Piotr Chromniak, WIMiIP, grupa 1

Sieci neuronowe oraz zastosowanie ich w rozpoznawaniu wzorców

Moim projektem jest sieć neuronowa będąca zdolna do rozpoznawania 25 znaków które są charakterystyczne dla alfabetu angielskiego. Jako że miałem okazję wypróbować szereg typów sieci, zaprezentuje tutaj moje wnioski oraz uzasadnię końcowy wybór.

Sieciami które przetestowałem są Samoorganizujące się mapy, WTA oraz wsteczna propagacja z czego dwie pierwsze to sieci bez nauczyciela a ostatnia to sieć z nauczycielem. Moim naturalnym wyborem na początku była siec opierająca się na wielu warstwach i algorytmie wstecznej propagacji błędu. Wynikało to z faktu, że posiadamy wzorce prawidłowego sygnału zatem najefektywniejszym sposobem byłoby użycie tego typu sieci. Natomiast dla eksperymentu zastosowałem także sieci bez nauczyciela.

Reprezentacja problemu

Litery w moim problemie są reprezentowane przez pole pikseli, w tym przypadku to 6x4 pix co daje łącznie pole 24 pikseli. Jest to relatywnie mała ilość ale nic nie stoi na przeszkodzie zwiększyć ilość danych wejściowych w przyszłości w razie potrzeby. Wynika to z faktu że zastosowane sieci z punktu widzenia oprogramowania są bardzo łatwo skalowalne co pozwala dostosować ilość wejść w zależności od sprzętu na jakim pracujemy oraz dokładności jaką oczekujemy od sieci.

Problem charakteryzował się stała ilością próbek, 25 czyli tyle ile rozpoznawanych liter. Każda litera był reprezentowana przez 24 piksele w które w programie były prezentowane jako wektor 24 danych o wartościach 0 bądź 1. W przypadku sieci ze wsteczną propagacją wyjściem było 25 wyjść, z którego każdy był kojarzony z jednym ze znaków. Każde wyjście można było interpretować jak pewnego rodzaju miara prawdopodobieństwa że dany sygnał to dana litera. W przypadku SOM oraz WTA wejścia były takie same natomiast wyjścia miały inny sens. Algorytmy te znajdowały skupiska danych – środek ciężkości danego zbioru i na podstawie sygnału wejściowego przypisywał sygnał do jednego z klastrów. W efekcie każdy sygnał miał przydzielany swoją grupę.

WTA

Sieć typu Winner Takes All oblicza poprawkę po przetworzeniu sygnału na podstawie odległości wektora sygnału od wektora wag neuronu. Nazwa sugeruje że dla każdego sygnału zwycięski neuron jest tylko jeden i tylko on jest modyfikowany jeśli chodzi o jego wagi. Efektem tej sieci jest klastryzacja danych. W moim przypadku oznacza to, że litery o pewnych podobnych odległościach od wektora wag danego neuronu(a zatem klastra) zostaną przypisane do tej samej klasy danych. Dzięki temu nasze dane są posegregowane ze względu na podobieństwo.

SOM

W przypadku biblioteki AForge WTA jest wyjątkowym przypadkiem sieci SOM ponieważ ilość neuronów oraz skala zmian była aplikowana tylko do jednego neuronu. Gdzie w przypadku SOM modyfikowanych jest wiele neuronów w zależności od tego jaki dystans propagacji poprawki wybierzemy. Dystans propagacji jest określany w AForge jako promień. SOM posiada ciekawą właściwość pozwalając mapować wielowymiarowe problem( w moim przypadku 24- wymiarowy problem, gdzie każdy wymiar ma ten sam sens co reszta) do dwóch wymiarów. W efekcie otrzymujemy mapę przedstawiającą podobieństwo sygnałów które jest przedstawiane w postaci odległości i względnej orientacji między klastrami danych zwizualizowanymi na mapie.

