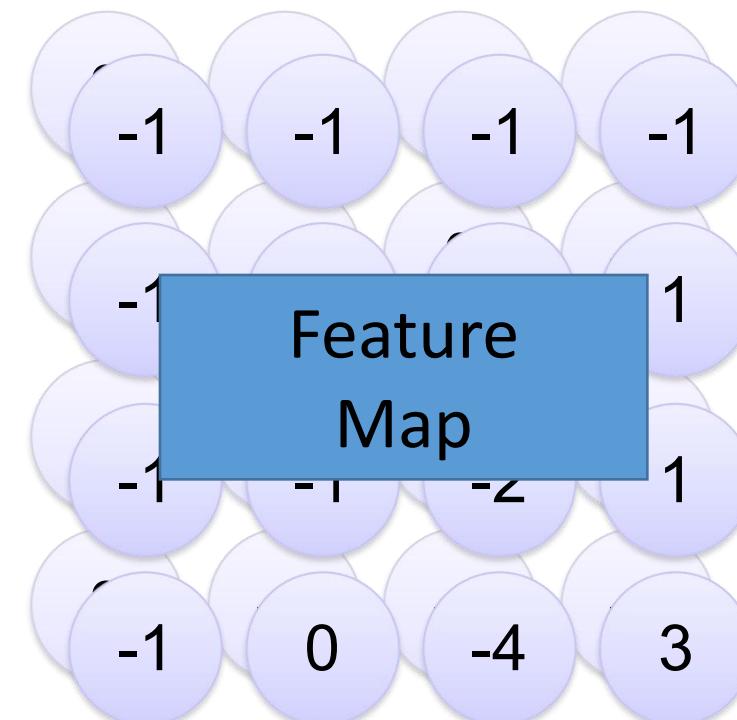
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

-1	1	-1
-1	1	-1
-1	1	-1

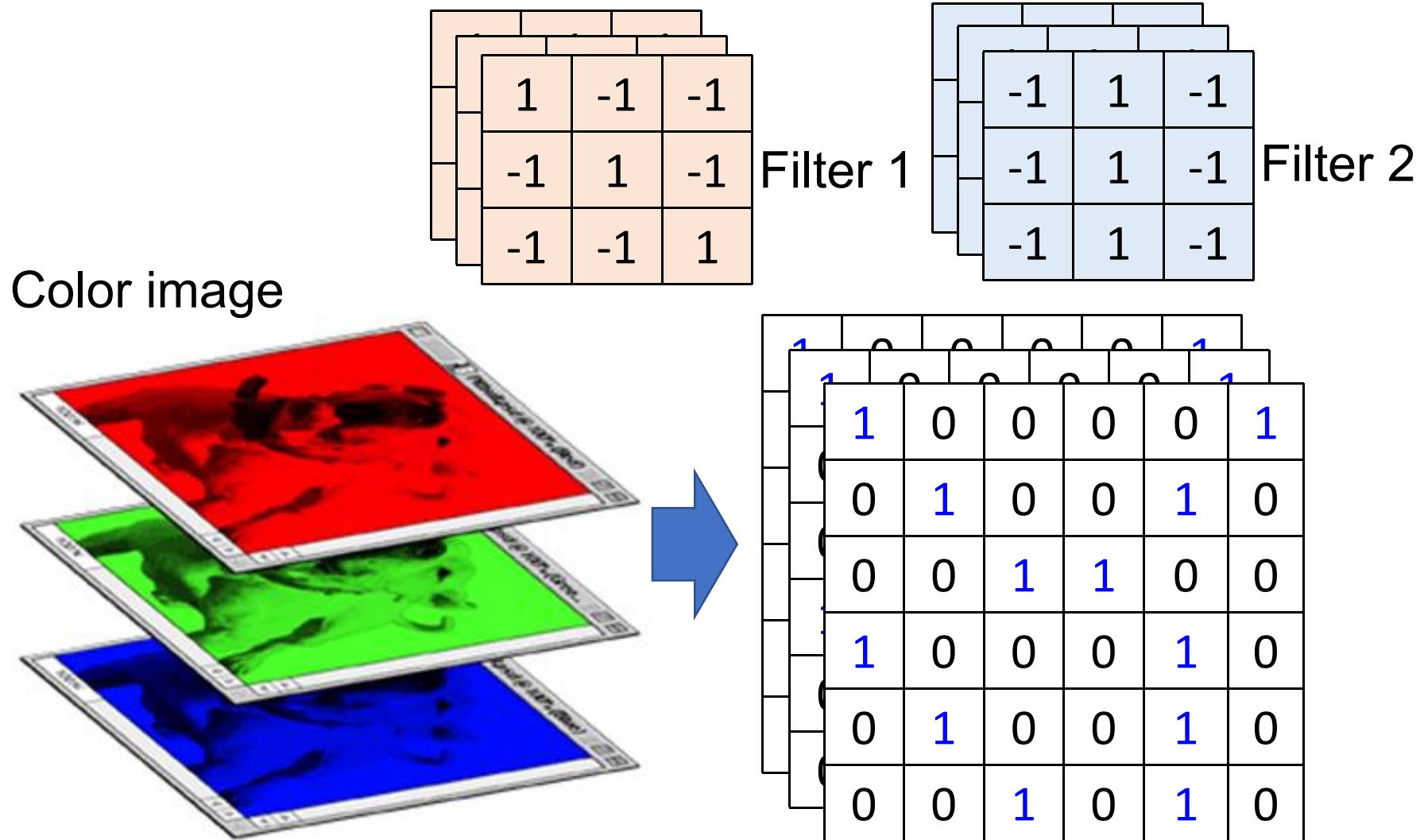
Filter 2

Repeat this for each filter

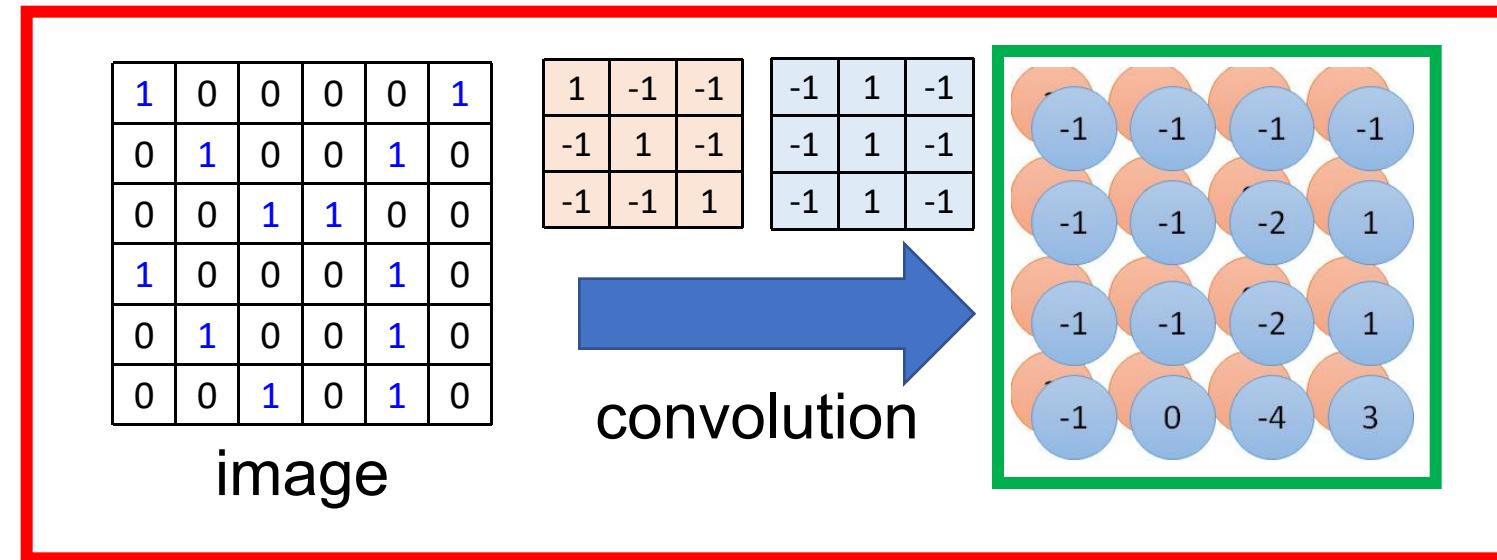


Two 4 x 4 images
Forming 2 x 4 x 4 matrix

Color image: RGB 3 channels

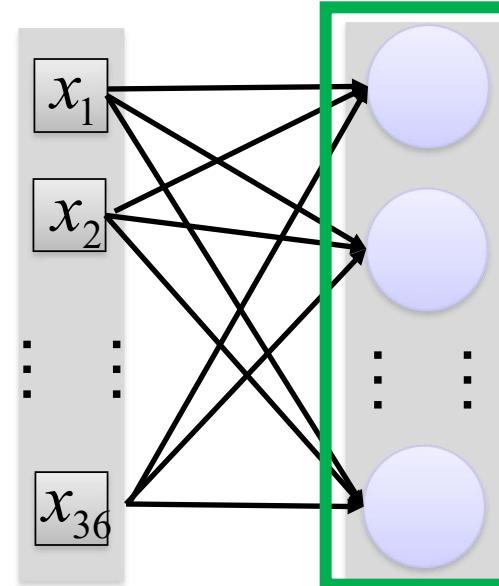


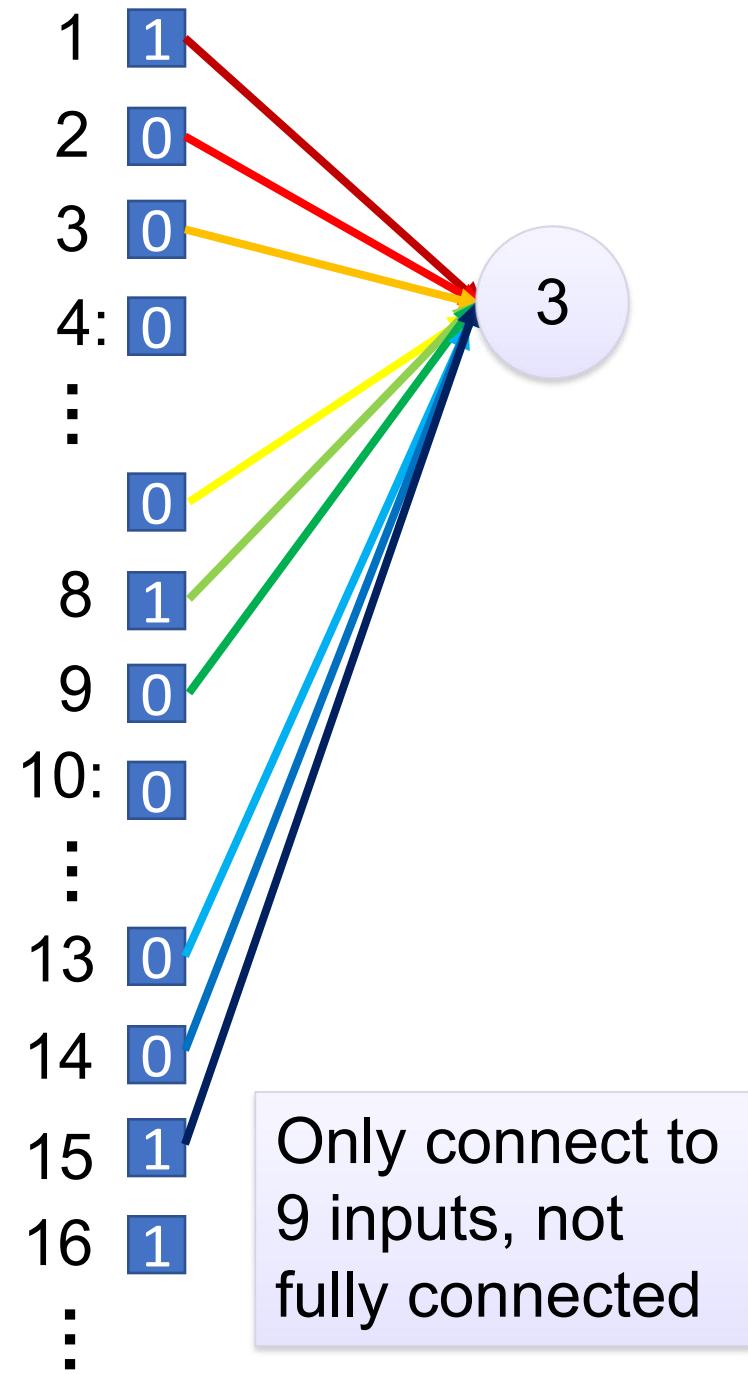
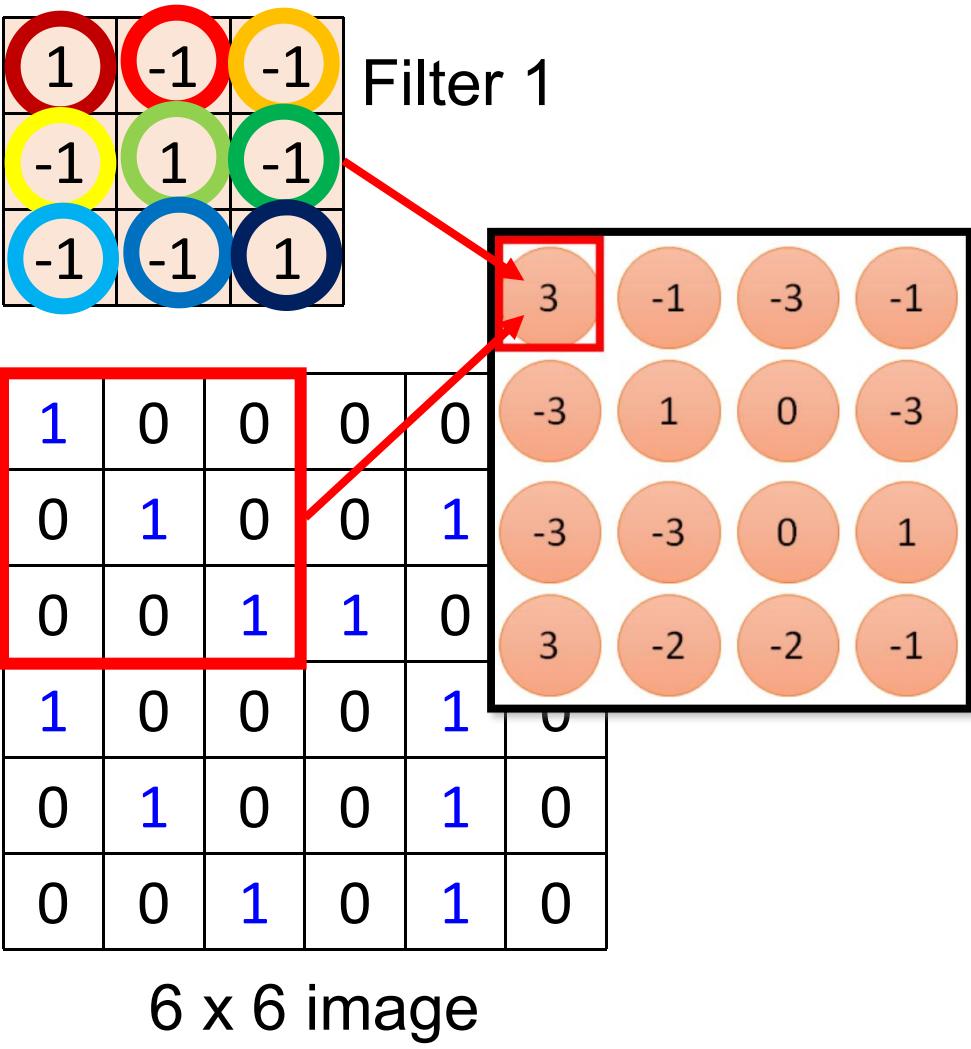
Convolution v.s. Fully Connected

