

ANÁLISIS DE LA INFLUENCIA DE CAPAS DE BATCH NORMALIZATION EN LA RED SOCO

Por Antonio Jesús García Nieto

CONTROL DE VERSIONES

[illegible]

TABLA DE CONTENIDO

| | |
|---------------------------------|---|
| Control de versiones..... | 2 |
| Introducción | 4 |
| Planteamiento del estudio | 4 |
| Resultados | 5 |
| Conclusión del experimento..... | 6 |
| Bibliografía..... | 6 |

INTRODUCCIÓN

En este experimento estudiaremos la variación de la precisión en función de la introducción de capas de normalización de lotes (conocidas como capas de BN o BatchNormalization). Nos basaremos en algunos estudios como (Chen et al., 2019; Ioffe & Szegedy, 2015). Este estudio, entre otras cosas, realiza una exploración en lo referente a optimizadores y ratios de aprendizaje para el problema específico de la detección y clasificación de lesiones de piel.

La hipótesis que pretendemos validar en este estudio es:

1. *Existe un optimizador que tiene un mejor desempeño que RMSProp para nuestro problema.*

PLANTEAMIENTO DEL ESTUDIO

El estudio consistirá en una comparación de la precisión media entre distintas configuraciones de la red SOCO.

La configuración general será:

Se seguirá el mismo esquema de entrenamiento que en el experimento 1. Recordemos que el este es:

- El conjunto de entrenamiento = source (60% del total del conjunto de datos),
- El conjunto de reentrenamiento = train (70% del 40% restante del conjunto de datos),
- El conjunto de test = test (21% del 40% restante del conjunto de datos)
- El conjunto de validación = validación (9% del 40% restante del conjunto de datos)

Los hiperparámetros serán los siguiente:

- Ratio de aprendizaje de 0.0001, para todos los optimizadores.
- Épocas: 15 para entrenamiento y para transferencia.
- Callbacks: Se detiene el entrenamiento si la red consigue peores resultados en 5 ocasiones consecutivas.
- Tamaño de lote: 16
- Optimizador Amsgrad

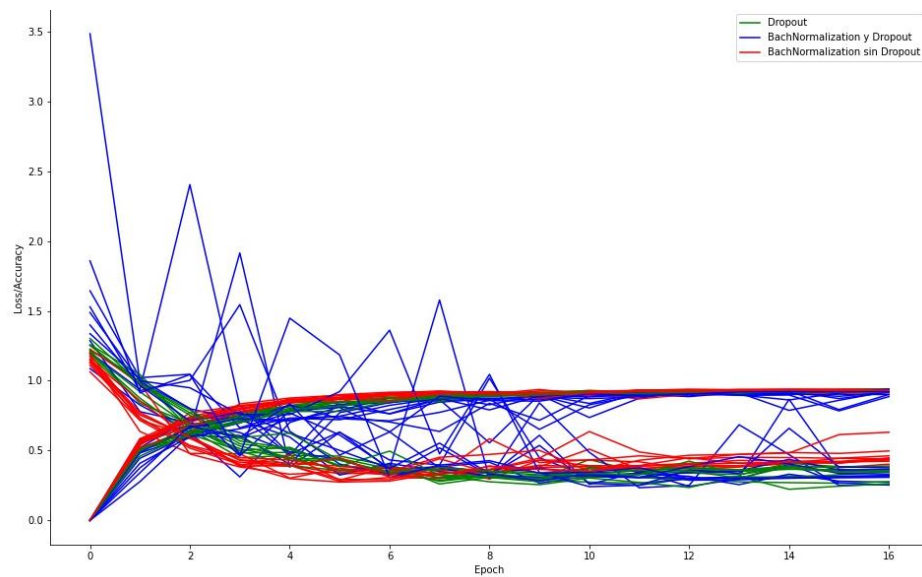
Se realizarán tres pruebas:

1. El modelo de la red SOCO según el experimento dos. Es decir, la red SOCO con PReLU, Dropout.
2. Una modificación de este modelo añadiéndole capas de normalización por lotes.
3. Una modificación del modelo inicial quitándole la capa de dropout y añadiéndole capas de normalización por lotes.

Para asegurar la robustez de la prueba se entrenará 10 veces cada modelo con cada configuración.

