Convolutional Neural Networks

CS5242

Lee Hwee Kuan & Wang Wei

Teaching assistant:

Connie Kou Khor Li, Ji Xin, Ouyang Kun

cs5242@comp.nus.edu.sg

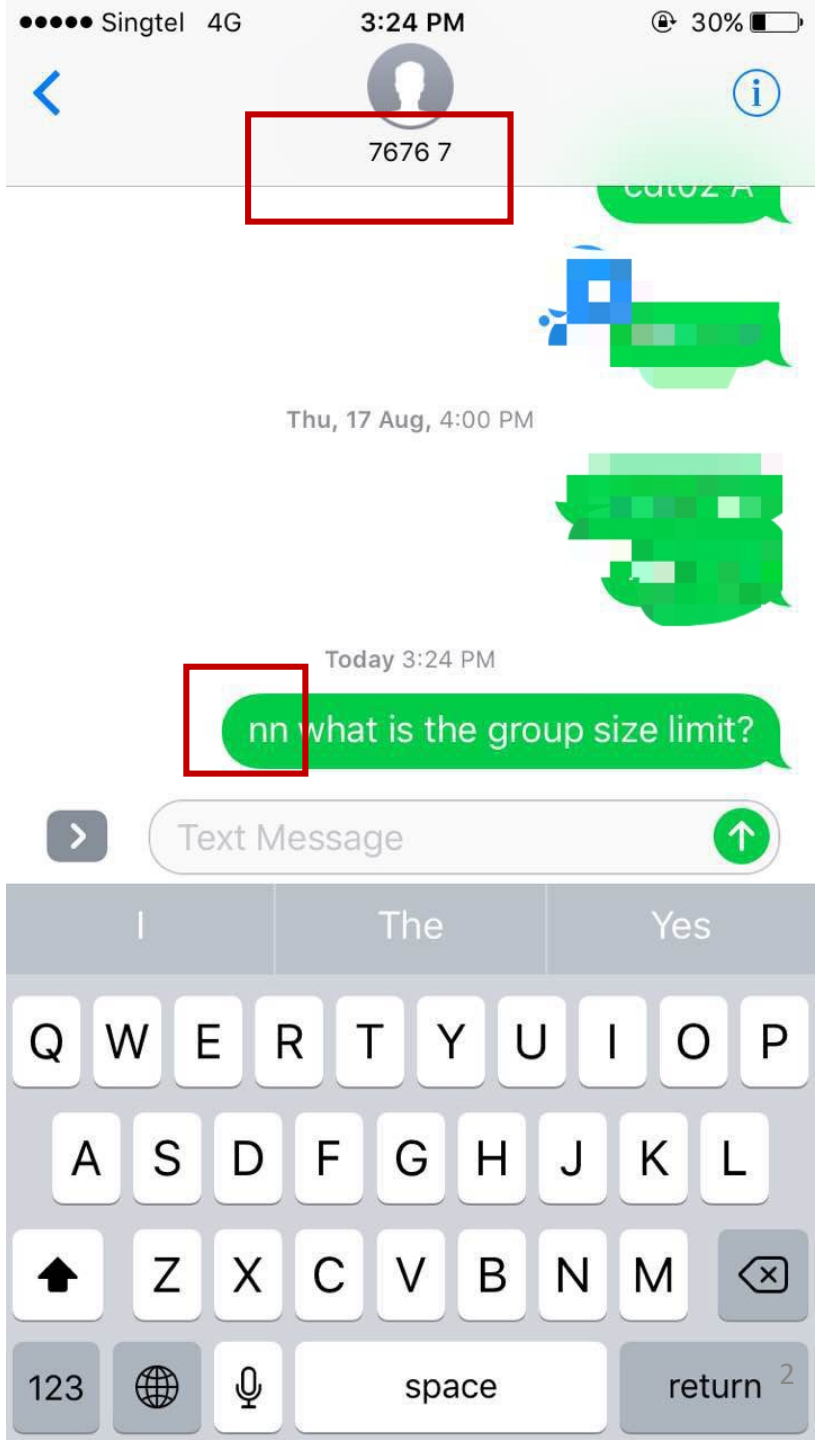
Questions

SMS to **76767** **OR** https://peerq.nus.edu.sg.

