
Proyecto Final Statistical Learning II

Ricardo Javier Mendoza Villatoro
Universidad Galileo de Guatemala
21002007@galileo.edu

Abstracto

El siguiente documento esta formado por 3 mini proyectos de Deep Learning. En la primera parte podemos ver una Red Neuronal simple que nos da la opcion a servir como recomendador de canciones hacia un agente en especifico partiendo de sus gustos anteriores como referencia. La segunda parte la toma una Red Neuronal Convolutacional la cual nos ayuda a clasificar 5 de las mas conocidas enfermedades en la pupila humana, el entrenamiento del modelo se dio a partir de imagenes de pupilas con las enfermedades presentes. Como ultima parte del proyecto contamos con una Red Neuronal Recurrente la cual tiene como tarea el estimar el comportamiento del precio del cierre del día de la criptomoneda mas famosa, Bitcoin, el entrenamiento esta basado revisar los ultimos 5 dias de comportamiento en el precio y poder dar una estimación de en que precio cerrara Bitcoin.

1 Recomendación de canciones utilizando una ANN

El objetivo de esta primera etapa estuvo centrado en la predicción del gusto de un agente sobre una canción, es decir una decision binaria, en donde el agente selecciona con número 1 si una canción fue de agrado o con un número 0 si la canción no fue de agrado para el agente. Lo que el modelo tratara de imitar es un sistema de recomendación ya conocido por todos los servicios de streaming de musica que se encuentran en el mercado, pues estos servicios ofrecen muy discretamente el servicio de recomendar canciones que no han sido escuchadas en su plataforma por el agente, esperando que el agente manifieste su agrado para asi permanecer utilizando la plataforma de streaming de musica.

1.1 Set de Datos

Para el entrenamiento del modelo se utilizara un set de datos que fue compartido publicamente por medio de una competencia de Kaggle. <https://www.kaggle.com/geomack/spotifyclassification> Dicho Dataset fue extraido por medio del API publica de Spotify, por lo que el set de datos contiene columnas que describen a una canción tal y como un servicio de musica caracteriza una canción.

Variables Independiente

- acousticness
- danceability
- duration in ms
- energy
- instrumentalness
- liveness
- loudness
- mode

- speechiness
- tempo
- time signature
- song title
- artist

Variable Dependiente

- target

Como es de esperarse para este entrenamiento se hizo un pre procesamiento de los datos para eliminar aquellas columnas que funcionan como indentificador como el nombre de la canción, el ID interno de spotify, además de configurar el set de datos en el formato indicado para su entrenamiento.

1.2 Experimentación

Para la etapa de experimentación básicamente se entrenaron varios modelos dentro de los cuales lo variante fueron algunos aspectos grandes como arquitecturas o hiper parámetros de entrenamiento. A continuación se describirán las arquitecturas y la modificación de hiper parámetros realizada en cada experimentación así como también la hipótesis que se lograba experimentar.

Arquitectura 1

La primera red neuronal artificial estuvo compuesta de la siguiente forma:

1. Capa densa de 64 neuronas, activada por la función Relu, los pesos inicializados de forma uniforme.
2. Capa densa de 1 neurona, activada por la función Sigmoid.

1.2.1 Experimento 1

Hipótesis La arquitectura de únicamente una capa oculta con 64 neuronas activadas por la función relu proveerán una exactitud superior al 70% tras entrenar durante 50 epochs, con un batch size de 32 observaciones, utilizando el optimizador adam y la función de costo binary crossentropy.

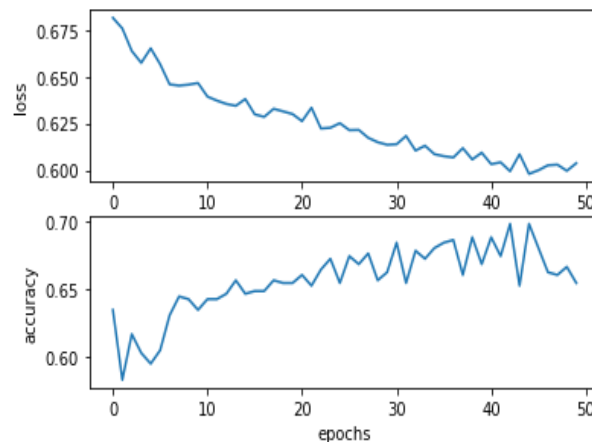


Figure 1: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

1.2.2 Experimento 2

Hipótesis La arquitectura de únicamente una capa oculta con 64 neuronas activadas por la función relu proveerán una exactitud superior al 70

