Sprawozdanie - sieci Kohonena

Dominik Rafacz

4/27/2020

Oświadczenie o samodzielności pracy

Potwierdzam samodzielność powyższej pracy oraz niekorzystanie przeze mnie z niedozwolonych źródeł. *Dominik Rafacz*

Przygotowanie

Załadowanie niezbędnych pakietów

```
library(MIOwAD)      # pakiet z sieciami
library(dplyr)       # transformacja danych
library(ggplot2)     # wykresy
library(patchwork)   # laczenie wykresow
```

Wczytywanie danych

```
X_2d <- read.csv("../data/kohonen/hexagon.csv")
X_3d <- read.csv("../data/kohonen/cube.csv")
```

Labortorium 7

Cel: zbudowanie dwuwymiarowych sieci Kohonena i dopasowanie ich do danych 2d i 3d

Wykonamy eksperyment, dopasowując sieci Kohonena do danych ze zbioru 2d i 3d. Będziemy dopasowywać sieci o rozmiarze 10 x 10, korzystając z dwóch możliwych funkcji sąsiedztwa – funkcji gaussowskiej i tzw. meksykańskiego kapelusza, a także z pięciu możliwych wartości parametru skali.

```
ngh_funs <- c(
  "gauss" = gauss,
  "mexican_hat" = mexican_hat
)

scale_par <- c(
  `0.1` = 0.1,
  `0.5` = 0.5,
  `1` = 1,
  `5` = 5,
  `10` = 10
)

# trenujemy sieci dla zbioru 2d dla każdej funkcji dla każdego parametru
```

```
nets <- list(lab7 = list())
nets[["lab7"]][["2d"]] <- nlapply(ngh_funs, function(fun)
  nlapply(scale_par, function(par)
    kohonen_network(X_2d[, 1:2], 10, 10, lambda = 10, ngh_fun = fun, scale = par)
    ))
```

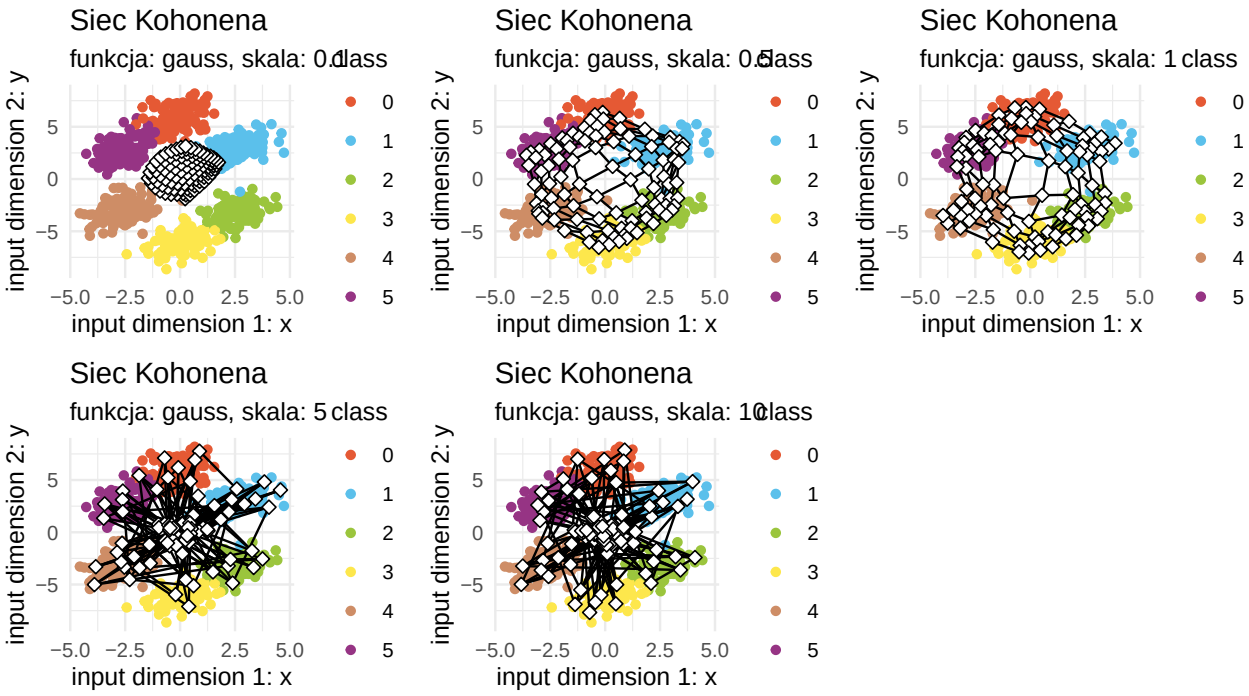
```
# trenujemy sieci dla zbioru 3d dla każdej funkcji dla każdego parametru
nets[["lab7"]][["3d"]] <- nlapply(ngh_funs, function(fun)
  nlapply(scale_par, function(par)
    kohonen_network(X_3d[, 1:3], 10, 10, lambda = 10, ngh_fun = fun, scale = par)
    ))
```

Teraz wygenerujemy wykresy do wizualizacji efektów treningu.

