DEEPSPEECH

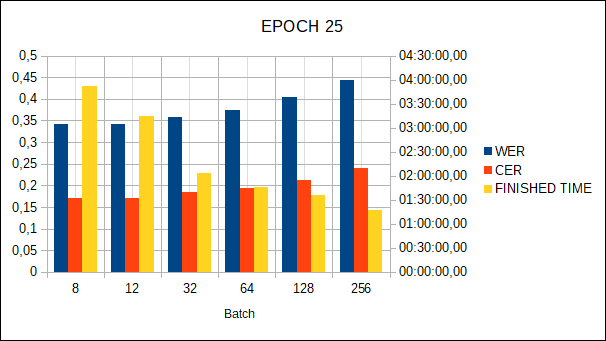
# INTRODUCCIÓN

# BATCH

## Primeros pasos

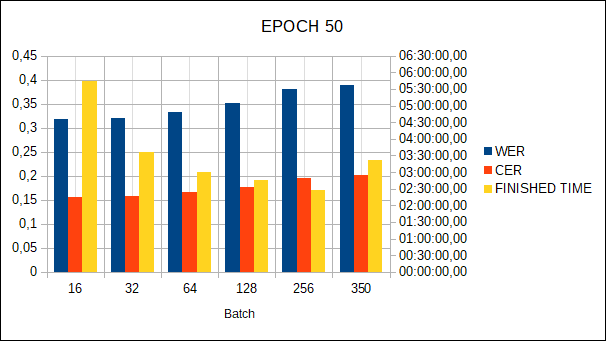
Después de hacer varios cálculos con el deepspeech he reunido los resultados en un gráfico. Únicamente he cambiado el parámetro BATCH para ver cuál es el mejor para el modelo. Se puede ver que cuanto mayor sea el Batch, peor es el resultado y menos tiempo necesita para realizar el proceso. Pero si analizamos los procesos independientemente (en cada hoja de Excel esta una gráfica) vemos que desde el batch 64 hacia arriba el modelo no converge suficientemente y no se pueden valorar los resultados. Por eso he repetido el proceso, pero con 50 iteraciones. Decir también que cuando el BATCH es 512 el modelo da un error de memoria. Por eso nos hemos parado en 256.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BATCH** | WER | CER | FINISHED TIME | LOSS |
| 8 | 0,341365 | 0,171572 | 03:52:14,23 | 43,253345 |
| 12 | 0,3411 | 0,171097 | 03:14:17,15 | 43,396267 |
| 32 | 0,357695 | 0,183656 | 02:03:49,10 | 45,415836 |
| 64 | 0,375015 | 0,193967 | 01:46:02,27 | 48,720901 |
| 128 | 0,403776 | 0,212446 | 01:36:19,14 | 51,855141 |
| 256 | 0,443143 | 0,239326 | 01:17:10,10 | 56,358032 |



Después de hacer el mismo experimento podemos sacar varias conclusiones. En comparación con la anterior, en esta el tiempo no se acorta siempre que agrandamos el BATCH.. De aquí podemos deducir que el BATCH óptimo sera entre 32, 64, 128 o 256. Con estas tres no podemos sacar nada en clave porque, aunque el resultado se empeora con las 50 iteraciones, el tiempo va bajando. Si vemos las gráficas de cada modelo, todavía no podemos decir que se ha convergido. La siguiente prueba será calcular los modelos comentados con el mismo tiempo de proceso. Aunque con si cogemos 512 como tamaño del batch nos da un error de memoria, hemos probado con el tamaño 350, que es inferior al error y superior al 256. Con este tamaño no nos da un error de memoria, pero necesita más tiempo para procesar y encima nos da peor resultado. Por eso hemos establecido el tamaño 256 como limite superior.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BATCH** | WER | CER | FINISHED TIME | LOSS |
| 16 | 0,318112 | 0,156406 | 05:45:11,06 | 40,16819 |
| 32 | 0,320111 | 0,157494 | 03:36:01,57 | 40,77356 |
| 64 | 0,333106 | 0,166037 | 03:00:19,23 | 42,939884 |
| 128 | 0,35094 | 0,177093 | 02:44:50,87 | 45,171425 |
| 256 | 0,381122 | 0,196271 | 02:27:50,60 | 48,785892 |
| 350 | 0,389535 | 0,202139 | 03:22:16,89 | 49,840111 |



