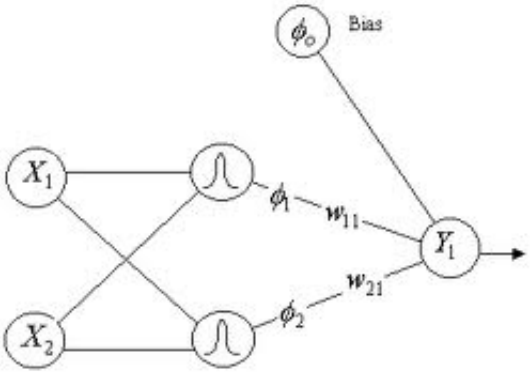
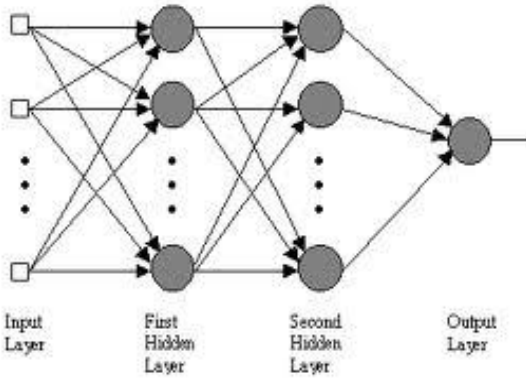
Metody klasyfikacji

Metody klasyfikacji

Do typowych metod klasyfikacji i regresji należy zaliczyć klasyfikator kNN, drzewa decyzji jak również systemy indukcji reguł. Algorytmy drzew decyzji jak również systemy indukcji reguł bazujące na algorytmie sekwencyjnego pokrywania wykorzystują w trakcie uczenia algorytmy przeszukiwania, podobnie też działają algorytmy selekcji prototypów dla klasyfikatora kNN. Jednak w szczególności dla zmiennych numerycznych najlepszymi rozwiązaniami zwykle są algorytmy oparte o minimalizację ciągłej funkcji celu. Należą do nich m.in. sieci neuronowe typu MLP, sieci neuronowe typu RBF, klasyfikator SVM oraz różne inne. W przypadku sieci neuronowych proces optymalizacji sprowadza się zwykle do wykorzystania algorytmów gradientowych, które dążą do minimalizacji funkcji celu zdefiniowanej w postaci błędu średniokwadratowego.



a) Sieć typu RBF



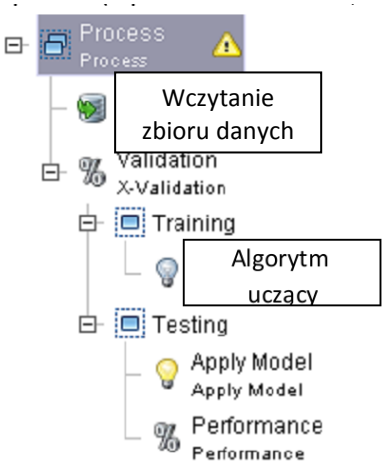
b) Sieć typu MLP

Takie rozwiązanie ma jednak wadę, którą jest brak pewności odnalezienia ekstremum globalnego, czyli znalezienia najlepszych możliwych wag. Wszystko bowiem zależy od punktu startu procesu optymalizacji. Alternatywne rozwiązanie zaproponowano w algorytmie SVM (algorytm ten z punktu widzenia architektury jest równoważny sieci neuronowej RBF). W algorytmie tym optymalizowany problem ma postać funkcji kwadratowej, dzięki czemu nie ma problemów odnalezienia minimum globalnego. Pewne ograniczenia jednak dotyczą problemu rozwiązania dużej złożoności obliczeniowej.

Najprostszymi klasyfikatorami są model liniowy i kwadratowy. Ich działanie polega na próbie dyskryminacji klas za pomocą hiperpłaszczyzny seprarującej (model liniowy) oraz funkcji kwadratowej (model kwadratowy). Takie proste rozwiązanie często pozwala na uzyskanie bardzo dobrych rezultatów, lepszych niż wykorzystanie zaawansowanych modeli predykcyjnych.