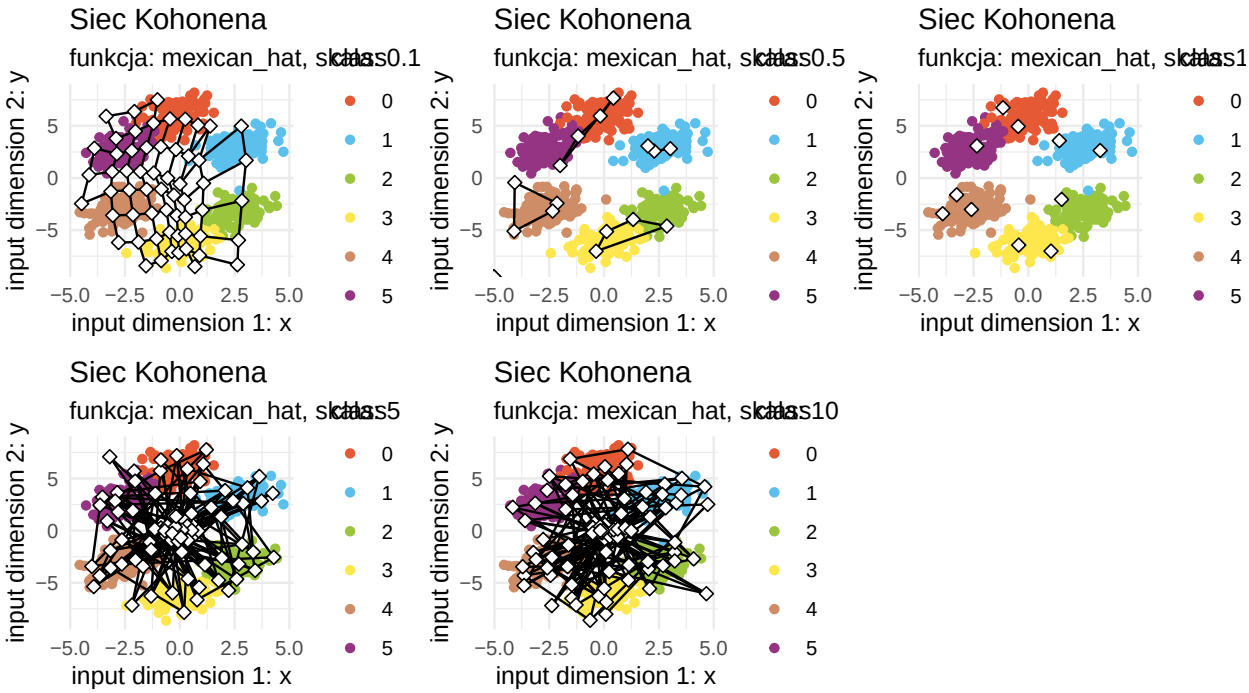
```
plots <- list(lab7 = list())
plots[["lab7"]][["2d"]] <- nlapply(names(ngh_funs), function(fun)
  nlapply(names(scale_par), function(par)
    plot_kohonen(generate_net_plot_data(nets[["lab7"]][["2d"]][[fun]][[par]]), X_2d) +
      ggtitle("Siec Kohonena", paste0("funkcja: ", fun, ", skala: ", par))
  ))

plots[["lab7"]][["3d"]] <- nlapply(names(ngh_funs), function(fun)
  nlapply(names(scale_par), function(par) {
    dat <- generate_net_plot_data(nets[["lab7"]][["3d"]][[fun]][[par]])
    p_xy <- plot_kohonen(dat, X_3d, inp_dims_plot = c(1, 2), inp_class = 4)
    p_zy <- plot_kohonen(dat, X_3d, inp_dims_plot = c(3, 2), inp_class = 4)
    p_xz <- plot_kohonen(dat, X_3d, inp_dims_plot = c(1, 3), inp_class = 4)
    (p_xy + p_zy) /
      (p_xz + plot_spacer())
  }))
```

Przyjrzyjmy się teraz samym wykresom.

```
plots[["lab7"]][["2d"]][["gauss"]][["0.1"]] +
  plots[["lab7"]][["2d"]][["gauss"]][["0.5"]] +
  plots[["lab7"]][["2d"]][["gauss"]][["1"]] +
  plots[["lab7"]][["2d"]][["gauss"]][["5"]] +
  plots[["lab7"]][["2d"]][["gauss"]][["10"]]
```




```
plots[["lab7"]][["2d"]][["mexican_hat"]][["0.1"]] +
  plots[["lab7"]][["2d"]][["mexican_hat"]][["0.5"]] +
  plots[["lab7"]][["2d"]][["mexican_hat"]][["1"]] +
  plots[["lab7"]][["2d"]][["mexican_hat"]][["5"]] +
  plots[["lab7"]][["2d"]][["mexican_hat"]][["10"]]
```