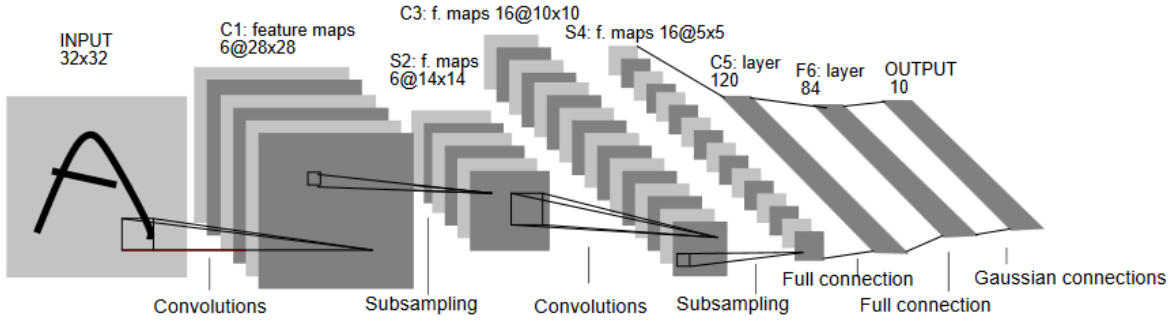
Content: <code><space><answer or question>

For exmaple: "**nn** what is the group size limit"

code question



History



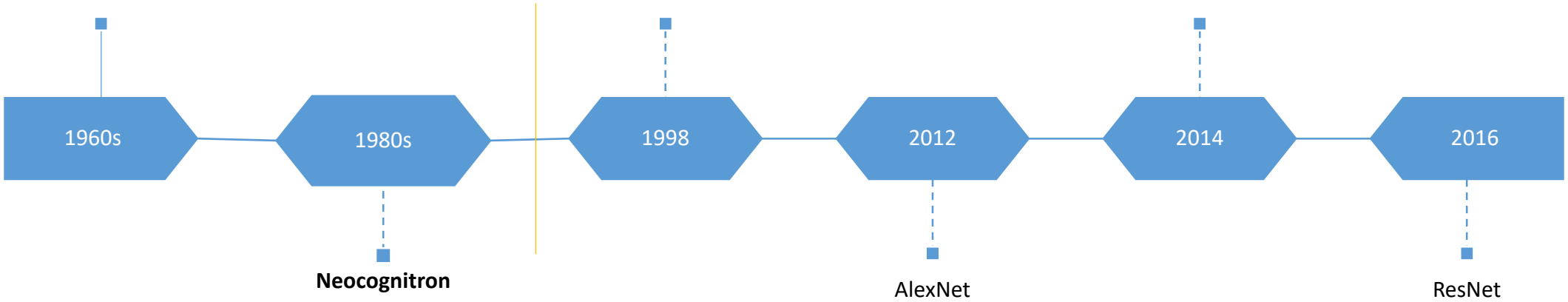
Source from [1]

Visual cortex

Receptive field

Simple cells

complex cells



"sandwich" architecture (SCSCSC…)
simple cells: modifiable parameters
complex cells: perform pooling

Roadmap

- Convolution and Pooling
  - 1D convolution
  - 2D convolution
  - Pooling
- Architectures
  - AlexNet, VGG, InceptionNet, ResNet, DenseNet, XceptionNet
- Training
  - Activation functions
  - Normalization functions
  - Hyper-parameters
- Applications
  - Classification, Detection, NeuralStyle

Intended learning outcome

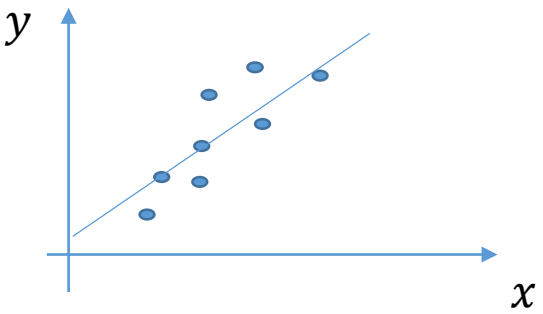
Know	Know the difference between convolution and MLP
Understand	Understand the characteristic of convolution layers
Calculate	Calculate the size of convolution outputs and kernel size
Implement	Implement 1D and 2D conv operations from scratch

From MLP to Convolution

- House price prediction (https://www.kaggle.com/harlfoxem/housesalesprediction)
  - location x_1
  - # bedrooms x_2
  - # bathrooms x_3
  - # size(sqft) x_4
  - Etc.
- MLP
  - $y = relu(x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4 + b)$

$?