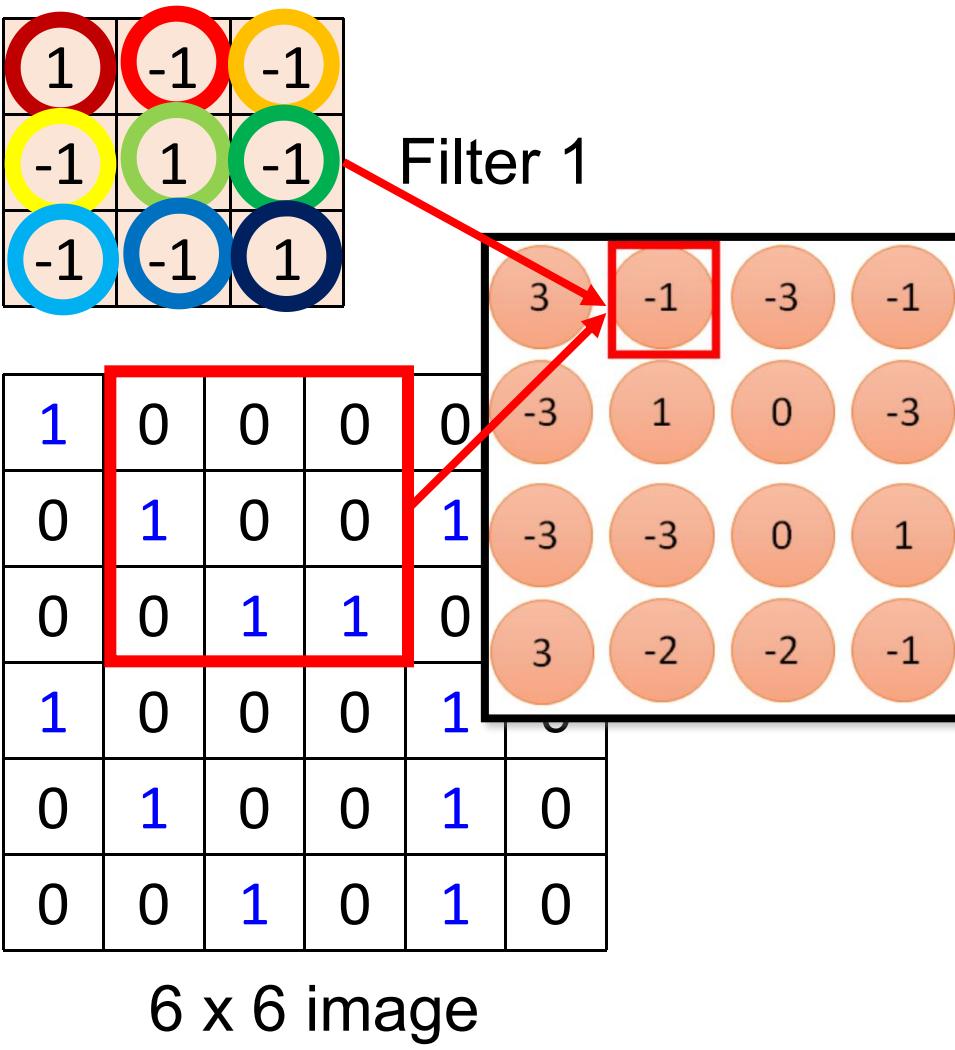


Fully-
connected

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

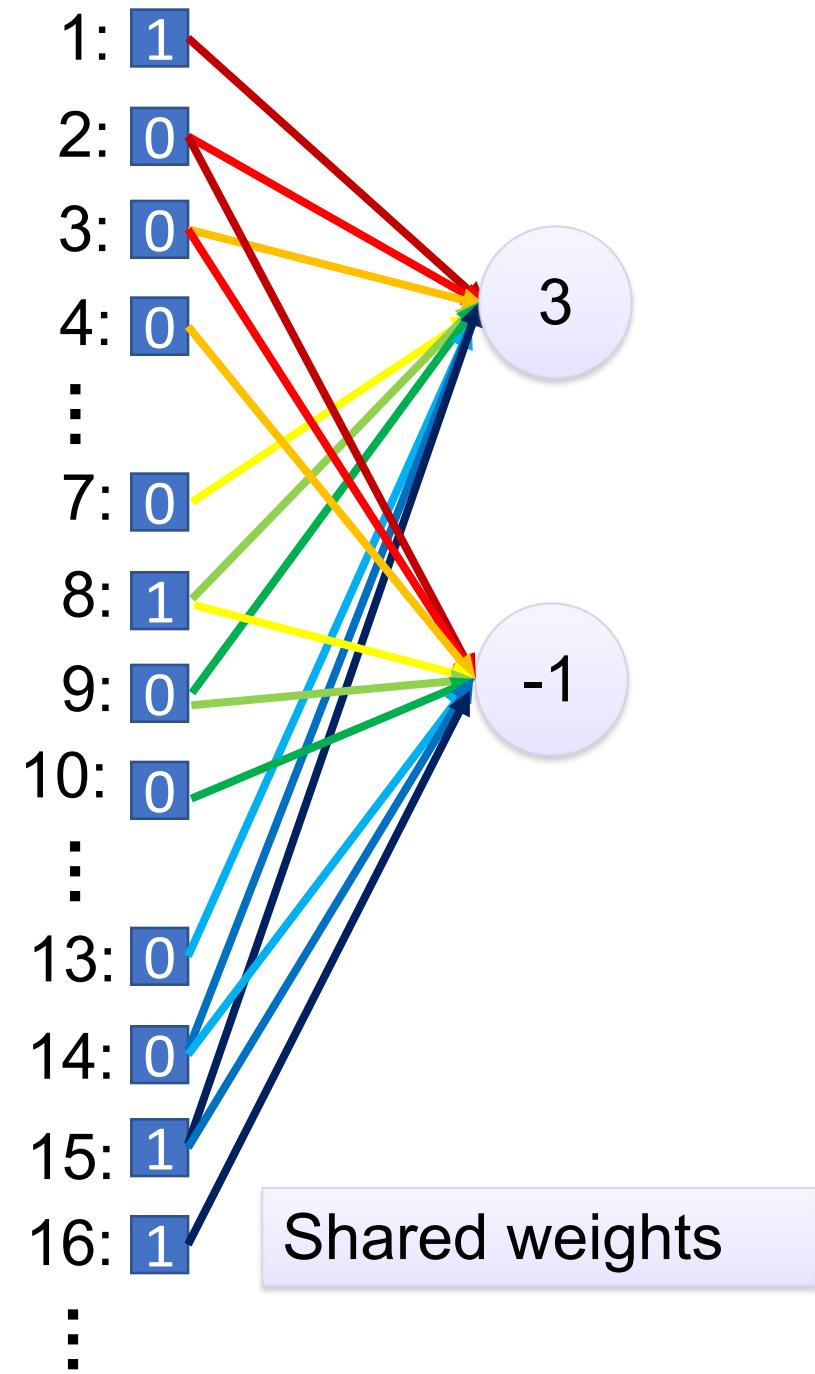






Fewer parameters

Even fewer parameters



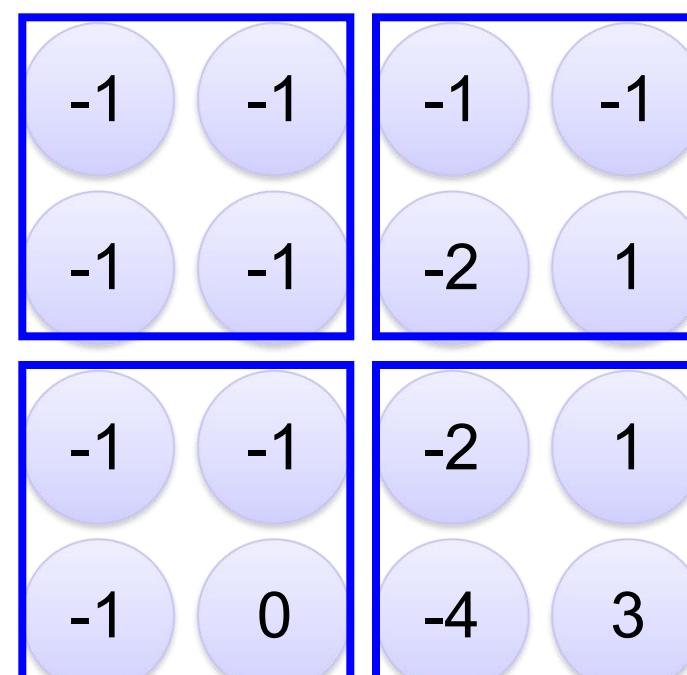
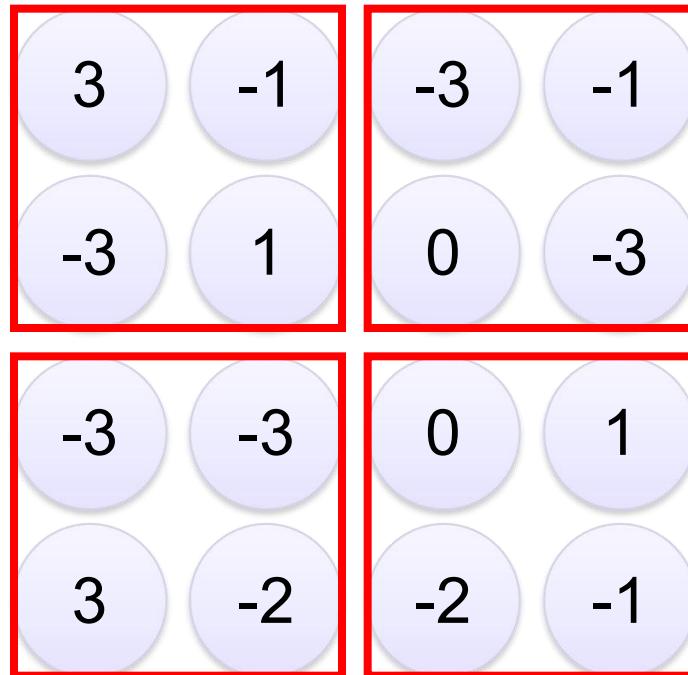
Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