RESULTADOS

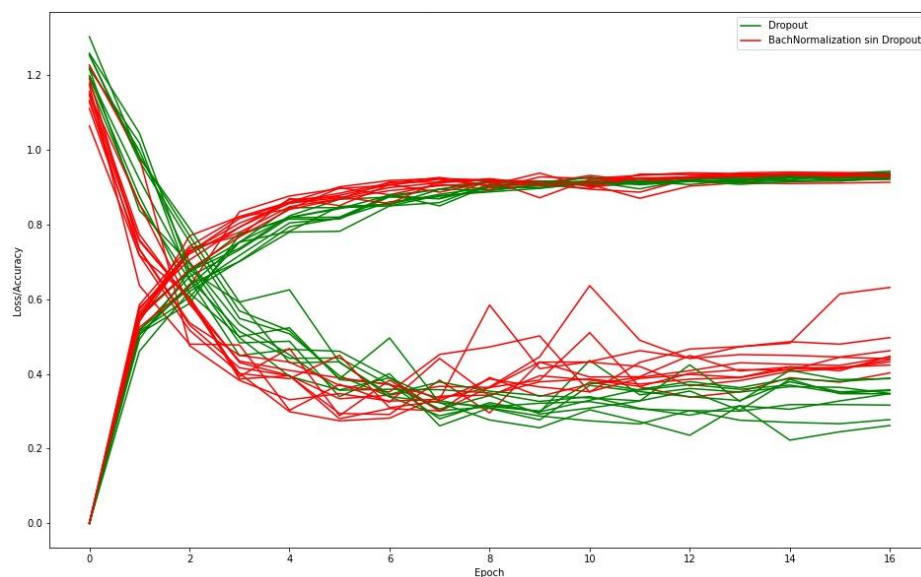
En primer lugar, se comparará la precisión sobre los datos de entrenamiento y el valor de pérdida de todos los modelos entrenados.



Llama la atención el comportamiento del modelo al aplicar ambas técnicas. Se puede observar como en este caso el comportamiento es más volátil e inestable, no obstante, tanto la pérdida como la precisión parece tender a la estabilidad a medida que se suceden las épocas.

Si descartamos este modelo y nos centramos en los que aplican solo Dropout o capas de BN, podremos dar con la siguiente imagen.

Como detalles importantes, parece que la red tiende a conseguir una mejor precisión al aplicar técnicas de BN mientras que las técnicas de Dropout consiguen una mayor pérdida.



Los resultados se pueden resumir en la siguiente tabla.

| | Dropout | BachNormalization y Dropout | BachNormalization sin Dropout |
|---|-------------------------------|-----------------------------|-------------------------------|
| 0 | 0.925926 | 0.910178 | 0.932925 |
| 1 | 0.921260 | 0.925343 | 0.938174 |
| 2 | 0.927384 | 0.923301 | 0.927676 |
| 3 | 0.929425 | 0.916010 | 0.932925 |
| 4 | 0.931175 | 0.916594 | 0.924176 |
| 5 | 0.932050 | 0.924468 | 0.932925 |
| 6 | 0.927384 | 0.920968 | 0.933800 |
| 7 | 0.929717 | 0.885389 | 0.933508 |
| 8 | 0.929717 | 0.930009 | 0.931759 |
| 9 | 0.936716 | 0.919510 | 0.939049 |
| | | | |
| | Dropout | 0.929076 | |
| | BachNormalization y Dropout | 0.917177 | |
| | BachNormalization sin Dropout | 0.932692 | |

Se puede apreciar que la técnica que da un mayor desempeño es la de aplicar solamente capas de BachNormalization.

CONCLUSIÓN DEL EXPERIMENTO

Los resultados obtenidos han sido muy semejantes a los del estudio (Jayalakshmi & Kumar, 2019). Adam ha tomado un buen desempeño, y en ambos casos superior a RMSProp.

El rendimiento se incrementa en un 2.6% cuando se usa Amsgrad frente a RMSProp. Por lo tanto, podemos afirmar que la hipótesis enunciada se cumple, y que en lugar de usar RMSProp podemos pasar a utilizar Amsgrad como optimizador.

No obstante, este estudio nos deja otra pequeña conclusión en el aire, y es que 30 épocas de entrenamiento (15 mediante scratching y 15 mediante transfer learning) no son suficientes para conseguir el mejor valor en cuanto a precisión media. Una idea clave para siguientes estudios puede ser trabajar con un rango variable de épocas mediante planificación del entrenamiento o paradas preestablecidas en el algoritmo de optimización.

BIBLIOGRAFÍA

- Chen, G., Chen, P., Shi, Y., Hsieh, C.-Y., Liao, B., & Zhang, S. (2019). Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks. *ArXiv*. <http://arxiv.org/abs/1905.05928>
- Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. PMLR. <http://proceedings.mlr.press/v37/loff15.html>
- Jayalakshmi, G. S., & Kumar, V. S. (2019, February 1). Performance analysis of convolutional neural network (CNN) based cancerous skin lesion detection system. *ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings*. <https://doi.org/10.1109/ICCIDS.2019.8862143>