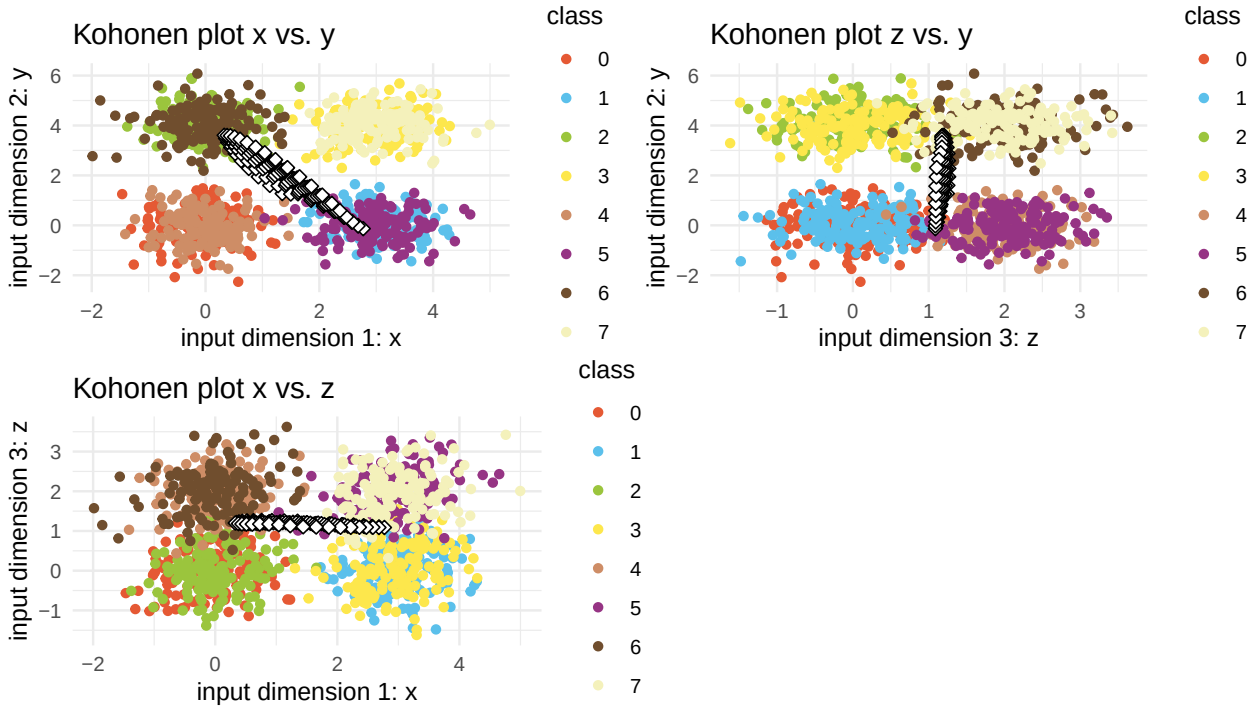


Na pierwszym zestawie wykresów widzimy sieci dopasowane do zbioru z użyciem funckji gaussowskiej z róznym parametrem skali, na drugim funckji meksykańskiej z tymi samymi parametrami. Możemy zauważyć:

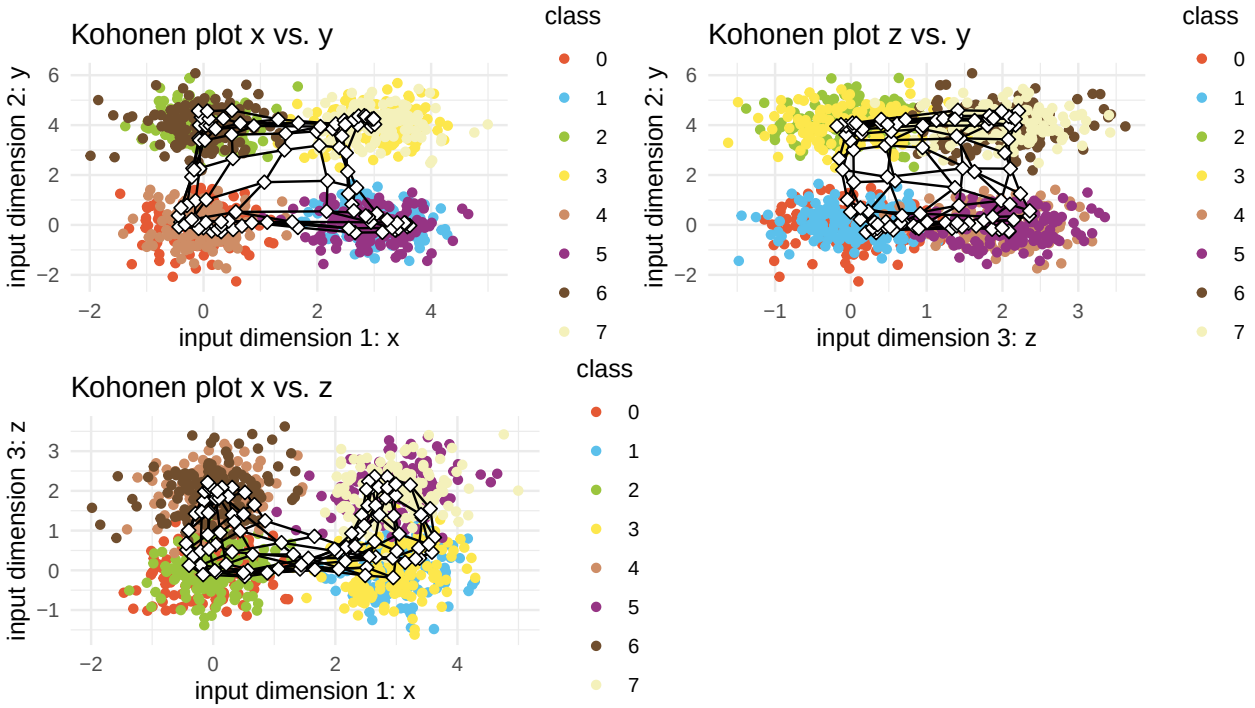
- Dla małego parametru skali, sieć nie dopasowuje się prawie w ogóle do danych.
- Bez modyfikacji skali, dopasowanie jest odpowiednie dla funkcji gaussowskiej
- Dla dużego parametru skali, relacja sąsiedztwa neuronów prawie nie ma w ogóle znaczenia i siatka jest bardzo chaotyczna.
- Funkcja meksykańska "wyrzuca" wiele neuronów poza obszar danych przy nieodpowiednim doborze parametrów.