Why Pooling

- Subsampling pixels will not change the object

bird



Subsampling

bird



We can subsample the pixels to make image smaller



fewer parameters to characterize the image

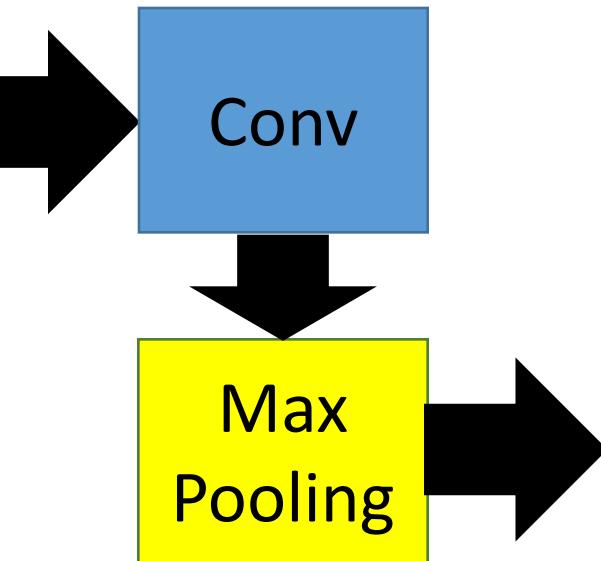
CNN as Fully Connected Network

- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

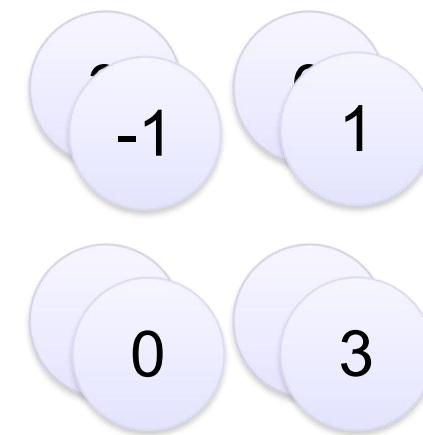
Max Pooling

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

6 x 6 image



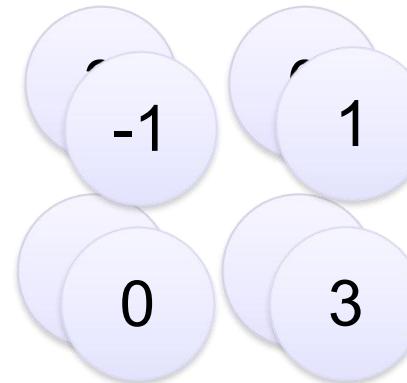
New image
but smaller



2 x 2 image

Each filter
is a channel

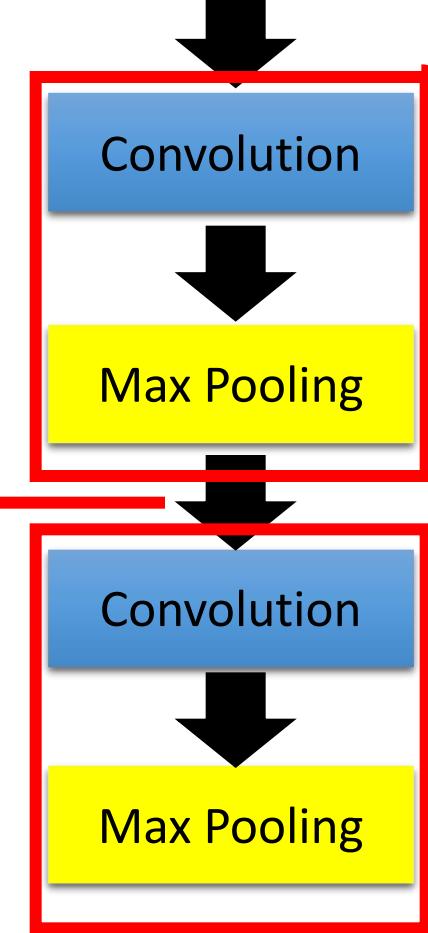
The Whole CNN



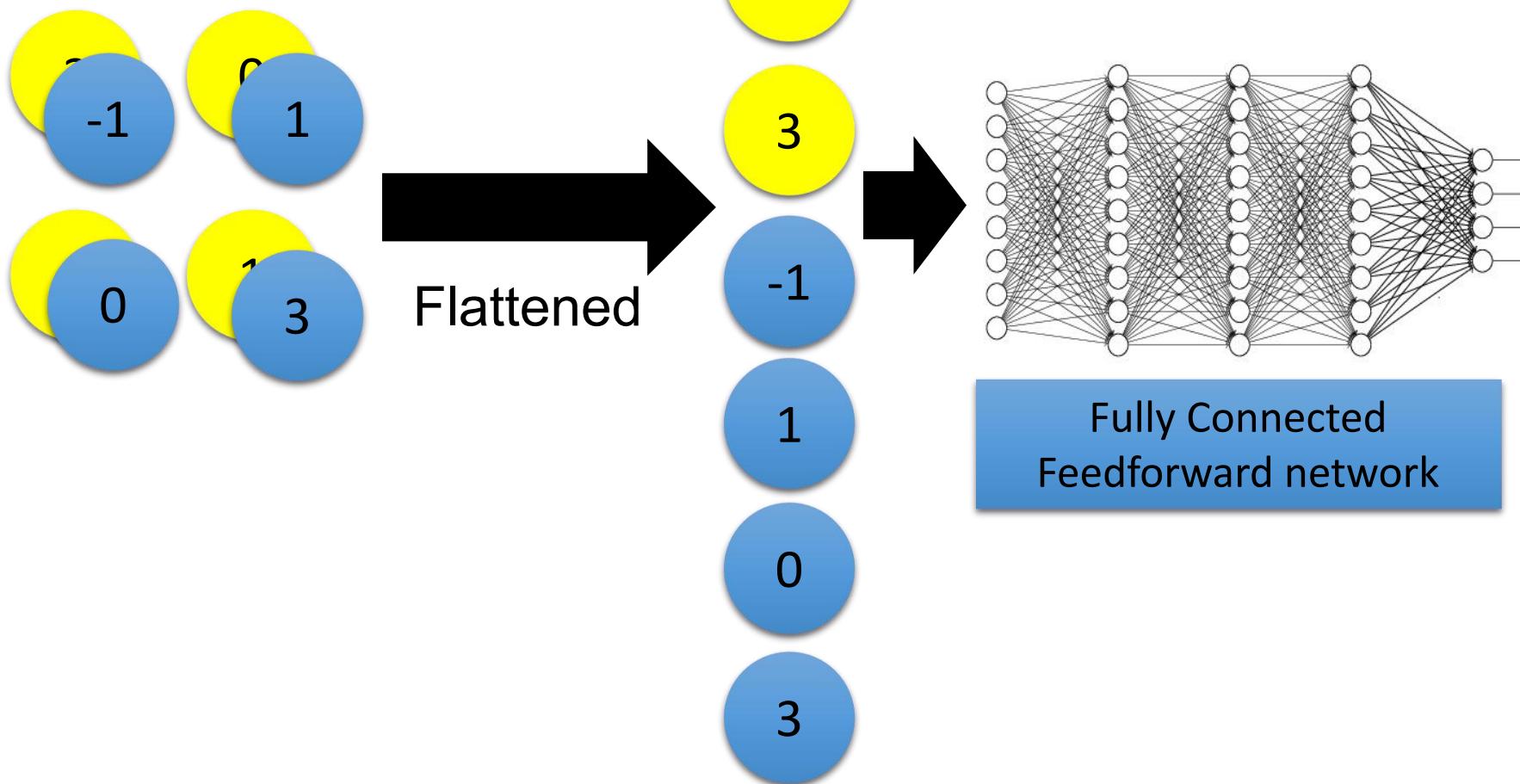
A new image

Smaller than the original image

The number of channels is the number of filters

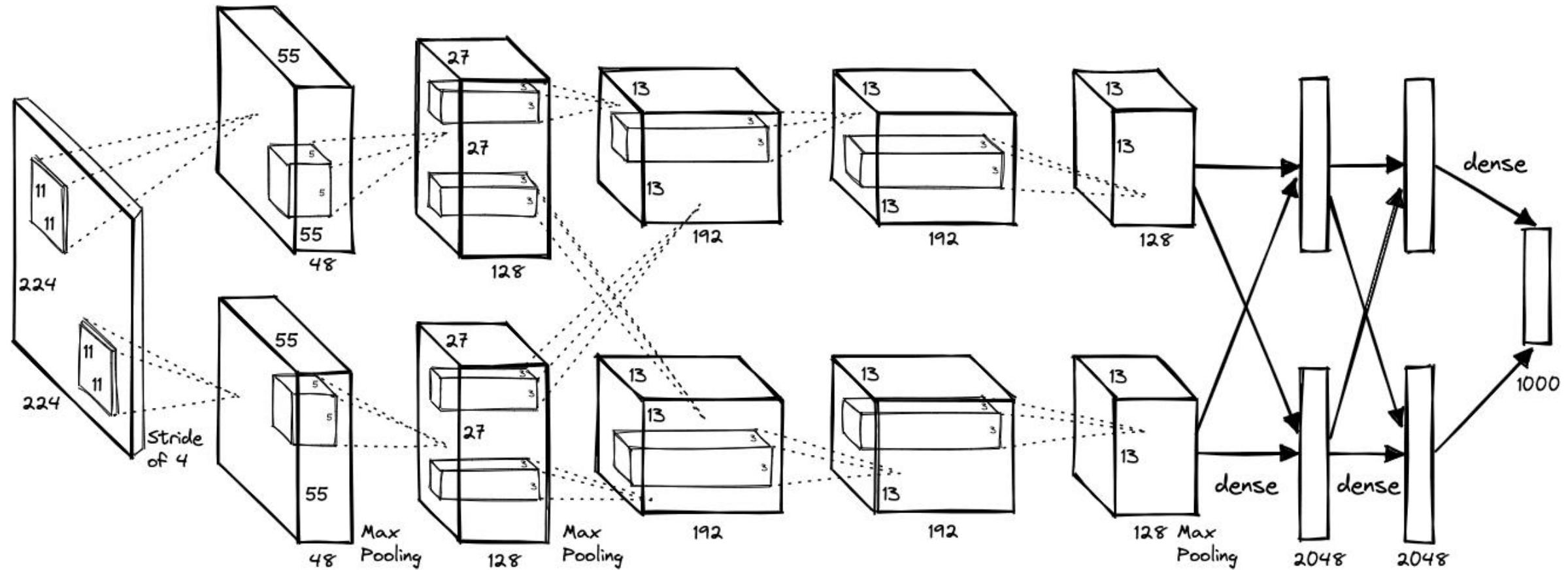


Flattening



ImageNet Classification with Deep Convolutional Neural Networks

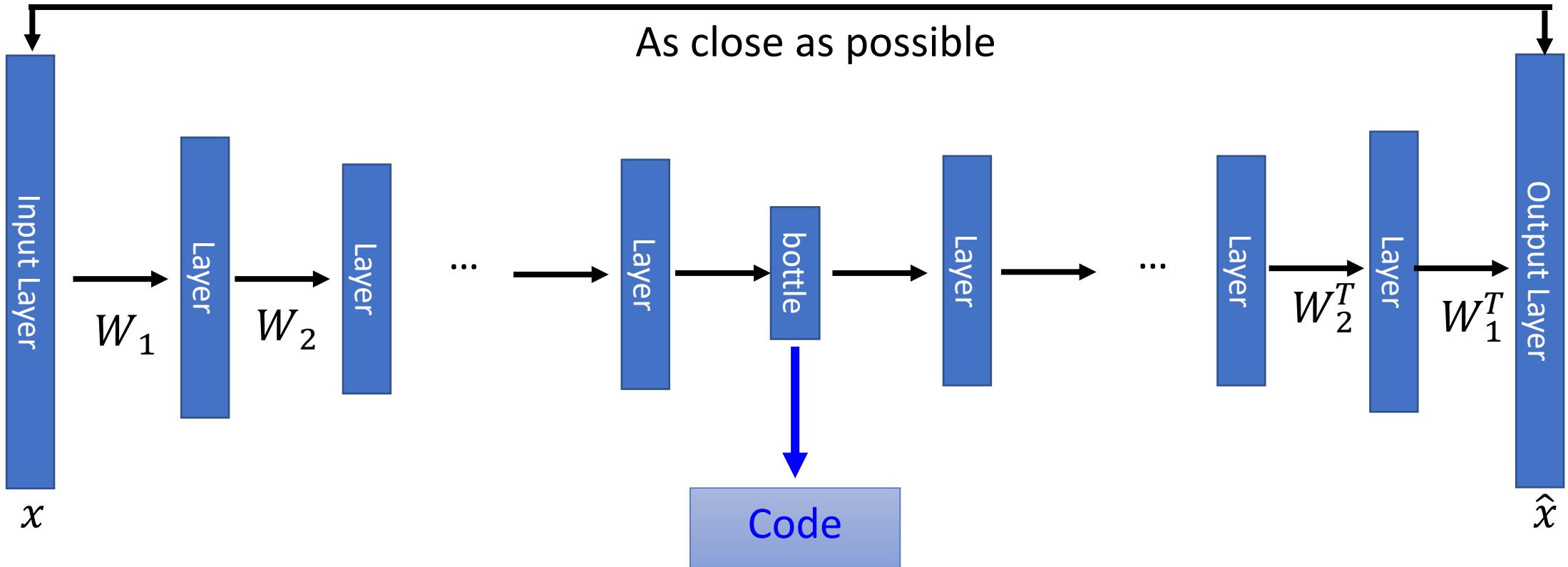
[Krizhevsky, Sutskever, Hinton, 2012]



AlexNet

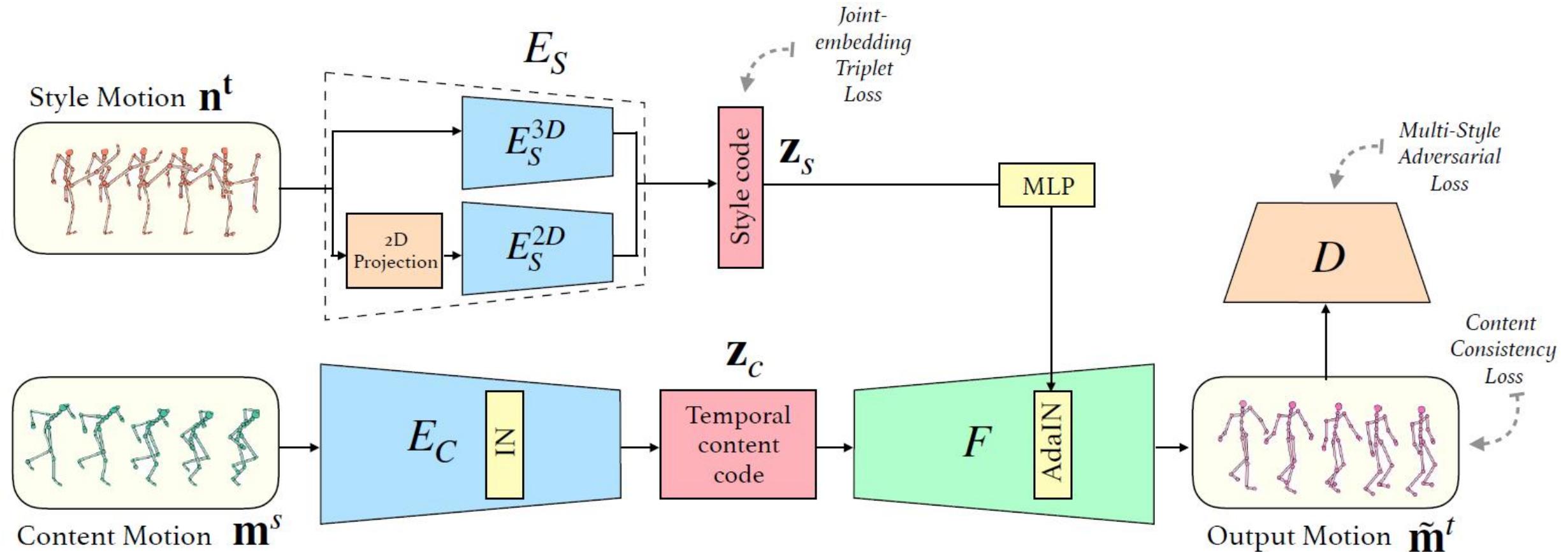
Reducing the dimensionality of data with neural networks.

[Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. Science, 2006]



Unpaired Motion Style Transfer from Video to Animation

[Kfir, SIGGRAPH 2020]

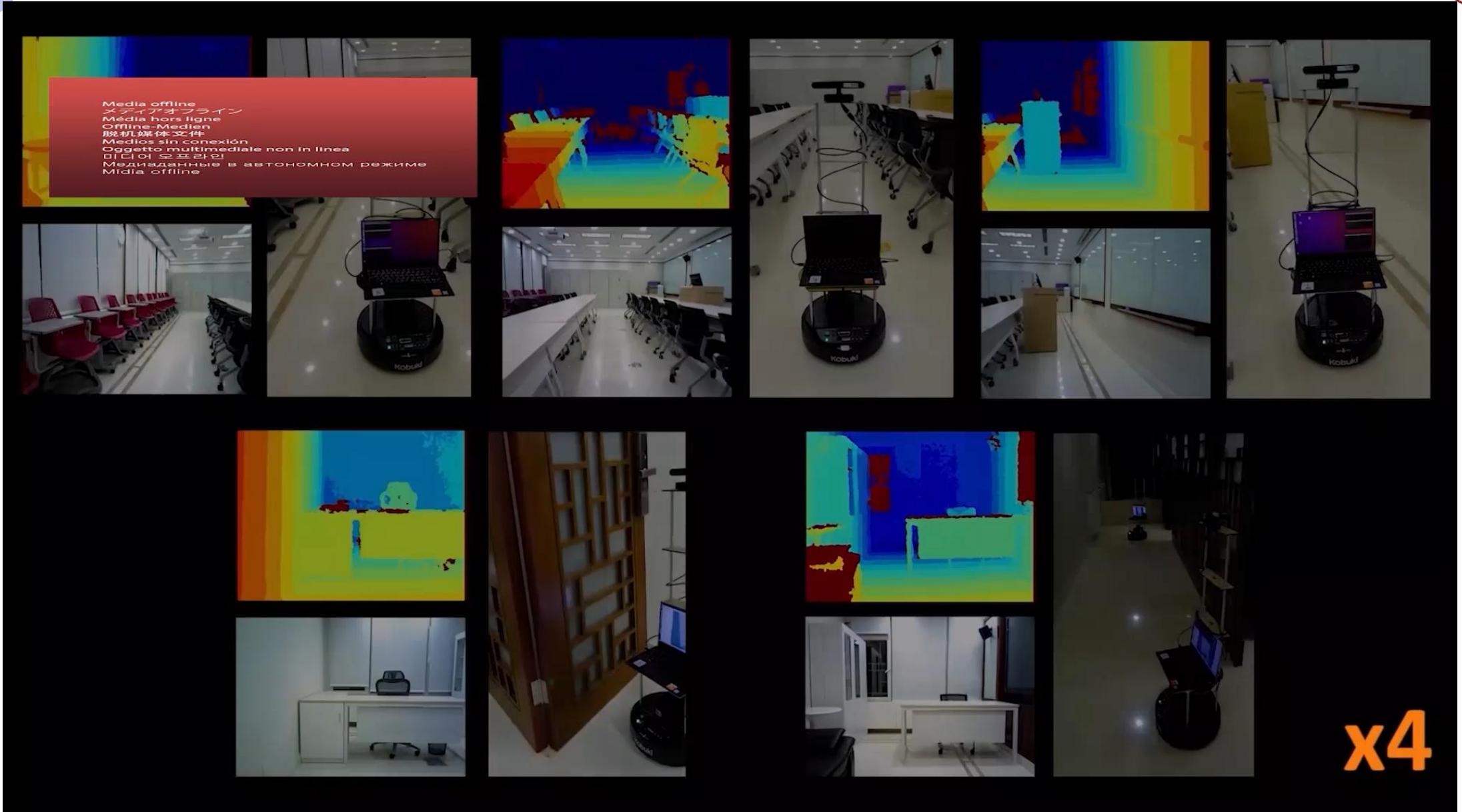




Deep Learning Applications



Deep Learning Applications





Deep Learning for NLP Tasks

1-of-N Encoding

apple = [1 0 0 0 0]

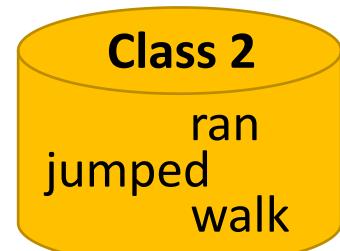
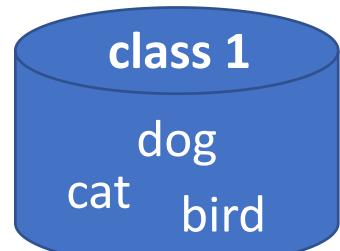
bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

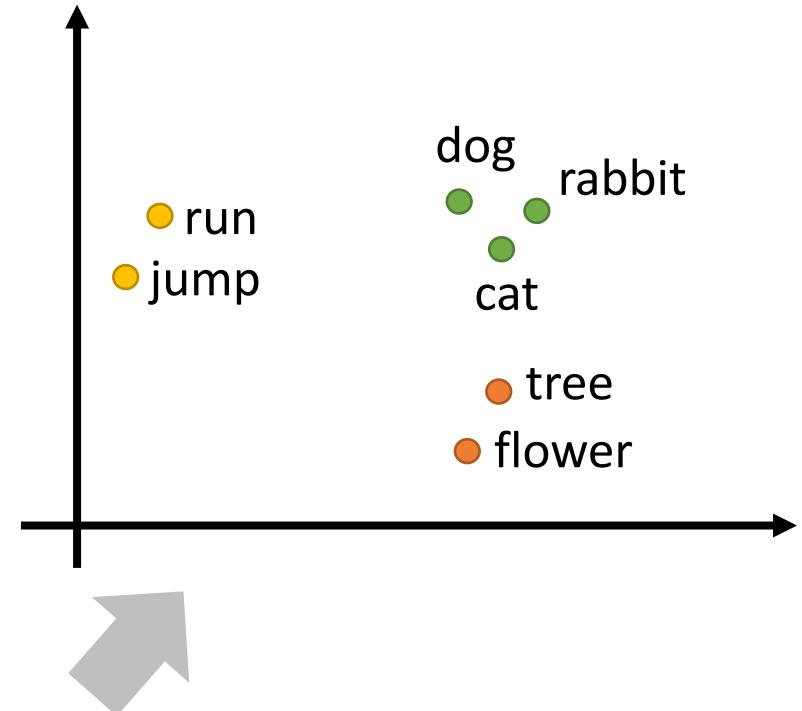
dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word Class