Source from http://www.jwbrealestatecapital.com/real-estate-rental-properties-for-sale-in-north-jacksonville/



From MLP to Convolution

- Predict the digit in an image? (http://yann.lecun.com/exdb/mnist/)
- Image representation
  - height 28 pixels, width 28 pixels
  - size=28x28=784
- Multi-layer perceptron (MLP)
  - $h^i = relu\left(h^{i-1}W^i + b^i\right), W^i \in R^{|\boldsymbol{h^{i-1}}|\times|\boldsymbol{h^i}|}, b^i \in R^{|h^i|}, h^0 = x$
  - $y = softmax(h^{n-1})$

Source from: deeplearning.net/tutorial/

From MLP to Convolution

- Problems:
  - $W^i \in R^{|\boldsymbol{h^{i-1}}| \times |\boldsymbol{h^i}|}$
    - 2500x2000=5,000,000
  - Too many parameters → overfitting

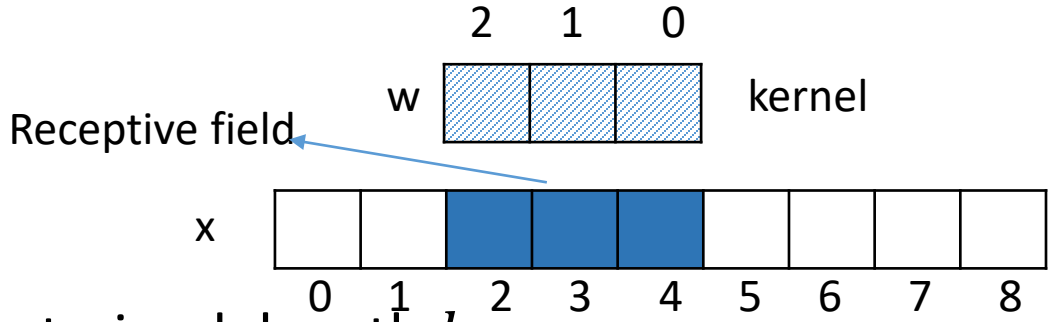
NN architecture	Dataset	Distortions	Test Error [%]
MLP:2500-2000-1500-1000-500-10	MNIST	no	1.47
MLP:2000-2000-2000-2000-2000-2000-2000-10	MNIST	no	1.531 ± 0.051
MLP:1500-1500-1500-1500-1500-1500-1500-10	MNIST	no	1.513 ± 0.052
MLP:1000-1000-1000-1000-1000-1000-1000-1000-1000-10	MNIST	no	1.628 ± 0.035
MLP:1000-1000-1000-1000-1000-1000-1000-1000-10	MNIST	no	1.542 ± 0.052
MLP:1000-1000-1000-1000-1000-1000-1000-10	MNIST	no	1.517 ± 0.069
MLP:1000-1000-1000-1000-1000-1000-10	MNIST	no	1.529 ± 0.078
MLP:1000-1000-1000-1000-1000-10	MNIST	no	1.571 ± 0.046
MLP:1000-1000-1000-1000-10	MNIST	no	1.549 ± 0.038
MLP:1000-1000-1000-10	MNIST	no	1.650 ± 0.030
MLP:500-500-500-500-500-500-500-10	MNIST	no	1.744 ± 0.038
MLP:500-500-500-500-500-500-10	MNIST	no	1.702 ± 0.064
MLP:500-500-500-500-500-10	MNIST	no	1.719 ± 0.069
MLP:500-500-500-500-10	MNIST	no	1.728 ± 0.028
MLP:500-500-500-10	MNIST	no	1.765±0.040
MLP:2000-1500-1000-500-10	MNIST	5% translation	0.94
MLP:2500-2000-1500-1000-500-10	MNIST	affine + elastic	0.35
MLP committee:2500-2000-1500-1000-500-10	MNIST	affine + elastic	0.31
CNN 20M-40M-60M-80M-100M-120M-150N	MNIST	affine + elastic	0.35

Source from: http://people.idsia.ch/~ciresan/results.htm

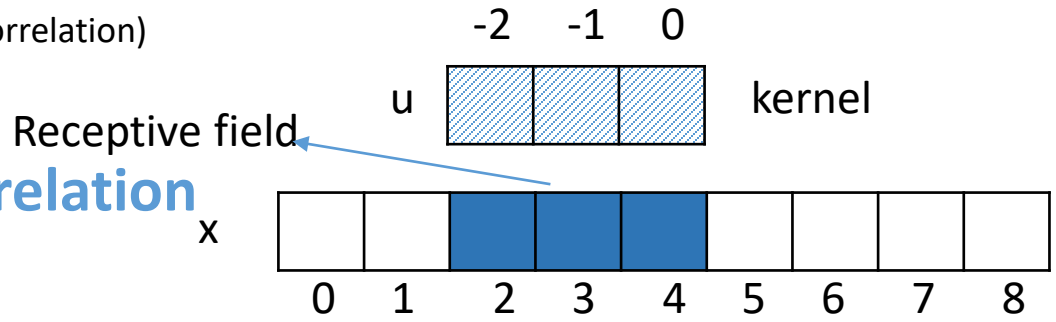
Convolution

- Location estimation
  - At time t, the server will receive the location x_t from a sensor, which may include some noise.
  - The location estimation relies on recent locations more than historical locations with weights $w_0 \geq w_1 \geq, \ldots w_{k-1}$, with their sum=1.
  - The estimated location
    - $y_t = w_0 \times x_t + w_1 \times x_{t-1} + \ldots + w_{k-1} \times x_{t-(k-1)}$
    - $= \sum_{i=1}^{k} w_i \times x_{t-i}$
    - $= \sum_{i=-\infty}^{\infty} w_i \times x_{t-i}$, $(w_i = 0 \; for \; i < 0 \; or \; i \geq k)$

