

Xarxes neuronals recurrents per l'anàlisi de sentiments:
classificació binària de ressenyes de pel·lícules



Gabriel Pombo Hillenius
Tutor: Toni Lozano Bagén

21 de juliol de 2023

Índex

1	Introducció	3
2	Xarxes Neuronals	5
3	Xarxes Neuronals Recurrents	7
3.1	Arquitectures	13
4	Cas pràctic	15
4.1	Metodologia	19
4.2	RNN senzilles	21
4.3	RNN complexes	22
4.4	Resultats	23
5	Conclusions	24

1 Introducció

L'anàlisi de sentiments a través del processament de llenguatge natural és un camp d'estudi que té com a objectiu categoritzar el sentiment d'un text en positiu, negatiu o neutre. Donades les grans quantitats de text que hi ha actualment i la impossibilitat de categoritzar-lo manualment, el desenvolupament en tècniques d'aprenentatge automàtic i l'entrenament de models amb conjunts de dades etiquetades han mostrat grans avenços en tasques de classificació sentimental de correus electrònics, publicacions a xarxes socials, ressenyes i xats d'atenció al client. Les empreses han utilitzat l'entrenament de models de categorització de textos per monitoritzar les seves marques i identificar tendències a les opinions dels clients.

Les xarxes neuronals recurrents han demostrat adaptar-se a les estructures seqüencials dels textos i les relacions d'àmplia distància entre variables per les seves habilitats en capturar patrons complexos, relacions a les dades i informació contextual. S'adapten bé als textos ja que poden processar-los sense la necessitat de tenir la mateixa extensió entre ells [8], sense el requeriment d'un pre-processat complex i amb grans volums de dades a través de la computació paral·lela. Són capaces d'aprendre les diferents relacions jeràrquiques d'un text entre paraules, frases i corpus. L'objectiu d'aquest projecte és estudiar les xarxes neuronals i en concret el processament de llenguatge natural (NLP) per l'anàlisi de sentiments. Es compararan xarxes neuronals recurrents senzilles de classificació binària de sentiments amb xarxes neuronals recurrents més complexes.

En aquest projecte es treballarà amb una base de dades de 50,000 ressenyes de pel·lícules d'IMDB [5], una pàgina web que emmagatzema informació de pel·lícules i on els usuaris poden valorar les pel·lícules del 0 al 10. En particular és una base de dades recollida per la Universitat d'Stanford que serveix per fer classificació binària de textos en sentiments positius i negatius.

Conté 25,000 ressenyes d'entrenament i 25,000 de test.

Per a categoritzar la base de dades s'han classificat com a positives les ressenyes de puntuacions més grans o iguals que 7 (sobre 10) i com a negatives les menors o iguals que 4 (sobre 10). La distribució binària (positiu-negatiu) està balancejada i el nombre de ressenyes per pel·lícula està limitat a 30 per a evitar tenir correlació entre ressenyes. Per tal de que la xarxa neuronal no associï els termes propis de pel·lícules concretes a una etiqueta, les dades d'entrenament i test contenen pel·lícules disjunctes.

Per al processament de dades s'utilitzarà el programari Python, específicament, la llibreria Numpy [6] i per la creació i avaluació dels models de xarxes neuronals la llibreria Tensor Flow Keras [4].

2 Xarxes Neuronals

Les xarxes neuronals són models computacionals compostos per neurones artificials interconnectades que serveixen per entrenar models d'aprenentatge automàtic i han tingut molta popularitat en els últims anys per a la resolució complexa de problemes. Són especialment útils en el reconeixement de patrons a les dades, l'aprenentatge de dades i la predicció. S'adapten a diferents àmbits i tipus de dades: per a les dades tabulars s'utilitzen les xarxes neuronals totalment connectades (FCNN), per les dades no tabulars en format d'imatges les xarxes neuronals convolucionals (CNN) i per a les dades seqüencials les xarxes neuronals recurrents (RNN).

El seu funcionament està inspirat en el funcionament de la neurona cerebral i el seu mecanisme de comunicació i es fan càlculs computacionals - aplicació d'una funció per passar d'un valor d'entrada a un valor de sortida.

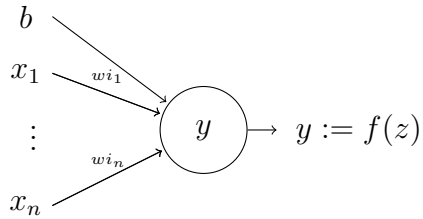


Figura 1: Estructura d'una neurona

L'entrada d'una neurona poden ser un o diversos valors $X = x_1, x_2, \dots, x_n$ als que se'ls hi apliquen uns pesos $W = w_{i_1}, w_{i_2}, \dots, w_{i_n}$ i un biaix b (que determinen la influència de les dades d'entrada a la sortida) i es combinen amb una funció d'entrada z , al resultat de la qual se li s'aplica una funció d'activació f per obtenir una sortida y que és passada a la següent neurona.

