

APR: Práctica sobre Redes Neuronales Multicapa (MLP)

Roselló Beneitez, N. U.; Roselló Oviedo, M.

6 de Enero de 2020

Índice

1. Descripción de la práctica	3
2. Redes neuronales en MNIST	3
3. Conclusiones	5

Índice de figuras

1. Valores de dimensionalidad reducida mediante PCA contra valores de neuronas en la capa oculta, usando el 50 % de las muestras para entrenamiento.	3
2. Valores de dimensionalidad reducida mediante PCA contra valores de neuronas en la capa oculta, usando el 80 % de las muestras para entrenamiento.	4
3. Visualización gráfica de los resultados para 50 %.	4
4. Visualización gráfica de los resultados para 80 %.	5
5. Comparativa de los resultados para 50 % vs 80 %.	5

1. Descripción de la práctica

En esta práctica se ha trabajado con redes neuronales multicapa en Octave mediante la librería *nnet*, la cual dispone de sus propios métodos auxiliares con los que entrenar un *multilayer perceptron*.

La primera parte de la práctica, de carácter didáctico e iniciativo, consiste en experimentar con el corpus *hart* para familiarizarse con la metodología de la librería, y no se haya incluida en esta memoria. La segunda parte, la cual desarrollaremos a continuación, se centra en la clasificación de dígitos de la base de datos MNIST.

2. Redes neuronales en MNIST

Se ha entrenado la red neuronal con diversos valores de porcentaje de muestras destinadas al entrenamiento, concretamente 50 % y 80 %; el resto de los valores de entrenamiento se han destinado al conjunto de validación del modelo.

Puesto que los recursos de los equipos disponibles no eran ilimitados, han habido ejecuciones para las que no ha sido posible obtener un resultado, bien porque el entrenamiento ha sido terminado por el programa en el entrenamiento, o bien porque no se disponía de la suficiente memoria RAM como para almacenar todos los cálculos en memoria, por lo que el programa utilizaba memoria virtual procedente del disco y como consecuencia el tiempo del entrenamiento aumentaba de manera drástica.

Para ambos porcentajes de entrenamiento, se han realizado experimentos variando tanto la reducción de dimensionalidad mediante PCA como el número de neuronas en la capa oculta (todos los experimentos cuentan con una sola capa oculta). El error obtenido se muestra en cada correspondiente casilla de las tablas siguientes tablas:

MNIST kernel polinómico		
Grado	% error	% IC ($\alpha = 0.05$)
2	1.95	$\pm 0.1106\%$
3	2.09	$\pm 0.1145\%$
4	2.63	$\pm 0.1280\%$
5	3.35	$\pm 0.1440\%$
6	4.12	$\pm 0.1590\%$
7	5.27	$\pm 0.1788\%$
8	6.36	$\pm 0.1953\%$
9	7.42	$\pm 0.2097\%$

Figura 1: Valores de dimensionalidad reducida mediante PCA contra valores de neuronas en la capa oculta, usando el 50 % de las muestras para entrenamiento.

MNIST kernel polinómico		
Grado	% error	% IC ($\alpha = 0.05$)
2	1.95	$\pm 0.1106\%$
3	2.09	$\pm 0.1145\%$
4	2.63	$\pm 0.1280\%$
5	3.35	$\pm 0.1440\%$
6	4.12	$\pm 0.1590\%$
7	5.27	$\pm 0.1788\%$
8	6.36	$\pm 0.1953\%$
9	7.42	$\pm 0.2097\%$

Figura 2: Valores de dimensionalidad reducida mediante PCA contra valores de neuronas en la capa oculta, usando el 80 % de las muestras para entrenamiento.

MNIST kernel polinómico		
Grado	% error	% IC ($\alpha = 0.05$)
2	1.95	$\pm 0.1106\%$
3	2.09	$\pm 0.1145\%$
4	2.63	$\pm 0.1280\%$
5	3.35	$\pm 0.1440\%$
6	4.12	$\pm 0.1590\%$
7	5.27	$\pm 0.1788\%$
8	6.36	$\pm 0.1953\%$
9	7.42	$\pm 0.2097\%$

Figura 3: Visualización gráfica de los resultados para 50 %.

