UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

CIÈNCIA DE DADES APLICADA AL CONJUNT DE DADES CLÍNIQUES MIMIC-IV

Autor: Francesc Parcerisas Vela

Director: Dra. Laura Igual Muñoz
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 10 de juny de 2024

Resum

Aquest estudi explora l'ús de models predictius d'aprenentatge automàtic en l'anàlisi de registres clínics de pacients a les unitats de cures intensives. Malgrat les limitacions en recursos computacionals, hem obtingut resultats prometedors en classificació utilitzant una arquitectura transformer [10], amb AUCROC de 0.808 i AUCPRC de 0.404.

Índex

1 Introducció

Fent un repàs de la literatura científica respecte a models predictius d'aprenentatge automàtic, hom pot trobar nombrosos exemples d'apliacions en els camp de la medicina, específicament, en l'anàlisi de registres clínics [9], [14], [17] durant les estades a la unitat de cures intensives (UCI). Aquestes aplicacions han demostrat ser extremadament útils en la detecció de certs trets fenotípics, el càlcul de la probabilitat de supervivència del pacient o l'estimació de la llargada de l'estada hospitalària. Les eines que proporcionen són un complement perfecte pels professionals mèdics als que permeten estalviar temps i realitzar les seves funcions de forma més eficient.

L'objectiu d'aquest treball és el d'examinar quins són aquests models predictius, quins són els conceptes teòrics que hi ha al darrere i quins són els resultats que proporcionen. A banda, també ens proposem entrenar un model que implementi l'arquitectura transformer [8] fent ús de la base de dades MIMIC-IV.

Estructura de la Memòria

Estructurarem aquesta memòria en els blocs següents:

1. En un primer bloc explicarem la composició de la base de dades MIMIC-IV, comunment utilitzada per a l'entrenament de models predictius [9], i n'explorarem les característiques.

2. Veurem quins són els conceptes teòrics en els que es fonamenten aquests models predicius. Concretament explicarem el funcionament de les xarxes neuronals recurrents (RNN) i els transformers.

3. Finalment mirarem quins altres treballs s'han realitzat respecte predicció de la mortalitat amb models d'aprenentatge automàtic i com podem implementar una nova solució prenent com a base la feina feta fins ara [10].

2 Base de dades MIMIC-IV

MIMIC (Medical Information Mart for Intensive Care) és una base de dades relacional que conté dades anonimitadzades d'estades hospitalàries de pacients reals admesos a la Unitat de Cures Intensives (UCI) de l'hospital Beth Israel Deaconess Medical Center de Boston (BIDMC), Massachusetts, Estats Units d'Amèrica. Les dades poden ser estan disponibles de forma pública [1] després de realitzar un curs introductori. En aquesta secció exposarem els asepctes fonamentals d'aquesta base de dades, explicarem els origens de les dades i com estan organitzades.

La versió de la base de dades que farem servir durant aquest treball és la 2.2. de MIMIC-IV, que pren dades d'estades de pacients entre 2008 i 2019 [1]. Aquestes dades s'obtenen de MetaVision, un sistema de gestió d'informació clínica creat per iMDSoft, que emmagatzema i mostra les dades al costat del llit dels pacients. En aquesta última versió de la base de dades s'utilitza únicament MetaVision, a diferència de MIMIC-III, que combina dades d'altres sistemes. A la secció 2.1.3 entrarem una mica més en detall en les diferències que hi ha entre MIMIC-IV i les versions anteriors.

La creació de la base de dades ha estat portada a terme, essencialment, en tres passos [1]:

- *Adquisició*: dels pacients admesos al departament d'emergències o una de les unitats de cures intensives BIDMC, n'extreuen les dades de les bases de dades corresponents en cada cas. Creen una taula mestra amb tots els pacients admesos entre 2008 i 2019 i filtren les taules d'estades hospitalàries únicament per considerar estades d'aquest pacients.
- *Preparació de les dades*: rrealitzen una reorganització del conjunt de dades per a facilitar l'anàlisi de dades retrospectiu. Aquest pas engloba desnormalitació de les taules, eliminació de registres d'auditoria, i reorganització en un nombre menor de taules. Tot i això, han realitzat passos addicionals de neteja de les dades, assegurant així una representació fefaent del conjunt de dades al món real.

[1] `https://physionet.org/content/mimiciv/2.2/icu/#files-panel`

- *Anonimització*: d'acord amb la Llei de Responsbiltiat i Portabilitat de l'Assegurança de Salut (Health Insurance Portability and Accountability Act, HIPAA)[3], qualsevol dada que permeti identificar als pacients ha de ser anonimitzada. Els identificadors reals de les taules han estat substituits per xifres aleatòries, obtenint com a resultat nombres entersn anonimitzats per cada pacient, hospitalització i estada a la UCI. En els casos necessàris, han aplicat un algoritme d'anonimització de text pla a les dades de text pla per eliminar qualsevol tipus d'informació sanitària protegida. Finalment, les dates i les marques temporals han estat intercanviades de forma aleatòria en un punt del futur fent servir un increment en dies. Un únic increment temporal ha estat aplicat a cada pacient, de manera que per cada un d'ells, les dades temporals són internament coherents [1]. És a dir, la diferència entre dos unitats de temps (del mateix pacient) del conjunt de dades originals es conserva en el conjunt de dades anonimitzades. De forma contrària, però, les dades de dos pacients diferents no són comparables pel que fa a unitats temporals. Dos pacients admesos l'any 2130, no necessàriament han estat admesos al mateix any de la realitat .

2.1 Visió general de MIMIC-IV

La versió 2.2. de la base de dades de MIMIC-IV conté dades de 299.712 pacients, 431.231 admissions hospitalàries i 73.181 estades a la unitat de cures intensives. La base de dades està dividida en 5 mòduls en funció de la tipologia i l'origen de les dades. Aquests són:

- *hosp*: Dades generals a nivell hospitalari. Aquestes inclouen: informació general del pacient i les admissions, proves de laboratori, microbiologia i administració de medicació.

- *icu*: Dades d'estades a la unitat de cures intensives. Aquestes dades inclouen les medicions fetes durant les estades a la UCI que provenen del sistema MetaVision de iMDSoft. Les taules de MetaVision han esta desnormalitzades per crear un sistema en forma d'"estrella"amb les taules *icustays* i *d_items* com a eixos centrals [1]. Aquestes s'enllacen amb un conjunt de dades que incorporen el sufix -*events* i que, com el seu nom indica, emmagatzemen dades referents als diferents esdeveniments que succeeixen durant l'estada a la

UCI. Per exemple: *inputevents* inclou les entrades intravenoses i de fluits i, *outputevents* inclou mesures fetes al pacient.

- *ed*: Dades provinents del departament d'emergències (ED) de l'hospital. Algeunes de les dades que incluen són: mesures de signes vitals o raons per a l'admissió i el triatge.
- *cxr*: Taules de cerca i metadades que enllacen MIMIC-IV amb el conjunt de dades d'imatges mèdiques MIMIC-CXR. Les imatges d'aquest conjunt són estudis de radiologies de pit. Aquests estudis poden contenir imatges des de diferents angles i estar acompanyades d'informes de text pla semiestructurat.
- *note*: Anotacions clíniques compostes de text anonimitzat.

A continuació aprofundirem en el contingut dels moduls *hosp* i *icu* que contenen les dades més importants del conjunt i són amb les que utilitzarem al llarg d'aquest treball. Detallem també els volums de les taules i els seus camps més importants.

2.1.1 Mòdul *hosp*

Tal i com ja hem introduït, el mòdul *hosp* conté dades referents als registres mèdics electrònics (Electronic Health Records, EHR) dels pacients. Les medicions que s'hi poden trobar són, majoritàriament, fetes durant l'estada hospitalària, si bé algunes d'elles són fetes fora[1].

Al mòdul *hosp* hi ha 21 taules, entre les quals hi trobem les taules *patients*, on hi ha dades demogràfiques dels pacients; *admissions*, per les hospitalitzacions i; *transfers*, pet les transferències intra-hospitalàries. En aquestes taules (i en d'altres) trobarem els camps *subject_id*, que és un identificador únic, enter, per cada pacient i; *hadm_id*, que és un altre identificador enter que representa una única estada hospitalària corresponent a un pacient.

Com ja hem explicat, les dades temporals, per motius de privacitat, estan anonimitzades. Aquestes es troben a les columnes *anchor_year* i *anchor_year_group*. La columna *anchor_year* conté aquesta dada temporal anonimitzada, en un moment entre els anys 2100 i 2200. La columna *anchor_year_group*, en canvi, conté un rang de tres anys entre 2008 i 2019 en el que va passar l'estada hospitalària a la realitat.

Per exemple, considerem un pacient amb un valor a *anchor_year* de 2123 i a *anchor_year_group*, de 2010-2012. Aquesta estada de 2123, realment ha tingut lloc en algun punt entre els anys 2010 i 2012. A més, la columna *anchor_age* conté l'edat del pacient l'any *anchor_year*, si bé els pacients més grans de 89 anys, apareixen amb un valor de 91.

Altres informacions que podem trobar en aquest mòdul són medicions de laboratori, a les taules *labevents* i *d_labitems*; cultius microbiològics, a *microbiologyevents* i *d_micro*; administració de medicació, a *emar* i *emar_detail*; prescripcions mèdiques, a *prescriptions* i *pharmacy*; informació hospitalària, a *diagnoses_icd*, *d_icd_diagnoses*, *procedures_icd*, *d_icd_procedures*, *hcpcsevents*, *d_hcps* i *drgcodes*; registres mèdics en línia, a *omr* i; informació de servei, a *services*. A la Taula 1 podem veuer detallades les dimensions de cadascuna de les taules més importants del mòdul *hosp*.

Taula	Dimensions	Contingut
admissions	(431.231, 16)	Informació sobre estades de cada pacient
diagnoses_icd	(4.756.326, 5)	Diagnòstic de cada admissió hospitalària
labevents	(58.131.956, 16)	Resultats de proves de laboratori de cada pacient
microbiologyevents	(1.396.224, 25)	Informació sobre cultius microbiològics
patients	(299.712, 6)	Dades demogràfiques de cada pacient

Taula 1: Resum de les taules més importants del mòdul *hosp*.

2.1.2 Mòdul *icu*

El mòdul *icu* conté les dades referents a les estades a la unitat de cures intensives extretes dels monitors MetaVision. La taula *icustays* conté les dades generals d'aquestes estades, mentre que cada taula amb el sufix -*events* conté dades d'algún tipus d'esdeveniment: *datetimeevents*, totes les dades temporals de medicions a la UCI; *ingredientevents*, ingredients de les administracions que es fan al pacient; *inputevents*, administracions que es fan als pacients; *outputevents*, medicions realitzades als pacients i; *procedureevents*, procediments duts a terme durant l'estada a la UCI. A *chartevents* hi trobem totes les dades de pacients durant l'estada a la UCI.

A la Taula 2 mostrem les dimensions de les taules principals del mòdul *icu*. En capítols posteriors entrarem en detall en els valors que hi ha en aquestes taules.

2.1.3 Diferències amb versions anteriors de MIMIC

Respecte la seva predecessora MIMIC-III, MIMIC-IV incorpora certes millores, sobretot pel que fa a simplificació de l'estructura i l'incorporació de nous elements que ajuden a enriquir les dades [5].

Els origens de les dades ara són més actualitzats, contenen dades compreses entre els anys 2008 i 2019, en contraposició a MIMIC-III que utilitzava dades dels anys 2001 a 2012. En versions anteriors s'extreien les dades, a banda del sistema MetaVision, del sistema CareVue. Ara únicament queda MetaVision, cosa que simplifica l'estructura eliminant les taules referents a CareVue.

L'estructura de la base de dades ha estat modificada per separar-la en mòduls i identificar els origens de cada tipus de dades. A MIMIC-III, en canvi, totes les taules formaven part d'un sol conjunt no diferenciat. Per exemple, en aquesta nova versió, es presenta la taula *chartevents* al mòdul *icu* per què conté únicament dades de les estades a la UCI; *labevents* té un origen hospitalari, tot i contenir dades de medicions i, per tant, es presenta de forma separada.

Un detall complet de tots els canvis realitzats al llarg de les diferents versions de MIMIC, així com de les diferents actualitzacions que ha sofert, pot ser trobat a

Taula	Dimensions	Contingut
chartevents	(313.645.063, 11)	Total de dades de pacients durant l'estada a la UCI
icustays	(73.181, 8)	Informació sobre les estades a la UCI de cada pacient
datetimeevents	(7.112.999, 10)	Dades temporals de medicions a la UCI per cada pacient
ingredientevents	(12.229.408, 17)	I Ingredients de les administricacions realitzades a la UCI
inputevents	(8.978.893, 25)	Administracions realitzades a la UCI per cada pacient
outputevents	(4.234.967, 9)	Medicions realitzades a la UCI per cada pacient
procedureevents	(696.092, 22)	Procediment realitzats durant l'estada a la UCI de cada pacient

Taula 2: Resum de les taules més importants del mòdul *icu*.

la pàgina de la documentació [2].

2.2 Anàlisi exploratori de les dades

A continuació ens proposem analitzar el conjunt de dades MIMIC-IV, explorant-ne les característiques més importatns i com estan distribuïdes segons les diferents tipologies de pacient. En particular, volem posar atenció en la llargada de l'estada a la UCI (Length Of Stay, LOS) i la mortalitat.

Comencem explorant el mòdul *hosp* i com estan distribuïdes les edats dels pa-

[2] `https://mimic.mit.edu/docs/iv/about/changelog/`

cients. A la Figura 1 podem veure el nombre de pacients per edat, la variable *anchor_age* de la taula *patients*. Observem com no hi ha entrades de pacients amb edat < 18 i comja hem explicat, si tenen edat ≥ 89, tenen per defecte el valor 91.

