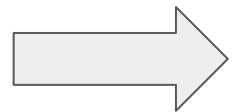




SCAN ME

Introducción y motivación

En la actualidad, muchos sitios de comercio en línea dependen de trabajadores manuales para etiquetar y clasificar sus productos, lo que puede resultar lento y costoso. Una aplicación de clasificación de imágenes automatizada podría ayudar a agilizar este proceso y a mejorar la precisión de las clasificaciones, lo que podría beneficiar tanto a los sitios de comercio en línea como a sus clientes.



ZAPATILLA

Como solución se han implementado los siguientes métodos:

1. SVM (Support Vector Machine) usando OVO y OVA
2. MLP (Multilayer Perceptron) de Scikit-learn
3. MLP (Multilayer Perceptron) de Tensorflow

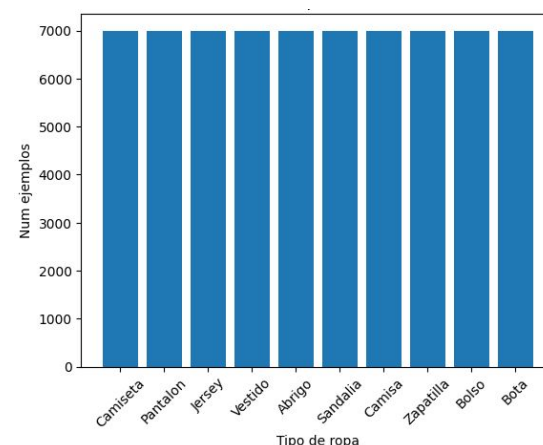
Conjunto de datos

Para este proyecto, hemos elegido utilizar el dataset MNIST fashion debido que cuenta con 10 clases diferentes, lo que lo hace adecuado para un problema de clasificación de imágenes. Además, las imágenes tienen un tamaño pequeño de 28x28, lo que significa que son fáciles de procesar y requieren menos cómputo que imágenes más grandes.

Es importante señalar que es posible utilizar otros datasets para este proyecto aunque es importante tener en cuenta que el cambio de dataset podría requerir más cómputo y afectaría al tiempo de entrenamiento y a la precisión del modelo.

Número de ejemplos: 70000 (repartidos entre 10 clases)

- Train: 60000
- Test: 10000



Marco experimental

Para este proyecto, hemos utilizado dos modelos de aprendizaje automático diferentes: máquinas de vectores de soporte (SVM) y redes neuronales multinivel (MLP). Además, hemos utilizado un modelo MLP implementado en TensorFlow, ya que al utilizar la GPU este modelo es capaz de entrenarse más rápidamente.

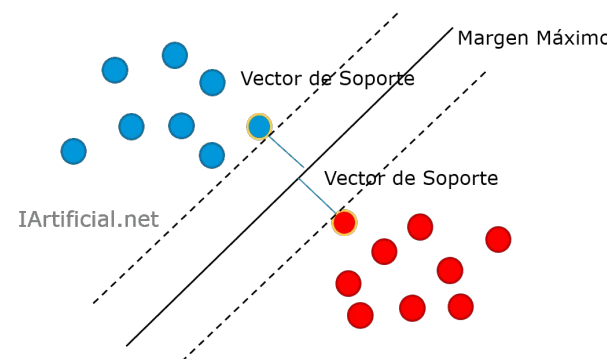
MODELOS:

SVM:

- OVO (One vs One)
- OVA (One vs All)

Optimización SVM:

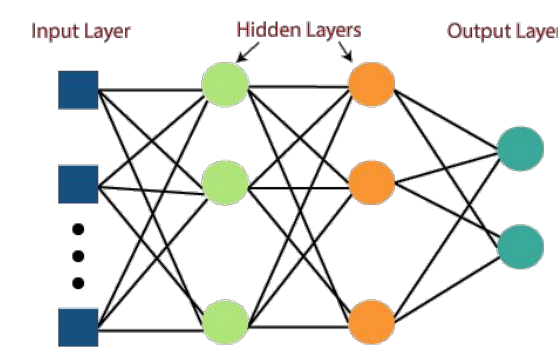
- C
- Gamma



MLP:

Optimización MLP:

- Tamaño capas ocultas
- Ratio de aprendizaje
- N° iteraciones
- Optimizador
- Mini-batch