MNIST kernel polinómico		
Grado	% error	% IC ($\alpha = 0.05$)
2	1.95	$\pm 0.1106\%$
3	2.09	$\pm 0.1145\%$
4	2.63	$\pm 0.1280\%$
5	3.35	$\pm 0.1440\%$
6	4.12	$\pm 0.1590\%$
7	5.27	$\pm 0.1788\%$
8	6.36	$\pm 0.1953\%$
9	7.42	$\pm 0.2097\%$

Figura 4: Visualización gráfica de los resultados para 80 %.

MNIST kernel polinómico		
Grado	% error	% IC ($\alpha = 0.05$)
2	1.95	$\pm 0.1106\%$
3	2.09	$\pm 0.1145\%$
4	2.63	$\pm 0.1280\%$
5	3.35	$\pm 0.1440\%$
6	4.12	$\pm 0.1590\%$
7	5.27	$\pm 0.1788\%$
8	6.36	$\pm 0.1953\%$
9	7.42	$\pm 0.2097\%$

Figura 5: Comparativa de los resultados para 50 % vs 80 %.

3. Conclusiones

Como se puede ver cuando se analizan los porcentajes de error y los intervalos de confianza, la mayoría de los errores presentan una diferencia estadísticamente significativa usando el 80 % de las muestras para entrenamiento con respecto a usar el 50 % de muestras. Esto indica, como parece lógico, que es útil tener un porcentaje más elevado dedicado al entrenamiento que a la validación, lo cual tiene sentido ya que al introducir más muestras en el conjunto de validación se

están desaprovechando muestras perfectamente válidas para el entrenamiento. Según lo visto en la teoría, el conjunto de validación debería establecer cuándo se pasa de generalizar a sobreentrenar el modelo, por lo que en nuestra opinión un conjunto tan grande como el de pruebas debería ser suficiente en la mayoría de los casos para evitar un desaprovechamiento de los datos de entrenamiento. Cabe recordar que en nuestras ejecuciones el tamaño del conjunto de validación era de 30000 y de 12000 muestras, esto es, un 50 % y un 20 % del conjunto de entrenamiento respectivamente.

Con respecto a la variación del error con respecto al número de dimensiones proyectadas con PCA y al número de neuronas en la capa oculta, se puede ver un descenso notable del error a medida que tanto las dimensiones proyectadas con PCA como el número de neuronas en la capa oculta aumenta. No es beneficioso para el modelo, sin embargo, aumentar el número de una de estas dos componentes manteniendo la otra marginalmente mínima, como se puede apreciar en las tablas obtenidas. Lo óptimo es encontrar un balance entre ambas componentes: el mejor resultado del error para cada porcentaje de entrenamiento coincide en ser aquel obtenido mediante 40 neuronas en la capa oculta y la proyección a 30 dimensiones con PCA para el conjunto de entrenamiento, tanto al 50 % (con error 4.28 %) y al 80 % (con error 3.40 %), respectivamente.

Por último, cabe mencionar que el modelo conexionista no siempre es la solución óptima a todos los problemas. La menor tasa de error encontrada ha sido de un 3.40 %, mientras que las tasas de error encontradas en prácticas anteriores han sido de un 2.20 % para la *Práctica 1 sobre Mixtura de gaussianas* y de un 1.95 % para la *Práctica 2 sobre Máquinas de vector soporte*. Esto no quiere decir que estas dos técnicas sean mejores que las redes neuronales, sino que son mejores *para este problema específico de clasificación de dígitos*. Es por esto que es necesario un cierto conocimiento experto sobre las técnicas de aprendizaje automático, además de los mejores *toolkits* que las implementan, con el fin de saber cuáles son sus mejores aplicaciones.