La xarxa neuronal connecta les diferents neurones i les organitza en capes: una capa d'entrada, una de sortida i entre mig les capes ocultes.

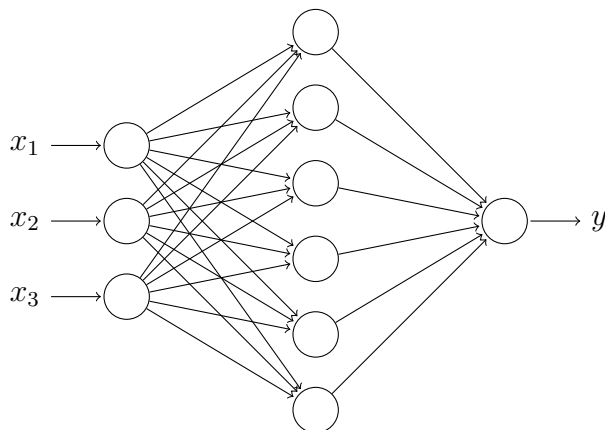


Figura 2: Estructura d'una xarxa neuronal

L'entrenament de xarxes neuronals tracta d'ajustar els pesos per reduir la funció de cost (loss function) a través del mètode de retropropagació [2], que consisteix en introduir dades d'entrenament d'entrada a la xarxa, computar-la amb pesos i biaix inicialment aleatoris i obtenir la sortida. De la diferència entre la sortida predita i la sortida desitjada es calcula l'error. Posteriorment, es propaga l'error fins la capa d'entrada i utilitzant l'algoritme del descens del gradient es van actualitzant els pesos per minimitzar la funció de cost.

En l'algoritme del descens del gradient, el gradient indica la direcció cap a on hi ha més magnitud de canvi, per tant, com la idea és minimitzar la funció de cost, s'actualitzaran els pesos movent-se en direcció contrària al gradient. A l'hora d'actualitzar el pesos en direcció contrària, multiplicarem el gradient per la taxa d'aprenentatge, que depenent de la seva magnitud, farà que avanci amb passos més o menys grans.

3 Xarxes Neuronals Recurrents

Les xarxes neuronals recurrents s'introdueixen per al tractament de dades seqüencials i sèries temporals. Ho fan a partir del concepte de recurrència, on al contrari de les xarxes neuronals tradicionals, la informació també pot fluir de la capa de sortida a la capa d'entrada, guardant així informació sobre elements passats de la seqüència a l'estat ocult de la xarxa [7]. Per altra banda, també tenen la capacitat de tractar dades seqüencials de diferents longituds, alhora que respecten la importància de l'ordre en la seqüència.

S'adapten bé al processament de llenguatge natural: traducció automàtica (Google Translate), anàlisi de sentiments, predicció de text... També al reconeixement de veu (Siri), l'anàlisi i predicció de sèries temporals o inclús la generació de música.

L'input de la cel·la recurrent són el valor x_t de la seqüència en el pas t i l'estat ocult de la xarxa al pas anterior s_{t-1} . L'output són la sortida o resposta h_t per a aquest punt i l'estat de la xarxa actualitzat per al pas t .

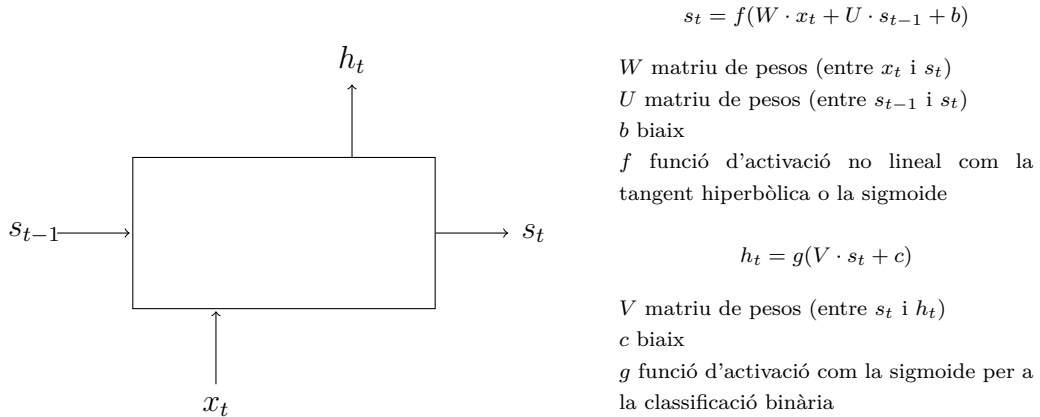


Figura 3: Estructura d'una neurona recurrent

Per a les xarxes, l'entrada són els elements de la seqüència, que van sent processats en ordre alhora que s'emmagatzema la informació corresponent a l'estat ocult. Per a que l'entrada de la xarxa neuronal processi les seqüències de text, previament les paraules han de ser vectoritzades numèricament.