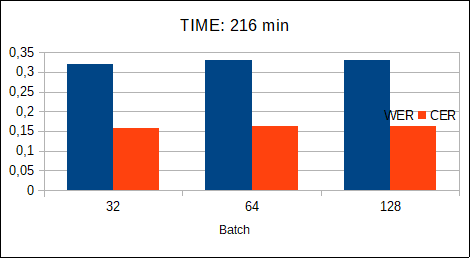
## Segundo experimento

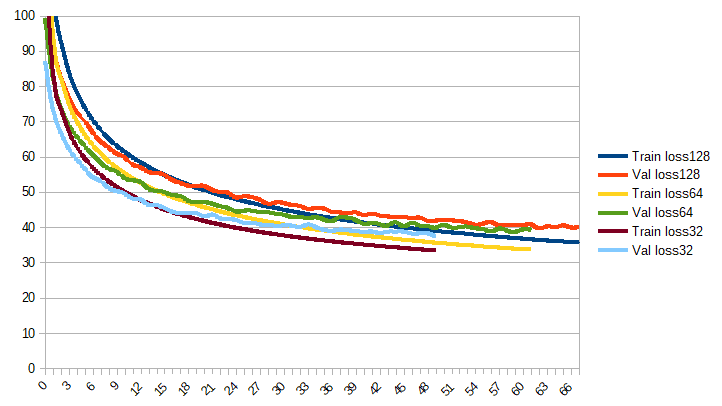
Ahora calcularemos los errores variando el Batch pero estableciendo un mismo tiempo de proceso.

### 330 minutos

En este experimento hemos realizados pruebas para ver los resultados siempre con un tiempo establecido, en este caso de 330 minutos. Vemos que en los tres casos los resultados son parecidos, aunque pensamos que con el tamaño 128 no se ha convergido lo suficiente. Por eso hacemos otro experimento pero en este caso el tiempo es de 7 horas y 30 minutos, y encima hemos añadido el tamaño 256.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BATCH** | WER | CER | FINISHED TIME | LOSS |
| 32 | 0,320111 | 0,157494 | 03:36:01,57 | 40,77356 |
| 64 | 0,329324 | 0,163408 | 03:47:26,77 | 41,993813 |
| 128 | 0,330473 | 0,163705 | 03:37:19,23 | 42,524307 |



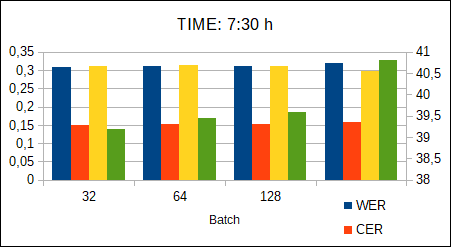


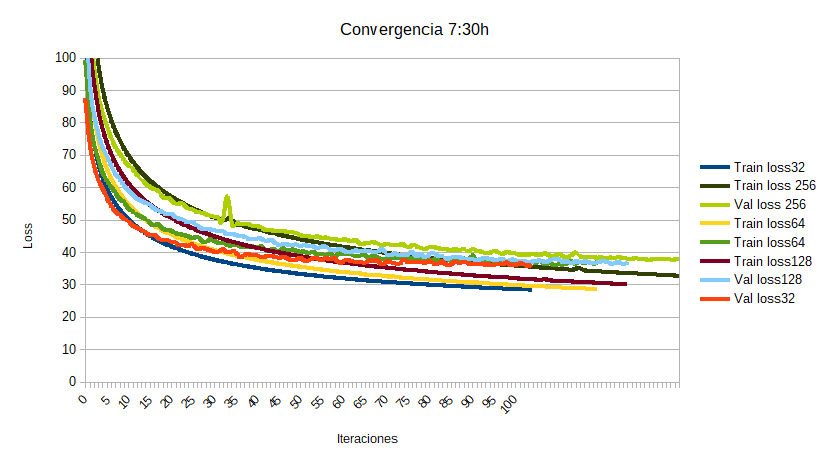
No converge. Por eso no muestra los resultados bien, más largo.