Word Embedding





Deep Learning for NLP Tasks

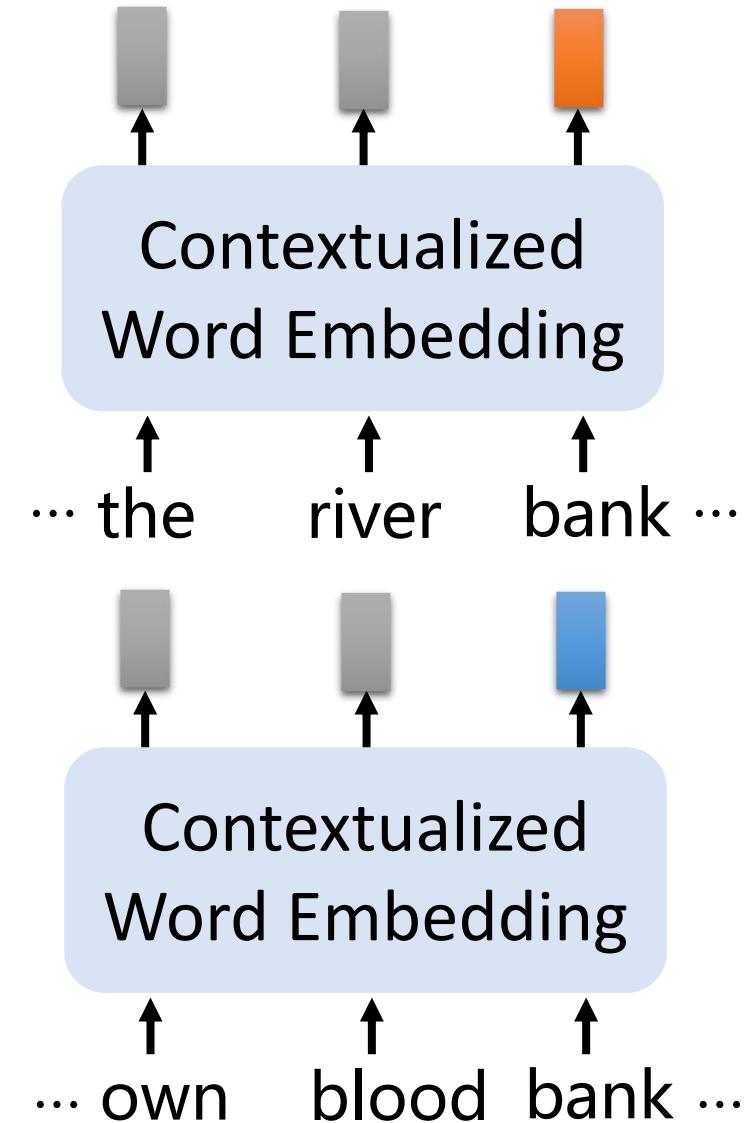
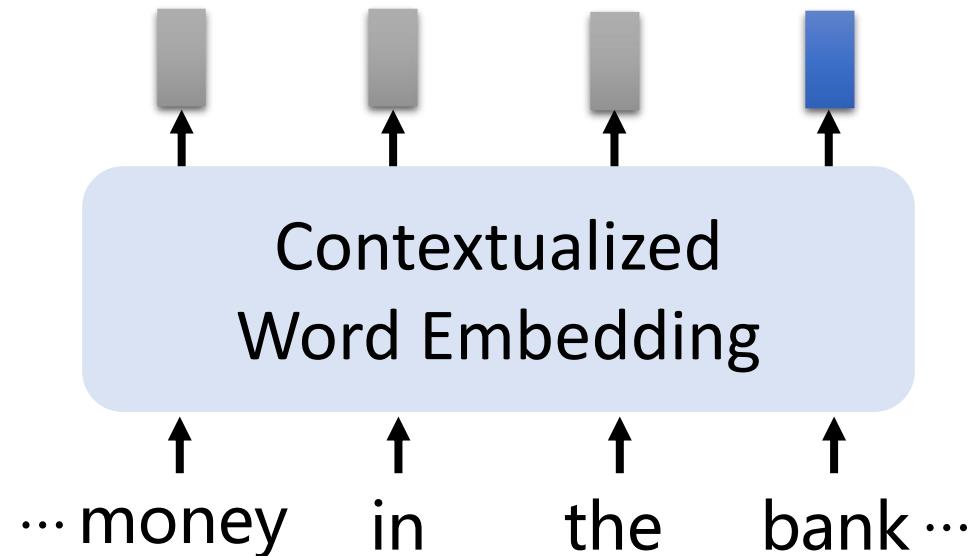


- Have you paid that money to the bank yet ?
- It is safest to deposit your money in the bank .
- The victim was found lying dead on the river bank .
- They stood on the river bank to fish.
- The hospital has its own blood bank.
- The third sense or not?



Deep Learning for NLP Tasks

- Each word token has its own embedding, even though it has the same word type
- The embeddings of word tokens also depend on its context

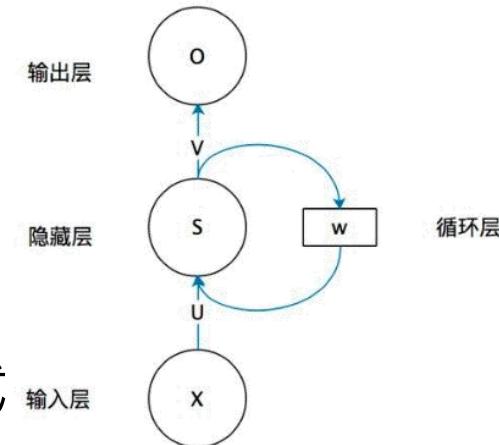
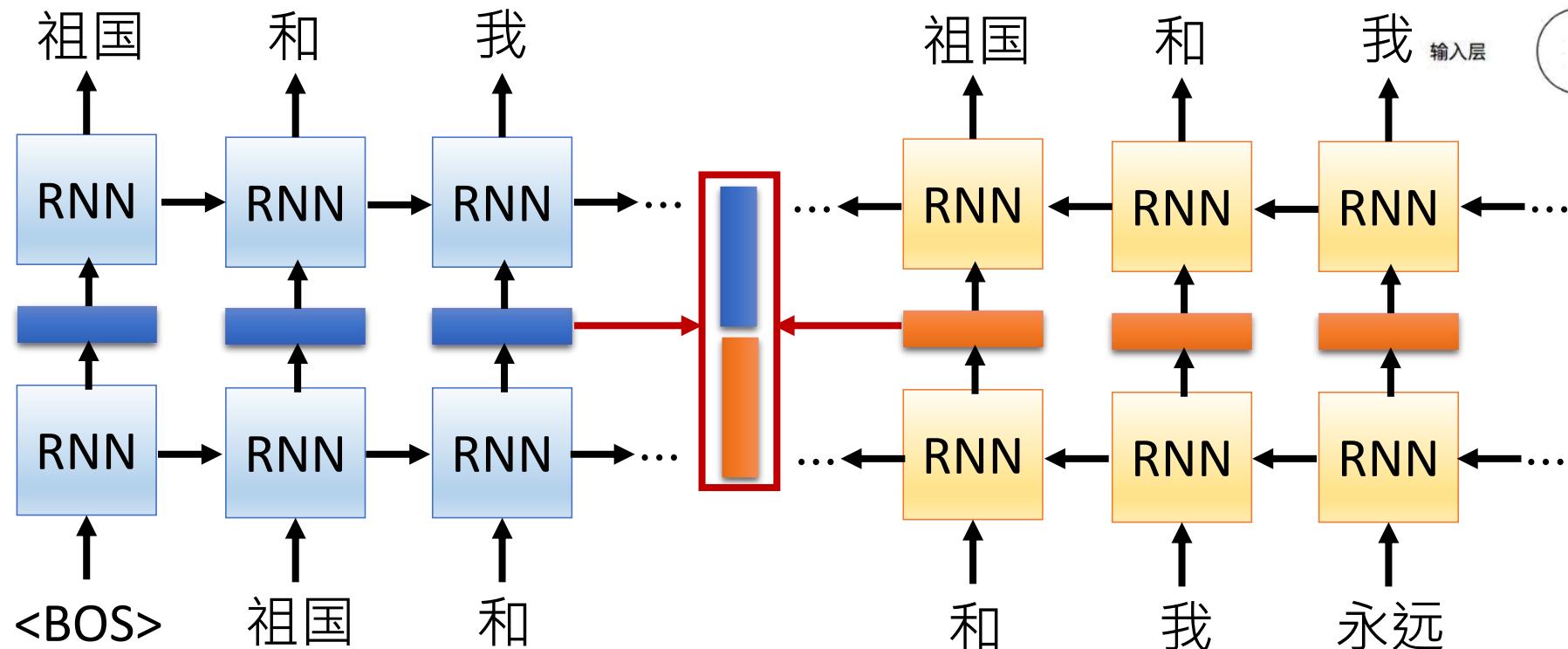


Embeddings from Language Model (ELMO)



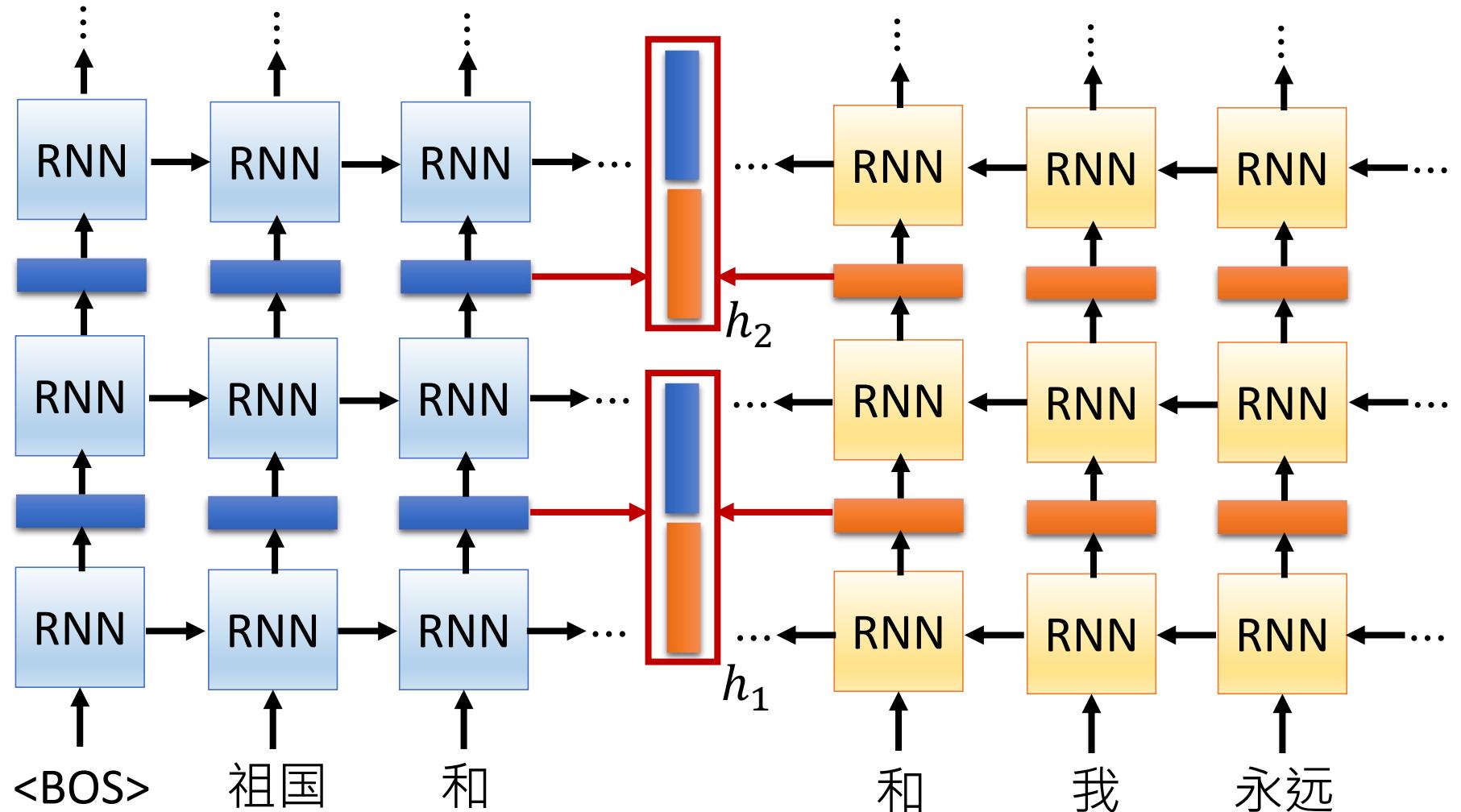
RNN-based language models (trained from lots of sentences)

e.g. given “祖国 和 我 永远 在一起”



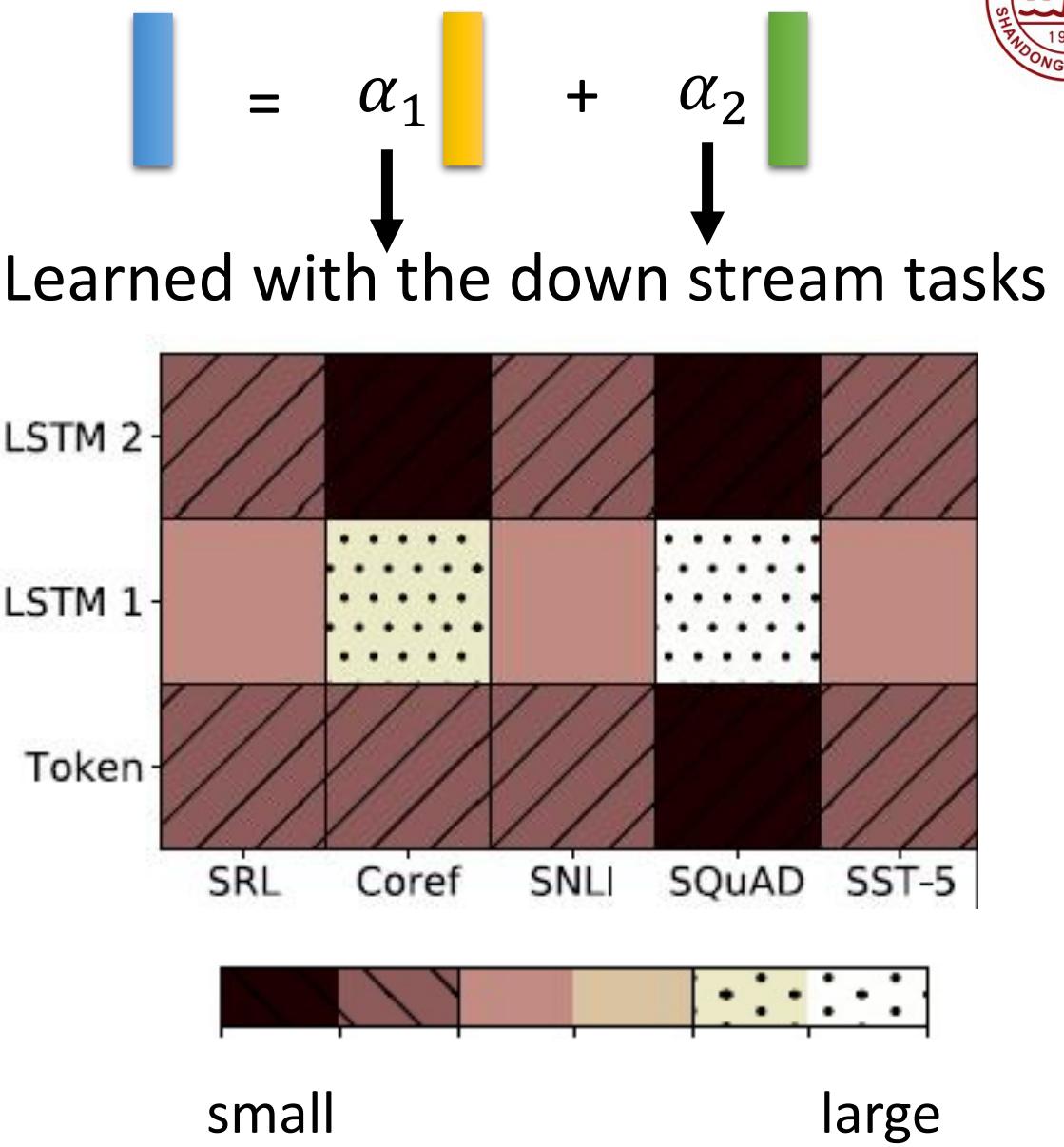
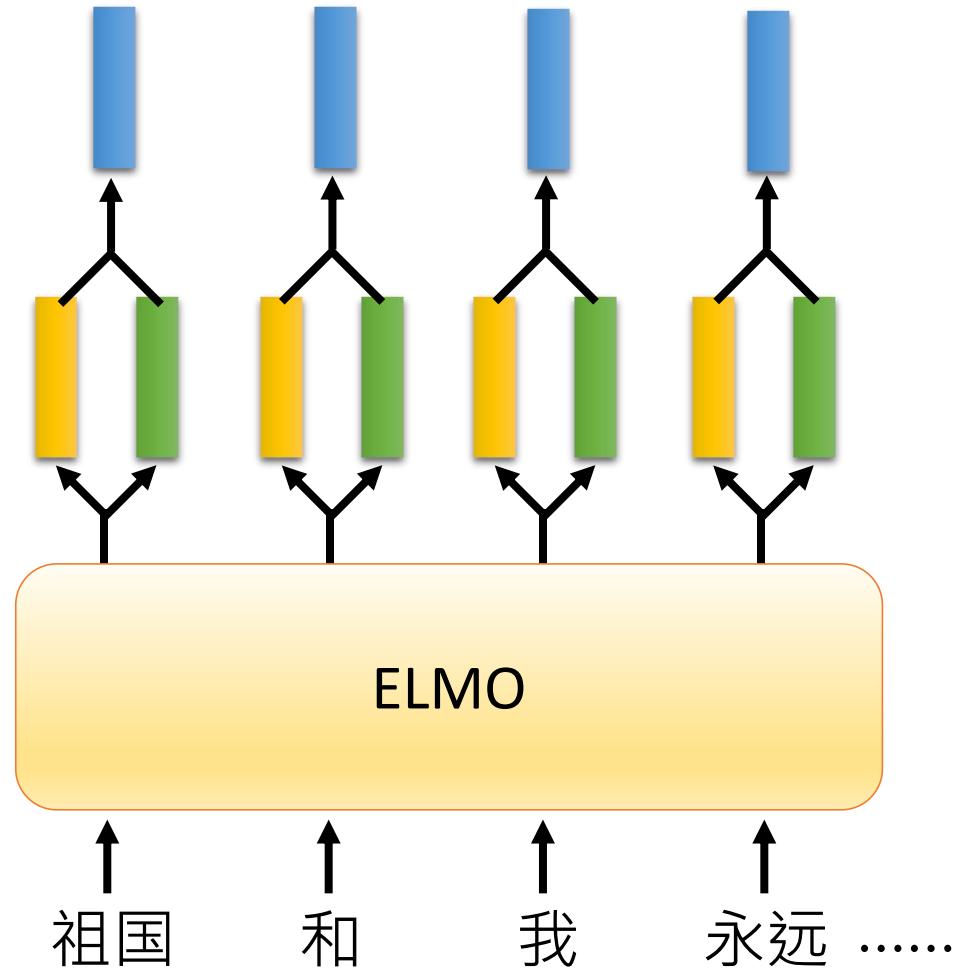
Each layer in deep LSTM can generate a latent representation.