Możemy zauważyć, że na dopasowaniach dla najlepszej liczby parametrów neurony klastrują się podobnie do klastrów danych.

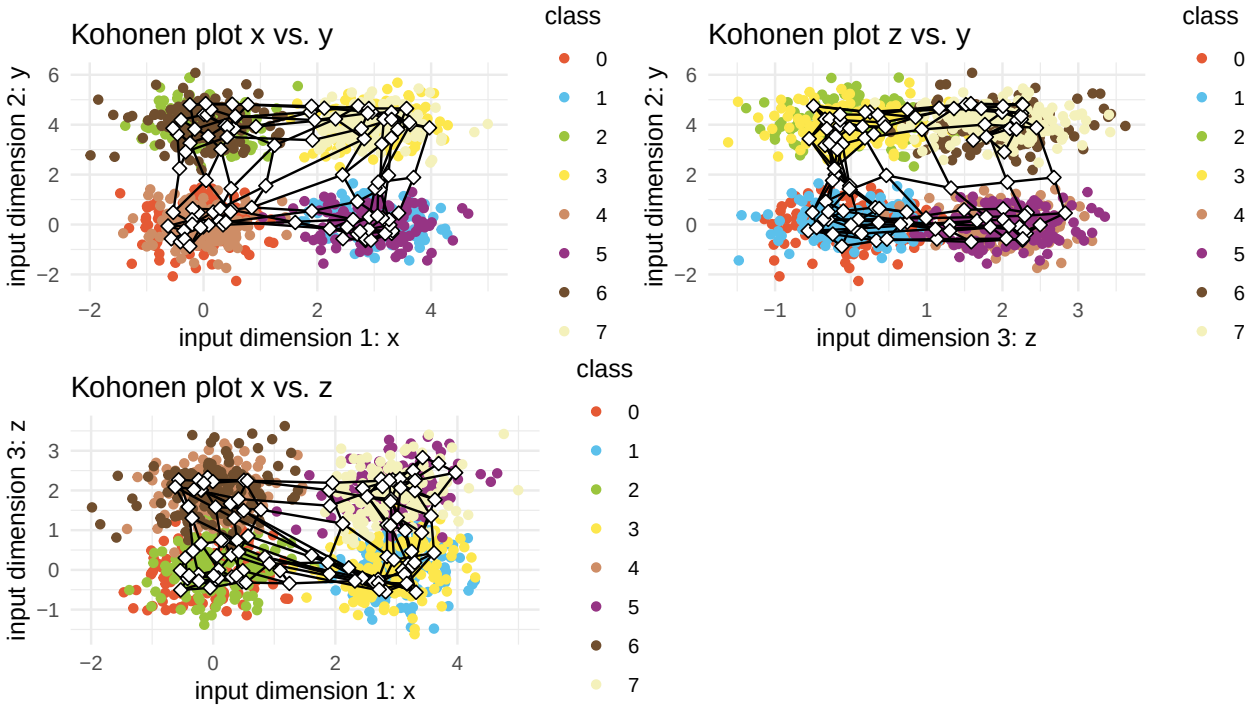
```
plots[["lab7"]][["3d"]][["gauss"]][["0.1"]]
```



