MACHINE LEARNING

Trabajo 1: Redes Neuronales para regresión

Este trabajo está orientado a predecir una variable **continua** a través de redes neuronales. La predicción de variables discretas se hará en el trabajo futuro de Clasificación.

Normas para la realización del trabajo

1) El trabajo es individual.
2) Se entregará el trabajo en pdf en una copia digital enviada al Campus Virtual.
3) El trabajo deberá estar explicado (no basta con responder a las cuestiones), indicando, si es necesario, el código utilizado. Se valora la claridad de exposición en el informe y la estructura. Puede contener Anexos de datos y gráficos o no, todo según vuestro criterio. El trabajo es libre, con lo cual se agradece sentido común.
4) El trabajo está automáticamente suspenso (aunque se podría recuperar en Septiembre incluyendo examen oral) si se da al menos una de las siguientes circunstancias: i) La presentación y explicaciones son escasas. ii) Los modelos comprobados son demasiado limitados. iii) Se observan signos de copia de otros trabajos de otros alumnos.

Se construirá el modelo de redes sobre **una** matriz de datos a elegir voluntariamente, salvo los que falten a 3 clases que tendrán que hacerlo sobre dos matrices, o a 5 o más, que tendrán que hacerlo sobre tres matrices.

En principio se pide que el archivo tratado al menos 300 observaciones, al menos inicialmente 5 variables input y que haya al menos una variable categórica (antes del proceso de selección de variables).

En el Anexo tenéis donde buscar archivos de interés.

La búsqueda del dataset es responsabilidad del alumno. Cuanto más complejo mejor. No responderé a cuestiones del tipo "¿Está bien este dataset para el trabajo?"

Cuestiones generales a responder

Se trata de conseguir obtener el mejor modelo de Red neuronal, estable en cuanto a su performance, y en comparación con un modelo de regresión.

En términos generales,

a) Se requiere un proceso de depuración previa y selección de variables (no se pide que sea demasiado exhaustivo porque el énfasis hay que hacerlo en la construcción de la red).
b) Se requiere la comparación entre la mejor red y regresión; se comparará con diferentes particiones y semillas.
c) Se comprobará el efecto de la variación del número de nodos, y eventualmente del algoritmo.
d) Se probará si es preciso, modelos con diferentes variables
e) Se comparará red neuronal básica por defecto, red con early stopping (tunear el numero de iteraciones), y regresión.
f) Se prefiere validación cruzada repetida a cualquier otro método de comparación de modelos (training-validación repetida o no, validación cruzada sin repetir)
g) El trabajo estará suspenso si se cumple alguna de las siguientes condiciones: si no se hace remuestreo, si no se realiza el tuneo de parámetros,como el número de nodos, iteraciones , learning rate (weight decay), si no se compara con regresión, si no se explica el proceso de selección de variables , o si no se hace al menos una breve descripción y depuración de datos.

h) Se puede utilizar el programa SAS o R o Python (aportado por el profesor o personal) y cualquier modificación o uso de código. Se puede utilizar solo Enterprise Miner pero la nota estará limitada a 6 en este caso.

Fecha de entrega: 3 de Abril

Nota aproximada del trabajo según lo que se haga:

1) Todos los apartados correctamente, proceso básico de depuración y selección de variables, trabajo realizado solo con E Miner- remuestreo con training-test repetido, **nota 5-6**
2) Todos los apartados correctamente, proceso básico de depuración y selección de variables, trabajo realizado con E Miner y R, validación cruzada repetida, **nota 6-8**
4) Todos los apartados correctamente, buen proceso de depuración y selección de variables, trabajo realizado con E Miner y con R, validación cruzada repetida, estudio y tuneo con diferentes sets de variables, datos interesantes o complicados, buena descripción del problema y conclusiones (añadir a las conclusiones tabla básica del modelo de regresión). **nota 8-10**

ANEXO

WEBS DE DATASETS PARA APLICAR TÉCNICAS DE MACHINE LEARNING

RECOMENDADOS PARA EMPEZAR, MUY BIEN ESTRUCTURADOS

****

https://archive.ics.uci.edu/
https://sci2s.ugr.es/keel/datasets.php
https://vincentarelbundock.github.io/Rdatasets/datasets.html

****

1) En uci y keel están los archivos ordenados por clasificación o regresión.