Which one should we use???





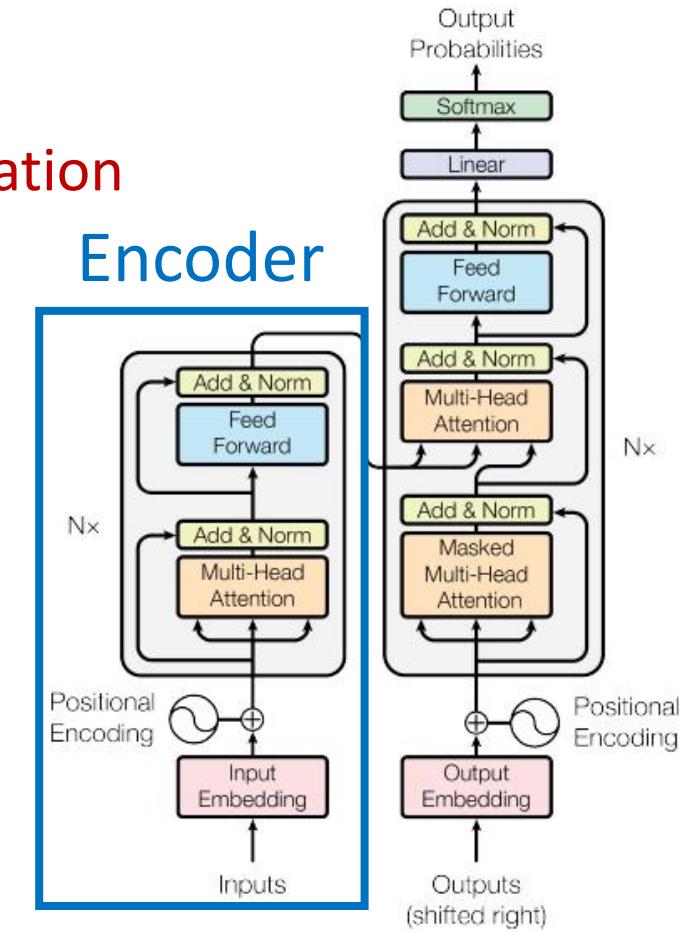
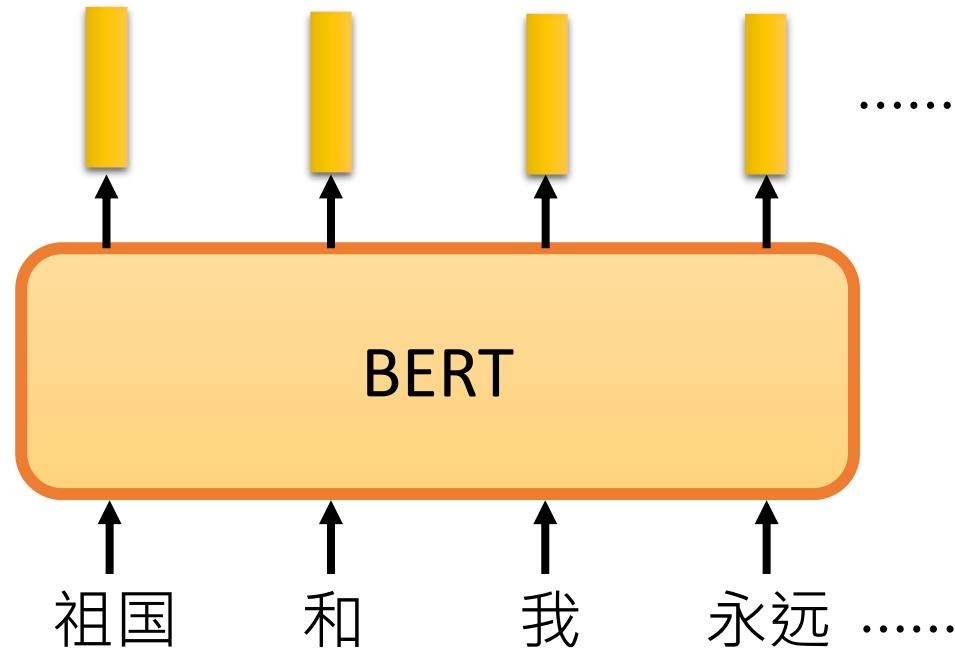
ELMO



Bidirectional Encoder Representations from Transformers

BERT = Encoder of Transformer

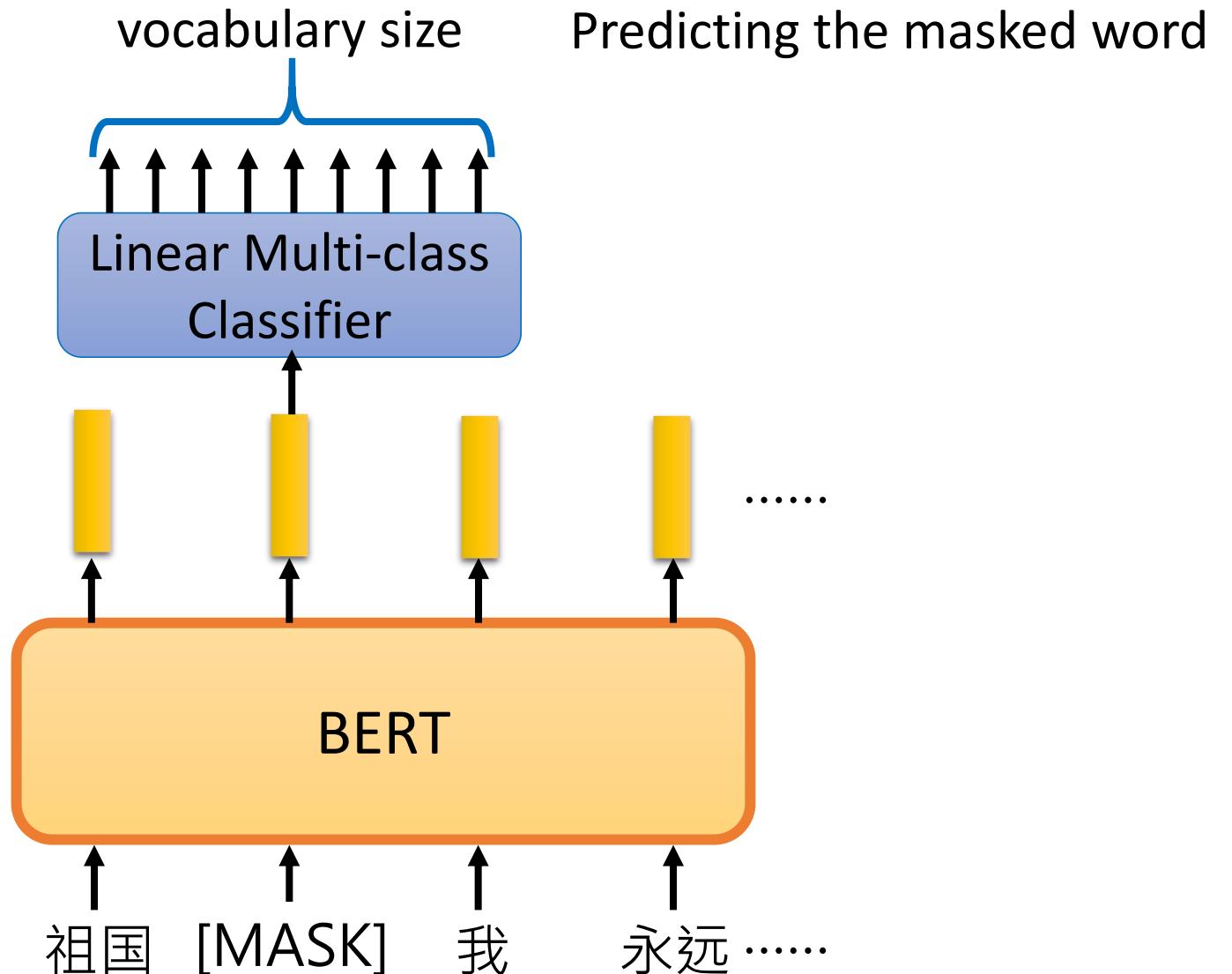
Learned from a large amount of text without annotation