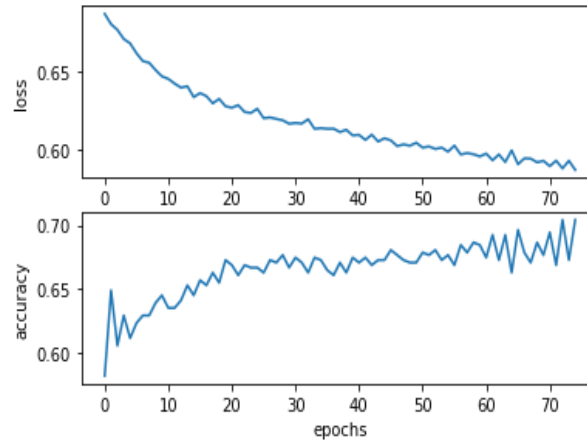


Figure 2: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

1.2.3 Experimento 3

Hipótesis La arquitectura de unicamente una capa oculta con 64 neuronas activadas por la funcion relu proveeran una exactitud superior al 90

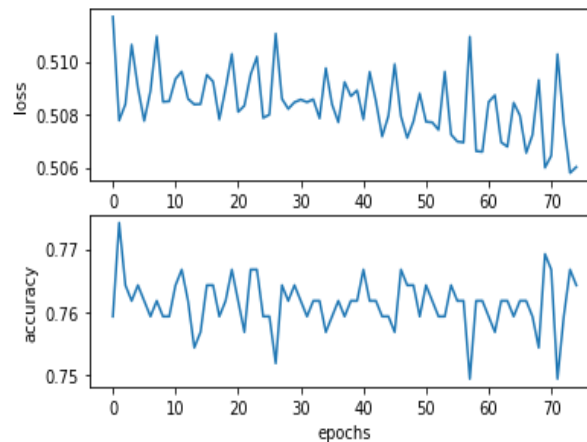


Figure 3: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

Arquitectura 2

1. Capa densa de 32 neuronas, activada por la función Tanh, los pesos inicializados de forma uniforme.
2. Capa densa de 16 neuronas, activada por la función Tanh, los pesos inicializados de forma uniforme.
3. Capa densa de 1 neurona, activada por la función Sigmoid.

1.2.4 Experimento 4

Hipótesis La arquitectura de unicamente 2 capas ocultas, 1 capa con 32 neuronas, la segunda compuesta por 16 neuronas, ambas activadas por la funcion tanh proveeran una exactitud superior al 70% tras entrenar durante 50 epochs, con un batch size de 32 observaciones, utilizando el optimizador adam y la funcion de costo binary crossentropy.

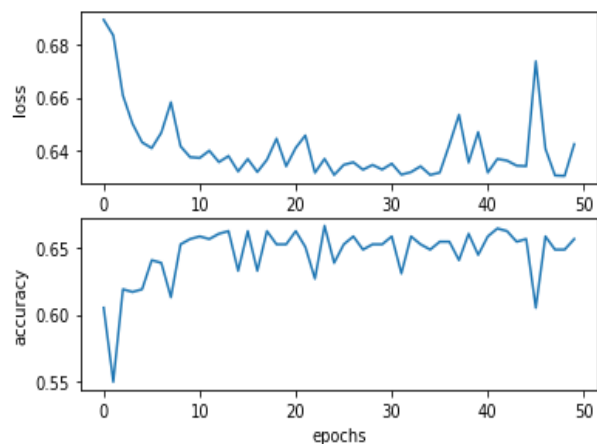


Figure 4: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

1.2.5 Experimento 5

Hipótesis La arquitectura de unicamente 2 capas ocultas, 1 capa con 32 neuronas, la segunda compuesta por 16 neuronas, ambas activadas por la funcion tanh proveeran una exactitud superior al 80% tras entrenar durante 50 epochs, con un batch size de 128 observaciones, utilizando el optimizador adam y la funcion de costo binary crossentropy.

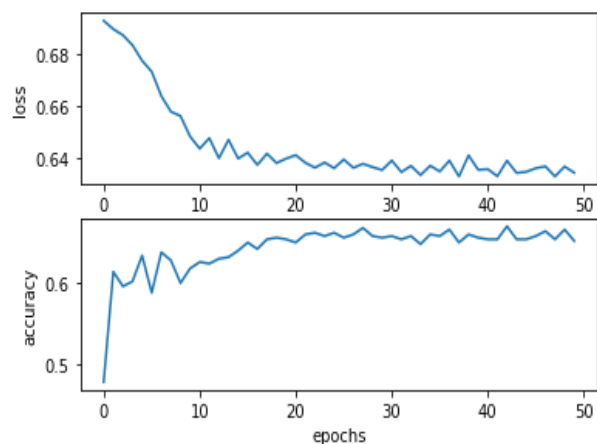


Figure 5: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

Arquitectura 3

1. Capa densa de 64 neuronas, activada por la función Tanh, los pesos inicializados de forma uniforme.
2. Capa densa de 32 neuronas, activada por la función Tanh, los pesos inicializados de forma uniforme.
3. Capa densa de 16 neuronas, activada por la función Tanh, los pesos inicializados de forma uniforme.
4. Capa densa de 1 neurona, activada por la función Sigmoid.