En l'entrenament de la xarxa, es fixarà una longitud màxima per a les seqüències que es processaran: si la seqüència concreta té menys valors que el valor fixat, s'ompliran amb zeros. En canvi, si la seqüència en té més, s'haurà de tallar la seqüència descartant-ne una part.

La sortida de la xarxa dependrà de la tasca concreta que s'estigui fent. Pot ser l'estat ocult en cadascun dels passos de la xarxa, per exemple en casos de reconeixement d'entitats nombrades (NER), on la xarxa processa la seqüència de text d'entrada i l'estat ocult a cada pas és utilitzat per localitzar entitats en forma de persones, organitzacions, llocs...

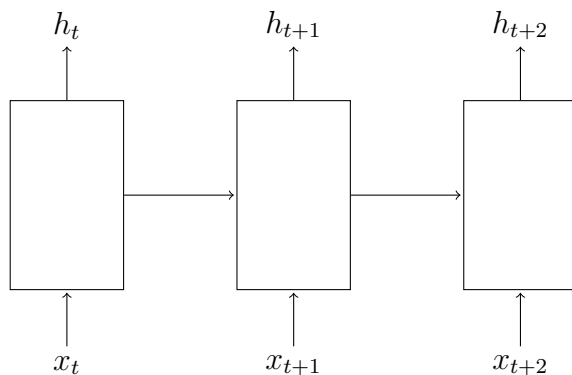


Figura 4: RNN amb diverses dades d'entrada i sortida

La sortida també pot ser directament l'output de la xarxa en l'últim pas, per exemple en casos com l'anàlisi de sentiments en seqüències de text, on s'obindrà una distribució de probabilitat sobre les 3 categories: positiu, negatiu o neutre.

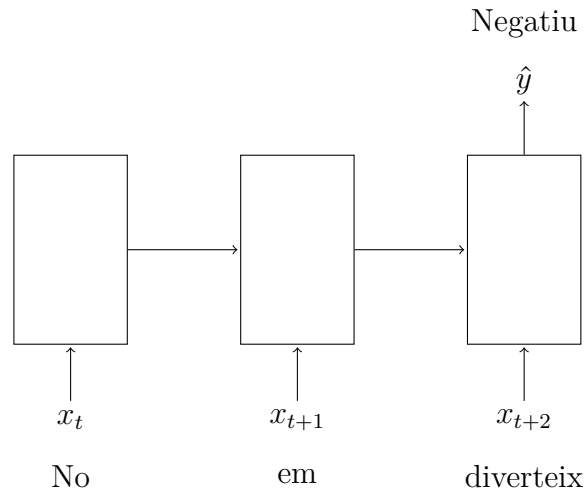


Figura 5: RNN amb diverses dades d'entrada i una única sortida

A la generació o predicció de text, on l'objectiu és processar una seqüència de text i obtenir la següent o següents paraules que genera la xarxa, un cop tractada la seqüència d'entrada estarem interessats en observar les sortides de la xarxa. Al afegir entrades amb valors nuls al final de la seqüència podrem obtenir la predicció de les següents paraules de la seqüència.

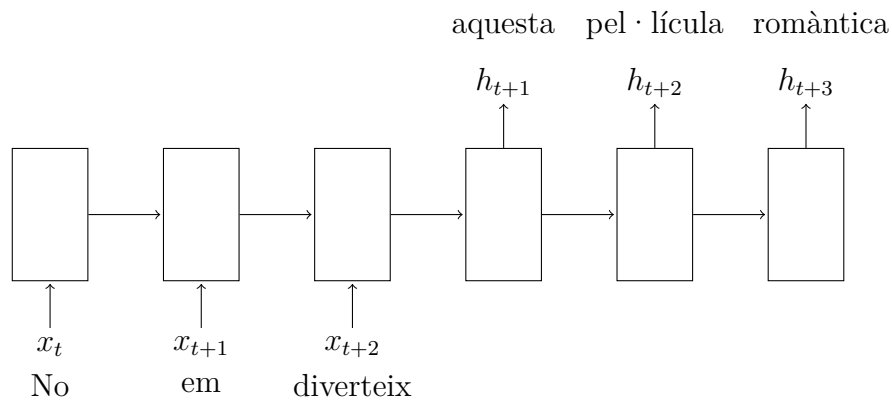


Figura 6: RNN amb diverses dades d'entrada i sortida

Els pesos i el biaix de la xarxa neuronal recurrent són compartits entre les diferents cel·les i es mantenen a cada pas, permetent així que la xarxa

pugui tractar seqüències de longituds variables i que l'estat ocult pugui codificar informació sobre elements passats.