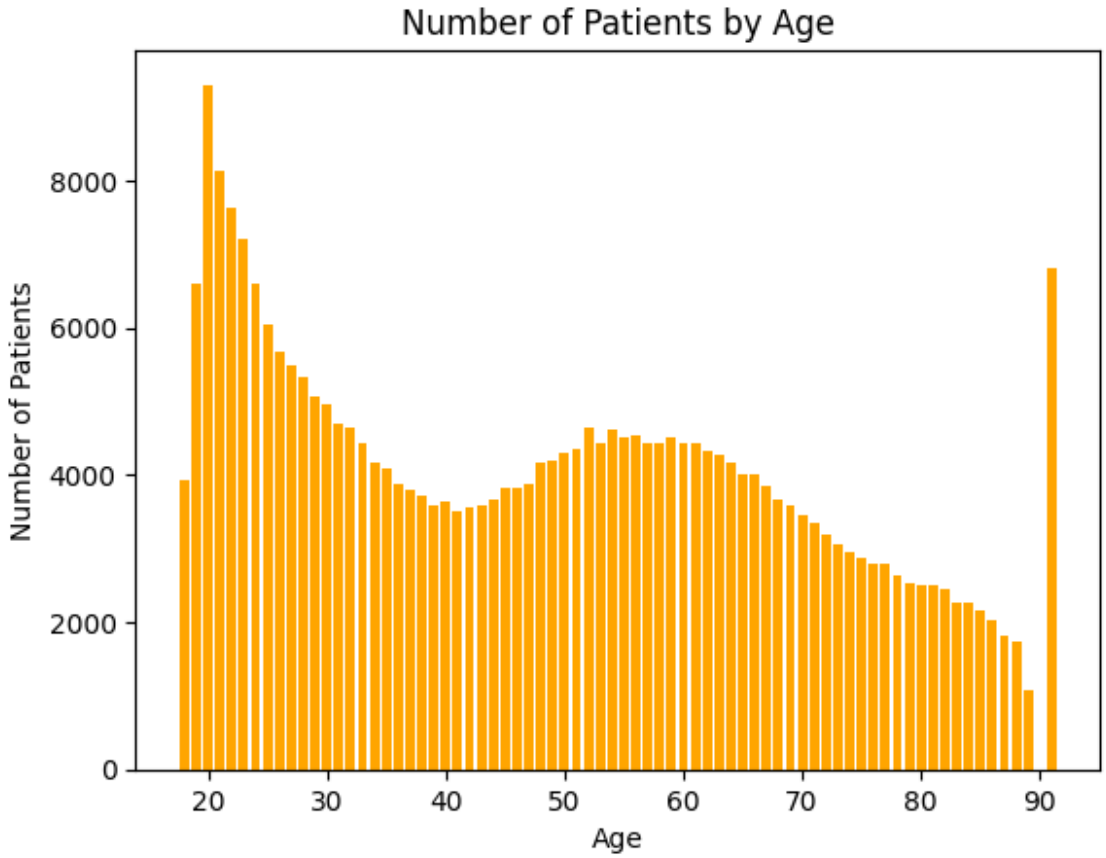


Figura 1: Nombre de pacients per edat

Podem considerar, també, veure la mortalitat en funció del rang d'edat mirant la columna *dod* de la taula *patients*. Dels 299.712 pacients que hi al conjunt de dades, un 90%, (270.636) no han tingut mort intrahospitalària. Aquí cal notar que *dod*, a MIMIC-IV, només indica morts que han passat durant l 'estada a l'hospital, a diferència d'altres versions on també s'hi recollien morts després de l'alta hospitalària. A la Figura 2 podem veure com tal i com indica la intuició, la mortalitat augmenta quan agumenta l'edat dels pacients.

Per acabar aquest anàlisi inicial per edats dels pacients, midem com es comporta la llargada l'estada a la UCI (LOS). A la Figura 3, mostrem un gràfic de dispersió que mostra aquesta relació. Podem veure també, que a nivell global, l'estada mitjana és de 3,45 dies amb una desviació estàndard de 4,92. Tenim una mediana de

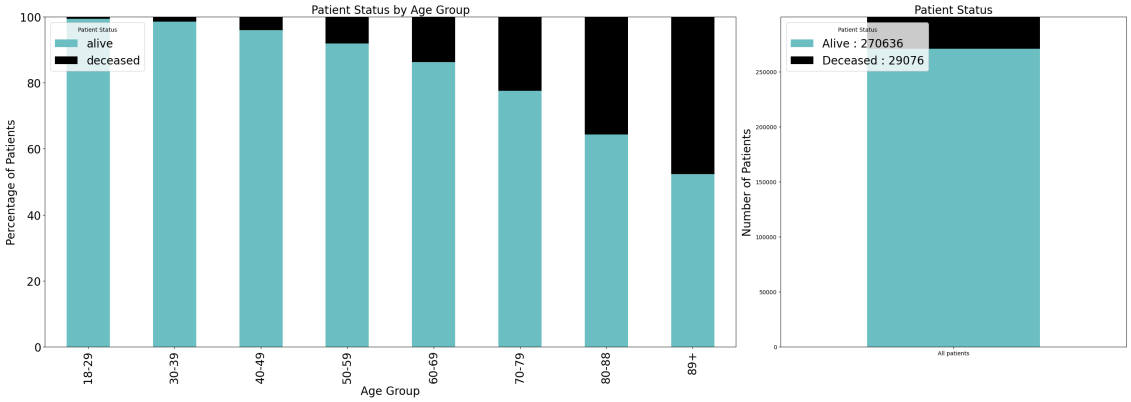


Figura 2: A l'esquerra: nombre de pacients i mortalitat per rang d'edat. A la dreta: mortalitat total del conjunt de dades.

1.92 dies. Podem concloure, per tant, que en general l'estades a la UCI són curtes, de menys d'una setmana, tot i que en el conjunt de dades en podem trobar de més de 100 dies.

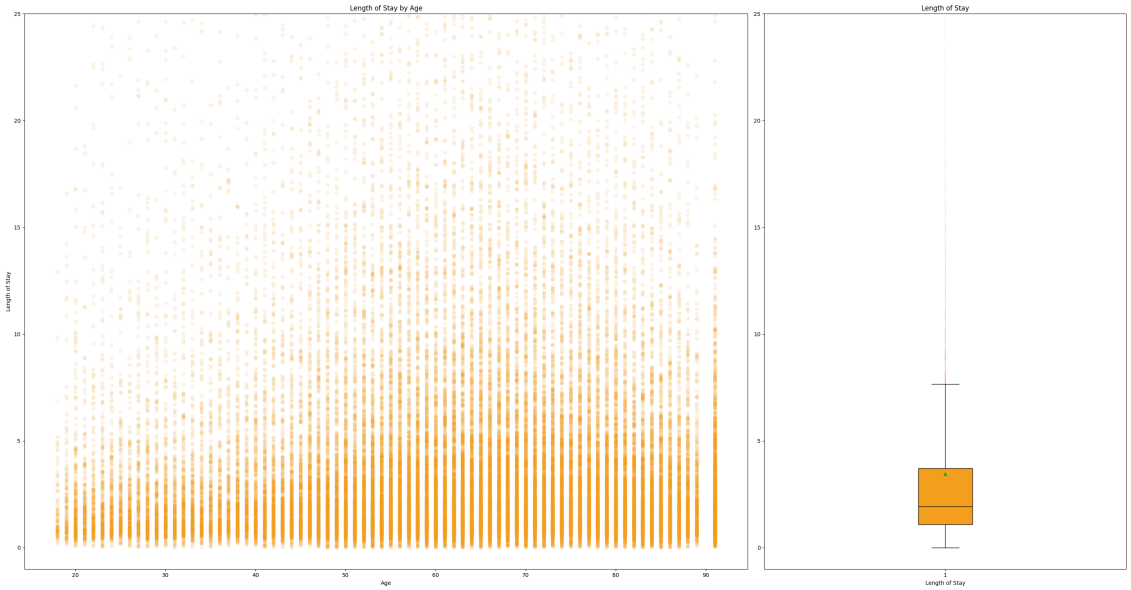


Figura 3: LOS per edat. A l'esquerra: diagrama de dispersió del LOS en funció de l'edat. A la dreta: diagrama de caixa del LOS total del conjunt de dades.

A banda de fixar-nos en l'edat dels pacients també podem fixar-nos en el seu diagnòstic. A la taula *diagnoses_icd* tenim el diagnòstic per cada pacient i admissió hospitalària. A la columna *icd_code* hi ha un codi segons ICD (International Coding Definitions) que representa un diagnòstic concret, juntament amb la versió d'aquesta

icd_version	diagnòstics diferents	diagnòstics totals
9	9072	2766877
10	16757	1989449

Taula 3: Nombre de diagnòstics per versió d'ICD a MIMIC-IV

codificació *icd_version*. Al conjunt de dades, tenim codificacions 9 i 10. A la taula 3 veiem quants diagnòstics tenim per cada versió ICD.

Cada admissió hospitalària pot tenir més d'un diagnòstic, ordenats amb ordre de prioritat *seq_num*. Aquest ordre de priporitat és indicatiu sobretot per valors baixos, per valors alts, perd significància. Per una mateixa admissió, un diagnòstic amb $seq\_num = 6$, pot no ser més prioritari que un diagnòstic amb $seq\_num = 5$ [5]. Els codis ICD poden ser classificats en capítols [6], denotant un rang més general de diagnòstics. A la Figura 4 veiem quantes admissions hi ha per capítol de la versió ICD = 9 amb un $seq\_num = 1$ (el més prioritari). També hi podem veure la mortalitat en funció d'aquests rangs. Per fer aquest càlcul considerem la mort del pacient si ha mort durant l'admissió que té aquell diagnòstic.

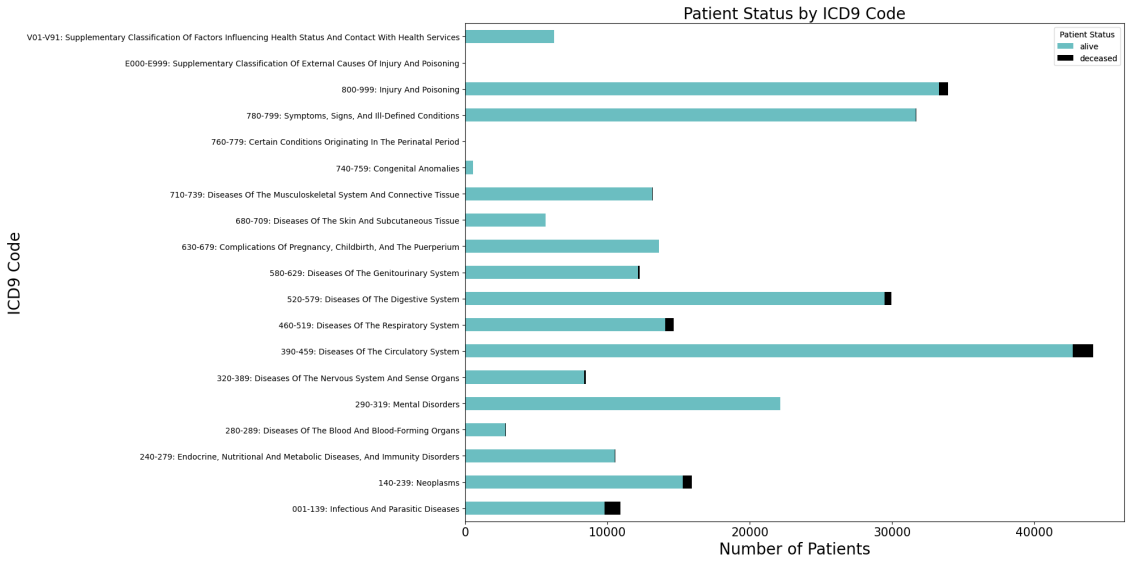


Figura 4: Mortalitat per diagnòstic, basat en ICD9.

Si bé hi ha alguns d'aquests conjunts de diagnòstics amb definicions vagues, permet fer-nos una idea de com està distribuït el conjunt de dades. Veiem que els

dos diagnòstics més freqüents són: Malalties del Sistema Circulatori (390-459) i Lesions i Intoxicació (800-999). Pel que fa a la mortalitat, podem veure que una part significativa de les morts venen per: Malalties del Sistema Circulatori (390-459) i Infeccions i Malalties Parasitàries (001-139).

3 Introducció als models predictius

En aquesta secció ens proposem fer un repàs a través de l'evolució dels diferents models predictius d'aprenentatge automàtic que s'han estat utilitzat i se segueixen utilitzant. En particular, pretenem introduïr aquells conceptes que avui en dia serveixen de base pels models més avançats que formen part de l'*state of the art* d'aquest camp. Un d'aquests models són els LLM (Large Language Models o Models de Llenguatge Extens) com GPT (Generative Pre-Trained Transformers o Transormadors Generatius Preentrenats) que gaudeixen d'àmplia popularitat en el moment d'escriure aquesta memòria.

En general, però, tot i l'interés que desperten aquestes tecnologies i la gran diversitat d'aplicacions pràctiques que implementen, en aquest treball volem posar atenció sobretot en les tasques de classificació.

3.1 Conceptes previs

Abans d'entrar en matèria amb els models de predicció, ens veiem amb la necessitat d'explicar una sèrie de conceptes teòrics previs.

