CIS 6930 Topics in Computing for Data Science
Week 7: Review Session + Discussion

10/5/2021

Yoshihiko (Yoshi) Suhara

2pm-3:20pm

This Week & Next Week

- Thu 10/7 Midterm exam (written exam **on campus**)
  - Time: 2pm-3:20pm
  - Location: LBR 252

- Fall Break

Week 7!

- ~~Week 1: Deep Learning Basics (Thu 9/9)~~
- ~~Week 2: AutoEncoder (Tue 9/14)~~
- ~~Week 3: Convolutional Neural Networks (Thu 9/16)~~
- ~~Week 4: GAN (Tue 9/21)~~
- ~~Week 5: Word embeddings: Word2vec, GloVe (Thu 9/23)~~
- ~~Week 6: Recurrent Neural Networks (Tue 9/28, Thu 9/30)~~
- **Week 7: Review/Project pitch & Mid-term (Tue 10/5, Thu 10/7)**
- Fall Break
- ...

Week 7!

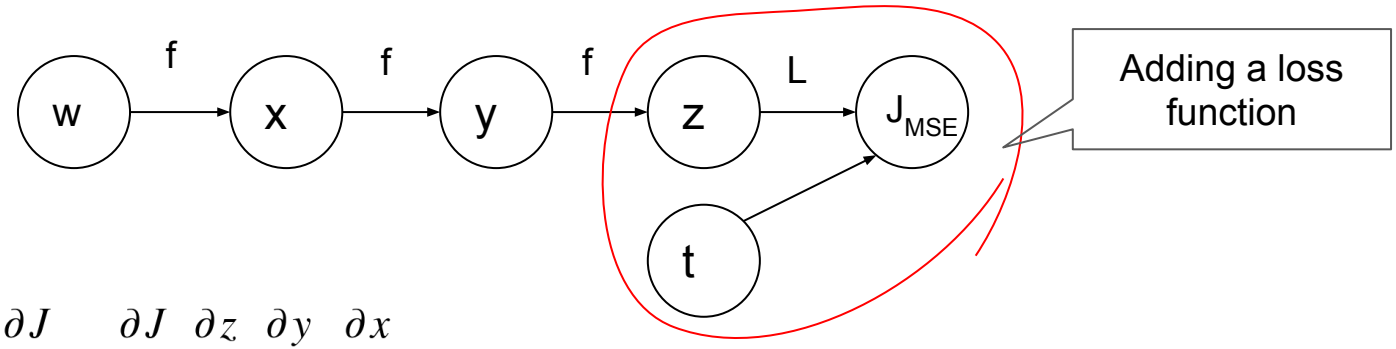
- **Week 1: Deep Learning Basics (Thu 9/9)**
- **Week 2: AutoEncoder (Tue 9/14)**
- **Week 3: Convolutional Neural Networks (Thu 9/16)**
- **Week 4: GAN (Tue 9/21)**
- **Week 5: Word embeddings: Word2vec, GloVe (Thu 9/23)**
- **Week 6: Recurrent Neural Networks (Tue 9/28, Thu 9/30)**
- Week 7: Review/Project pitch & Mid-term (Tue 10/5, Thu 10/7)
- Fall Break
- ...

This review session is to quickly go through **the basics** of the topics in the previous sessions and not necessarily cover every important topic

Week 1 Deep Learning Basics

Backpropagation from 3000 ft

- Multiplications of **the derivative** of each step + **Outputs** of the previous step



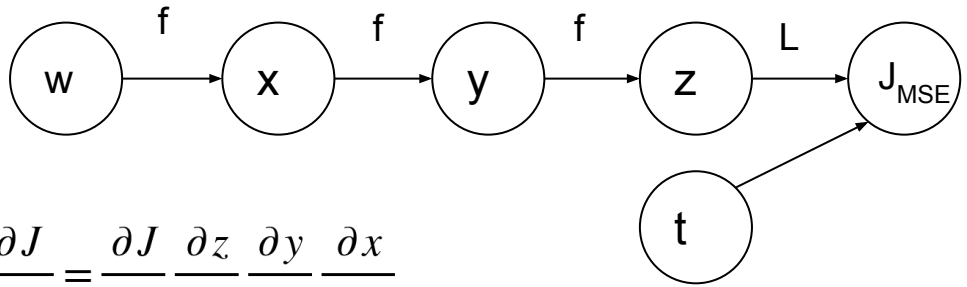
$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w}$$

$$= L'(z, t) f'(y) f'(x) f'(w)$$

$$= L'(f(f(f(w))), t) f'(f(f(w))) f'(f(w)) f'(w)$$

Differentiability is Key!

- The gradients of any parameters can be calculated **as long as the functions are differentiable!**
- Backpropagation = A gradient calculation method
  - i.e., can be coupled with any optimization method (e.g., SGD)
- Multiplications of **the derivative** of each step + **Outputs** of the previous step