Per l'entrenament de la xarxa, l'objectiu és com a la resta de xarxes neuronals trobar uns paràmetres que minimitzin la funció de cost. Però les RNN utilitzen l'algoritme de retropropagació en el temps, una modificació que s'adapta a la recurrència i permet que l'algoritme s'ajusti al fet que la informació flueixi de la capa de sortida a la capa d'entrada.

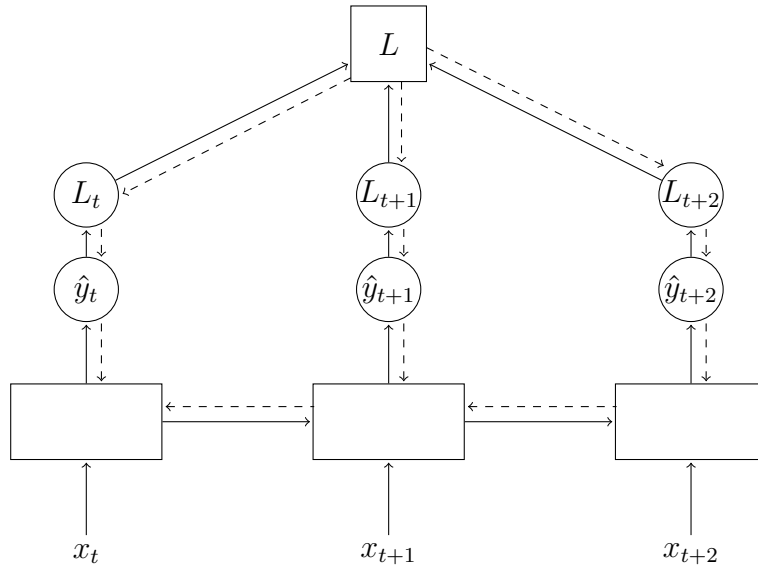


Figura 7: Algoritme de retropropagació en el temps

La diferència entre la sortida predita i la sortida desitjada resultarà en l'error que s'anirà propagant fins la capa d'entrada utilitzant l'algoritme del descens del gradient. Com que els paràmetres (pesos i biaix) es mantenen a tots els passos, els gradients es multiplicaran per si mateixos exponencialment i provocaran dos tipus de problemàtiques: desaparició o explosió del gradient, segons si la norma del gradient és major o menor que 1, provocant dificultats per a capturar dependències a llarg termini i inestabilitat en l'entrenament de la xarxa.

Per a solucionar el problema del gradient, les RNN han introduït l'ús de cel·les Long Short Term Memory (LSTM) [3] i Gated Recurrent Unit (GRU) [1], que amb una estructura de cel·la més complexa controlen la propagació del gradient i el flux d'informació a partir de l'ús de portes.

Les LSTM permeten preservar informació seqüencial en el temps, per exemple en l'anàlisi de sentiments en ressenyes, informació sobre elements inicials de la seqüència de text. A diferència de la cel·la recurrent explicada anteriorment, aquestes tenen com a entrada l'element de la seqüència en aquell pas i la concatenació de la sortida i l'estat ocult en el pas anterior. Tenen tres tipus de portes: la d'oblit, que decideix quina informació de la memòria en l'anterior pas es queda; la porta d'entrada, que selecciona amb quina informació s'ha d'actualitzar la memòria de la xarxa i per últim la porta de sortida, que determina quina informació donarà com a resposta la cel·la. Totes 3 permeten que el gradient flueixi sense interrupcions.

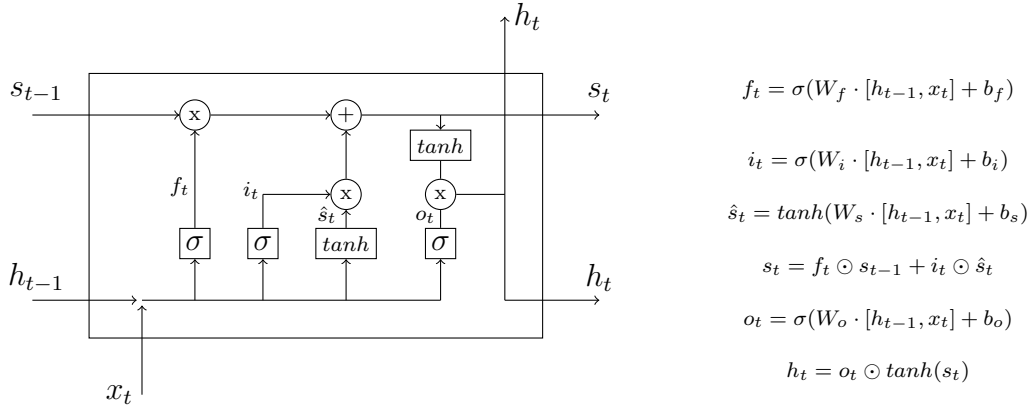


Figura 8: Cel·la Long Short Term Memory

On f_t porta d'oblit, i_t porta d'entrada, \hat{s}_t calcula quina informació exactament s'afegeix a la memòria, s_t actualitza la memòria, \odot producte de vectors element a element, o_t porta de sortida, h_t dona el vector de sortida i

els vectors d'entrada de les portes estan en $[0,1]$.

