Universidad de Concepción
Dirección de Postgrado
Facultad de Ingeniería - Ingeniería Civil Informática

PHASED LONG SHORT TERM MEMORY PARA CLASIFICACIÓN DE OBJETOS VARIABLES MUESTREADOS IRREGULARMENTE

Memoria para optar al grado de
INGENIERO CIVIL INFORMÁTICO

POR
Cristobal Donoso Oliva
CONCEPCIÓN, CHILE
Agosto, 2018

Profesor guía: Guillermo Cabrera Vives
Departamento de Ingeniería Informática y Ciencias de la Computación
Facultad de Ingeniería
Universidad de Concepción

A mis padres, por su sacrificio y confianza. A mi abuela, por sus almuerzos y compañía incondicional. A mi hermana, por ser mi ejemplo a seguir. A mi compañera de caídas y victorias; quien trasnochó conmigo en más de una oportunidad. A mis amigos, los cuales fueron mi cable a tierra en todo momento. A mis profesores del colegio, por enseñarme lo valioso que es el conocimiento y la crítica. A mis profesores, María Angélica Pinninghoff y Ricardo Contreras, quienes desde temprano me apoyaron e impulsaron en la investigación. Al profesor Andreas Polimerys, por sus críticas y reflexiones. A mi tutor, Guille Cabrera, por su apoyo, entrega y enseñanzas. A Luis Bello, por su altruismo en cada ayudantía extraoficial o logia de estudio. A mis compañeros de universidad, con quienes estudiamos y compartimos en tocatas y paseos. A mis mascotas quienes permanecen en lo más profundo de mis recuerdos. A todos mis cercanos, y aquellos que pude haber olvidado.

Resumen

En los últimos años, la cantidad de datos astronómicos capaces de ser almacenados ha crecido rápidamente, llevando a la necesidad de manejar grandes cantidades de datos obtenidos a partir de telescopios y del uso de técnicas analíticas tales como la espectrometría y la fotometría. Surge así el campo de la astroinformática, que involucra el empleo de distintas técnicas propias de las ciencias de la computación y su aplicación a problemas del ámbito astronómico. En la presente investigación, se utilizaron redes neuronales recurrentes para la clasificación de supernovas y otras estrellas variables, mediante el procesamiento de curvas de luz fotométricas muestreadas irregularmente. Con el propósito de mejorar el desempeño de la clasificación en términos de exactitud y puntaje f1, se utilizó un modelo con unidades Phased Long Short Term Memory (PLSTM). Posteriormente, se realizó un estudio comparativo entre este modelo y el desarrollado por Charnock y et al. (2017), en el que fueron empleadas dos capas ocultas de Long Short Term Memory (LSTM). Las pruebas se realizaron utilizando curvas de luz simuladas correspondientes a la fotometría de varios tipos de supernovas, y curvas reales correspondientes a la fotometría de otros tipos de estrellas. Para las curvas de luz correspondientes a supernovas, se obtuvo un 80 % de exactitud y 76 % de puntaje f1 para el modelo con PLSTM, mientras que para el modelo con LSTM se obtuvo un 87 % de exactitud y 85 % de puntaje f1. En tanto, para las curvas de luz reales se obtuvo un 60 % de exactitud para las PLSTM y un 56 % de exactitud para las LSTM. Los resultados obtenidos a través de la experimentación indican que el modelo con PLSTM no mejora el rendimiento obtenido por el modelo con LSTM para la clasificación de supernovas. Por otra parte, es posible concluir preliminarmente que el modelo con PLSTM obtiene un mejor desempeño al utilizar una mayor cantidad de unidades ocultas sobre el conjunto de curvas reales.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

INTRODUCCIÓN

El uso de grandes volúmenes de datos es hoy un paradigma ya establecido y aplicado en múltiples disciplinas. Los datos describen fenómenos del mundo observable; así, desde la abstracción numérica podemos encontrar patrones y/o estructuras adyacentes que posibilitan la realización de tareas de aprendizaje.

En el campo de la astronomıía, gran parte de los investigadores utilizan imágenes captadas por telescopios para estudiar el universo; sin embargo, las capacidades humanas resultan insuficientes al momento de analizar la enorme cantidad de datos generados. En este contexto, se estima que para el año 2021 el Large Synoptic Survey Telescope (LSST) entregará alrededor de 15 TB de imágenes en bruto por noche [7]. Así, emerge el nuevo campo de la astro-informática, que emplea técnicas provenientes desde las ciencias de la computación para el descubrimiento de nuevos conocimientos [4].

Uno de los grandes desafiíos en el estudio del universo se encuentra relacionado con la clasificación de objetos astronómicos; en particular, la expansión del universo o el Big Crunch, puede ser determinada a través de la medición del brillo o corrimientos al rojo sobre supernovas -explosiones estelares que generan un brillo notable en el espacio- lo que permite diferenciarlas fácilmente de otros objetos luminosos [19].

En astronomía, se estudia la composición química de objetos luminosos a través de la espectrometría, técnica analítica cuyo fundamento se encuentra en el hecho de que tanto elementos como compuestos químicos emiten una intensidad de luz determinada a distintas longitudes de onda. Tales intensidades son medidas y

registradas a través del uso de un instrumento llamado espectrofotómetro, y los datos así obtenidos se visualizan en un espectro cuyos ejes corresponden al flujo de fotones difractados a distintas longitudes de onda. A partir de estos espectros las supernovas pueden ser clasificadas en dos grandes grupos, en función de la presencia (tipo II) o ausencia (tipo I) de hidrógeno; una vez llevada a cabo esta clasificación, se continúan realizando sub-clasificaciones de acuerdo a los distintos componentes que aparecen en el espectro [10]. Sin embargo, la aplicación de la espectrometría resulta de alto costo en cuanto al uso de datos, debido a que mide y registra la intensidad de la luz en cada longitud de onda en el rango completo del espectro electromagnético. Así, una buena alternativa metodológica se encuentra en la fotometría, que consiste en la medición del flujo de luz de un objeto en un rango de la radiación electromagnética, y utilizar los datos obtenidos para clasificar distintas curvas de luz .

Ahora bien, una curva de luz está compuesta por magnitudes de brillo (en distintas longitudes de onda) asociadas a un tiempo específico en el que se realizó la observación. La imagen 1.1 muestra una curva de luz para una supernova tipo Ia. Una forma de obtener una curva de luz es a partir de la medición de la intensidad

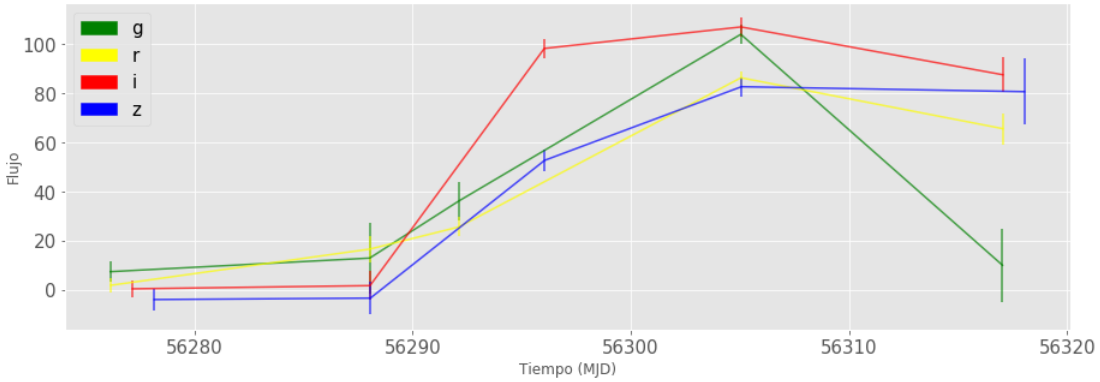


Figura 1.1: Supernova representada por cuatro curvas de luz en distintas bandas (griz). Cada banda comprende un rango de longitud de onda distinto

de cada píxel en una imagen; diremos entonces, que una curva de luz contiene

magnitudes de brillo que representan el flujo de fotones para distintas imágenes en el tiempo. Se ha demostrado que las supernovas pueden ser clasificadas en función de sus curvas de luz, que corresponden a series de tiempo [20]

Las series de tiempo son muy importantes para propósitos de clasificación. A través de la correlación entre magnitudes, es posible detectar estructuras o dinámicas que subyacen implícitamente a través del tiempo [5].

Se han propuesto diversos métodos estadísticos para trabajar con series de tiempo, los que pueden ser paramétricos o no paramétricos [21]. El uso de estadísticos en un modelo paramétrico puede significar un gran costo asociado. Dado un conjunto finito de parámetros ϕ, futuras predicciones , x, son independientes de los datos observados

$$P(x|\phi, D) = p(x|\phi) \tag{1.1}$$

donde D corresponde a los datos. Luego, ϕ capturará todo lo que se conoce de los datos. La complejidad del modelo estará limitada aun cuando la cantidad de datos no lo esté. Esto los hace poco escalables. Desde ese punto de vista, métodos no paramétricos, como las redes neuronales, ajustan estocásticamente una función discriminativa asumiendo un espacio infinito de dimensiones para ϕ. La cantidad de información que capture la función puede crecer a medida que crece la cantidad de datos de datos D, convirtiéndolo en un método más flexible [22].

