Certamen 1 Machine Learning

Pablo Sebastián Jerez Agurto - Gabriel Ignacio Briceño Novoa
Universidad de Concepción
pablojerez2016@udec.cl - gbriceno2016@udec.cl
Concepción

1. Introducción

El presente artículo muestra la réplica de modelos de machine learning presentados en el manuscrito *A deep learning ensemble model for wildfire susceptibility mapping* preparados por Alexandra Bjanes, con el fin de estudiar, evaluar y comparar resultados de la susceptibilidad de incendios en la Octava región del BioBio. Los datos han sido proporcionados por la autora del manuscrito y los modelos a implementar y optimizar son el CNN-1 y CNN-2. Finalmente se realizará una comparación con el modelo presentado en el artículo *A new approach of deep neural computing for spatial prediction of wildfire danger at tropical climate áreas*, utilizando los mismos datos y metodología.

2. Metodología

El data set proporcionado por Alejandra Bjanes está compuesto por una capa de clasificación incendio/no incendio y otras 14 capas correspondiente al terreno, relacionadas con variables meteorológicas, topográficas, vegetación y variables antropogénicas. Éstas serán extraídas en instancias de 7x7 píxeles. Para un correcto análisis, se revisa y limpia el dataset de datos que puedan entorpecer el aprendizaje. Se eliminan instancias que posean capas nulas y luego se balancea el dataset de acuerdo con la variable de interés (incendio/no incendio). De este modo se analizan 9728 instancias compuestas por 21 capas de 7x7 píxeles.

2.1. Set de entrenamiento y set de testeo

Se crean el grupo de entrenamiento y el de prueba (80 % y 20 % respectivamente). Ambos conjuntos se dividen en 5 pliegues para utilizarlos en 5-Fold cross validation en los posteriores modelos de entrenamiento y prueba.

2.2. Definición de Hipermodelos e hiperparámetros

Se implementan los hipermodelos para las tres arquitecturas a comparar: CNN-1, CNN-2 y MLP. Se utiliza Keras Turner para facilitar un espacio de búsqueda y encontrar los mejores valores de hiperparámetros. Se definen los algoritmos de optimización bayesiana, hiperbanda y búsqueda aleatoria. Los hiperparámetros a prueba son los siguientes:

Modelo	Optimizador	Learning rate	Batch size
CNN1	{Adam, SGD}	{0, 0.1, 0.01, 0.001, 0.0001}	{16, 32, 64}
CNN2	{Adam, SGD}	{0, 0.1, 0.01, 0.001, 0.0001}	{16, 32, 64}
MLP	{Adam, SGD}	{0, 0.1, 0.01, 0.001, 0.0001}	{16, 32, 64}

Cuadro 1: Hiperparámetros explorados para cada Arquitectura.

Modelo	Dropout	L1	L2
CNN1	{0, 0.2, 0.5}	{0, 0.1, 0.01, 0.001}	{0, 0.1, 0.01, 0.001}
CNN2	{0, 0.2, 0.5}	{0, 0.1, 0.01, 0.001}	{0, 0.1, 0.01, 0.001}
MLP	{ - }	{ - }	{ - }

Cuadro 2: Regularizadores explorados para cada Arquitectura.

2.3. Optimización de Hiperparámetros con validación cruzada

Para cada batch size explorado(16, 32, 64) se probaron múltiples combinaciones de hiperparámetros para cada una de las arquitecturas. Luego de encontrar la mejor configuración de hiperparámetros para cada batch size, se reentrenó usando 5-Fold cross validation para obtener el mejor batch size en base al F1-Score por cada una de las arquitecturas. Finalmente se itera usando 5-Fold cross validation 3 veces en el mismo modelo, es decir, se iteró 15 veces hasta encontrar el modelo que encuentre la mejor semilla presentando el mejor F1-Score.

2.4. Matriz de confusión y F1-Score

Para evaluar los resultados de los modelos, se utilizan indicadores como la matriz de confusión y F1-Score. La matriz de confusión permite ver que tan bien puede clasificar el modelo los valores procesados, y el F1-Score permite evaluarla precisión y exhaustividad del modelo. Las filas representan el 'ytest' y las columnas el 'ypred'.

