



**Universitat de Girona**

ESCOLA POLITÈCNICA SUPERIOR

PROJECTE GEC

*Tècniques Avançades d'Intel·ligència Artificial*

*JOSEP LLUÍS ARCÓS ROSELL*

*MARIA BEATRIZ LÓPEZ IBAÑEZ*

Iuri Pons Frago, u1956242

# Índex

<b>1</b>	<b>Introducció</b>	<b>2</b>
<b>2</b>	<b>Xarxes Neuronals Artificials</b>	<b>3</b>
<b>3</b>	<b>Representació Numèrica d'un Text</b>	<b>5</b>
<b>4</b>	<b>Xarxes Neuronals Recurrents</b>	<b>7</b>
<b>5</b>	<b>Transformes</b>	<b>8</b>
<b>6</b>	<b>Realització del Projecte</b>	<b>10</b>

# 1 Introducció

Aquest projecte de Tècniques Avançades d'Intel·ligència Artificial tracta sobre l'aprenentatge d'un corrector d'errors gramatical (*CEG*; o *GEC* en anglès *Grammatical Error Correction*).

Aquests correctors, usualment, treballen amb llenguatges naturals creats per l'ésser humà. La disciplina de la informàtica que estudia i analitza els llenguatges naturals és el processament del llenguatge natural (*PLN*; o *NLP* del seu nom en anglès, *Natural Language Processing*).

Partint de la premissa que els models *GEC* són costosos d'entrenar, sigui per temps o per capacitat computacional, i nosaltres fem servir una computadora de gamma mitja, el nostre objectiu és entendre, analitzar i aprofundir en aquesta disciplina.

Abans d'entrar en matèria, per entendre millor el conceptes que treballarem a continuació, hauríem de fer una reflexió sobre què comporta crear un *GEC*. Com sabem tots, el llenguatge d'una màquina és el binari. Però, la tasca que volem resoldre es basa en un llenguatge completament diferent.

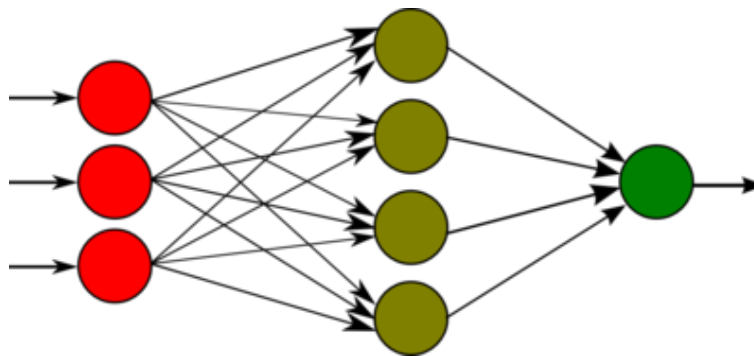
Cada *NLP* funciona diferent, tenen diferents alfabetes, s'estructuren diferent i, inclús, l'ordre d'escriptura i lectura són diferents. Tot i tenir aquestes diferències, tots comparteixen una característica molt important, segueixen un conjunt de regles. El problema és que cada llenguatge té moltes regles i cada regla, normalment, té excepcions. A part, el context i l'ambigüitat són molt rellevant i són aquests motius pels quals es fa impossible, o molt tediós, codificar un sol llenguatge humà.

Tot i que hi ha molts programadors que no han descartat el fet de programar totes les regles d'un *NLP*, aquí només ens centrarem en la potència del *machine learning*. Concretament, farem servir una metodologia que fa molt de temps que s'està investigant i desenvolupant que són les xarxes neuronals.

## 2 Xarxes Neuronals Artificials

Una **xarxa neuronal artificial** (XNA), també anomenada xarxa neuronal simulada o senzillament xarxa neuronal (denominada habitualment en anglès com *ANN*) és un paradigma d'aprenentatge i processament automàtic inspirat en la forma en què funciona el sistema nerviós dels animals.