Sieć wielowarstwowa z algorytmem wstecznej propagacji

Jest to sieć ucząca się z nauczycielem. Modyfikacja wag neuronu jest obliczana na bazie różnicy między wartością oczekiwaną a wartością przewidzianą. Następnie błąd ten jest rzutowany wstecz a miarą proporcjonalności tego rzutu jest waga danego wejścia. Czynnikiem decydującym o wielkości poprawki jest też współczynnik uczenia oraz pochodna funkcji aktywacyjnej.

Testy

Zdecydowałem się omówić WTA oraz SOM w jednej części ponieważ z racji specyfikacji sieci mamy do czynienia z błędem absolutnym będącym odległością wektora wejścia i wektora wag. Nie możemy zatem bezpośrednio porównywać wyników z wynikami dla sieci z nauczycielem, w którym zastosowana jest metryka MSE które jest bardziej naturalna i więcej mówi o mierze błędu.

WTA:

SOM:

Podsumowanie:

|  |  |  |
| --- | --- | --- |
|  | SOM | WTA |
| Ilość iteracji | 19393 | 226 |
| Końcowy błąd | 15,98 | 25,27 |

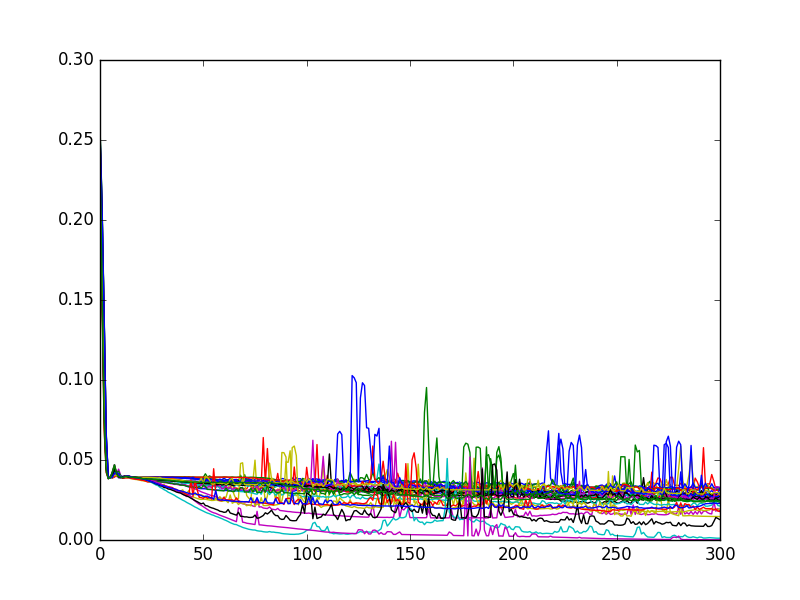
Jak widzimy WTA szybciej uczy się klastrować dane ale także szybciej osiąga minimum jeśli chodzi o funkcję błędu, natomiast SOM potrafi zejść do mniejszego progu błędu jednak nieproporcjonalnym kosztem ponieważ zajmuje to około 20k iteracji.

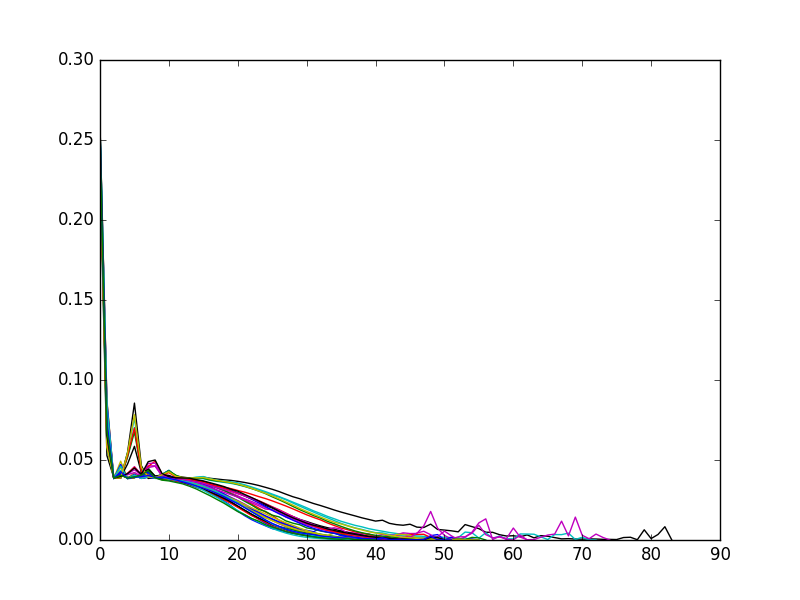
Ostatecznie zastosowanie sieci bez nauczyciela okazuje się mało przydatne ponieważ priorytetem w tym problemie jest bardzo precyzyjne rozróżnianie wzorców. Klastryzacja danych może być pomocna w przypadku SOM ponieważ dzięki mapie 2D jesteśmy w stanie zobaczyć pewne podobieństwa w literach natomiast sam fakt przynależności do pewnej klasy nic nam nie daje.