1.2.6 Experimento 6

Hipótesis La arquitectura de unicamente 2 capas ocultas, 1 capa con 64 neuronas, la segunda compuesta por 32 neuronas, ambas activadas por la funcion relu proveeran una exactitud superior al

93 70% tras entrenar durante 50 epochs, con un batch size de 32 observaciones, utilizando el optimizador
94 adam y la funcion de costo binary crossentropy.

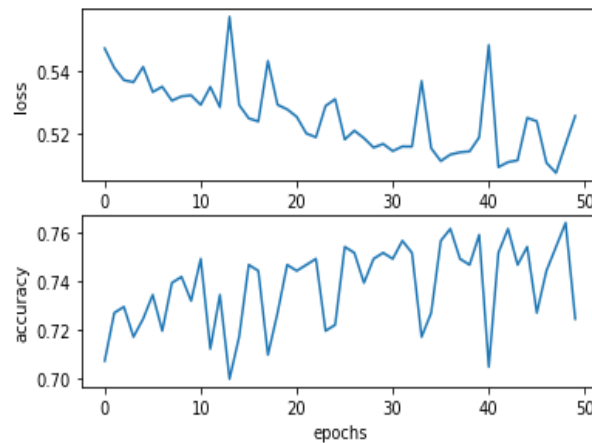


Figure 6: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

95 1.2.7 Experimento 7

96 **Hipótesis** La arquitectura de unicamente 2 capas ocultas, 1 capa con 64 neuronas, la segunda
97 compuesta por 32 neuronas, ambas activadas por la funcion relu proveeran una exactitud superior
98 al 70% tras entrenar durante 50 epochs, con un batch size de 128 observaciones, utilizando el
99 optimizador adam y la funcion de costo binary crossentropy.

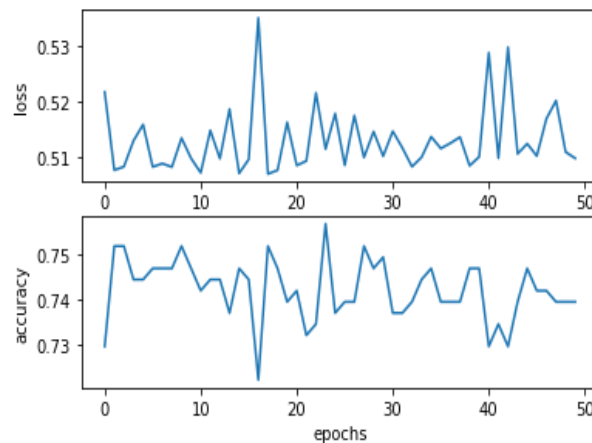


Figure 7: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

100 1.2.8 Experimento 8

101 **Hipótesis** La arquitectura de unicamente 2 capas ocultas, 1 capa con 64 neuronas, la segunda
102 compuesta por 32 neuronas, ambas activadas por la funcion relu proveeran una exactitud superior
103 al 70% tras entrenar durante 50 epochs, con un batch size de 256 observaciones, utilizando el
104 optimizador adam y la funcion de costo binary crossentropy.

105 Arquitectura 4

- 106 1. Capa densa de 128 neuronas, activada por la función Relu, los pesos inicializados de forma
107 uniforme.
- 108 2. Capa de normalización

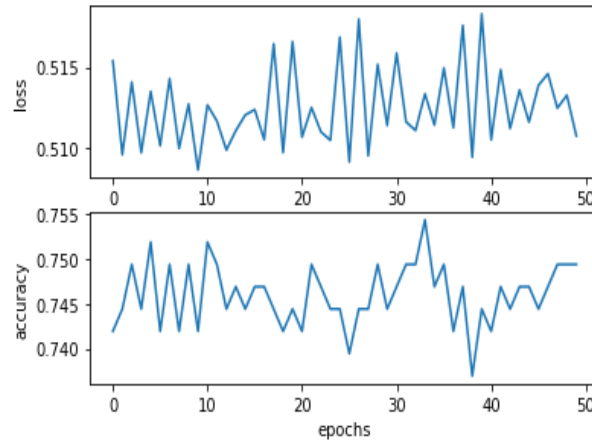


Figure 8: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

- 109 3. Capa densa de 64 neuronas, activada por la función Relu, los pesos inicializados de forma
- 110 uniforme.
- 111 4. Capa de normalización
- 112 5. Capa densa de 32 neuronas, activada por la función Relu, los pesos inicializados de forma
- 113 uniforme.
- 114 6. Capa de castigo al 10%
- 115 7. Capa densa de 1 neurona, activada por la función Sigmoid.