Convolution and Cross-Correlation

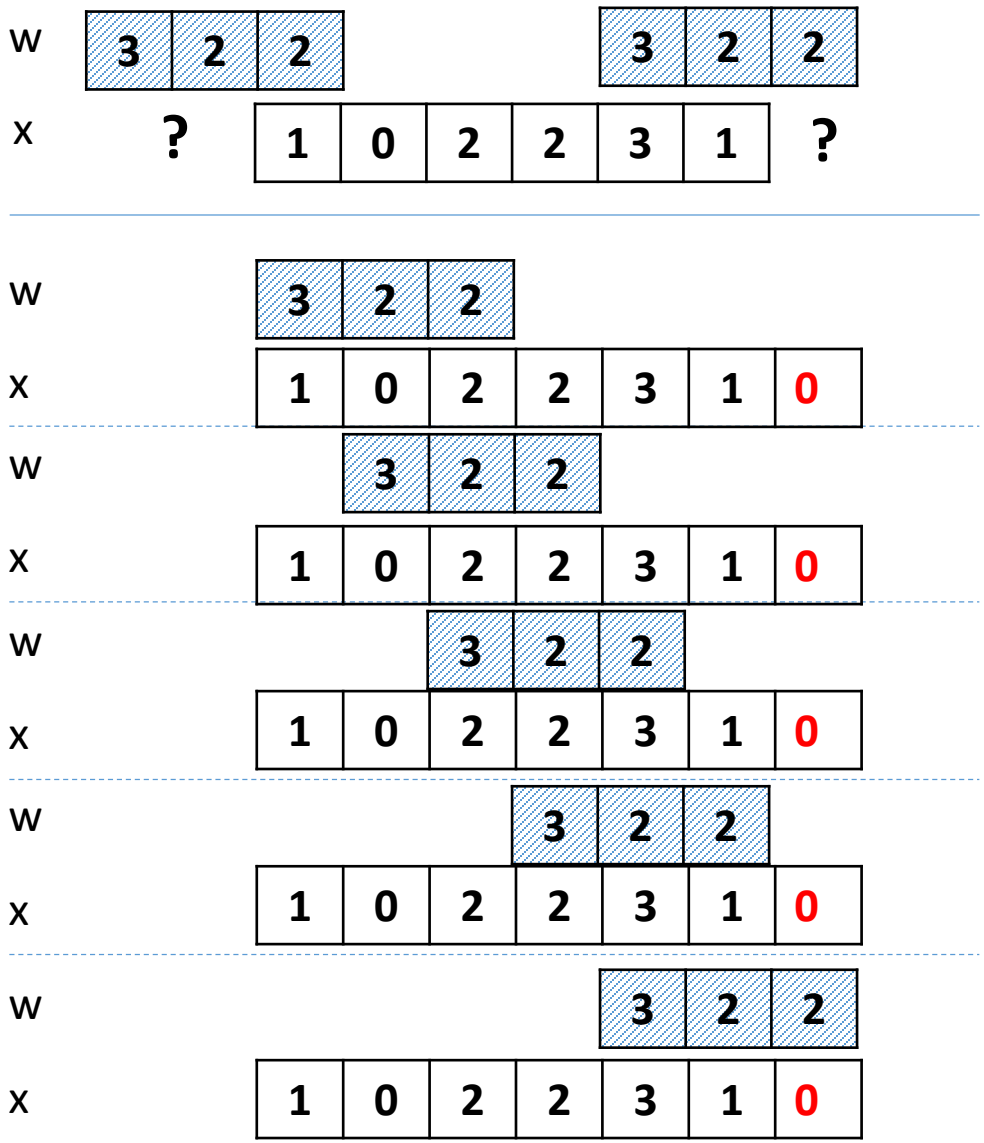


- Formal definition
  - $y_t = \sum_{i=-\infty}^{\infty} w_i \times x_{t-i}$
  - w is called kernel; the parameters to be trained; length k
  - x is the input; length l
  - the input area, i.e. t, t-1, … t-(k-1) is called the receptive field for CNN
  - y_t is the output feature; length o
- Cross-correlation (https://en.wikipedia.org/wiki/Cross-correlation)
  - $y_t = \sum_{i=-\infty}^{\infty} u_i \times x_{t+i}, u_i = w_{-i}$
  - **In CNN, convolution refers to cross-correlation**



