
Recurrent Neural Network - Sentiment Analysis of Financial News

Edgar Emilio González y González
emilio.gonzalez@galileo.edu.gt

Abstract

1 El mercado bursátil se mueve constantemente y en ello tiene un gran impacto las
2 noticias, que llevan a invertir o no en las acciones de las empresas, que llevan a los
3 inversionistas a comprar o vender acciones, en este análisis se busca determinar
4 como están las noticias y determinar si es un buen momento para poder invertir o no
5 en compras de acciones. El análisis de sentimiento tiene como objetivo determinar
6 la como se sienten los inversionistas respecto a las noticias financieras, a partir
7 de una fuente de datos y buscando una buena toma de decisiones. Este trabajo se
8 centra en la aplicación del análisis de sentimientos en las noticias financieras. Se
9 evalúan dos técnicas alternativas, ambas con Keras pero con distintos parámetros
10 que nos ayudarán a determinar cual tiene una mejor predicción sobre los datos
11 entregados.

12 1 Dataset

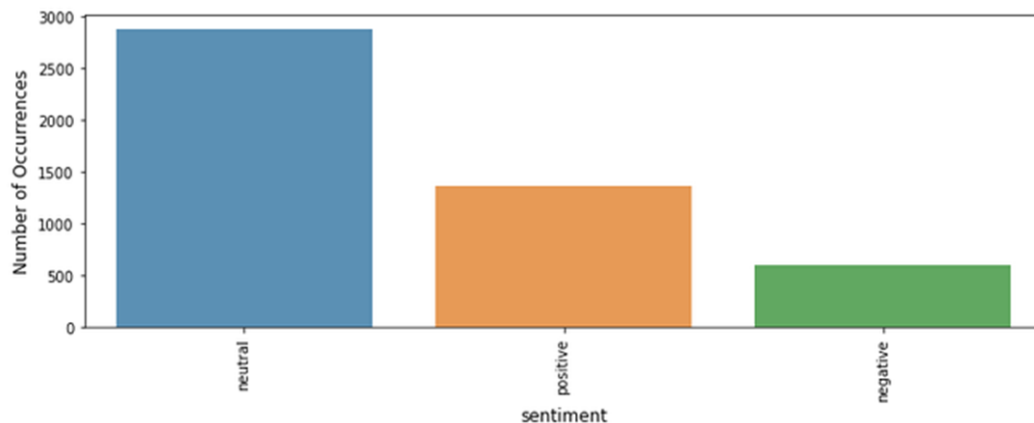
13 Se utiliza un dataset público(facilitado en una competencia de Kaggle) que tiene 3 categorías distintas
14 en los comentarios:

15 * Positivo

16 * Negativo

17 * Neutral

18 En el avance del proyecto veremos cambios necesarios en la data para poder incrementar la capacidad
19 de segmentar los comentarios.



20

2 Models

El modelo principal utilizado es de Keras con optimizador Adam, adicional a este e utilizaron 3 distintos modelos de sklearn para ver cual de ellos performa mejor con nuestro problema y poder comparar entre sklearn y keras cual tiene mejor resultados y mejor tiempo de procesamiento, siempre en busqueda que nuestro MLP sea más eficiente y con mejores predicciones.

2.1 Algoritmo ADAM:

es una extensión del descenso de gradiente y un sucesor natural de técnicas como AdaGrad y RMSProp que adapta automáticamente una tasa de aprendizaje para cada variable de entrada para la función objetivo y suaviza aún más el proceso de búsqueda mediante el uso de una media móvil exponencialmente decreciente del gradiente para realizar actualizaciones a las variables.

Adam está diseñado para acelerar el proceso de optimización, p. Ej. disminuir el número de evaluaciones de funciones requeridas para alcanzar los óptimos, o para mejorar la capacidad del algoritmo de optimización, p. resultar en un mejor resultado final.

Esto se logra calculando un tamaño de paso para cada parámetro de entrada que se optimiza. Es importante destacar que cada tamaño de paso se adapta automáticamente al rendimiento del proceso de búsqueda en función de los gradientes (derivadas parciales) encontrados para cada variable.

