

Algunas dudas y fallos en la entrega intermedia

Cuidado con lo que decís en la documentación:

- Todo lo que afirméis debe estar soportado por vuestros resultados. Afirmaciones contradictorias con vuestros resultados resultarán en calificaciones (mucho) más bajas. No copiéis ciegamente lo que os dan los LLM, comprobar que es coherente con vuestros resultados.
- Todos los conceptos que empleéis deben estar adecuadamente definidos y explicados. Si me decís "la función de activación X tiene mejor equilibrio entre estabilidad y rendimiento" entonces me tenéis que explicar qué queréis decir con estabilidad y con rendimiento, y cómo se mide ese equilibrio para decir que la función X ha demostrado ser "mejor" en ese equilibrio. Fijaos que nosotros hemos estado hablando de test_accuracy y tiempo de entrenamiento, dos magnitudes muy concretas y fáciles de medir. Si afirmáis que un modelo es más estable que otro, tenéis que dejar claro a qué os referís con estabilidad, ¿significa que tiene menor desviación estándar en test_accuracy entre varias repeticiones? ¿o que tiene menor diferencia entre el máximo y el mínimo? ¿o tarda siempre más o menos lo mismo? ¿o que la curva de evolución tiene menos vaivenes? Las conclusiones que no cumplan con este criterio no se tendrán en cuenta por considerarse no fundamentadas.

¿Número de épocas en cada experimento?

- Es mejor que utilicéis un EarlyStopping bien configurado desde la tarea MLP3 en adelante.
- Hacer pruebas con el mismo número de épocas (sin EarlyStopping) para modelos diferentes darán resultados distorsionados si los distintos modelos necesitan distinto número de épocas para converger.

¿Cómo hacer las gráficas?

- Las de evolución de (train_)accuracy, (train_)loss, val_accuracy, val_loss deben tener las cuatro líneas en una sola gráfica, con dos ejes verticales, uno para accuracy y otro para loss. Es decir: una sola gráfica con cuatro curvas. Si tienes dos gráficas para esas cuatro curvas tendrás menos puntuación.
- Las de barras, que comparan varios modelos, deben mostrar test_accuracy y tiempo de entrenamiento en dos ejes verticales.
- En ambos tipos de gráficas los ejes verticales deben tener la escala ajustada para aprovechar la superficie de la gráfica y apreciar mejor las diferencias. Por ejemplo, si el valor mínimo de accuracy en la serie a plotear es 42% y el mayor 46%, el eje vertical de accuracy podría empezar en 40% y terminar en 48% o 50% (pero no empezar en 0% y terminar 100%).

¿Dónde incluir las gráficas?

- Las de evolución del entrenamiento: en MLP2 (para mostrar dónde se produce el sobreentrenamiento, y para ver si los distintos EarlyStopping parecen estar parando el entrenamiento en el momento adecuado) y en cualquier tarea en la que veas algo interesante que comentar (p.ej. si en una tarea detectaste que el EarlyStopping necesitaba un reajuste, o si dos modelos muestran líneas de evolución muy diferentes y se puede deducir algo de ello).
- Las de barras para comparar test_accuracy y tiempo de entrenamiento: en todas las tareas en las que compares distintos modelos para escoger uno mejor.
- Matriz de confusión: al menos una vez en alguna tarea, y en cualquier tarea en la que veas que dos modelos producen matrices suficientemente diferentes y hay algo interesante que comentar.

¿Cuándo reajustar los parámetros en las tareas CNN*?

- Se pueden empezar a reajustar desde la tarea CNN1, no hace falta esperar a la CNN3.
- Solo es necesario comentar los resultados de esas pruebas, no es necesario documentarlas en detalle como se hacía en las tareas MLP*, aunque puedes documentar lo que te parezca más interesante (por ejemplo, si ves que un parámetro tiene ajustes óptimos muy diferentes entre MLP y CNN, y puedes comentar algo sobre a qué se deberá).

¿Límite de filtros y neuronas por capa en CNN3?

- La suma de los filtros de las capas Conv2D antes de la primera capa de Pooling2D tiene que ser menor o igual que 32.
- La suma de los filtros de las capas Conv2D después de la primera capa de Pooling2D tiene que ser menor o igual que 128.
- La suma de las neuronas de las capas Dense debe ser menor o igual que 1000.

¡Todo el código en un único archivo python!

- Todo el código necesario para producir los experimentos y las gráficas que se comenten en la documentación debe estar en un único archivo python, con todo el código ejecutable al final del archivo (después de “if __name__”).
- Si quieras mostrar código que no se utiliza para producir documentación pero fue interesante en algún punto de tu experimentación puedes incluirlo en otro archivo o en el mismo.