



# Tarea 2

## Recursive Neural Network

Autor: Matías Meneses C.  
Profesor: Alex Bergel  
11 de octubre de 2017  
Santiago, Chile.

## Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Software utilizado . . . . .	3
<b>2. Preparación de Input</b>	<b>4</b>
<b>3. Construcción de Red Neuronal</b>	<b>4</b>
3.1. Forward step . . . . .	5
3.2. Backpropagation . . . . .	5
3.3. Update . . . . .	5
<b>4. Experimentos</b>	<b>5</b>
<b>5. Resultados</b>	<b>6</b>
5.1. Sample con 1 epoch . . . . .	6
5.2. Sample con 100 epoch . . . . .	7
5.3. Sample con 200 epoch . . . . .	8
5.4. Samples con 450 epoch . . . . .	8
5.4.1. Método estándar . . . . .	8
5.5. Método de máxima probabilidad . . . . .	8
5.6. Método híbrido . . . . .	9
<b>6. Análisis de Resultados</b>	<b>9</b>
<b>7. Conclusión</b>	<b>9</b>

## 1. Introducción

El objetivo de esta tarea es realizar una red neuronal avanzada respecto a la implementada en la primera tarea del curso.

Se decidió realizar una Recursive Neural Network (RNN) que sea capaz de generar poemas, entrenándola previamente con poemas de Pablo Neruda.

Se realizaron diversas pruebas para analizar la efectividad de la predicción, la sintáxis y semántica del texto generado. Las pruebas fueron realizadas variando los parámetros de la red para comprobar su impacto.

La inspiración y fuente de información para la implementación fue extraída de estos dos portales de internet:

- The Unreasonable Effectiveness of Recurrent Neural Networks
- Recurrent Neural Networks Tutorial

En este informe se explican brevemente las partes de una RNN.

### 1.1. Software utilizado

Se utilizó el lenguaje de programación Python para realizar la implementación de la RNN, en conjunto con la librería numpy para realizar el cálculo matricial.

El programa resultante fue ejecutado en una máquina con Linux (Fedora 26), intel core i5 con 8 GB de RAM.

## 2. Preparación de Input

Se cuenta con un set de 232 poemas de Pablo Neruda, con un total de 361321 caracteres. El archivo viene con marcadores especiales para separar poemas, estrofas y versos.

Se procesa el archivo de tal forma que se genere un vector que contenga todos los versos, en formato de vector con el código correspondiente a cada caracter, el cual fue calculado previamente.

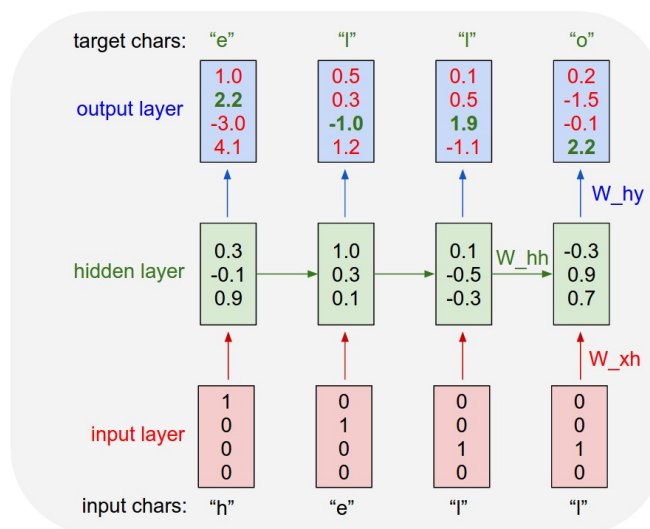


Figura 1: Diagrama de la RNN a construir

## 3. Construcción de Red Neuronal

A diferencia de la tarea anterior, se usaron matrices para representar las distintas capas de la red neuronal, con el objetivo de realizar los algoritmos de forma eficiente realizando operaciones sobre matrices. Los siguientes parámetros matriciales caracterizan a la red:

- $WXH$ : representa los pesos entre el input ( $X$ ) y la capa oculta ( $H$ )
- $WHH$ : representa los pesos entre neuronas de la capa oculta. Son la "memoria" de la RNN
- $WHY$ : representa los pesos entre la capa oculta y el output ( $Y$ )
- $bH$ : bias de la capa oculta
- $bY$ : bias del output

Se construyó la red de tal forma que permita cargar estos parámetros desde archivos, con el objetivo de realizar sampling posterior, o continuar con el entrenamiento de la red.