116 1.2.9 Experimento 9

117 **Hipótesis** La arquitectura descrita a continuación tras entrenar durante 175 epochs, con un batch size
 118 de 1000 observaciones, utilizando el optimizador adamax y la función de costo binary crossentropy
 119 proveerán una exactitud superior al 80

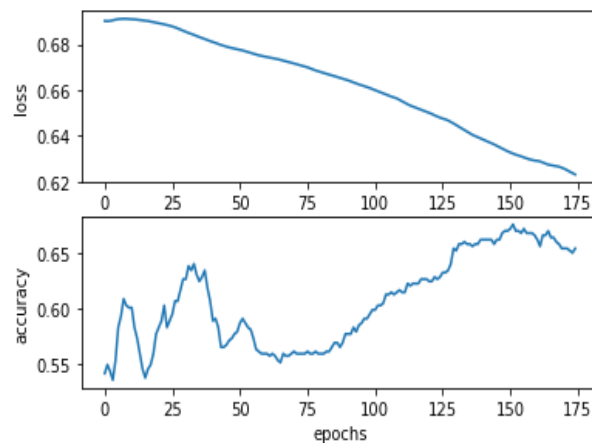


Figure 9: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

120 1.3 Conclusiones

121 Los resultados de las experimentaciones anteriores mostraron que el predecir el gusto de un agente
 122 basado en las características de sus canciones (con like y sin like) es posible por medio de una red
 123 neuronal vanilla, esto hasta cierto nivel ya que agregamos y sacrificamos un poco de nuestra métrica

124 de evaluacion la exactitud con la razon de no sobre ajustar el modelo a la relativa pequena cantidad
125 de canciones disponibles para nuestro agente.

126 Con el ultimo experimento (el mejor) se logro tener una exactitud del 80% sobre el set de validacion
127 mostrado a continuacion. Bajo la arquitectura ganadora descrita a continuacion:

- 128 1. Capa densa de 64 neuronas, activada por la función Tanh, los pesos inicializados de forma
129 uniforme.
- 130 2. Capa densa de 32 neuronas, activada por la función Tanh, los pesos inicializados de forma
131 uniforme.
- 132 3. Capa densa de 16 neuronas, activada por la función Tanh, los pesos inicializados de forma
133 uniforme.
- 134 4. Capa densa de 1 neurona, activada por la función Sigmoid.

135 Se logro disminuir el error a 0.14, teniendo una exactitud del 92% en el entrenamiento y de 80% en
136 el set de validacion, si nos damos la variacion entre estos resultados no es grande, pero tampoco es
137 pequena es decir que al incluir la capa de dropout agregamos ruido que nos ayudo a generalizar un
138 poco mejor, esto se demuestra mucho mas con los resultados de los experimentos anteriores donde
139 ambos resultados no cambiaban. Ahora ya con 80% de exactitud podemos concluir que el modelo
140 puede tomar canciones aleatorias y clasificarlas para ser mostradas o no hacia el agente, en teoria el
141 80% de las canciones mostradas o recomendadas deberian de ser aceptadas o queridas por el agente

142 2 Detección de enfermedades en el ojo con CNN

143 La detección temprana de enfermedades en los ojos es crucial para conseguir una visión duradera a
144 través de los años, es por esto que para esta parte se desarrolló una Red Convolución Neuronal la cual
145 nos permitirá clasificar entre 5 tipos de enfermedades en los ojos, esto basado en imágenes médicas
146 de la retina.

147 2.1 Set de Datos

148 Nuestro set de datos en esta parte fue extraido de la Competencia Retinal Disease Classification de
149 Kaggle. <https://www.kaggle.com/andrewmvd/retinal-disease-classification/code>.

150 **Variables Independiente** Imagenes de 150x150 píxeles a color

151 **Variable Dependiente** El tipo de enfermedad

- 152 1. Diabetic Retinopathy
- 153 2. Age-related Macular
- 154 3. Macular Hole
- 155 4. Retinal Dystrophie
- 156 5. Myopia

157 Para este proyecto se realizó un pre procesmiento de las imagenes, se utilizaron tecnicas de transfor-
158 macion de imagenes para asi lograr un modelo mas generalizado y ademas de que el dataset no es
159 de tan gran tamaño. Para aumentar el tamaño del dataset se tomaron las imagenes y se realizaron
160 sus transformadas con respecto a cambio de perspectiva, se voltearon horizontal y verticalmente en
161 diferentes angulos. Ademas se redujo el tamaño de las imagenes para asi simplificar el rendimiento
162 del modelo.