****

2) En la web Rdatasets_
https://vincentarelbundock.github.io/Rdatasets/datasets.html
Archivos de más de 1200 observaciones, lista elaborada por mí:

Item	Rows	Cols	modelo
BEPS	1525	10	clasificación
Caravan	5822	86	clasificación
Gunnels	1592	10	clasificación
Hdma	2381	13	clasificación
Hmda	2381	13	clasificación
VerbAgg	7584	9	clasificación
WVS	5381	6	clasificación
Wells	3020	5	clasificación
YouthRisk2007	13387	6	clasificación
azcabgptca	1959	6	clasificación
dengue	2000	13	clasificación
flchain	7874	11	clasificación
mexico	1359	33	clasificación
mifem	1295	10	clasificación
monica	6367	12	clasificación
ohio	2148	4	clasificación
spam7	4601	7	clasificación
student	9679	13	clasificación
turnout	2000	5	clasificación
voteincome	1500	7	clasificación
Car	4654	70	clasificación multiclase
Chile	2700	8	clasificación multiclase
Kakadu	1827	22	clasificación multiclase
msqR	6411	79	correspondencias
colon	1858	16	cox
cricketer	5960	8	cox
mgus2	1384	10	cox
nwtco	4028	9	cox
BudgetFood	23972	6	regresión
BudgetItaly	1729	11	regresión
BudgetUK	1519	10	regresión
Computers	6259	10	regresión
DoctorContacts	20186	15	regresión
HI	22272	13	regresión
InstInnovation	6208	25	regresión
Males	4360	12	regresión
Males	4360	12	regresión
MathPlacement	2696	16	regresión
MedExp	5574	15	regresión
PatentsRD	1629	7	regresión
SLID	7425	5	regresión
SaratogaHouses	1728	16	regresión
Schooling	3010	28	regresión
Snmesp	5904	8	regresión
Star	5748	8	regresión

VietNamH	5999	11	regresión
Vocab	30351	4	regresión
Wage	3000	11	regresión
Wages	4165	12	regresión
Wages	4165	12	regresión
Workinghours	3382	12	regresión
azpro	3589	6	regresión
baseball	21699	22	regresión
diamonds	53940	10	regresión
mdvis	2227	13	regresión
medpar	1495	10	regresión
nlschools	2287	6	regresión
rwm5yr	19609	17	regresión
science	1385	7	regresión
DoctorAUS	5190	15	regresión muchos dep posibles ver web
OFP	4406	19	regresión varias dependientes
VietNamI	27765	12	regresión y clasi
Gestation	1236	23	regresión y clasificación
NCbirths	1450	15	regresión y clasificación

ZIPEADOS POCO EXPLICADOS

http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets?CGISESSID=10713f6d891653ddcbb7ddbdd9cffb79
https://www.cs.waikato.ac.nz/ml/weka/datasets.html

DATOS COMPLICADOS PERO INTERESANTES

https://www.nature.com/sdata/
http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

CONCURSOS

https://www.kaggle.com/
https://www.drivendata.org/competitions/
http://www.chalearn.org/challenges.html
https://www.kdd.org/kdd-cup

OTROS (google, microsoft, etc.)

https://datasetsearch.research.google.com/
https://aws.amazon.com/public-datasets/
https://msropendata.com/

Frequently Asked Questions (FAQ) sobre el Trabajo

1) Estructura

1) ¿Hay un esquema concreto para el trabajo?

En principio hay un esquema básico, sobre él se puede trabajar y alterar cosas:

1) Descripción de los datos.

2) Depuración y codificación, estandarización (estandarización si se usa R, porque el SAS la hace internamente) y creación de dummies.

3) Estudio de selección de variables; selección primaria del mejor set o mejores sets bajo regresión lineal, usando validación cruzada (CV) repetida y boxplot, examinando gráficamente sesgo y varianza.

4) Con el set (o sets) de variables más interesantes, tunear la red

Con Miner se puede tunear el numero de nodos, el algoritmo, función de activación, iteraciones. El learning rate solo lo he conseguido tunear con el algoritmo bprop o rprop. No os pido que lo hagáis.

En R:

En este caso solo se puede tunear el número de nodos, decay y la maxit o número de iteraciones. Pero lo considero suficiente de momento.

Realizar el proceso de tuneado y presentar boxplots basados en CV repetida.