Training of BERT

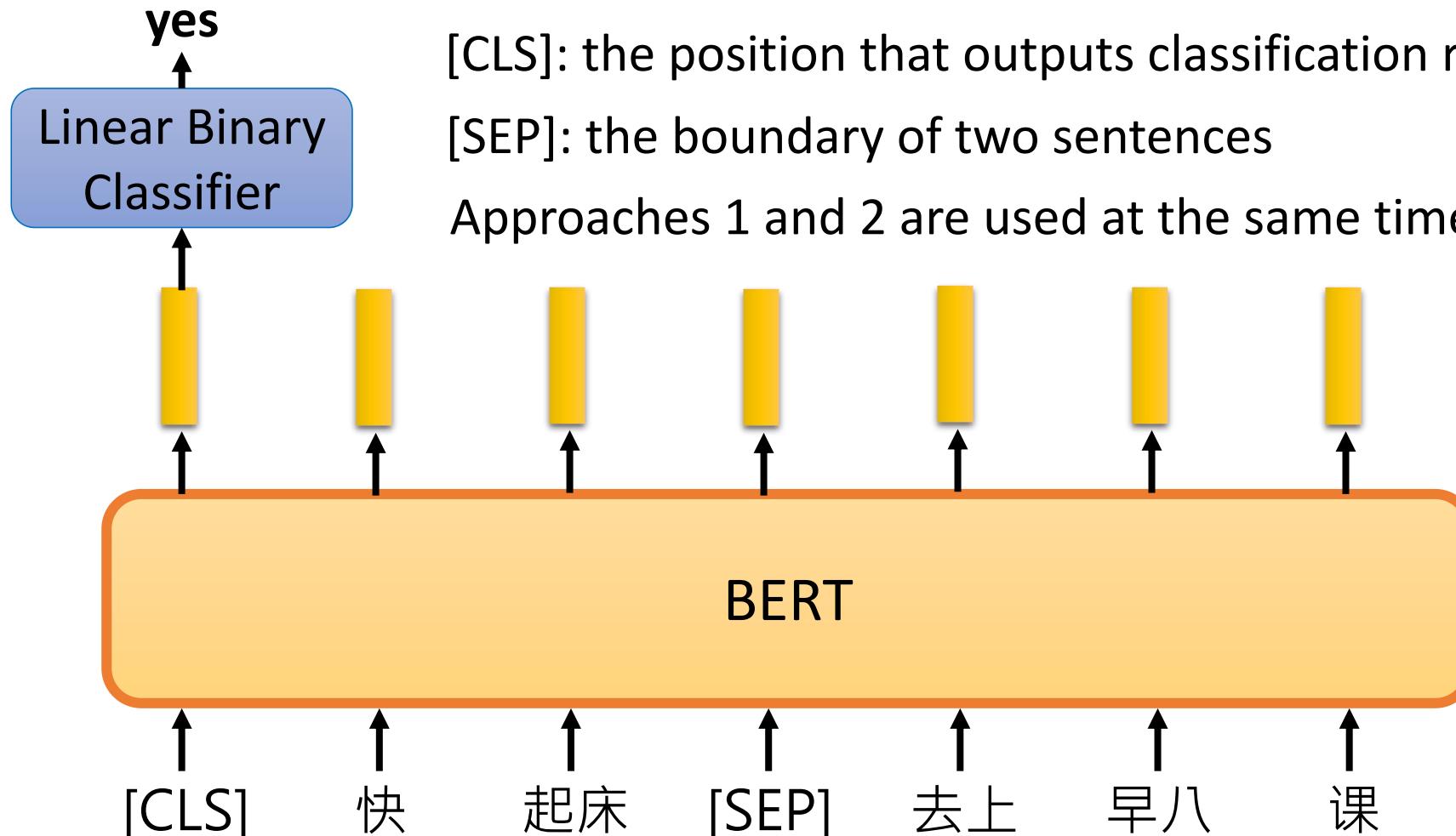
Approach 1: Masked LM





Training of BERT

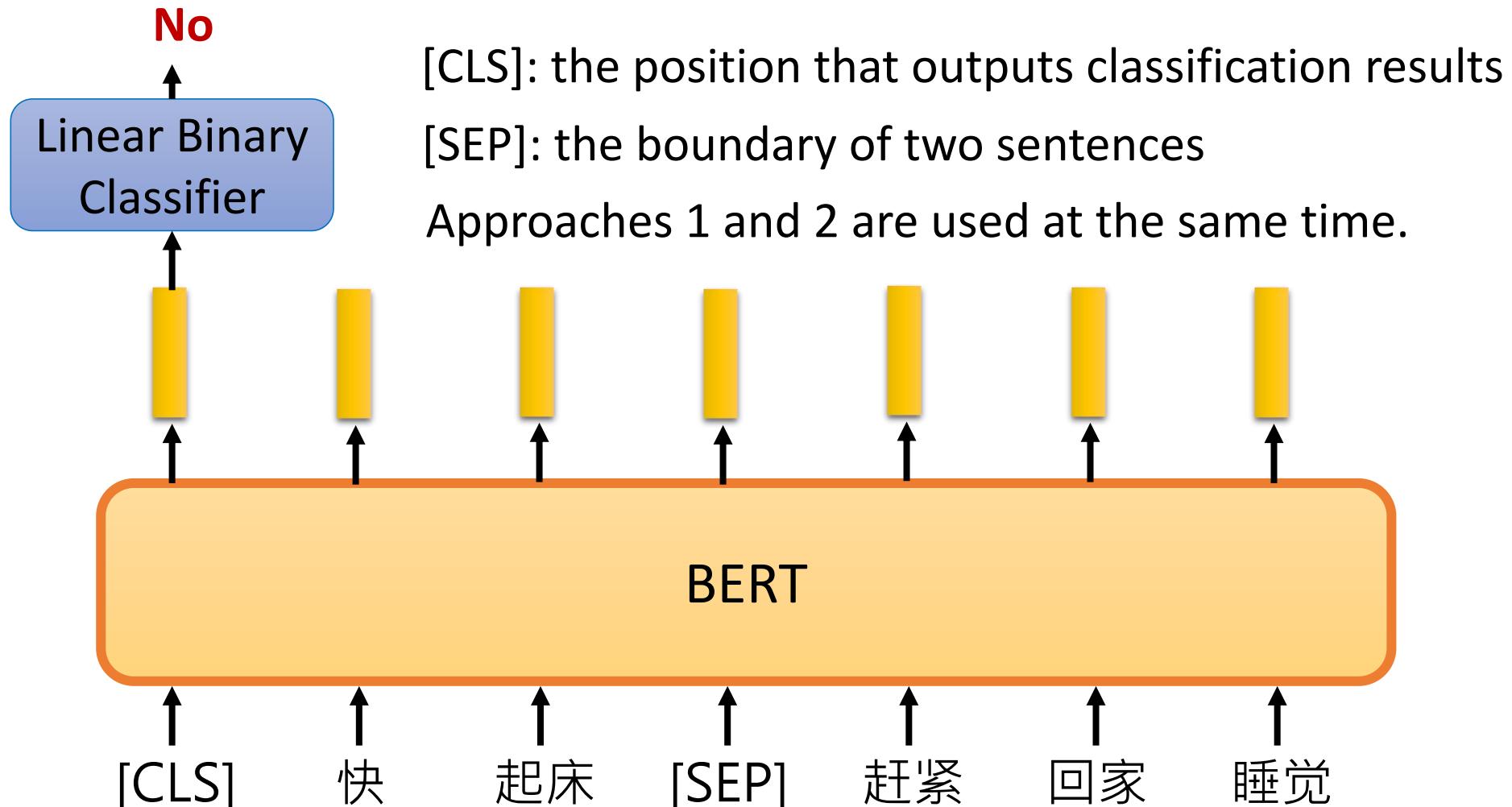
Approach 2: Next Sentence Prediction





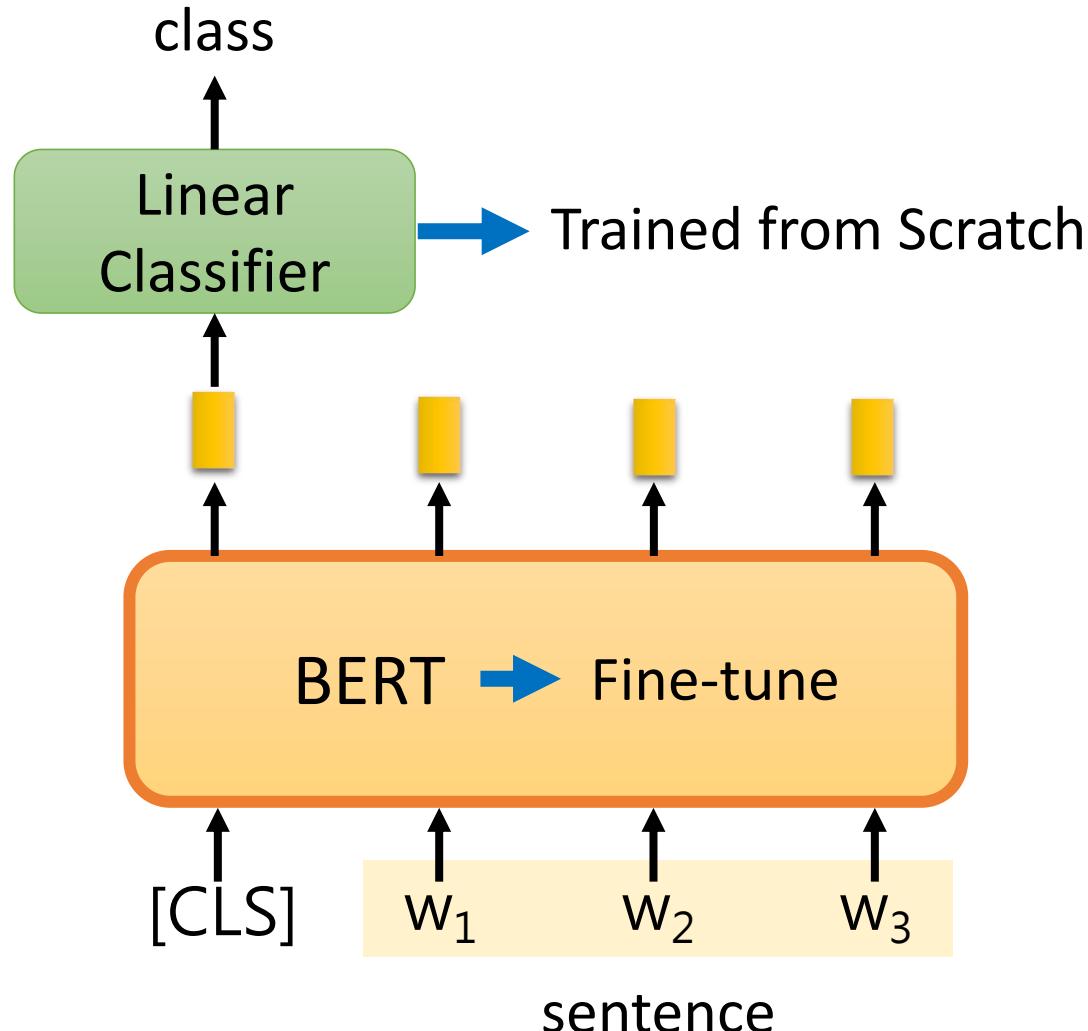
Training of BERT

Approach 2: Next Sentence Prediction





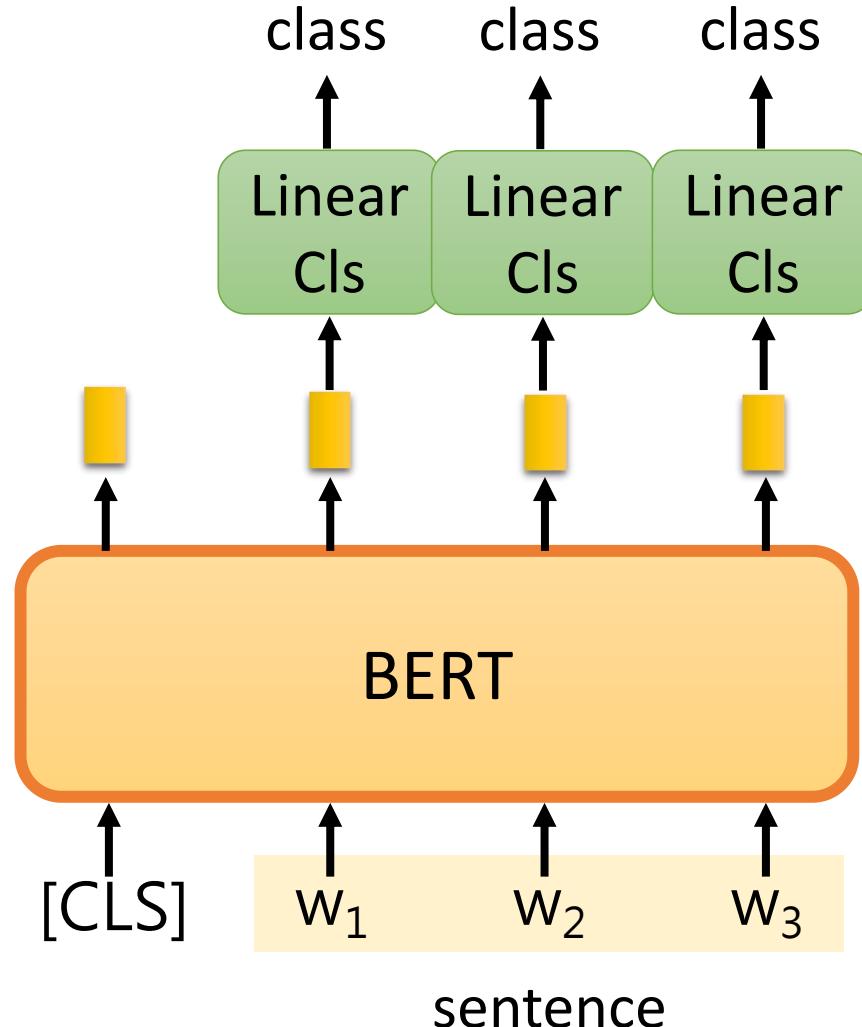
How to use BERT – Case 1



Input: single sentence,
output: class

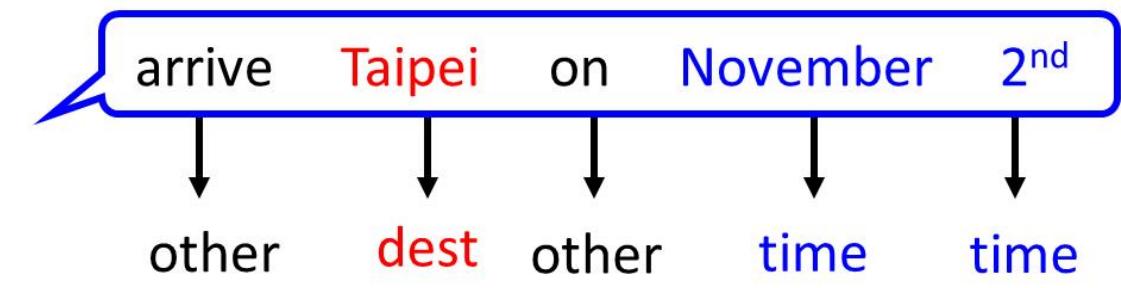
Example:
Sentiment analysis (our HW),
Document Classification

How to use BERT – Case 2



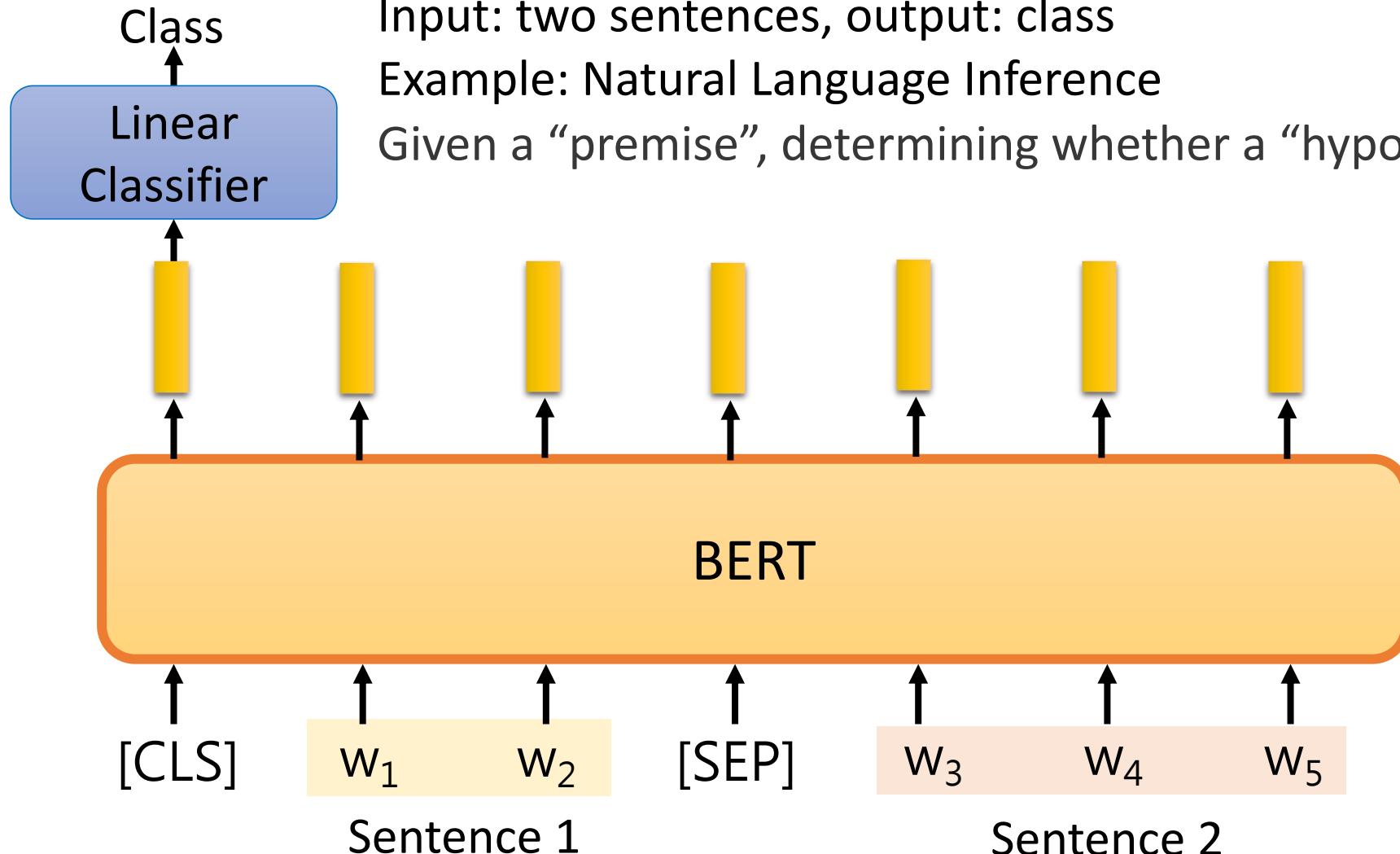
Input: single sentence,
output: class of each word

Example: Slot filling





How to use BERT – Case 3



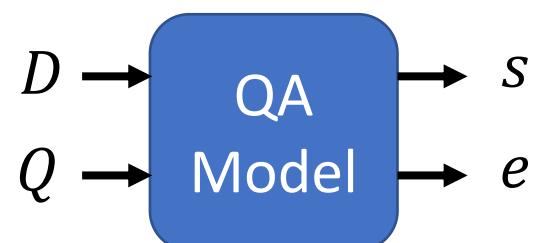


How to use BERT – Case 4

Extraction-based Question Answering (QA)
(E.g. SQuAD)

Document: $D = \{d_1, d_2, \dots, d_N\}$

Query: $Q = \{q_1, q_2, \dots, q_N\}$



output: two integers (s, e)

Answer: $A = \{q_s, \dots, q_e\}$

In meteorology, precipitation is any product of the condensation of 17 atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain are called "showers".