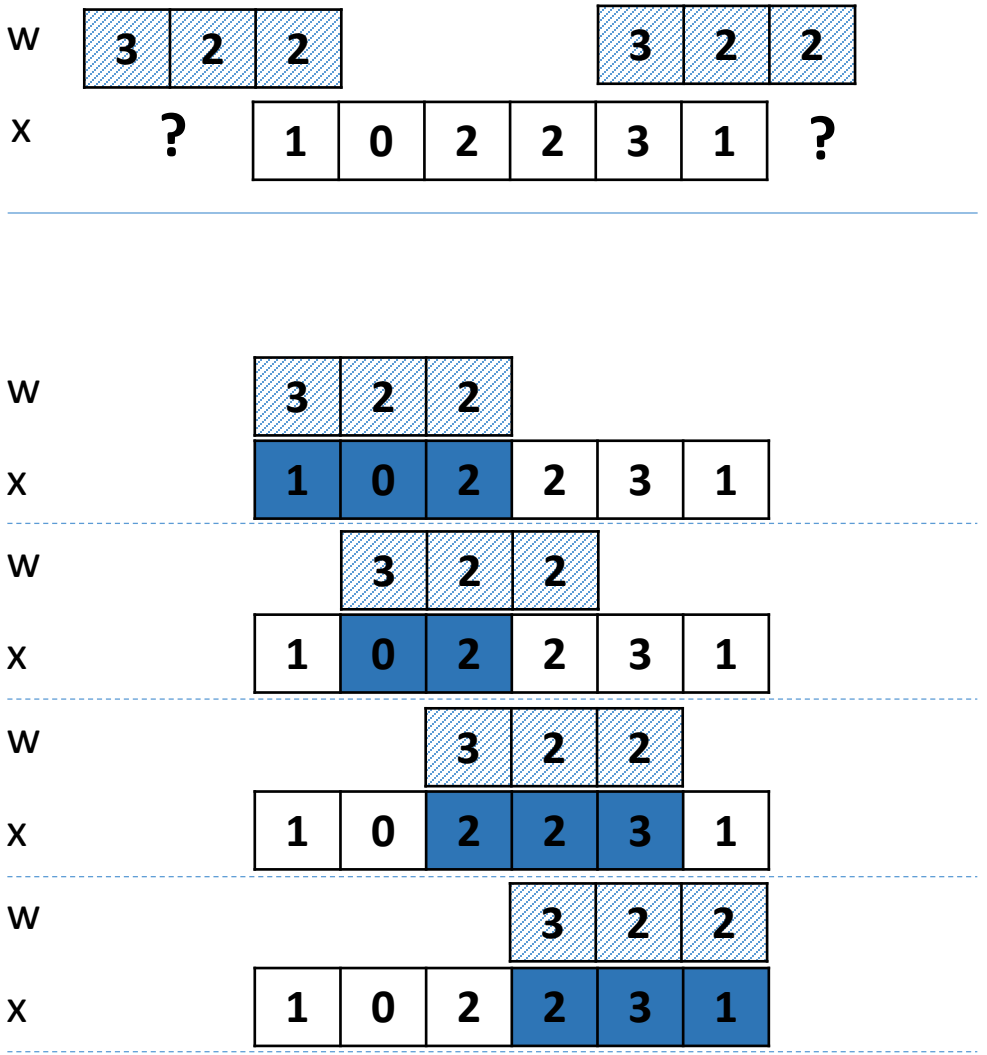
Padding (why?)

- Manual padding (p)
  - Kernel size/length: k
  - Input length: l
  - # outputs $o = l + p - k + 1$
  - Output feature values for $p = 1$
    - 3x1+2x0+2x2=7
    - 3x0+2x2+2x2=8
    - 3x2+2x2+2x3=16
    - 3x2+2x2+2x1=14
    - 3x3+2x1+2x0=11
  - Torch, PyTorch, Caffe, SINGA



