



27.11.2017r.

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Podstawy sztucznej inteligencji

Sprawozdanie numer 4
Uczenie sieci regułą Hebba

Inżynieria Obliczeniowa
Urszula Ślusarz
nr. indeksu: 286132

1. Cel ćwiczenia

Celem ćwiczenia jest poznanie działania reguły Hebba dla sieci jednowarstwowej na przykładzie grupowania liter alfabetu.

2. Wstęp teoretyczny

Reguła Hebba

Czasami metoda uczenia wg praw Hebba nazywana jest też uczeniem korelacyjnym (correlation learning) ponieważ zmierza ona do takiego dopasowania wag, aby uzyskać najlepszą korelację między sygnałami wejściowymi, a zapamiętanym w formie wartości wag „wzorcem” sygnału, na który neuron ma reagować.

Sieć ucząca się (bez nadzoru) wg reguły Hebba, osiąga na ogół dobre wyniki i całkiem samoczynnie grupuje wejściowe sygnały w „kategorie” sensownie odpowiadające klasom podobnych do siebie sygnałów wejściowych **X**. Jednak efekt ten nie jest nigdy pewny, gdyż dość istotnie zależy od początkowego stanu sieci (początkowych przypadkowo wybranych wartości wag) decydującego o tym, jak w początkowym etapie uczenia zaczną się krystalizować ośrodki przyszłych grup.

Posługując się przykładem: **sieć trenowana w celu rozpoznawania ręcznie pisanych liter** nauczy się je identyfikować, ale nie wiemy z góry, który neuron sygnalizować będzie „A”, a który „B”. Mało tego: nie mamy gwarancji, że każdej literze odpowiadać będzie tylko jeden neuron - bardziej prawdopodobne jest, że kilka neuronów nauczy się rozpoznawać jako „swoje” „A”. Gorszą lecz również prawdopodobną sytuacją jest, że któraś z liter nie będzie w ogóle rozpoznawana. Sposobem na poradzenie sobie z tą sytuacją może być stosowanie większej liczby neuronów niż ilość planowanych do rozpoznania wzorców.

Reguła Hebba posiada istotną wadę, mianowicie prowadzi do procesu rozbieżnego (wagi są przez cały czas zwiększane). Możemy zapobiec rozbieżności prostej reguły Hebba ograniczając wzrost wektora wag **W**. Dobre rozwiązanie zaproponował Oja modyfikując regułę Hebba, tak że osiągnięcie stanu stabilnego w przestrzeni wag jest zagwarantowane. Wagi modyfikujemy według wzoru:

Jeśli aktywny neuron A jest cyklicznie pobudzany przez neuron B, to staje się on jeszcze bardziej czuły na pobudzenie tego neuronu.

Konsekwencją stwierdzenia Hebba jest następująca, dwuczęściowa reguła:

1. Jeżeli neurony A i B połączone synapsą są pobudzane jednocześnie (synchronicznie) to połączenie synaptyczne je łączące jest wzmacniane.
2. Jeżeli neurony A i B połączone synapsą są pobudzane niejednocześnie (asynchronicznie) to połączenie synaptyczne je łączące podlega osłabieniu.

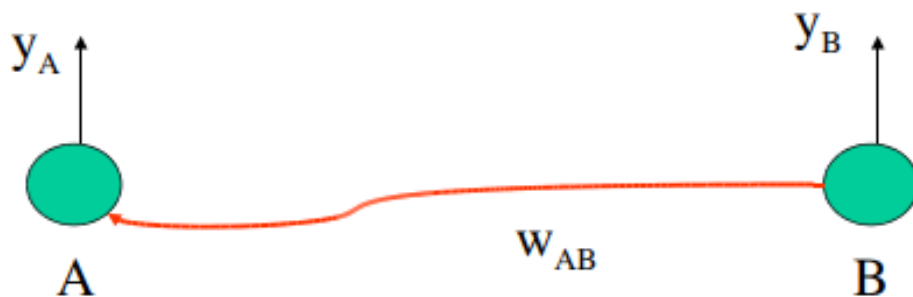
Zmiana wag połączenia pomiędzy neuronem A i B według reguły Hebba przebiega w następujący sposób:

$$w_{AB}(k+1) = w_{AB}(k) + \eta y_A(k) y_B(k)$$

gdzie:

- w_{AB} - waga połączenia synaptycznego pomiędzy neuronem A i B
- y_A - stan aktywacji neuronu A
- y_B - stan aktywacji neuronu B

$\forall \eta$ - współczynnik uczenia (dodatni)



3. Wykonane kroki scenariusza

✓ W pierwszej kolejności wygenerowałam dane uczące i testujące, które zawierają 20 dużych liter w postaci dwuwymiarowej tablicy np. 5x7 pikseli dla jednej litery.

Litery wybrane przeze mnie: **A C D E F G H K M N O P R S T U W X Y Z**

Dane uczące, poniższe tabele przedstawiają matryce 20 liter po konwersji zero-jedynkowej:

```
letterA= [0 0 1 0 0 ...
          0 1 0 1 0 ...
          0 1 0 1 0 ...
          1 0 0 0 1 ...
          1 1 1 1 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ]';
```

```
letterC= [0 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 0 ...
          1 0 0 0 0 ...
          1 0 0 0 0 ...
          1 0 0 0 1 ...
          0 1 1 1 0 ]';
```

```
letterD = [1 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 1 1 1 0 ]';
```

```

letterE = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 1 ]';

```

```

letterF = [1 1 1 1 1 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 1 1 1 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ...
           1 0 0 0 0 ]';

```

```

letterG = [0 1 1 1 0 ...
           1 0 0 0 1 ...
           1 0 0 0 0 ...
           1 0 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           0 1 1 1 0 ]';

```

```

letterH = [1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 1 1 1 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ...
           1 0 0 0 1 ]';

```

```

letterK = [1 0 0 0 1 ...
           1 0 0 1 0 ...
           1 0 1 0 0 ...
           1 1 0 0 0 ...
           1 0 1 0 0 ...
           1 0 0 1 0 ...
           1 0 0 0 1 ]';

```

```

letterM= [1 0 0 0 1 ...
          1 1 0 1 1 ...
          1 0 1 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ]';

```

```

letterN= [1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 1 0 0 1 ...
          1 0 1 0 1 ...
          1 0 0 1 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ]';

```

```

letterO= [0 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          0 1 1 1 0 ]';

```

```

letterP= [1 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 1 1 1 0 ...
          1 0 0 0 0 ...
          1 0 0 0 0 ...
          1 0 0 0 0 ]';

```

```

letterR= [1 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 1 1 1 0 ...
          1 0 1 0 0 ...
          1 0 0 1 0 ...
          1 0 0 0 1]';

```

```

letterS= [0 1 1 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 0 ...
          0 1 1 1 0 ...
          0 0 0 0 1 ...
          1 0 0 0 1 ...
          0 1 1 1 0]';

```

```

letterT= [1 1 1 1 1 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0]';

```

```

letterU= [1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          0 1 1 1 0]';

```

```

letterX= [1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 0 0 1 ...
          1 0 1 0 1 ...
          1 0 1 0 1 ...
          0 1 0 1 0]';

```

```

letterW= [1 0 0 0 1 ...
          1 0 0 0 1 ...
          0 1 0 1 0 ...
          0 0 1 0 0 ...
          0 1 0 1 0 ...
          1 0 0 0 1 ...
          1 0 0 0 1]';

```

```

letterY= [1 0 0 0 1 ...
          1 0 0 0 1 ...
          0 1 0 1 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0 ...
          0 0 1 0 0]';

```

```

letterZ= [1 1 1 1 1 ...
          0 0 0 0 1 ...
          0 0 0 1 0 ...
          0 0 1 0 0 ...
          0 1 0 0 0 ...
          1 0 0 0 0 ...
          1 1 1 1 1]';

