Instituto Tecnológico y de Estudios Superiores de Monterrey

TC3006C.101

Inteligencia Artificial Avanzada para la ciencia de datos. (Gpo. 101)

Momento de retroalimentación

Análisis y Reporte sobre el desempeño del modelo.

Profesores:

Dr. Ivan Mauricio Amaya Contreras

Realizado por : Javier de Golferichs - A01139500

11 de septiembre de 2022

1. Introducción

La implementación seleccionada para esta entrega es la segunda entrega, que usa el framework de redes neuronales. La librería usada para el modelo son sklearn.neural_network.

Para este se realiza un ciclo `for` que itera cambiando los parámetros de las redes para así cambiar evaluar el sesgo, varianza y nivel de ajuste de las implementaciones.

El código inicia con la conexión a drive, la importación del dataset (wines.data) y la definición de los datos de entrada y datos de salida (la clase a la que corresponde). Se utilizan todas las características disponibles en el conjunto para hacer la evaluación.

2. Análisis de la implementación

A continuación se presenta la función en la que se obtienen y almacenan los puntajes y matrices de confusión para los parámetros de test y train.

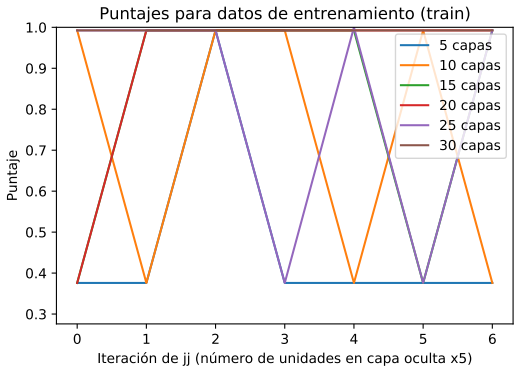
```
cont = 0
for ii in range (5,40, 5):
  if cont == 1:
    scores_train.append(jj_scores_train)
    scores_test.append(jj_scores_test)
    almacenamiento_conf_mat_train.append(jj_mc_train)
    almacenamiento_conf_mat_test.append(jj_mc_test)
  cont = 1
  jj_scores_train = []
  jj_scores_test = []
  jj_mc_train = []
  jj_mc_test = []
  for jj in range(5,40,5):
    cross_val_predict,
    nnRE = MLPClassifier(hidden_layer_sizes=(ii,jj), ## cambiar estos parámetros
                         activation='logistic', verbose=False, solver='adam',
                         learning_rate='adaptive', max_iter=2000)
    nnRE.fit(X_train,y_train)
    jj_scores_train.append(nnRE.score(X_train, y_train))
    jj_scores_test.append(nnRE.score(X_test, y_test))

    jj_mc_train.append(confusion_matrix(y_train,cross_val_predict(nnRE,X_train,y_train, cv =
        10)))
    jj_mc_test.append(confusion_matrix(y_test,cross_val_predict(nnRE,X_test,y_test, cv = 10)))
```

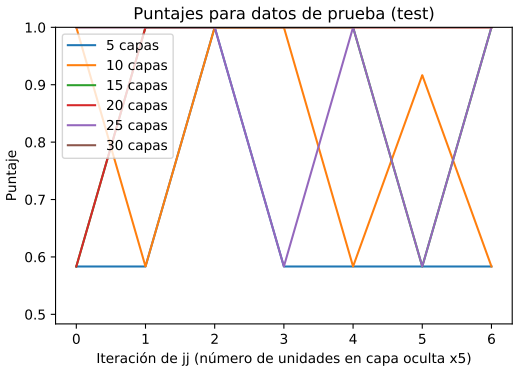
Listing 1: Obtención y almacenamientos de puntajes y matrices de confusión. Nótese que el rango utilizado va desde 5 capas hasta 35 capas, y jj va desde 5 unidades por capa hasta 35.

Ya corrida esta sección del código, se encuentran los valores de los puntajes y matrices de confusión para train y test.

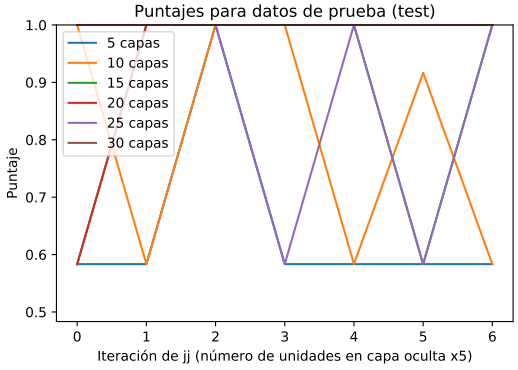
En la fugura 1, se observa el cambio en los puntajes de las tres divisiones del conjunto de datos, que se presentan en las figs. 2 to 4.



(a)



(b)



(c)

Figura 1: 1a Cambio de puntajes en set de datos de entrenamiento. 1b Cambio de puntajes en set de datos de prueba. 1c. Cambio de puntajes en set de datos de prueba.

