|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelo** | **Accuracy** | **F1-Score** | **Recall** | **Precisión** |
| modelo\_1capa\_Adam\_lr1e-3\_rand | 0,821656 | 0,770492 | 0,734375 | 0,810345 |
| modelo\_5capa\_Adam\_lr1e-3\_rand | 0,770701 | 0,723077 | 0,734375 | 0,712121 |
| modelo\_8capa\_Adam\_lr1e-3\_rand | 0,802548 | 0,730435 | 0,65625 | 0,823529 |
| modelo\_100capa\_Adam\_lr1e-3\_rand | 0,592357 | 0 | 0 | 0 |
|  |  |  |  |  |

**Análisis de los modelos y conclusiones**

**Los datos muestran el rendimiento de diferentes arquitecturas de redes neuronales entrenadas con el optimizador Adam y una tasa de aprendizaje de 1e-3. Aquí están las observaciones clave:**

1. **Rendimiento en función del número de capas:**
   * **modelo\_1capa\_Adam\_lr1e-3\_rand tiene una accuracy de 82.16%, con un F1-Score de 0.77, lo que indica un buen equilibrio entre precisión y recall.**
   * **modelo\_5capa\_Adam\_lr1e-3\_rand tiene una accuracy de 77.07%, con un F1-Score de 0.72, mostrando un rendimiento ligeramente inferior al modelo de 1 capa.**
   * **modelo\_8capa\_Adam\_lr1e-3\_rand mejora la precisión (82.54%) y tiene un F1-Score de 0.73, aunque su recall es más bajo (0.656), lo que indica que puede estar perdiendo algunas instancias positivas.**
   * **modelo\_100capa\_Adam\_lr1e-3\_rand tiene un rendimiento deficiente, con una accuracy de 59.23% y un F1-Score de 0, lo que sugiere que el modelo no está clasificando correctamente.**
2. **Impacto del número excesivo de capas:**
   * **Los modelos con 100 capas muestran que aumentar la profundidad no siempre mejora el rendimiento. Probablemente, estos modelos sufren de:**
     + **Sobreajuste extremo, lo que impide generalizar a datos de prueba.**
     + **Degradación del gradiente, dificultando el entrenamiento adecuado.**
     + **Explosión o desaparición del gradiente, lo que impide que las capas profundas aprendan correctamente.**
3. **Modelo con mejor rendimiento:**
   * **El modelo de 1 capa tiene el mejor equilibrio entre precisión (81.03%), recall (73.43%) y F1-Score (0.77).**
   * **Aunque el modelo de 8 capas tiene una precisión ligeramente superior, su recall es más bajo, lo que indica que no está capturando correctamente todas las instancias positivas.**

**Conclusión general**

**El modelo de 1 capa es el más eficiente en este conjunto de experimentos, proporcionando un buen equilibrio entre precisión y recall, sin sufrir los problemas observados en los modelos excesivamente profundos. Aumentar demasiado la cantidad de capas no garantiza un mejor rendimiento y puede llevar a problemas de entrenamiento.**

**Si quieres optimizar aún más el modelo, podríamos probar:**

* **Reducir la tasa de aprendizaje para modelos más profundos.**
* **Uso de regularización como Dropout o L2 para evitar sobreajuste.**
* **Otra inicialización de pesos, como Xavier o He, para mejorar la estabilidad en redes profundas.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Trat.** | **Optimizador** | **Tasa de aprendizaje** | **Pesos iniciales** | **Épocas** | **Exactitud entrenamiento** | **Exactitud validación** | **Exactitud prueba** |
| 1 | Adam | 0,001 | random | 100 | 0,79617 | 0,815287 | 0,821656 |
| 2 | Adam | 0,001 | random | 100 | 0,842681 | 0,840764 | 0,770701 |
| 3 | Adam | 0,001 | random | 100 | 0,835841 | 0,828025 | 0,802548 |
| 4 | Adam | 0,001 | random | 100 | 0,590971 | 0,592357 | 0,592357 |

Los datos muestran el rendimiento de distintos tratamientos de un modelo de aprendizaje automático utilizando el **optimizador Adam** con una **tasa de aprendizaje de 0.001** y **pesos iniciales aleatorios**. Aquí están las observaciones clave:

1. **Exactitud en entrenamiento vs. validación vs. prueba**:
   * **Tratamiento 1** tiene una **exactitud de prueba de 82.16%**, lo que indica un buen rendimiento general.
   * **Tratamiento 2** muestra una **exactitud de entrenamiento más alta (84.26%)**, pero su rendimiento en prueba cae a **77.07%**, lo que podría sugerir un leve sobreajuste.
   * **Tratamiento 3** tiene una **exactitud de prueba de 80.25%**, con valores de entrenamiento y validación relativamente cercanos, lo que indica estabilidad.
   * **Tratamiento 4** tiene un rendimiento significativamente inferior (**59.23% en prueba**), lo que sugiere que el modelo no está aprendiendo correctamente.
2. **Posible sobreajuste en algunos tratamientos**:
   * El **Tratamiento 2** tiene una alta exactitud en entrenamiento (**84.26%**) pero una caída en prueba (**77.07%**), lo que podría indicar que el modelo está memorizando los datos de entrenamiento en lugar de generalizar bien.
   * El **Tratamiento 3** tiene una menor diferencia entre entrenamiento y prueba, lo que sugiere un mejor equilibrio.
3. **Tratamiento con mejor rendimiento**:
   * **Tratamiento 1** parece ser el más **consistente**, con una **exactitud de prueba de 82.16%**, lo que indica que generaliza mejor.
   * **Tratamiento 3** también es una opción viable, con una **exactitud de prueba de 80.25%** y una menor diferencia entre entrenamiento y prueba.

**Conclusión general**

El **Tratamiento 1** es el más eficiente en este conjunto de experimentos, proporcionando un buen equilibrio entre **exactitud en entrenamiento, validación y prueba**. El **Tratamiento 2** muestra signos de sobreajuste, mientras que el **Tratamiento 4** tiene un rendimiento deficiente.

Para mejorar el rendimiento del modelo, podríamos probar:

* **Reducir la tasa de aprendizaje** para mejorar la estabilidad.
* **Aplicar regularización** como Dropout o L2 para evitar sobreajuste.
* **Ajustar el número de épocas** para evitar que el modelo aprenda demasiado los datos de entrenamiento.