1. **Changing Learning rate**

Training iteration = 10

n\_input = 3

n\_hidden = 512

Optimizer = RMSpropOptimizer

Table: Accuracy

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Learning  Rate | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 0.001 | 19.0 | 25.2 | 38.0 | 47.1 | 57.8 | 61.3 | 69.2 | 69.7 | 72.4 | 76.2 |
| 0.01 | 29.3 | 41.8 | 45.3 | 50.0 | 53.8 | 62.1 | 59.0 | 60.1 | 60.5 | 60.7 |
| 0.1 | 5.00 | 11.8 | 18.9 | 13.4 | 14.4 | 18.7 | 19.5 | 21.7 | 21.3 | 19.4 |
| 1.0 | 4.40 | 5.10 | 7.6 | 9.1 | 7.1 | 9.2 | 8.2 | 8.5 | 8.7 | 9.7 |

Table: Loss

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Learning  Rate | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 0.001 | 4.51 | 2.99 | 2.11 | 1.83 | 1.47 | 1.35 | 1.10 | 1.10 | 1.05 | 0.879 |
| 0.01 | 4.10 | 2.41 | 2.22 | 2.12 | 1.88 | 1.45 | 1.68 | 1.56 | 1.75 | 1.83 |
| 0.1 | 11.1 | 9.03 | 8.31 | 9.96 | 8.91 | 7.43 | 7.32 | 7.31 | 8.00 | 7.98 |
| 1.0 | 110.8 | 96.7 | 86.0 | 79.1 | 72.0 | 75.3 | 76.4 | 77.9 | 74.3 | 78.0 |

Conclusion: A better result is obtained at a lower learning rate.

1. **Changing optimizer**

Training iteration = 10

n\_input = 3

n\_hidden = 512

Learning rate = 0.001

Table: Accuracy

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Optimizer | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| RMSprop  Optimizer | 19.0 | 25.2 | 38.0 | 47.1 | 57.8 | 61.3 | 69.2 | 69.7 | 72.4 | 76.2 |
| Adam  Optimizer | 22.0 | 30.4 | 48.2 | 56.4 | 68.4 | 70.8 | 74.5 | 76.8 | 76.6 | 78.9 |
| RMSprop  Optimizer | 13.1 | 30.5 | 33.9 | 48.0 | 57.5 | 64.5 | 71.0 | 69.3 | 75.9 | 75.4 |
| Gradient  Decent  Optimizer | 8.3 | 19.9 | 19.2 | 24.5 | 29.4 | 37.13 | 34.0 | 33.8 | 39.7 | 39.7 |

Table: Loss

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Optimizer | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| RMSprop  Optimizer | 4.51 | 2.99 | 2.11 | 1.83 | 1.47 | 1.35 | 1.10 | 1.10 | 1.05 | 0.879 |
| Adam  Optimizer | 4.31 | 2.44 | 1.59 | 1.27 | 0.94 | 0.79 | 0.70 | 0.62 | 0.58 | 0.55 |
| Rmsprop  Optimizer | 4.87 | 2.61 | 2.25 | 1.77 | 1.53 | 1.33 | 1.08 | 1.16 | 1.00 | 0.93 |
| Gradient  Decent  Optimizer | 4.92 | 3.08 | 3.10 | 2.92 | 2.51 | 2.11 | 2.23 | 2.27 | 1.98 | 1.99 |

Conclusion: The best result was obtained when using AdamOptimizer

1. **Changing n\_input**

Training iteration = 10

Learning rate = 0.001

n\_hidden = 512

Optimizer = AdamOptimizer

Table: Accuracy

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n\_input | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 2 | 7.3 | 16.9 | 24.7 | 30.9 | 36.1 | 40.9 | 45.0 | 44.5 | 47.5 | 54.4 |
| 3 | 22.0 | 30.4 | 48.2 | 56.4 | 68.4 | 70.8 | 74.5 | 76.8 | 76.6 | 78.9 |
| 4 | 16.0 | 46.5 | 63.3 | 75.3 | 79.6 | 84.7 | 86.5 | 85.4 | 85.3 | 88.0 |
| 5 | 19.0 | 48.7 | 70.5 | 88.7 | 90.2 | 91.2 | 92.1 | 95.2 | 88.5 | 94.7 |

