Zgodnie z regułą Hebba zmiana wagi neuronu odbywa się proporcjonalnie do iloczynu jego sygnału wejściowego i wyjściowego:

η – współczynnik uczenia

=sygnały wejściowy i wyjściowy neuronu

Regułę Hebba możemy stosować zarówno do nauki w trybie z nauczycielem jak i bez nauczyciela. W przypadku uczenia z nauczycielem wartość sygnału wyjściowego zastępuje się wartością oczekiwaną dla tego neuronu.

Ponieważ w każdym cyklu uczącym następuje proces sumowania aktualnych wartości wag i skończonego przyrostu :

W wyniku reguły Hebba wagi mogą przyjmować wartości dowolnie duże. Aby poprawić stabilność uczenia się przy aktualizacji wag przyjmuje się wartość zmniejszoną o współczynnik zapominania γ:

|  |  |
| --- | --- |
| 0 0 0 1 0 0 0  0 0 1 0 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0 |  |
| 0 1 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 0 0 |  |
| 0 0 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 1 0  0 0 1 1 1 0 0 |  |
| 0 1 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 0 0 |  |
| 0 1 1 1 1 1 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 1 1 1 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 1 1 1 1 0 |  |
| 0 1 1 1 1 1 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 1 1 1 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0 |  |
| 0 0 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 1 1 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 0 1 1 1 0 0 |  |
| 0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0 |  |
| 0 0 1 1 1 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 1 1 1 0 0 |  |
| 0 0 0 0 0 1 0  0 0 0 0 0 1 0  0 0 0 0 0 1 0  0 0 0 0 0 1 0  0 0 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 0 1 1 1 0 0 |  |
| 0 1 0 0 0 1 0  0 1 0 0 1 0 0  0 1 0 1 0 0 0  0 1 1 0 0 0 0  0 1 1 0 0 0 0  0 1 0 1 0 0 0  0 1 0 0 1 0 0  0 1 0 0 0 1 0 |  |
| 0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 1 1 1 1 0 |  |
| 0 1 0 0 0 1 0  0 1 1 0 1 1 0  0 1 0 1 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0 |  |
| 0 1 0 0 0 1 0  0 1 1 0 0 1 0  0 1 1 0 0 1 0  0 1 0 1 0 1 0  0 1 0 1 0 1 0  0 1 0 0 1 1 0  0 1 0 0 1 1 0  0 1 0 0 0 1 0 |  |

|  |  |
| --- | --- |
| 0 0 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 0 1 1 1 0 0 |  |
| 0 1 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0  0 1 0 0 0 0 0 |  |
| 0 0 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 0 1 0 1 0  0 1 0 0 1 1 0  0 0 1 1 1 0 0 |  |
| 0 1 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 1 1 1 1 0 0  0 1 0 1 0 0 0  0 1 0 0 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0 |  |
| 0 0 1 1 1 0 0  0 1 0 0 0 1 0  0 1 0 0 0 1 0  0 0 1 1 0 0 0  0 0 0 0 1 0 0  0 0 0 0 0 1 0  0 1 0 0 0 1 0  0 0 1 1 1 0 0 |  |
| 0 1 1 1 1 1 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0  0 0 0 1 0 0 0 |  |

Dane wejściowe to 20 wielkich liter alfabetu utworzonych na matrycy o rozmiarach 7x8.

|  |  |  |
| --- | --- | --- |
| litera | z zapominaniem | |
| Przed | Po |
| A | -0.4023 | 0.9548 |
| B | -0.7798 | -0.6892 |
| C | 0.8438 | -0.1254 |
| D | -0.4526 | -0.9672 |
| E | 0.9332 | -0.7519 |
| F | -0.2745 | 0.993 |
| G | 0.9615 | 0.6063 |
| H | 0.0487 | 0.8884 |
| I | 0.1861 | -0.0925 |
| J | -0.1898 | 0.7185 |
| K | -0.9995 | 0.9596 |
| L | 0.1082 | 0.1625 |
| M | 0.9443 | 0.8505 |
| N | -0.4511 | -0.6978 |
| O | 0.5563 | -0.9129 |
| P | -0.7055 | 0.9597 |
| Q | 0.6326 | -0.8966 |
| R | 0.5583 | -0.4178 |
| S | 0.1702 | -0.7683 |
| T | -0.0155 | 0.3964 |

Widać, że podobne znaki są grupowanie razem, przykładowo dla liter B i E wyjścia neuronu są bardzo zbliżone. Podobna sytuacja jest w przypadku liter O i Q.

|  |  |  |
| --- | --- | --- |
| litera | bez zapominania | |
| Przed | Po |
| A | -0.5388 | -0.488 |
| B | -0.696 | 0.9169 |
| C | -0.7328 | 0.7585 |
| D | -0.6894 | 0.1293 |
| E | -0.9897 | 0.8805 |
| F | -0.937 | -0.8528 |
| G | 0.7179 | 0.991 |
| H | -0.6477 | 0.98 |
| I | 0.3041 | 0.6434 |
| J | 0.3614 | 0.351 |
| K | -0.9641 | 0.068 |
| L | -0.8363 | -0.3093 |
| M | -0.1917 | 0.34 |
| N | -0.9256 | 0.212 |
| O | 0.2701 | 0.8503 |
| P | 0.9412 | 0.8402 |
| Q | 0.8523 | 0.8854 |
| R | -0.8144 | 0.8957 |
| S | 0.6503 | 0.8885 |
| T | 0.6412 | 0.5509 |

W przypadku uczenia bez współczynnika zapominania sytuacja wyglądała bardzo podobnie. Wyjścia podobnych liter są zbliżone do siebie.