Es tracta d'un sistema d'interconnexió de neurones en una xarxa que col·labora per produir un estímul de sortida. Aquest conjunt de neurones artificials interconnectades utilitza un model matemàtic o computacional de processament de dades basat en una aproximació connexionista per a la computació.



*Figura 1: Xarxa neuronal artificial amb 3 neurones d'entrada, 4 neurones en la seva capa oculta i una neurona de sortida.*

Sense entrar en detalls del funcionament tan complex que tenen les xarxes neuronals, podem destacar que l'objectiu d'aquestes és donar una sortida respecte als paràmetres d'entrada introduïts mitjançant una selecció de paràmetres i càlculs matemàtics.

També cal saber els avantatges que tenen aquestes xarxes, per saber si realment són una bona eina per resoldre el nostre problema:

- **Aprenentatge:** Tenen l'habilitat d'aprendre mitjançant una etapa que s'anomena *etapa d'aprenentatge*. Aquesta ho fa sobre un conjunt de dades d'entrada i les compara amb les dades de sortida que se li ha indicat prèviament.
- **Auto organització:** Una xarxa neuronal crea la seva pròpia representació de la informació al seu interior, sense que el programador se n'hagi de preocupar.
- **Tolerància a fallades:** Atès que una xarxa neuronal emmagatzema la informació de forma redundant, aquesta pot seguir responnent de manera acceptable fins i tot si es fa malbé parcialment.
- **Flexibilitat:** Una xarxa neuronal pot gestionar canvis no importants en la informació d'entrada (ex. si la informació d'entrada és la imatge d'un objecte, la resposta corresponent no pateix canvis si la imatge canvia una mica la seva brillantor o l'objecte canvia lleugerament).

- Temps real: L'estructura d'una xarxa neuronal és paral·lela, per la qual cosa si això és implementat amb ordinadors o en dispositius electrònics especials, es poden obtenir respostes en temps real.
- Escalabilitat: poden adaptar-se a qualsevol problema d'una determinada àrea.

Veient tots els avantatges que tenen les *XNA*, arribem a la conclusió que efectivament són una molt bona eina per dur a terme la resolució del nostre problema. Però, hi ha un punt molt important que s'ha comentat a la introducció. Com que les xarxes neuronals artificials estan implementades en els computadors, aquestes també treballen amb dades numèriques i nosaltres volem tractar text.

Quan es tracta de treballar amb imatges, àudios o simplement amb dades numèriques tenim molt clar com fer el tractament de dades si cal. Però quan es tracta d'un text i l'enteniment d'aquest no és tan senzill.

### 3 Representació Numèrica d'un Text

Arran del problema que hem trobat anteriorment, ens hem de posar a pensar com podem representar un text numèricament. Segurament, una de les primeres solucions que sorgeixen a qualsevol persona és assignar un nombre a cada paraula. Aquest ja pot ser a l'atzar o per ordre, però és important que cada paraula tingui un nombre diferent i les paraules amb el mateix lexema comparteixin nombre.

Així doncs el següent text el podríem representar de la següent manera:

El pare està amb el seu fill.

25 15 32 65 25 8 3

Figura 2: Text representat numèricament amb nombres escollits a l'atzar.

Ara ja hem pogut resoldre el problema de la representació numèrica d'un text, però és el correcte. Com millor funcionen els sistemes d'aprenentatges, sigui artificial o humà, és mitjançant la lògica i la correlació entre les dades. Si nosaltres volem fer entendre a una màquina un llenguatge humà, doncs ha de saber que les paraules tenen relació i similitud.

Segons la representació numèrica de la figura 2, veiem que la paraula *està* té un valor que duplica el valor de la paraula *pare*. I la diferència entre *pare* i *fill* és més gran que la diferència entre *fill* i *seu*, cosa que no hauria de ser així.

Arribem a la conclusió, que per representar un conjunt de paraules no podem fer servir nombre a l'atzar. Doncs la solució que es va proposar fa uns anys és representar cada paraula vectorialment. Enlloc de tenir un valor per cada paraula, tindrem un vector de mida  $m$ .