3. Experimentación y Resultados

3.1. Métricas en set de entrenamiento

Los hiperparámtros obtenidos en el espacio de busqueda de los hipermodelos para cada una de las arquitecturas se aprecian en el cuadro 3:

Modelo	Optimizador	Learning rate	Dropout	L1	L2	Batch size
CNN1	Adam	0.001	0.5	0.001	0.01	32
CNN2	Adam	0.01	0.5	0.001	0.001	32
MLP	Adam	0.01	-	-	-	16

Cuadro 3: Hyperparámetros escogidos para cada Arquitectura.

Los resutlados obtenidos para los datos de entrenamiento en cada uno de los modelos escogidos se aprecian en el Anexo.

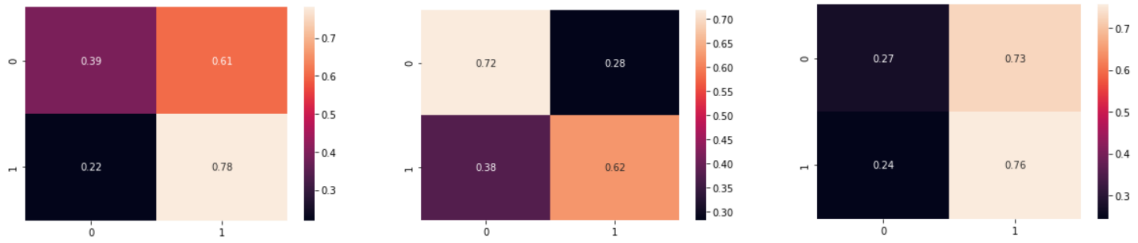
Se logra evidenciar que el modelo CNN-2 presenta mejores resultados a diferencia del CNN-1 y MLP para los datos de entrenamiento.

Modelo	Accuracy train	Accuracy val
CNN1	mean: 0.5526 std: 0.0191	mean: 0.5905 std: 0.0120
CNN2	mean: 0.6832 std: 0.0057	mean: 0.6651 std: 0.0123
MLP	mean: 0.5426 std: 0.0117	mean: 0.5246 std: 0.0356

Cuadro 4: Resultados de entrenamiento de modelos CNN-1, CNN-2, MLP.

3.2. Métricas en set de testeo

Los resultados obtenidos evidencian que el mejor modelo es el CNN-2 configurado con optimizador Adam, learning rate 0.01, dropout de 0.5, con regularizador l1 y l2 0.001 presentando un f1 score de 0.6551.



(a) $CNN-1$ (b) $CNN-2$ (c) MLP

Figura 1: Matrices de confusión data test CNN-1, CNN-2, MLP

Modelo	TP	TN	FP	FN	f1 score	Recall	Precision
CNN1	0.78	0.39	0.61	0.22	0.6540	0.7802	0.5629
CNN2	0.62	0.72	0.28	0.38	0.6551	0.6242	0.6893
MLP	0.76	0.27	0.73	0.24	0.6092	0.7556	0.5104

Cuadro 5: Resultados de testeo de modelos CNN-1, CNN-2, MLP.

4. Conclusión

Las arquitecturas CNN-1 y CNN-2 presentaron mejores rendimientos a diferencia de la arquitectura MLP, permitiendo mapear con mayor efectividad un mapa de susceptibilidad de riesgo de incendio. El modelo seleccionado es el CNN-2, con valores de accuracy y f1 score superiores a CNN-1, presentados en el Cuadro 4 y 5 de las secciones 3.1 y 3.2 respectivamente.

Los datos predichos por el modelo CNN-2 tienen un amplio rango de suceptibilidad. El rango de mayor incertidumbre de las predicciones se encuentra entre 0.3 y 0.6, dado que el porcentaje de falsos negativos que se encuentran en el rango de predicciones menor a 0.3 es cercano a 0, lo mismo ocurre con el porcentaje de falsos positivos que se encuentran en el rango de predicciones mayor a 0.6.