```

Liniowym rozwinięciem tych matryc będą wektory wejściowe ciągu uczącego:

```

189 - in=[0 0 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1
190      1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1
191      1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1
192      1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1
193      0 0 0 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1
194      1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0
195      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
196      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
197      0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
198      1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1
199      1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0
200      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0
201      0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0
202      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
203      1 0 1 0 0 0 1 0 1 1 1 1 1 0 0 1 1 0 0 0
204      1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0
205      1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0
206      1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 1
207      1 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0
208      1 0 1 0 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0
209      1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0
210      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
211      0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0
212      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
213      1 0 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 0 0 0
214      1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1
215      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
216      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0
217      0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
218      1 1 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 1 0 0
219      1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0 1
220      0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1
221      0 1 1 1 0 1 0 0 0 0 1 0 0 1 1 1 0 0 1 1
222      0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1
223      1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0 1 0 1];
224      %A C D E F G H K M N O P R S T U W X Y Z

```

Ciąg uczący składa się z 20 następujących wektorów uczących :

Wektor wejściowy - 35 wartości

Wektor wyjściowy - 20 wartości

Wyjściowa (out) macierz, liczba 1 reprezentuje występowanie danej litery w odpowiedniej kolumnie:

```

226 - out=[ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
227         0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
228         0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
229         0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
230         0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
231         0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
232         0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
233         0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
234         0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
235         0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
236         0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
237         0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
238         0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
239         0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
240         0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
241         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
242         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
243         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
244         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
245         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];

```

Dane testujące:

```

248 - testujacyA=[0;1;1;1;0;1;0;0;0;1;1;0;0;0;1;1;1;1;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;];
249             %1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
250             %A C D E F G H K M N O P R S T U W X Y Z
251 - testujacyC=[0;1;1;1;0;1;0;0;0;1;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;1;0;1;1;0;];
252             %0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
253             %A C D E F G H K M N O P R S T U W X Y Z
254 - testujacyD=[1;1;1;1;0;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;1;1;0;];
255             %0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
256             %A C D E F G H K M N O P R S T U W X Y Z
257 - testujacyE=[1;1;1;1;1;1;0;0;0;0;1;0;0;0;0;1;1;1;1;0;1;0;0;0;0;1;0;0;0;0;1;1;1;1;];
258             %0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
259             %A C D E F G H K M N O P R S T U W X Y Z
260 - testujacyF=[1;1;1;1;1;1;0;0;0;0;1;0;0;0;0;1;1;1;1;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;];
261             %0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
262             %A C D E F G H K M N O P R S T U W X Y Z
263 - testujacyG=[0;1;1;1;0;1;0;0;0;1;1;0;0;0;0;1;0;1;1;1;1;0;0;0;1;1;0;0;0;1;0;1;1;0;];
264             %0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
265             %A C D E F G H K M N O P R S T U W X Y Z
266 - testujacyH=[1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;1;1;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;];
267             %0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
268             %A C D E F G H K M N O P R S T U W X Y Z
269 - testujacyK=[1;0;0;0;1;1;0;0;1;0;1;0;1;0;0;1;1;0;0;0;1;0;1;0;0;1;0;0;1;0;0;1;0;];
270             %0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
271             %A C D E F G H K M N O P R S T U W X Y Z
272 - testujacyM=[1;0;0;0;1;1;1;0;1;1;1;0;1;0;1;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;];
273             %0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
274             %A C D E F G H K M N O P R S T U W X Y Z
275 - testujacyN=[1;0;0;0;1;1;0;0;0;1;1;1;0;0;1;1;0;1;0;1;1;0;0;1;1;1;0;0;0;1;1;];
276             %0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
277             %A C D E F G H K M N O P R S T U W X Y Z