Per evitar que les paraules quedin relacionades de manera incorrecte, farem que el vector sigui tan llarg com paraules hi hagi en el nostre vocabulari, així doncs  $m = \text{mida del vocabulari}$ . Per exemple, si tinguéssim tres paraules les representàrem com es mostra a la següent figura.

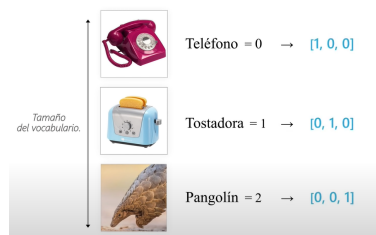


Figura 3: Representació vectorial One-Hot Encoding de 3 paraules diferents.

A la figura 3, veiem que ara cada paraula té la seva pròpia dimensió i la distància entre elles és la mateixa, així doncs hem eliminat el malentés de relació de paraules, però tampoc és veritat que la relació entre cada paraula és la mateixa.

El que hem de fer doncs, és trobar la relació entre les diferents paraules d'un llenguatge, però altre cop, això és una feina molt complicada per un ésser humà. Per tant, farem servir una màquina per dur a terme a aquesta tasca.

Això vol dir que farem servir un model entrenat per relacionar paraules vectorialment, per després fer servir aquestes representacions per entrenar un altre model per entendre un llenguatge humà. Aquest primer model rep el nom del concepte *Embedding*.

## 4 Xarxes Neuronals Recurrents

Ara ja tenim clar com representar una paraula, però encara no ha acabat el problema. Està clar que depenent del model entrenat per classificar les paraules i del text d'entrada que li donem per entrenar-se, tindrà un criteri d'avaluació en concret. Això vol dir, que podrem representar qualsevol paraula.

La qüestió ara és que per entendre una oració no només val classificar cada paraula segons un corpus de dades gegant. També s'ha d'analitzar el context i la relació de les paraules en aquesta frase.

Per poder tractar una oració tenint en compte cada paraula d'aquesta, s'ha fet servir durant molt de temps les **xarxes neuronals recurrents** (XNR; o bé en anglès *RNN*, acrònim de *Recurrent Neural Network*).

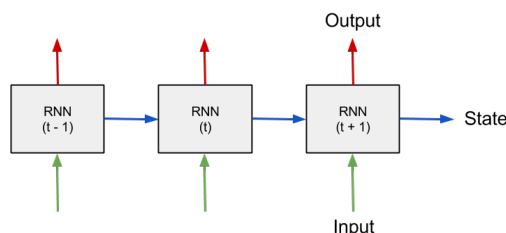


Figura 4: Esquema del comportament d'una RNN.

El comportament d'una RNN, com podem veure a la figura anterior, es basa a tractar les dades d'entrada de manera seqüencial i que cada element que s'analitza tingui en compte els anteriors. Així doncs, li donem un significat a l'estructura de la frase i donem context a oracions com:

*La taronja és rodona; La rodona és taronja*

Com podem observar, en aquestes dues frases es fan servir les mateixes quatre paraules, això vol dir que segons la representació que li havíem donat a cada paraula, haurien de ser iguals. Però, hi ha una evidència molt clara per nosaltres, els humans, i és el canvi de subjecte.

En fer servir una RNN, analitzaria que el verb *és* de la primera frase té relació amb *la taronja*, i el verb *emphés* de la segona té relació amb *la rodona*. Així doncs, ja hem resolt el problema del context i els models són capaços d'entendre'l. A excepció de les frases extenses...



## 5 Transformes

Sí, ja havíem arribat a una solució i sí, el tractament de dades seqüencial és la metodologia correcta per resoldre aquest tipus de problemes. Però resulta que com més extensa és l'oració que volem analitzar, menys relació tenen les paraules del principi amb les del final. Per exemple:

*L'ocell recollia pals grans i petits per poder acabar de construir el seu niu.*

En aquesta oració, el **sintagma nominal** *el seu niu* fa referència a *l'ocell*. Però la xarxa neuronal ha hagut de tractar tantes paraules pel mig que ha perdut el context. És per això que s'ha d'acabar de matisar el comportament d'una RNN.