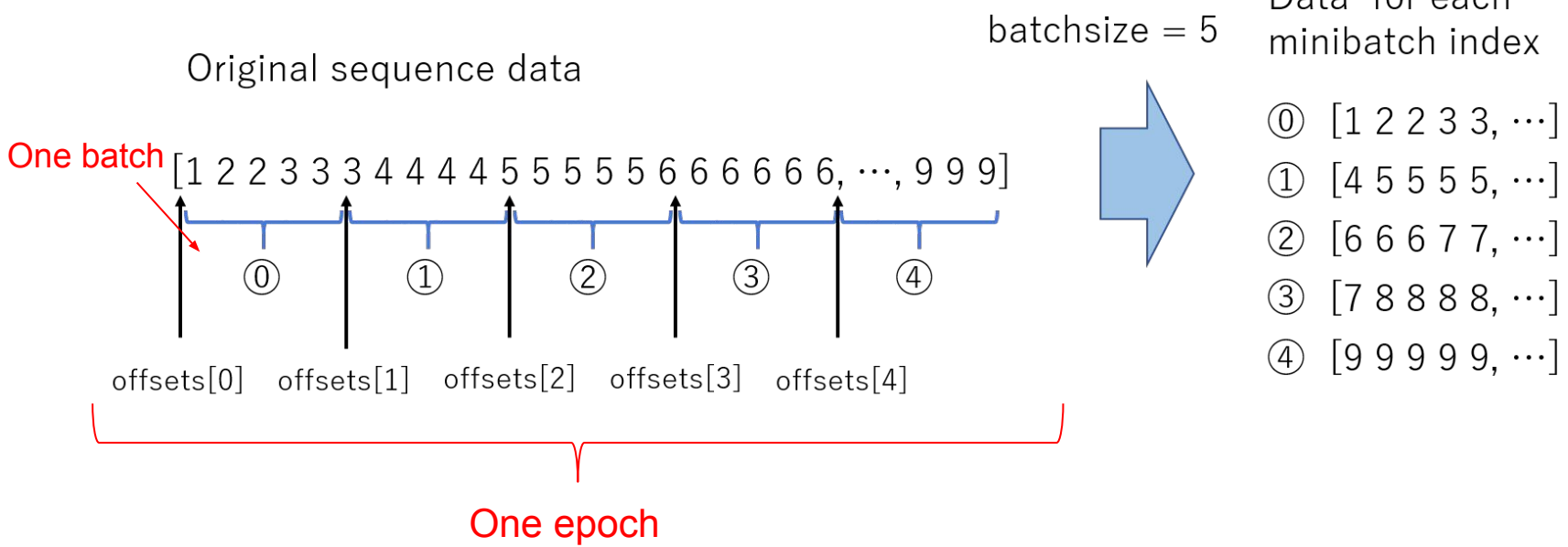


$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w}$$

$$= L'(z, t) f'(y) f'(x) f'(w)$$

$$= L'(f(f(f(w))), t) f'(f(f(w))) f'(f(w)) f'(w)$$

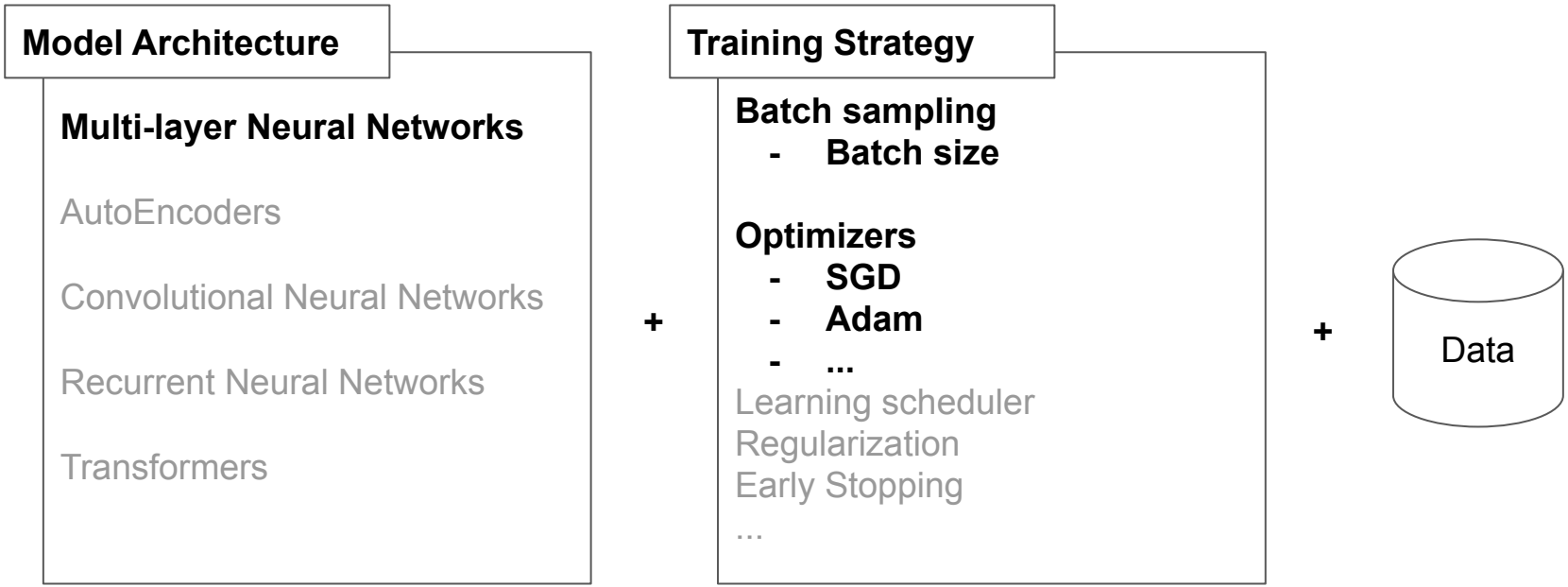
How Mini-batch Sampling Works?



http://corochann.com/training-rnn-with-simple-sequence-dataset-1304.html

Basic Deep-Learning Building Blocks

- **(A) Model Architecture + (B) Training Strategy + (c) Data**
- i.e., What to Optimize and How to Optimize



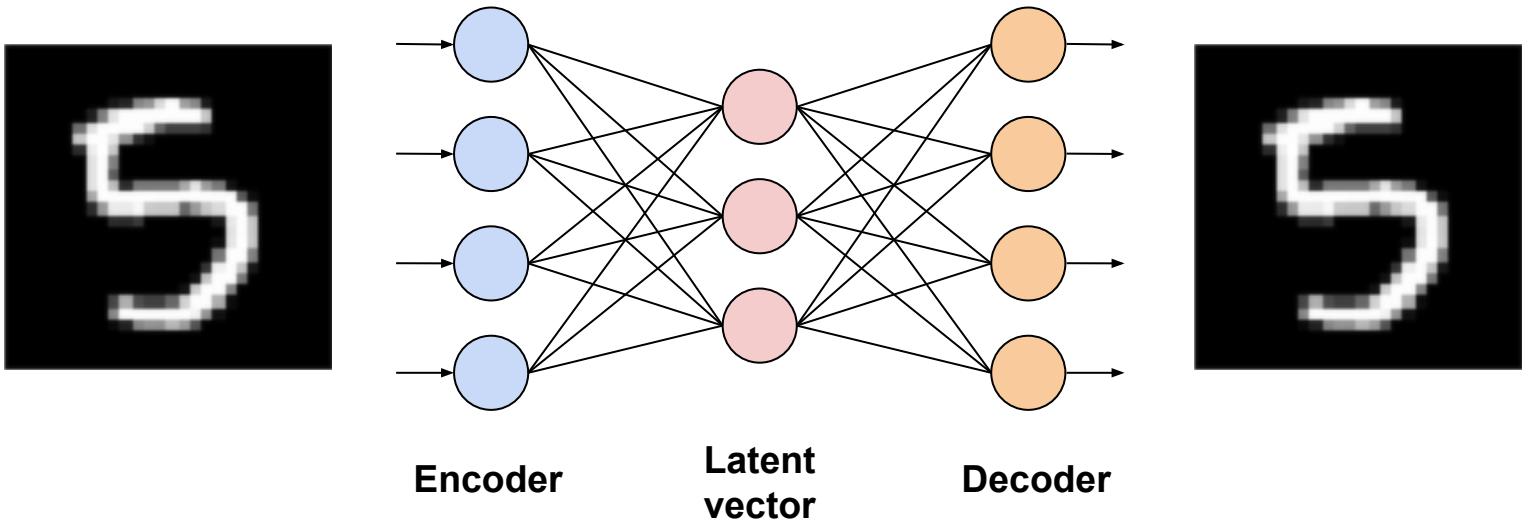
Deep-Learning Building Blocks: Starter Kit