Las Redes Neuronales (en adelante NN, del inglés Neural Network) son un modelo matemático bioinspirado en el proceso de aprendizaje humano. Requieren una gran cantidad de datos para ser entrenadas, de lo contrario se sobreajustan. En astronomía, la cantidad de datos manejada permite el uso de este tipo de modelos [23]. Por otro lado, las NN en su arquitectura más simple, no considera el tiempo. Las redes neuronales recurrentes (en adelante RNN, del inglés Recurrent Neural Network) añaden ciclos en la capa oculta donde cada ciclo representa un tiempo distinto. Así, las RNN se constituyen como una extensión de las NN, que utiliza y mantiene información temporal para entrenar la red. Sin embargo, existen

problemas cuando la entrada posee dependencias muy largas entre los tiempos [2]. Actualmente, el estado del arte de las RNN hace uso de unidades Long Short Term Memory (en adelante LSTM) cuya estructura permite mantener eficientemente una memoria a largo plazo. Se han propuesto nuevos modelos para aquellos casos particulares en los que las LSTM no rinden lo suficiente; por ejemplo, en el caso de muestreos irregulares. Para mejorar el rendimiento, en el año 2016, Daniel Neil propuso un nuevo modelo denominado Phased LSTM (en adelante PLSTM) [18] cuya arquitectura asume una periodicidad en el muestreo, añadiendo así, la capacidad de aprender en un rango de tiempo real en qué momento debería haber una observación, y de esta forma aumentar el índice de convergencia del modelo.

En la búsqueda de un buen modelo de clasificación, en el año 2010 se lanzó un desafío cuyo objetivo era la clasificación de supernovas utilizando simulaciones de curvas de luz en distintos filtros [15]. En el año 2017 y utilizando el mismo conjunto de datos, Charnock y colaboradores resolvieron el desafío de clasificación de supernovas utilizando una LSTM, alcanzando la mejor exactitud conocida a la fecha [6].

En la presente investigación se muestra el problema de clasificación de supernovas con el propósito de comparar el desempeño en términos de exactitud y puntaje f1 del modelo con PLSTM contra el modelo con LSTM. Adicionalmente, utilizamos objetos correspondientes a distintas estrellas variables extraídas desde el conjunto de datos MACHO [1] con el propósito de extender el análisis comparativo sobre un dominio real. A continuación, se muestra el desarrollo de la investigación: En el capitulo 2 se expone la hipótesis del proyecto, los objetivos -tanto específicos y como generales- y metodología. En el capitulo 3 se desarrollan los conceptos astronómicos más importantes como también los modelos utilizados; esto es desde, NN simple a LSTM y PLSTM. En el capitulo 4 presentamos una revisión de los datos y el preprocesamiento de estos en la sección 4.1. La sección 4.2 presenta las diferentes arquitecturas utilizadas. Finalmente, en la sección 5 se muestran los experimentos y resultados y en el capitulo 6 algunas conclusiones importantes.

Capítulo 2

HIPÓTESIS Y OBJETIVOS

2.1. Hipótesis

El uso de redes neuronales recurrentes con unidades PLSTM en el procesamiento de curvas de luz muestreadas irregularmente para la clasificación de objetos luminosos variables permite obtener un mejor desempeño en términos de exactitud y puntaje f1 en comparación con el empleo de redes neuronales recurrentes con unidades LSTM.

2.2. Objetivos

2.2.1. Objetivos Generales

Comparar empíricamente el desempeño de clasificación multiclase entre un modelo que utiliza unidades PLSTM y otro que utiliza unidades LSTM para el procesamiento de datos fotométricos simulados (supernovas) y reales (MACHO).

2.2.2. Objetivos Específicos

1. Extender el dominio de clasificación que emplean Charnock y colaboradores a 8 tipos de supernovas

2. Estandarizar los datos de curvas de luz correspondientes a supernovas y estrellas reales para su procesamiento en ambos modelos de redes neuronales recurrentes

3. Comparar el desempeño de clasificación, en términos de exactitud y puntaje f1, entre el modelo con unidades PLSTM y el modelo con unidades LSTM utilizando 8 tipos de supernovas

4. Comparar el desempeño de clasificación, en términos de exactitud y puntaje f1, entre el modelo con unidades PLSTM y el modelo con unidades LSTM sobre los subconjuntos de datos extraídos a partir de las curvas reales de MACHO.

Capítulo 3

MARCO TEÓRICO

3.1. Supernovas, Corrimiento al rojo y Técnicas de medición

Las supernovas son transientes cuya naturaleza consiste en la explosión de una estrella debido a una gran cantidad de energía concentrada en su núcleo. En astronomía, se utiliza el término transiente para hacer referencia a un objeto que es observable durante un periodo de tiempo limitado que va desde segundos hasta días; como es el caso de las supernovas. El fenómeno completo puede tardar millones de años, hasta que la estrella muera. Sin embargo, el núcleo colapsa en un cuarto de segundo y la onda de brillo puede iluminar algunos meses hasta que se desvanezca en años. Durante todo este tiempo, el observador realiza seguimientos intermitentes con el objetivo de capturar el pico de magnitud de luz. Luego, y con una periodicidad mayor, se captura la mengua de la curva. La imagen 3.1 muestra la diferencia entre puntos de luz, en función del tiempo, durante la explosión. Nótese que al principio de la curva, la diferencia entre tiempos de observación es menor que al final. Dado que el Universo se expande continuamente entonces la longitud de onda emitida por un objeto astronómico también incrementa. Al observar una supernova, esta se ve alterada por su corrimiento al rojo, dicho de otra manera incrementa proporcionalmente según la distancia percibida por el observador. En este trabajo consideramos cuatro intervalos estándar de longitud de onda según Sloan Digital Sky Survey (SDSS). La tabla 3.1 muestra las longitudes asociadas a cada filtro. Un filtro comprende un rango definido del espectro electromagnético absorbiendo fotones en un rango estándar.

En la introducción se hizo un pequeño esbozo sobre las características de una

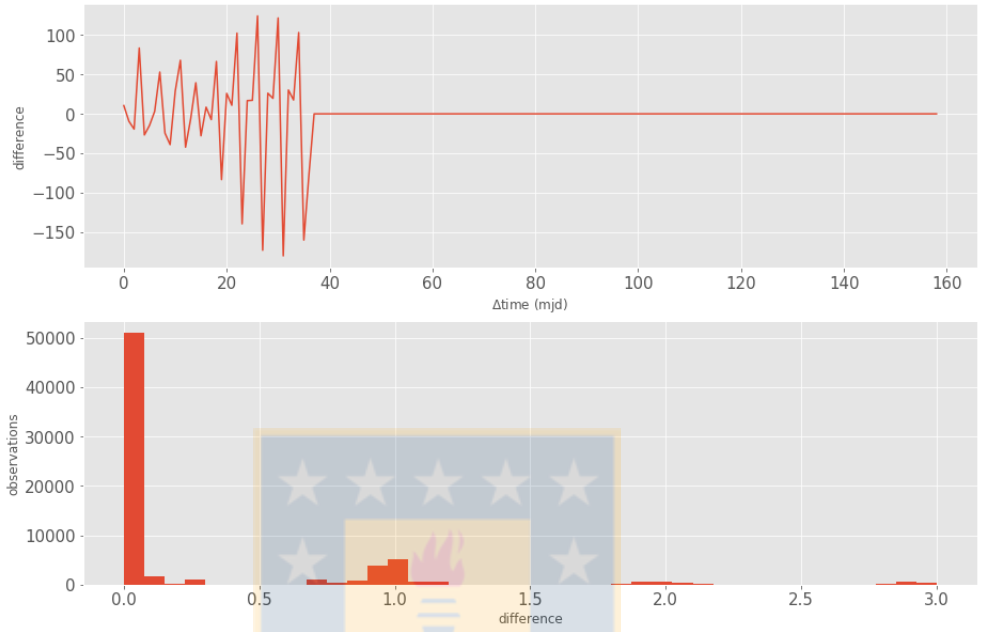


Figura 3.1: [Superior]Diferencia entre tiempos en función de la hora de observación. La diferencia crece en el mismo sentido del tiempo de observación. [Inferior] Cantidad de observaciones por diferencia de tiempo. La mayor cantidad de observaciones se concentran al comienzo del periodo de observación (donde la diferencia entre magnitudes es menor)

Banda	Longitud de onda (Angstroms)
Green (g)	4770
Red (r)	6231
Near Infrared (i)	7625
Infrared (z)	9134