El problema que hem de resoldre ara es denomina *falta de memòria* o *falta d'atenció*, i és que de fet, si no estem atents a l'inici de l'oració, no podrem saber la referència de les darreres paraules. Arran d'això, va sortir un nou *paper* que va capgirar tota la branca de *machine learning*, **Attention Is All You Need**.

El que ens presenten en aquest *paper*, a grosso modo, és el **Transformer**, el primer model de transducció de seqüència basat completament en atenció. Els dos principals punts forts dels transformers són els mecanismes d'atenció, que permet relacionar cadascun dels elements a tractar, i l'ús de la posició relativa i absoluta de cada element de la seqüència.

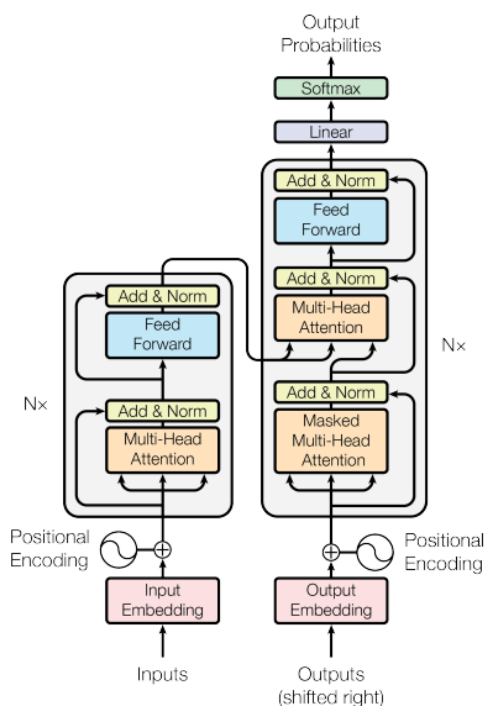


Figura 5: Arquitectura del Transformer.

També s'ha de tenir en compte l'excel·lent avenç de la capacitat d'entrenar aquests models amb temps significativament inferiors a les RNN. Això és degut al fet que ja no depenem del tractament seqüencial sinó que podem tractar els diferents elements paral·lelament.

Avui en dia, pràcticament tots els models que usaven xarxes neuronals recurrents han sigut substituïts per transformers, arribant a així a nou rècords de *machine learning* i aquests han pogut resoldre tasques que fins i tot no havien sigut entrenats per fer-ho.

Aquest és l'exemple de [GPT-3](#), un model transformer entrenat per generar text de tal manera que sigui molt semblant a com ho fem els humans. Aquest rep com a entrada la introducció o els tòpics d'un concepte i és capaç d'interpretar què es demana i realitzar la tasca.

Un exemple seria l'entrada de dues línies d'un diàleg i GPT-3 és capaç de continuar amb el diàleg. Com més específica sigui l'entrada que li donem, més exhaustiva serà la sortida que doni. El que no s'esperava és que aquest model tingués la capacitat de generar codi de programació a partir d'una entrada que defineix la tasca a realitzar.

Altres aplicacions que poden tenir l'ús de la tecnologia dels transformers podrien ser el processament d'imatges i la identificació de patrons per poder-les classificar, el motor de cerca que usen plataformes com *Google*, l'assistència de veu o robots domèstics, el mecanisme d'etiquetar, corregir o traduir textos, entre altres.

## 6 Realització del Projecte

Arribat en aquest punt, ja sabem que l'ús del transformer és una de les millors metodologies per resoldre la correcció d'errors gramaticals. La llibreria que recomana el *paper* que hem vist anteriorment del transformer és [Tensor2Tensor](#) administrada [Tensorflow](#).