Definició 3.1. *Sigui* $(\Omega, \mathcal{F}, \{f_\theta \in \Theta\})$ *un espai de probabilitats paramètric, amb* f_θ *funcions de densitat, i* $X = (X_1, ...X_n)$ *observacions independents d'una variable aleatòria. Llavors, si les funcions* f_θ *són contínues, definim la* ***funció de versemblança*** *com*

$$\begin{aligned} \mathcal{L} : \Omega \times \Theta &\longrightarrow \mathbb{R}_+ \\ \mathcal{L}(\theta) &= \mathcal{L}(\theta; x_1, \ldots, x_n) \\ &= f_{X_1,\ldots,X_n}(\theta; x_1, \ldots, x_n) \\ &= \prod_{i=1}^{n} f_{X_i}(x_i; \theta) \end{aligned} \tag{3.1}$$

Informalment, la funció de versemblança ens indica com de plausible és obtenir aquestes observacions donat un determinat $\theta \in \Theta$*. Podem extendre aquesta definició a funcions no contínues.*

A continuació introdïm les definicions de tres funcions d'activació que són utilitzades en models d'aprenentatge: la funció logística, la funció ReLU i la funció tangent hipebòlica.

Definició 3.2. *Sigui* $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *una funció (habitualment polinòmica de grau 1), llavors definim la* ***funció logística*** *com a un subtipus de funció sigmoidea* $\sigma(x) : \mathbb{R}^n \longrightarrow [0, 1]$ *:*

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}} \tag{3.2}$$

Fixem-nos que aquesta funció únicament pren valors entre 0 i 1. Això ens resultarà especialment útil per interpretar el resultat com a "percentatge de"quan es múltiplica el valor per un altre terme. Ho veurem a la secció 3.4.

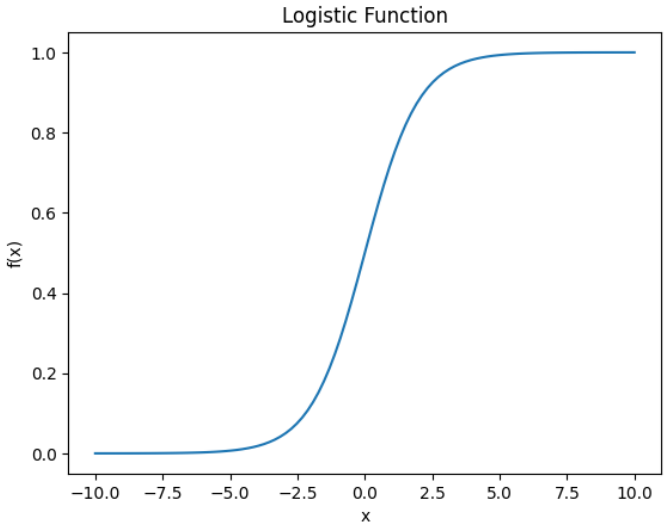


Figura 5: Exemple de funció logística

Definició 3.3. *Definim la funció* ***ReLU*** *(Rectified Linear Unit) com*

$$f : \mathbb{R} \longrightarrow \mathbb{R}_+$$
$$f(x) = \max(x, 0) = \begin{cases} 0 & si\ x \leq 0 \\ x & si\ x > 0 \end{cases}. \tag{3.3}$$

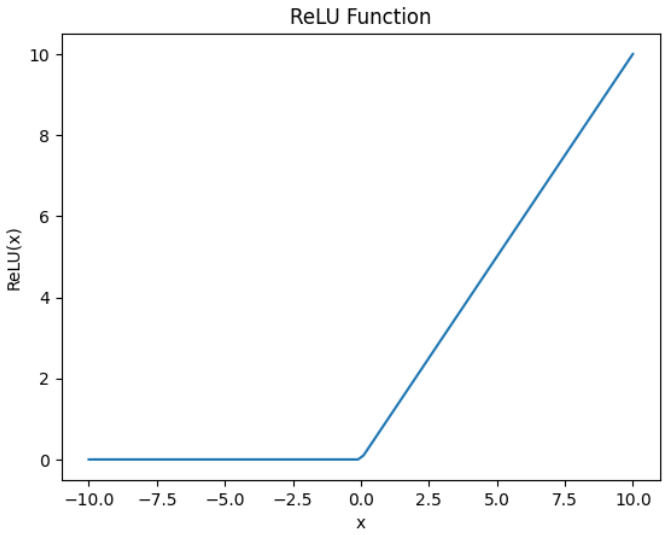


Figura 6: Exemple de funció ReLU

Definició 3.4. *Definim la funció* ***tangent hiperbòlica*** *com*

$$\begin{aligned} \tanh : \mathbb{R} &\longrightarrow [-1, 1] \\ \tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}}. \end{aligned} \tag{3.4}$$

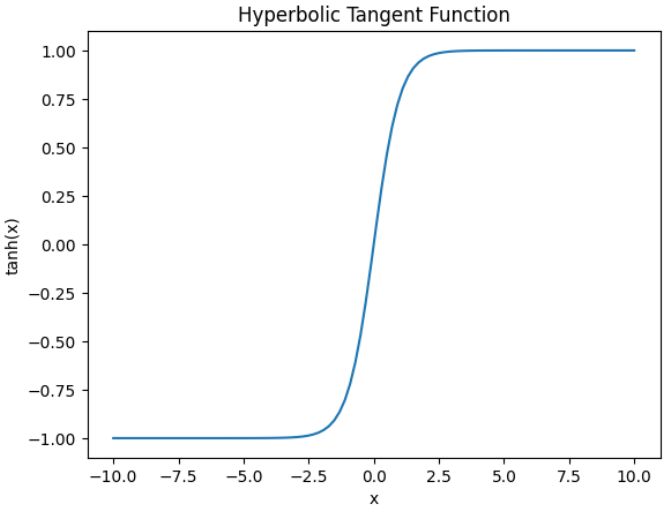


Figura 7: Exemple de funció tanh

Observació 1. Fixem-nos que per $f(x) = x$, tenim $\tanh(x) = 2 \cdot \sigma(2x) - 1$.

3.2 Regressió Logística (LR)

El model de Regressió Logística (Logistic Regression, LR) com a classificador binari pretén predir quina etiqueta escau més a un vector de variables d'entrada $x = (x_1, ..., x_n)$ amb l'ajuda d'una funció logística. Per fer-ho, assigna etiquetes 0 i 1 que es corresponen amb els valors extrems de la funció logística.

És a dir, considerem la funció $f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$ definida com

$$f(x) = b + \sum_{i=1}^{n} w_i x_i, \tag{3.5}$$

on $b, w_i, ..., w_n \in \mathbb{R}$. Anomenarem biax a b i pesos als w_i. Fem servir aquesta funció a la funció logística $\sigma(x) = \dfrac{1}{1 + e^{-f(x)}}$ per determinar la versemblança de que la variable d'entrada $x \in \mathbb{R}^n$ pertanyi o no a la classe etiquetada amb 1.

A partir d'aquí, ens trobem amb el problema de determinar quins són els millors paràmetres (pesos i biaixos) que permeten determinar millor l'etiqueta per a cada variable d'entrada. Existeixen diversos mètodes que troben aquests paràmetres òtpims. Aquí proposem el mètode de descens del gradient minimitzant la funció de pèrdua d'entropia creuada mitjana.

Suposem que tenim una mostra d'm observacions $x_1, ..., x_m$ i les seves etiquetes corresponents $y_1, ..., y_m \in \{0, 1\}$, i posem $\theta = (b, w_i, ..., w_n)$ pels paràmetres, llavors podem definir la funció de pèrdua d'entropia creuada mitjana com

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left(y_i \log(\sigma(x_i)) + (1 - y_i) \log(1 - \sigma(x_i)) \right). \tag{3.6}$$

i com ja hem dit volem trobar θ que faci aquesta funció mínima calculant a cada pas el gradient de $\mathcal{L}$ respecte θ.

3.3 Xarxes Neuronals Recurrents (RNN)

Suposem que com a variables d'entrada del model predictiu tenim dades de longitud diferent i en les que l'ordre és important. Per exemple una evolució temporal de valors com pot ser el canvi en el preu d'un producte financer o, en el cas d'aquest

treball, les diferents mesures mèdiques fetes durant l'estada hospitalària.

En aquests casos, no podem fer servir models senzills com la regressió logística o les xarxes neurnoals "estàndard". Necessitem fer servir algún tipus de xarxa que tingui en compte els valors passats. És a dir, que tingui memòria. Un cas de model que compleix aquestes característiques són les Xarxes Neuronals Recurrents (Recurrent Neural Networks, RNN). En aquesta explicació i en les posteriors assumirem que el lector està familiaritzat amb els conceptes bàsics de les xarxes neuronals com són el concepte de funció d'activació (si bé a 3.1 hem definit un parell d'elles) i retropropagació.

Considerem una seqüencia d'entrada $x_1, ..., x_k$ ordenada amb $x_i \in \mathcal{R}$. Per simplificar l'explicació, assumim que les variables d'entrada són unidimensionals. A cada temps t calculem la sortida h_{t+1} (anomenada estat ocult a temps t) a partir dels pesos w i els biaxios b però a diferència de les xarxes neuronals estàndard, aquí incorporem al càlcul el valor de la capa anterior h_t. És a dir, h_{t+1} té dependència de x_t però també de h_t. Formalment,

$$h_{t+1} = \sigma(f(x_t, h_t)), \tag{3.7}$$

on σ és una funció d'activació i f, una funció lineal amb pesos w_h, w_x i biaix b_k:

$$f(x_t, h_t) = w_h h_t + w_x x_t + b. \tag{3.8}$$

Notem que els pesos i els biaixos són els mateixos per cada temps t.

Tal i com es fa amb les xarxes neuronals estàndard, aquests pesos s'actualitzen amb retropropagació minimitzant l'error mitjançant descens del gradient.

3.3.1 Problema de l'explosió i desaparicíó del gradient

Les xarxes neuronals recurrents presenten un problema: el de l'explosió/desaparició del gradient. Considerem la xarxa neuronal que hem definit a 3.3. A cada pas de recurrència calculem el gradient d'una certa funció de pèrdua $\mathcal{L}(W)$ respecte el conjunt de pesos W:

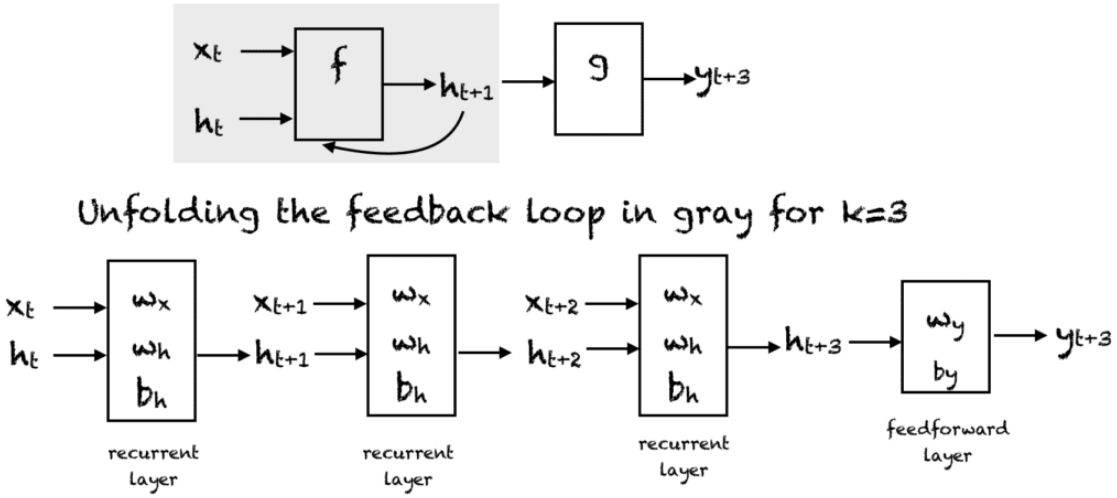


Figura 8: Configuració d'una RNN amb $k = 3$. A dalt sense desplegar. A baix desplegada. Imatge extreta de [2]

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{k} \frac{\partial \mathcal{L}_t}{\partial W}. \tag{3.9}$$

A partir de la retropropagació, a cada pas actualitzem el valor de W fent

$$W \leftarrow W - \alpha \left(\frac{\partial \mathcal{L}}{\partial W} \right). \tag{3.10}$$

A més, a cada temps t, podem posar la derivada parcial $\frac{\partial \mathcal{L}_t}{\partial W}$ com

$$\begin{aligned} \frac{\partial \mathcal{L}_t}{\partial W} &= \frac{\partial \mathcal{L}_t}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W} = \\ &= \frac{\partial \mathcal{L}_t}{\partial h_t} \prod_{s=2}^{t} \left(\frac{\partial h_s}{\partial h_{s-1}} \right) \frac{\partial h_1}{\partial W} \end{aligned} \tag{3.11}$$

i, com que $h_s = \sigma(f(x_{s-1}, h_{s-1})) = \sigma(w_h h_{s-1} + w_x x_{s-1} + b)$, podem calcular la derivada de h_s respecte h_{s-1}, aplicant la regla de la cadena, com

$$\begin{aligned} \frac{\partial h_s}{\partial h_{s-1}} =& \sigma'(w_h h_{s-1} + w_x x_{s-1} + b) \frac{\partial}{\partial h_{s-1}} (w_h h_{s-1} + w_x x_{s-1} + b) = \\ =& \sigma'(w_h h_{s-1} + w_x x_{s-1} + b) w_h. \end{aligned} \tag{3.12}$$

Substituint (3.12) a (3.11), tenim

$$\frac{\partial \mathcal{L}_t}{\partial W} = \frac{\partial \mathcal{L}_t}{\partial h_t} \prod_{s=2}^{t} (\sigma'(w_h h_{s-1} + w_x x_{s-1} + b) w_h) \frac{\partial h_1}{\partial W}. \tag{3.13}$$

Això provoca que si $\sigma'(w_h h_{s-1} + w_x x_{s-1} + b) w_h \leq 1$, el gradient tendirà a 0 ràpidament per valors grans de t, fent que pas de retropropagació sigui ínfim i reduïnt la convergència al resultat desitjat. A aquest fenòmen l'anomenem desaparició del gradient.