Cuadro 3.1: Longitudes de onda asociados a las bandas griz

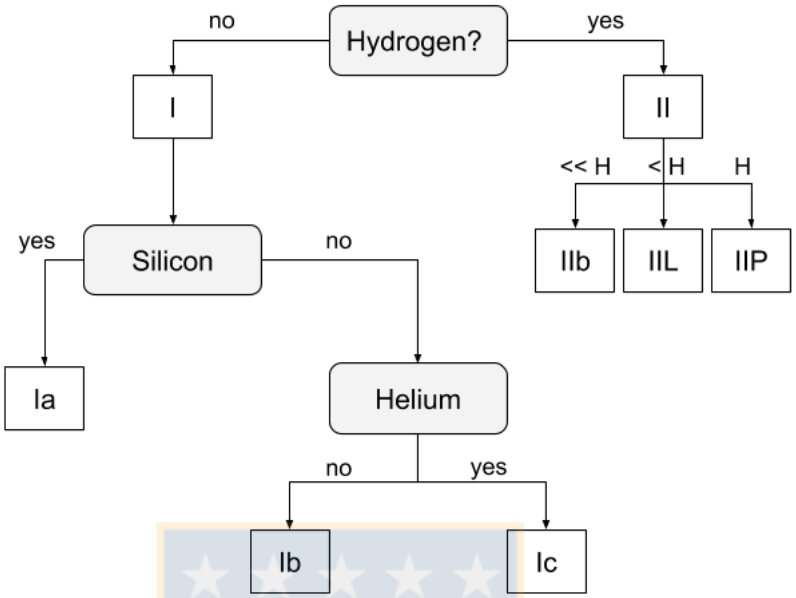


Figura 3.2: Árbol de clasificación jerárquico asociado a tipos de supernovas en función de su composición química. Desde la parte superior del árbol se ve la primera separación de tipos de supernovas según presencia o ausencia de Hidrógeno. La rama de la derecha (tipo II) genera subclases IIb, IIL IIP según la cantidad de Hidrógeno presente en el espectro. Por el otro lado, tenemos la supernova de tipo I cuya subdivisión está condicionada por la presencia de Silicio y Helio.

supernova. En específico, cada supernova pertenece a una clase -y esta clase puede ser o no una subclase de otra clase más general. La clase está asociada a la composición química de la supernova. El factor discriminante principal tiene que ver con la presencia de Hidrógeno (tipo I vs II). Una vez realizada la primera separación podemos clasificar en función del Silicio y Helio, así también como la cantidad de Hidrógeno que contiene la supernova. La figura 3.2 muestra el árbol de clasificación de las supernovas.

3.2. Redes Neuronales

En el año 1943, McCulloh y Pitts [17] presentaron el primer modelo matemático inspirado en el mecanismo biológico a través del cual los seres humanos desarrollan el aprendizaje. En particular, hablamos del reconocimiento de patrones mediante la observación de características de un fenómeno u objeto y las

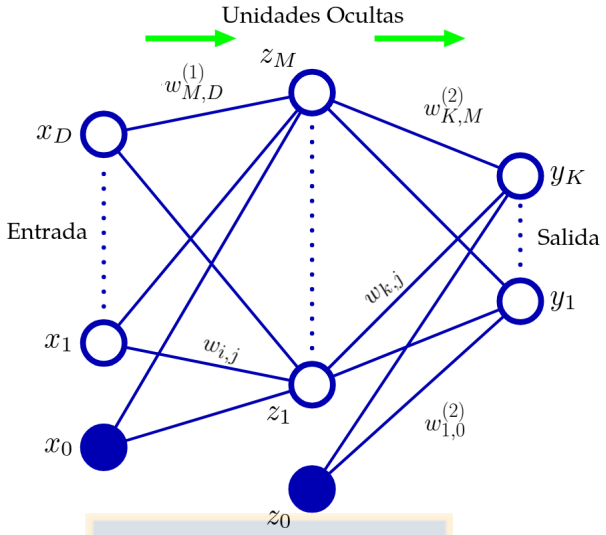


Figura 3.3: Red neuronal con una capa oculta. En la capa de entrada se encuentra el input x_D que conectan con la capa oculta Z_M por medio de los pesos W_{MD}. Luego, desde la capa oculta se traspasa una señal a la capa de salida y_K por medio de los pesos W_{KM} [3]

relaciones existentes entre ellas.

La salida de una red neuronal y está definida en función de su entrada x y los pesos w asociados a cada conexión neuronal a las distintas capas ocultas [3]. En la imagen 3.3 podemos ver una red neuronal con una capa oculta. Nótese que cada entrada x_i con $i = 0, ..., D$ representa una característica y está a su vez conecta con cada una de las neuronas por medio de los pesos w_{ji}. Al igual que en el cerebro, es necesario definir una función de activación que controle el funcionamiento asíncrono de las neuronas dado un estímulo. El estímulo se define como la combinación lineal entre la entrada y los pesos. Consecutivamente, el resultado de la operación pasará por una función h que controla el estado prendido o apagado de la neurona. Formalmente podemos decir que:

$$y_k(x, w) = \sigma(\sum_{j=1}^{M} w^{(2)}_{kj} h(\sum_{i=1}^{D} w^{(1)}_{ji} x_i + w^{(1)}_{j0}) + w^{(2)}_{k0}) \tag{3.1}$$

donde M es la cantidad de neuronas y D la dimensionalidad de la entrada (características observables). Nótese que w_{j0} y w_{k0} corresponden al sesgo asociado a cada neurona. La salida de cada neurona puede ser conectada con otra capa oculta de neuronas para aumentar el nivel de abstracción (modelo deep learning). Finalmente, tendremos la capa de salida y donde la combinación lineal, junto a los pesos, definirá la salida del modelo. En caso de estar trabajando con probabilidades, es necesario pasar el resultado por alguna función que normalice la salida.

3.2.1. Retropropagación del error

Una de los principales atractivos de las Redes Neuronales es su capacidad de aprendizaje basado en la propagación del error a través de los pesos. Una vez terminado el proceso de comunicación hacia adelante debemos comparar nuestra salida con una etiqueta [16]. Para calcular la pérdida entre la salida y la etiqueta usamos una función de costo, para clasificación no binaria podemos utilizar entropía cruzada categorica H la cual puede interpretarse como la longitud esperada en bits para identificar la clase real, comparando la distribución aproximada del modelo con la real. En la práctica, debido a que no conocemos la distribución real de los datos, calculamos esta medida desde las observaciones, comparando etiquetas reales y con etiquetas predichas $\hat{y}$ para un conjunto de N objetos y C clases [9].

$$H(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} (\mathbb{1}(y_{ij} \in C_j) log(p_{ij})) \tag{3.2}$$

Donde $\sum_j p_{ij} = 1$. Note que cuando realizamos clasificación de dos o más clases nuestro modelo entrega un vector de probabilidades de largo C. Luego la clase predicha $\hat{y}$ será

$$\hat{y} \ = \ argmax_j(p_{ij}), \ con \ j = 1, ..., C \tag{3.3}$$

Una vez calculado el error H debemos actualizar los pesos comunicando este valor hacia las capas anteriores. Este procedimiento se realiza mediante la derivación de la función de costo H respecto a los pesos W de cada capa $l = 0, 1, ..., L$ haciendo uso de la regla de la cadena [24] para calcular un vector de gradientes $\bigtriangledown_w$. Una

vez calculando los gradientes la regla iterativa de actualización se define como,

$$w^{t+1} = w^t - \eta \bigtriangledown_w H \tag{3.4}$$

donde η es la tasa de aprendizaje para controlar el descenso del gradiente; a mayor tasa mayor será el salto entre tiempos. Gráficamente, estamos actualizando los pesos en dirección contraria al gradiente, véase la figura 3.4.

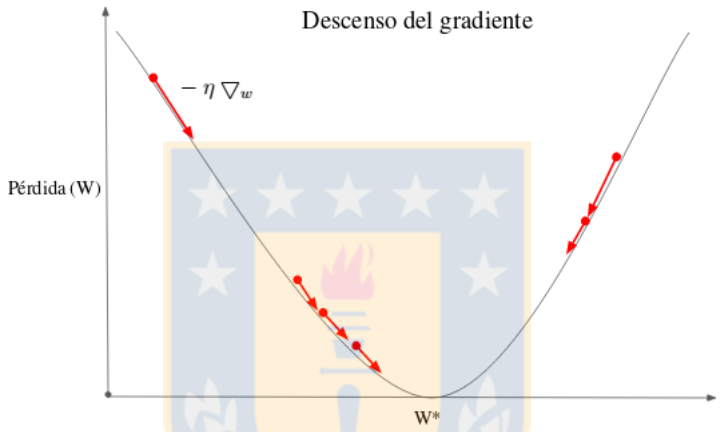


Figura 3.4: Representación gráfica del descenso del gradiente. W* representa la configuración de valores óptimos para los pesos. La flecha roja se mueve en magnitud y sentido según el gradiente. La longitud de cada flecha depende la magnitud de las derivadas mientras que el espaciado (o salto) entre flechas está determinado por la tasa de aprendizaje