### 3.1. Forward step

En este paso, se construyen las representaciones de *hot vector* del input, esto es, un array con 1 en la posición del caracter correspondiente, y 0 en el resto. Además, se realiza el cálculo de los output de las neuronas de la capa oculta y de salida, para finalmente calcular el vector distribución de probabilidad final. Para una RNN, las ecuaciones son las siguientes:

- $h_t = \tanh(WHX \cdot x_t + WHH \cdot h_{t-1} + bH)$
- $y_t = WHY \cdot h_t + bY$
- $p_t = \text{softmax}(y_t)$

Se escogió la tangente hiperbólica como función de activación, pues presenta gradientes más fuertes que el sigmoid, y tiene un rango mayor. Se utiliza *softmax* para calcular la distribución de probabilidad. 
$$\text{softmax}(x) = \frac{\exp x}{\exp \sum x}.$$

### 3.2. Backpropagation

En este paso se calculan los gradientes de los parámetros de la RNN. Notar que es necesario utilizar la derivada de la tangente hiperbólica  $\frac{d}{dx} \tanh x = 1 - \tanh^2 x$ .

### 3.3. Update

En este paso se actualiza el valor de los parámetros. El error fue calculado en los pasos anteriores usando *cross-entropy loss*, y la actualización se realiza usando Adagrad, un tipo de gradiente descendiente.

## 4. Experimentos

Se realizaron experimentos con 3 cantidades de epoch distintas. La primera con sólo un epoch, la segunda con 200 epoch y la tercera con 450 epoch, todas con 100 capas ocultas y un learning rate de 0,1. Se generó un texto de veinte mil caracteres para todas las configuraciones. Además, para la configuración más entrenada, se generó texto realizando variaciones en la decisión probabilística del siguiente caracter:

- Elección sobre distribución de probabilidad: estándar, usado en la generación de texto
- Elegir siempre el caracter con más alta probabilidad
- Híbrido entre las dos opciones anteriores: cuando está generando la primera letra de una palabra, usa la distribución de probabilidad, y cuando está generando caracteres dentro de una palabra, usa el caracter con más alta probabilidad

## 5. Resultados

### 5.1. Sample con 1 epoch

pradarenémir leraco.  
El sue po,  
deña  
ne uatiempa piocisnalcimidógbren mhjudestojapeque  
cabre de eliesMeno  
sebardabarasraas a,  
lara y emio.

Y ernelo erresterloguerer, lo conlotiteudocón  
s cor ilrarabra mielilo  
éja cartabre tertrora an os  
denebrar de quecaba  
ses ga:  
havia murtoresinta vera se la? : ve a lontecondo si  
predron.

Y astibontasijos y cunsqumararetetraegodo simar men  
arlesmia tu

## 5.2. Sample con 100 epoch

Ascididon  
sin mi di los pararás casa,  
huesta hi atolas sino. Ascós, estávilla.

Ahiencia he mi valque verrillos mi arilta paudos ye el  
agua  
qué mi vida,  
al amargamente,  
como netaza el peso.  
No tibirmanal, mi mis desiljas orgado  
que es esosos.

Entre emiste el quisando e viocó como impunovas,  
como pormies!  
Aho de hetenderes  
ena bor murés deseralma haja univas no que vieron  
que árblos, las estura sile segue el otor mara yo mada  
y entre, que porda, sobre océadas,  
nadio y este en la vistir, canas en casta, y mi comos en  
mi vieles entres al anumiras de viento resmbada,  
es yo sólo de otaúdas y masado dómilelo.

Anlos.  
Amando y modieno  
encagio, oria!  
Larnoy manto  
nos tenó bamo el mi amiqué lejos.  
Ases y, masta hay me la morvamensiguncarios,  
como volas que quelundido y del mundos, las hosa.  
Y  
pombras y ya, sin altas, ardes que mientos, ciego de  
Macho!  
Aques su vida se qué hace y cocinitan se hos que espira  
pocas como na corre su descia.