77

79

What causes precipitation to fall?

gravity

$s = 17, e = 17$

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupel

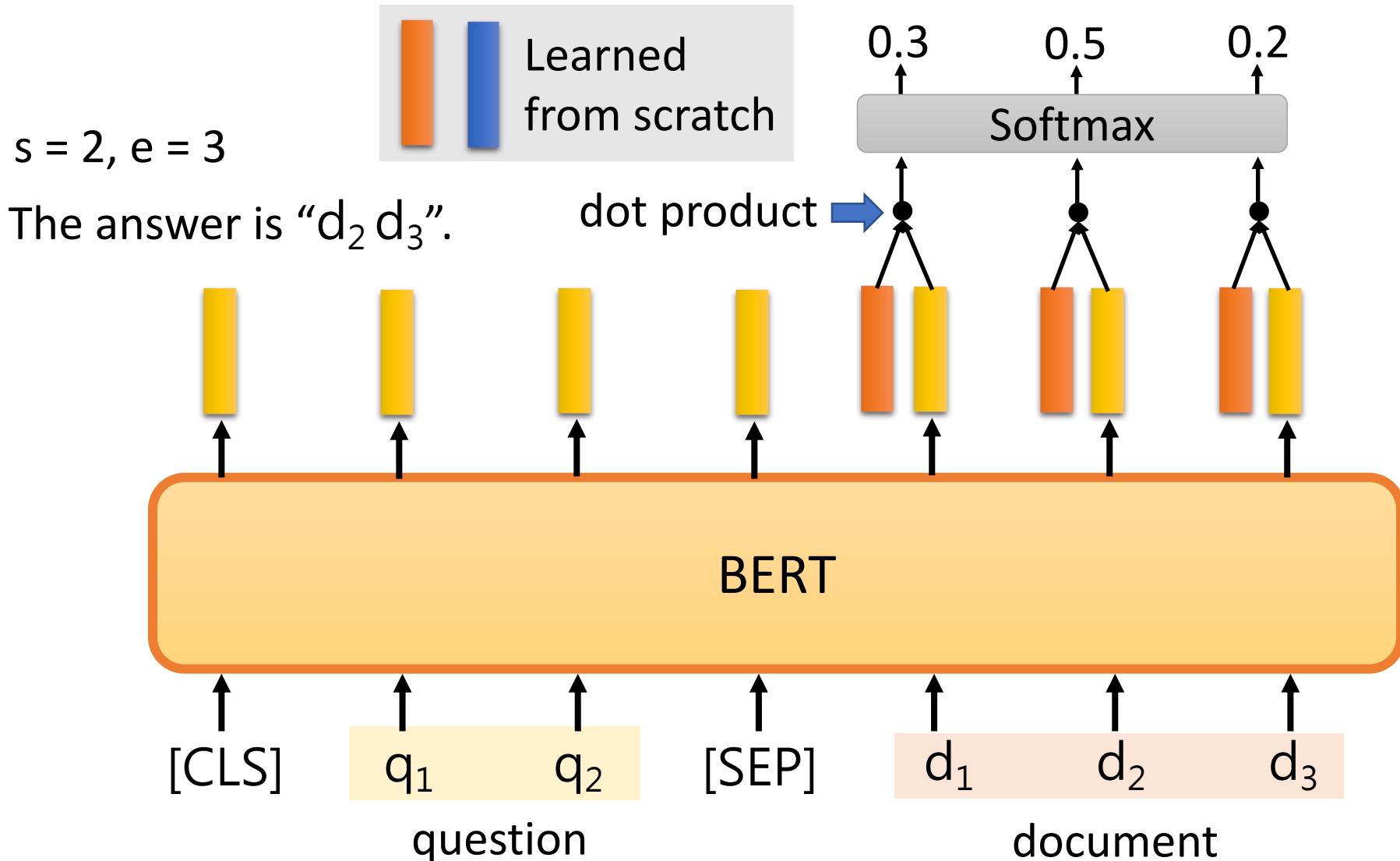
Where do water droplets collide with ice crystals to form precipitation?

within a cloud

$s = 77, e = 79$



How to use BERT – Case 4

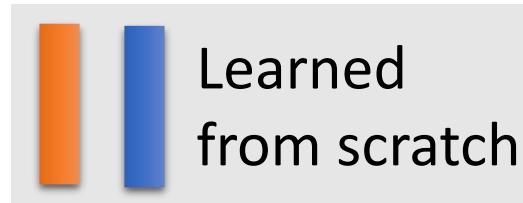




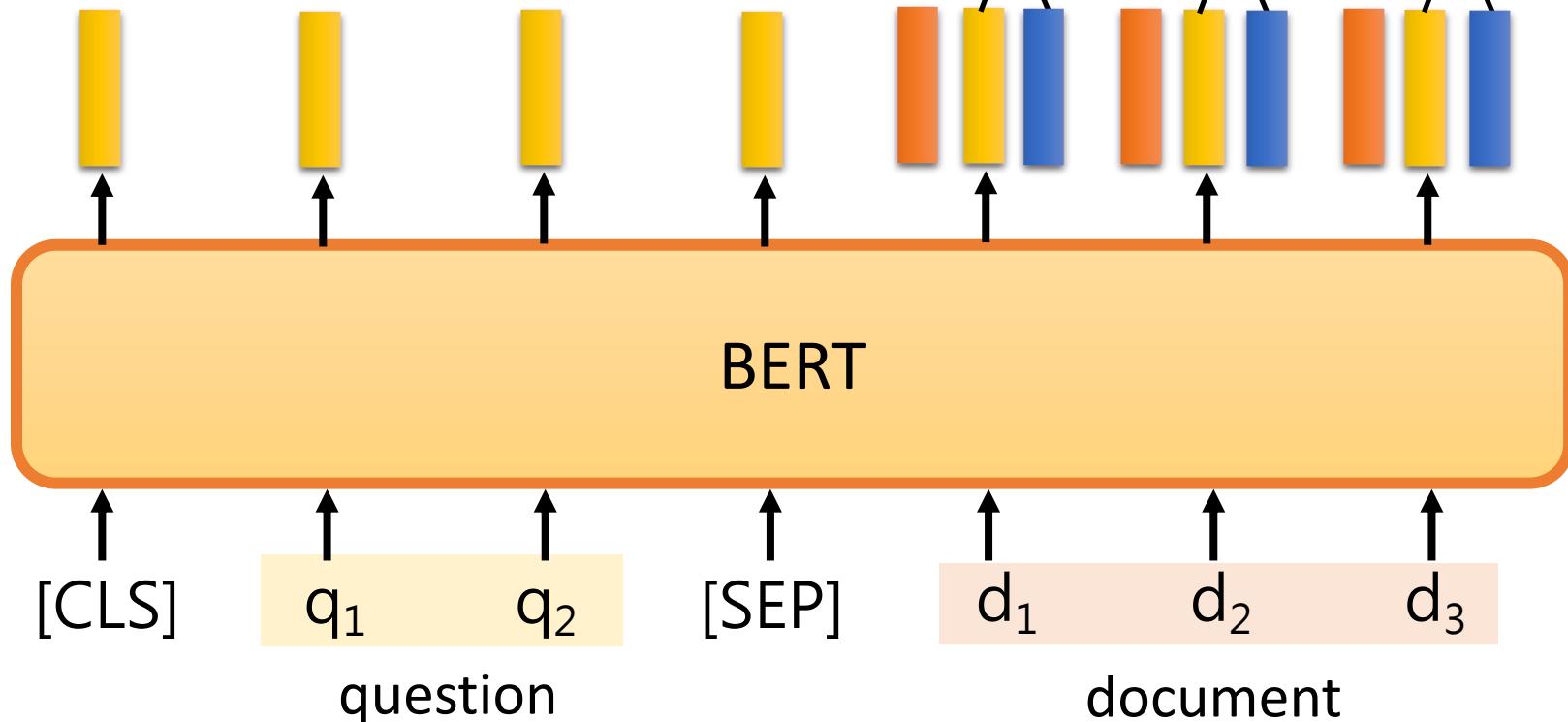
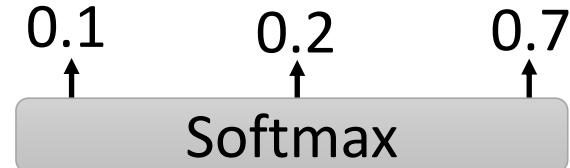
How to use BERT – Case 4

$s = 2, e = 3$

The answer is “ $d_2 d_3$ ”.

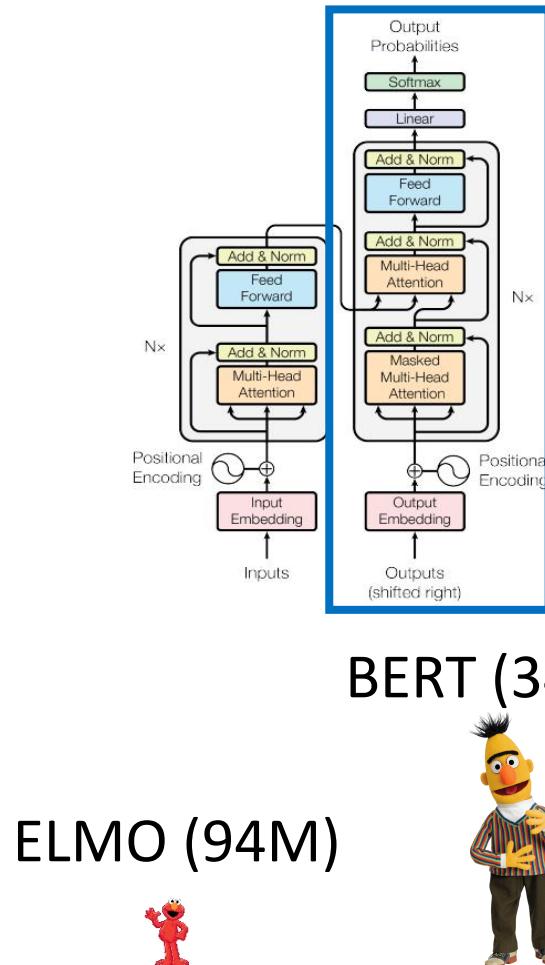


dot product →





Generative Pre-Training (GPT)



Transformer
Decoder

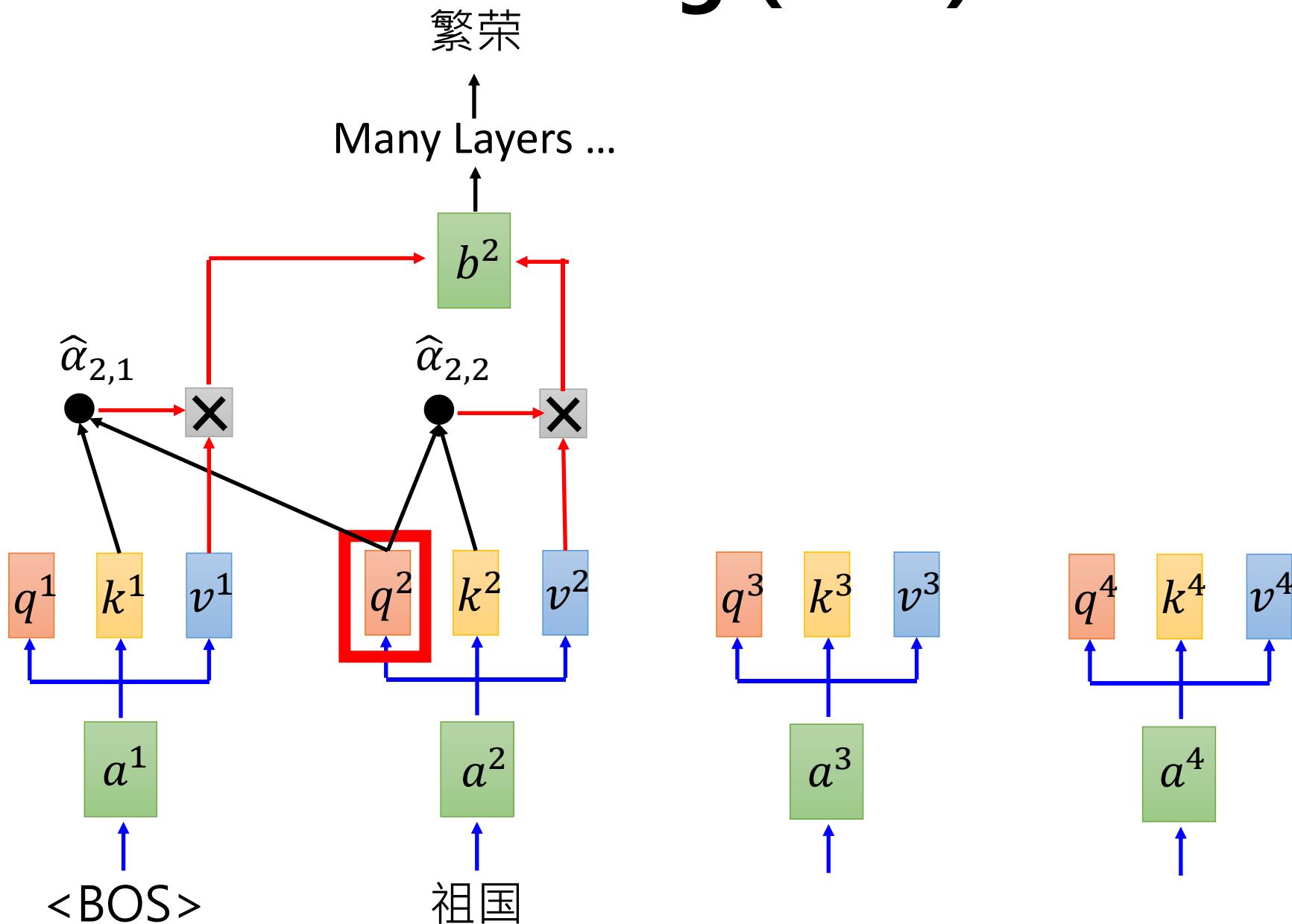


https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Source of image: <https://huaban.com/pins/1714071707/>



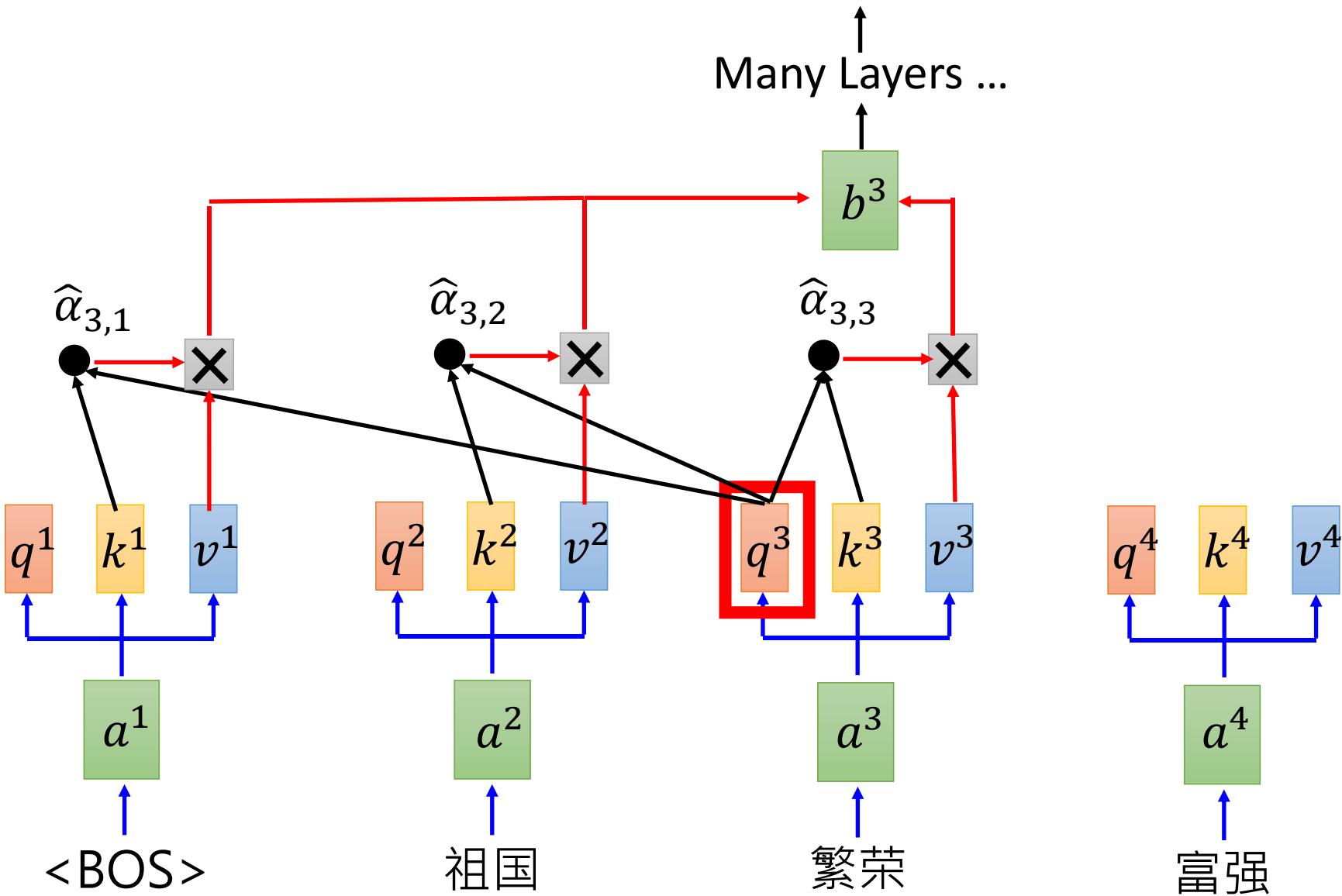
Generative Pre-Training (GPT)