El modelo CNN2 seleccionado, si bien presenta porcentajes de verdaderos positivos y verdaderos negativos aceptables (0.62 y 0.72), debería considerarse disminuir el threshold a 0.3 y así reducir los falsos negativos a 0.08, y en efecto, aumentar el porcentaje de falsos positivos a 0.73, ya que es preferible monitorear y controlar más áreas previniendo incendios en vez de reparar los daños causados por un siniestro no contemplado, que en este caso, y como se mencionó en el parrafo anterior, consideraría a las predicciones dentro del rango 0.3 a 0.6, que es donde se encuentran la mayor cantidad de falsos positivos y falsos negativos, y en definitiva en donde el modelo no logra separar de manera consistente los datos. Cabe destacar que el f1 score con un threshold de 0.3 aumenta de un 0.654 a un 0.695.

Para los otros modelos, es importante recalcar que sus predicciones tienen muy poca variabilidad entre ellas. Esto puede deberse a la naturaleza de los datos proporcionados, a la extracción de los datos o al tamaño de patchsize. Sería intereante encontrar y probar mayores configuraciones que puedan clasificar significativamente mejor estas áreas.

5. Anexo

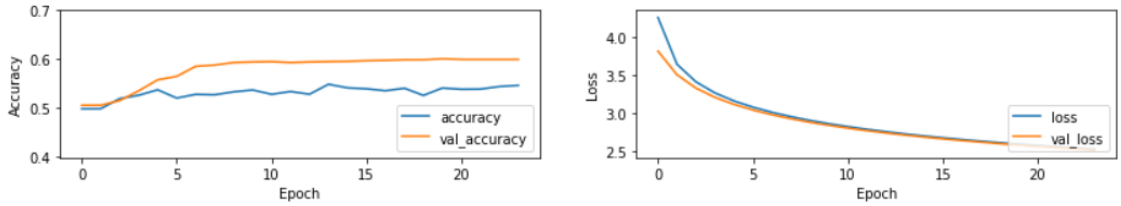


Figura 2: Resultados train model CNN-1
Fuente: Elaboración propia

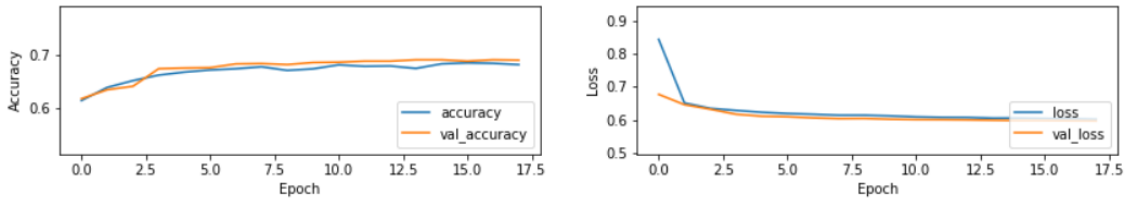


Figura 3: Resultados train model CNN-2
Fuente: Elaboración propia

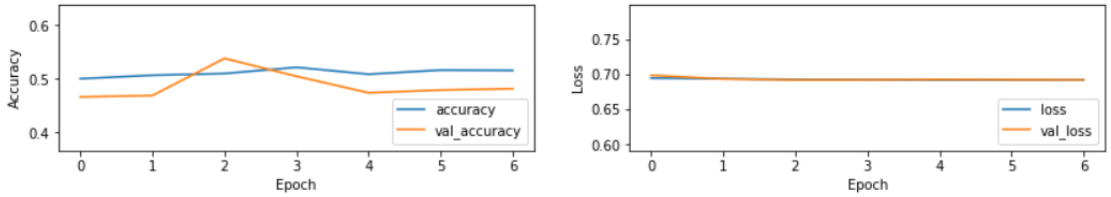


Figura 4: Resultados train model MLP
Fuente: Elaboración propia