163 2.2 Experimentación

164 Para la etapa de experimentación basicamente se entrenaron varios modelos dentro de los cuales lo
165 variante fueron algunos aspectos grandes como arquitecturas o hiper parametros de entrenamiento. A
166 continuación se describiran las arquitecturas y la modificación de hiper parametros realizada en cada
167 experimentación así como tambien la hipótesis que se lograba experimentar.

168 Arquitectura 1

- 169 1. Capa convolucional de 16 filtros de 3x3 activadas por la funcion Relu
170 2. Capa de MaxPooling con filtro de 2x2
171 3. Capa densa de 32 neuronas, activada por la función Relu.
172 4. Capa densa de 5 neuronas, activada por la función Sigmoid.

173 2.2.1 Experimento 1

174 **Hipótesis** La arquitectura de unicamente una capa convolucional con 16 filtros de 3x3, 1 capa oculta
175 con 32 neuronas, ambas capas activadas por la funcion relu proveeran una exactitud superior al 70%
176 tras entrenar durante 50 epochs, con un batch size de 256 observaciones, utilizando el optimizador
177 adam y la funcion de costo categorica crossentropy.

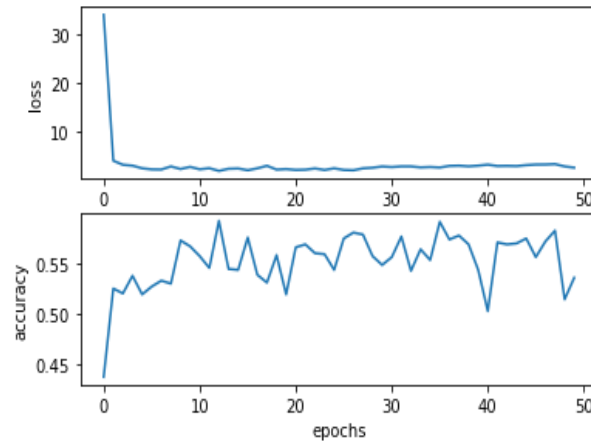


Figure 10: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

178 2.2.2 Experimento 2

179 **Hipótesis** La arquitectura de unicamente una capa convolucional con 16 filtros de 3x3, 1 capa oculta
180 con 32 neuronas, ambas capas activadas por la funcion relu proveeran una exactitud superior al 70%
181 tras entrenar durante 25 epochs, con un batch size de 512 observaciones, utilizando el optimizador
182 adam y la funcion de costo categorica crossentropy.

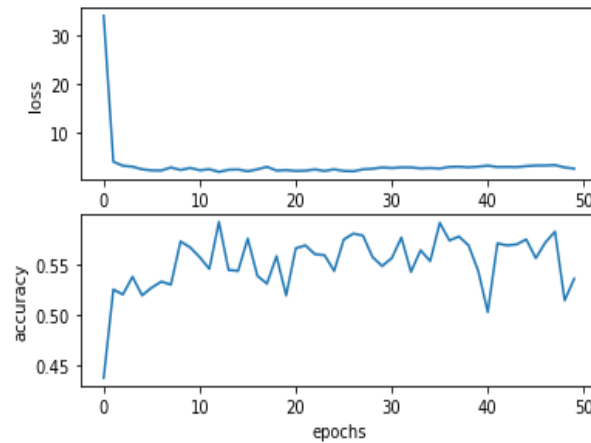


Figure 11: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

183 Arquitectura 2

- 184 1. Capa convolucional de 16 filtros de 3x3 activadas por la funcion Relu
- 185 2. Capa de MaxPooling con filtro de 2x2
- 186 3. Capa de Normalización
- 187 4. Capa densa de 32 neuronas, activada por la función Relu.
- 188 5. Capa de Normalización
- 189 6. Capa densa de 32 neuronas, activada por la función Relu.
- 190 7. Capa de Normalización
- 191 8. Capa de castigo al 40%
- 192 9. Capa densa de 5 neuronas, activada por la función Sigmoid.

193 2.2.3 Experimento 3

194 **Hipótesis** La arquitectura de unicamente una capa convolucional con 16 filtros de 3x3, 1 capa de
 195 maxpooling de 2x2 y 2 capas densas con 32 neuronas, ambas capas activadas por la funcion relu y
 196 normalizadas por una capa de Batch normalization seguido de 1 capa de dropout al 40% proveeran
 197 una exactitud superior al 85% tras entrenar durante 50 epochs, con un batch size de 256 observaciones,
 198 utilizando el optimizador adam y la funcion de costo categorical crossentropy.