Si, en canvi $\sigma'(w_h h_{s-1} + w_x x_{s-1} + b) w_h$ és un nombre gran (cosa que pot succeïr per valors grans de w_h) el gradient creixarà ràpidament generant el fenòmen de l'explosió del gradient.

El problema de l'explosió del gradient fa que les xarxes neuronals recurrents estàndard estiguin en desús. En canvi, van sorgir noves tècniques que permetien mitigar els efectes d'aquest problema. Una d'aquestes tènciques són els LSTMs [19], que introduïm a continuació.

3.4 Long short-term memory (LSTM)

Les unitats LSTM (Long short-term memory)[19] són una mica més complexes que les de les RNN estàndard i, a cada pas, a banda de connectar-se entre elles a través dels estats ocults h_t, també es connecten a través de la memòria a llarg termini. A més, cada unitat, conté quatre capes de recurrència (cadascuna amb els seus pesos i biaxos), a diferència de les RNN que únicament en tenien una.

Prenem l'exemple de la unitat LSTM central de la RNN de la figura 9. Tenim dues entrades de memòria (le de llarg termini a dalt i la de curt termini abaix. Com a sortides, també tenim la memòria a llarg termini i una memòria de curt termini que es correspon amb l 'estat ocult h_t. Sigui x_t l'observació d'entrada de temps t; W_f, W_c, W_i i W_o els pesos de cada una de les capes de recurrència; c_t, la memòria a llarg termini a temps t ; h_t, la memòria a curt termini o estat ocult a temps t i; $[h_{t-1}, x_t]$, el vector d'entrada a temps t. Llavors, tenim les quatre capes de recurrència següents representades a les figures 9, 10, 11, 12 amb les seves funcions d'activació. Són les següents:

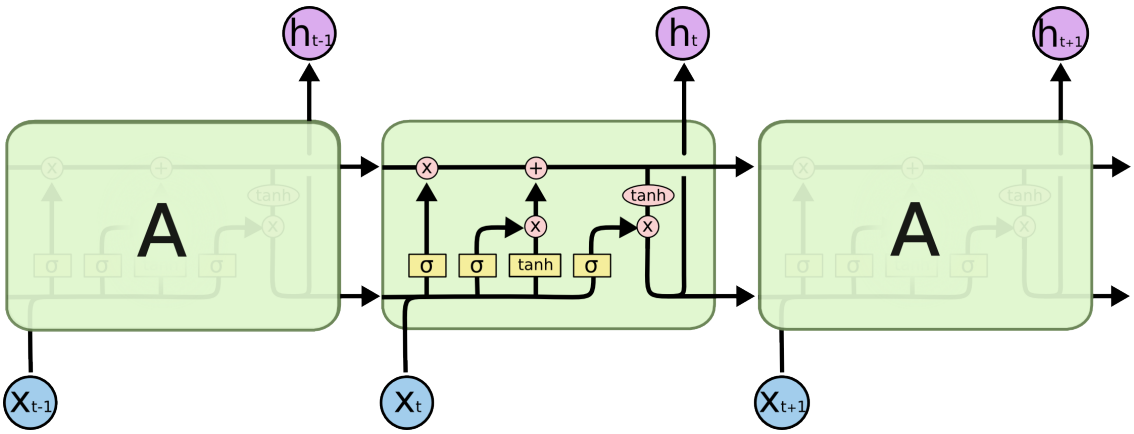


Figura 9: RNN amb unitats LSTM. Imatge extreta de [4]

- Porta d'oblit (*forget gate*): Prenent com a referència la figura 10, la sortida és:

$$f_t = \sigma(W_f[h_{t-1}, x_t]), \tag{3.14}$$

La forma d'interpretar aquest valor f_t és el percentatge de la memòria a llarg termini de la unitat anterior c_{t-1} que passarà. Recordem que la funció σ pren valors entre 0 i 1.

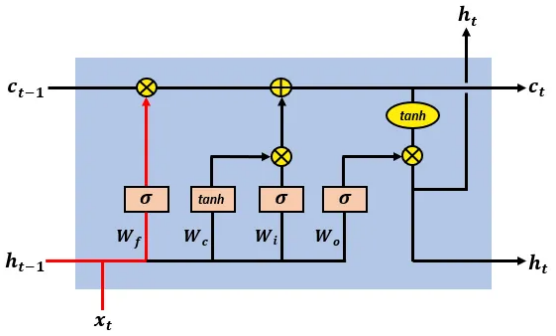


Figura 10: *Forget gate*. Imatge extreta de [7]

- Porta d'entrada (*input gate*): Prenem la figura 11. La porta d'entrada correspon a les capes 2 i 3. Obtenim els valors:

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t]) \tag{3.15}$$

i,

$$i_t = \tanh(W_i[h_{t-1}, x_t]). \tag{3.16}$$

Posteriorment calculem el producte $\tilde{c}_t \times i_t$. És a dir la unitat amb tanh calcula el valor que es sumarà a la memòria de llarg termini.

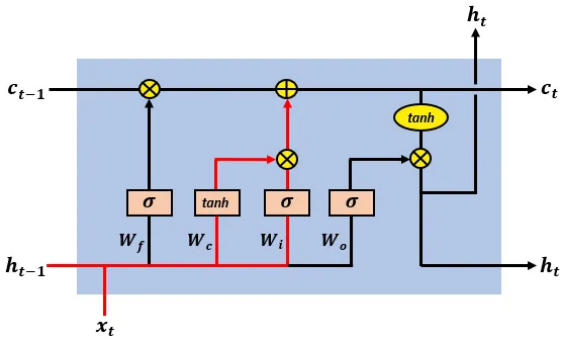


Figura 11: *Input gate*. Imatge extreta de [7]

- Porta de sortida (*output gate*): figura 12. Calculem el valor

$$o_t = \tanh(W_o[h_{t-1}, x_t]). \tag{3.17}$$

És a dir, calculem el percentatge de memòria a llarg termini que esdevindrà memòria a curt termini.

Finalment, calculem les sortides: la memòria a llarg termini

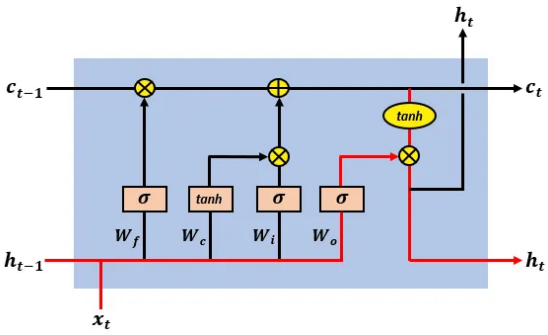


Figura 12: *Output gate*. Imatge extreta de [7]

$$c_t = c_{t-1} \times f_t + \tilde{c}_t \times i_t \tag{3.18}$$

i la memòria a curt termini o estat ocult

$$h_t = o_t \times \tanh(c_t). \tag{3.19}$$

A continuació aprotem un parell de pinzellades de com resolen els LSTMs el problema de la desaparició/explosió del gradient. Una explicació una mica més detallada pot ser trobada a l'article divulgatiu d'Arbel, N. [7].

Recordem que si tenim una RNN, podem calcular el gradient d'una funció de pèrdua respecte els pesos W com

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_{t=1}^{k} \frac{\partial \mathcal{L}_t}{\partial W}. \tag{3.20}$$

A cada temps t, tenim la derivada parcial

$$\begin{aligned} \frac{\partial \mathcal{L}_t}{\partial W} &= \frac{\partial \mathcal{L}_t}{\partial h_t}\frac{\partial h_t}{\partial c_t}\cdots\frac{\partial c_2}{\partial c_1}\frac{\partial c_1}{\partial W} = \\ &= \frac{\partial \mathcal{L}_t}{\partial h_t}\frac{\partial h_t}{\partial c_t}\prod_{s=2}^{t}\left(\frac{\partial c_s}{\partial c_{s-1}}\right)\frac{\partial c_1}{\partial W}. \end{aligned} \tag{3.21}$$

Anàlogament al que hem vist amb RNN tradicionals, és el productori central el que que fa que el gradient pugui desaparèixer. Calculem quins són els valors dels $\dfrac{\partial c_s}{\partial c_{s-1}}$ en el cas d'LSTMs. Recodem que $c_t = c_{t-1} \times f_t + \tilde{c}_t \times i_t$. Llavors obtenim

$$\frac{\partial c_s}{\partial c_{s-1}} = A_t + B_t + C_t + D_t, \tag{3.22}$$

on

$$\begin{aligned} A_t =& \sigma'(W_f[h_{t-1}, x_t])W_f o_{t-1} \times \tanh'(c_{t-1})c_{t-1}, \\ B_t =& f_t, \\ C_t =& \sigma'(W_i[h_{t-1}, x_t])W_i o_{t-1} \times \tanh'(c_{t-1})\tilde{c}_t, \\ D_t =& \sigma'(W_c[h_{t-1}, x_t])W_c o_{t-1} \times \tanh'(c_{t-1})i_t. \end{aligned} \tag{3.23}$$

De forma col·loquial, el fet de tenir un gradient composat com a suma de quatre

termes dificulta que aquests es facin tant petits que redueixin el gradient i desapareixi. Tal i com està dissenyada la unitat LSTM, aquests termes es compensen entre ells i per tant el productori $\prod_{s=2}^{t} \left(\frac{\partial c_s}{\partial c_{s-1}} \right)$ no tendeix a 0 per valors grans de k. Com diem, a l'article [7] hi ha més detall en els càlculs i la demostració.

El que ens interessa saber és que podem afegir un component de memòria a les unitats de les RNN per mitigar els efectes del problema de la desaparició del gradient i crear xarxes més grans i complexes.

3.5 Transformers

A l'article *Attention is all you need* de Vaswani, A. et al. [8], se'ns introduïa per primer cop una arquitectura de xarxa neuronal que bategen amb el nom de Transformer. Aquesta arquitectura, originalment, estava enfocada a tasques de processament de llenguatge per la traducció i, proposava un nou model de xarxa neuronal sense fer ús de recurrència. Aquest prescindir de recurrència, permet no haver d'introduir les dades d'entrada de forma seqüencial, com feiem amb RNNs i LSTMs, fent l'entrenament molt més paral·lelitzable i, per tant, reduïnt el temps d'execució del procés de forma significativa. Al llarg d'aquesta secció detallarem el model descrit per Vaswani, enfocat a llenguatge, i que posteriorment serà adaptat per Khader, F et al. per ser aplicat a dades clíniques multimodals [10].

En aquest model, que segueix una estructura *encoder-decoder*, el codificador mapeja una seqüència d'entrada de representacions simbòliques $(x_1, ..., x_n)$ a una seqüència de representacions contínues $\mathbf{z} = (z_1, ..., z_n)$, un element cada vegada. A la figura 13, podem veure un esquema general d'aquesta arquitectura on tant el codificador com el descodificador fan servir capes de *self-attention* i *fully-connected* punt a punt apilades.

Les piles *encoder* i *decoder* que proposa Vaswni són:

- *Encoder*: Composat de $N = 6$ piles de capes idèntiques amb dues subcapes cada una: la primera un mecanisme de *self-attention multi-head* i la segona una *feed-forward*. Després de cada capa, també, s'aplica una connexió residual conjuntament amb una normalització.

- *Decoder*: Composat també de $N = 6$ capes identìques i dues subcapes com a l'*encoder* i, una tercera capa d'enmascarament que assegura que les prediccions de la capa i depenguin únicament de les sortides connegudes a posicions $\leq i$.

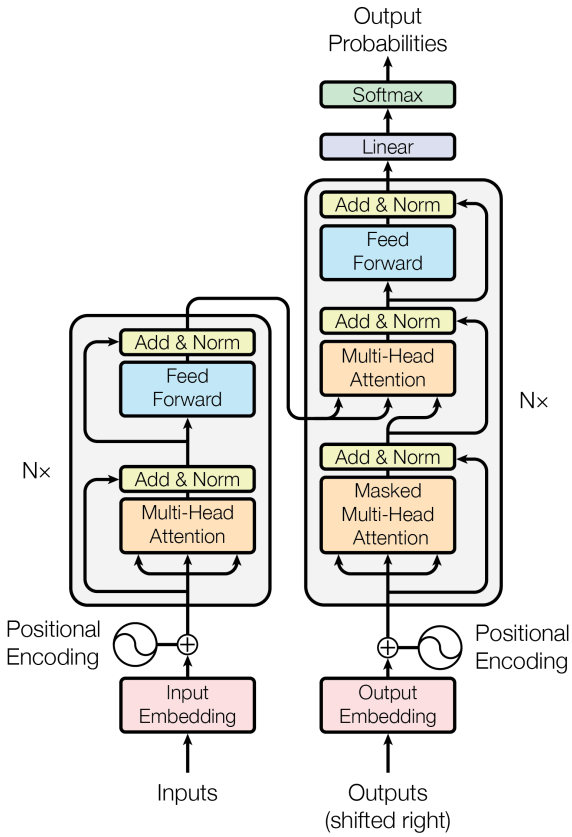


Figura 13: Arquitectura Transformer. A l'esquerra l'*encoder*, a la dreta el *decoder*. Imatge extreta de [8].