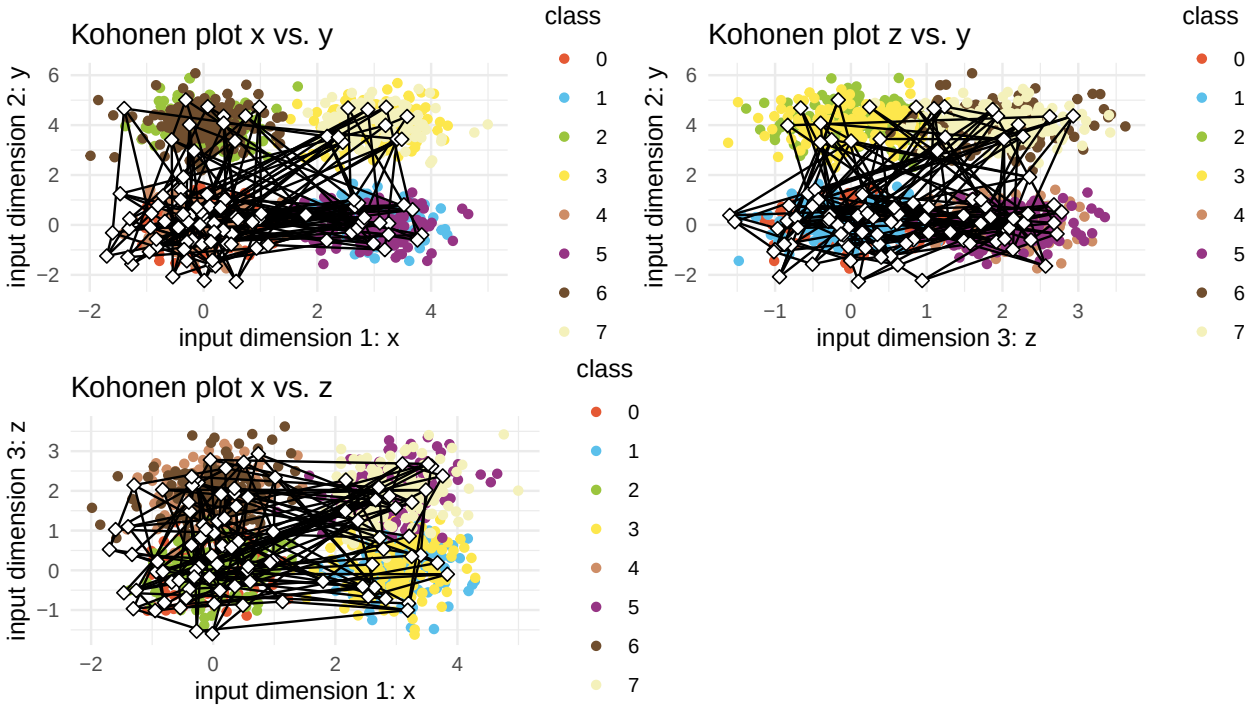

```
plots[["lab7"]][["3d"]][["gauss"]][["0.5"]]
```



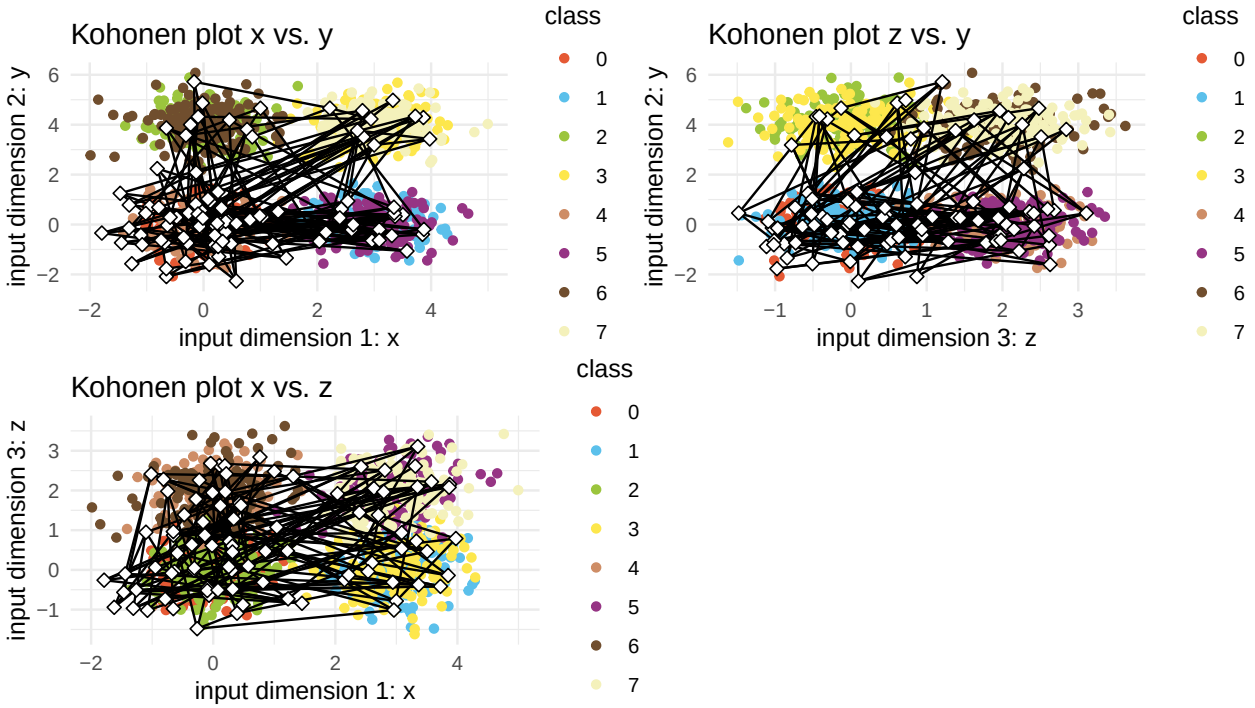

```
plots[["lab7"]][["3d"]][["gauss"]][["1"]]
```



