

Trabajo práctico 1: Optimización y Redes Neuronales

Juan José Cordero Gómez
Escuela de computación
Instituto Tecnológico de Costa Rica

Luis Diego Hidalgo Blanco
Escuela de computación
Instituto Tecnológico de Costa Rica

Ricardo Sánchez Alpizar
Escuela de computación
Instituto Tecnológico de Costa Rica

2. (30 PUNTOS) PERCEPTRÓN MULTI-CAPA PARA DETECCIÓN DE GLAUCOMA EN IMÁGENES DE FONDO DE OJO

1.a) Ejecute el entrenamiento 10 veces por 15 épocas por corrida, y reporte la tasa de aciertos, falsos positivos y falsos negativos promedio y su desviación estándar para esas 10 corridas.

Para empezar, se presenta una tabla resumen de la ejecución de 10 corridas utilizando el perceptrón multi-capas con distintos valores para el LR:

TABLE I
RESUMEN DE RESULTADOS PARA MLP CON DIFERENTES LR.

| Iteración | LR | # Aciertos | # Fallos | Falsos positivos | Falsos negativos | Precisión (%) |
|-------------------------------|-------|------------|----------|------------------|------------------|---------------|
| 1 | 0.1 | 105 | 105 | 105 | 0 | 50.0 |
| 2 | 0.1 | 105 | 105 | 105 | 0 | 50.0 |
| 3 | 0.01 | 185 | 25 | 22 | 3 | 88.0952 |
| 4 | 0.01 | 167 | 43 | 0 | 43 | 79.5238 |
| 5 | 0.03 | 126 | 84 | 84 | 0 | 60.0 |
| 6 | 0.03 | 117 | 93 | 93 | 0 | 55.7143 |
| 7 | 0.001 | 129 | 81 | 79 | 2 | 61.4286 |
| 8 | 0.001 | 113 | 97 | 97 | 0 | 53.8095 |
| 9 | 0.003 | 155 | 55 | 52 | 3 | 73.8095 |
| 10 | 0.003 | 179 | 31 | 13 | 18 | 85.2381 |
| Desviación estándar: 29.2646. | | | | | | |

Seguidamente, se muestra una tabla con la ejecución de 10 corridas empleando únicamente el factor de LR = 0.01:

TABLE II
RESUMEN DE RESULTADOS PARA MLP CON UN LR CONSTANTE DURANTE 10 PRUEBAS.

| Iteración | LR | # Aciertos | # Fallos | Falsos positivos | Falsos negativos | Precisión (%) |
|-------------------------------|------|------------|----------|------------------|------------------|---------------|
| 1 | 0.01 | 184 | 26 | 20 | 6 | 87.619 |
| 2 | 0.01 | 184 | 26 | 20 | 6 | 87.619 |
| 3 | 0.01 | 177 | 33 | 28 | 5 | 84.2857 |
| 4 | 0.01 | 188 | 22 | 6 | 16 | 89.5238 |
| 5 | 0.01 | 156 | 54 | 53 | 1 | 74.2857 |
| 7 | 0.01 | 164 | 46 | 45 | 1 | 78.0952 |
| 6 | 0.01 | 163 | 47 | 45 | 2 | 77.619 |
| 8 | 0.01 | 133 | 77 | 77 | 0 | 63.3333 |
| 9 | 0.01 | 175 | 35 | 0 | 35 | 83.3333 |
| 10 | 0.01 | 188 | 22 | 3 | 19 | 89.5238 |
| Desviación estándar: 17.4789. | | | | | | |

3. (40 PUNTOS) REDES CONVOLUCIONALES PARA DETECCIÓN DE GLAUCOMA EN IMÁGENES DE FONDO DE OJO

1. (20 puntos) Implemente el filtro de “Unsharp masking” para la mejora de las imágenes, según lo especificado en el material del curso.

a) Compruebe y comente su uso para las imágenes de fondo de ojo, mostrando los resultados. Use al menos dos valores distintos de la ganancia λ .

: El filtro de unsharp masking, realza los bordes de las imágenes y ayuda en los modelos de clasificación de imágenes. Para aplicar este filtro, se emplea la siguiente formula:

$$G = U + (U * N) \times \lambda$$

Donde:

U = Imagen original.

λ = Coeficiente de ganancia.

N = Núcleo Gaussiano de desenfoque (blur).

$(U * N)$ implica la convolución entre la imagen original y el filtro Gaussiano de desenfoque (blur).

El detalle del código empleado se encuentra en Jupyter notebook llamado: TP1 - UnsharpMask-AlexNet.ipynb.

Una vez implementado el código esto es un ejemplo de la salida obtenida al emplear diferentes coeficientes de ganancia:

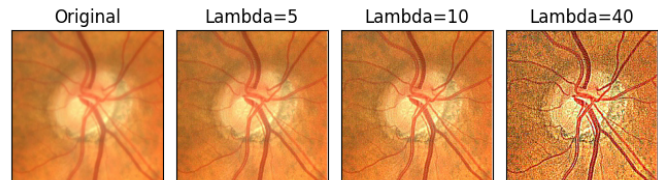


Fig. 1. Ejemplo de uso de filtro de Unsharp masking.