### 450 minutos

En este ejemplo vemos que independientemente el tamaño de batch que definimos, los resultados son casi iguales. Por eso en nuestro caso escogeríamos el batch más pequeño porque ocupa menos espacio. Aun así, hay otra variable a estudiar. Hemos hecho los experimentos con todos los parámetros establecidos y lo único que hemos cambiado ha sido el tamaño Batch. Pero si cambiamos el Learning Rate, cambiarían los resultados?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BATCH** | WER | CER | FINISHED TIME | LOSS |
| 32 | 0,308384 | 0,150217 | 07:29:08,47 | 39,189232 |
| 64 | 0,311761 | 0,152518 | 07:32:15,64 | 39,453438 |
| 128 | 0,312237 | 0,1524231 | 07:30:00,00 | 39,596123 |
| 256 | 0,319157 | 0,156645 | 07:06:46,35 | 40,806568 |





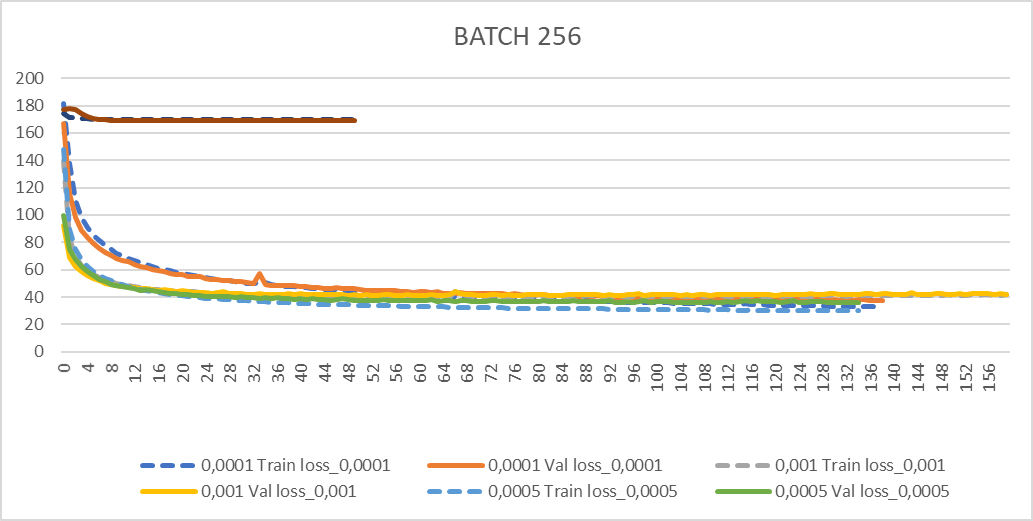
# LEARNING RATE

**¿Qué es Learning Rate?**

En el aprendizaje automático y la estadística, la tasa de aprendizaje es un parámetro de ajuste en un algoritmo de optimización que determina el tamaño del paso en cada iteración mientras se mueve hacia una función de pérdida mínima.

Por eso tenemos que probar con cada tamaño establecido de batch, si alteramos el Learning Rate, los resultados mejoran o empeoran.

En el primer caso, hemos aumentado y disminuido 10 veces el LR establecido que era 0,0001. Esto es, hemos hecho tres pruebas con diferentes tamaños. En la grafica podemos ver que si aumentamos el parámetro el modelo se convierte inestable. En cambio, si disminuimos, el modelo tarda más en convergirse y no mejora el resultado obtenido.



Analizando los experimentos podemos deducir que con los tamaños 32 y 64, el LR óptimo es el 0,0001 pero con 128 y 256 hay diferentes conclusiones. Vemos que en el caso de 128 aumentando el parámetro por 10, no mejora el resultado, pero aumentándolo por 5, los resultados son muy parecidos o mejores. En el caso de 256, si aumentamos por 10 en las primeras iteraciones hay una mejoría, pero según asa el tiempo los resultados tienden a igualarse.