3.3. Redes Neuronales Recurrentes

Para estudiar la correlación entre datos distribuidos en el tiempo es necesario mantener información contextual. El contexto es información útil desde tiempos pasados, la cual nos permite hacer una inferencia en el tiempo actual. Las redes recurrentes tienen una matriz de pesos adicional - además de las de entrada y salida - por cada paso de tiempo; esa matriz representa el contexto. Las redes neuronales recurrentes añaden un ciclo en cada neurona permitiendo así, mantener información en el tiempo mediante una nueva matriz de pesos [13]. La imagen 3.5 muestra la estructura del modelo. Nótese ahora que la entrada es un tensor

de tres dimensiones donde cada vector en la dimensión temporal contiene una magnitud. Durante el proceso de entrenamiento debemos retropropagar el error

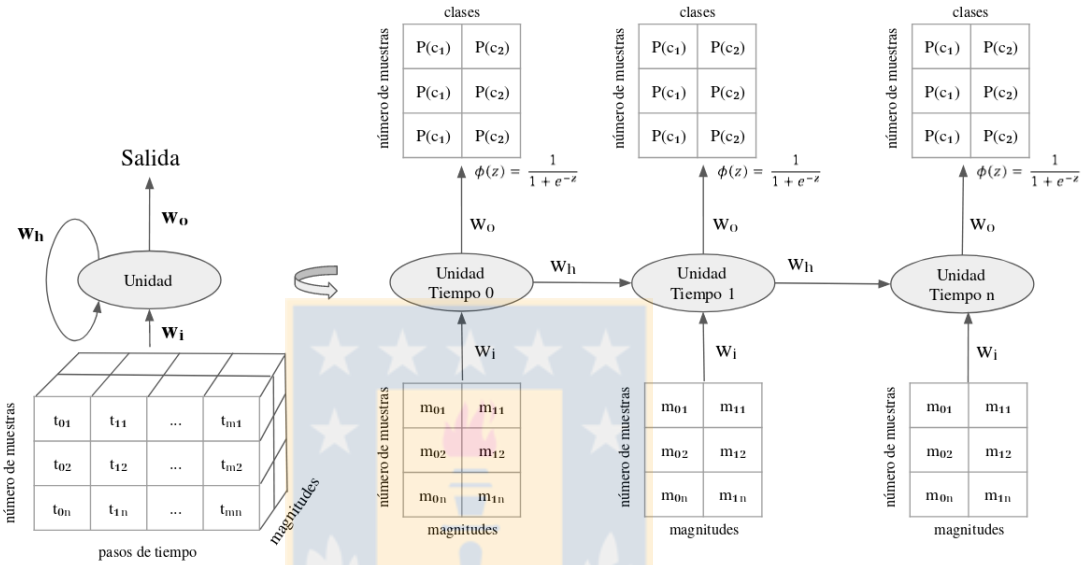


Figura 3.5: [Izquierda] Red neuronal recurrente comprimida. Se puede ver la entrada como una matriz tridimensional donde para cada muestra se tiene tiene una serie de tiempos con sus respectivas magnitudes. [Derecha] Red neuronal recurrente desenrollada. Se puede observar cada tiempo dentro del ciclo. Note que por cada tiempo está entrando una matriz con magnitudes asociadas a cada muestra en el tiempo actual. W_i, W_o y W_h son los pesos asociados a la entrada la salida y las capas ocultas respectivamente. La salida de cada unidad puede ser evaluada en alguna función ϕ con el objetivo de acotar el resultado (por ej. cuando trabajamos con probabilidades)

calculado entre la salida del modelo y la etiqueta real [11]. En el contexto de RNN, la dependencia entre distintos tiempos que se encuentran muy espaciados genera problemas de explosión o desaparición del gradiente [2]. Para lidiar con este problema, Sepp Hochreiter y Jürgen Schmidhuber, proponen una nueva unidad (neurona) cuya estructura permite mantener largas dependencias en el tiempo [12].

3.4. Long Short Term Memory

Las unidades LSTM son un tipo de neurona cuyo comportamiento permite almacenar información a largo plazo por naturaleza [12]. El núcleo de esta unidad consiste en un estado de celda (del inglés cell state), sobre el cual escribiremos o borraremos información relevante desde todos los pasos de tiempo. Como se puede ver en la imagen 3.6 cada unidad está compartiendo el mismo estado de celda. La celda será modificada conforme vayamos añadiendo o eliminando información durante el proceso de entrenamiento. Realizaremos operaciones de escritura y

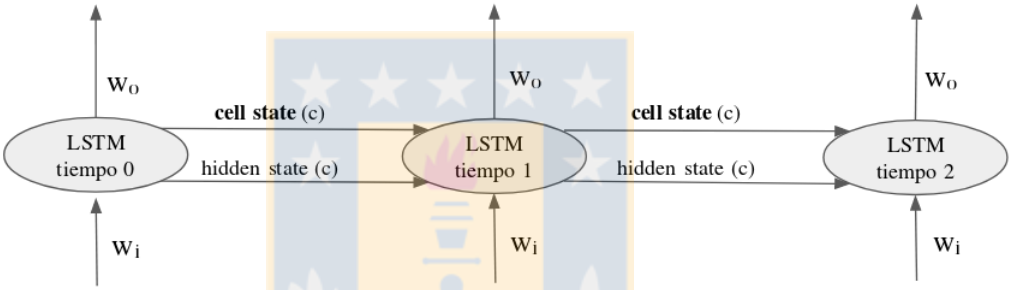


Figura 3.6: Simple representación de una red neuronal recurrente con unidades LSTM. Se puede ver que el estado de celda (cell state), al igual que el oculto (hidden state), cruza a través de los pasos de tiempo

borrado sobre el estado de celda por medio de compuertas específicas. La imagen 3.7 muestra tres compuertas principales: olvidof_t, recuerdo s_t y salida o_t. La idea es simple,

1. Compuerta del olvido (f_t): considera la entrada en un tiempo t para eliminar información del estado de celda que ya no será relevante para el futuro.

2. Compuerta del recuerdo (s_t): guarda en el estado de celda la información que puede ser útil desde la entrada en el tiempo t.

3. Compuerta de salida (o_t): genera una predicción para el tiempo siguiente basándose en el estado de celda actual. También conocida como estado oculto (del inglés hidden state).

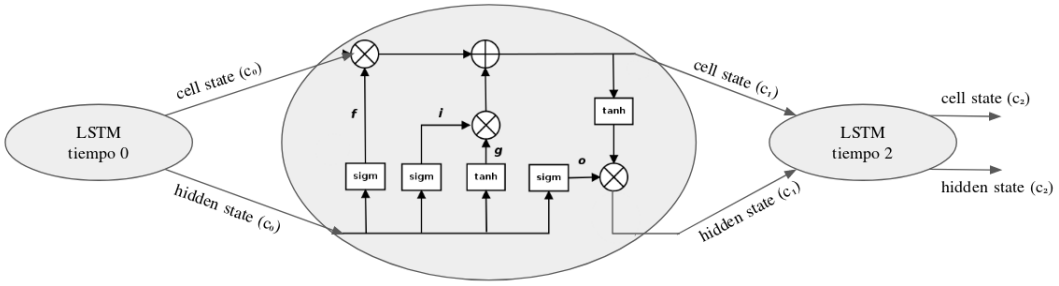


Figura 3.7: LSTM en detalle. Se puede ver cada una de las puertas interactuando por medio de las operaciones y funciones correspondientes.

Adicionalmente, podemos encontrar una compuerta de entrada cuya función consiste en transformar la entrada mediante combinaciones lineales. Cada una de estas compuertas puede ser vista como una red neuronal simple aprendiendo cada una de las acciones descritas. Matemáticamente, cada estado se define como:

$$c_t = f_t \cdot c_{t-1} + s_t \cdot \sigma(x_t W_{cx} + h_{t-1} W_{ch} + b_f) \tag{3.5}$$

$$h_t = o_t \cdot tanh(c_t) \tag{3.6}$$

y cada compuerta

$$f_t = \sigma(x_t W_{fx} + h_{t-1} W_{fh} + c_{t-1} W_{fc} + b_f) \tag{3.7}$$

$$s_t = \sigma(x_t W_{sx} + h_{t-1} W_{sh} + c_{t-1} W_{sc} + b_s) \tag{3.8}$$

$$o_t = \sigma(x_t W_{ox} + h_{t-1} W_{oh} + c_{t-1} W_{oc} + b_o) \tag{3.9}$$

Para la compuerta de olvido (3.7), x_t es la entrada en el tiempo actual t. W_{fx}, W_{fh}, W_{fc} son los pesos para la compuerta de olvido f_t con respecto a la entrada x_t, al estado oculto h_{t-1} y el estado de celda c_t. b_f es el sesgo de la compuerta. La misma lógica siguen las demás compuertas s_t (3.8) y o_t (3.9).
Note que el estado de celda (3.5) elimina contenido desde c_{t-1} mediante el producto punto por f_t (3.7) y añade nueva información desde s_t haciendo el producto punto con $\sigma(x_t W_{cx} + h_{t-1} W_{ch} + b_f)$ que aprende los pesos W_{cx} y W_{ch} desde el pasado h_{t-1} y el presente x_t.

