Zadanie 3a – klasifikácia

Máme 2D priestor, ktorý má rozmery X a Y, v intervaloch od -5000 do +5000. V tomto priestore sa môžu nachádzať body, pričom každý bod má určenú polohu pomocou súradníc X a Y. Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste). Každý bod patrí do jednej zo 4 tried, pričom tieto triedy sú: red (R), green (G), blue (B) a purple (P). Na začiatku sa v priestore nachádza 5 bodov pre každú triedu (dokopy teda 20 bodov). Súradnice počiatočných bodov sú:

```
R: [-4500, -4400], [-4100, -3000], [-1800, -2400], [-2500, -3400] a [-2000, -1400]

G: [+4500, -4400], [+4100, -3000], [+1800, -2400], [+2500, -3400] a [+2000, -1400]

B: [-4500, +4400], [-4100, +3000], [-1800, +2400], [-2500, +3400] a [-2000, +1400]

P: [+4500, +4400], [+4100, +3000], [+1800, +2400], [+2500, +3400] a [+2000, +1400]
```

Vašou úlohou je naprogramovať klasifikátor pre nové body – v podobe funkcie classify(int X, int Y, int k), ktorá klasifikuje nový bod so súradnicami X a Y, pridá tento bod do nášho 2D priestoru (s farbou podľa klasifikácie) a vráti triedu, ktorú pridelila pre tento bod. Na klasifikáciu použite k-NN algoritmus, pričom k môže byť 1, 3, 7 alebo 15.

Na demonštráciu Vášho klasifikátora vytvorte testovacie prostredie, v rámci ktorého budete postupne generovať nové body a klasifikovať ich (volaním funkcie classify). Celkovo vygenerujte 40000 nových bodov (10000 z každej triedy). Súradnice nových bodov generujte náhodne, pričom nový bod by mal mať zakaždým inú triedu (dva body vygenerované po sebe by nemali byť rovnakej triedy):

- R body by mali byť generované s 99% pravdepodobnosťou s X < +500 a Y < +500
- G body by mali byť generované s 99% pravdepodobnosťou s X > -500 a Y < +500
- B body by mali byť generované s 99% pravdepodobnosťou s X < +500 a Y > -500
- P body by mali byť generované s 99% pravdepodobnosťou s X > -500 a Y > -500

  (Zvyšné jedno percento bodov je generované v celom priestore.)

Návratovú hodnotu funkcie classify porovnávajte s triedou vygenerovaného bodu. **Na základe týchto porovnaní vyhodnoťte úspešnosť** Vášho klasifikátora pre daný experiment.

Experiment vykonajte 4-krát, pričom zakaždým Váš klasifikátor použije iný parameter k (pre k = 1, 3, 7 a 15) a vygenerované body budú pre každý experiment rovnaké.

Vizualizácia: pre každý z týchto experimentov vykreslite výslednú 2D plochu tak, že vyfarbíte túto plochu celú. Prázdne miesta v 2D ploche vyfarbite podľa Vášho klasifikátora.

Dokumentácia musí obsahovať opis konkrétne použitého algoritmu a reprezentácie údajov. Uveďte aj vizualizácie viacerých pokusov. V závere zhodnoťte dosiahnuté výsledky ich porovnaním.

Poznámka 1: Je vhodné využiť nejaké optimalizácie na zredukovanie zložitosti:

- Pre hľadanie k najbližších bodov si môžeme rozdeliť plochu na viaceré menšie štvorce, do ktorých umiestňujeme body s príslušnými súradnicami, aby sme nemuseli vždy porovnávať všetky body, ale len body vo štvorci, kde sa nachádza aktuálny bod a susedných štvorcoch.
- Je možné tiež využiť len základnú plochu a vyhľadávať k najbližších len z bodov, ktoré sú v nejakej vzdialenosti od klasifikovaného bodu. Či už je to ako kružnica alebo štvorec. Počiatočnú veľkosť kružnice/štvorca volíme podľa množstva bodov, ktoré sú aktuálne v priestore. Kružnica by mala obsahovať aspoň k bodov, štvorec aspoň 2k, aby to bolo určite k najbližších.
- Je tiež možné využiť algoritmus na hľadanie práve k najmenších hodnôt.
- Na hľadanie najbližších susedov v dvojrozmernom priestore sú vhodné aj k-d stromy.
- PyPy je implementácia programovacieho jazyka Python. PyPy často beží rýchlejšie ako štandardná implementácia CPython, pretože PyPy používa just-in-time kompilátor. Pypy nepodporuje niektoré grafické knižnice.

Poznámka 2: Úlohu je možné riešiť aj pomocou neurónovej siete, pričom je vhodné použiť nejaký framework (napríklad PyTorch).

Príklad vizualizácie: (rôzne pokusy vždy pre k=1, 3 a 7, 15)