Les cel·les GRU també utilitzen portes per a controlar el flux d'informació i la propagació del gradient. Però, simplifiquen l'estructura en dues portes: reinici i actualització. Com que tenen menys paràmetres, l'entrenament no requereix un conjunt de dades tant gran.

La diferència entre les dues portes és que la d'actualització determina quina part de la memòria al pas anterior s'ha de passar en combinació amb l'entrada de la cel·la a la memòria del pas actual. En canvi, la de reinici determina quina part de la memòria al pas anterior s'ha de deixar enrere abans de calcular la memòria actual.

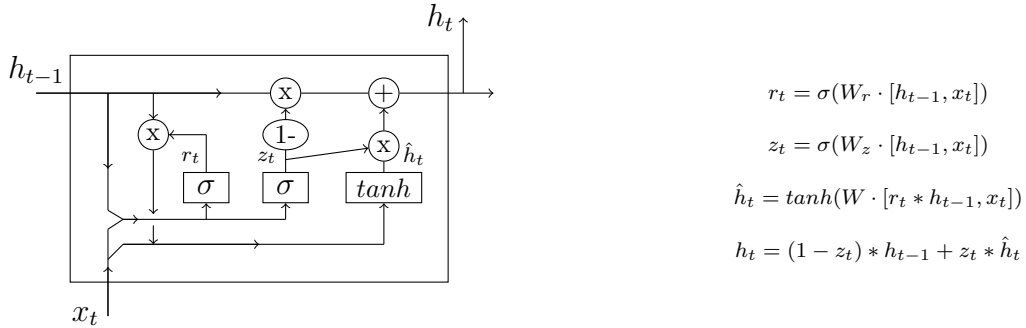


Figura 9: Cel·la Gated Recurrent Unit

On z_t porta d'actualització, r_t porta de reinici i h_t calcula l'estat de la xarxa en el pas actual t .

3.1 Arquitectures

Dins les xarxes neuronals recurrents trobem diferents arquitectures o estructures de xarxa. La més senzilla és la RNN bàsica, que té l'estructura estàndard d'una recurrència i la informació seqüencial d'entrada es passa només en una sola direcció i ordre, de passat a futur.

La RNN bidireccional, no només processa en la direcció de passat a futur, sinó que també introdueix la informació amb l'ordre invertit, d'elements futurs a passats. Això permet capturar la informació del context de la seqüència tant del passat com del futur i tenir en compte els dos ordres d'entrada de les dades.

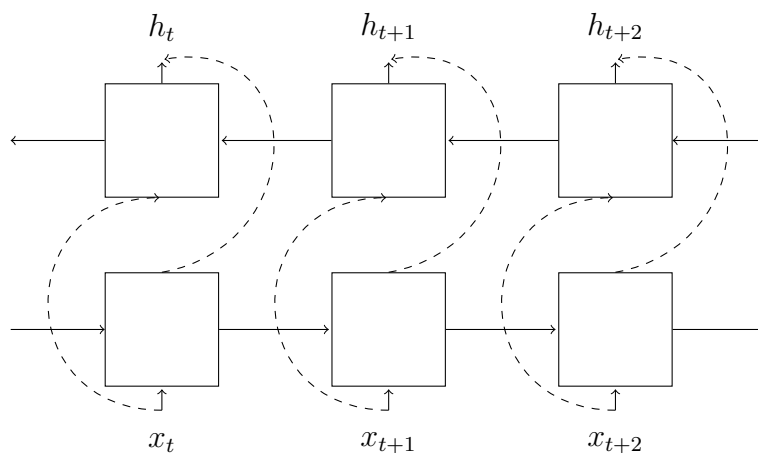


Figura 10: RNN amb arquitectura bidireccional

La RNN profunda, permet capturar relacions complexes entre elements de dades seqüencials a llarg termini. Les diferents cel·les d'una capa comparteixen paràmetres, però entre les diferents capes no són compartits i poden tenir la seva pròpia estructura interna, com poden ser les LSTM o les GRU. Requereixen conjunts de dades grans perquè utilitzin molts paràmetres. L'output d'una capa serà utilitzat per la següent capa, permetent així

que les últimes capes capturin informació a llarg termini sobre elements d'entrada anteriors.