3.5. Phased Long Short Term Memory

Las Phased Long Short Term Memory (PLSTM) [18] son una extensión de las LSTM cuya estructura les permite aprender la periodicidad del muestreo. Para realizar esto, las PLSTM añaden una puerta del tiempo k_t cuya función será dar paso completo, parcial o nulo al flujo de información.

Consideremos nuevamente la ecuación 3.5 y 3.6. Bajo este nuevo modelo no necesitaremos actualizar a cada tiempo los nuevos estados, por lo tanto, definimos nuevos estados candidatos como c'_l y h'_l, los cuales estarán condicionados por el resultado de la compuerta del tiempo k_l:

$$c'_l = f_l \cdot c_{l-1} + i_l \cdot \sigma(x_l W_{cx} + h_{l-1} W_{ch} + b_f) \tag{3.10}$$

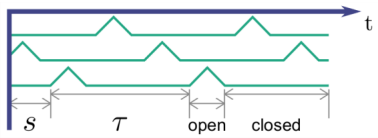
$$c_l = k_l \cdot c'_l + (1 - k_l) \cdot c_{l-1} \tag{3.11}$$

$$h'_l = o_l \cdot tanh(c_l) \tag{3.12}$$

$$h_l = k_l \cdot h'_l + (1 - k_l) \cdot h_{l-1} \tag{3.13}$$

donde ϕ representa la fase de abertura dentro del ciclo rítmico

$$k_l = \begin{cases} \frac{2\phi}{r_{on}} & \text{if } \phi < \frac{1}{2} r_{on} \\ 2 - \frac{2\phi}{r_{on}} & \text{if } \frac{1}{2} r_{on} < \phi < r_{on} \\ \alpha \phi_t & \text{otherwise} \end{cases}$$



$$\phi = \frac{(t_l - s) mod \tau}{\tau} \tag{3.14}$$

τ controla el periodo real de la señal, r_{on} controla el periodo de abertura, s define el desplazamiento inicial del muestreo y t es el tiempo actual.

Intuitivamente, las PLSTM aprenden el periodo de muestreo en los datos, lo cual permite discriminar pasos de tiempo y acelerar el aprendizaje.

Capítulo 4

METODOLOGÍA

Se utilizó una metodología de comparación experimental que consta de dos etapas principales. En primer lugar, se preprocesaron tanto las curvas de luz de supernovas como las de objetos reales definiendo un número máximo de observaciones por muestra. Se normalizó el instante de observación en MJD (del inglés Modified Julian Day) de cada curva de luz calculando la diferencia entre los tiempos por la primera observación en cada serie de tiempo. Para el caso particular de las supernovas, se completaron datos faltantes utilizando valores aleatorios desde una distribución uniforme. En segundo lugar se implementó la arquitectura del estado del arte de Charnock et al. (2017) [6]. De manera consecutiva se realizaron las pruebas correspondientes utilizando PLSTM sobre las distintas clases de supernovas. Luego, se emplearon datos reales correspondientes a distintos tipos de estrellas variables con una mayor cantidad de observaciones (50, 60, 70, 80 y 100). En ambos experimentos se realizó una búsqueda exhaustiva sobre la cantidad de neuronas que optimiza la exactitud de los resultados.

4.1. Sobre los datos

4.1.1. Simulación de Supernovas

Se utilizaron simulaciones de 8 tipos de supernovas[15]. La tabla 4.1 muestra la cantidad de supernovas por tipo. Cada supernova está descrita por parámetros de la simulación y magnitudes en 4 bandas. La figura 4.1 muestra parte de la tabla de observaciones generada.

Indice	Tipo	Cantidad
0	Ia	20275
1	II	48075
2	Ibc	1050
3	IIn	3115
4	IIP	745
5	IIL	1705
6	Ib	5835
7	Ic	4480

Cuadro 4.1: Cantidad de supernovas por clase

	id_sn	time	RA	DECL	MWEBV	REDSHIFT FINAL	g	r	i	z	g_e	r_e	i_e	z_e	SIM REDSHIFT	class
0	393133	1.10850	34.5	-5.5	0.0227	0.695	36.7300	8.8670	-16.9800	-1.8180	30.750	3.599	7.060	2.926	0.7001	2
1	393133	2.09000	34.5	-5.5	0.0227	0.695	-5.3990	0.4444	6.1790	3.0800	9.315	3.863	6.318	2.922	0.7001	2
2	393133	3.10750	34.5	-5.5	0.0227	0.695	0.6672	4.0030	-0.3458	-0.5002	7.694	2.655	4.790	3.017	0.7001	2
3	393133	7.13700	34.5	-5.5	0.0227	0.695	6.9330	-3.9910	1.7490	-2.4030	3.789	1.758	3.200	2.936	0.7001	2
4	393133	8.15075	34.5	-5.5	0.0227	0.695	0.4684	-2.3700	0.3952	2.4730	2.000	1.875	2.783	3.190	0.7001	2

Figura 4.1: Datos asociados a las observaciones de una supernova. [De izquierda a derecha] Identificador de la supernova, diferencia de tiempo, ascensión (del inglés RA), declinación, exceso esperado de B-V debido a la extinción por polvo en la Vía Láctea (MWEBV, del inglés Milky Way Extintion B-V), desplazamiento al rojo original, magnitud en banda g, magnitud en banda r, magnitud en banda i, magnitud en banda z, desplazamiento al rojo simulado, Clase.

Preprocesamiento de los datos

Los datos fueron preprocesados de manera que sirviesen como entrada a la red neuronal recurrente. Los dos problemas principales se detallan a continuación.

- Ausencia de Observaciones: La simulaciones consideran datos faltantes (del inglés missing values) en cada tiempo. Esto quiere decir que para un mismo tiempo, digamos t_2, podemos tener una o más bandas sin magnitud, como se ve en la tabla 4.2. Para solucionar esto simplemente rellenamos el espacio con un valor aleatorio distribuido uniformemente y acotado por los valores más cercanos; en el ejemplo anterior, para rellenar i_2 tomaríamos un valor aleatorio, distribuido uniformemente, desde $]i_1, i_3[$

- Distinta cantidad de observaciones: La cantidad de observaciones para una

time	...	g	r	i	z
t_1	...	g_1	r_1	i_1	z_1
t_2	...	g_2	r_2		z_2
t_3	...	g_3	r_3	i_3	z_3

Cuadro 4.2: Missing values en la observación

supernova es independiente del resto. Por lo tanto, podemos tener distintas longitudes para la serie de tiempo. Como las redes neuronales realizan operaciones matriciales, es fundamental que la dimensionalidad de entrada sea definida y no varíe durante el entrenamiento. Para resolver este problema utilizamos una técnica de enmascaramiento la cual nos permite manejar largos variables de series de tiempo. En la práctica realizamos los siguientes pasos:

- se definió un largo máximo para todas las series de tiempos considerando las curvas de luz en el conjunto de muestras
- se completó con 0's aquellas curvas de luz que tenían menos observaciones que el máximo. Luego, se cortaron las serie de tiempo en la última capa efectiva (antes del primer 0 de la máscara); esto con motivo de retropropagar el error sin considerar la máscara.

4.1.2. Conjunto de datos reales MACHO

MACHO (del inglés Massive Compact Halo Object) es un conjunto de observaciones captadas por el gran telescopio de Melbourne entre los años 1992 y 1999 [1]. Reúne un gran conjunto de estrellas variables; la imagen 4.2 muestra ejemplos de curvas de luz para cada uno de los objetos astronómicos en el conjunto de datos. La cantidad de observaciones promedio es de alrededor de 1000 observaciones por objeto. Durante la clasificación, el procesamiento de observaciones para una RNN se vuelve intratable cuando se maneja una gran cantidad de pasos de tiempos ($\sim$ 500 observaciones) [14]. En ese contexto, cortamos las curvas de luz generando subconjuntos con menos observaciones (50, 60, 70, 80 y 100). La tabla 4.3 muestra la cantidad de objetos utilizados en la fase de entrenamiento

(70 %), validación (20 %) y prueba (10 %).