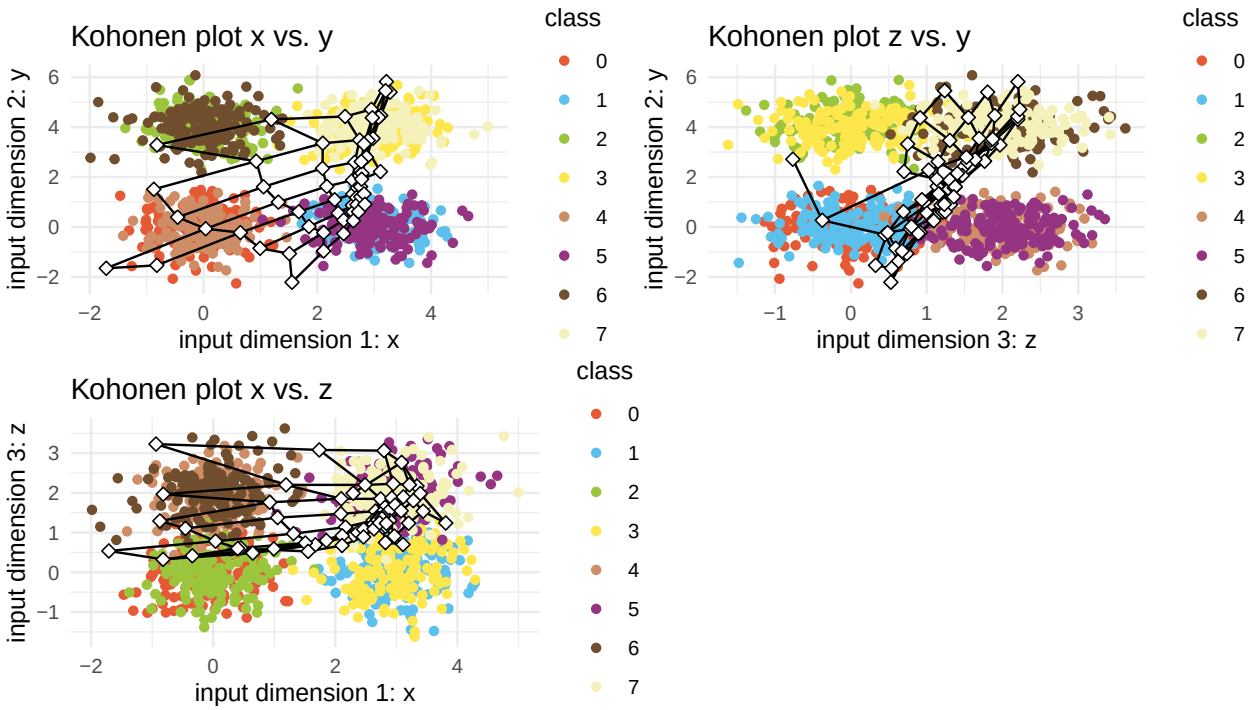

```
plots[["lab7"]][["3d"]][["gauss"]][["5"]]
```



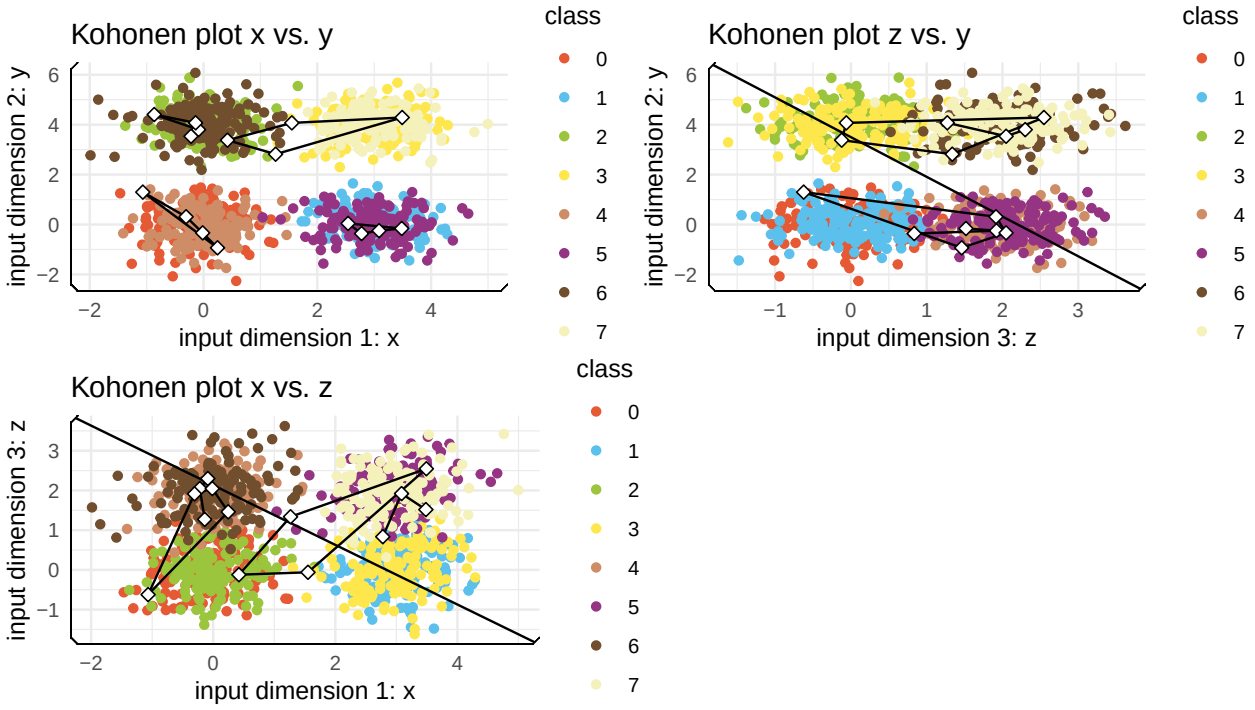

```
plots[["lab7"]][["3d"]][["gauss"]][["10"]]
```