Comparar resultados con SAS Miner, teniendo en cuenta que la diferencia puede deberse al remuestreo. Y que en Miner solo se puede hacer train test repetido y por tanto los resultados no son igual de fiables.

2) Presentación y comentarios

¿Hay algunas normas concretas sobre presentación y comentarios?

Normalmente se deja al alumno actuar según su sentido común. Se agradece que se muestren pequeños scripts y partes de código intercaladas entre el texto y gráficos. También que se explique suficientemente bien el proceso y toma de decisiones en los tuneados, selección de variables, etc. Se espera que el trabajo pueda servir como documento de referencia al autor en un futuro, para recordar conceptos o técnicas olvidadas.

Respecto a la presentación, solo se pide un pdf. Se pueden añadir anexos a voluntad.

3) Selección de variables

1) ¿Puedo usar otros métodos de selección de variables como la Selección del EMiner aparte de stepwise simple y repetido, RBF y otros ?

Sí, pero al menos tienes que compararlo con el set de variables obtenido con un método clásico estándar como stepwise AIC o BIC. Esto es obligatorio.

Por otra parte, no solo hay que tener en cuenta el resultado empírico que observamos por CV, sino que si un set de variables tiene muchas más variables que otro pero la ganancia en error es muy pequeña, estaremos posiblemente sobreajustando a pesar de que empíricamente en nuestras pruebas parezca mejor.

2) Tengo tres sets de variables tentativos, ¿puedo aplicarlos y tunear las redes en cada uno por separado y comparar?

Sí, es correcto. No es necesario si hay un set claramente mejor que los otros en regresión.

4) ¿Es necesario crear dummies antes de la selección de variables?

Yo lo recomiendo porque hacerlo así da modelos más finos. En particular, por este orden:

1) Crear dummies

2) Eliminar dummies con pocas observaciones con valor 1 (menos de 20-30 por ejemplo).

Eventualmente crear nuevas dummies a partir de uniones de dummies y usar éstas ("si Valencia=1 o Almería=1 entonces Valmeria=1"). Esto último si el conocimiento del contexto lo hace lógico.

4) Redes

¿ Es estricto calcular como mínimo 30 observaciones por parámetro?

Es simplemente una protección, 20 observaciones es menos seguro. Por otro lado, existe lo que se llama la maldición de la dimensionalidad: a más variables input, exponencialmente más observaciones se necesitan para poder representar la relación funcional entre la variable dependiente y las variables input.

Por lo tanto, si tienes 30 variables input en tu modelo no necesitas solo 20 o 30 observaciones por parámetro, posiblemente sea necesario tener 100 para protegerse más, por la alta dimensión del espacio de variables input. No hay fórmulas exactas, se podría decir que 20 observaciones por parámetro para un tamaño moderado de 5 variables input, 40 para 10 variables input, etc. Y se evaluará la ganancia en error. Como se comenta en otras cuestiones, se considera lo empírico (CV) pero también la protección y cuánto ganamos en bajar el error haciendo el modelo más complicado. Si no ganamos lo suficiente, tomamos el modelo más sencillo.

5) Decisiones en tuneado y modelos

A veces no eliges los valores que te recomienda caret o que parecen óptimos viendo el error. Por ejemplo, eliges 5 nodos en una red cuando caret prefiere 15 nodos, pues le dan menor error en sus pruebas de validación cruzada . ¿Cuáles son los criterios para decidirse?

En este caso, seguramente la diferencia de error entre 5 nodos y 15 sea tan pequeña que no merezca la pena usar el modelo más complicado (15 nodos) frente al más sencillo (5 nodos).

En principio los criterios empíricos (es decir, seleccionar directamente el hiperparámetro con menor valor de error promedio en validación cruzada) no son suficientes.

A menudo un tuneado nos da un error por CV relativamente más bajo con una cantidad de parámetros o nodos demasiado alto. Eso ocurre porque las técnicas de remuestreo tampoco son perfectas, y las debemos considerar simultáneamente con medidas de protección ante el sobreajuste. Por lo tanto, los criterios que se deben de tener en cuenta en el proceso de tuneado y selección de modelos deben ser varios:

1) Lo empírico, teniendo en cuenta sesgo y también varianza

2) Aspectos numéricos de sentido común. Por ejemplo, en las redes considerar modelos con al menos 20-30 observaciones por parámetro. .