```



```

278 - testujacyO=[0;1;1;1;0;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;0;1;1;1;0;];
279 - %0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
280 - %A C D E F G H K M N O P R S T U W X Y Z
281 - testujacyP=[1;1;1;1;0;1;0;0;0;1;1;0;0;0;1;1;1;1;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;];
282 - %0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
283 - %A C D E F G H K M N O P R S T U W X Y Z
284 - testujacyR=[1;1;1;1;0;1;0;0;0;1;1;0;0;0;1;1;1;1;0;1;0;1;0;0;1;0;0;1;0;1;0;0;0;1;];
285 - %0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
286 - %A C D E F G H K M N O P R S T U W X Y Z
287 - testujacyS=[0;1;1;1;0;1;0;0;0;1;1;0;0;0;0;0;1;1;1;0;0;0;0;0;1;1;0;0;0;1;0;1;1;1;0;];
288 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
289 - %A C D E F G H K M N O P R S T U W X Y Z
290 - testujacyT=[1;1;1;1;1;0;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;0;1;0;0;0;0;0;1;0;0;0;];
291 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
292 - %A C D E F G H K M N O P R S T U W X Y Z
293 - testujacyU=[1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;1;1;0;];
294 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
295 - %A C D E F G H K M N O P R S T U W X Y Z
296 - testujacyW=[1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;0;0;1;1;0;1;0;1;1;0;1;0;1;0;1;0;];
297 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
298 - %A C D E F G H K M N O P R S T U W X Y Z
299 - testujacyX=[1;0;0;0;1;1;0;0;0;1;0;1;0;1;0;0;0;1;0;0;0;1;0;1;0;1;0;0;0;1;1;0;0;0;1;];
300 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
301 - %A C D E F G H K M N O P R S T U W X Y Z
302 - testujacyY=[1;0;0;0;1;1;0;0;0;1;0;1;0;1;0;0;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;1;0;0;];
303 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
304 - %A C D E F G H K M N O P R S T U W X Y Z
305 - testujacyZ=[1;1;1;1;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;1;0;0;0;0;1;1;1;1;1;];
306 - %0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
307 - %A C D E F G H K M N O P R S T U W X Y Z

```

Litery są utworzone na matrycy **5 x 7** czyli na **35 polach**.

I tyle wejść ma sieć, mogą zostać wypełnione tylko liczbą 0(min) albo 1(max);

```

5 - min_max=[0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
6 -          0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
7 -          0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
8 -          0 1; 0 1; 0 1;0 1;0 1];

```

✓ Kolejnym krokiem było przygotowanie (implementacja lub wykorzystanie gotowych narzędzi) jednowarstwowej sieci oraz reguły Hebba z i bez współczynnika zapomnienia.

```

6
7 - net = newff(PR,S,{'tansig','trainlm','learnh'});
8
9

```

```

56 - lp.dr = 0.5; |
57 - lp.lr = 0.9;
58
59 - dw=learnh([0],in,[0],[0],out,[0],[0],[0],[0],[0],lp,[0]);
60
61 - net.trainParam.epochs = 2500;
62 - net.trainParam.goal = 0.01;
63 - net.trainParam.lr=0.5;
64
65 - net=train(net, in, dw);

```


Funkcja newff

newff - Tworzenie wielowarstwowej jednokierunkowej sieci neuronowej, złożonej z neuronów o nieliniowych funkcjach aktywacji.

Funkcja ta tworzy wielowarstwową sieć neuronową: każda warstwa składa się z zadanej liczby neuronów o nieliniowych funkcjach aktywacji (jakkolwiek funkcje aktywacji w poszczególnych warstwach mogą mieć również postać liniową).

Wywołanie funkcji:

NET = NEWFF(PR, [S1 S2...SNL], ...{TF1 TF2...TFNL}, BTF, BLF, PF)

WEJŚCIE:

PR - macierz o wymiarach $R \times 2$, gdzie R jest liczbą wejść sieci (współrzędnych wektorów wejściowych); pierwsza kolumna zawiera minimalne wartości kolejnych współrzędnych wektorów wejściowych, druga kolumna – maksymalne wartości tych współrzędnych.

Si - liczba neuronów w i -tej warstwie sieci; liczba warstw wynosi $N1$.

TFi - nazwa funkcji aktywacji neuronów w i -tej warstwie sieci (zmienna tekstowa); domyślna = 'tansig' (tangens hiperboliczny); dopuszczalne wartości parametru TF to: 'tansig' i 'logsig' i 'purelin'.

BTF - nazwa funkcji, wykorzystywanej do treningu sieci (zmienna tekstowa); domyślnie BTF = 'trainlm' (metoda Levenberga-Marquardta)

BLF - nazwa funkcji, wykorzystywanej do wyznaczania korekcji wag sieci podczas treningu (zmienna tekstowa); domyślnie BLF = 'learngd'; dopuszczalne wartości parametru BLF to: 'learngd' (gradient prosty) i 'learngdm' (gradient prosty z momentum).

