**Pembelajaran Mesin Mendalam - Neural Networks from scratch**

|  |  |  |
| --- | --- | --- |
| **Nama** | Audi Husen Mustofa | 568878 |

https://github.com/audihus/Neural-Network-from-Scratch

**Eksperimen 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Mode** | **Weight Initialization** | **Activation Function** | **Learning Rate** | **Train MAE** | **Val Mae** |
| 1 | Batch Gradient Descent | Glorot (Xavier) Initialization | Relu | 0.1 | 1.033.837 | 930.341 |
| TanH | 913.359 | 860.277 |
| Leaky Relu | 1.291.562 | 1.078.813 |
| 2 | Stochastic Gradient Descent | Glorot (Xavier) Initialization | Relu | 0.1 | 613.015 | 598.343 |
| TanH | 722.565 | 671.555 |
| Leaky Relu | 603.178 | 752.198 |
| 3 | Mini Batch Gradient Descent | Glorot (Xavier) Initialization | Relu | 0.1 | 873.233 | 880.281 |
| TanH | 721.924 | 685.725 |
| Leaky Relu | 861.622 | 825.594 |

Berdasarkan tabel eksperimen di atas:  
**Untuk Batch Gradient descent**, fungsi aktivasi yang menghasilkan error paling rendah adalah TanH dengan train error 913.359, disusul Relu dengan skor 1.033.837, dan terakhir adalah aktivasi fungsi Leaky Relu dngan skor 1.291.562. dan untuk validasinya pun tanH cenderung lebih rendah errornya. Ini karena weight initialization Xavier cenderung lebih cocok dengan fungsi aktivasi TanH daripada Relu dan Leaky Relu sehingga error dari aktivasi fungsi TanH lebih rendah

**Untuk Stochastic Gradient Descent,** fungsi aktivasi yang menghasilkan error terendah adalah Leaky Relu untuk train error dan Relu untuk val error. Selain itu jika diperhatikan terjadi overfitting pada fungsi aktivasi Leaky Relu.

**Untuk Mini Batch Gradient Descent,** fungsi aktivasi yang menghasilkan error terendah untuk train adalah TanH sama seperti pada Batch Gradient Descent. Dan untuk val errornya didapatkan error terendah dengan fungsi aktivasi TanH juga.

**Eksperimen 2**

**Learning Rate diubah menjadi 0.01**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Mode** | **Weight Initialization** | **Activation Function** | **Learning Rate** | **Train MAE** | **Val Mae** |
| 1 | Batch Gradient Descent | Glorot (Xavier) Initialization | Relu | **0.01** | 1.957.086 | 1.823.650 |
| TanH | 2.047.321 | 1.759.033 |
| Leaky Relu | 1.483.390 | 1.232.794 |
| 2 | Stochastic Gradient Descent | Glorot (Xavier) Initialization | Relu | **0.01** | 708.216 | 736.924 |
| TanH | 671.463 | 711.732 |
| Leaky Relu | 684.312 | 677.231 |
| 3 | Mini Batch Gradient Descent | Glorot (Xavier) Initialization | Relu | **0.01** | 934.045 | 928.443 |
| TanH | 864.580 | 839.258 |
| Leaky Relu | 912.256 | 922.116 |

Setelah learning rate diubah menjadi 0.01, pada Batch Gradient descent nilai errornya secara keseluruhan cenderung meningkat signifikan. Pada Batch Gradient Descent dan Stochastic Gradient descent nilai error terendah dihasilkan dari aktivasi fungsi Leaky relu, tetapi pada Mini Batch Gradient Descent nilai error terendah dihasilkan dari fungsi aktivasi TanH.

**Eksperimen 3**

Mengubah learning rate menjadi 0.001 dan weight initialization he (Kaiming initialization)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Mode** | **Weight Initialization** | **Activation Function** | **Learning Rate** | **Train MAE** | **Val Mae** |
| 1 | Batch Gradient Descent | He (Kaiming initialization) | Relu | **0.001** | 2.308.078 | 2.448.487 |
| TanH | 3.330.511 | 3.523.753 |
| Leaky Relu | 2.292.137 | 2.504.274 |
| 2 | Stochastic Gradient Descent | He (Kaiming initialization) | Relu | **0.001** | 800.370 | 804.579 |
| TanH | 924.365 | 861.212 |
| Leaky Relu | 809.135 | 801.096 |
| 3 | Mini Batch Gradient Descent | He (Kaiming initialization) | Relu | **0.001** | 1.921.969 | 2.026.495 |
| TanH | 2.198.623 | 2.201.966 |
| Leaky Relu | 1.846.773 | 2.021.623 |

Ketika menggunakan weight initialization he (Kaiming initialization) error terendah tidak lagi dipegang oleh fungsi aktivasi TanH, ini karena he lebih cocok untuk fungsi aktivasi Relu dan Leaky Relu.