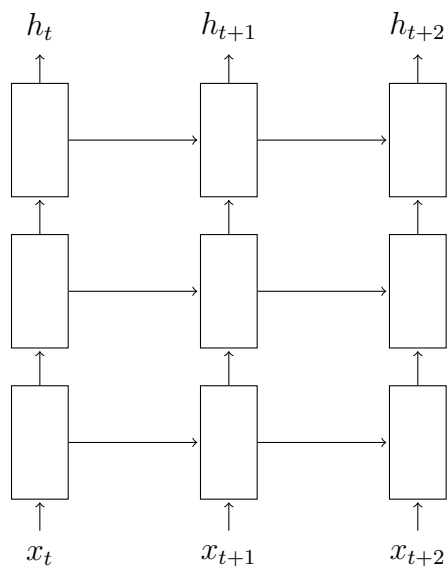


Figura 11: RNN amb arquitectura profunda

4 Cas pràctic

La base de dades "Large movie review dataset" [5] de la Universitat d'Stanford es va crear l'any 2011 i conté 50,000 ressenyes de pel·lícules de la web cinematogràfica IMDB. Cada ressenya ha estat escrita en anglès per algun dels usuaris de la web i tenen diferents extensions de la seqüència de text, per tant, s'ha de determinar una longitud màxima de la seqüència d'entrada de text per tal de poder entrenar la xarxa. Actualment es pot obtenir la base de dades ja pre-processada: cada paraula ve indexada amb un número (s'ha limitat a les 10,000 més freqüents) a partir de la tokenització, s'han eliminat les puntuacions i transformat els salts de línia, s'ha convertit tot el text a minúscules i eliminat els números.

Exemple de ressenya positiva:

"# real cool smart movie i loved # colors especially the purple car al-ice # is wise and wonderful as stella i liked # reference to how her face had gotten # the # dance scene is brilliant really liked this one"

Exemple de ressenya negativa:

"# a rating of 1 does not begin to express how dull depressing and re-lentlessly bad this movie is"

La funció de cost o loss function més adequada per aquest cas pràctic és la d'entropia creuada binària (binary cross entropy loss). On y_i és el sentiment a predir, \hat{y}_i el sentiment predit i i és cada ressenya del total de N .

$$L = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Per l'algoritme del descens del gradient es poden utilitzar optimitzadors que aporten una major velocitat d'entrenament a la minimització de la funció de cost. Per l'anàlisi de sentiments el més popular és ADAM (Adaptive Moment Estimation) que combina les avantatges d'altres optimitzadors: Momentum (es té en compte la mitjana de la decaiguda exponencial de gradients anteriors per ajustar la direcció d'actualització dels paràmetres) combinat amb l'ús d'una velocitat d'aprenentatge adaptativa similar a RMSProp (adaptant-la a partir de la mitjana de la decaiguda exponencial dels gradients anteriors).

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t$$

On m_t és el primer moment dels gradients anteriors, v_t el segon moment, β_1 i β_2 les seves corresponents mitjanes de decaiguda exponencial, \hat{m}_t i \hat{v}_t les seves correccions, g_t el gradient al moment t , θ_t els paràmetres resultants al moment actual i η la taxa d'aprenentatge (learning rate), que determinarà la velocitat d'ajust dels pesos a l'hora de minimitzar la funció de cost amb l'algoritme del descens del gradient.

Un altre punt important és definir la mida del batch (batch size), que és el número de ressenyes que es processaran abans d'actualitzar els paràmetres i les èpoques (epochs), que son el número de vegades que es processaran les dades.

Per la capa de sortida, s'utilitzarà la funció d'activació Sigmoide o

logística, que donarà per sortida una probabilitat entre 0 (negativa) i 1 (positiva), sent la classificació del sentiment de la ressenya.

$$\text{sigmoide}(x) = \frac{1}{1 + e^{-x}}$$

El sobreajust o overfitting, es dona quan la xarxa entrenada s'especialitza a les dades d'entrenament però en canvi amb les dades test o noves dades la xarxa no classifica correctament les ressenyes, ja que s'han capturat patrons específics de les dades d'entrenament i no generals.

Per evitar el sobreajust hi ha diverses propostes que s'adapten òptimament a aquest tipus de casos pràctics. El dropout és una capa que anul·la l'efecte d'algunes neurones durant l'entrenament de la xarxa i força a la resta de neurones a aprendre sobre una representació menys específica. El dropout rate és un paràmetre que defineix el percentatge de neurones que es posaran a 0 de manera aleatòria a cada iteració.

