Deep Learning

Sequential Models

Edgar F. Roman-Rangel.
edgar.roman@itam.mx

Digital Systems Department.
Instituto Tecnológico Autónomo de México, ITAM.

March 5th, 2021.

Outline

Sequential data

Recursion layers

Definition

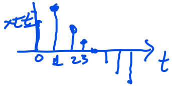
Data whose samples hold some sort of temporal relationship. E.g.,

- Voltage.

$n =$ indice del tiempo ; $x[n] =$ señal

$t =$ $x[t] =$

- Voice.



- Time series: stock price, pollution measurements.

$x[t] = [x_1 \ x_2 \ \cdots \ x_n]$

- Text.
- DNA.
- Movies (combined spatial-time relationships).

Order might be important:

"I work at Google" vs. "I google at work".

Mathematically

Each data point is a time series, of the form:

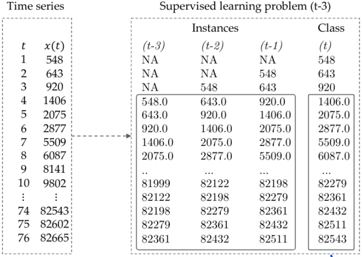
$$\mathbf{x} = [x(t=1), x(t=2), \ldots, x(t=T)],$$

where each time sample $x(t)$ might be multivariate,

$$x(t) = [x_1, x_2, x_M].$$

This same format might apply to inputs, outputs, or both.

Examples



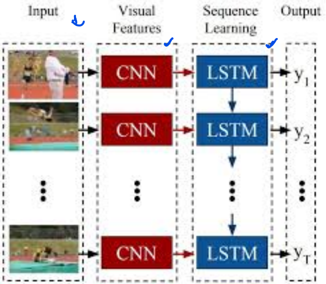
Time series

t	$x(t)$
1	548
2	643
3	920
4	1406
5	2075
6	2877
7	5509
8	6087
9	8141
10	9802
⋮	⋮
74	82543
75	82602
76	82665

Supervised learning problem (t-3)

Instances			Class
(t-3)	*(t-2)*	*(t-1)*	*(t)*
NA	NA	NA	548
NA	NA	548	643
NA	548	643	920
548.0	643.0	920.0	1406.0
643.0	920.0	1406.0	2075.0
920.0	1406.0	2075.0	2877.0
1406.0	2075.0	2877.0	5509.0
2075.0	2877.0	5509.0	6087.0
..	...	...	...
81999	82122	82198	82279
82122	82198	82279	82361
82198	82279	82361	82432
82279	82361	82432	82511
82361	82432	82511	82543

x y

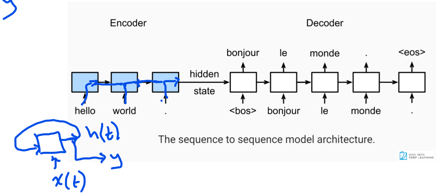


x

"I love this movie.
I've seen it many times
and it's still awesome." →

y

"This movie is bad.
I don't like it it all.
It's terrible." →



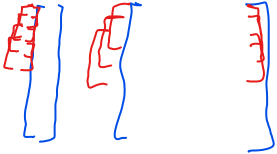
The sequence to sequence model architecture.

For all non-numerical data, we must first find a numerical representation.

Processing

Consider a phrase as a matrix of shape $[\overset{T}{\cancel{M}} \times N]$, with $\overset{T}{\cancel{M}}$ rows (words), each represented by a vector of length N.

1D convolution



$$\mathbf{y}(t) = \sum_{k=t-n}^{t} \mathbf{w}_k^T \mathbf{x}(k).$$

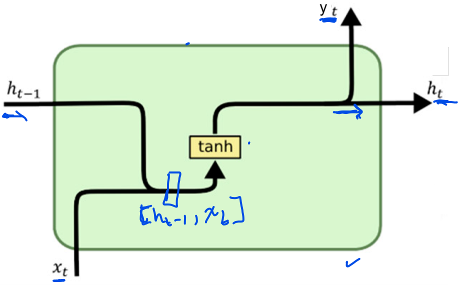
Apply convolution only on the word's axis, i.e., combine word representations.

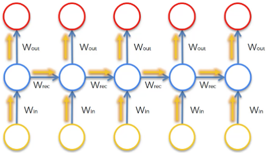
Recursion

$$\mathbf{y}(t) = \underset{\uparrow}{\mathbf{w}_0^T} \underline{\mathbf{x}(t)} + \sum_{\underline{k=t-n}}^{t} \mathbf{w}_k^T \underline{\mathbf{y}(k)}.$$

Apply recursion on the word's axis, i.e., accumulate word representations.

Recursion layer



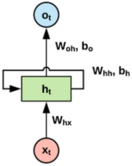


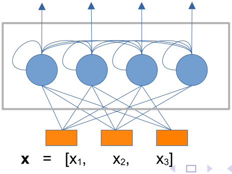
A hidden layer is nothing but a layer of perceptrons.

Number of parameters

$$\mathbf{h}_t = \sigma\left([\mathbf{w}_h, \mathbf{w}_x] \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix}\right).$$

- $N = |\mathbf{w}_h| + |\mathbf{w}_x| + |b|$, where $|\cdot|$ is cardinality. $\mathbf{w}_h, \mathbf{w}_x, b$, are matrices of recursion, input, and bias weights.
- $|\mathbf{w}_h| = |p_h|^2$: number of hidden perceptrons squared.
- $|\mathbf{w}_x| = |p_h| \times |\mathbf{x}|$: number of hidden perceptrons times number of features in the input vector.
- Optionally: add the number of weights in the output $|\mathbf{w}_o|$.