- Layers
  - Linear Layer
- Activation functions
  - Logistic sigmoid, tanh
  - ReLU, Leaky ReLU
- Optimizers
  - SGD w/wo Momentum)
  - Adam

Week 2 Autoencoders

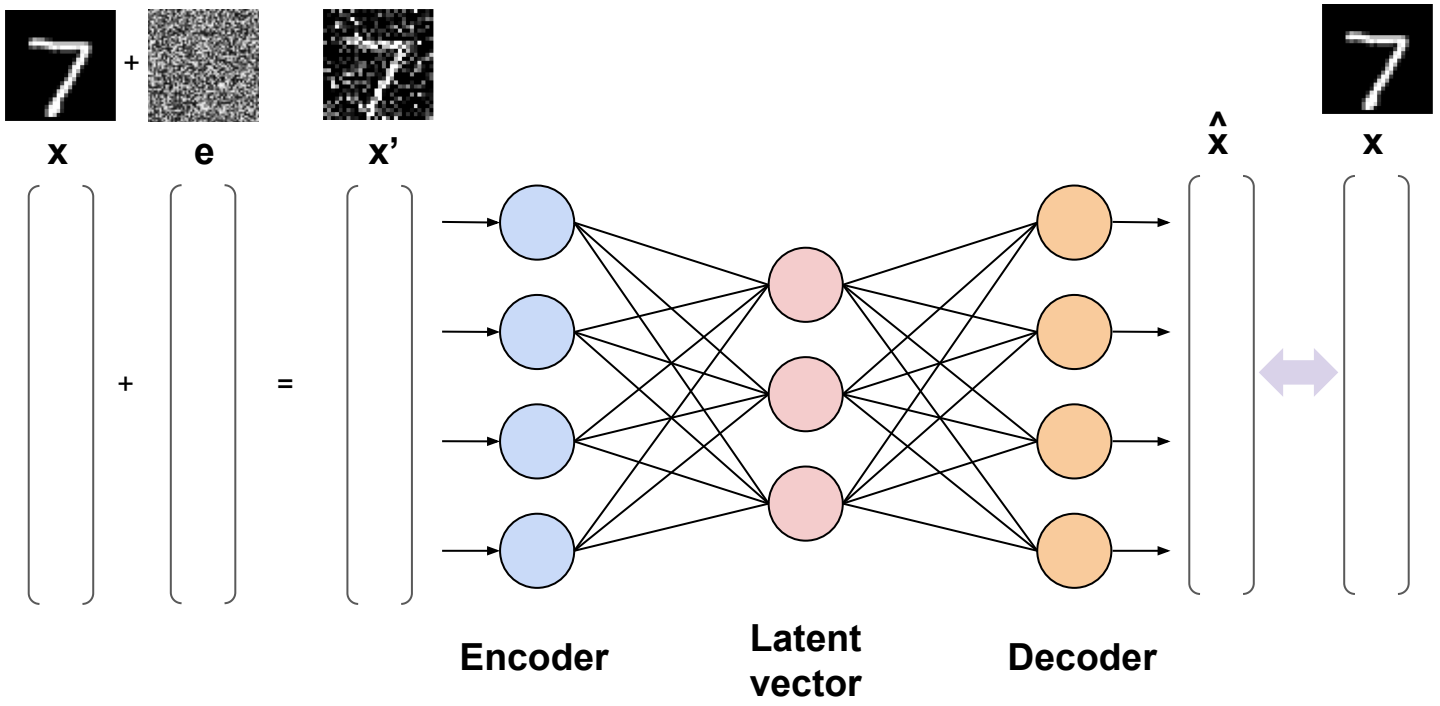
Recap: Autoencoders

- **Encoder-decoder models** that **learn to reconstruct the original data**



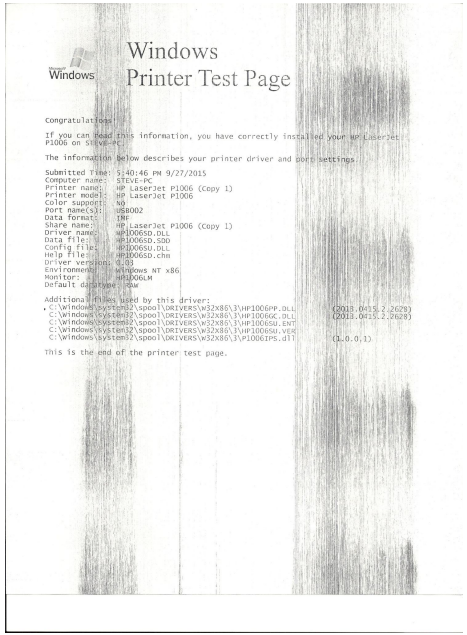
Recap: Denoising Autoencoder

- Learn to reconstruct the original data from **corrupted input**



Recap: Autoencoder Applications

- Image restoration problems in general

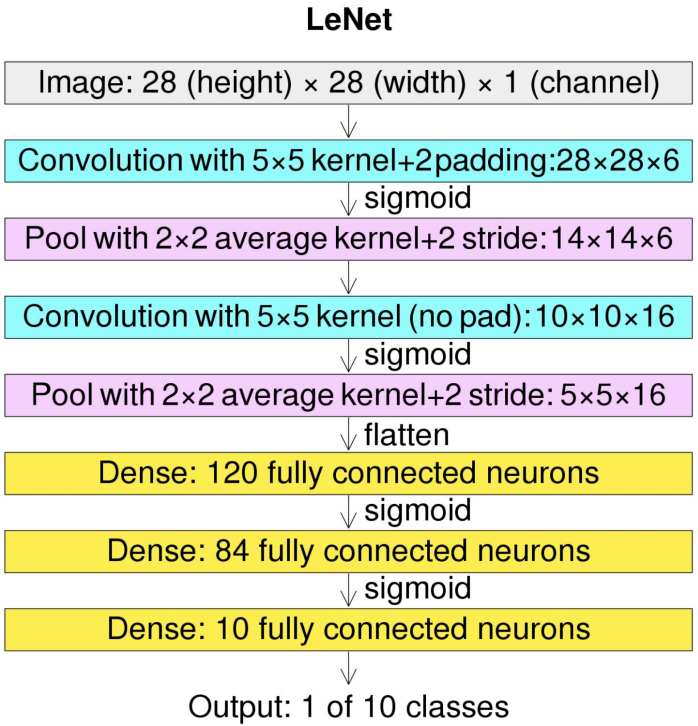


How do you train Autoencoder models for the problems?