En termes generals podem dir que una funció d'atenció és aquella que proporciona una sortida vectoraial per cada terna (*query*, *key*, *value*), on *query*, *key* i *value* són també vectors [8]. Vegem que aquests noms ens evoquen conceptes d'ús de bases de dades amb els que ja estem familiaritzats.

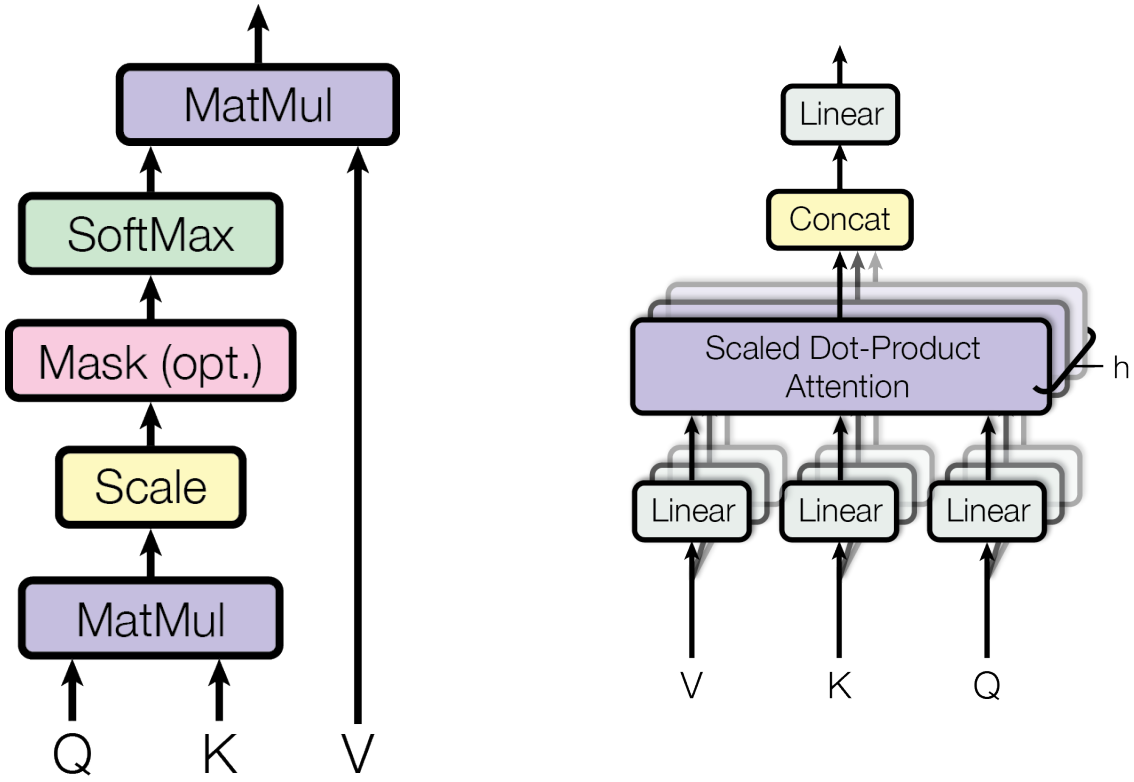
Vaswani et al. proposen dos tipus d'atenció:

- Atenció escalada de producte escalar: siguin Q, K matrius composades per vectors de mida d_k, corresponents a les *queries* i les *keys*, i sigui V una matriu composada per vectors de mida d_v, corresponent als *values*. Llavors, la sortida generada per l'atenció escalada de producte escalar és

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{3.24}$$

  Una representació d'aquesta funció està a 14a.

- Atenció *multi-head*: en lloc de fer servir una única capa d'atenció de producte escalar per vectors *query*, *key* i *value* de mida d_{model}, decideixen projectar



(a) Atenció escalada de producte escalar

(b) Atenció *multi-head*

Figura 14: Funcions d'atenció. Imatges extretes de [8]

linealment cada vector a h vectors de mida d_k i mida d_v, aplicar-los una atenció de producte escalar i concatenar-ne les sortides. Formalment, la sortida de l'atenció multihead és

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_n)W^O, \tag{3.25}$$

on $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, $W_i^Q, W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ i $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

A la figura 14b podem veure una representació d'aquesta configuració.

Seguint l'estructura que podem veure representada a 13, l'altre subcapa que fa servir l'arquitectura Transformer és una *feed-forward fully-connected* aplicada a cada posició de forma independent. Formalment,

$$\text{FFN}(x) = \max(0, xW_+b_1)W_2 + b_2. \tag{3.26}$$

Finalment, també afegeixen una transformació lineal i passen les sortides per

una funció softmax per tal d'obtenir valors de probabilitat.

La capa que ens ha mancat explicar és la de codificació posicional (*positional encoding* a 13). Degut a prescinidir de recurrència i convolucions, l'arquitectura Transformer necessita una forma d'injectar informació sobre la posició en la que es troba una determinada entrada [8]. És per això que per cada posicio pos, sumen a l'entrada incrustada de mida i,

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}}) \tag{3.27}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}}) \tag{3.28}$$

3.6 Mètriques d'avaluació

En aquesta secció, adjuntem l'explicació de les mètriques AUC-ROC i AUC-PRC, que són comunment utilitzades per avaular tasques de classificació binàries (i poden ser exteses a tasques de classificació multimodal). AUC (**A**rea **U**nder the **C**urve), com el seu nom indica és el valor de l'àrea sota la corba, en aquest cas les corbes ROC i PRC:

- ROC (**R**eceiver **O**perating **C**haracteristic) és una corba que representa la VPR (raó de veritables positius) en funció de la FPR (raó de falsos negatius). VPR i FPR es defineixen com:
  - $VPR = \dfrac{VP}{VP + FN}$, on VP és el nombre de positius predits correctament i FN el nombre de falsos negatius. És a dir, el denominador és el nombre total de positius reals.
  - $FPR = \dfrac{VN}{VN + FP}$, on VN és el nombre de negatius predits correctament i FP el nombre de falsos positius. És a dir, el denominador és el nomber total de negatius reals.
- PRC (**P**recision-**R**ecall **C**urve) és una corba que representa la precisió (Precision) en funció de l'exhaustivitat (Recall). La precisió P i l'exhaustivitat R es defineixen com:
  - $P = \dfrac{VP}{VP + FP}$, on VP és el nombre de positius predits correctament i FP el nombre de falsos positius.
  - $R = \dfrac{VP}{VP + FN}$, on VP és el nombre de positius predits correctament i FN el nombre de falsos negatius.

La taula de confusió 4 pot ajudar a clarificar aquests conceptes.

Predit / Real	Positiu	Negatiu
Positiu	VP	FN
Negatiu	FP	VN

Taula 4: Taula de confusió

Fem servir els termes positiu i negatiu per designar la pertinença o no a una classe arbitrària.

4 Treballs relacionats amb la predicció de la mortalitat

En aquest apartat exposarem altres treballs relacionats que intenten realitzar tasques de predicció de mortalitat i LOS des de diferents punts de vista aplicant models predictius d'aprenentatge automàtic. Principalment, tractarem els treballs *Multitask learning and benchmarking with clinical time series data* de Harutyunyan et al. [9] , que implementa RNN amb GRUs per realitzar tasques de predicció de la mortalitat i; *Medical transformer for multimodal survival prediction in intensive care: integration of imaging and non-imaging data* de Khader, F. et al. [10], que fa servir transformers.

4.1 Breu exlicació del *benchmarking* de Harutyunyan

L'article *Multitask learning and benchmarking with clinical time series data* de Harutyunyan et al. [9] ha esdevingut un punt de referència per realitzar anàlisi comparatiu de models de predicció amb dades clíniques. En l'article, utilitzen la base de dades MIMIC-III i una arquitectura de model basada en LSTM per realitzar quatre tasques de predicció:

- Predicció de la mortalitat hospitalària a partir de les primeres 48 hores d'una estada a la UCI. Aquesta tasca està plantejada com a una tasca de classificació binària. La mètrica d'avaluació principal és AUC-ROC.
- Predicció de la descompensació. És a dir mirar de predir si la salut d'un pacient empitjorarà notablement durant les pròximes 24 hores d'estada a la UCI. La definició que fan aquí de la tasca és predir la mortalitat en les següents 24 hores per cada hora d'estada a la UCI. La mètrica d'avaluació principal és AUC-ROC.
- Predicció de l'estada a la UCI (LOS): predir el temps que manca a la UCI per cada hora d'estada. Es planteja aquesta tasca com a classifiació en 10 classes o intervals de temps (una per estades inferiors a un dia, 7 segemnts de llargada un dia per cada dia de la primera setmana, un segment per estades entre 1 i 2 setmanes i, un segments per estades majors a 2 setmanes). La mètrica principal d'avaluació és el coeficient kappa de Cohen.

- Classificació fenotípica. Determinar quina de les condicions de cures agudes són presents en una estada hospitalària d'un pacient. La tasca és una classificació múltietiqueta i la mètrica d'avaluació principal és la mitjana AUC-ROC.

4.2 Aprenentatge per transferència

Els models d'aprenentatge profund, com les RNN han demostrat àmpliament la seva eficàcia a l'hora d'efectuar diverses tasques de predicció amb dades clíniques. A banda d'altres problemes, aquests models són vulnerables al sobreajustament, sobretot quan les dades d'entrenament són escasses.

Al llarg d'aquesta secció explicarem la solució que proposen Gupta, P. et al. [17]: l'aprenentatge per transferència (*transfer learning*) aplicat a dades clíniques. D'acord amb els autors, l'aprenentatge per transferència és capaç de mitigar el fenòmen del sobreajustament al transferir el coneixement d'una xarxa ben entrenada en una tasca, a una altra xarxa enfocada a una tasca similar però amb un volum de dades d'entrenament força inferior.

Ha estat demostrat que xarxes pre-entrenades poden ser utilitzades a un ampli ventall de tasques de naturalesa similar[17]. Fins i tot, la transferència de pesos procedents de xarxes entrenades en tasques radicalment diferents pot arribar a ser millor que l'inicialització de pesos de forma aleatòria.

Gupta, P. et al. proposen dues aproximacions a l'aprenentatge per transferència: (i) en primer lloc l'extracció de característiques d'una xarxa pre-entrenada (en aquest cas, TimeNet [18]) i utilitzar-les per construïr models enfocats a una tasca concreta i (ii) l'inicialització d'una xarxa d'aprenentatge profund amb paràmetres d'una altra xarxa prèviament entrenada i ajustar-la a partir de dades etiquetades enfocant-la a la tasca objectiu. En aquesta secció, pretenem condensar de forma més o menys breu, els resultats publicats per aquests autors.

4.2.1 La xarxa neuronal pre-entrenada TimeNet

La xarxa neuronal TimeNet pretén ser una xarxa neuronal de propòsit general aplicada expecíficament a extreure característiques de sèries temporals [18]. Aquesta xarxa ha estat entrenada amb 18 conjunts de dades diferents però ha demostrat de ser de gran utilitat a l'hora de realitzar tasques de classificació amb 30 conjunts de dades aliens als fets servir durant la fase d'entrenament.

TimeNet conté 3 capes recurrents amb 60 unitats recurrents controlades (GRUs)

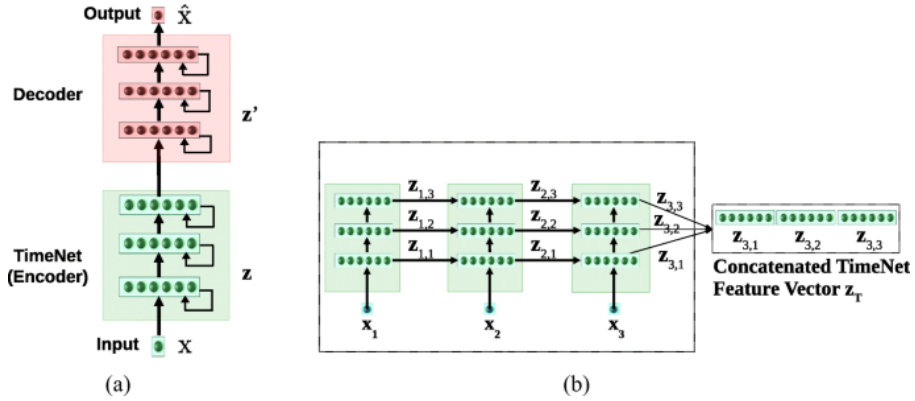


Figura 15: (a) Codificador-Descodificador TimeNet. (b) Extractor de vector de característiques de TimeNet. Imatges extretes de [17].

cadascuna (no detallarem el funcionament de les GRU, només cal saber que són similars a LSTM 3.4) i, per tant, genera vectors de característiques de mida 180. També fa ús d'un codificador f_E i un descodificador entrenats conjuntament a partir d'un sistema seq2seq. Tal i com es mostra a la figura 15, donat un conjunt de dades d'entrada, aquestes es codifiquen per generar el vector de característiques que farà servir el classificador. Aquesta codificació permet obtenir, donat un input de llargada variable $x_1, ..., x_\tau$, un vector de característiques de llargada fixa $z_T = f_E(x_1, ..., x_\tau; \mathbf{W}_E)$ a partir del conjunt de paràmetres $\mathbf{W}_E$ del codificador. Aquest vector de característiques pot ser usat després per entrenar un classificador més senzill. És degut a la gran utilitat que proporciona TimeNet que Gupta, P. et al. decideixen aplicar-la, enfocant-la a l'objectiu de la seva publicació.