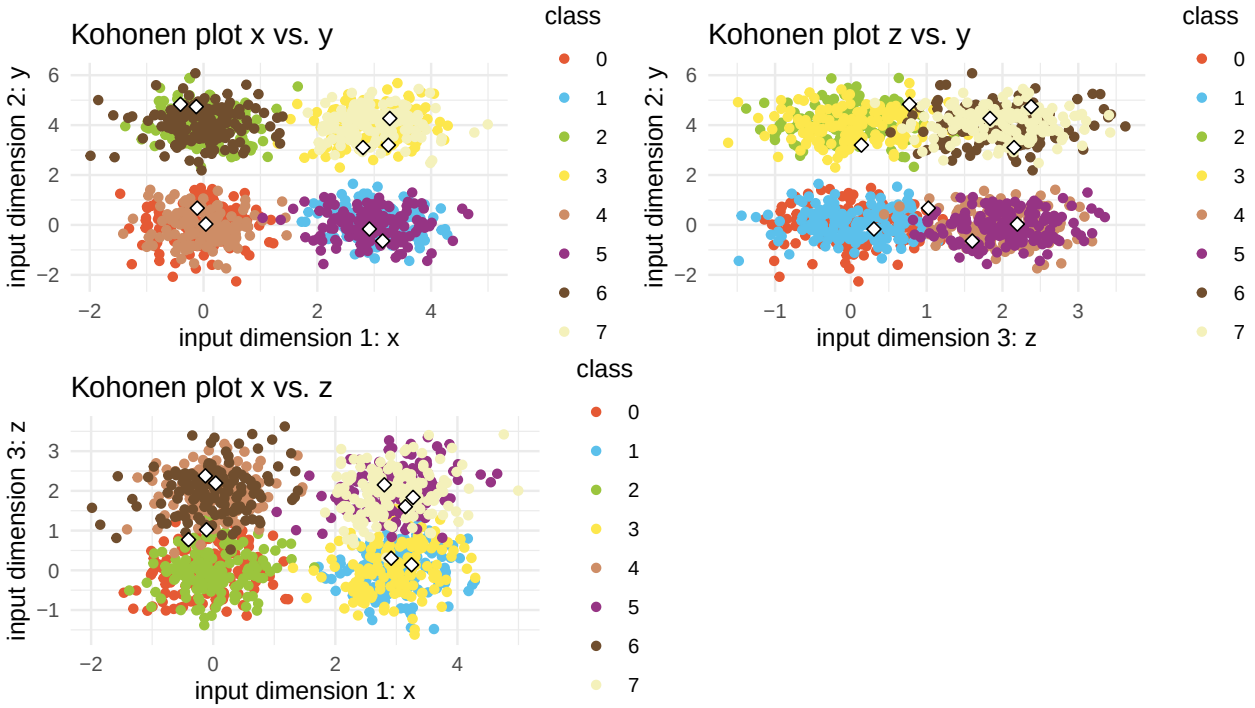

```
plots[["lab7"]][["3d"]][["mexican_hat"]][["0.1"]]
```