Week 3 Convolutional Neural Networks

Key Concepts in CNN

- Convolution
- Pooling
- Fully-connected Layer



Convolution: Terminology

$x_{1,1}$	$x_{1,2}$	…	$x_{1,M}$
…			
…			
$x_{N,1}$	…		$x_{N,M}$

Input image /Feature map

*

w_1	w_2
w_3	w_4

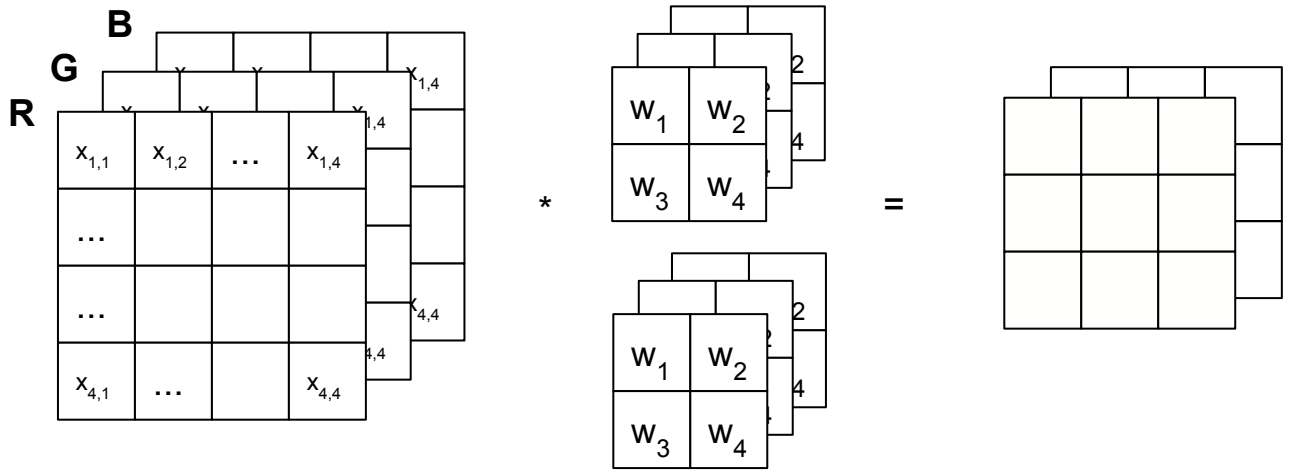
Convolutional filter (aka kernel)

=

Feature map

Convolutions on RGB image

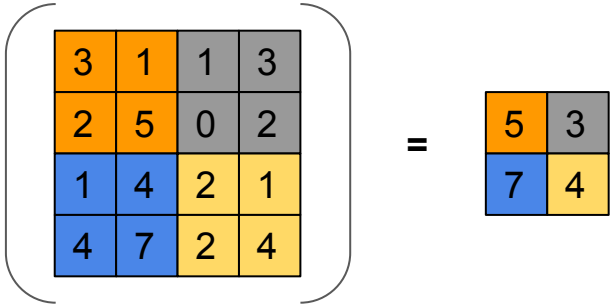
```
Conv2d(in_channels=3, out_channels=2, kernel_size=2, stride=1, padding=0)
```



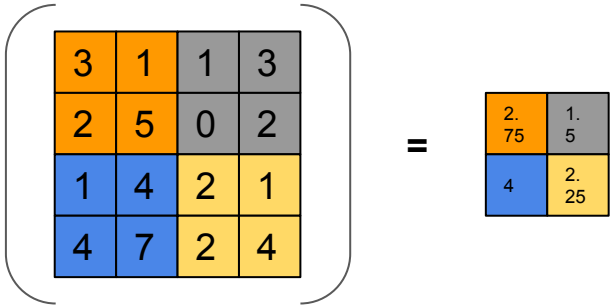
(1, 3, 4, 4) (2, 3, 2, 2) (1, **2**, 3, 3)