Algorithm: Generalized Adam

S0. Initialize $m_0 = 0$ and x_1

For $t = 1, \dots, T$, **do**

S1. $m_t = \beta_{1,t}m_{t-1} + (1 - \beta_{1,t})g_t$

S2. $\hat{v}_t = h_t(g_1, g_2, \dots, g_t)$

S3. $x_{t+1} = x_t - \frac{\alpha_t m_t}{\sqrt{\hat{v}_t}}$

End

37

2.2 Algoritmo SGD:

es un método iterativo para optimizar una función objetivo con propiedades de suavidad adecuadas (por ejemplo, diferenciable o subdiferenciable). Puede considerarse como una aproximación estocástica de la optimización del descenso del gradiente, ya que reemplaza el gradiente real (calculado a partir del conjunto de datos completo) por una estimación del mismo (calculado a partir de un subconjunto de datos seleccionado al azar). Especialmente en problemas de optimización de alta dimensión, esto reduce la carga computacional, logrando iteraciones más rápidas en el comercio para una tasa de convergencia más baja. [1]

Si bien la idea básica detrás de la aproximación estocástica se remonta al algoritmo de Robbins-Monro de la década de 1950, el descenso de gradiente estocástico se ha convertido en un método de optimización importante en el aprendizaje automático.

Algorithm 1 Pseudo code of Stochastic Gradient Descent algorithm

```
1  Enter initial values ( $\epsilon_k, \theta$ )
2  while
3  Sampling  $\{x^{(1)}, \dots, x^{(m)}\}$  from Dataset by using  $\{y^{(1)}, \dots, y^{(m)}\}$  Labels
4   $\hat{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(x^{(i)}, \theta), y^{(i)})$ 
5   $\theta \leftarrow \theta - \epsilon_k \hat{g}$ 
6  End while
```

49

2.3 Algoritmo Nadam:

es una extensión del algoritmo de Adam que incorpora el impulso de Nesterov y puede resultar en un mejor rendimiento del algoritmo de optimización.

El algoritmo fue descrito en el artículo de 2016 por Timothy Dozat titulado “Incorporando el impulso de Nesterov en Adam”. Aunque una versión del documento se redactó en 2015 como un informe del proyecto de Stanford con el mismo nombre.

Momentum agrega un promedio móvil en declive exponencial (primer momento) del gradiente al algoritmo de descenso del gradiente. Esto tiene el impacto de suavizar las funciones objetivas ruidosas y mejorar la convergencia.

Adam es una extensión del descenso de gradiente que agrega un primer y segundo momento del gradiente y adapta automáticamente una tasa de aprendizaje para cada parámetro que se optimiza. NAG es una extensión del impulso en el que la actualización se realiza utilizando el gradiente de la actualización proyectada del parámetro en lugar del valor de la variable actual real. Esto tiene el efecto de ralentizar la búsqueda cuando se encuentra el óptimo en lugar de sobrepasar, en algunas situaciones.

Algorithm 2 Nesterov-accelerated Adaptive Moment Estimation (Nadam)

Require: $\alpha_0, \dots, \alpha_T; \mu_0, \dots, \mu_T; \nu; \epsilon$: Hyperparameters

$\mathbf{m}_0; \mathbf{n}_0 \leftarrow 0$ (first/second moment vectors)

while θ_t not converged **do**

$\mathbf{g}_t \leftarrow \nabla_{\theta_{t-1}} f_t(\theta_{t-1})$

$\mathbf{m}_t \leftarrow \mu_t \mathbf{m}_{t-1} + (1 - \mu_t) \mathbf{g}_t$

$\mathbf{n}_t \leftarrow \nu \mathbf{n}_{t-1} + (1 - \nu) \mathbf{g}_t^2$

$\hat{\mathbf{m}} \leftarrow (\mu_{t+1} \mathbf{m}_t / (1 - \prod_{i=1}^{t+1} \mu_i)) + ((1 - \mu_t) \mathbf{g}_t / (1 - \prod_{i=1}^t \mu_i))$

$\hat{\mathbf{n}} \leftarrow \nu \mathbf{n}_t / (1 - \nu^t)$

$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha_t}{\sqrt{\hat{\mathbf{n}}_t + \epsilon}} \hat{\mathbf{m}}_t$