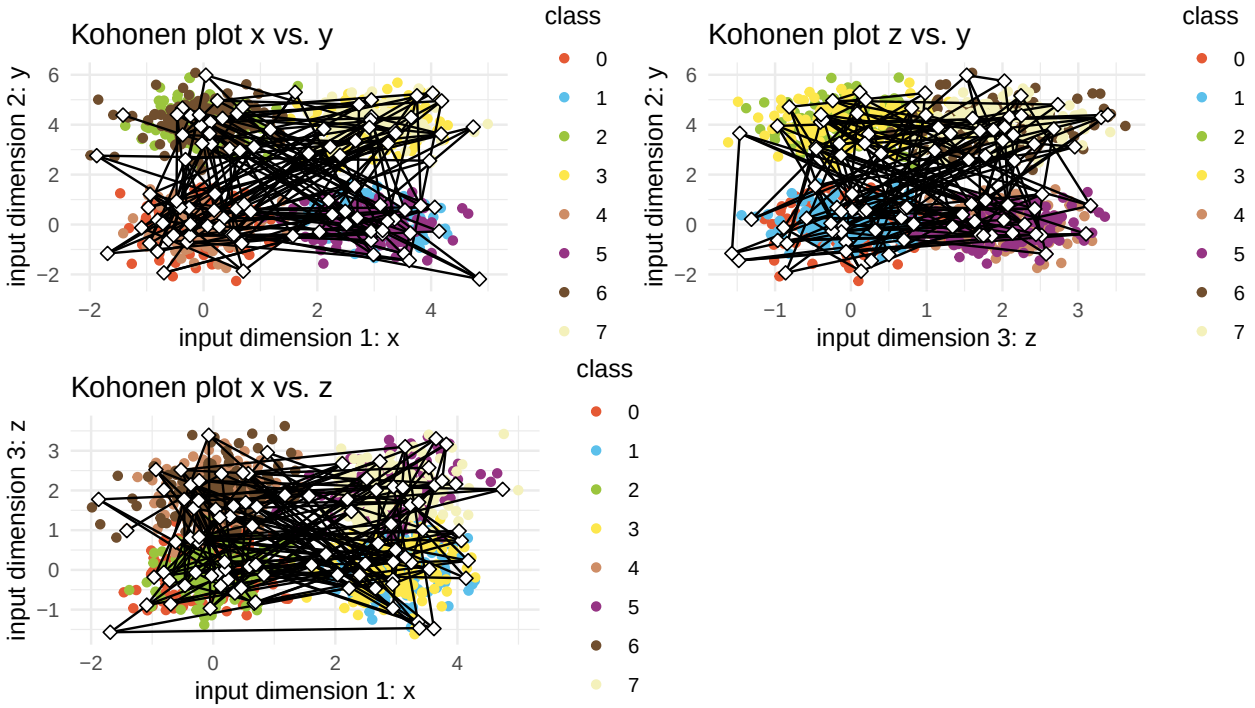

```
plots[["lab7"]][["3d"]][["mexican_hat"]][["0.5"]]
```



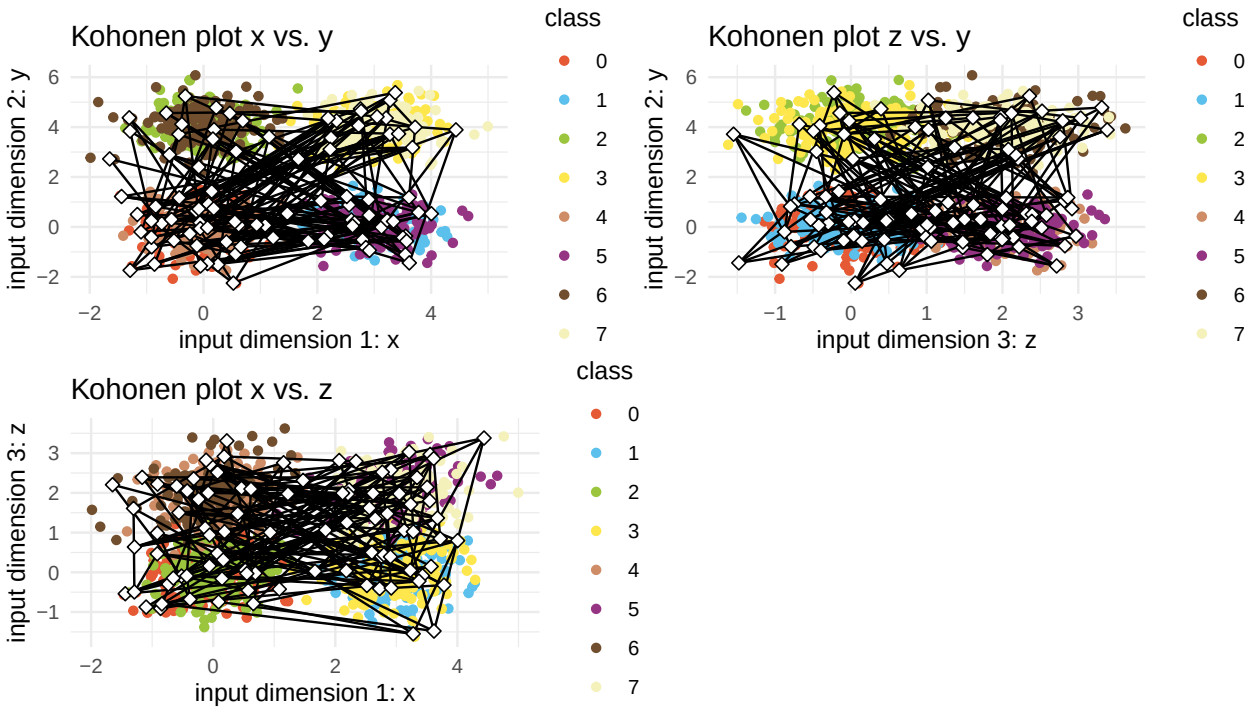

```
plots[["lab7"]][["3d"]][["mexican_hat"]][["1"]]
```




```
plots[["lab7"]][["3d"]][["mexican_hat"]][["5"]]
```




```
plots[["lab7"]][["3d"]][["mexican_hat"]][["10"]]
```



Tutaj każdy wykres to tak naprawdę zestaw trzech wykresów, prezentujących rzuty przestrzeni wejściowej na poszczególne płaszczyzny wyznaczone przez pary osi. Wykresy nie są aż tak czytelne (trudniej zobaczyć strukturę sieci), ale możemy wyciągnąć podobne wnioski jak w przypadku sieci 2d.