4.2.2 Escenaris a l'hora d'aplicar l'aprenentatge per transferència

Considerem els conjunts de dades d'instàncies de sèries temporals D_S i D_T corresponents a uns conjunts origen S i objectiu T, amb $D_S = \{(x_S^i, y_S^i)\}_{i=1}^{N_S}$ on els $x_S^i := x^i = \{x_1^i, ..., x_\tau^i\}$, $x_t^i \in \mathbb{R}^n$ vectors n-dimensionals, són les sèries temporals, $y_S^i := y^i = \{y_1^i, ..., y_K^i\} \in \{0,1\}^K$ les etiquetes corresponents a la sèrie temporal i K és el nombre de tasques binàries de classificació. De forma anàloga, $D_T = \{(x_T^i, y_T^i)\}_{i=1}^{N_T}$, tal que $N_T << N_S$ i $y_T^i \in \{0,1\}$ de tal manera que la tasca objectiu és una tasca de classificació binària.

Direm que D_S i D_T són del mateix *domini* si els n paràmetres de D_S i D_T són els mateixos. També direm que la *tasca* de D_S i D_T és la mateixa si el nombre de classes objectiu a y_S i y_T és el mateix i semànticament són equivalents.

A partir d'aquí, Gupta, P. et al. plantegen dos escenaris:

1. *Adaptació del domini:* D_S conté sèries temporals de múltiples dominis provinents d'arxius públics i D_t conté sèries temporals de dades clíniques. L'objectiu és entrenar una RNN de forma no supervisada amb el conjunt D_S i adaptar el model pre-entrenat a la tasca objectiu, fent servir el conjunt D_T i aplicant entrenament supervisat.

2. *Adaptació de la tasca:* Tant D_S com D_T contenen sèries temporals de dades clíniques tals que els paràmtres de x_S i x_T corresponen als n mateixos camps clínics (e.g. ritme cardíac, nivells d'oxigen, etc.) però, y_S correspon a diverses tasques i y_T correspon a una tasca diferent però que manté algun tipus de relació amb les altres. L'objectiu en aquest cas és pre-entrenar un model RNN amb el conjunt D_S via entrenament no supervisat en les diverses tasques de y_S i, adaptar-lo a la tasca objectiu fent servir D_T via entrenament supervisat.

4.2.3 Adaptació del domini

Per dur a terme la tècnica d'adaptació del domini fent servir un extractor de característiques de sèries temporals pre-existent, TimeNet en aquest cas, tindrem en compte dues consideracions clau: (i) ampliar TimeNet per realitzar tasques de classificació de sèries temporals mèdiques multivariants i (ii) adaptar les característiques de TimeNet enfocades a tasques específiques del món mèdic. TimeNet pot ser adaptat a aquestes tasques afegint un classificador lineal tradicional abans de l'extracció de característiques, com es mostra a la Figura 16.

Considerem el conjunt $D_T = \{(x_T^i, y_T^i)\}_{i=1}^{N_T}$ on x_T^i és multivariant i $y_T^i \in \{0, 1\}$ tal que la tasca objectiu és una tasca de classificació binària i N_T el nombre de sèries temporals. D'aquesta manera podem considerar una tasca de classificació multi-etiqueta com un conjunt de classificacions binàries. A continuació es detalla la forma de construir una tasca de classificació binària.

Per cada sèrie temporal multivariant $x = \{x_1, ..., x_\tau\}$ on $x_t \in \mathbb{R}^n$, considerem n sèries univariants $x_j = \{x_{j1}, ..., x_{j\tau}\}$, $j = 1, ..., n$. Fent servir el codificador f_E de TimeNet, per cada x_j, obtenim el vector $z_j = f_E(x_j; W_E) \in \mathbb{R}^c$ on $c = 180$ i així obtenir el vector de característiques $z = [z_1, ..., z_n] \in \mathbb{R}^{n \times c}$.

Aquest vector z és usat després com a entrada al model de classificació. Com que la mida del vector és un nombre gran, amb l'intenció de quedar-nos amb les característiques més rellevants, l'etiqueta que li assignem és $\hat{y} = w \cdot z$, $w \in \mathbb{R}^m$. Els pesos w són obtinguts mitjançant la funció LASSO:

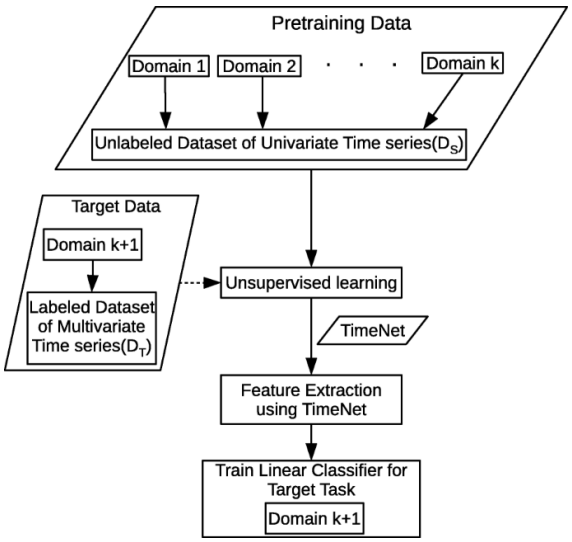


Figura 16: Escenari d'adaptació del domini. Imatge extreta de [17]

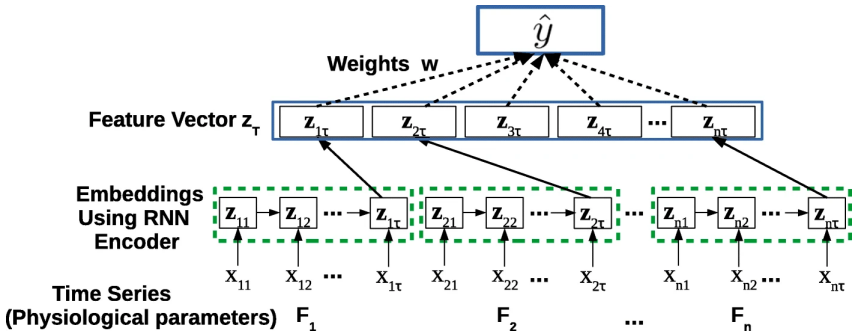


Figura 17: Extracció de caracerístiques amb TimeNet. Imatge extreta de [17]

$$\arg\min_{w} \frac{1}{N} \sum_{i=1}^{N} (y_i - w \cdot z_i)^2 + \alpha ||w||_1, \tag{4.1}$$

on $y_i \in \{0, 1\}$, α és una constant de control dispersió i $||w||_1 = \sum_{j=1}^{n} \sum_{k=1}^{c} |w_{jk}|$ és la norma L_1.

Per determinar la rellevància de cadascuna de les n característiques en cru que teníem inicialment, podem computar els *scores* de rellevància r_j, $j = 1, ..., n$ a partir dels pesos w:

$$r_j = \sum_{k=1}^{c} |w_{jk}|. \tag{4.2}$$

Podem normalitzar aquests *scores* via min-max per obtenir

$$r'_j = \frac{r_j - r_{min}}{r_{max} - r_{min}} \in [0, 1]. \tag{4.3}$$

Gupta et al. entrenen RNNs de la forma que acabem de descriure fent servir com a conjunt de dades MIMIC-III, amb $n = 76$ característiques en cru i mesures cada hora durant 48 hores ($\tau = 48$) plantejant les tasques de predicció de (i) presència de certs fenotips i (ii) mortalitat hospitalària i fent servir $\alpha = 0.0001$. Obtenen els resultats mostrats a les taules 5 i 6.

Predicció fenotípica	Micro AUROC	Macro AUROC
LR	0.799 (0.796, 0.803)	0.739 (0.734, 0.743)
LSTM	0.821 (0.818, 0.825)	0.770 (0.766, 0.775)
TimeNet-48	0.812 (0.808, 0.815)	0.761 (0.757, 0.765)
TimeNet-All	0.813 (0.810, 0.817)	0.764 (0.759, 0.768)
TimeNet-48-Eps	0.820 (0.817, 0.824)	0.772 (0.768, 0.777)
TimeNet-All-Eps	0.822 (0.819, 0.825)	0.775 (0.771,0.779)

Taula 5: Resultats d'entrenament de l'escenari d'adaptiació del domini per predicció fenotípica [17].

Predicció de mortalitat	AUROC	AUPRC
LR	0.848 (0.828, 0.868)	0.474 (0.419, 0.529)
LSTM	0.855 (0.835, 0.873)	0.485 (0.431, 0.537)
TimeNet-48	0.852 (0.831, 0.872)	0.519 (0.467, 0.571)

Taula 6: Resultats d'entrenament de l'escenari d'adaptiació del domini per predicció de mortalitat [17].

4.2.4 Adaptació de la tasca

A diferència de l'escernari d'adaptació del domini, l'escenari d'adaptació de la tasca consisteix en transferir l'aprenentatge adquirit a través d'una sèries de tasques a una tasca objectiu relacionada amb aquestes. Durant aquesta secció explicarem l'entrenament d'una RNN a la que Gupta et al. anomenen HealthNet [17] com a classificador binari en múltiples tasques de forma simultània.

Considerem $D_S = \{(x_S^i, y_S^i)\}_{i=1}^{N_S}$ i $D_T = \{(x_T^i, y_T^i)\}_{i=1}^{N_T}$ conjunts de dades de mateix domini procedents d'una base de dades de registres clínics. x_S és una sèrie temporal multivariant i $y_S \in \{0,1\}^K$ amb K el nombre de tasques de classificació binàries, $N_T << N_S$ i $y_T \in \{0,1\}$ una tasca de classificació binària. Llavors, el procediment proposat per Gupta és el d'incialment entrenar HealthNet en les N tasques de D_S i considerar les aproximacions següents per adaptar el model a la tasca objectiu de D_T:

1. Ajustar els paràmetres de HealthNet d'alguna forma adient per tal de que sigui entrenada correctament en la tasca objectiu.

2. Entrenar un classificador de regressió logística més senzill enfocat a la tasca objectiu i les característiques de HealthNet.

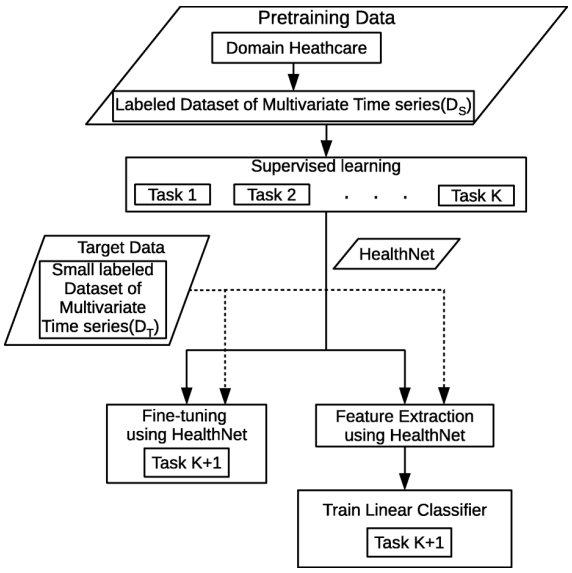


Figura 18: Escenari d'adaptació de la tasca. Imatge extreta de [17]

Per a obtenir HealthNet entrenen una RNN multicapa amb L capes recurrents que contenen unitats de recurrència per mapejar x_S a y_S com es mostra a la figura 19.

Considerem ara, $z_{t,l} \in \mathbb{R}^h$ la sortida de la capa oculta l-èssima a temps t, $z_t = [z_{t,1}, ..., z_{t,L}] \in \mathbb{R}^m$ l'estat ocult a temps t i h el nombre d'unitats de recurrència a cada capa, $m = h \times L$. Llavors, els paràmetres de la RNN són obtinguts via descens estocàstic del gradient minimitzant la funció de pèrdua d'entropia creuada $\mathcal{L}$:

$$z_\tau = f_E(x : W'_E), \hat{y} = \sigma(W_C z_{\tau,L} + b_C), \tag{4.4}$$

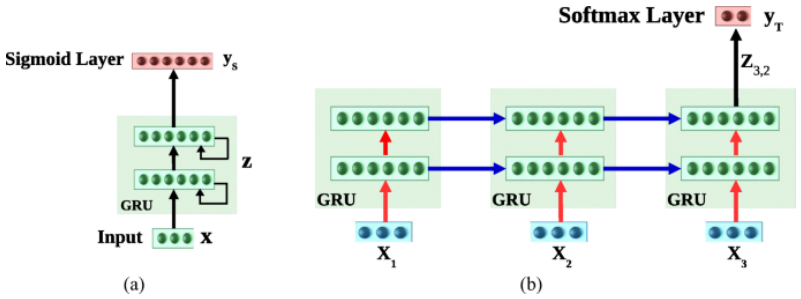


Figura 19: Entrenament de HealthNet. Imatge extreta de [17]

$$C(y_k, \hat{y}_k) = y_k \cdot log(\hat{y}_k) + (1 - y_k) \cdot log((1 - \hat{y}_k)), \tag{4.5}$$

$$\mathcal{L} = -\frac{1}{N_S \times K} \sum_{i=1}^{N_S} \sum_{k=1}^{K} C(y_k, \hat{y}_k), \tag{4.6}$$

on $\sigma(x) = \frac{1}{(1+e^{-x})}$ és la funció d'activació sigmoide, $\hat{y}$ és un estimador per l'objectiu y, W'_E són paràmtres de les capes recurrents i W_C i b_C són paràmetres de la capa de classificació.

Fan servir els paràmteres de les capes recurrents de HealthNet (W'_E) i noves capes de classificació binària (W'_C i b'_C) per inicialitzar els paràmetres de la nova RNN enfocada a la tasca objectiu. Per obtenir probabilitats de les classes de la nova tasca de classificació fan $y^i = softmax(W'_C z'^i_{\tau,L} + b'_C)$, on $z'^i_{\tau,L}$ és la sortida de les unitats de recurrència a la última capa L a l'última unitat temporal τ.