Como se puede observar diferentes valores del coeficiente de ganancia λ resultan en diferentes saturaciones, es necesario realizar el ajuste según el objetivo del modelo y acorde a las imágenes del data set.

Con $\lambda = 10$ parece ser un balance adecuado donde es evidente el realce de los bordes con respecto a la imagen original.

El $\lambda = 5$ parece tener un efecto poco perceptible que podría no aportar mucho al procesamiento del modelo.

Por otro lado el $\lambda = 40$ luce bastante saturado lo cual podría ser perjudicial para el modelo en términos del “ruido” que se observa en la figura 1.

2. (20 puntos) Implemente manualmente (especificando las capas) en pytorch la arquitectura de AlexNet. Entrene la red usando el conjunto de datos de imágenes de fondo de ojo. Calibre los hiper-parámetros necesarios para obtener los mejores resultados posibles y repórtelos. Ejecute el entrenamiento 10 veces por 15 épocas por corrida, y reporte la tasa de aciertos, falsos positivos y falsos negativos promedio y su desviación estándar para esas 10 corridas.

b) Compare los resultados respecto a lo obtenido con el perceptrón multi-capas y coméntelos.

: Se procede a realizar varias pruebas con el modelo con el fin de encontrar un learning rate (LR) que más lo beneficia y produce resultados más precisos, para lo cual se prueban dos veces los siguientes LR:

- 0.0
- 0.01
- 0.03
- 0.001
- 0.003

Los resultados de ambas pruebas son promediados para determinar el LR más efectivo para el modelo.

El detalle de cada entrenamiento y prueba del modelo se encuentra en el Jupyter notebook llamado: TP1 - UnsharpMask-AlexNet.ipynb.

A continuación se muestra el resumen de los resultados, todas las pruebas se realizaron con 210 imágenes, tabla: III:

TABLE III
RESUMEN DE RESULTADOS PARA ALEXNET CON DIFERENTES LR.

| Iteración | LR | # Aciertos | # Fallos | Falsos positivos | Falsos negativos | Precisión (%) |
|-------------------------------|-------|------------|----------|------------------|------------------|---------------|
| 1 | 0.1 | 178 | 32 | 29 | 3 | 84.7619 |
| 2 | 0.1 | 169 | 41 | 11 | 30 | 80.4762 |
| 3 | 0.01 | 204 | 6 | 5 | 1 | 97.1429 |
| 4 | 0.01 | 196 | 14 | 13 | 1 | 93.3333 |
| 5 | 0.03 | 192 | 18 | 11 | 7 | 91.4286 |
| 6 | 0.03 | 187 | 23 | 13 | 10 | 89.0476 |
| 7 | 0.001 | 197 | 13 | 10 | 3 | 93.8095 |
| 8 | 0.001 | 200 | 10 | 6 | 4 | 95.2381 |
| 9 | 0.003 | 199 | 11 | 8 | 3 | 94.7619 |
| 10 | 0.003 | 201 | 9 | 5 | 4 | 95.7143 |
| Desviación estándar: 10.6212. | | | | | | |

De la tabla anterior se puede identificar el LR de 0.01 como uno de los que mejores resultados produce, por lo que se procede a hacer 10 repeticiones con dicho LR y a calcular la desviación estándar del mismo:

TABLE IV
RESUMEN DE RESULTADOS PARA ALEXNET CON UN LR CONSTANTE DURANTE 10 PRUEBAS.

| Iteración | LR | # Aciertos | # Fallos | Falsos positivos | Falsos negativos | Precisión (%) |
|------------------------------|------|------------|----------|------------------|------------------|---------------|
| 1 | 0.01 | 201 | 9 | 9 | 0 | 95.7143 |
| 2 | 0.01 | 200 | 10 | 8 | 2 | 95.2381 |
| 3 | 0.01 | 200 | 10 | 8 | 2 | 95.2381 |
| 4 | 0.01 | 199 | 11 | 6 | 5 | 94.7619 |
| 5 | 0.01 | 199 | 11 | 10 | 1 | 94.7619 |
| 6 | 0.01 | 1898 | 21 | 20 | 1 | 90 |
| 7 | 0.01 | 194 | 16 | 3 | 13 | 92.381 |
| 8 | 0.01 | 202 | 8 | 5 | 3 | 96.1905 |
| 9 | 0.01 | 197 | 13 | 3 | 10 | 93.8095 |
| 10 | 0.01 | 202 | 8 | 8 | 0 | 96.1905 |
| Desviación estándar: 3.8483. | | | | | | |

Como es de esperar los resultados empleando el mismo LR son bastante consistentes, aún cuando han sido entrenados tomando diferentes muestras de imágenes y por cada corrida se realiza un reinicio de kernel con el de que las corridas sean lo más “limpias” posibles.

Finalmente y con el fin de facilitar la interpretación de resultados se unifican los resultados de cada pila de pruebas por cada modelo.