Mean and Max Pooling

Max-pooling

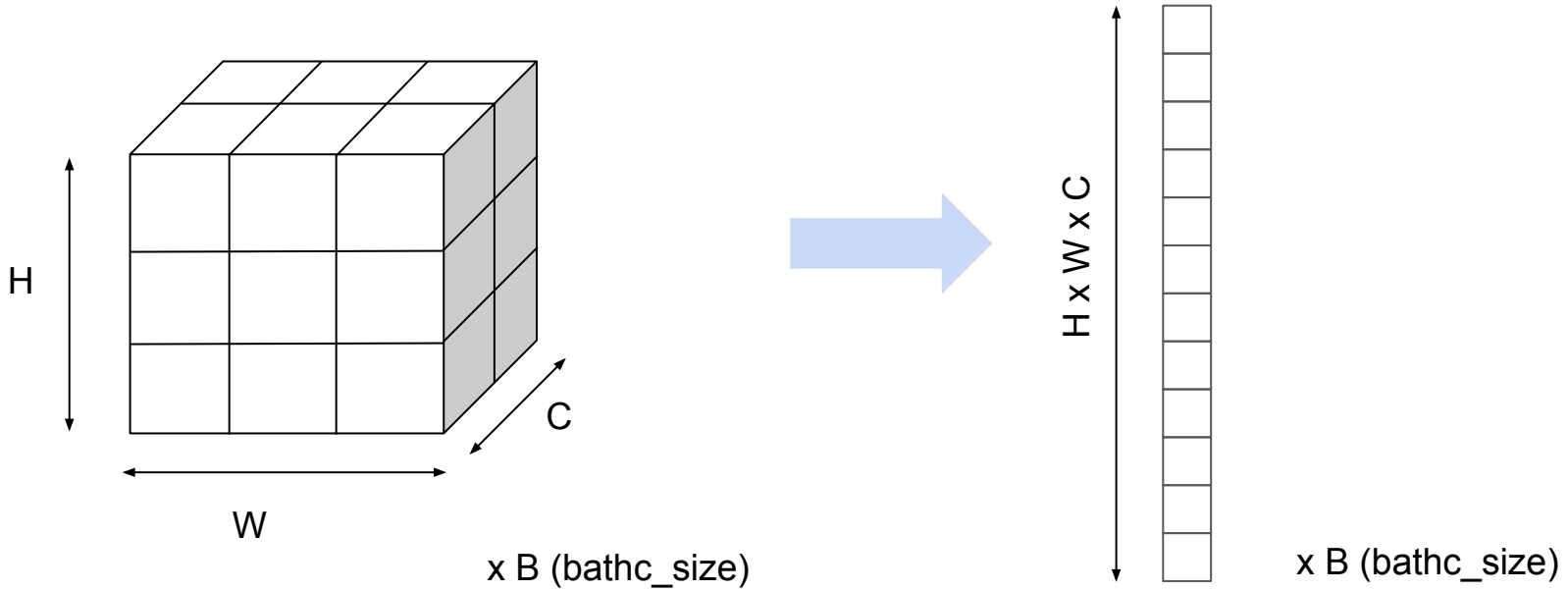


Mean-pooling

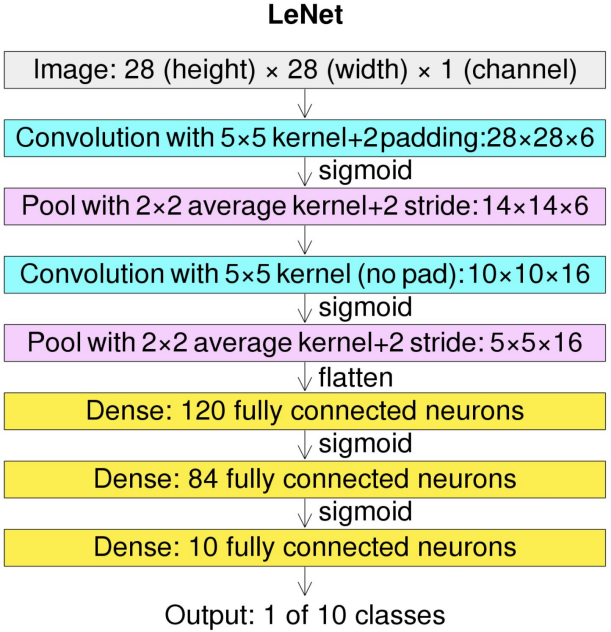


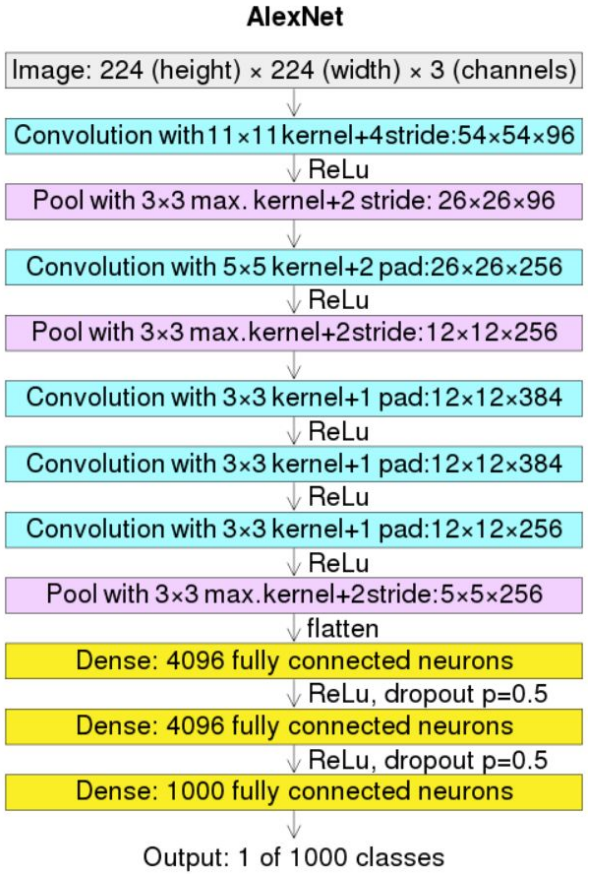
Flattening Feature Maps into Vectors

- Feature maps are not directly compatible with `Linear` layers
  - (batch_size, #channel, height, width) =! (batch_size, dim)



Model Design → Blueprint → PyTorch code

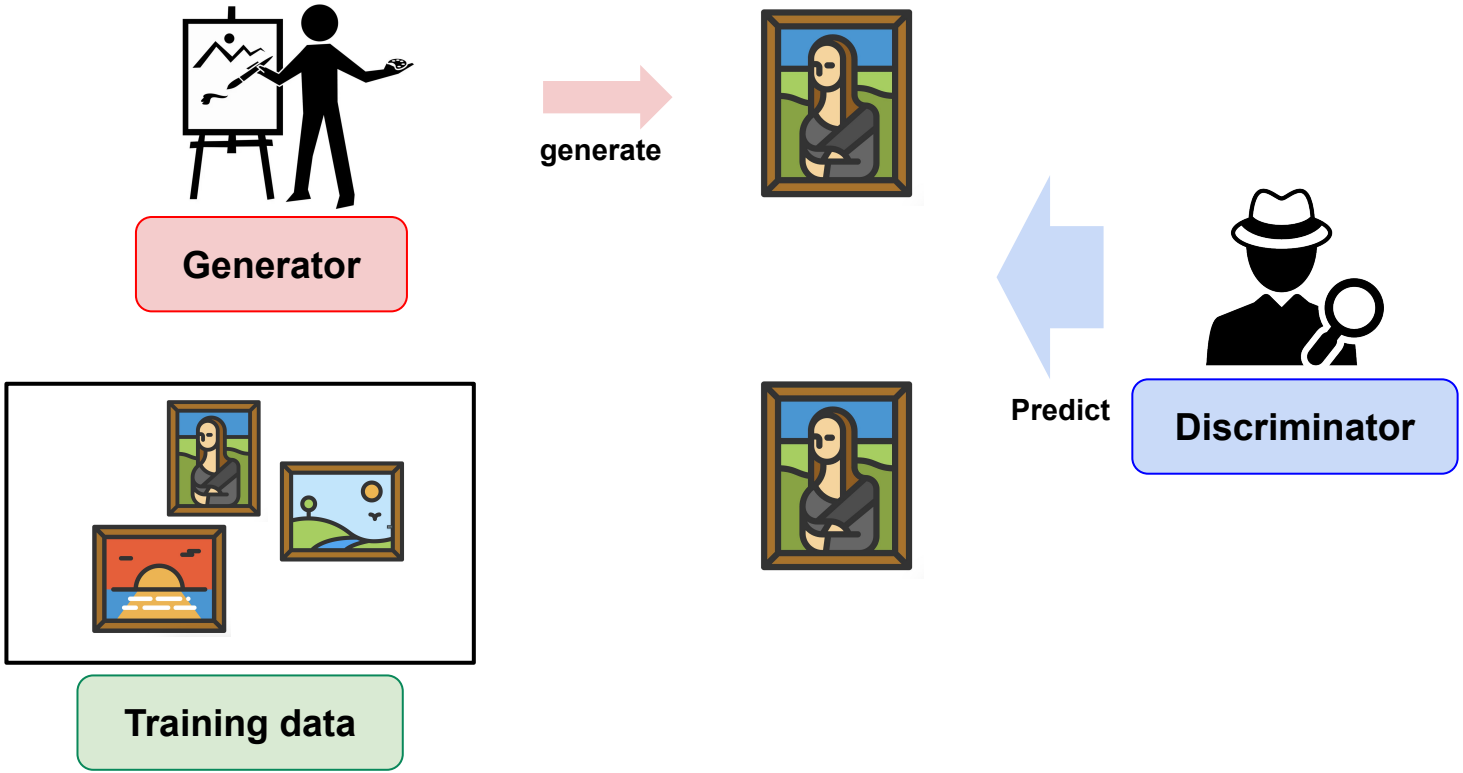




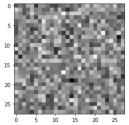
Week 4 Generative Adversarial Networks

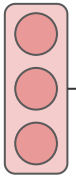
Generative Adversarial Network (GAN)

