Lab 5: Convolution

10/14/2022

Linear Shift Invariant Systems

- *Linear* systems all obey the principles of homogeneity and superposition
  - Linear operators *respect certain relationships between input elements, meaning that those relationships are preserved in the output*
    - L(a*x) = a*L(x) → respects/preserves scaling
    - L(x + y) = L(x) + L(y) → respects/preserves addition
- *Shift Invariant* systems respect a different relationship between input elements...“shifting”
  - L(shift(x)) = shift(L(x))
  - What the heck is a shift? Shifts change the position of elements in the input vector
    - shift(x[i]) = x[i + n] for some fixed value of n
    - Shift with n = 2:

$$\text{shift}\left(x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

LSI Systems as Matrices

- Linear System → We can describe the transformation with a matrix
  - First column is output of system in response to first basis/impulse vector
  - Subsequent columns are outputs in response to the other basis vectors, but these inputs are by definition shifted versions of the same vector → for LSI systems we only need to test the output to one input.

$$L\left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} k1 \\ k2 \\ 0 \\ 0 \end{pmatrix} \Longrightarrow M = \begin{pmatrix} k1 & 0 & 0 & 0 \\ k2 & k1 & 0 & 0 \\ 0 & k2 & k1 & 0 \\ 0 & 0 & k2 & k1 \end{pmatrix}$$

LSI Systems as Matrices

- LSI system/convolution matrices also have a nice interpretation from the “rows,” perspective: the output at a given index is the inner product of part of the input with a reversed order copy of the kernel

$$\begin{pmatrix} S\_1 \\ S\_2 \\ S\_3 \\ s\_4 \end{pmatrix}$$

$$\begin{pmatrix} k1 & 0 & 0 & 0 \\ \boxed{k2 \;\; k1} & 0 & 0 \\ 0 & \boxed{k2 \;\; k1} & 0 \\ 0 & 0 & \boxed{k2 \;\; k1} \\ 0 & 0 & 0 & k2 \end{pmatrix} \begin{pmatrix} out\_1 \\ out\_2 \\ out\_3 \\ out\_4 \\ out\_5 \end{pmatrix}$$

- So, in the usual way that we use the dot product to be a similarity measure, we can say that the output of an LSI (or equivalently, *the convolution of the input signal with the characteristic kernel*) at a given position, is a measure of how similar part of the input is to the kernel
- This is the “sliding dot product,” interpretation.

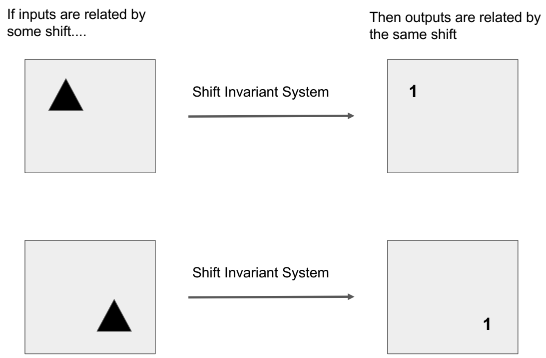
LSI Systems as Matrices

- Terminology:
  - The matrix that represents a LSI system is called a ***“convolution matrix.”***
  - The response to the first impulse vector, or the first column of the convolution matrix, is called the ***“impulse response,”*** or ***“kernel,”*** of the operator.
  - We will often use the phraseology: “the output of an LSI system is the convolution of the input signal with the kernel.”

Why Bother?

- When there is meaningful structure *between elements of the input signal*, describing the input-output function as an LSI system (or alternatively, describing the output as *the convolution of the input with some fixed kernel*) ensures that same structure is preserved in the output signal.
  - What kinds of signals have the type of structure that convolution respects?
    - Time-series data, images, time-series of images, etc.
- Practical Benefits
  - Reduced degrees of freedom (overfitting/model complexity reduction)

$$M = \begin{pmatrix} k1 & 0 & 0 & 0 \\ k2 & k1 & 0 & 0 \\ 0 & k2 & k1 & 0 \\ 0 & 0 & k2 & k1 \end{pmatrix}$$

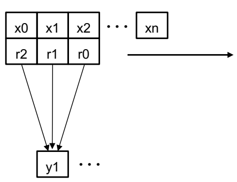


Finite Dimensional Signals: Boundary Conditions

- Recall our definition of shift invariance
    - shift(x[i]) = x[i + n] for some fixed value of n
    - Or in general for finite LSI systems the definition of convolution is
    - $y[n] = \sum_m x[m]r[n-m] = \sum_k x[n-k]r[k]$ What happens when k > n?
- For finite dimensional inputs, this definition cannot make sense everywhere…
    - Defining "shift invariance," in these trouble spots ⇔ deciding on a set of boundary conditions
    - 3 most common are
        - Zero Padding ('same', 'valid', 'all')
        - Circular Boundary Conditions
        - Mirrored Boundary Conditions

Zero Padding: 'full', 'same', and 'valid'

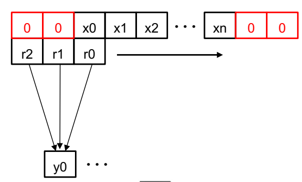
- 'valid' option: only consider the outputs where all terms in definition are known



- Only consider outputs where the kernel is entirely within the signal

Zero Padding: 'full', 'same', and 'valid'

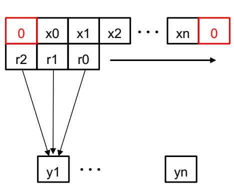
- 'full' option: use every possible datapoint, assuming the unknown values are zero.



$$\sum_k x[n-k]r[k]$$

Zero Padding: 'full', 'same', and 'valid'

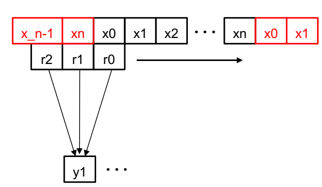
- 'same' option: zero pad on each side by an amount that makes the output size equal to the input size.



- It is often convenient for either theoretical or practical reason for the output to have the same dimensionality as the input.

Circular/Periodic Boundary Conditions

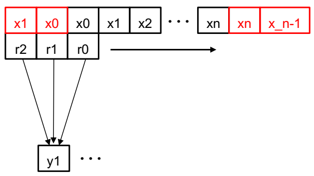
- Rather than assume unknown values are zero, assume values "wrap around"



- Similar to full but with different assumption about missing values

Mirrored Boundary Conditions

- Padded values are the “reflection,” or values near the boundary



- Similar to full but with different assumption about missing values

2-D Extensions

- Everything discussed so far has a direct extension to the two dimensional case:
  - Shift invariance: LSI systems are invariant to shifts in both directions (two dimensional data structures, i.e. images, can be shifted in two directions independently).
  - Padding: zeros surround data structure on both sides
  - Other boundary conditions: can be independently applied to each dimension
    - Visual: a 360 degree panoramic photo might be taken to have circular boundary conditions along one dimension (horizontal), but use zero padding along the other