A continuación los resultados empleando diferentes LR en cada modelo:

TABLE V
COMPARACIÓN DE MLP Y ALEXNET CON LR DIFERENTES.

| Iter. | LR | MLP # Aciertos | AlexNet # Aciertos | MLP # Fallos | AlexNet # Fallos | MLP Falsos pos. | AlexNet Falsos pos. | MLP Falsos neg. | AlexNet Falsos neg. | MLP Precisión (%) | AlexNet Precisión (%) | Diferencia (%) | Dif. Prom. (%) |
|-------|-------|----------------|--------------------|--------------|------------------|-----------------|---------------------|-----------------|---------------------|-------------------|-----------------------|----------------|----------------|
| 1 | 0.1 | 105 | 201 | 105 | 9 | 105 | 3 | 0 | 0 | 50 | 95.7143 | 45.7143 | 45.4762 |
| 2 | 0.1 | 105 | 200 | 105 | 10 | 105 | 8 | 0 | 2 | 50 | 95.2381 | 45.2381 | |
| 3 | 0.01 | 185 | 200 | 25 | 10 | 22 | 8 | 3 | 2 | 88.0952 | 95.2381 | 7.1429 | 11.1905 |
| 4 | 0.01 | 167 | 199 | 43 | 11 | 0 | 6 | 43 | 5 | 79.5238 | 94.7619 | 15.2381 | |
| 5 | 0.03 | 126 | 199 | 84 | 11 | 84 | 10 | 0 | 1 | 60 | 94.7619 | 34.7619 | 34.5238 |
| 6 | 0.03 | 117 | 1898 | 93 | 21 | 93 | 20 | 0 | 1 | 55.7143 | 90 | 34.2857 | |
| 7 | 0.001 | 129 | 194 | 81 | 16 | 79 | 3 | 2 | 13 | 61.4286 | 92.381 | 30.9524 | 36.6667 |
| 8 | 0.001 | 113 | 202 | 97 | 8 | 97 | 5 | 0 | 3 | 53.8095 | 96.1905 | 42.381 | |
| 9 | 0.003 | 155 | 197 | 55 | 13 | 52 | 3 | 3 | 10 | 73.8095 | 93.8095 | 20 | 15.4762 |
| 10 | 0.003 | 179 | 202 | 31 | 8 | 13 | 8 | 18 | 0 | 85.2381 | 96.1905 | 10.9524 | |

En general como se puede observar en la tabla anterior los resultados entre cada modelo con el mismo LR varían sustancialmente, en algunos casos con una diferencia promedio de hasta un 45% (diferencia promediada entre las dos iteraciones empleando el mismo LR). A excepción del caso del LR 0.001, donde el MLP muestra el mejor comportamiento y aquí la diferencia promedio es de un 11%.

Dado que en las iteraciones con el LR de un 0.001 presentan la menor diferencia promedio entre modelos, se procede a realizar 10 corridas de ambos modelos con el mismo LR para efectos de comparación y cálculo de la desviación estándar en términos más semejantes:

TABLE VI
COMPARACIÓN DE MLP Y ALEXNET CON LR CONSTANTE.

| Iteración | LR | MLP # Aciertos | AlexNet # Aciertos | MLP # Fallos | AlexNet # Fallos | MLP Falsos positivos | AlexNet Falsos positivos | MLP Falsos negativos | AlexNet Falsos negativos | MLP Precisión (%) | AlexNet Precisión (%) |
|----------------------|------|----------------|--------------------|--------------|------------------|----------------------|--------------------------|----------------------|--------------------------|-------------------|-----------------------|
| 1 | 0.01 | 184 | 201 | 26 | 9 | 20 | 9 | 6 | 0 | 87.619 | 95.7143 |
| 2 | 0.01 | 184 | 200 | 26 | 10 | 20 | 8 | 6 | 2 | 87.619 | 95.2381 |
| 3 | 0.01 | 177 | 200 | 33 | 10 | 28 | 8 | 5 | 2 | 84.2857 | 95.2381 |
| 4 | 0.01 | 188 | 199 | 22 | 11 | 6 | 6 | 16 | 5 | 89.5238 | 94.7619 |
| 5 | 0.01 | 156 | 199 | 54 | 11 | 53 | 10 | 1 | 1 | 74.2857 | 94.7619 |
| 6 | 0.01 | 164 | 189 | 46 | 21 | 45 | 20 | 1 | 1 | 78.0952 | 90 |
| 7 | 0.01 | 163 | 194 | 47 | 16 | 45 | 3 | 2 | 13 | 77.619 | 92.381 |
| 8 | 0.01 | 133 | 202 | 77 | 8 | 77 | 5 | 0 | 3 | 63.3333 | 96.1905 |
| 9 | 0.01 | 175 | 197 | 35 | 13 | 0 | 3 | 35 | 10 | 83.3333 | 93.8095 |
| 10 | 0.01 | 188 | 202 | 22 | 8 | 3 | 8 | 19 | 0 | 89.5238 | 96.1905 |
| Desviación estándar: | | 17.4789 | | 3.8483 | | | | | | | |

De los resultados anteriores se puede concluir que aún usando el LR más efectivo en el modelo del MLP, AlexNet sigue siendo un modelo más superior para resolver el problema de detección de glaucoma en las imágenes de fondo de ojo.