PF - funkcja wyznaczająca wartość wskaźnika jakości treningu sieci jednokierunkowej (zmienna tekstowa); domyślnie PF = 'mse' (błąd średniokwadratowy); parametr ten może oznaczać dowolną różniczkowalną funkcję błędu, np. 'msereg' (suma błędu średniokwadratowego i kwadratów wag sieci – metoda regularyzacji wag) lub 'sse' (suma kwadratów błędów).

WYJŚCIE:

NET - struktura (obiekt) zawierająca opis architektury, metod treningu, wartości liczbowe wag oraz inne parametry wielowarstwowej sieci jednokierunkowej.

tansig- tangens hiperboliczny (domyślna)

trainlm- domyślna metoda Levenberga-Marquardta

learnh- jest funkcją uczenia wagi Hebba

Krótki opis:

net.trainParam.epochs – maksymalna liczba trenowanych epok

net.trainParam.goal- cel wydajności

net.trainParam.lr- jest to wskaźnik uczenia się

[dW, LS] = learnh(W,P,Z,N,A,T,E,gW,gA,D,LP,LS) bierze kilka wejść,

W	S-by- Rwaga macierzy (lub S-by- 1stronniczości wektora)
P	R-by- Qwektory wejściowe (lub ones(1,Q))
Z	Sprzez Qważone wektory wejściowe
N	S-by- Qwektory wejściowe netto
A	S-by- Qwektory wyjściowe
T	Sdo Qwarstw docelowych wektorów
E	S-w- Qwarstwowe wektory błędów
gW	S- przez Rgradient w odniesieniu do wydajności
gA	S-by- Qwyjściowy gradient w odniesieniu do wydajności
D	S- przez - Sodległości neuronów
LP	Parametry uczenia się, brak, LP = []
LS	Stan nauki, początkowo powinien być = []

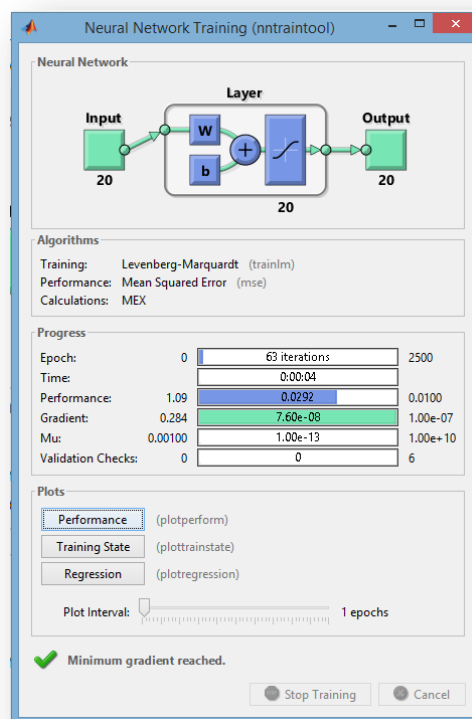
W moim przypadku należało jednak używać tylko tych parametrów, które wymagały zmiany obliczeniowej.