L'early stopping és una tècnica que atura l'entrenament del model abans de que es produeixi el sobreajust, que es determina quan la funció de cost deixa de minimitzar-se amb les dades de test (la xarxa s'ha adaptat massa a les dades específiques d'entrenament). En aquest cas, es defineix un paràmetre que son el nombre d'èpoques que es faran abans d'aturar l'entrenament des del punt en que la funció de cost no millora.

Per finalitzar amb les tècniques de reducció de l'overfitting, la regularització introdueix al càlcul de la funció de cost un paràmetre que modifica el resultat de forma que els pesos es calculen sense sobreajustar-se a les dades d'entrenament.

Hi ha dos tipus:

$$\text{L1 (Lasso)} = L_0 + \lambda \sum_{i=1}^n |w_i|$$

$$\text{L2 (Ridge)} = L_0 + \frac{\lambda}{2n} \sum_{i=1}^n w_i^2$$

On L_0 és la funció de cost original, w_i els pesos, λ el paràmetre de regularització i n el nombre de pesos de la xarxa.

Per últim, s'haurà de definir la mesura d'avaluació de la classificació del model. A partir de la matriu de confusió es poden definir diverses mesures d'avaluació:

	Valor predit positiu	Valor predit negatiu
Valor real positiu	Veritable Positiu (TP)	Fals Negatiu (FN)
Valor real negatiu	Fals Positiu (FP)	Veritable Negatiu (TN)

Taula 1: Matriu de confusió

L'exactitud o accuracy representa la quantitat de prediccions correctes sobre el total de casos examinats i és una mesura òptima per avaluar la classificació sentimental del model.

$$\text{Exactitud} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

4.1 Metodologia

L'objectiu del cas pràctic d'aquesta recerca és comparar models en xarxes neuronals senzilles i simples (RNN simple, LSTM simple o GRU simple) amb models més sofisticats que incorporen els diversos elements que s'han investigat a la recerca per obtenir millors resultats en l'anàlisi de sentiments en ressenyes de pel·lícules.

Com s'ha esmentat anteriorment, el conjunt de dades està dividit en 25,000 d'entrenament i 25,000 de test. En altres casos es poden realitzar procediments de divisió del conjunt de dades a través de la validació creuada o altres tècniques més avançades, però en aquest, les dades procedeixen d'una divisió simple amb la condició de no tenir ressenyes comunes entre els dos conjunts.

Per la xarxa neuronal, l'entrada serà una seqüència de paraules que conformarà una ressenya i la sortida serà la probabilitat de sentiment positiu.

El primer pas serà determinar una longitud màxima per a totes les ressenyes (longitud d'entrada). Les ressenyes tenen una extensió mitjana de 234 paraules, una desviació mitjana de 173 i l'extensió màxima i mínima son 7 i 2494. S'ha determinat a partir de diverses proves que la longitud d'entrada seran 500 paraules.

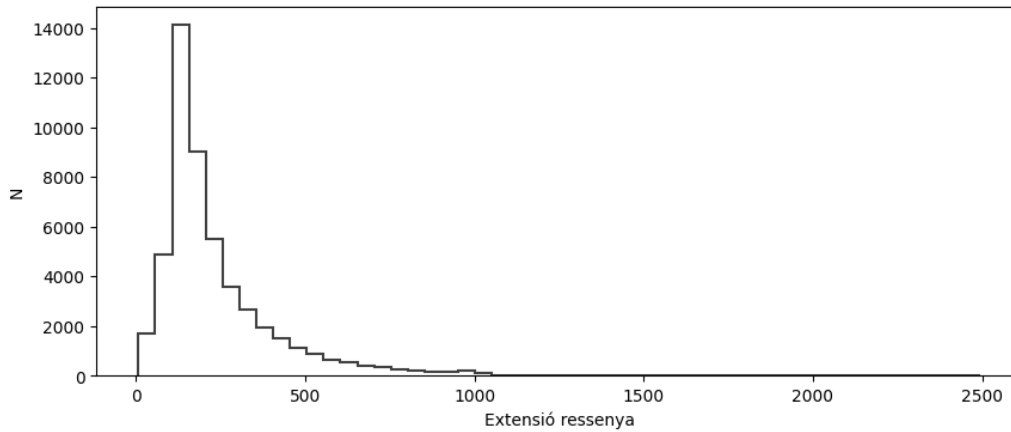


Figura 12: Distribució extensions de les ressenyes

Per altra banda, la xarxa utilitza una capa d'"embedding" que transforma les paraules que estan en índex a tensors-vectors amb una dimensió vectorial de 16 (amb alguna excepció que millorava el model) i s'ha determinat una mida del vocabulari d'aquesta capa de 10,000 paraules diferents, donat que el nombre de paraules úniques a les ressenyes és 9998.

Les capes ocultes dependran de l'arquitectura de cada model, però en canvi totes les xarxes tindran una capa final densa amb una neurona de sortida (classificació binària) i una funció d'activació sigmoide.