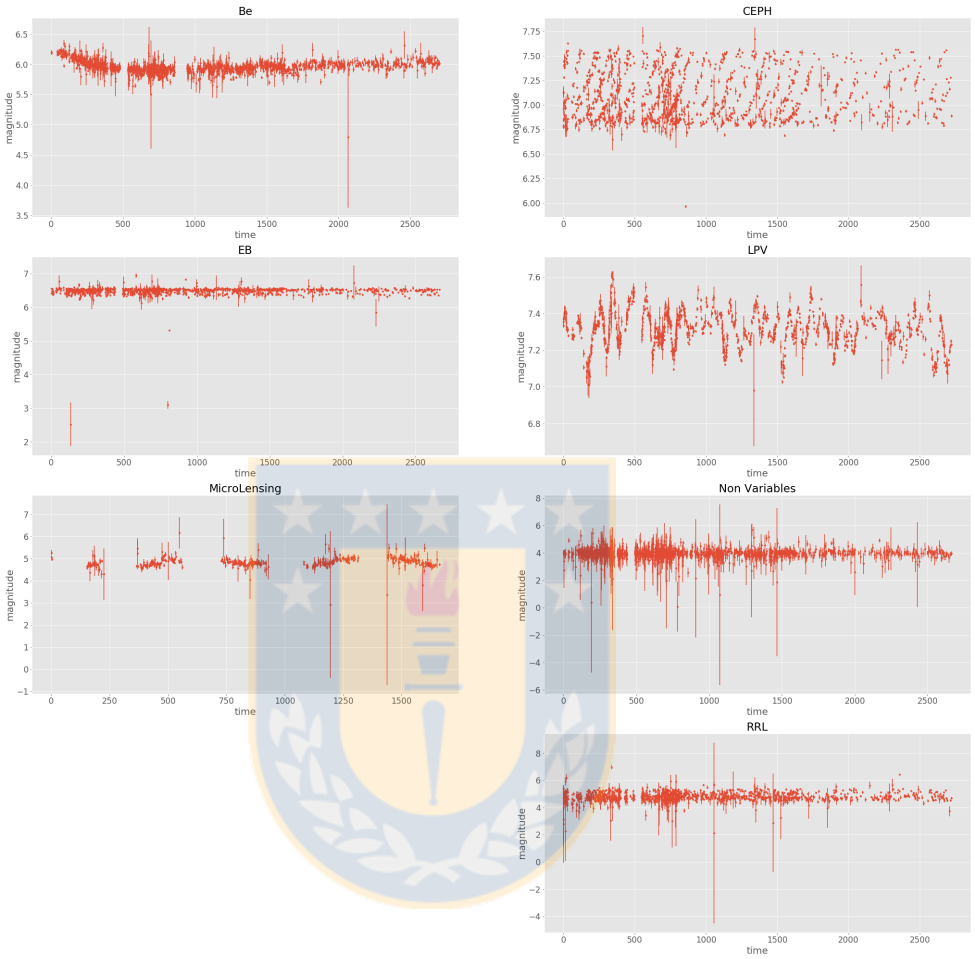


Figura 4.2: Curvas de luz utilizadas durante el entrenamiento. Estas fueron extraídas para cada una de las clases desde conjunto de datos MACHO

4.2. Arquitecturas Utilizadas

La arquitectura de Charnock [6] consiste en una LSTM con dos capas ocultas utilizando Dropout para regularizar [25]. La concatenación de dos capas de LSTM consiste en introducir la salida de una como la entrada de la otra; no es necesario realizar transformaciones de la salida en este caso.

Todas las redes neuronales aprenden un modelo complejo de varios parámetros. Más aún, durante el entrenamiento estamos modificando los pesos y por ende la

Objeto	Cantidad
Binaria Eclipsante	10240
Cefeida	10100
Estrella Be	10200
RR Lyrae	10834
Invariables	10354
Microlente Gravitacional	10062
Estrella variable de periodo largo	10834

Cuadro 4.3: Composición del conjunto de datos MACHO completo

geometría de nuestro modelo cambia. Algunos problemas que pueden ser generados tienen que ver con la forma de la entrada en cada paso de tiempo; es posible que el gradiente explote o desaparezca [2]. Una técnica destinada a resolver estos problemas es Batch Normalization [8] de manera que se normalicen los parámetros del modelo en cada paso de tiempo. Charnock no utiliza Batch Normalization en su modelo original, sin embargo, nosotros entrenamos utilizando esta técnica.

El modelo propuesto utiliza dos capas PhasedLSTM conectadas entre ellas. Cabe señalar que las Phased LSTM reciben dos matrices de entrada:

- las magnitudes y errores asociadas a la curva de luz
- el tiempo para cada una de las observaciones

En la figura 4.3 se puede ver el modelo final de Charnock et al. 2017 cuya composición consiste en utilizar dos capas ocultas con 16 neuronas en cada compuerta. Por medio de la capa dropout las neuronas se encienden y apagan con una probabilidad de 0,5. Además, antes de cada tiempo estamos añadiendo la capa de Batchnormalization que transforma la entrada para cada LSTM. La figura 4.4 muestra la arquitectura del modelo propuesto. Las unidades Phased LSTM tienen aplicadas las mismas técnicas de regularización (Dropout y Batch Normalization). Note que, en ambos modelos, solo estamos considerando la última capa para realizar la clasificación. Eventualmente podríamos generar una predicción por cada tiempo, no obstante, esto escapa de los objetivos de este trabajo.

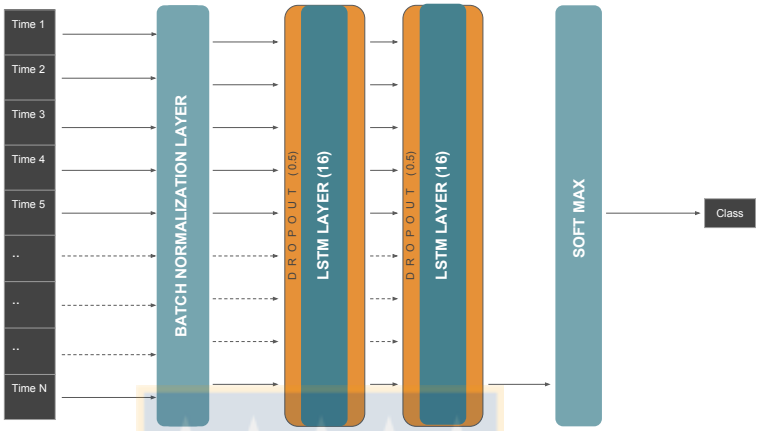


Figura 4.3: Arquitectura de Charnock con Batch Normalization

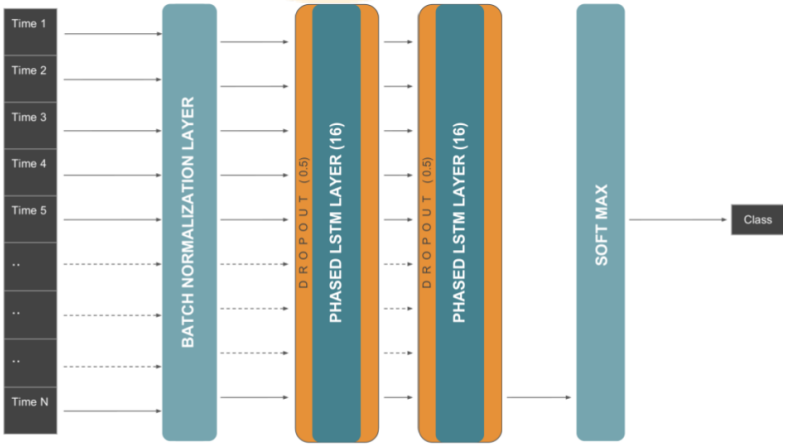


Figura 4.4: Arquitectura propuesta

Capítulo 5

EXPERIMENTOS Y RESULTADOS

5.1. Métricas

Antes de describir las métricas es necesario definir los casos que puede entregar nuestro clasificador. Cuando el número de etiquetas es mayor a 2 (clasificación multiclase) utilizamos una matriz de confusión. Las etiquetas reales están representadas por columnas mientras que las predichas se encuentran en las filas. La ecuación 5.1 muestra una matriz de confusión de n clases reales y d clases predichas por el modelo. Usualmente, $n = d$.

$$M = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix} \tag{5.1}$$

En general, podemos definir los siguientes casos:

- Verdaderos Positivos: $\sum_i^n \sum_j^d CM_{i=j}$
- Falsos Positivos: $\sum_i^n M_{i \neq j}$
- Falsos Negativos: $\sum_j^d M_{i \neq j}$

La clasificación óptima puede ser vista como una matriz diagonal.

5.1.1. Exactitud

$$Exactitud = \frac{Verdaderos\ Positivos}{Cantidad\ total\ de\ muestras} \tag{5.2}$$

5.1.2. Puntaje F1

$$Precision : \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos\ +\ Falsos\ Positivos} \tag{5.3}$$

$$Exhaustividad : \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos\ +\ Falsos\ Negativos} \tag{5.4}$$

$$F1 = 2 \times \frac{precision \times exhaustividad}{precision + exhaustividad} \tag{5.5}$$

5.1.3. Función de Costo para Calcular la Perdida

La función utilizada fue la de entropía cruzada categórica descrita anteriormente en la ecuación 3.2. Esta función recibe de entrada la salida de la ultima capa de la RNN cuya función de activación es una SoftMax

$$P(y = k|x) = \frac{e^{x^T w_k}}{\sum_{k=1}^{K} e^{x^T w_k}} \tag{5.6}$$

donde K representa la cantidad de clases, x^T el vector de activaciones de la ultima capa y w_k los pesos asociados a cada clase k

5.2. Experimentos sobre el Conjunto de Supernovas

La fase de entrenamiento comprendió 200 épocas. Cada época corresponde a un paso completo (y retropropagación) del conjunto total de datos a través la red. Se utilizaron 42640 muestras para la fase de entrenamiento, 25584 para prueba y 17056 para validación. Solo durante la fase de entrenamiento ajustamos los pesos, el conjunto de prueba nos permite ver que tan bien generaliza nuestro modelo. Dicho de otra manera son datos que nunca ha visto el modelo y por lo tanto no ha ajustado sus pesos considerándolos. Utilizando el conjunto de validación podemos ver convergencia en la perdida y, en función de esto, podemos detener el entrenamiento. El criterio de convergencia utilizado fue tal que la desviación estándar del promedio de los pesos en 4 épocas seguidas no superara los 0.05 Finalmente, se utiliza un conjunto de prueba -el cual no tiene relación alguna con

el criterio de convergencia- para medir el desempeño final del modelo.