lp.dr- współczynnik zapominania

lp.lr- współczynnik uczenia się

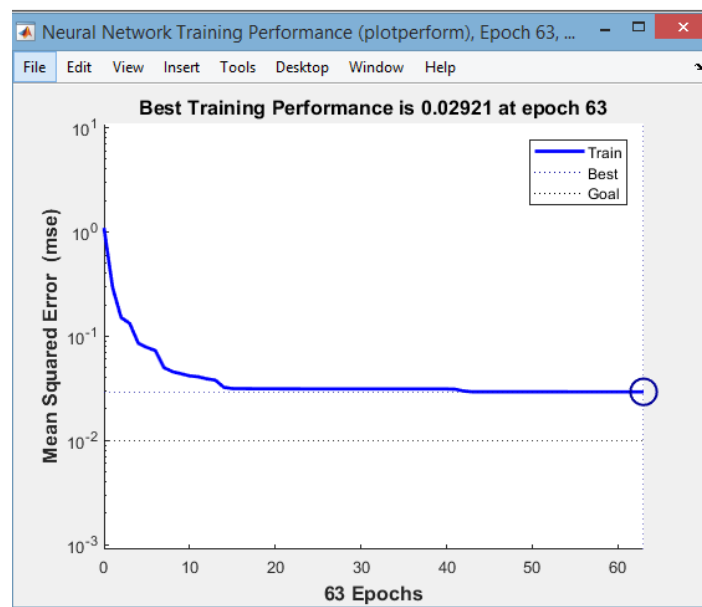
Pewnym problemem w podstawowej metodzie Hebb'a jest to, że wagi mają tendencję do przyjmowania dużych wartości, gdyż w każdym cyklu uczącym dodajemy przyrost dw. Jedną z metod poprawy tej reguły jest użycie tzw. współczynnika zapominania $0 \leq \alpha \leq 1$, który zmniejsza znaczenie aktualnych wag.

✓ Kolejnym krokiem było przetestowanie sieci dla różnych współczynników uczenia i zapominania.



Nauka wg metody Hebb'a została wykonana w 63 epokach.

Plots-> Performance:



W tabelce przedstawię wyniki dla różnych współczynników uczenia i zapominania wyniki uzyskane wg reguły Hebba, następnie wypiszę na jakie grupy zostały podzielone dane litery i według jakich cech zostało to uczynione.

Współczynnik zapominania= 0.001

Współczynnik uczenia	0.001	0.1
Liczba epok	29	13
Litera:		
A	2.5000e-04	0.0250
D	-1.0000	1
E	5.0000e-04	0.0500
F	7.5000e-04	0.0749
G	-2.5000e-04	-0.0250
H	-1	0.1000
K	7.5002e-04	0.0749
N	0.0010	0.1000
O	-7.4607e-14	-2.2204e-16
P	9.8373e-04	0.1000
R	0.0010	0.1000
S	-1	-0.0250
T	0.0010	1.0000
U	7.5000e-04	0.0751
Y	1.0000e-03	0.1000

Współczynnik zapominania= 0.1

Współczynnik uczenia	0.001	0.1
Liczba epok	12	18
Litera:		
A	2.5000e-04	1
D	1	1
E	5.0000e-04	0.0500
F	7.5000e-04	0.0749
G	-2.5061e-04	-0.0250
H	0.0010	0.1000
K	7.5000e-04	0.0749
N	0.0010	0.1000
O	4.4409e-16	9.2444e-12
P	0.0010	0.1000
R	0.0010	0.1000
S	-2.5000e-04	-0.0250
T	0.0010	0.1000
U	7.4999e-04	0.0751
Y	-1.0000	0.1000

4.Podsumowanie

Sieć uczona wg. metody Hebba na ogół uzyskuje dobre wyniki i samoczynnie grupuje sygnały wejściowe w "kategorie". Efekt ten nie jest jednak zawsze pewny, gdyż istotnie zależy od stanu początkowego sieci. Jak widzimy w przedstawionej przeze mnie tabelce, która zawiera odpowiedź zgodną z sumowaną liczbą parametrów dla danej litery. W pierwszym zestawieniu, który zawiera współczynnik zapominania równy 0.001 przy współczynniku uczenia 0.001, litery takie jak A i G są przedstawione jako jedna kategoria, następnie U i F również mają wartości takie same. Jak widzimy metoda Hebba to oto zadanie wykonała prawidłowo i w krótkim czasie, ponieważ od 12 do 29 epokach jest w stanie zakończyć zadanie. Dane umieszczone wyżej oraz różnorodność wyników, jest zasługą dokonywanych zmian w parametrach oraz wprowadzeniu współczynnika zapomnienia, ponieważ wadą tej reguły jest wykładniczy wzrost wag przy wielokrotnej prezentacji takiego samego wymuszenia. Efektem tego jest nasycenie neuronu, dla uniknięcia takiej sytuacji modyfikuje tę regułę przez wprowadzenie współczynnika zapomnienia γ . Odpowiedni dobór wartości γ umożliwia powstrzymanie niekontrolowanego wzrostu wag.