$|\mathbf{w}_h|$ = 4x4 = 16

$|b|$ = 4

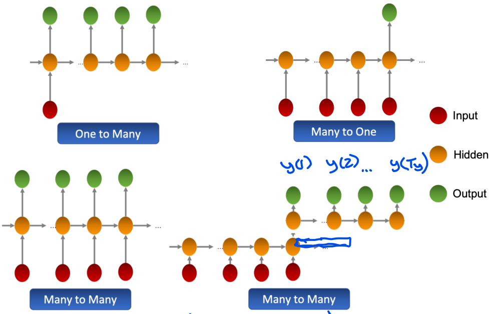
$|\mathbf{w}_x|$ = 3x4 = 12

N = 32

Recursion types

$y(t)$; $t=1...T$

$x(t)$; $t=1$



$x(1)$ $x(2)$ ... $x(T_x)$

- ▶ One-to-many: x: image, y: text description.
- ▶ Many-to-one: x: text phrase, y: sentiment class.
- ▶ Many-to-many: x: climate measurement, y climate forecast.
- ▶ Many-to-many: x: source language, y translation.

Some characteristics

- Can model a sequence of data.
- Samples are assumed to depend on previous ones.
- Vanishing gradient through time.
- Truncated backprop (in time).
- Short-term memory.

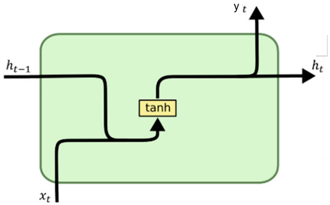
Outline

Sequential data

Recursion layers

RNN

Recurrent Neural Network: Rumelhart, 1986.

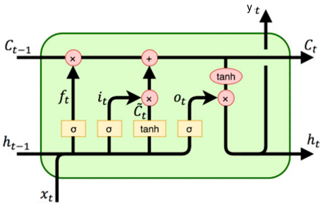


- $y_t = h_t = \sigma(\mathbf{w}[\mathbf{x}_t, \mathbf{h}_{t-1}])$.

$$\frac{\partial E}{w_i(t)} = \frac{\partial E}{\partial \sigma_T()} \cdot \frac{\partial \sigma_T()}{\partial w_{i,T}} \cdot \frac{\partial w_{i,T}}{\partial \sigma_{T-1}()} \cdot \frac{\partial \sigma_{T-1}()}{\partial w_{i,T-1}} \cdots$$

LSTM

Long Short-term Memory: Hochreiter and Schmidhuber, 1997.



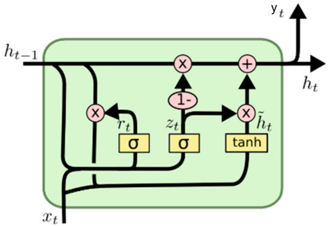
- $y_t = h_t = o_t \odot \tanh(C_t)$,
- $o_t = \sigma(\mathbf{w}_o[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $f_t = \sigma(\mathbf{w}_f[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $i_t = \sigma(\mathbf{w}_i[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $\tilde{C}_t = \tanh(\mathbf{w}_C[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$.

Biases are omitted for simplicity.
Also, all variables are vectors.

Think of each section as a gate: input gate, forget gate, output gate, and cell (state) gate.

GRU

Gated Recurrent Unit: Cho et al., 2014.



- $z_t = \sigma(\mathbf{w}_z[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $r_t = \sigma(\mathbf{w}_r[\mathbf{x}_t, \mathbf{h}_{t-1}])$,
- $\tilde{h}_t = \tanh(\mathbf{w}_h[\mathbf{x}_t, r_t \times \mathbf{h}_{t-1}])$,
- $y_t = h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t$.

Bidirectional RNN's (BRNN)

Schuster and Paliwal, 1997. "Bidirectional Recurrent Neural Networks".

Looking up ahead in the future, might provide richer information for some temporal signals, as opposed to only looking at the past. E.g.,

He said, "Teddy bears are on sale!"
vs.
He said "Teddy Roosevelt was a great president!"

- Getting information from the future.
- Two RNN's, forward and backward processing of the signal.
- Useful for non-causal signals.

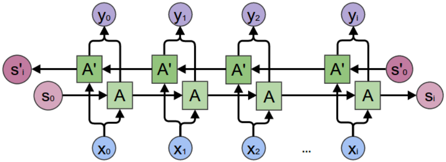
BRNN

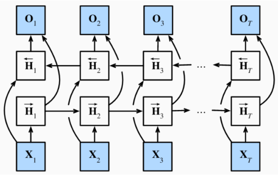
Change

$$y(t) = f(x(t), x(t-1), \dots, x(0), y(t-1), \dots, y(0)),$$

into

$$y(t) = f(\mathbf{x}, y(0), \dots, y(t-1), y(t+1), \dots, y(T)).$$

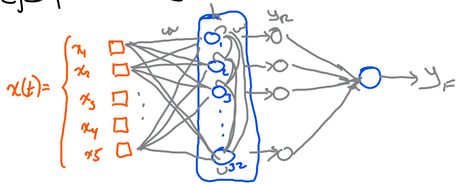




- $\overrightarrow{\mathbf{h}}(t) = \sigma(\mathbf{x}(t)\mathbf{w}_x^{(f)} + \overrightarrow{\mathbf{h}}(t-1)\mathbf{w}_h^{(f)}),$
- $\overleftarrow{\mathbf{h}}(t) = \sigma(\mathbf{x}(t)\mathbf{w}_x^{(b)} + \overleftarrow{\mathbf{h}}(t+1)\mathbf{w}_h^{(b)}),$
- $y(t) = \sigma([\overrightarrow{\mathbf{h}}(t), \overleftarrow{\mathbf{h}}(t)]\mathbf{w}_y).$

Q&A

Ejemplo del código



Thank you!

`edgar.roman@itam.mx`