Do wykonania w RapidMiner

1. Zbuduj Schemat do testów jak na rysunku Pamiętaj aby wczytać zbiór danych w odpowiedni sposób (Operator Read CSV). Do testów wykorzystaj test krzyżowy.



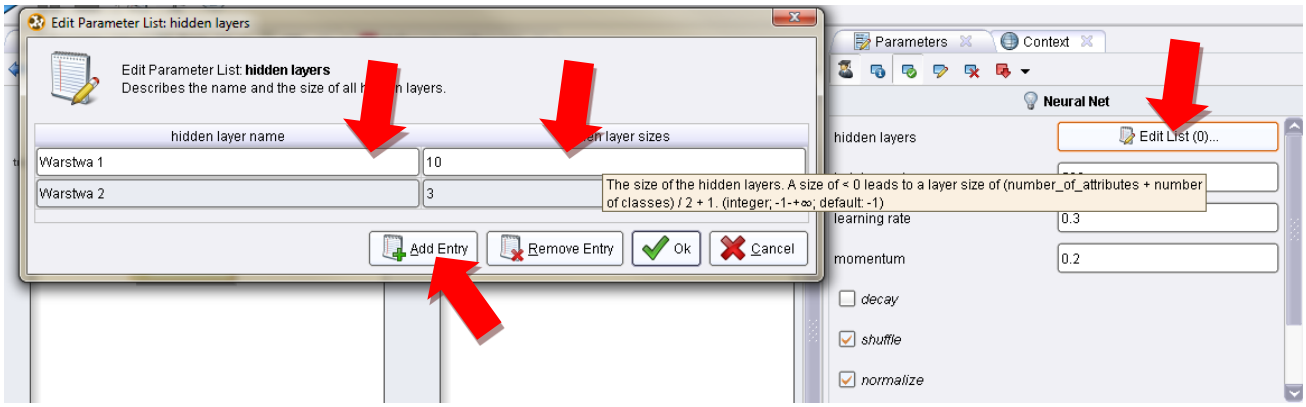
Dla ułatwienia prowadzenia obliczeń i testów wykorzystaj operatory *Loop Parameters* oraz Log.

Testy wykonaj na zbiorach danych: zbior5.csv

2. Dokonaj weryfikacji jakości działania klasyfikatora k-NN, oraz wpływu parametru *k* na jakość uzyskanych wyników predykcji. W tym celu:
    a. Umieść operator k-NN w miejsce „Algorytm uczący”
    b. Dokonaj zmian wartości parametru k w zakresie 1-20
    c. Wyniki przedstaw w formie graficznej nanosząc na rysunku ustawienia konfiguracyjne oraz uzyskaną dokładność
    d. Wnioski
3. Dokonaj weryfikacji działania klasyfikatora SVM i wpływu jego parametrów na jakość uzyskiwanych wyników
    a. Wstaw operator Support Vector Machine (LibSVM) w miejsce bloczku *Algorytm uczący*
    b. W opcja konfiguracyjnych SVM’a ustaw:*kernel type* = *rbf* a następnie zmieniaj parametry *C* oraz *gamma*. Parametr *C* decyduje tutaj o szerokości marginesu zaufania i określa maksymalną wartość wagi jaka może zostać okreslona dla danego wektora. , parametr *gamma określa* szerokość funkcji jądrowej
    c. Weryfikacji dokonaj dla ustawień z zakresu:
    C: 0.001 do 100 zmieniając wartości C z krokiem x10(6 wartości –0.001,0.01,0.1,1 itd)
    Gamma:0.001 do 10, tak aby zbadać również 5 różnych wartości –
    UWAGA: W obydwu przypadkach jeśli korzystasz z operatora Loop Parameter użyj skali logarytmicznej dla wartości
    d. Powtórz obliczenia dla SVM’a z ustawionym parametrem: *kernel type* = *linear.* Dla takiego parametru zmieniaj tylko wartości parametru *C.* Pozostałe parametry nie mają zastosowania dla modelu liniowego
    e. Wyniki przedstaw w formie graficznej nanosząc na rysunku ustawienia konfiguracyjne oraz uzyskaną dokładność
    f. Skomentuj wnioski
4. Zweryfikuj działanie modelu liniowego *Linear Discriminant Analysis.* Zwróć uwagę, że operator ten nie ma żadnych ustawień konfiguracyjnych. Pamiętaj że klasyfikator liniowy stara się podzielić przestrzeń hiperpłaszczyzną.
5. Dokonaj weryfikacji jakości działania sieci neuronowej, oraz wpływu rozmiaru sieci (liczba warstw oraz liczba wektorów w warstwie) na jakość uzyskiwanych wyników.