Table: Loss

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n\_input | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 2 | 4.37 | 2.69 | 2.38 | 1.97 | 1.79 | 1.62 | 1.47 | 1.43 | 1.36 | 1.16 |
| 3 | 4.31 | 2.44 | 1.59 | 1.27 | 0.94 | 0.79 | 0.70 | 0.62 | 0.58 | 0.55 |
| 4 | 4.68 | 1.98 | 1.19 | 0.80 | 0.664 | 0.483 | 0.41 | 0.49 | 0.41 | 0.349 |
| 5 | 4.62 | 1.72 | 0.989 | 0.427 | 0.356 | 0.351 | 0.247 | 0.149 | 0.433 | 0.150 |

Conclusion: A better result is obtained at higher values of n\_input

1. **Changing n\_hidden**

Training iteration = 10

Learning rate = 0.001

n\_input = 5

Optimizer = AdamOptimizer

Table: Accuracy

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n\_hidden | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 500 | 14.9 | 44.9 | 77.0 | 88.2 | 92.7 | 91.8 | 89.6 | 95.3 | 93.9 | 94.2 |
| 512 | 19.0 | 48.7 | 70.5 | 88.7 | 90.2 | 91.2 | 92.1 | 95.2 | 88.5 | 94.7 |
| 550 | 15.6 | 44.9 | 75.8 | 87.7 | 91.4 | 91.4 | 94.5 | 92.7 | 89.8 | 93.1 |
| 600 | 12.8 | 47.0 | 72.8 | 86.2 | 89.50 | 91.9 | 89.6 | 93.2 | 94.4 | 92.4 |
| 650 | 15.0 | 51.0 | 77.8 | 84.3 | 93.0 | 86.6 | 91.0 | 93.0 | 93.5 | 88.6 |

Table: Loss

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n\_hidden | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| 500 | 4.71 | 1.89 | 0.774 | 0.448 | 0.274 | 0.329 | 0.416 | 0.135 | 0.169 | 0.140 |
| 512 | 4.62 | 1.72 | 0.989 | 0.427 | 0.356 | 0.351 | 0.247 | 0.149 | 0.433 | 0.150 |
| 550 | 4.83 | 1.87 | 0.820 | 0.467 | 0.342 | 0.295 | 0.198 | 0.256 | 0.371 | 0.169 |
| 600 | 5.13 | 1.94 | 0.877 | 0.511 | 0.388 | 0.305 | 0.402 | 0.201 | 0.163 | 0.221 |
| 650 | 5.14 | 1.78 | 0756 | 0.549 | 0.268 | 0.522 | 0.30 | 0.171 | 0.170 | 0.398 |

Conclusion: A better result is obtained at lower values of n\_hidden

1. **Changing number of stack**

Training iteration = 10

Learning rate = 0.001

n\_hidden = 500

n\_input = 5

Optimizer = AdamOptimizer

Table : Accuracy

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LSTM | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| One stack | 14.9 | 44.9 | 77.0 | 88.2 | 92.7 | 91.8 | 89.6 | 95.3 | 93.9 | 94.2 |
| Two Stack | 23.5 | 57.9 | 82.4 | 88.1 | 93.1 | 88.5 | 92.8 | 92.7 | 90.7 | 93.7 |

Table: Loss

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LSTM | Ite  1 | Ite  2 | Ite  3 | Ite  4 | Ite  5 | Ite  6 | Ite  7 | Ite  8 | Ite  9 | Ite  10 |
| One stack | 4.71 | 1.89 | 0.774 | 0.448 | 0.274 | 0.329 | 0.416 | 0.135 | 0.169 | 0.140 |
| Two Stack | 3.70 | 1.59 | 0.610 | 0.396 | 0.229 | 0.340 | 0.242 | 0.167 | 0.356 | 0.202 |

Conclusion: A better result was obtained when using one stack LSTM