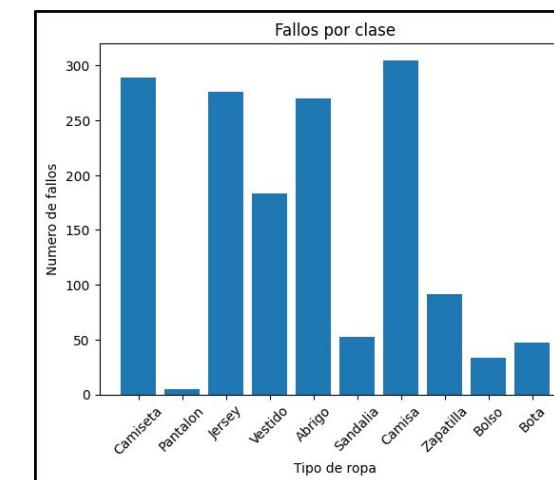


Estudio experimental

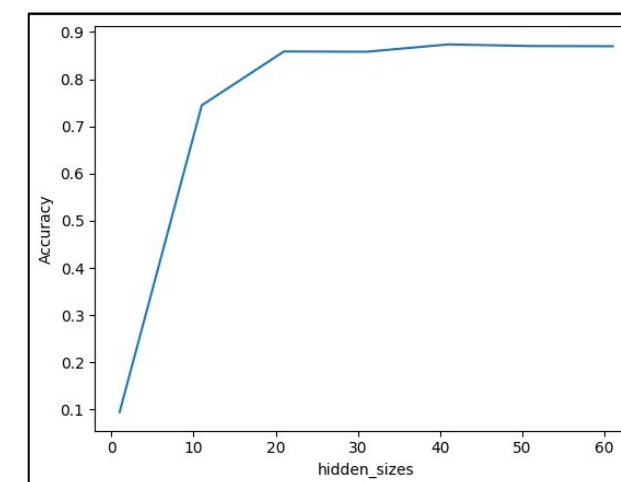
RESULTADOS:

RESULTADOS POR MODELO		Acc	Tiempo
SVM	OVO --> Gamma 'scale' C = 1.0	0.889	6.12 min
	OVA --> Gamma 'scale' C = 1.0	0.889	5.8 min
MLP	Capas ocultas --> 121 Alpha --> 0.01 Iteraciones --> 250 Optimizador --> Adam Modo --> por Lotes	0.864	5.25min
	Capas ocultas --> (100,30,20) Alpha --> 0.01 Iteraciones --> 250 Optimizador --> Adam Modo --> por Lotes	0.885	7.32 min
	Capas ocultas --> (100,60,20) Alpha --> 0.01 Iteraciones --> 250 Optimizador --> Adam Modo --> mini-batch de 1000	0.872	8.56 min

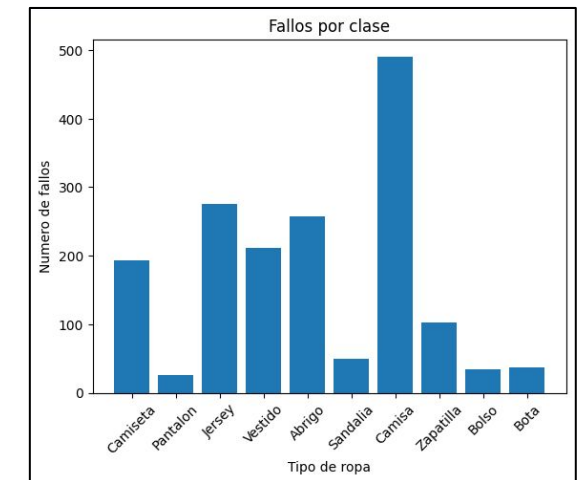
Fallos por clase SVM:



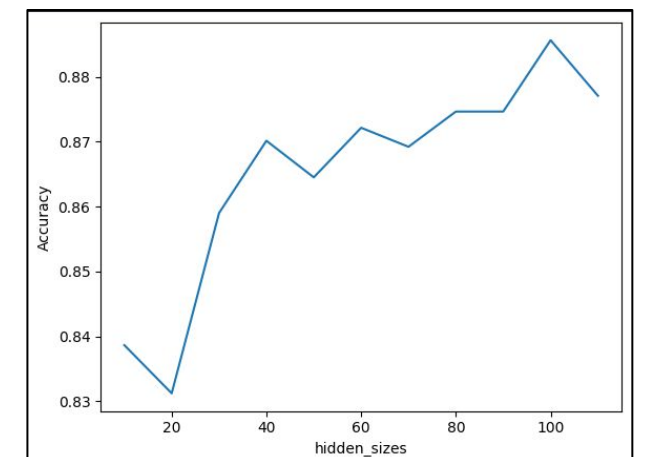
Accuracy respecto a una única capa oculta MLP:



Fallos por clase MLP:



Accuracy respecto a variación de la primera capa oculta MLP (X,30,20):



Como modelo final podríamos elegir tanto MLP o la SVM ya que ambas dan resultados muy parecidos. Con ninguno logramos pasar del 90% de precisión ya que debido a la baja resolución de las imágenes y a que hay 4 clases bastante parecidas entre sí (camisa, camiseta, abrigo, jersey) fallamos bastantes (los gráficos mostrados arriba). El SVM falla las 4 clases más o menos por igual, pero el MLP falla sobre todo en la clase camisa. Por lo tanto, puede que el MLP sea más interesante ya que con centrarse en las camisas se reducen en gran medida los errores de clasificación. En general, ambos son modelos que generalizan bastante bien y que podrían usarse en un entorno real.

Conclusiones y líneas futuras

En conclusión, hemos desarrollado una aplicación de clasificación de imágenes utilizando dos modelos de aprendizaje automático diferentes y un modelo MLP implementado en TensorFlow. Hemos evaluado estos modelos utilizando el dataset MNIST fashion y hemos comparado sus resultados.

En un futuro, podría ser interesante explorar la posibilidad de utilizar videos en lugar de imágenes y modelos más complejos de convolución para la clasificación de imágenes. Esto podría permitir una mayor precisión en la clasificación y podría ser útil en aplicaciones que involucren objetos en movimiento, como el reconocimiento de ropa en una tienda.