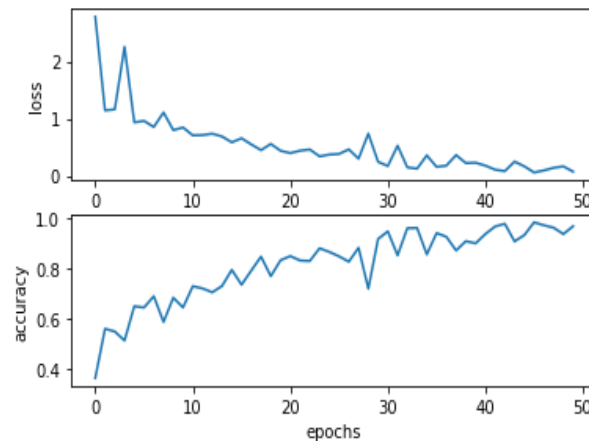


Figure 12: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

199 2.3 Conclusiones

200 Para validar la generalización del modelo se evaluó este en un set de datos antes no vistos, en el cual
 201 alcanzó una exactitud de 96%, es decir que la clasificación de las 5 enfermedades antes mencionadas
 202 es una tarea confiable al utilizar este modelo. La arquitectura responsable de este resultado está
 203 descrita a continuación:

- 204 1. Capa convolucional de 16 filtros de 3x3 activadas por la funcion Relu
- 205 2. Capa de MaxPooling con filtro de 2x2
- 206 3. Capa de Normalización
- 207 4. Capa densa de 32 neuronas, activada por la función Relu.
- 208 5. Capa de Normalización
- 209 6. Capa densa de 32 neuronas, activada por la función Relu.
- 210 7. Capa de Normalización
- 211 8. Capa de castigo al 40%
- 212 9. Capa densa de 5 neuronas, activada por la función Sigmoid.

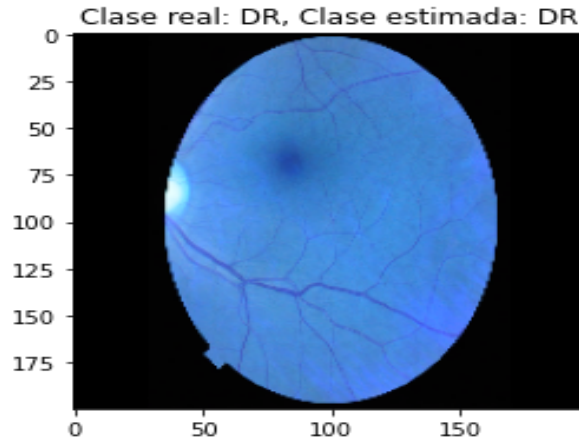


Figure 13: Ejemplo de clasificación sobre retina con problema relacionado a diabetes.

213 3 Predicción del Precio de Bitcoin con RNN

214 La predicción del precio de las criptomonedas es un tema de interés en estos días debido a su relevante
 215 oportunidad de sacar beneficio del trading. Para esta parte utilizaremos una Red Neuronal Recurrente
 216 para así poder estimar el precio del bitcoin en el tiempo. Para lograr esto, alimentaremos a nuestro
 217 modelo con observaciones que están compuestas por el precio de cierre de Bitcoin en los últimos 5
 218 días pasados y la idea es poder predecir el precio de cierre el día siguiente.

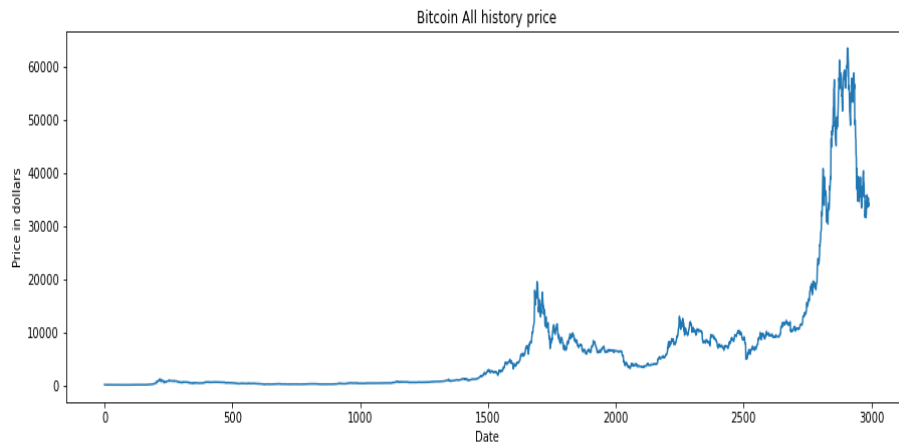


Figure 14: Precio de Bitcoin en su historia

219 3.1 Set de Datos

220 Nuestro set de datos en esta parte fue extraído de la Competencia Cryptocurrency Historical Pricesn
 221 de Kaggle. [https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory?](https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory?select=coin_Bitcoin.csv)
 222 [select=coin_Bitcoin.csv](https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory?select=coin_Bitcoin.csv). Originalmente la fuente de los datos es Yahoo Finance.