a. Wstaw operator NeuralNet w miejsce Algorytm Uczący na schemacie

b. Dokonaj jego konfiguracji, w tym celu kliknij na hidden layers w opcjach konfiguracyjnych operatora NeuralNet. Następnie korzystając z przycisku Add Entry dodawaj kolejne warstwy określając ich nazwy (np. Warstwa 1, Warstwa 2) i wpisując określoną liczę neuronów w określonej warstwie (kolumna Hidden Layer Size)



c. Modyfikuj ustawienia dotyczące liczby warstw ukrytych (od 1 do 2) oraz liczby neuronów w poszczególnych warstwach (od 2 do 10)

d. Weryfikacji dokonaj następujących konfiguracji

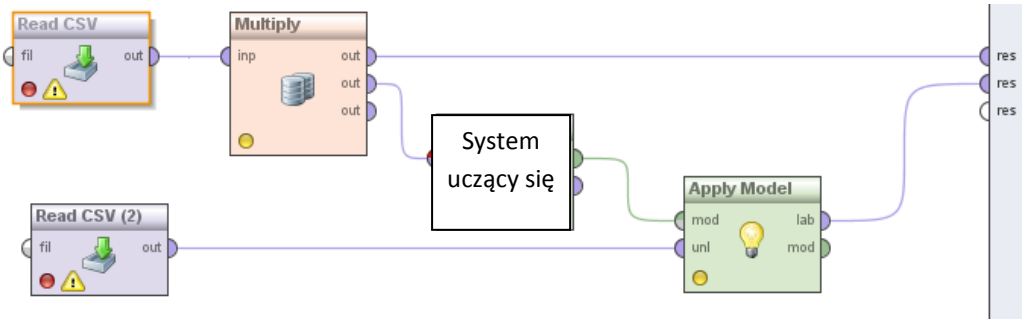
Lp	Warstwa	Liczba neuronów
1	Warstwa1	2
2	Warstwa 1	5
3	Warstwa1	10
4	Warstwa 1 Warstwa 2	10 5
5	Warstwa 1 Warstwa2	5 10
6	Warstwa1 Warstwa2 Warstwa3	5 5 5

e. Podczas obliczeń ustaw *learning rate* na wartość 0.7

f. Wyniki przedstaw w formie graficznej nanosząc na rysunku ustawienia konfiguracyjne oraz uzyskaną dokładność

g. Wnioski

6. Ponieważ większość z załączonych zbiorów jest zbiorami w przestrzeni 2D, dla każdego z modeli zbudowanych w późniejszych zadaniach dokonaj weryfikacji wizualnej uzyskanych wyników. W tym celu zbuduj proces:



Gdzie operator *Read CSV* odpowiada za wczytanie zbioru uczącego, natomiast operator *Read CSV(2)* odpowiada za wczytanie zbioru *zbior_test.csv.* Następnie dla uzyskanych wyników dokonaj wizualizacji *Plot view -> Scatter plot* z ustawieniem atrybutów att1 i att2 oraz ustawionym kolorem odpowiednio jako *Label* dla zbioru (ExampleSet (Multiple)) oraz *Predicted(Label) dla zbioru (ExampleSet (Read CSV(2)))*

Zaobserwuje kształt granicy decyzji uzyskanego modelu uczącego. Zadanie to powtórz dla najgorszej i najlepszej konfiguracji każdego z modeli predykcyjnych.

7. Zrób zbiorcze porównanie najlepszych wyników różnych klasyfikatorów. Czy można wskazać jeden najlepszy?