Generator + Discriminator + Training data

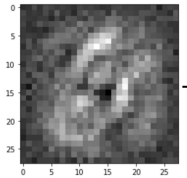


GAN Architecture





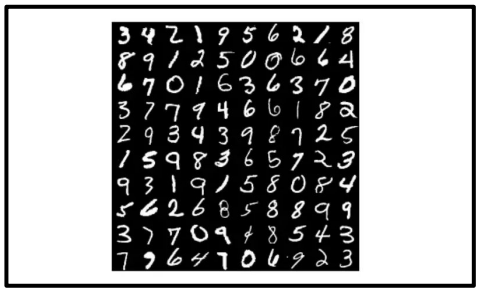
Decoder



Generator

Encoder

Real/Fake



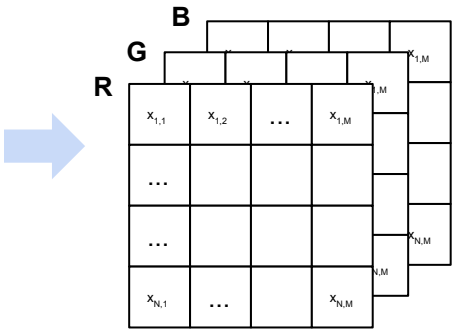
Training data

Discriminator

Week 5 NLP Basics & Word Embeddings

What Makes Text Processing Difficult for Computers?

- Symbolic representations cannot be directly converted into vector/tensor representations
  - cf. Images



Images

"A cat is starting at me" → ???

Text

One-hot Vector Representation

- Assign one dimension to each word

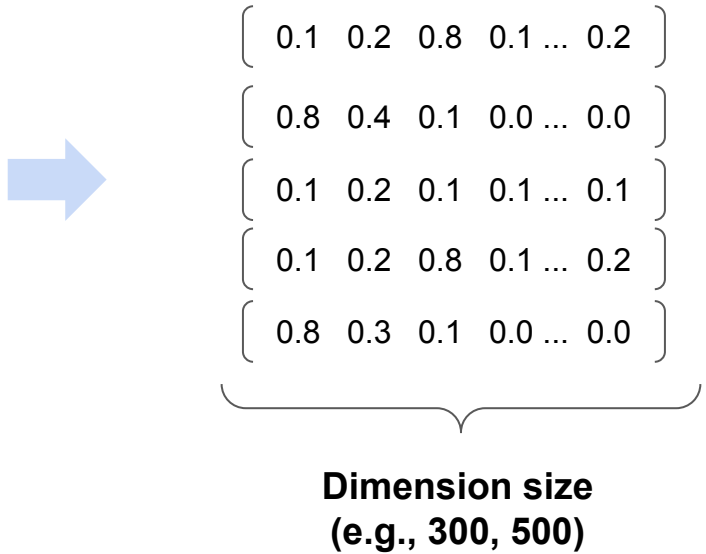
“A cat and a dog”

a	cat	and	dog		
1	0	0	0	...	0
0	1	0	0	...	0
0	0	1	0	...	0
1	0	0	0	...	0
0	0	0	1	...	0

Vocabulary size

Word Embeddings: *Dense* Vector Representations

"A cat and a dog"

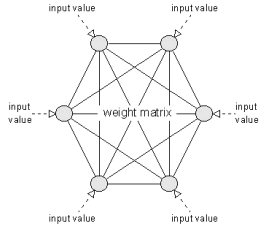


Dense vector representations

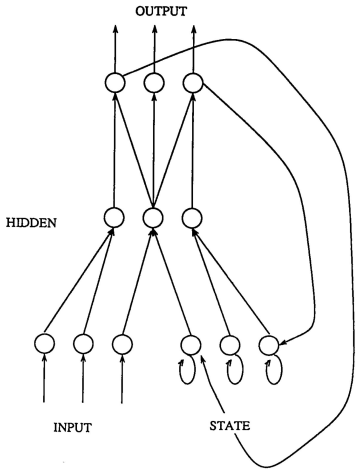
Week 6 Recurrent Neural Networks

Recurrent Neural Networks for Sequential Input

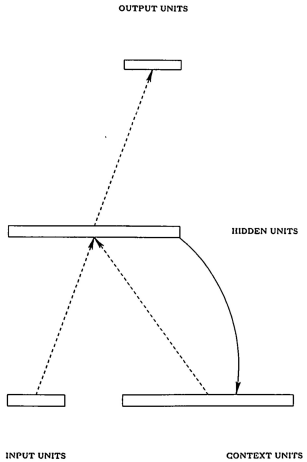
- Inspired by Hopfield Network (1982)
- “Simple” Recurrent Neural Networks
  - Jordan Net (1986)
  - **Elman Net (1990)**