223 **Variables Independiente** Precio de cierre del Bitcoin en los últimos 5 días.

224 **Variable Dependiente** El precio de cierre del Bitcoin en el día número 6.

225 Para este proyecto se realizó un pre procesamiento ya que para este modelo se definió que con 5 días
 226 atrás se iba a predecir el 6 de cada rango, por lo que el dataset tiene en sí 5 columnas una por cada
 227 precio de cierre de los días anteriores y la 6 columna pertenecerá al precio de cierre del día número 6.

228 3.2 Experimentación

229 Para la etapa de experimentación basicamente se entrenaron varios modelos dentro de los cuales lo
230 variante fueron algunos aspectos grandes como arquitecturas o hiper parametros de entrenamiento. A
231 continuación se describiran las arquitecturas y la modificación de hiper parametros realizada en cada
232 experimentación así como tambien la hipótesis que se lograba experimentar.

233 Arquitectura 1

- 234 1. Capa recurrente LSTM de 5 unidades
- 235 2. Capa de castigo al 10%
- 236 3. Capa densa de 64 neuronas, activada por la función Relu.
- 237 4. Capa de castigo al 10%
- 238 5. Capa densa de 16 neuronas, activada por la función Relu.
- 239 6. Capa de castigo al 10%
- 240 7. Capa densa de 1 neuronas, activada por la función lineal.

241 3.2.1 Experimento 1

242 **Hipótesis** La arquitectura de una capa LSTM de 5 neuronas, 2 capas densas con activacion relu
243 seguidas de capas de castigo dropout al 10% proveeran almenos una disminucion en el RMSE por
244 debajo de 0.5 tras entrenar por 100 epochs, con un batch size de 128 observaciones, utilizando el
245 optimizador adam y MSE como funcion de error.

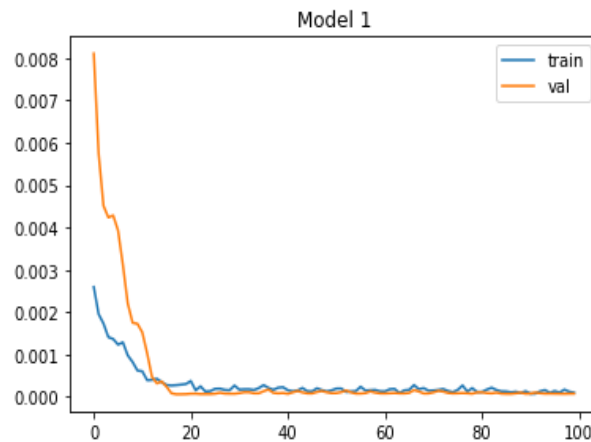


Figure 15: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

246 3.2.2 Experimento 2

247 **Hipótesis** La arquitectura de una capa LSTM de 5 neuronas, 2 capas densas con activacion relu
248 seguidas de capas de castigo dropout al 10% proveeran almenos una disminucion en el RMSE por
249 debajo de 0.001 tras entrenar por 50 epochs, con un batch size de 256 observaciones, utilizando el
250 optimizador adam y MSE como funcion de error.

251 Arquitectura 2

- 252 1. Capa recurrente LSTM de 5 unidades
- 253 2. Capa densa de 32 neuronas, activada por la función Relu.
- 254 3. Capa densa de 16 neuronas, activada por la función Relu.
- 255 4. Capa de castigo al 10%
- 256 5. Capa densa de 1 neuronas, activada por la función lineal.

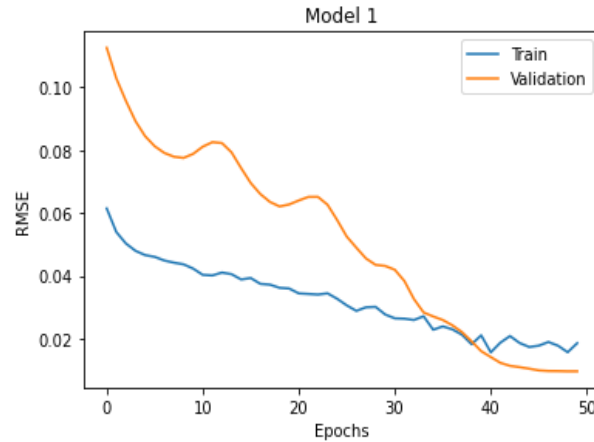


Figure 16: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

3.2.3 Experimento 3

Hipótesis La arquitectura de una capa LSTM de 5 neuronas, 2 capas densas con activacion relu seguidas de 1 unica capa de castigo dropout al 20% proveeran almenos una disminucion en el RMSE por debajo de 0.05 tras entrenar por 50 epochs, con un batch size de 64 observaciones, utilizando el optimizador adam y MSE como funcion de error.