3) Ante diferencias muy pequeñas entre resultados empíricos, escoger el modelo más sencillo (con menos parámetros).

4) Ante diferencias muy pequeñas entre resultados empíricos, escoger los modelos clásicos (logística, regresión lineal) ante los algoritmos de machine learning.

5) Otras consideraciones de índole práctica sobre el modelo construido:

a) Ante la duda, eligiremos el modelo en el que las variables introducidas sean más lógicas, tengan más sentido para el investigador. Igualmente se debería verificar que el signo del parámetro asociado tenga sentido, en la tabla de la logística (o regresión lineal en caso de predicción de v. continua).

b) Ante la duda, eligiremos el modelo en que las variables introducidas tengan una medición más lógica y directa (suelen ser variables continuas) o más fáciles o baratas de medir, y siempre que esté claro que en sucesivas campañas y tomas de datos van a poder ser medidas con seguridad con los mismos criterios. Y ante la duda las que tengan menos missing.

Por último, a menudo la performance de un algoritmo o tuneado no difiere significativamente de otro. Aunque para fortalecer la intuición y la visión gráfica no hemos tenido en cuenta la comparación de la performance con contrastes de hipótesis estadísticos, se podrían también tener en cuenta.

Algunos errores típicos en los trabajos

1) Considerar las semillas y/o número de grupos de CV (validación cruzada) como un parámetro de los modelos.

Este es un error grave que ocurre en un 5% del alumnado. Las semillas de aleatorización y el número de grupos son solamente un marco, esquemas de muestreo para comparación.

a) En principio todos las pruebas de algoritmos y tuneados deben llevarse a cabo con el mismo esquema (mismas semillas, mismo número de grupos) para EVITAR que la diferencia que observamos entre un algoritmo y otro sea debida a la reestructuración de la muestra sobre la que se prueban las predicciones, y no al buen o mal funcionamiento de cada algoritmo.

b) Otro error típico relacionado es cuando un alumno, para comprobar la estabilidad de una solución-algoritmo, representa varias cajas (box plot) en paralelo, cada una construida con diferentes semillas. Intento muy loable, pero ahí realmente lo que se está viendo es la variabilidad o varianza de los errores al variar la semilla. En realidad se está representando esa variabilidad del error en paralelo en varias cajas, en vez de representarla en vertical que es lo más apropiado.

c) Más grave es cuando se presenta la comparación de un solo algoritmo o varios en varias cajas, cada caja construida con *diferente* número de grupos de validación cruzada.

En este caso no se está llegando a buenas conclusiones. Pues salvo fluctuaciones aleatorias, el error de CV es menor cuanto mayor sea el número de grupos, pues al aumentar los grupos, se está construyendo el modelo con más observaciones y por lo tanto será más eficiente al predecir el grupo excluido.

2) No poner conclusiones .

Ocurre poco frecuentemente. Se debe tomar una decisión sobre uno o varios modelos adecuados y dar alguna explicación o reflexión personal.

3) No hacer depuración ni buena selección de variables.

Ocurre poco frecuentemente. No pido explicaciones exhaustivas en la depuración, pero sí que hagáis un mínimo esfuerzo en la selección de variables.

4) Automatizar todo el proceso, dejando a SAS o R el tuneado estricto.

Ocurre poco frecuentemente. Como podéis ver en las FAQ, la idea es combinar lo empírico con sentido común.

En el caso de los hiperparámetros (nodos o iteraciones en la red) en tuneado, aparte de la protección contra el sobreajuste, hay que buscar patrones en los gráficos y resultados. Cuando se fija un

hiperparámetro al valor 0.0345 como óptimo uno tiene que estar más o menos seguro de que en un entorno de ese valor se tendrían resultados similares. En general es mejor buscar un entorno estable en un error aceptable en un rango de valores del hiperparámetro, que un punto exacto de valor del hiperparámetro que nos da óptimo error, porque este punto realmente depende de nuestra muestra y estructura de remuestreo.

Igualmente en cualquier estudio de hiperparámetros y comparación de algoritmos nunca hay que perder de vista los valores numéricos del error, pues las diferencias de error pueden ser mínimas en la práctica aunque los gráficos, al estar a escala, nos parezcan muy diferentes.