end while

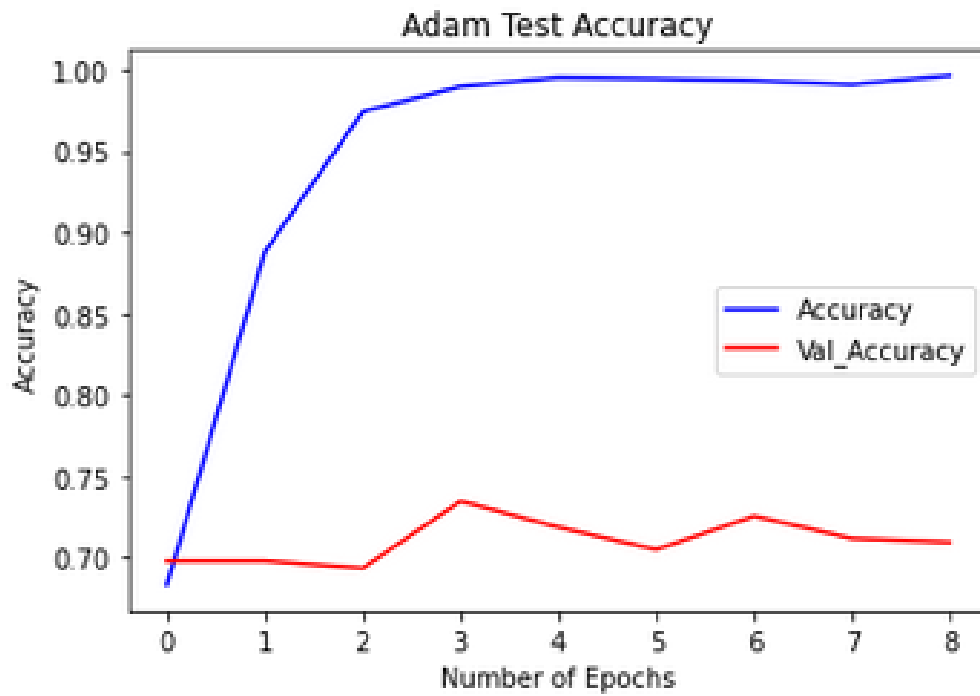
return θ_t

65

3 Resultados

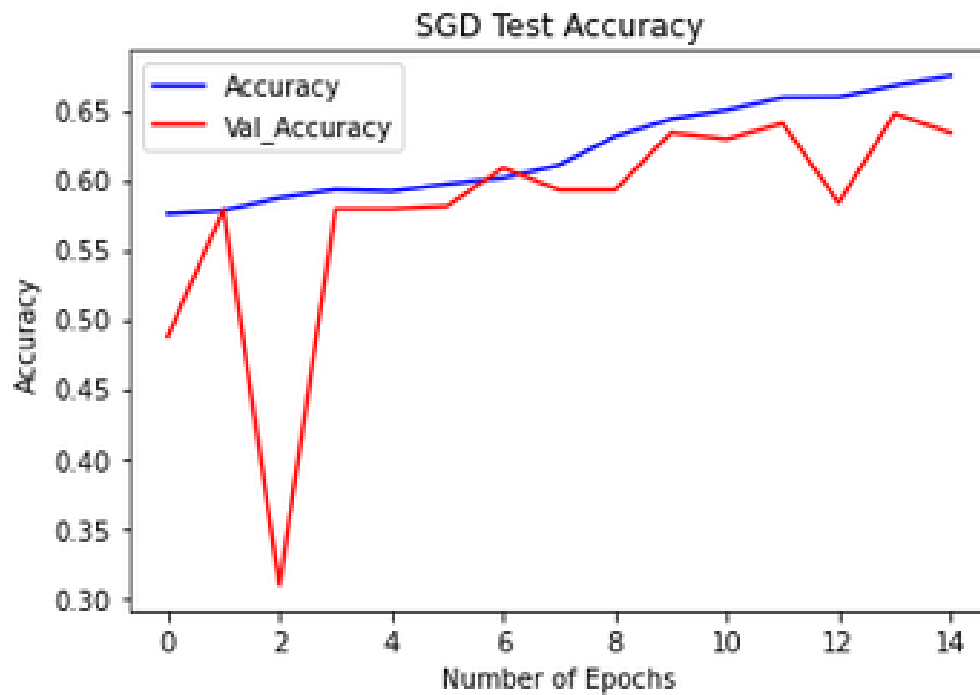
En las dos herramientas utilizadas, tuvimos mejores resultados con Keras que con tensorflow, a continuación el resumen del accuracy, tanto de las 3 pruebas distintas en Tensorflow como la única realizada con Keras:

*TF - ADAM:



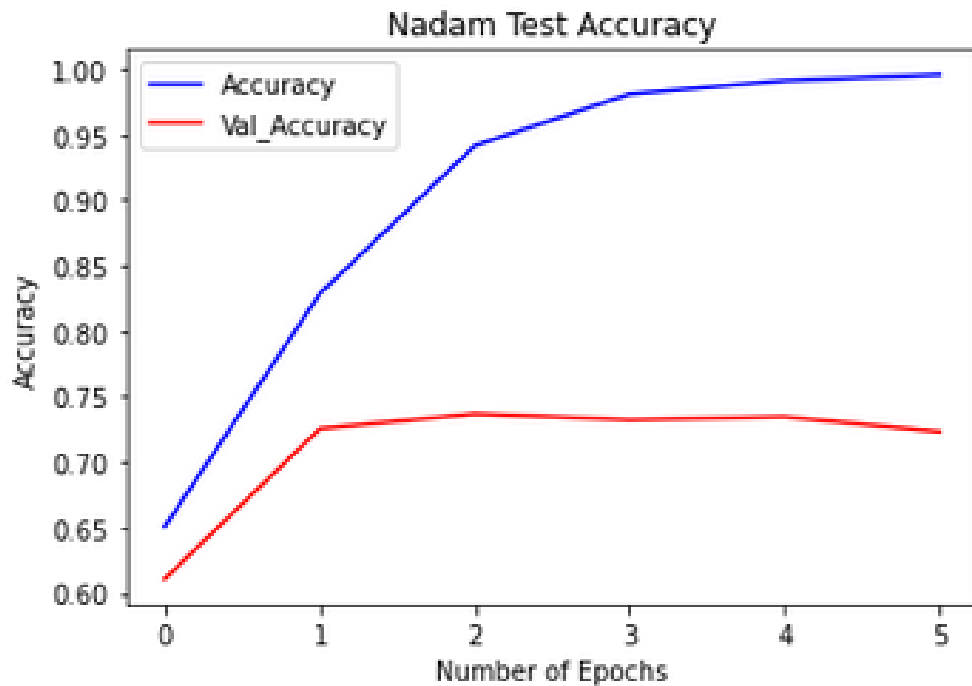
71

72 *TF - SGD:



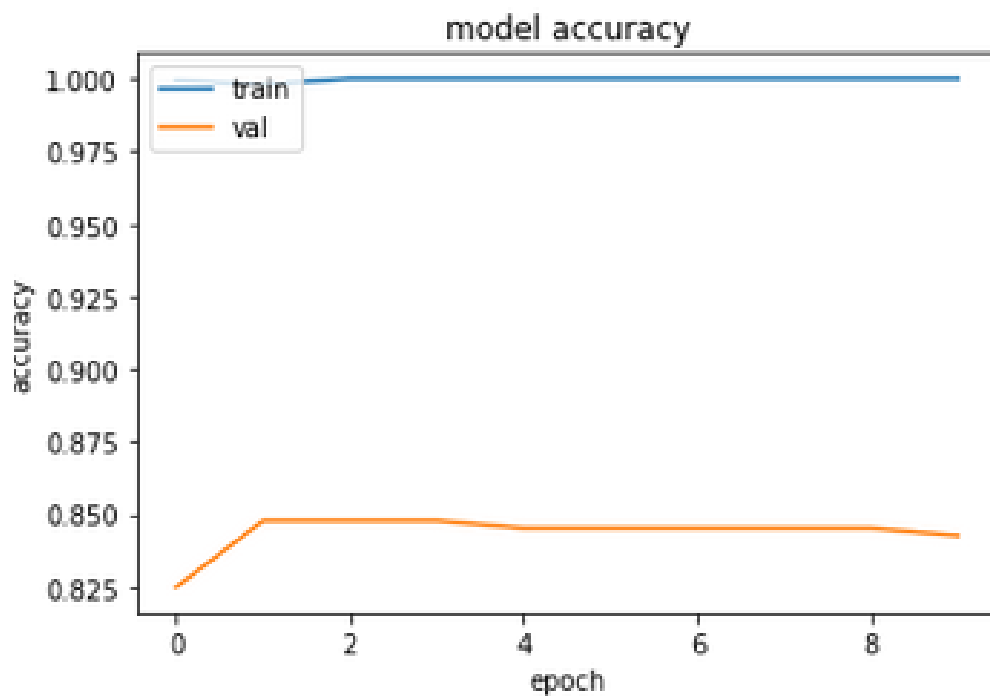
73

74 * TF - NADAM:



75

76 * KERAS:



77

78 4 Conclusiones

79 Se tuvieron buenos resultados utilizando el framework de keras, pero un punto importante en esto fue
 80 eliminar los neutros en nuestro dataset, esto permitió que mejorara y evolucionara mejor el algoritmo,
 81 tanto en velocidad como en capacidad de distinguir los positivos y negativos.

82 **5 Referencias**

- 83 VARMEDJA, Dejan, et al. Credit card fraud detection-machine learning methods. En 2019 18th
84 International Symposium INFOTEH-JAHORINA (INFOTEH). IEEE, 2019. p. 1-5.
- 85 PILLAI, Thulasyammal Ramiah, et al. Credit card fraud detection using deep learning technique.
86 En 2018 Fourth International Conference on Advances in Computing, Communication Automation
87 (ICACCA). IEEE, 2018. p. 1-6.
- 88 ARUN, Gurumurthy Krishnamurthy; VENKATACHALAPATHY, Kaliyappan. Intelligent feature
89 selection with social spider optimization based artificial neural network model for credit card fraud
90 detection. IIOABJ, 2020, vol. 11, no 2, p. 85-91.