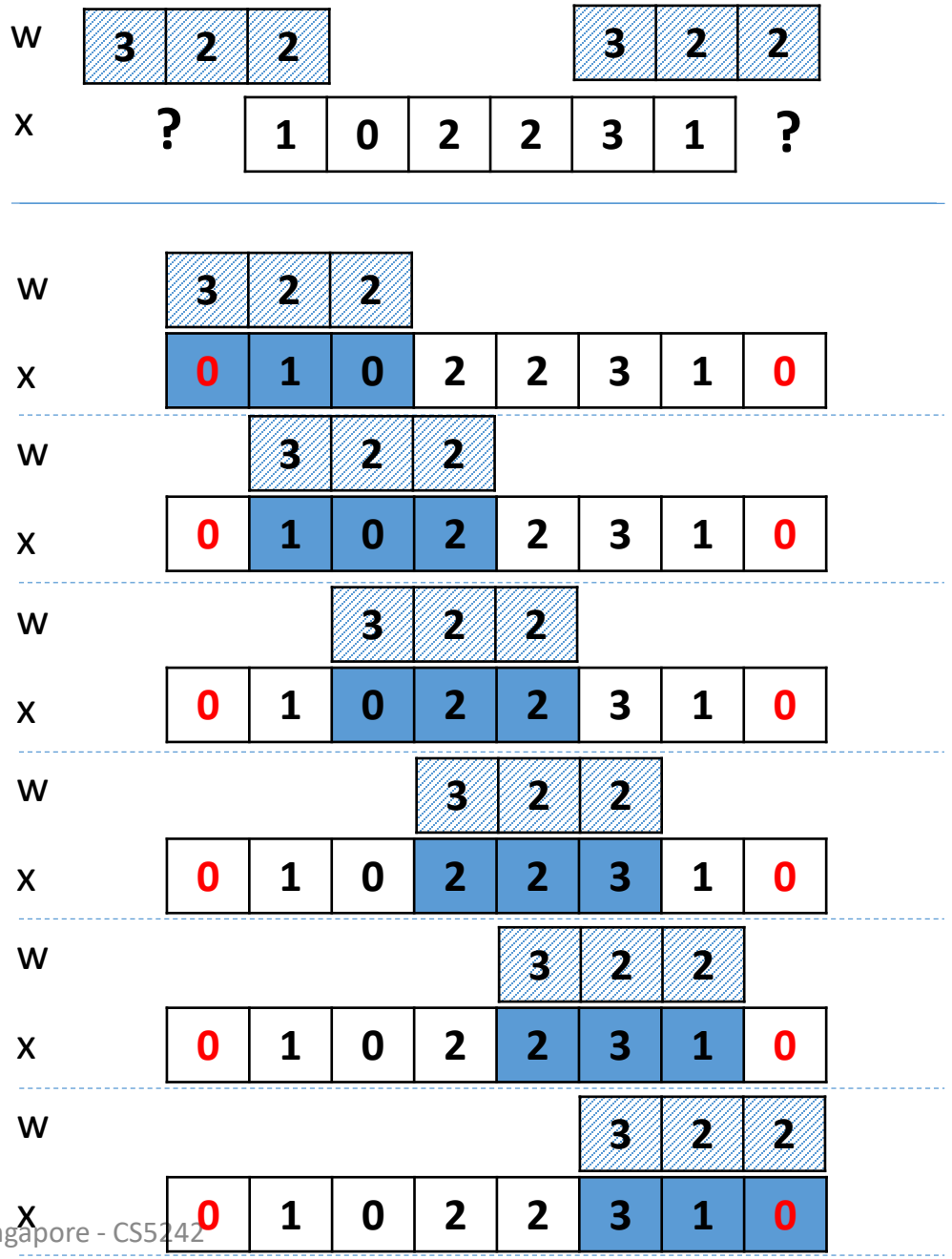
Padding

- Valid/No padding ($p = 0$)
  - # inputs denoted as l
  - # outputs $o = l - k + 1$=6-3+1=4
  - Output feature values
    - 3x1+2x0+2x2=7
    - 3x0+2x2+2x2=8
    - 3x2+2x2+2x3=16
    - 3x2+2x2+2x1=14
  - Outputs become shorter
  - TensorFlow, Keras



Padding

- Same padding (p?)
  - $l + p - k + 1 = l$
  - $p = k - 1$
    - Left padding = $\lfloor p/2 \rfloor$
    - Right padding = $\lceil p/2 \rceil$
  - Output values
    - 3x0+2x1+2x0=2
    - 3x1+2x0+2x2=7
    - 3x0+2x2+2x2=8
    - 3x2+2x2+2x3=16
    - 3x2+2x3+2x1=14
    - 3x3+2x1+2x0=11
  - TensorFlow, Keras



Question

- Given the input length, kernel size, and padding size, what is the output length?
  - qn1: $l = 8, k = 3, p = 2, o =?$
  - qn2: $l = 224, k = 5, p = 1, o =?$

- Given the input length, kernel size and padding type, what is the padding size and output length?
  - $l = 224, k = 5,$
  - qn3: Valid, $p =?$
  - qn4: Valid, $o =?$
  - qn5: Same, $p =?$
  - qn6: Same, $o =?$