Siguin W'_{EF} i W'_{ER} els pesos no recurrents i recurrents de les capes recurrents. Els paràmetres són entrenats alhora minimitzant la funció de pèrdua d'entropia creuada amb regularitzador via descents estocàstic del gradient. Considerem dues tècniques de regularització donades per models d'ajustament de pèrdua $\mathcal{L}_1$ i $\mathcal{L}_2$:

$$\mathcal{L}_1 = -\frac{1}{N_T} \sum_{i=1}^{N_T} C(y, \hat{y}) + \lambda ||W'_{EF}||_1, \tag{4.7}$$

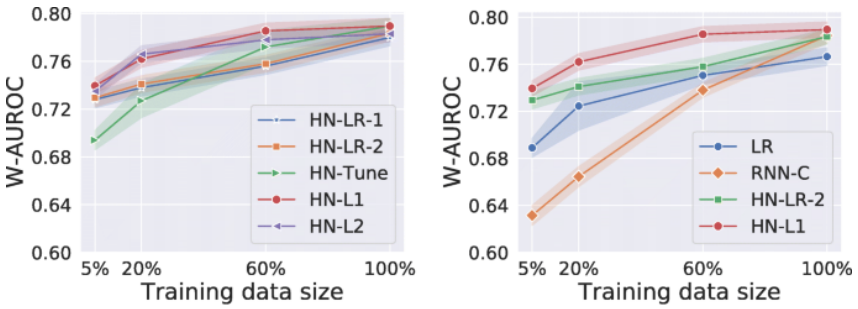
$$\mathcal{L}_2 = -\frac{1}{N_T} \sum_{i=1}^{N_T} C(y, \hat{y}) + \lambda ||W'_{EF}||_2, \tag{4.8}$$

on $\hat{y}$ és la probabilitat de la classe positiva, $||W'_{EF}|| = \sum_{j=1}^{m} |W_j|$ és la norma L_1 i $||W'_{EF}||_2 = \sum_{j=1}^{m} {W_j}^2$ és la norma L_2.

Finalment, donada una entrada $x \in D_T$, l'estat z_τ és el que Gupta et al. acaben fent servir com a entrada per l'entrenament del model de regressió logística. La probabilitat de la classe positiva l'obtenim com $\hat{y} = \sigma(w'_C z_\tau + b'_C)$, on w'_C i b'_C són paràmetres obtinguts minimitzant la versemblança de pèrdua $\mathcal{L}'$:

$$\mathcal{L}' = -\frac{1}{N_T} \sum_{i=1}^{N_T} C(y, \hat{y}) + \lambda ||w'_C||_1. \tag{4.9}$$

Tal i com fan amb l'escenari d'adaptació al domini, en aquest escenari també fan servir la base de dades MIMIC-III, amb $n = 76$ i $\tau = 48$. Considerem entrenar HealthNet enfocada a la mateixa tasca de detecció fenotípica que adaptació del domini però amb 20 dels 25 fenotíps. Els altres 5 els faran servir per avaluar la capacitat de transferència de HealthNet a una nova tasca. A la figura 20, en podem veure els resultats.



(a) Transfer learning approaches (b) Comparison with baseline

Figura 20: Resultats de la classficació aplicant adaptació de la tasca [17]

4.2.5 Comparació entre l'adaptació de la tasca i l'adaptació del domini

Per poder comparar els dos escenaris és necessari aplicar-los a una situació en que tots dos es puguin aplicar. Per exemple, en un escenari en que no es tenen dades de tipus mèdic el millor serà fer servir adaptació del domini amb TimeNet[17]. Si en canvi es tenen un nombre petit de característiques de tipus mèdic pot ser millor considerar el fet d'entrenar una RNN enfocada específicament a l'ambit mèdic i aplicar adaptació de la tasca. Per comparar els dos escenaris, doncs, Gupta et añ. les tasques objectiu de detectar presència o absència de 5 fenotips i obtenim els resultats de la figura 21.

Quan comparem els dos escenaris, podem observar com, quan el nombre de tasques origen es redueix, els resultats de l'adaptació de la tasca són inferiors als d'adaptació del domini amb TimeNet. Quan hi ha prou dades originals, en canvi, el

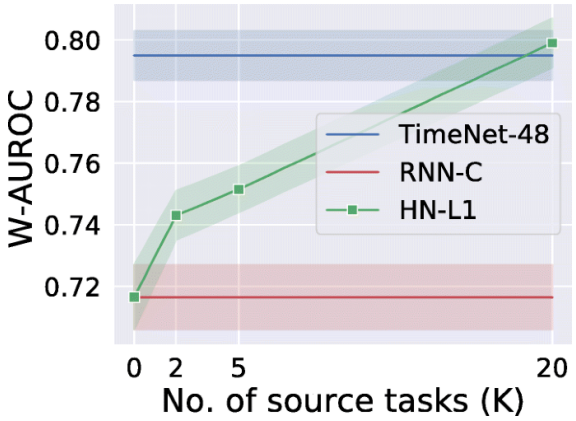


Figura 21: Comparació d'escenaris d'aprenentatge per transferència

rendiment s'equipara. Gupta, P et al. també remarquen que el cost computacional d'adaptar TimeNet a dades mèdiques és força més barat i ràpid que el d'entrenar un model com HealthNet de zero i després ajustar-lo a la nova tasca. Els temps amb les màquines fetes servir han estat de 10 minuts i 3 hores respectivament.

Després de comparar els diferents mètodes, queda palès que l'aprenentatge per transferència és capaç de proporcionar resultats molt positius i amb costos d'entrenament força més baixos que els d'entrenar una RNN tradicional.

5 Transformers mèdics per la tasca de predicció de la mortalitat

L'ús de l'arquitectura de xarxes neuronals de Transformers, si bé altament popular en models de processament de llenguatge natural (LNP), presenta una bibliografia no massa abundant pel que fa a tasques de predicció amb dades clíniques i, específicament fent ús de la base de dades MIMIC-IV. Exemples d'ús d'aquesta arquitectura per tasques de predicció clínica són:

- *Mortality prediction using medical time series on TBI patients* de Fonseca, J. et al. [11], on s'explora la possibilitat d'utilitzar MIMIC-III per determinar la mortalitat de pacients amb TCE (traumatisme cranioencefàlic) comparant una arquitectura basada en transformer i en LSTM.
- *Medical transformer for multimodal survival prediction in intensive care: integration of imaging and non-imaging data* de Khader, F. et al. [10], on es fa ús de models basats en transformers per predir la mortalitat combinant dades de registres mèdics de MIMIC-IV amb radiografies de pit de MIMIC-CXR.

És aquest últim article que detallarem a continuació, sobretot, per aportar una prespectiva aplicant a pacients amb un diagnòstic més ampli i amb un conjunt de dades més actualitzat. A l'article proposen tres experiments aportant diferents tipus de dades per realitzar les tasques de predicció: (i) ús únicament de registres mèdics, (ii) ús únicament d'imatges i (iii) ús combinat de registres mèdics i imatges. Aquest primer experiment, a banda, el replicarem amb els nostres propis recursos i estarem en disposició de comparar els resultats amb d'altres models predictius de l'*state of the art* de la matèria així com amb el benchmarking de Harutyunyan, H. et al. [9].

Per processar les dades, de forma que es puguin inocorporar origens de dades multimodals, segueixen el procediment marcat per Hayat, N.[14], incloent les 17 medicions clíniques de la taula 7. Pel que fa al processament d'imatges, es normalitzen les radioagrafies per seguir els estàndards marcats per ImageNet[15] i es reajusten a una mida de 384×384, que permetrà que siguin usades en models preentrenats. Finalment, la divisió de les dades consisteix en: conjunt d'entrenament, 72%; conjunt de validació, 8% i; conjunt d'avaluació, 20%.

Khader et al. proposen treballar sobre l'arquitectura transformer de Vaswani et al. [8], dissenyant-ne un de nou al que anomenaran MeTra (Medical Transformer) i,

Variable	Tipus
Taxa de reompliment capil·lar	Categòrica
Pressió arterial diastòlica	Contínua
Fracció d'oxigen inspirat	Contínua
Escala de coma de Glasgow—obertura ocular	Categòrica
Escala de coma de Glasgow—resposta motora	Categòrica
Escala de coma de Glasgow—resposta verbal	Categòrica
Escala de coma de Glasgow—total	Categòrica
Glucosa	Contínua
Freqüència cardíaca	Contínua
Alçada del cos	Contínua
Pressió arterial mitjana	Contínua
Saturació d'oxigen	Contínua
Freqüència respiratòria	Contínua
Pressió arterial sistòlica	Contínua
Temperatura	Contínua
Pes corporal	Contínua
pH	Contínua

Taula 7: Llistat de medicions clíniques escollides a [10] per realitzar els experiments de predicció de la mortalitat que incorporen registres clínics.

en particular, pels problemes en que s'incorpora problemes de visió artificial, sobre l'arquitectura de Dosovitsky et al. [13]. Com ja hem explicat a 3.5, els transformers treballen amb tokens, és per això que MeTra pren variables d'entrada d'imatges $x_{\text{CXR}} \in \mathbb{R}^{H \times W}$, on H i W són l'altura i l'amplada; de les que extreu, mitjançant un nucli de visió, les característiques $z_{\text{CXR}} \in \mathbb{R}^{N \times D}$, on N és el nombre de tokens i D, la dimensionalitat de la representació latent de cada token. De la mateixa manera, els vectors de paràmetres clínics $x_{\text{CP}} \in \mathbb{R}^{K \times T}$, on K és el nombre de paràmetres i T el nombre d'unitats de temps de la sèrie temporal, són projectats linealment a $x_{\text{CP}} \in \mathbb{R}^{M \times D}$ per tenir la mateixa dimensionalitat que els tokens d'imatges. Les dues representascions, llavors, són concatenades per obtenir la representació latent $z_{\text{MULTI}} \in \mathbb{R}^{(N+M) \times D}$. Els mecansimse de *self-attention* que processa les variables d'entrada, no té en compte els elements de la seqüència. Per resoldre-ho, es defi-

neixen $N + M$ tokens de mida D que A continuació també s'afegeix un token de classe CLS prefixat a x_{MULTI} i la representació multimodal resultant és processa mitjançant un codificador transformer on les capes de de *self-attention* permeten les transferències intermodals. Finalment, un perceptró multicapa amb funció d'activació sigmoide genera la sortida p_{SURVIVAL}, que indica la versemblança de que aquell pacient, durant aquella esta a la UCI, sobrevisqui o no. A la figura 22 podem veure un esquema de l'arquitectura.

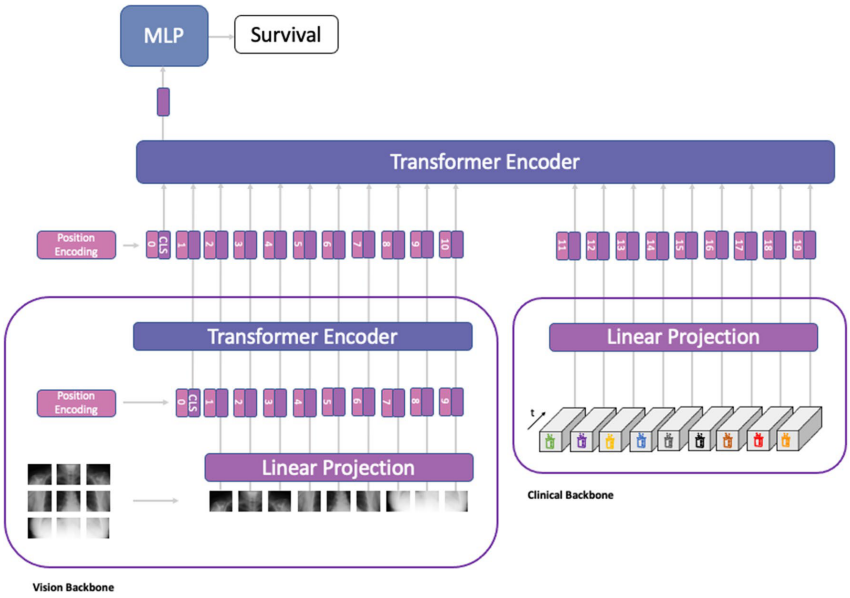


Figura 22: Arquitectura de MeTra. Imatge extreta de [10]

Per la versió amb l'experiment que treballa únicament amb registres clínics, MeTra(CP) s'inicialitzen a 0 els valors corresponents a x_{CXR} i, de la mateixa manera, per la versió que treballa únicament amb imatges, MeTra(CXR), s'inicialitzen a 0 els valors corresponents a x_{CP}. Com a onjectiu d'entrenament es minimitza la funció de pèrdua d'entropia creuada de la forma

$$\mathcal{L} = y \log(p_{\text{SURVIVAL}} + (1 - y) \log(1 - p_{\text{SURVIVAL}}), \tag{5.1}$$

on $y \in \{0, 1\}$ és l'etiqueta real que indica la supervivència, essent 1 que el pacient mor durant l'estada i 0 que el pacient sobreviu.

La mida del conjunt de dades d'entrenament que agafen per realitzar l'experiment és de 6.125 pacients de la versió 1.0 de MIMIC-IV (4.396 pacients d'entrenament; 472, de validació i; 1257, de testeig), amb un percentatge de morta-

litat del 16%. Les sèries temporals corresponents a les medicions durant l'estada hospitalària, s'agafen totes durant les primeres 48 hores en intervals d'una hora. És a dir, tindrem els vectors d'entrada $x_{\text{CP}} \in \mathbb{R}^{17\times 48}$. Al repositori de github https://github.com/FirasGit/MeTra hi ha el codi per reproduir els experiments.