### 5.3. Sample con 200 epoch

Nudrisa:  
el pré andán y dombeso  
conéjo cuanzante.  
Cabré tu piema milera,  
paledas len y escallón que o petán misto  
mirnendo  
pascuina de mero interañás enden en el ototeda su  
talallos vestidos,  
un umbre las miriderne co pilce al llagamosemos  
tría que escantas  
con abitan  
paraceadorda miempo derdidetrigos:  
termira bramando caíderte la sigo y no se vejen  
entrecentimer una picierto az minto rodriempa  
parabaro así caía!

### 5.4. Samples con 450 epoch

#### 5.4.1. Método estándar

Tes que estmpándo,  
con las aque do,  
yolmado decico en leño por taste hacio  
crilles cruel voz el rigredo mía el charas  
algosoha que el otravas,  
hastados,  
con munez  
y y las hoy de su sanos pájaces  
este tantirilos,  
suaroto qué contío viven lego morone?

Conturamente me lojas alga, hojas azul mundo,  
mungaros ola, los nocentia muertes, el alvezas,  
de un cricales:  
con los campanados y ar al raíz  
si por las despetud.

### 5.5. Método de máxima probabilidad

Es el mar en la estrella en la estrella en la estrella en la  
estrella en la estrella en la estrella ...



## 5.6. Método híbrido

No sol se sube la luz que me desperas en el tiempo un  
corazón solo por el hombre las cosas de la que al palente  
interio de oro de este rosas se para ola de tristeza que la  
luna con un día a una al palente están gallar o la orrar  
los piedras algo mi para hora de mi inveridos de humo y  
el océan la pallas las sembante de despado solo que se  
la verdidos los piedras por las como para de alviel como  
bajo estar como invieron jon alegre me para al palente  
de un las hojas boca la llena de la invia un sus por una  
un golpera de la una venido a como el hombre con las  
despado a por como en la recesta en que con alma en  
la orgulas triste gotalla con despera con alma los alma  
de palo de la madieron en veces no interan grande las  
como en la vida en la estrella de la pallas paso un noche  
la luna en la tierra tu sangre azul por invidado en todo  
la contigo en la muerte la invierto manos de sueron y  
...

## 6. Análisis de Resultados

Se observa de los resultados que la sintáxis (estructura) de los poemas fue generada de forma satisfactoria, sin haber diferencias significativas respecto a la cantidad de entrenamiento.

Con respecto al contenido de los poemas, se observa que mientras más entrenada está la red, más palabras correctas logra generar, pero que aún con 450 epoch sigue generando muy pocas palabras existentes.

Con respecto a la variación en la decisión probabilística, la elección del caracter con más alta probabilidad lleva a un loop de usualmente 3 palabras. Estas palabras, sin embargo, son existentes. Esta pista fue la que nos llevó a realizar el tercer método híbrido de predicción. Con este método, se generaron muchas palabras existentes pero se perdió la estructura de un poema y la puntuación del lenguaje. Esto tiene sentido, ya que es más probable que una letra esté acompañada de más letras que de un signo de puntuación o un marcador especial.

Los distintos samples pueden ser generados ejecutando los archivos *sample.method.py filename*, y la red puede ser entrenada ejecutando el archivo *train.py*.

## 7. Conclusión

Se concluye que la RNN fue capaz de genera poemas sintácticamente correctos, no así con la semántica, generando en su mayoría palabras inexistentes. Se cree que una mayor cantidad de entrenamiento, así como un dataset mayor puede generar textos de mayor calidad. Se debe considerar como un factor importante que el vocabulario de un poeta es muy amplio.

Queda pendiente la realización de otras pruebas con distintos dataset y en inglés, para contrastar los resultados considerando el tamaño del vocabulario y las características del idioma. Además, es deseable realizar

esta implementación utilizando alguna librería de cálculo en GPU como Theano.

Como comentario personal, fue una tarea muy entretenida, con la que me hubiera gustado tener más tiempo para realizar diversas pruebas, pero eso ya sería más un proyecto semestral que una tarea :) .