Wsteczna propagacja w kilku topografiach

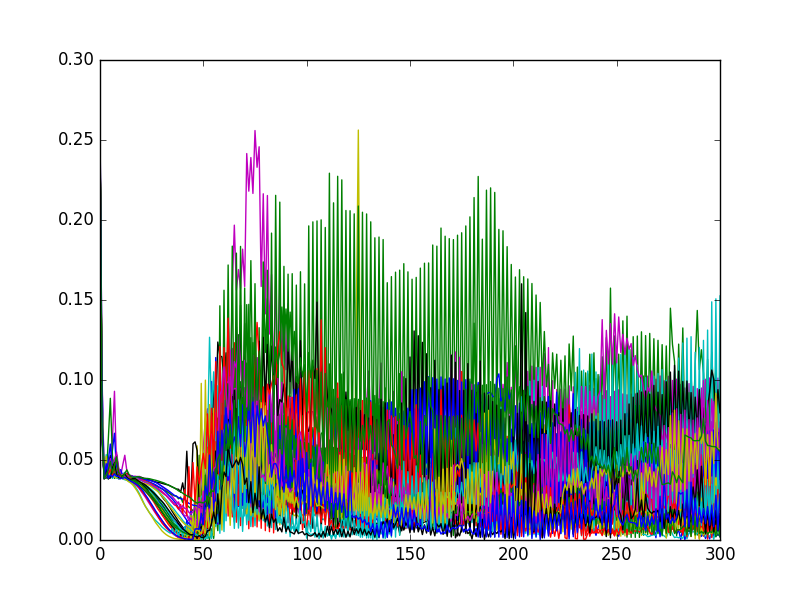
\* Na poniższych wykresach oś pionowa reprezentuje błąd MSE a pozioma ilość epok

Topologia 24-4-25:

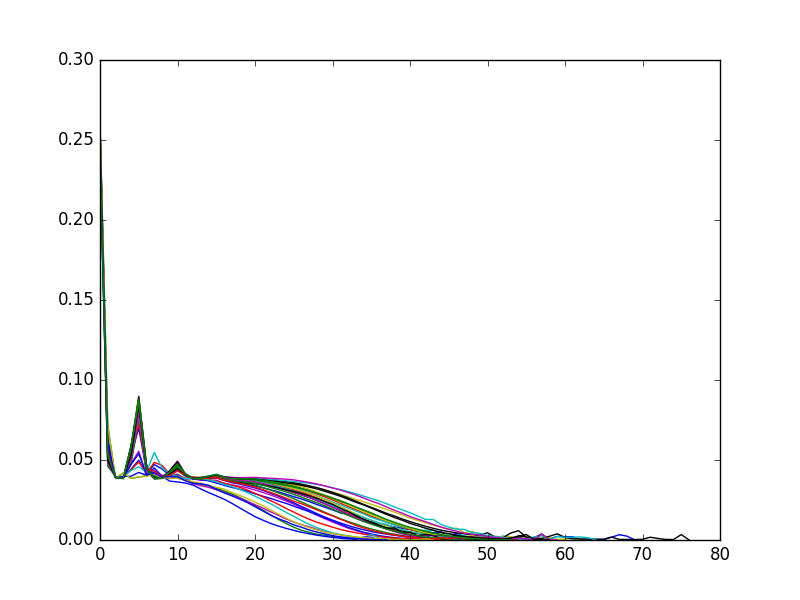


Topologia 24-20-25: 