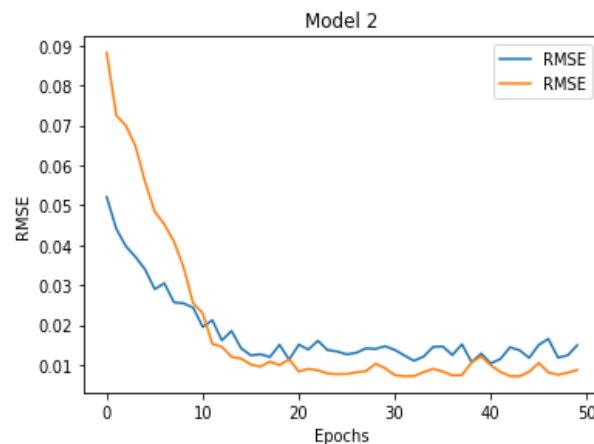


Figure 17: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

3.2.4 Experimento 4

Hipótesis La arquitectura de una capa LSTM de 5 neuronas, 2 capas densas con activacion relu seguidas de 1 unica capa de castigo dropout al 20% proveeran almenos una disminucion en el RMSE por debajo de 0.005 tras entrenar por 100 epochs, con un batch size de 64 observaciones, utilizando el optimizador adam y MSE como funcion de error.

3.3 Conclusiones

El precio del bitcoin es algo que depende de muchos factores externos, por lo que tratarlo de predecir como una serie de tiempo ajena a factores externos talvez no es una buena idea, pero a pesar de que su comportamiento como en la primera grafica muestra un comienzo algo pronunciado alista, en los ultimos tiempos bitcoin ha tenido un comportamiento algo desordenado en el que no se parece en nada a su historia, en cualquier día puede dar un moviemiento muy brusco y lo que realmente

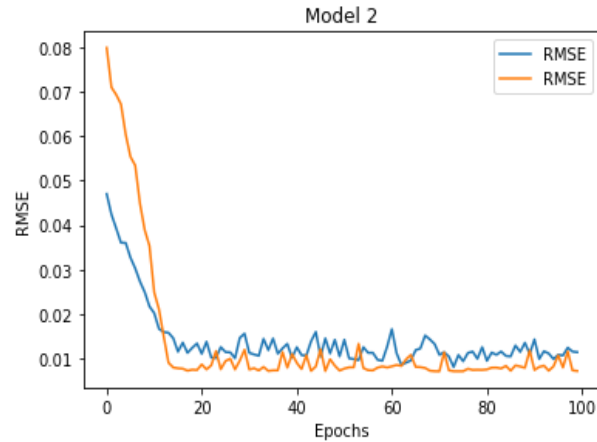


Figure 18: Error y exactitud en set de validación durante entrenamiento en arquitectura 1

273 interesaria ver es en que posicion queda su precio al final del dia. Creo que lo mas importante para
 274 inversores es poder tener una idea de en donde terminara el precio al final del dia, cosa que por su
 275 comportamiento se ve algo dificil pero lo que si nos da es la certeza de si sera alcista o no el precio
 276 final lo que al final es una buena primera senal en si invertir o no.

277 A continuacion se muestra una grafica del modelo con data de validacion es decir que el modelo
 278 nunca habia visto y es interesante como al principio bajo un comportamiento algo normal se comporta
 279 bien el modelo, pero cuando se llega a los puntos donde es algo desordenado no logra llegar a predecir
 280 el precio exacto, pero si el comportamiento.

281 Este resultado se obtuvo de entrenar el modelo2 al entrenar al modelo con ventanas de 5 dias
 282 hacia atras y tratando de predecir unicamente un dia adelante, el error se redujo hasta llegar a una
 283 estacionalidad en donde ya no disminuia el error solamente oscilaba en un minimo de RMSE de
 284 0.0011.

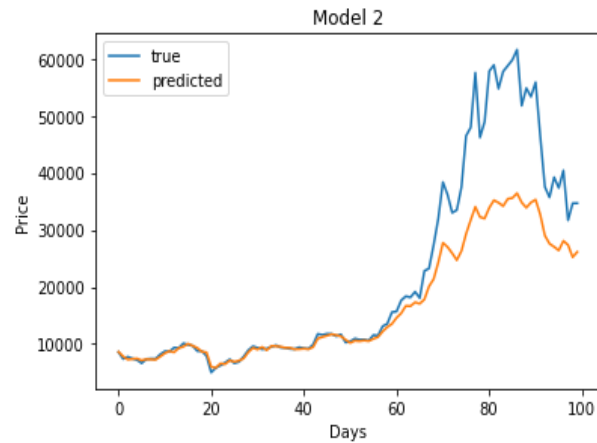


Figure 19: Error y exactitud en set de validación durante entrenamiento en arquitectura 1