Na wejściu problemu mamy obraz o wymiarach 28x28 pikseli, który prezentuje odręcznie zapisaną cyfrę (zakres od 0-9).

Każdy piksel posiada wartość z zakresu 0-255. Jeżeli wartość piksela jest równa 0 to na wejściu podajemy sygnał równy 0, w przeciwnym razie sygnał równy 1.

Do nauczenia i przetestowania sieci została wykorzystana baza MNIST (<http://yann.lecun.com/exdb/mnist/>), która zawiera 60 000 przykładów danych uczących i 10 000 danych testowych.

Na wyjściu problemu mamy 10 różnych sygnałów reprezentujących poszczególne cyfry.

Do rozwiązania problemu została wykorzystana wielowarstwowa sieć neuronowa oraz uczenie metodą wstecznej propagacji błędów (backpropagation).

Liczba neuronów w warstwie wejściowej jest równa **784** (28 pikseli x28 pikseli). Jeden piksel to jedno wejście, jeden neuron.

Liczba neuronów w warstwie wyjściowej jest równa **10**. Liczba cyfr do rozróżnienia.

Do wyznaczenie ilości neuronów w warstwie ukrytej użyty został wzór:

Źródło: http://www.ai.c-labtech.net/sn/pod\_prakt.html

gdzie:

Nwu - ilość neuronów w warstwie ukrytej,

Nwwe - ilość neuronów w warstwie wejściowej,

Nwwy - ilość neuronów w warstwie wyjściowej.

Struktura warstw wygląda następująco **784x90x10**.

Użyta została funkcja aktywacyjna sigmoidalna (zwana też krzywą logiczną) ze względu na to, iż jest ona najczęściej stosowana. Przyjmuje ona wartości pomiędzy 0 a 1 i posiada łatwą do obliczenia i ciągłą pochodną.

Wzór funkcji:

Pochodna funkcji:

Gdzie:

β – domyślnie równy 1

e – liczba Eulera

Ze względu na dużą ilość danych uczących współczynnik uczenia **k** zmienia się dynamicznie podczas działania programu:

0.7 - początkowa wartość,

0.5 - po 2 000 iteracji,

0.4 - po 5 000 iteracji,

0.3 - po 10 000 iteracji,

0.2 - po 15 000 iteracji,

0.1 - po 25 000 iteracji,

0.05 - po 45 000 iteracji.

Po nauczeniu i przetestowaniu sieci danymi testującymi udało się poprawnie rozpoznać **95.3%** wszystkich cyfr.

Wagi wejść i biasy neuronów są generowane losowo z zakresu (-0.5, 0.5).

Schemat procesu uczenia (metoda backpropagation) jest następujący:

1. Podajemy sygnał treningowy na wejściu.
2. Dla każdego neuronu obliczamy wartość wyżej wymienionej funkcji aktywacyjnej oraz jej pochodną.
3. Obliczamy błędy dla wszystkich neuronów.
   1. Błędy neuronów wyjściowych, są równe: wartość oczekiwana – wartość otrzymana,
   2. Błędy dla pozostałych neuronów, są obliczane poprzez zsumowanie iloczynu wag oraz błędów z neuronów, do których wysyłają sygnał.
4. Korygujemy wagi oraz biasy według wzorów:

gdzie:

w – waga,

k – współczynnik nauczania,

E – błąd,

y’(x) – pochodna,

y – wartość wejściowa.

1. Kontynuujemy od punktu 1. do wyczerpania danych uczących.