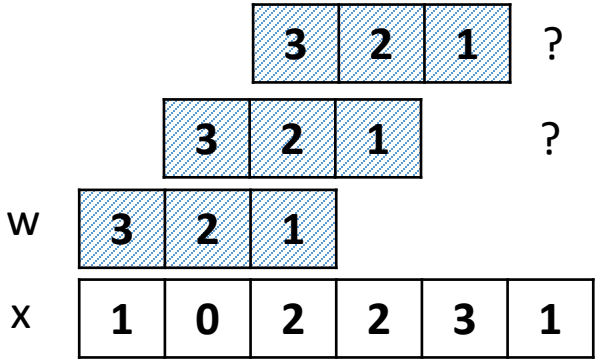
Question

- Given the input length, kernel size, and padding size, what is the output length?
  - qn1: $l = 8, k = 3, p = 2, o = 8$
  - qn2: $l = 224, k = 5, p = 1, o = 221$

- Given the input length, kernel size and padding type, what is the padding size and output length?
  - $l = 224, k = 5,$
  - qn3: Valid, $p = 0$
  - qn4: Valid, $o = 220$
  - qn5: Same, $p = 4$
  - qn6: Same, $o = 224$

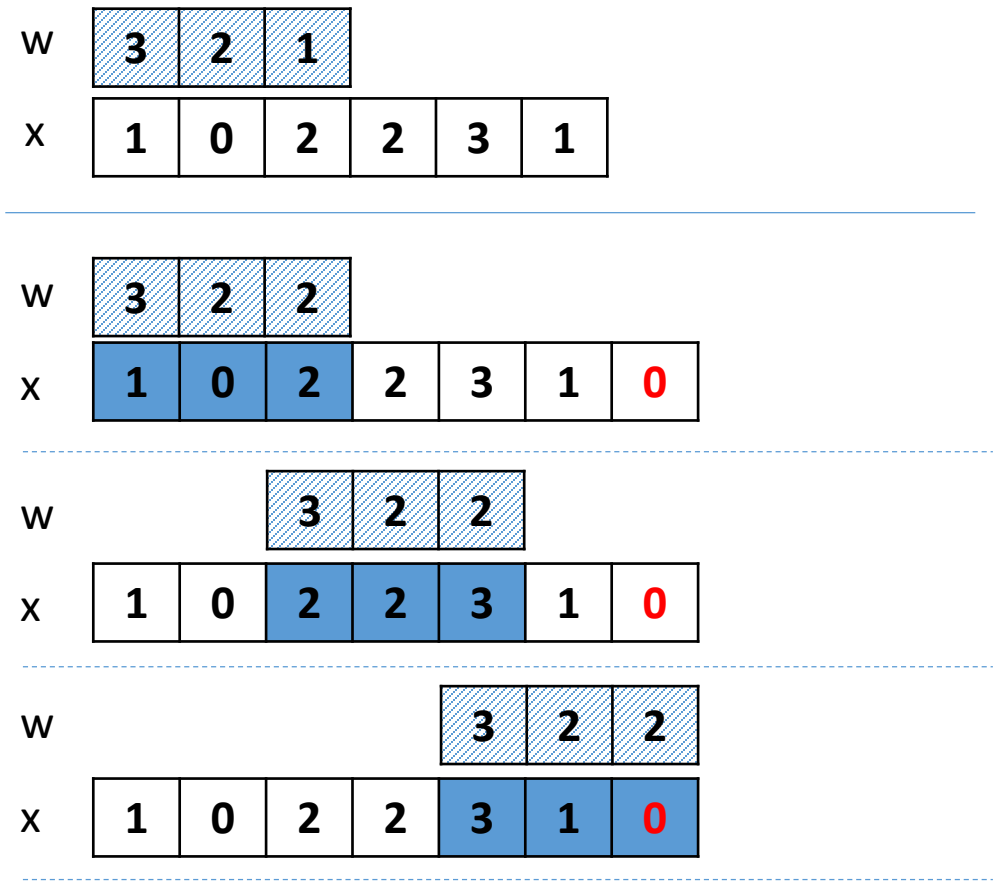
Stride (Why?)

- How many steps to move towards the next receptive field
  - s=1, every receptive field is considered -> many outputs
  - s>1, some receptive fields are skipped.
    - Miss some (redundant) information
    - Faster
    - Fewer outputs