Generative Pre-Training (GPT)

富强



GPT-1和BERT的对比

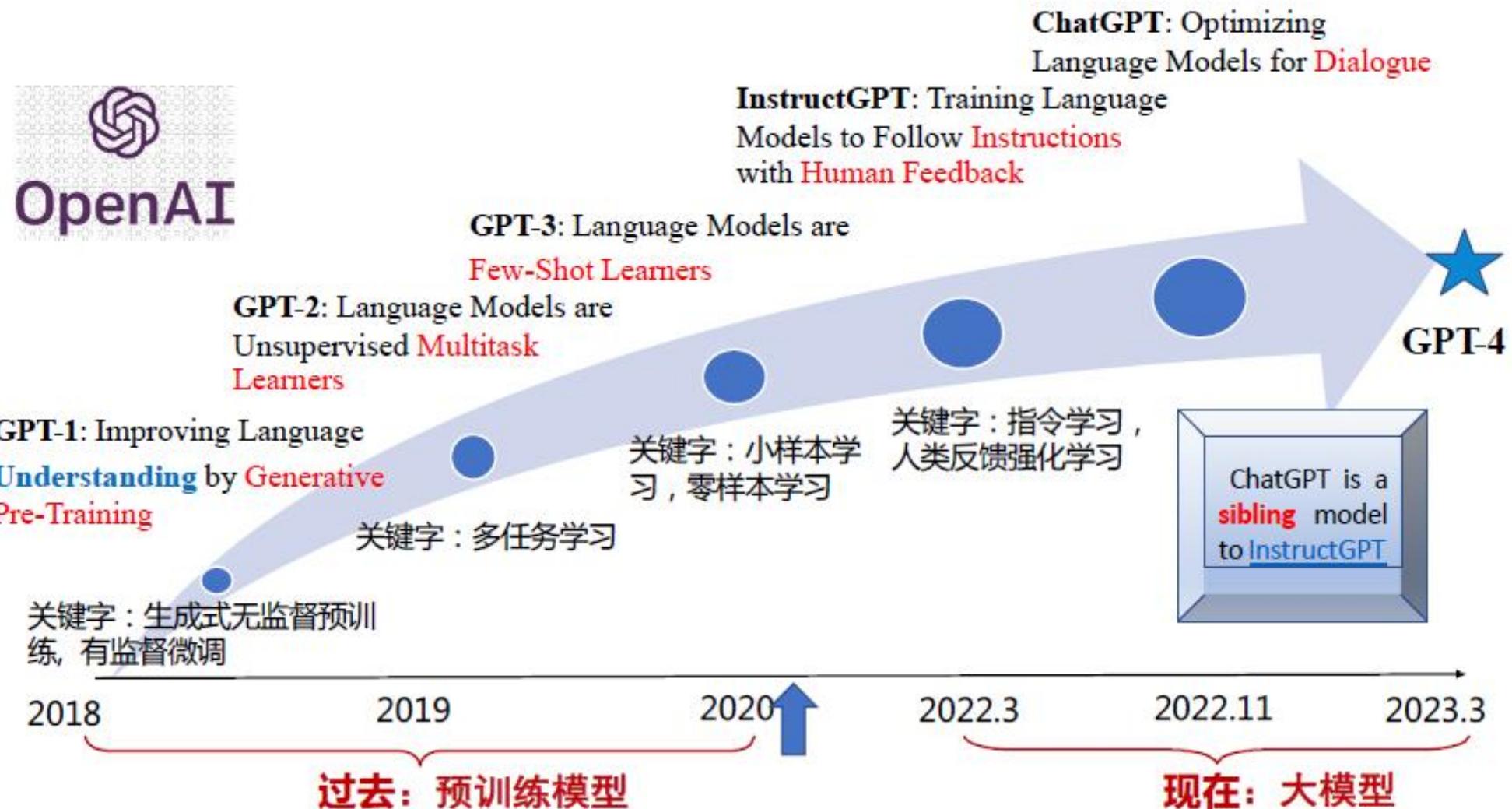


	BERT	GPT-1
提出时间	2018年10月11日	2018年6月
模型结构	双向 (Encoder)	单项 (Decoder)
主要应用	自然语言理解	自然语言生成
预训练任务	“完形填空”	“词汇接龙”
预训练难度	相对易	相对难
模型大小	340M	117M
模型性能	相对优	相对差
代码开源	是	否



由于BERT开源而GPT不开源，在后面很长一段时间内BERT生态圈一直比GPT好很多

ChatGPT是2022年11月美国人工智能公司OpenAI所推出的生成式对话预训练模型。它通过对话的形式进行交互，对话的形式使得其能够回答后续问题，承认自己的错误，质疑不正确的前提，并拒绝不适当的请求。



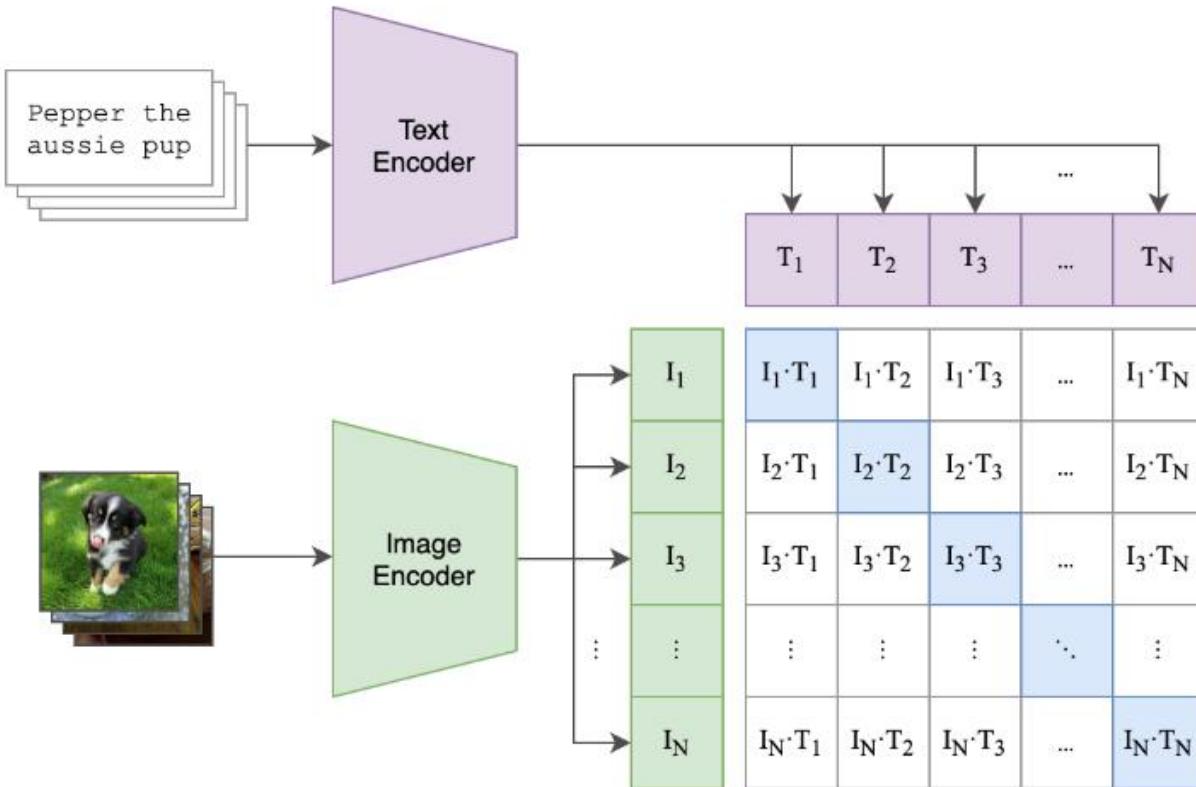
模型性能和参数规模不断提升，应用领域更加广泛，通用性持续加强

Learning Transferable Visual Models From Natural Language Supervision

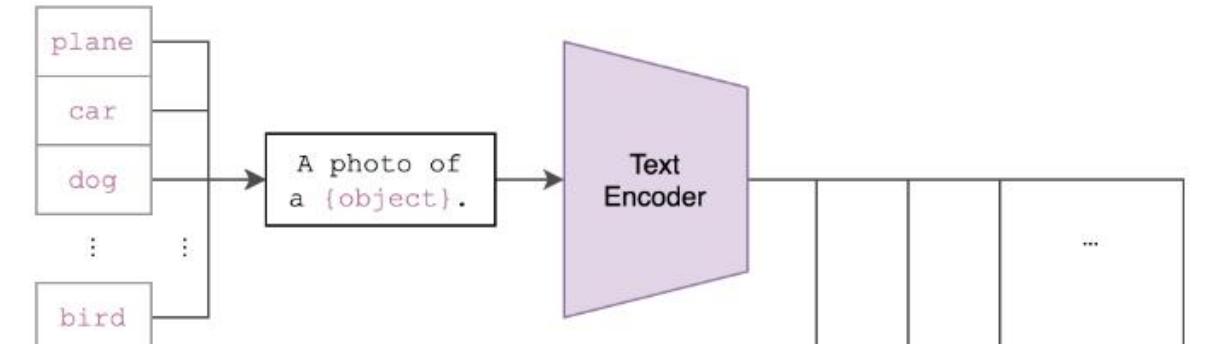
[Alec, Jong, ..., 2021]

- Pretrained visual-concept model
- Zero-shot transfer of the model to downstream tasks

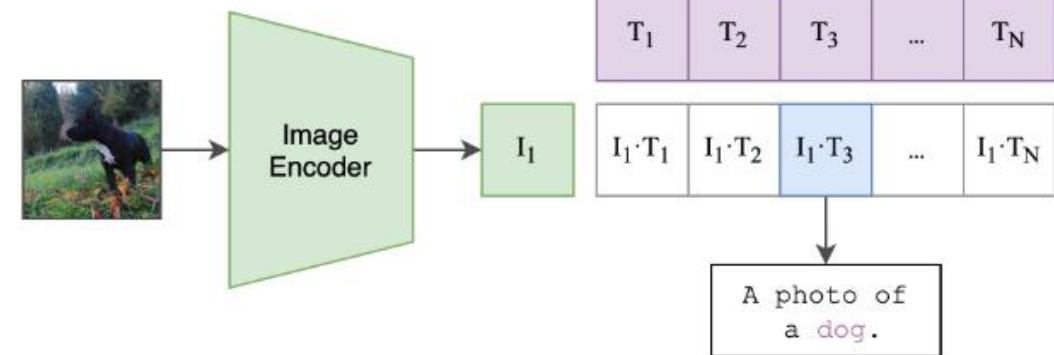
(1) Contrastive pre-training

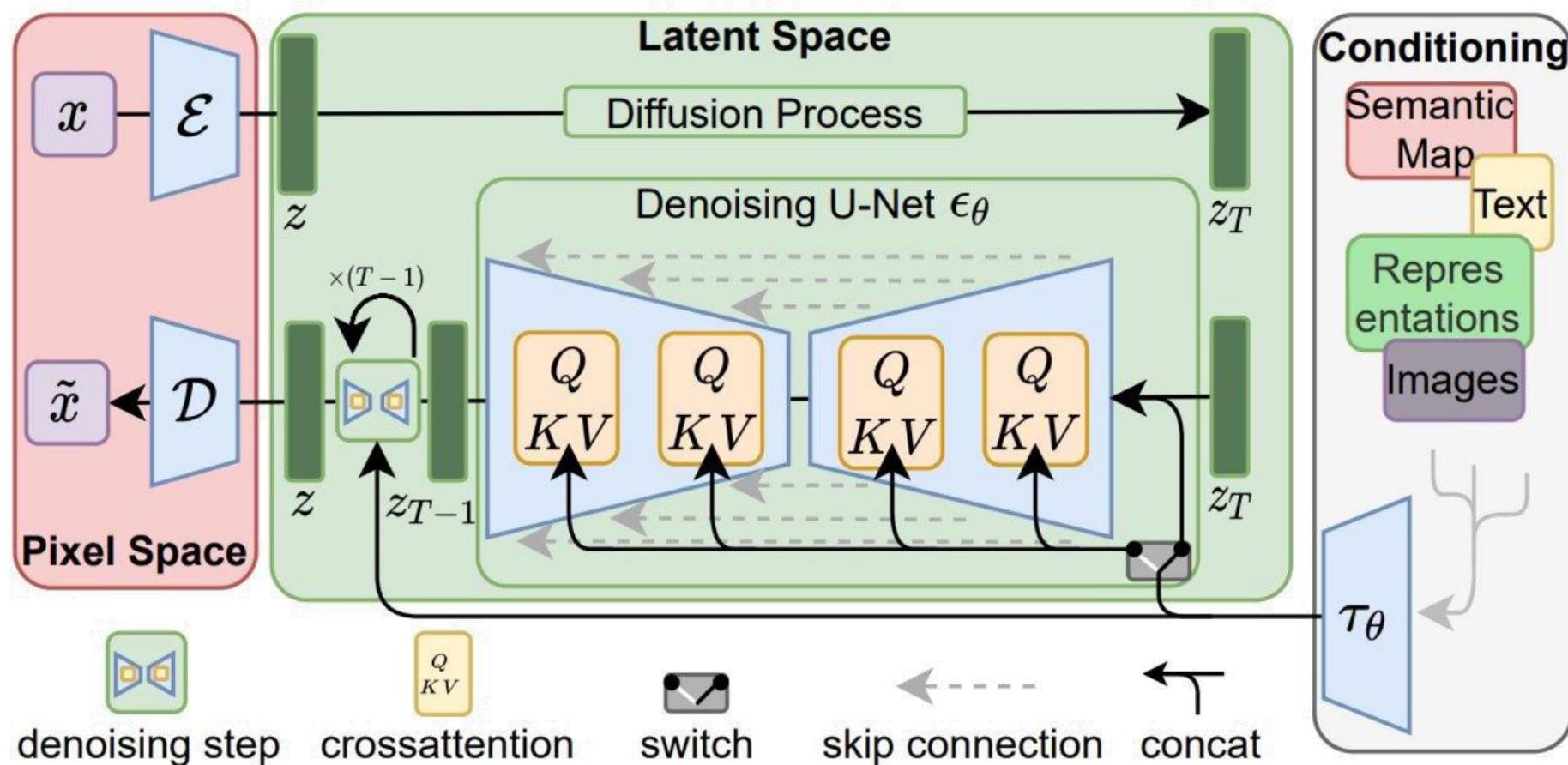


(2) Create dataset classifier from label text



(3) Use for zero-shot prediction





Course Outline

6 Personal assignments



6 Group assignments



上课日期	授课内容	实验内容	周次
20240905	课程入门、大数据探索式分析	/	第一周
20240912	课程实践项目介绍、项目组队测试、项目经验谈	项目成员集结	第二周
20240919	科研实践入门、数据采样与降维	项目管理工具制定项目计划、Pandas数据采样实践	第三周
20240926	数据质量管理	Pandas数据质量实践	第四周
20241003	/	/	第五周
20241010	众包与电子表格	电子表格实践	第六周
20241017	可视化设计	可视化设计实践	第七周
20241024	统计分析方法与工具	统计方法实践	第八周
20241031	中期汇报 (论文+项目进展) 1	中期进展报告	第九周
20241107	中期汇报 (论文+项目进展) 2	BERT实践环境配置	第十周
20241114	机器学习方法与工具	BERT实践	第十一周
20241121	人机交互方法与工具	Canis/Cast/Libra实践	第十二周
20241128	普适计算	手机移动数据采集与分析	第十三周
20241205	大规模数据分析系统	SPARK实践	第十四周
20241212	如何撰写项目论文	大项目收尾	第十五周
20241219	项目结题报告1	大项目验收	第十六周

Thank You