5.2.1. Ajustando hiperparámetros

La primera parte de la experimentación consistió en definir empíricamente el mejor número de neuronas para los modelos. Recordemos que cada unidad tiene compuertas que pueden ser vistas como redes neuronales con pesos independientes. El número de neuronas ocultas determinan la cantidad de pesos de cada compuerta. La tabla 5.1 muestra los resultados para ambos modelos y distintos números de neuronas. El máximo se logra con 128 neuronas, sin embargo, el modelo crece en complejidad y por ende en tiempo de entrenamiento. La elección para este experimento fueron 64 neuronas.

	LSTM	Phased LSTM
16	0.846	0.756
32	0.866	0.800
64	0.860	0.831
128	0.869	0.835

Cuadro 5.1: Exactitud vs cantidad de neuronas

5.2.2. Ejecutando los Modelos

La figura 5.1 muestra las matrices de confusión asociadas a cada modelo y la tabla 5.2 resume los resultados para F1 score, pérdida (entropía cruzada categórica) y exactitud. En la figura 5.2 se puede ver las curvas de aprendizaje. Por motivos de rendimiento debemos separar el conjunto total de datos en pequeños lotes (del inglés batch) e ir sumando las pérdidas parcialmente. Para este

	F1	Perdida	Exactitud
LSTM	0.85	0.51	0.87
PLSTM	0.76	0.64	0.80

Cuadro 5.2: Puntajes obtenidos después de la fase de entrenamiento sobre el conjunto de prueba

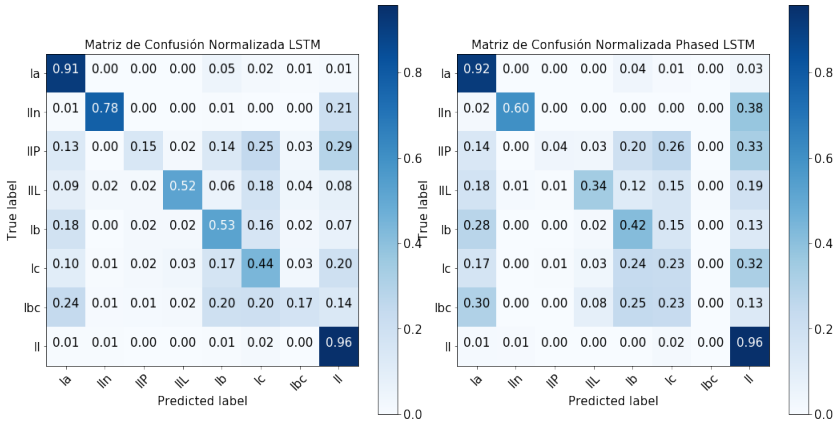


Figura 5.1: Matrices de confusión sobre el conjunto de prueba considerando LSTM y PLSTM con 64 neuronas

experimento utilizamos un batch de 200 muestras. Las iteraciones corresponden a la cantidad de veces que un batch pasa por la red.

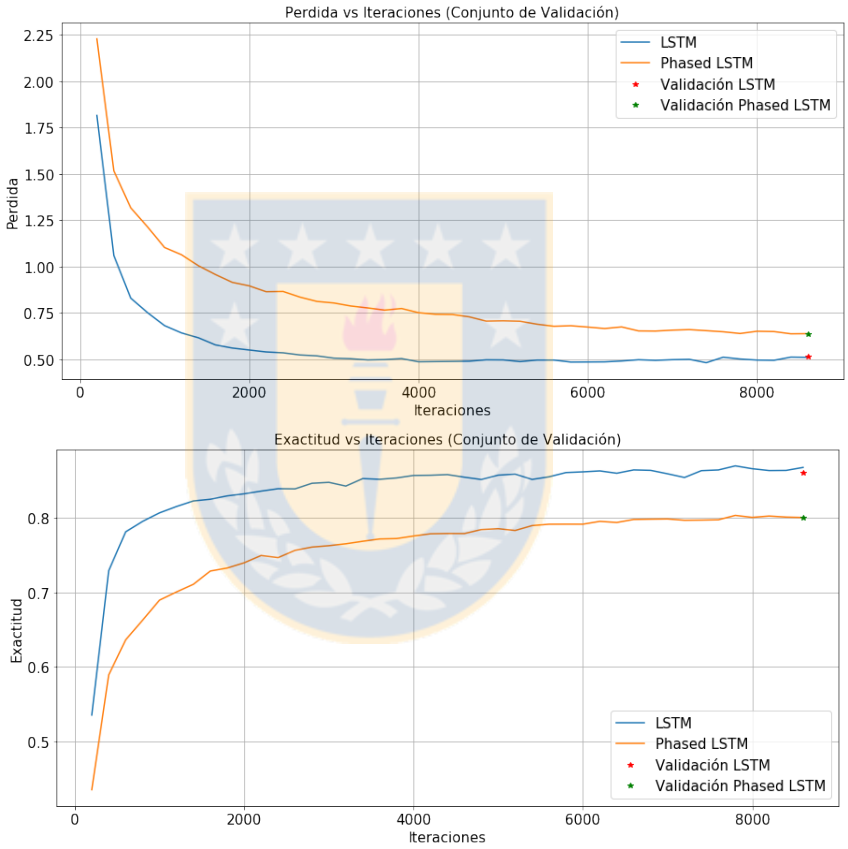


Figura 5.2: Curvas durante el proceso de entrenamiento para el conjunto de Validación y Prueba

5.3. Experimento sobre el conjunto de datos MACHO

Utilizamos distintos largos para las series de tiempo: 50, 60, 70, 80 y 100 observaciones. La cantidad de neuronas escaló desde 64, 128, 256 hasta 512.
El periodo de entrenamiento fue de 200 épocas, cantidad suficiente para mostrar tendencia de crecimiento en términos de exactitud. El tamaño del batch utilizado fue de 1024 muestras. Las figuras 5.4 y 5.5 muestran la curva de aprendizaje sobre el conjunto de validación.

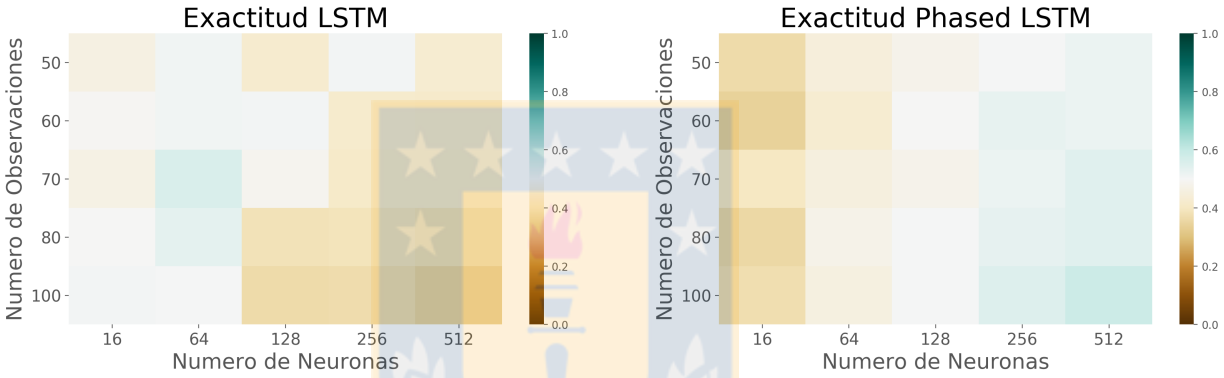


Figura 5.3: Exactitud vs Número de observaciones y neuronas en la capa oculta. Matriz comparativa entre LSTM y PLSTM

5.4. Discusión

El uso de PLSTM para el experimento con supernovas aumenta la perdida y en consecuencia la exactitud disminuye comparativamente. Las PLSTM aprenden la periodicidad del muestreo desde un espacio de tiempo real, por lo que permiten la actualización del estado de celda solo en un pequeño porcentaje del ciclo de apertura. La capacidad de memoria -en la compuerta del tiempo - está directamente relacionada con la cantidad de neuronas. El desempeño de las PLSTM radica en el número de neuronas. Esto quiere decir, a mayor cantidad de neuronas mejor la exactitud de nuestro modelo. Como se puede ver en la imagen 5.3 existe un comportamiento inverso entre ambos modelos. Esto nos permite afirmar - de manera temprana - que las Phased LSTM tienen ventaja comparativa respecto a las LSTM cuando la entrada crece en número de observaciones y/o el modelo se