L'optimitzador escollit ha estat ADAM, la funció de cost d'entropia creuada binària i la mesura d'avaluació del rendiment l'accuracy. Els models s'entrenaran amb 10 èpoques i una mida del batch de 64.

4.2 RNN senzilles

Les xarxes neuronals senzilles de les que partirà la comparació han tingut la següent configuració:

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 16)	160,000
RNN simple	(0, 64)	5,184
Dense	(0, 1)	65

Taula 2: RNN simple

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 8)	80,000
LSTM	(0, 32)	5,248
Dense	(0, 1)	33

Taula 3: LSTM simple

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 16)	160,000
GRU	(0, 64)	15,744
Dense	(0, 1)	65

Taula 4: GRU simple

4.3 RNN complexes

Les xarxes neuronals que incorporen els elements que més milloren el rendiment amb la base de dades IMDB han estat LSTM i GRU bidireccionals i tenen les següents configuracions:

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 16)	160,000
LSTM bidireccional	(0, 128)	41,472
Dense	(0, 1)	129

Taula 5: LSTM bidireccional

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 16)	160,000
GRU bidireccional	(0, 128)	31,488
Dense	(0, 1)	129

Taula 6: GRU bidireccional

Per últim, també s'han entrenat arquitectures profundes, tot i que no han millorat el rendiment, per exemple la configuració RNN profunda:

Capa	Dimensió de sortida	Paràmetres
Embedding	(0, 500, 16)	160,000
RNN simple	(0, 500, 64)	5,184
RNN simple	(0, 500, 64)	8,256
RNN simple	(0, 64)	8,256
Dense	(0, 1)	65

Taula 7: RNN profunda

4.4 Resultats

A la taula 8 es poden observar els resultats dels entrenaments de les diferents xarxes neuronals recurrents avaluats amb l'accuracy:

Mètode	Paràmetres	Accuracy
RNN simple	165,249	0.50
LSTM simple	85,281	0.70
GRU simple	175,809	0.50
RNN bidireccional	170,497	0.63
LSTM bidireccional	201,601	0.86
GRU bidireccional	191,617	0.86
RNN profunda	181,761	0.50
LSTM profunda	246,849	0.58
GRU profunda	225,729	0.50

Taula 8: Resultats cas pràctic

Les xarxes neuronals bidireccionals amb LSTM o GRU han pogut millorar el rendiment (amb un 0.86 d'accuracy) respecte les xarxes neuronals recurrents simples i profundes on l'accuracy ha estat en alguns casos del 0.50 i per tant no han estat capaçes d'aprendre per a predir el sentiment de la ressenya.

5 Conclusions

L'anàlisi de sentiments amb seqüències de text ha mostrat la seva utilitat davant la impossibilitat de categoritzar manualment les grans quantitats de text que hi ha disponibles actualment. Amb aquest cas pràctic on s'ha treballat amb la base de dades IMDB de la Universitat d'Stanford sobre ressenyes de pel·lícules [5], les xarxes neuronals recurrents han demostrat adaptar-se a les estructures seqüencials dels textos i les relacions d'àmplia distància entre variables per les seves habilitats en capturar patrons complexos, relacions a les dades i informació contextual. S'han adaptat bé al context de les ressenyes i han pogut processar-les sense la necessitat de tenir la mateixa extensió entre elles, sense un pre-processat complex i millorant la classificació mitjançant l'ús de diversos enfocaments innovadors en el camp.

S'ha pogut concloure que els diferents avenços en aquests camps i el desenvolupament de xarxes neuronals recurrents amb cel·les LSTM o GRU (que amb una estructura de cel·la més complexa controlen la propagació del gradient i el flux d'informació) combinades amb arquitectures bidireccionals (que capturen la informació de la seqüència tant del passat com del futur), permeten predir correctament un 86% de les ressenyes amb el seu sentiment associat, millorant un 26% i 16% respecte a la xarxa neuronal recurrent simple.

Referències bibliogràfiques

- [1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Universite de Montreal*, 2014.
- [2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Springer, 2008.
- [3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.
- [4] Tensor Flow Keras. Llibreria python tf.keras.layers. https://www.tensorflow.org/api_docs/python/tf/keras/layers. 2023.
- [5] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. 2011.
- [6] Numpy. Llibreria python numpy. <https://numpy.org>. 2023.
- [7] Anna Bosch Rue, Jordi Casas Roma, and Toni Lozano Bagen. *Deep learning: principios y fundamentos*. UOC, 2019.
- [8] Amit Kumar Sharma, Sandeep Chaurasia, and Devesh Kumar Srivastava. Sentimental short sentences classification by using cnn deep learning model with fine tuned word2vec. *Elsevier*, 2020.