A continuación se presentan los vectores de los puntajes y su cambio, en donde cada renglón representa un aumento de 5 capas ocultas (5 a 35 capas), y cada columna el número de unidades por capa (de 5 a 35).

```
[[0.37593985 0.9924812  0.37593985 0.37593985 0.9924812  0.9924812 ]
 [0.37593985 0.37593985 0.9924812  0.9924812  0.9924812  0.9924812 ]
 [0.9924812  0.9924812  0.9924812  0.9924812  0.9924812  0.9924812 ]
 [0.37593985 0.9924812  0.9924812  0.9924812  0.37593985 0.9924812 ]
 [0.37593985 0.37593985 0.9924812  0.9924812  1.         0.9924812 ]
 [0.37593985 0.9924812  0.37593985 0.9924812  0.37593985 0.9924812 ]
 [0.37593985 0.37593985 0.9924812  0.9924812  0.9924812  0.9924812 ]]
```

Figura 2: Puntajes de entrenamiento

```
[[0.42424242 0.90909091 0.42424242 0.42424242 0.90909091 0.93939394]
 [0.42424242 0.42424242 0.90909091 0.90909091 0.93939394 0.90909091]
 [0.93939394 0.90909091 0.90909091 0.90909091 0.93939394 0.93939394]
 [0.42424242 0.90909091 0.90909091 0.93939394 0.42424242 0.93939394]
 [0.42424242 0.42424242 0.90909091 0.93939394 0.93939394 0.93939394]
 [0.42424242 0.93939394 0.42424242 0.93939394 0.42424242 0.93939394]
 [0.42424242 0.42424242 0.93939394 0.93939394 0.93939394 0.90909091]]
```

Figura 3: Puntajes de validación

```
[[0.58333333 1.         0.58333333 0.58333333 1.         1.        ]
 [0.58333333 0.58333333 1.         1.         1.         1.        ]
 [1.         1.         1.         1.         1.         1.        ]
 [0.58333333 1.         1.         1.         0.58333333 1.        ]
 [0.58333333 0.58333333 1.         1.         1.         1.        ]
 [0.58333333 0.91666667 0.58333333 1.         0.58333333 1.        ]
 [0.58333333 0.58333333 1.         1.         1.         1.        ]]
```

Figura 4: Puntajes de prueba

2.1. Análisis de sesgo, varianza y nivel de ajuste del modelo

La varianza y el sesgo son inversamente proporcionales, tal que si uno aumenta, por el otro disminuye.

Si hay underfitting se encuentra que hay alto sesgo, y overfitting, se tiene alta varianza.

Para la evaluación de esto se hace una comparación de los puntajes de los tres subconjuntos (entrenamiento, validación y prueba).

En la primera columna de los subconjuntos 2 y 3, se encuentra que se tiene un puntaje bajo en la mayoría de los casos, mientras que en el tercer renglon (caso de las 15 capas con 5 unidades por capa) se tiene un puntaje considerablemente bueno de .99 para entrenamiento y .93 para validacion.

Se obtienen los promedios de los puntajes por categoría de número de capas utilizadas, en donde se observa como el aumento de capas mejora el promedio de los puntajes obtenidos (donde lo que se promedia son los puntajes de al iterar el número de unidades por capa).

```
[0.46401719 0.72824919 0.81632653 0.90440387 0.81740064 0.9924812 ]
```

Figura 5: Promedios de puntajes por número de capas de entrenamiento

```
[0.4978355  0.70562771 0.77489177 0.85714286 0.78787879 0.93073593]
```

Figura 6: Promedios de puntajes por número de capas de validación.

```
[0.64285714 0.80952381 0.88095238 0.94047619 0.88095238 1.        ]
```

Figura 7: Promedios de puntajes por número de capas de prueba.

Ya obtenidos los valores de los puntajes vistos en las figuras figs. 5 to 7, se encuentra que claramente aumentar el número de capas implica un mayor puntaje, al grado de que para los conjuntos de prueba se llega a que se está encontrando con puntaje de 1 para varias de las implementaciones.

Se considera que el modelo que presenta un nivel de sesgo medio es en el que se usan 35 capas ocultas, esto debido a que los puntajes obtenidos para los subconjuntos de entrenamiento y validación superan el puntaje de .90 en todos los casos de unidades. Así como que para los datos de prueba se encuentra que se tiene puntajes perfectos.

Con respecto a la varianza se encuentra que tiene una varianza media, aceptable, debido a que el sesgo no es bajo ni alto.

Se encuentra que no se tiene un buen balance para los casos de 35 capas ocultas debido a que los puntajes obtenidos presentan errores aceptables en los subconjuntos de entrenamiento y de validación.

3. Técnica de regularización

La técnica de regularización involucra el aumento de capas ocultas a 35, lo cual pudo ser encontrado a traves del análisis realizado en la sección anterior.

Se puede observar en los arreglos de las figs. 2 to 4, en los promedios de las figs. 5 to 7 y en las gráficas, en donde se observa que la línea café, que corresponde a este número de capas ocultas, tiene puntajes aceptables para los subconjuntos de entrenamiento, prueba y varianza.