A la taula 8 resumim els resultats obtinguts per Khader et al., en les diferents versions de l'experiment, comparades amb altres models que realitzen tasques similars, en termes d'AUCROC i AUCPRC. En algun d'aquests casos però, la comparació s'ha de prendre amb pinces ja que no sempre s'estan utilitzant volums de dades ni tipus de dades d'ordre similar, si més no, permet fer-nos una idea del potencial que té l'arquitectura transformers per aquest tipus de tasques predictives.

Mètode	**AUCROC**	**AUCPRC**
MeTra(CP + CXR)	0.863 [0.835, 0.889]	0.594 [0.526, 0.662]
MeTra(CP)	0.785 [0.751, 0.819]	-
MeTra(CXR)	0.811 [0.779, 0.841]	-
Early	0.827 [0.801, 0.854]	0.485 [0.417, 0.555]
Joint	0.825 [0.798, 0.853]	0.506 [0.436, 0.574]
MedFuse (PT)	0.841 [0.813, 0.868]	0.544 [0.477, 0.609]
MedFuse (OPTIMAL)	0.865 [0.837, 0.889]	0.594 [0.526, 0.655]
MedFuse (RI)	0.817 [0.785, 0.846]	0.471 [0.404, 0.545]

Taula 8: Comparació de resultats obtinguts en classifiació de la mortailitat hospitalària. Aquests models es fan servir de comparativa per què també combinen dades multimodals. Entre [,] hi ha indicat l'interval de confiança del 95%.Els valors de les mètriques han estat extrets de [14][10].

6 Experimentació

Inicialment, en aquest punt del projecte, havíem plantejat tres experiments per determinar la mortalitat hospitalària fent servir tècniques predictives que apliquin models de classifiació d'aprenentatge automàtic diferents:

- Replicar el primer experiment de Gupta et al.[17] (secció 4.2.3) utilitzant el conjunt de dades actualitzat de la versió 2.2 de MIMIC-IV.
- Realitzar l'experiment de Khader et. al [10] (secció 3.5) aplicant l'arquitectura transformer únicament a registres de medicions clíniques i ampliant la mida del conjunt de dades
- Seleccionar un subconjunt de pacients que presentin una característica fenotípica concreta per posar-nos en l'escenari d'adaptació a la tasca (secció 4.2.4) utilitzant l'arquitectura transformer de Khader en lloc de RNN com fa Gupta.

El primer experiment era relativament simple conceptualment: agafar una RNN entrenada en sèries temporals de tot tipus i plantejar l'escenari d'adaptació al domini (4.2.3) de la mateixa manera que Gupta et al. en la tasca de predicció de mortalitat, però utilitzant MIMIC-IV, a diferència d'ells que utilitzaven MIMIC-III i veure si, aquesta actualització en el conjunt de dades millorava els resultats obtinguts. Finalment, però, no ha pogut ser realitzat. Per fer-ho necessitavem una implementació d'alguna RNN entrenada en sèries temporals com la TimeNet[18] de Malhotra et al. tal i com feien Gupta, P. et al. No disposàvem de prou capacitat computacional per entrenar-ne una de zero i vam recórrer a buscar una RNN *off-the-shelf*. Fins on tenim constància, els autors tampoc n'han proporcionat cap que es trobi en disponibilitat per ser usada de forma pública i tampoc n'hem trobat cap que s'hi assembli en la forma i el propòsit. Si bé hem estat treballant amb dues implementacions fetes per part dels usuaris de GitHub: paudan[20] i kirarenctaon[21], després de múltiples esforços fallits no hem estat capaços d'adaptar-les a les nostres necessitats.

Hem realitzat el segon experiment de Khader et al. (secció 3.5). Hem utilitzat el codi que publiquen a la seva pàgina de github [16] i hem realitzat el processament de les dades tal com marquen. Nosaltres però, per limitacions en termes d'espai de capacitat computaciona i espai d'emmagatzematge, hem decidit utilitzar només els registres mèdics de MIMIC-IV en lloc de la combinació de registes i imatges. És

per això que, aquí sí, hem decidit no limitar-nos als 6.125 que tenen els dos tipus de dades i utilitzar 21.741 que sumen 25.083 estades a la UCI en total.

L'implementació [16] ha estat realitzada fent servir els recursos de la llibreria *Pytorch Lightning* de *Python*, un *framework* d'aprenentatge profund per investigadors per ser utilitzat de forma senzilla [22]. La configuració del model és exactament la mateixa que la de Khader 22. La divisió entre conjunt d'entrenament, de validació i d'avaluació també ha estat la mateixa: 72%, 8% i 20% respectivament. Hem entrenat el model en 200 epochs i fent servir 30 fils d'execució a un Lenovo Thinkpad amb un IntelCore i5.

Tampoc hem pogut realitzar l'últim experiment en que plantejavem entrenar un model amb l'arquitectura transformer de Khader fent servir un subconjunt de dades que presentin un tret fenotípic concret fer *fine-tuning* per adaptar el model a un conjunt de dades més ampli i actualitzat com és MIMIC-IV. L'objectiu era veure com es podia aplicar l'escenari d'adaptació a la tasca [17] (secció 4.2.4) a una xarxa amb transformers en lloc de RNN. Finalment, però, degut a limitacions de capacitat computacional i temps no l'hem pogut a terme. El temps per realitzar l'entrenament era limitat i el segon experiment que hem realitzat havia suposat un ús intensiu dels recursos (figures 23,24 i 25) del Lenovo Thinkpad que era necessari per a la realització d'altres tasques.

7 Resultats

Com acabem de descriure a l'apartat anterior, hem entrenat el model amb el disseny de Khader, amb 200 epochs amb una mitjana de 30 fils d'execució diferents. El temps total d'entrenament ha estat de 71 hores, 19 minuts i 40 segons repartits en quatre etapes diferents en un processador 11th Gen Intel(R) Core(TM) i5-1135G7 de 2.42 GHz. Els resultats que hem obtingut en termes de precisió en la classificació, sobre el conjunt d'avaluació, fent servir les mètriques AUCROC i AUCPRC el podem veure a la taula 9.

AUCROC	AUCPRC
0.808 [0.79, 0.826]	0.404 [0.394, 0.423]

Taula 9: Resultats obtinguts sobre el conjunt d'avaulació pel transformer mèdic

Khader et al. repeteixen l'experiment 100 vegades, nosaltres únicament l'hem realitzat un cop. Pel que fa a classificació, tot i les poques dades de les que disposem, el resultat és positiu. El model entrenat obté resultats en termes de precisió en la classificació, fent servir les mètriques d'avaluació AUCROC i AUCPRC, que estan al nivell dels de Khader [10] i Hayat [14], els resultats dels quals es mostren a la taula 8, essent entrenat únicament amb registres clínics. Pel que fa a eficiència de l'entrenament, no tenim referències amb que comparar-ho (Khader et al. no proporcionen dades sobre els temps d'exeució), però entenem que les 71 hores que ens ha trigat és un temps massa elevat. Hipotetizem que, tot i el temps que ha trigat l'entrenament, degut a la naturalesa paral·lelitzable de l'arquitectura transformer, aquest ha estat més curt del que hauria estat l'entrenament, en les mateixes condicions de hardware i mida de les dades, d'un model amb una arquitectura RNN.

De totes formes, suposem que aquest temps s'explica en gran mesura a les limitacions de maquinària de la que hem disposat. Un altre factor que pot haver allargat el temps d'entrenament és la grandària del conjunt de dades (unes 3.5 vegades més gran que el de Khader), però altra vegada, no tenim amb què comparar-ho.

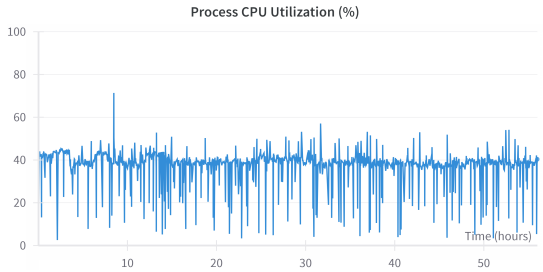


Figura 23: Percentatge de CPU utilitzat en la quarta etapa d'entrenament del model. Obtingut fent servir els recursos de `https://wandb.ai/`.

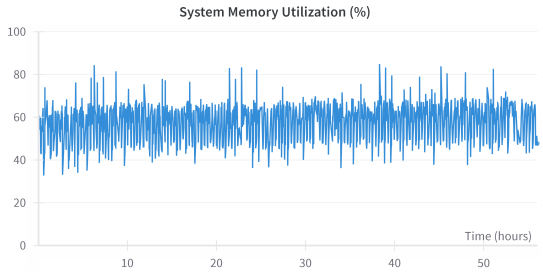


Figura 24: Percentatge de memòria del sistema utilitzada en la quarta etapa d'entrenament del model. Obtingut fent servir els recursos de `https://wandb.ai/`.

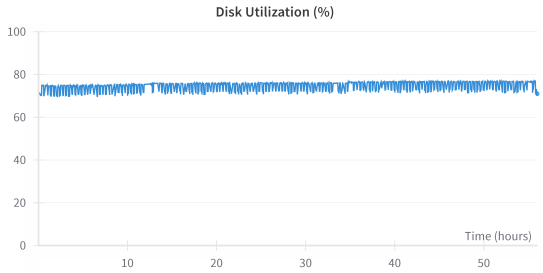


Figura 25: Percentatge de disc utilitzat en la quarta etapa d'entrenament del model. Obtingut fent servir els recursos de `https://wandb.ai/`.

8 Conclusions

En aquest treball, hem examinat l'ús de models predictius d'aprenentatge automàtic en el camp de la medicina, específicament en l'anàlisi de registres clínics de pacients a la unitat de cures intensives (UCI). Hem pogut repassar la literatura que forma part de l'estat de l'art i comprendre els fonaments teòrics que hi ha al darrere.

Tot i que no hem pogut realitzar tots els experiments que havíem plantejat inicialment, hem vist com l'ús de transformers presenta bons resultats de classificació. En el nostre cas hem obtingut valors AUCROC de 0.808 i AUCPRC de 0.404. No hem obtingut tant bons resultats pel que fa a temps d'execució, un dels potencials que té l'arquitectura transformer. Però com ja hipotetitzat, aquest resultat segurament és causa del hardware utilitzat.

En definitiva, aquest treball ens ha permès fer un pas endavant en la comprensió i aplicació dels models predictius d'aprenentatge automàtic en la predicció de la mortalitat hospitalària, mostrant especialment el potencial de les arquitectures transformers en l'anàlisi de dades clíniques.

Referències

[1] Johnson, A., et al. (2022). *MIMIC-IV, a freely accessible electronic health record dataset*. Scientific Data. Obtingut de https://www.nature.com/articles/s41597-022-01899-x

[2] Saeed, M. (n.d.). *An Introduction to Recurrent Neural Networks and the Math That Powers Them*. Machine Learning Mastery. Obtingut de https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/

[3] Health Insurance Portability and Accountability Act of 1996 (HIPAA).

[4] colah. (2015). *Understanding LSTM Networks*. colah's blog. Obtingut de https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] MIMIC-IV Documentation. (n.d.). Obtingut de https://mimic.mit.edu/docs/iv/

[6] ICD9 CODES. (n.d.). Obtingut de https://icd.codes/icd9cm

[7] Arbel, N. (n.d.). *How LSTM networks solve the problem of vanishing gradients*. DataDrivenInvestor. Obtingut de https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

[8] Vaswani, A., et al. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems, 30. arXiv:1706.03762.

[9] Harutyunyan, H., et al. (2019). *Multitask learning and benchmarking with clinical time series data*. Scientific Data. Obtingut de https://www.nature.com/articles/s41597-019-0103-9

[10] Khader, F., et al. (2023). *Medical transformer for multimodal survival prediction in intensive care: integration of imaging and non-imaging data*. arXiv:2107.11381.

[11] Fonseca, J., et al. (2023). *Mortality prediction using medical time series on TBI patients*.

[12] Wang, L. (2021). *Ciencia de datos para la predicción de mortalidad hospitalaria y duración de la estancia en la UCI con MIMIC-III*.

[13] Dosovitskiy, A., et al. (2021). *An image is worth 16x16 words: transformers for image recognition at scale*. International Conference on Learning Representations. arXiv:2010.11929.

[14] Hayat, N., et al. (n.d.). *MedFuse: Multi-modal fusion with clinical time-series data and chest X-ray images*. arXiv:2104.06525.

[15] Deng, J., et al. (2009). *ImageNet: A large-scale hierarchical image database*. IEEE Conference on Computer Vision and Pattern Recognition, 248–255. arXiv:0908.3387.

[16] Khader, F. (2023). *MeTra*. Obtingut de `https://github.com/FirasGit/MeTra`

[17] Gupta, P., et al. (2017). *Transfer Learning for Clinical Time Series Analysis Using Deep Neural Networks*. arXiv:1801.01544.

[18] Malhotra, P., et al. (2017). *TimeNet: Pre-trained deep recurrent neural network for time series classification*. arXiv:1706.08838.

[19] Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-term Memory*. Neural Computation, 9(8), 1735–1780.

[20] Danenas, P. *TimeNet*. GitHub. Obtingut de `https://github.com/paudan/TimeNet`

[21] Cho, S. *TimeNet*. GitHub. Obtingut de `https://github.com/kirarenctaon/timenet`

[22] Falcon, W.A. *PyTorch Lightning*. GitHub. Obtingut de `https://github.com/PyTorchLightning/pytorch-lightning`