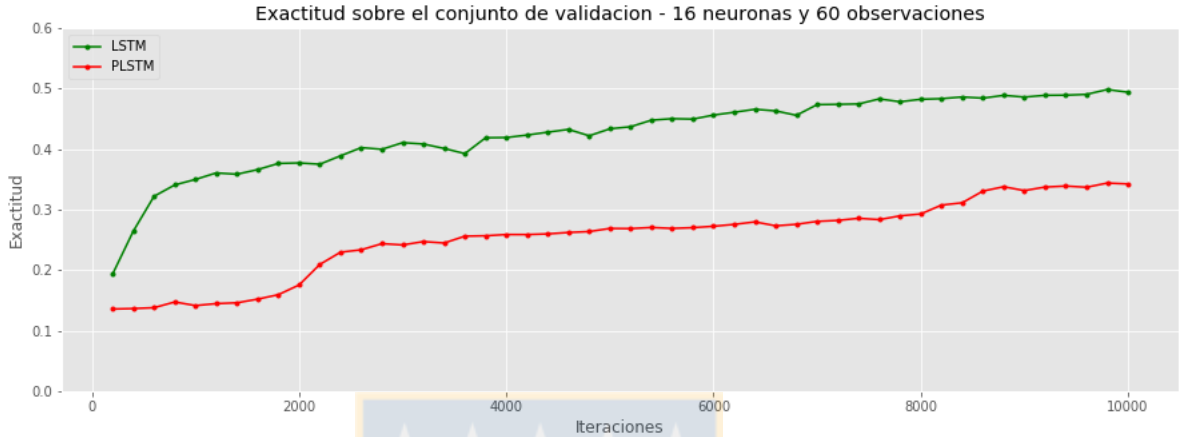


Figura 5.4: Curva de validación sobre un conjunto de neuronas pequeño (16). En este caso, LSTM supera en términos de exactitud a las PhasedLSTM

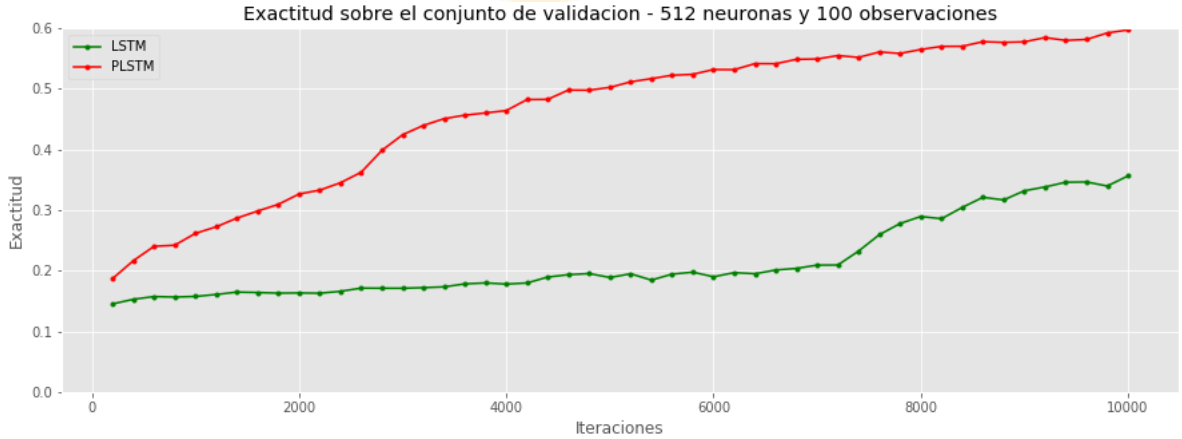


Figura 5.5: Curva de validación sobre un conjunto de neuronas grande (512). En este caso, Phased LSTM supera en términos de exactitud a las LSTM

complejiza aumentando el número de neuronas. Si bien es claro el comportamiento descrito anteriormente para los datos de supernovas y MACHO, la evaluación formal de esta hipótesis quedará para una investigación futura.

Capítulo 6

CONCLUSIONES

En este trabajo abordamos el problema de clasificación utilizando una metodología que consistió en implementar y comparar dos modelos de redes recurrentes con unidades LSTM y PLSTM sobre datos simulados de supernovas y reales de MACHO.

En la experimentación con supernovas se realizó una búsqueda exhaustiva sobre la cantidad de neuronas concluyendo de manera empírica que las LSTM se comportan mejor para todos los casos presentados. En particular seleccionamos 64 neuronas, cuyo comportamiento obtuvo obtuvo un 87 % de exactitud y un 85 % de puntaje F1 para las LSTM, y un 80 % y 70 % de puntaje F1 para las PLSTM.

En el segundo experimento consideramos curvas de luz reales desde el conjunto de datos MACHO. Con el objetivo de estudiar el desempeño en profundidad de las LSTM y PLSTM, seleccionamos 50, 60, 70, 80 y 100 observaciones. Al variar la cantidad de observaciones y neuronas (en ambos modelos) obtuvimos mejores resultados para la PLSTM con un 60 % de exactitud versus un 56 % de las LSTM. Por otro lado, los mejores resultados para ambos modelos no coinciden en el número de neuronas y observaciones. Podemos concluir que a mayor número de neuronas las PLSTM obtienen mejor comportamiento en términos de exactitud. No obstante, la cantidad de observaciones no muestra una tendencia visible desde los resultados; para 16 neuronas de tipo PLSTM los mejores resultados se encuentran en 80 y 60 observaciones.

En general, las PLSTM no constituyen una mejora per-se. Existe una relación entre la capacidad de aprendizaje de la compuerta del tiempo (cantidad de neuronas) y la cantidad de observaciones en la serie de tiempo. Como se muestra en los resultados, la elección del tipo de neurona no solo depende de los factores mencionados anteriormente, sino del tipo de problema y los datos a considerar. Únicamente, cuando usamos el conjunto de datos MACHO, las PLSTM convergen de manera mucho más rápido que las LSTM, aprendiendo la periodicidad del muestreo cuando la cantidad de neuronas es mayor.

Finalmente, de esta investigación se desprende una hipótesis respecto a la arquitectura de la red la cual define que a mayor cantidad de neuronas ocultas mejor el desempeño de las PLSTM. La validación y estudio de esta hipótesis serán analizadas en otra investigación futura.

Bibliografía

[1] Ch Alcock, RA Allsman, D Alves, TS Axelrod, AC Becker, DP Bennett, KH Cook, KC Freeman, K Griest, J Guern, et al. The macho project large magellanic cloud microlensing results from the first two years and the nature of the galactic dark halo. The Astrophysical Journal, 486(2):697, 1997.

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157–166, 1994.

[3] Christopher M Bishop. Pattern recognition and machine learning, 2006. 60(1):78–78, 2012.

[4] Kirk D Borne. Astroinformatics: data-oriented astronomy research and education. Earth Science Informatics, 3(1-2):5–17, 2010.

[5] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. Journal of the American statistical Association, 65(332):1509–1526, 1970.

[6] Tom Charnock and Adam Moss. Deep recurrent neural networks for supernovae classification. The Astrophysical Journal Letters, 837(2):L28, 2017.

[7] LSST Dark Energy Science Collaboration et al. Large synoptic survey telescope: dark energy science collaboration. arXiv preprint arXiv:1211.0310, 2012.

[8] Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. arXiv preprint arXiv:1603.09025, 2016.

[9] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. Annals of operations research, 134(1):19–67, 2005.

[10] JB Doggett and D Branch. A comparative study of supernova light curves. The Astronomical Journal, 90:2303–2311, 1985.

[11] Jiang Guo. Backpropagation through time. Unpubl. ms., Harbin Institute of Technology, 2013.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[13] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8):2554–2558, 1982.

[14] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078, 2015.

[15] Richard Kessler, Bruce Bassett, Pavel Belov, Vasudha Bhatnagar, Heather Campbell, Alex Conley, Joshua A Frieman, Alexandre Glazov, Santiago González-Gaitán, Renée Hlozek, et al. Results from the supernova photometric classification challenge. Publications of the Astronomical Society of the Pacific, 122(898):1415, 2010.

[16] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160:3–24, 2007.

[17] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4):115–133, 1943.

[18] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In Advances in Neural Information Processing Systems, pages 3882–3890, 2016.

[19] Saul Perlmutter, G Aldering, M Della Valle, S Deustua, RS Ellis, S Fabbro, A Fruchter, G Goldhaber, DE Groom, IM Hook, et al. Discovery of a supernova explosion at half the age of the universe. Nature, 391(6662):51, 1998.

[20] Yu P Pskovskii. The photometric properties of supernovae. Soviet Astronomy, 11:63, 1967.

[21] Robert H Shumway and David S Stoffer. Time series analysis and its applications. Studies In Informatics And Control, 9(4):375–376, 2000.

[22] Brian L Smith, Billy M Williams, and R Keith Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. Transportation Research Part C: Emerging Technologies, 10(4):303–321, 2002.

[23] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big data: astronomical or genomical? PLoS biology, 13(7):e1002195, 2015.

[24] Paul Werbos. Beyond regression:"new tools for prediction and analysis in the behavioral sciences. Ph. D. dissertation, Harvard University, 1974.

[25] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.