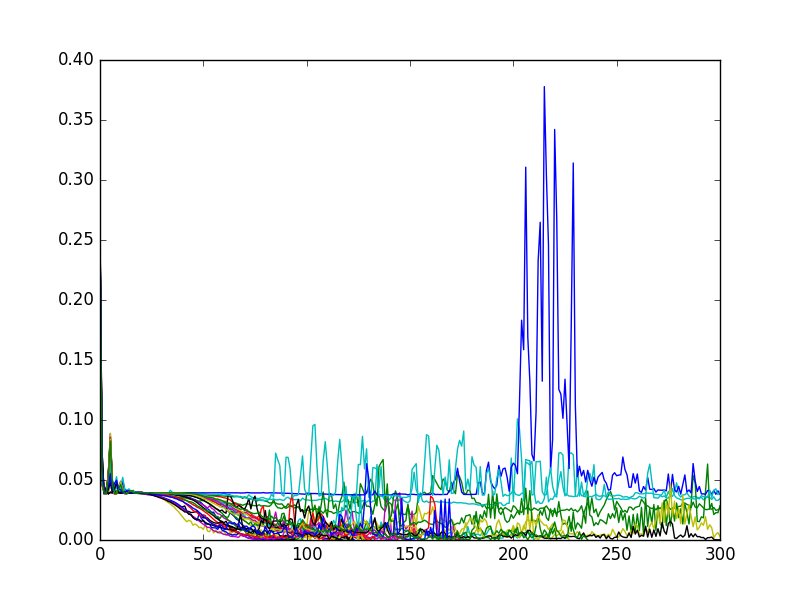
Topologia 24-20-25 – liniowa aproksymacja funkcji sigmoidalnej:



Topologia 24-24-25:



Topologia 24-24-24-25



Podsumowanie:

Sieci z nauczycielem uczą się o wiele szybciej z racji posiadania oryginalnych wzorców do których powinny dążyć. Topologie zostały dobrane losowo, dodatkowo dla jednej z nich została zmieniona funkcja aktywacyjna. Celem sieci było osiągnięcie błędu na poziomie 0.01. Sieci na których wykresach widzimy 300 iteracji, nie były w stanie sobie poradzić z osiągnięciem takiego błędu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 4 | 20 | 20-linear appr. | 24 | 24-24 |
| Ilość iteracji | 300+ | 45 | 300+ | 52 | 300+ |
| Błąd końcowy | 0,04-0,015 | 0,01 | 0,15-0,015 | 0,01 | 0,05-0,015 |

Jak widać najlepszą topologią z prezentowanych pięciu jest 24-20-25, która doszła do wyznaczonego poziomu błędu oraz jej funkcja(tj. sigmoidalna) charakteryzowała się relatywnie szybkim tempem uczenia.

Podsumowując dwa paragrafy testów – wybrana została przeze mnie sieć ucząca się z nauczycielem ponieważ mamy jasno określony wzorzec znaków co sprawia że uczenie też jest relatywnie szybkie i mało wymagające. Dodatkowo sieć wielowarstwowa pozwala nam modyfikować jej topologię w celu znalezienia lepszej pod względem szybkości uczenia. Co więcej sieć ta pozwala na dokładne rozróżnienie wzorca w przeciwieństwie do grupowania sygnałów które nie ma dla mnie praktycznego zastosowania w moim przypadku.

Taką sieć można wykorzystać na różne sposoby. Jej głównym celem jest oczywiście rozróżnianie znaków, natomiast żeby było to możliwe jest potrzebny system partycjonujący obraz. W moim wypadku na prostokąty o wielkości 6x4 pix, które następnie byłby analizowane prze sieć neuronową. Ma to swoje wady bo wybór obszaru do interpretacji nie jest najłatwiejszym zadaniem(oczywiście oprócz przypadku w którym mamy do czynienia z unormowanym dokumentem) za to sama analiza i rozpoznanie znaku przebiegałaby błyskawicznie.