Hopfield Network

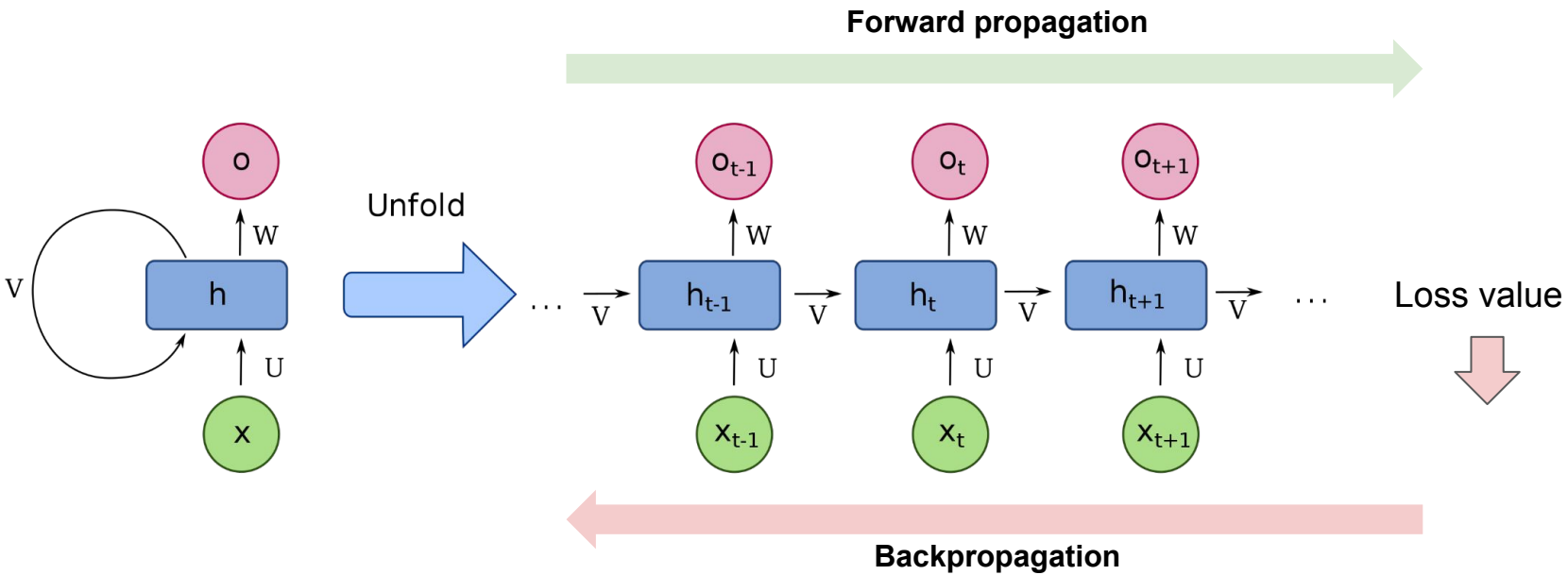


Jordan Net



Elman Net

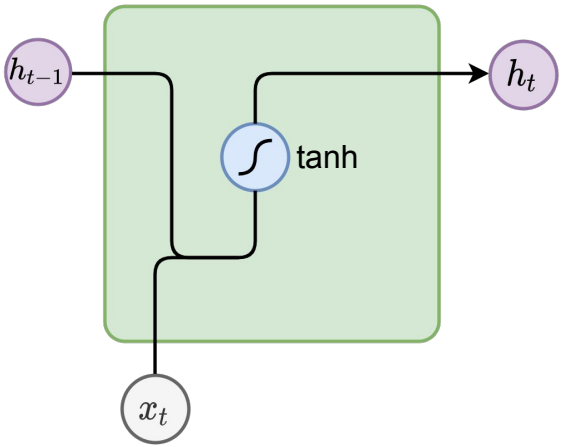
Unfolding RNN
Backpropagation Through Time (BPTT)



RNN Cell for Sequential Data

RNN = Elman Net

- Input value + previous hidden state → Next hidden state



$$h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{(t-1)} + b_{hh})$$

$$W_h\left[x_t; h_{(t-1)}\right] + b_h$$

Another notation

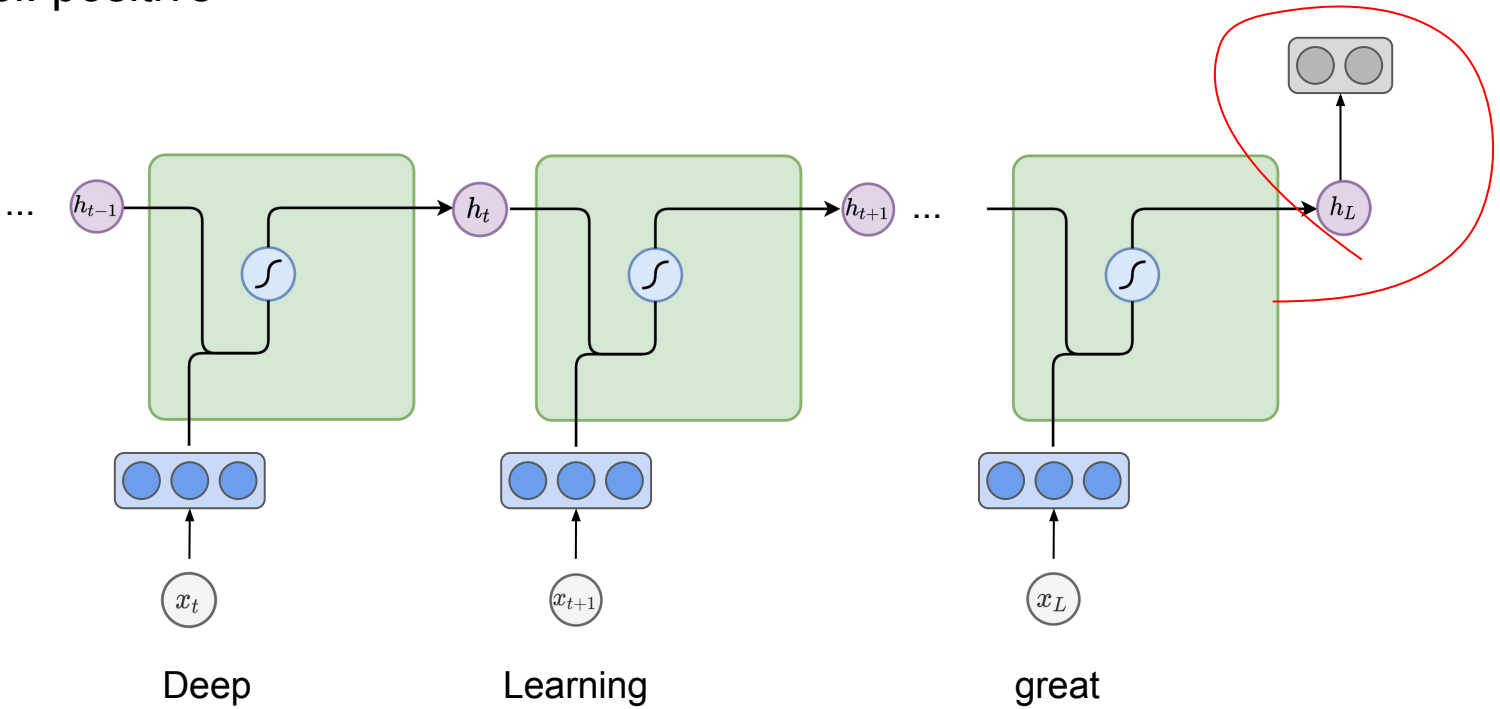
The figures are based on the following blog article (highly recommended!)