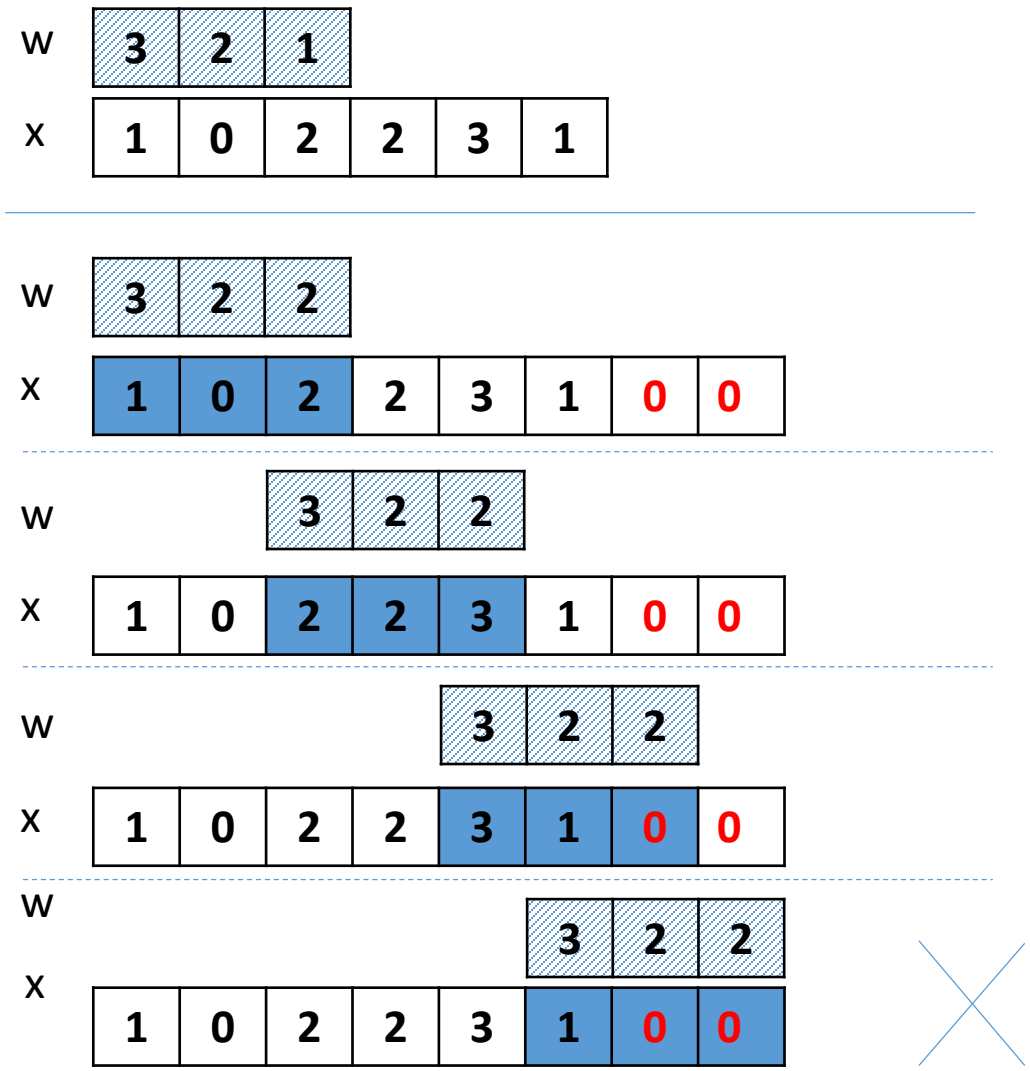
Stride

- Exact matching
  - With padding p (=1)
  - $o = \left\lfloor \frac{l+p-k}{s} \right\rfloor + 1$
  - (6+1-3)/2+1=3



Stride

- Not exact matching
  - With padding p (=2)
  - $o = \left\lfloor \frac{l+p-k}{s} \right\rfloor + 1$
  - (6+2-3)/2+1=3



Question

- Given the input length, kernel size, padding size, stride, what is the output length?
  - qn7: $l = 224, k = 5, p = 1, s = 2, o =$?

Question

- Given the input length, kernel size, padding size, stride, what is the output length?
  - qn7: $l = 224, k = 5, p = 1, s = 2, o = 111$

References and additional readings

- [1] LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). Proceedings of the IEEE. 86 (11): 2278–2324. doi:10.1109/5.726791. Retrieved October 7, 2016.
- http://cs224d.stanford.edu/
- http://cs231n.stanford.edu/
- Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep learning. MIT Press. http://www.deeplearningbook.org. Chapter 9.
- https://www.tensorflow.org/api_guides/python/nn#Notes_on_SAME_Convolution_Padding

Final Projects (40%)

- Projects will be held on Kaggle-in-class
  - Register using **your nus email** (https://inclass.kaggle.com/)
  - No data disclosure, no additional data
  - Each group <=2 students
  - Submission deadline: 04-Nov-2017 06:00 PM, Singapore Time
  - Report deadline: 11-Nov-2017 06:00 PM, Singapore Time
- Each group will be randomly assigned to
  - A computer vision project, https://inclass.kaggle.com/c/cs5242-project-1
  - A natural language processing project, https://inclass.kaggle.com/c/cs5242-project-2
- First task
  - Find a partner register your group information on IVLE (Project -> Final project).

Questions

- nt1 do you know transpose convolution?
- nt2 do you know separable convolution?
- nt3 do you know max pooling?
- nt4 do you know average pooling?
- nt5 do you know alexnet?
- nt6 do you know vgg?
- nt7 do you know resnet?