NUS | Computing
National University of Singapore

Eric Han
eric_han@nus.edu.sg
https://eric-han.com

Computer Science

T09 – 07 Nov 2024

Week 12

CS2109s TG35,36

Content I

Student Feedback on Teaching (SFT)

NUS Student Feedback https://blue.nus.edu.sg/blue/:

- Don't Mix module/grading/project feedback - **feedback only for teaching**.
- Feedback is confidential to university and anonymous to us.
- Feedback is optional but highly encouraged.
- Past student feedback improves teaching; see https://www.eric-han.com/teaching
  - ie. Telegram access, More interactivity.
- Your feedback is important to me, and will be used to improve my teaching.
  - Good > Positive feedback > Encouragement
    - Teaching Awards (nominate)
    - Steer my career path
  - Bad > Negative feedback (nicely pls) > Learning
    - Improvement
    - Better learning experience

Section 1: **ConvNets**

Question 1 [G]

Find the cross-correlation ('convolution' as per CNN), $\mathbf{x} \bigotimes \mathbf{W}$:

$$\mathbf{x} = \begin{bmatrix} 0.1 & 0.2 & 0.1 & 0.1 & 0.0 \\ 0.8 & 0.9 & 1.0 & 1.0 & 0.9 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0.9 & 1.0 & 1.0 & 0.8 & 1.0 \\ 0.0 & 0.1 & 0.1 & 0.2 & 0.0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 \\ -1.0 & -1.0 & -1.0 \end{bmatrix}$$

[@] What is the difference between cross-correlation and convolution? Why are most CNNs implemented as cross-correlation? Find the convolution, $\mathbf{x} * \mathbf{W}$.

Recap

- How to calculate 'convolution' as per CNN?

$$\mathbf{x} \bigotimes \mathbf{W} = \begin{bmatrix} -2.6 & -2.6 & -2.8 \\ -0.2 & 0.1 & 0.1 \\ 2.8 & 2.6 & 2.7 \end{bmatrix}$$

$$\mathbf{x} * \mathbf{W} = \begin{bmatrix} 2.6 & 2.6 & 2.8 \\ 0.2 & -0.1 & -0.1 \\ -2.8 & -2.6 & -2.7 \end{bmatrix}$$

Flip the filter in both dimensions (or rotate 180 degrees) to go between cross-correlation and convolution.

- In this case here the result is the negative of each other, but in general it is **NOT**.

Question 2

- Image input is $H \times W \times C = 224 \times 224 \times 3$
- First layer is Convolutional Layer with $C_1 = 96$ kernels of size 11×11, stride 4×4 without padding

Recap

How to calculate the output of a convolution?

- Image input is $H \times W \times C = 224 \times 224 \times 3$
- First layer is Convolutional Layer with $C_1 = 96$ kernels of size 11×11, stride 4×4 without padding

Recap

How to calculate the output of a convolution?

- Output height $=$ (Input height $+$ padding height top $+$ padding height bottom - kernel height) / (stride height) $+ 1$
- Output width $=$ (Output width $+$ padding width right $+$ padding width left - kernel width) / (stride width) $+ 1$

Question 2

- Image input is $H \times W \times C = 224 \times 224 \times 3$
- First layer is Convolutional Layer with $C_1 = 96$ kernels of size 11×11, stride 4×4 without padding

Recap

How to calculate the output of a convolution?

- Output height = (Input height + padding height top + padding height bottom - kernel height) / (stride height) + 1
- Output width = (Output width + padding width right + padding width left - kernel width) / (stride width) + 1

Answer

$$H_1 = \left\lfloor \frac{H - K + 2P}{S} \right\rfloor + 1 = 54$$

Similarly for $W_1 = 54$; There are 96 filters so, $54 \times 54 \times 96$

Question 3 [G]

Images are often batched B. B can take values such as $8, 16, 32, 64$.

- Comment on the output shape if we feed the large CNN in part (b) with a batch.
- What are the advantages of using a batch of images rather than a single image?
- [@] Impact of large/small batch sizes and how to determine the optimal size?

Images are often batched B. B can take values such as $8, 16, 32, 64$.

- Comment on the output shape if we feed the large CNN in part (b) with a batch.
- What are the advantages of using a batch of images rather than a single image?
- [@] Impact of large/small batch sizes and how to determine the optimal size?

Answer

- $B \times H_1 \times W_1 \times C_1$
- Using a batch of images is computationally efficient and more stable in gradient descent convergence.

Section 2: **CNN Receptive field**

2-layer convolutional neural network configuration:

- **Initial input**: 9×9 matrix
- **Layer 1**:
  - Kernel size: 5×5
  - Stride: 2×2
- **Layer 2**:
  - Kernel size: 2×2
  - Stride: 1×1

Calculate the receptive field sizes of neurons in both the first and second layers.

Recap [@]

1. What is CNN Receptive Field?
2. How to calculate the size?

2-layer convolutional neural network configuration:

- **Initial input**: 9×9 matrix
- **Layer 1**:
  - Kernel size: 5×5
  - Stride: 2×2
- **Layer 2**:
  - Kernel size: 2×2
  - Stride: 1×1

Calculate the receptive field sizes of neurons in both the first and second layers.