Illustrated Guide to LSTM's and GRU's: A step by step explanation

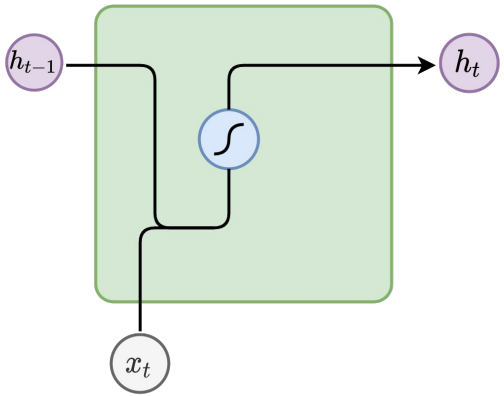
Text Classification Example

- Input: “Deep Learning is great”
- Label: positive

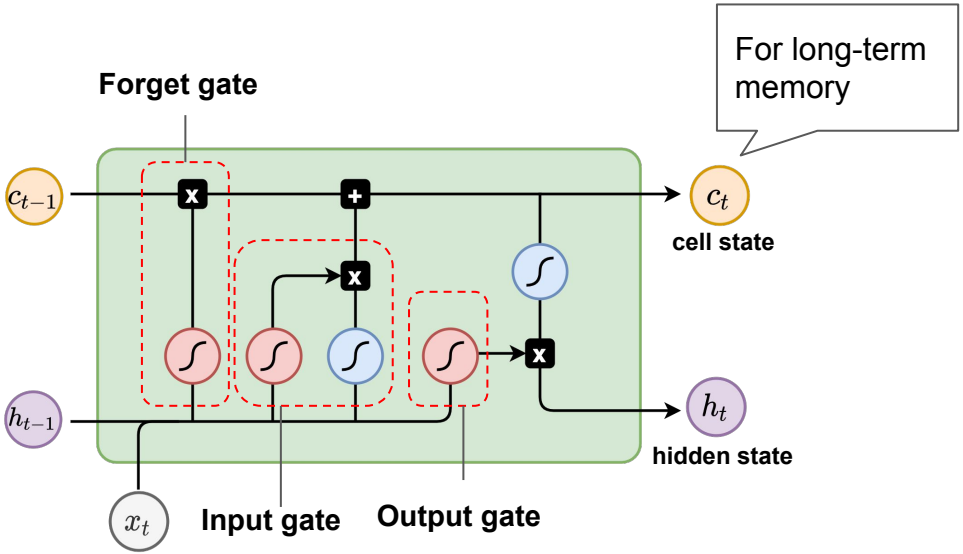
Any other options?



LSTM: Cell State + Gating Mechanism

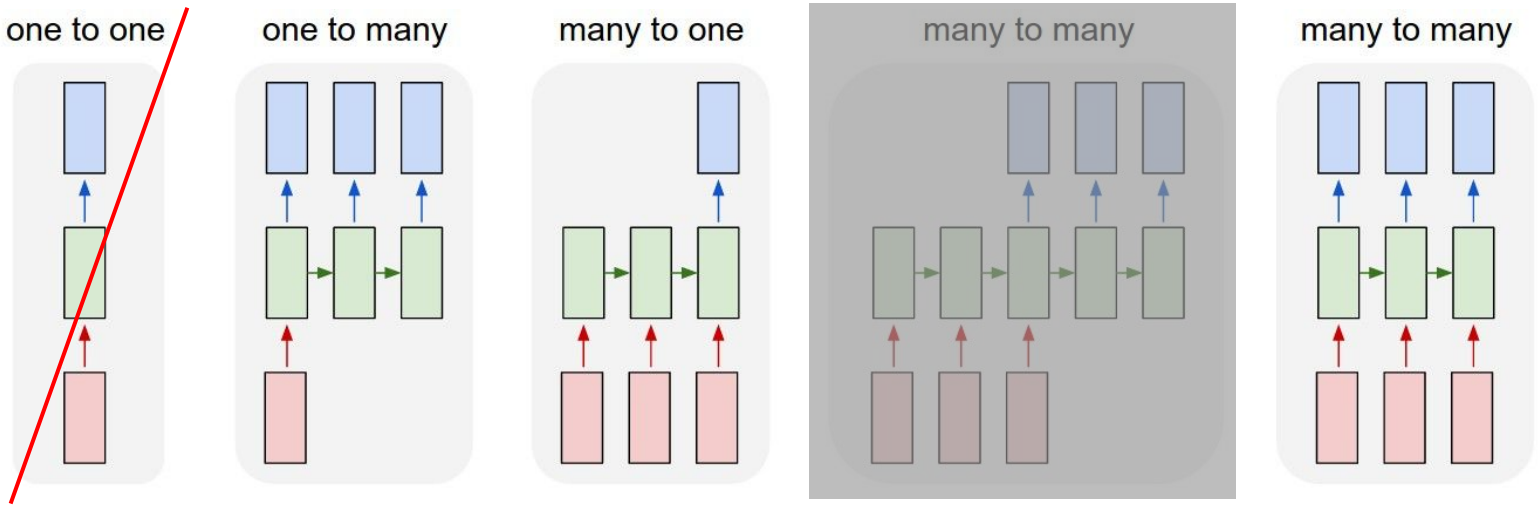


RNN Cell



LSTM Cell

Key Takeaway 1: RNN Application Patterns



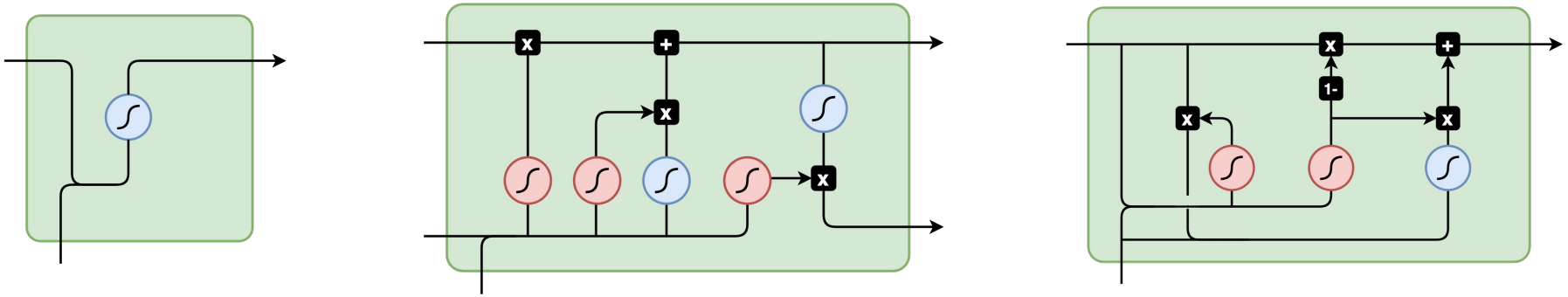
Single-input Generation?

Classification

Encoder-decoder model
(aka seq2seq)

Sequential tagging

Key Takeaway 2: Three Cells



Questions?

Discussions for Term-Project

- Any ideas?