Przetestowanie nauczonej sieci innym zestawem liter, które są delikatnie zniekształcone w stosunku do zestawu uczącego:

Dla 100 000 iteracji:

|  |  |
| --- | --- |
| Litery | |
| normalne | Zniekształcone |
| -0.0399 | -0.9155 |
| 0.5916 | 0.9006 |
| -0.5923 | -0.6784 |
| 0.9423 | 0.9439 |
| -0.2019 | -0.3261 |
| 0.447 | -0.8355 |
| 0.9364 | 0.1118 |
| 0.3503 | 0.4588 |
| 0.4286 | -0.245 |
| -0.5806 | -0.785 |
| 0.9333 | 0.5084 |
| -0.8036 | -0.7305 |
| -0.9384 | -0.2785 |
| -0.8284 | -0.7005 |
| -0.3019 | 0.9484 |
| -0.8891 | 0.9996 |
| -0.9913 | 0.2009 |
| 1 | 0.9814 |
| -0.9412 | -0.9707 |
| -0.9424 | -0.5008 |

Dla 5000000 iteracji:

|  |  |
| --- | --- |
| Litery | |
| normalne | Zniekształcone |
| 0.9548 | -0.3038 |
| -0.6892 | -0.4673 |
| -0.1254 | -0.685 |
| -0.9672 | 0.7861 |
| -0.7519 | -0.3855 |
| 0.993 | -0.7236 |
| 0.6063 | -1 |
| 0.8884 | -0.9197 |
| -0.0925 | 0.8587 |
| 0.7185 | -0.1058 |
| 0.9596 | -0.9894 |
| 0.1625 | 0.3054 |
| 0.8505 | -0.0062 |
| -0.6978 | 0.9477 |
| -0.9129 | -0.5568 |
| 0.9597 | -0.0455 |
| -0.8966 | -0.7682 |
| -0.4178 | -0.7282 |
| -0.7683 | 0.6847 |
| 0.3964 | -0.9965 |

Dla mniejszej ilości iteracji sieć lepiej radzi sobie z rozpoznawaniem zmienionych danych niż w przypadku bardzo dużej liczby iteracji.

Test dla bardzo dużej wartości współczynnika zapominania (wartość zbliżona do 1):

|  |  |  |  |
| --- | --- | --- | --- |
| litera | z zapominaniem | | |
| Przed | Po | inne litery |
| A | -0.9994 | -0.9548 | -0.4212 |
| B | -0.5952 | -0.6892 | 0.9987 |
| C | -0.3556 | -0.1254 | 0.4295 |
| D | -0.9781 | -0.9672 | 0.8014 |
| E | -0.2812 | -0.7519 | -0.7656 |
| F | 0.038 | 0.993 | -0.5402 |
| G | -0.7737 | 0.6063 | -0.595 |
| H | 0.9035 | 0.8884 | -0.9964 |
| I | 0.765 | -0.0925 | 0.8461 |
| J | 0.6227 | 0.7185 | 0.5855 |
| K | -0.9715 | 0.9596 | -0.5487 |
| L | 0.3587 | 0.1625 | -0.9361 |
| M | -0.9842 | 0.8505 | -0.0702 |
| N | -0.7156 | -0.6978 | 0.31 |
| O | -0.9305 | -0.9129 | -0.3192 |
| P | -0.2878 | 0.9597 | -0.3262 |
| Q | 0.1249 | -0.8966 | 0.9999 |
| R | -0.208 | -0.4178 | 0.359 |
| S | -0.6936 | -0.7683 | -0.6828 |
| T | 0.9441 | 0.3964 | 0.6875 |

Nie widać aby powstały jakiekolwiek grupy po procesie uczenia, wyniki różnią się nie znacznie w stosunku do początkowych.

**WNIOSKI:**

* Stosując metodę Hebba bez nauczyciela należy pamiętać, że nie mamy kontroli nad procesem uczenia. Sieć sama uczy się grupować dane dlatego nie jesteśmy w stanie przewidzieć wyniku. Możemy próbować ingerować w wyniki inicjując własnoręcznie wagi początkowe.
* Należy uważać, aby nie doprowadzić do przetrenowania sieci ustawiając zbyt dużą ilość iteracji, ponieważ wtedy sieć będzie miała problem z rozpoznawaniem nowych znaków, będzie zbyt wyspecjalizowana w rozpoznawaniu danych testowych.
* Zbyt duża wartość współczynnika zapominania powoduje, że sieć szybko zapomina tego czego się uczy i nauka nie przynosi pożądanych efektów.