Recap [@]

1. What is CNN Receptive Field?
2. How to calculate the size?

Region of the input image that influences a particular neuron in a given layer. Expanding with each additional layer as multiple receptive fields combine.

Answer 1

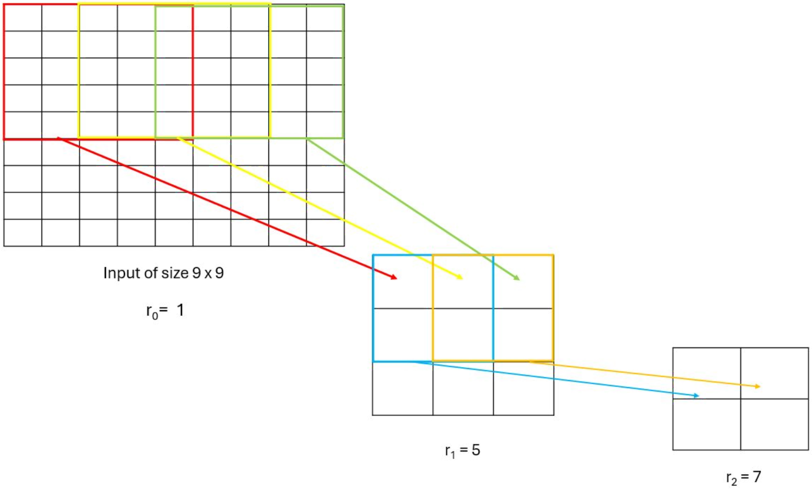

Figure 1: Simulate each layer / Calculate.

How does an increase in the receptive field affect the performance of a Convolutional Neural Network (CNN)?

Answer

1. **Feature Capture**:
   - Larger receptive fields allow deeper neurons to capture more of the input image, enhancing global feature detection and spatial context.
2. **Performance Benefits**:
   - **Pattern Recognition**: Improved recognition of patterns over larger areas.
   - **Contextual Awareness**: Beneficial for tasks needing broader context, like object detection and scene segmentation.

Enhancing performance in complex visual analysis.

Section 3: **RNN Design**

Identify the type of RNN model, Input/Output and examples required for the task:

1. Image Captioning
2. Stock Market Prediction
3. Language Translation

Recap

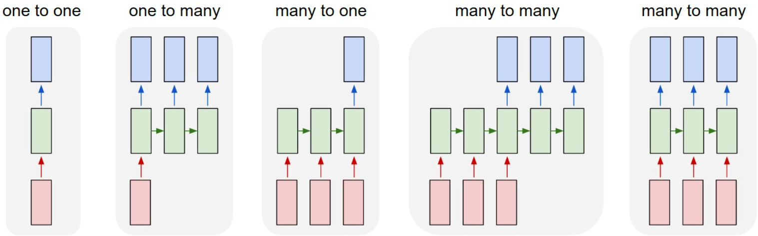

Figure 2: Rectangle is a vector and arrows represent matrix multiply; Input - red, output - blue and green - RNN's state. Taken from https://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Answer

1. One-to-many model
   - Input: One image.
   - Output: Multiple words as captions.
2. Many-to-one model
   - Input: Time series data of stock prices.
   - Output: Likelihood of price increase.
3. Many-to-many model
   - Input: Many words / code of language A.
   - Output: Many words / code of language B.

Examples:

1. Music generation
2. Text sentiment analysis
3. Phonetic transcription

Section 4: **CNN vs RNN**

Question 3 [G]

1. Performing sentiment analysis on Covid-19 posts on X. Explain what characteristics of RNN make it a standard model for sentiment analysis and which RNN model you want to use to tackle this problem.
2. Examine CNNs for sentiment analysis? Explain why or why not.
3. Examine RNNs for image processing? Explain why or why not.

Recap

- What are CNNs good at?
- What are RNNs good at?

Answer

1. RNN is the method for dealing with sequential input; Many/One RNN - Input: Sentence. Output: Sentiment(+/-).
2. Sentiment analysis strongly relies on context of the whole sentence; CNN convolution need many layers to detect higher level features to capture context.
   - I like durian
   - I do not like durain
   - I do not do not like durain
3. Window as a token and we can slide it across to generate the input.
   - https://karpathy.github.io/2015/05/21/rnn-effectiveness/

CNNs are very good at capture spatial structure - locality, ie. pixels near to each other are useful together - to recognize eye and layers above to compose the features.

Bonus Qn

Tasks

1. Implement `correlate2d(x,W)` and `convolve2d(x,W)` using `numpy`.
2. Calculate the values for question 1.
3. Compare it with `scipy.correlate2d(x,W, mode='valid')` and `scipy.convolve2d(x,W, mode='valid')`.

Buddy Attendance Taking

1. [@] and Bonus declaration is to be done here; You should show bonus to Eric.
2. Attempted tutorial should come with proof (sketches, workings etc...)
3. Random checks may be conducted.
4. Guest student should come and inform me.

Figure 3: Buddy Attendance: https://forms.gle/q5Secb3dHshmXNXd7

References I