Convolutions in 1D
What is a convolutional neural network?

Understanding Convolutional Neural Networks for NLP

When we hear about Convolutional Neural Network (CNNs), we typically think of Computer Vision. CNNs were responsible for major breakthroughs in Image Classification and are the core of most Computer Vision systems today, from Facebook’s automated photo tagging to self-driving cars.

Patterns; with more than one, you could look for patterns of patterns. Take the case of image recognition, which tends to rely on a contraption called a “convolutional neural net.” (These were elaborated in a seminal 1998 paper whose lead author, a Frenchman named Yann LeCun, did his postdoctoral research in Toronto under Hinton and now directs a huge A.I. endeavor at

Next, Zweig and co optimized their own deep-learning systems based on convolutional neural networks with varying number of layers, each of which processes a different aspect of speech. They then used the

digital input and send output to other nodes. Layers upon layers of these nodes make up so-called convolutional neural networks, which, with sufficient training data, have become better and better at identifying images.

Inside a Convolutional Neural Network

But how does this filtering work? The secret is in the addition of two new types of layers: convolutional and pooling layers. We’ll break the process down below, using the example of a network designed to do just one thing: determine whether a picture contains a grandma or not.
Convolution

\[(f \ast g)(t) = \int_{-\infty}^{\infty} f(t')g(t-t') \, dt'\]

Is Convolution that convoluted?
Let us start with discrete 1D signals!

For example, my systolic blood pressure measured with a wearable over 24 hours on 1/16/2016:
Interested in the largest drop?

We are moving a sliding window and at every position we take the scalar product between the mask and the signal.
Let us visualize it while in action:

We will call this operation correlation between two discrete signals and we can write it as

\[ y[n] = \sum_{k=1}^{N} s[k] h[k - n] \]
Convolution differs from correlation by a reflection:

We take the “mask” $d[k]$ and reflect it to $d[-k]$ and then we shift to $d[-(k-n)]=d[n-k]$. 

$$y[n] = \sum_{k=1}^{N} s[k] h[n - k]$$
The informal term for the convolution mask is “filter mask”, its values are called “filter weights” or convolution weights.

Let us do another operation on the systolic pressure signal.

Taking the local average over a window of two hours.

The averaging mask would look like $[\frac{1}{4} \frac{1}{4} \frac{1}{4} \frac{1}{4}]$

Does the convolution differ from correlation?
Let us move to the Academy of Music!
Sound is reflected creating multiple echos
Now let us take the song $s[n]$ and convolve it with the pistol $h[n]$

$$y[n] = \sum_{k=1}^{N} s[k] h[n - k]$$
Linear Shift-Invariant Systems
In the last lecture the song $s[n]$ and convolve it with the pistol $h[n]$

$$y[n] = \sum_{k=1}^{N} s[k] h[n - k]$$
Consider it as a system with input and output. In the previous examples, input might have been
- Audio like a song
- Blood pressure measurements
- In general any 1D signal

The systems’ view
Linear System

• We will say that a system is linear if

\[ f_1(t) \rightarrow T \rightarrow g_1(t) \]

then

\[ af_1(t) + bf_2(t) \rightarrow T \rightarrow ag_1(t) + bg_2(t) \]

\[ f_2(t) \rightarrow T \rightarrow g_2(t) \]
Example

\[ T\{f\}(t) = f(t) - f(t-1) \] is linear:

\[ g_1(t) = f_1(t) - f_1(t-1) \]
\[ g_2(t) = f_2(t) - f_2(t-1) \]

\[ T\{a f_1 + b f_2\}(t) = [a f_1(t) + b f_2(t)] - [a f_1(t-1) - b f_2(t-1)] \]
\[ = a(f_1(t) - f_1(t-1)) + b(f_2(t) - f_2(t-1)) \]
\[ = aT\{f_1\}(t) + bT\{f_2\}(t) \]
What is non-linear?

\[ g(t) = \max(f(t), f(t-1), f(t-2)) \]
Shift-invariant (or time-invariant) system

\[ f(t) \rightarrow T \rightarrow g(t) \quad \text{then} \quad f(t - t_0) \rightarrow T \rightarrow g(t - t_0) \]

Examples:

- \( T\{f\}(t) = f(t) - f(t - 1) \) is shift-invariant.

\[
T\{f(t - t_0)\} = f(t - t_0) - f(t - t_0 - 1) \\
= g(t - t_0)
\]

- \( g(t) = tf(t) \) is not shift-invariant.
Is there a formula for describing linear shift-invariant systems?

Yes, the convolution! Discrete or continuous

\[ g(t) = \int_{-\infty}^{\infty} f(t')h(t-t')dt' \quad g[n] = \sum_{k=-\infty}^{\infty} f[k]h[n-k] \]
What is a filter?
Last lecture’s main result: Linear shift-invariant systems can be written as convolutions!

\[ g(t) = \int_{-\infty}^{+\infty} f(t')h(t-t')dt' = f(t) \ast h(t) \]

But what exactly is this \( h(t) \) or \( h[n] \) that we call filter?
Box function

\[
\text{rect}(t) = \begin{cases} 
1, & |t| \leq 1/2 \\
0, & \text{otherwise}
\end{cases}
\]

\[
\text{rect}(t/a) = \begin{cases} 
1/a, & |t| \leq a/2 \\
0, & \text{otherwise}
\end{cases}
\]
Dirac function

\[ \delta(t) = \lim_{a \to 0} \frac{1}{a} \text{rect}(t/a) \]

It follows from definition of the Dirac function that

\[ \delta(t) = 0 \quad \text{for all } t \neq 0 \]

and

\[ \int_{-\infty}^{+\infty} \delta(t) dt = 1 \]
Absorption property

\[ \int_{-\infty}^{+\infty} \delta(t) f(t) dt = f(0) \]

or in a more general form:

\[ \int_{-\infty}^{+\infty} \delta(t) f(t_0) dt = f(t_0) \]
What happens when the input to an LSI is a Dirac?

A filter $h(t)$ or $h[n]$ is the response of the system to the Dirac impulse.
Example. Let an LSI system with impulse response defined as

\[ h(t) = \frac{1}{2}(\delta(t + 1) - \delta(t - 1)) \]

\[
g(t) = \int_{-\infty}^{+\infty} f(t-u)h(u)du
\]

\[
= \int_{-\infty}^{+\infty} f(t-u)\left(\frac{1}{2}(\delta(u + 1) - \delta(u - 1))\right)du
\]

\[
= \frac{1}{2} \int_{-\infty}^{+\infty} f(t-u)\delta(u + 1)du - \frac{1}{2} \int_{-\infty}^{+\infty} f(t-u)\delta(u - 1)du
\]

\[
= \frac{1}{2}(f(t + 1) - f(t - 1))
\]
Dirac description of sampling

• What do we get discrete signals out of analog?
Sampling is multiplication with the comb

Sampling is a multiplication of the signal by the comb function $\sum_{n=-\infty}^{\infty} \delta(t - nT)$, where $T$ is the sampling interval.

After sampling, we forget about the $T$: we just obtain a sequence of numbers, that we note $f_s[k]$. 

$W_T(t)$ 

$S(t-nT)$