

SUNS Cvičenie č.2

Ing. Pavol Marák

29. septembra 2015

Obsah

1	Zadanie č.1	2
1.1	Znenie zadania	2
1.2	Aproximácia funkcie MLP sieťou [4b]	2
1.3	Aproximácia funkcie RBF sieťou [4b]	2
1.4	Vzorový postup riešenia pre MLP sieť	3
1.5	Ukážka vzorovej dokumentácie k zadaniu	7
1.6	Funkcie na aproximáciu	8

1 Zadanie č.1

1.1 Znenie zadania

Úloha Použite viacvrstvový perceptrón (MLP) a RBF sieť na aproximáciu priebehu funkcie. Na implementáciu použite vhodný programovací prostriedok (Matlab prípadne implementácie neurónových sietí v iných jazykoch). Každý študent bude mať pridelenú vlastnú funkciu (konkrétne jej graf), ktorú bude aproximovať obidvomi sieťami. Je nutné graficky znázorniť výsledky aproximácie funkcie.

Termín odovzdania: Na cvičení v 4. týždni (12. - 16. október 2015)

1.2 Aproximácia funkcie MLP sieťou [4b]

1. Nájdite funkčný predpis vašej funkcie $f(x)$ z grafu. Vytvorte množinu bodov definičného oboru X a množinu prislúchajúceho oboru hodnôt Y . Vykreslite funkciu a porovnajte s dodaným grafom. **[0.5b]**
2. Vytvorte MLP sieť a nastavte počet neurónov skrytej vrstvy. **[0.5b]**
3. Rozdeľte údaje na trénovaciu množinu, validačnú množinu a testovaciu množinu (tzv. cross-validácia). Takto rozdelené údaje použite v procese trénovania, aby sa predišlo tzv. pretrénovaniu siete. **[0.5b]**
4. Natrénujte MLP sieť tak, aby sa naučila zobrazenie $y = f(x)$ dané grafom vašej funkcie. **[0.5b]**
5. Testujte sieť na testovacej vzorke (teda pre sieť neznámych údajoch, na ktorých sa sieť netrénovala). **[0.5b]**
6. Zobrazte v jednom grafe funkciu $f(x)$ a jej aproximáciu neurónovou sieťou. **[0.5b]**
7. Zobrazte grafický vývoj chyby na trénovacej a validačnej množine počas trénovania siete. V grafe jasne označte jednotlivé krivky. **[0.5b]**
8. Otestujte aspoň 3 rôzne konfigurácie MLP siete (počet neurónov, aktivačné funkcie, trénovacie funkcie, atď.). **[0.5b]**

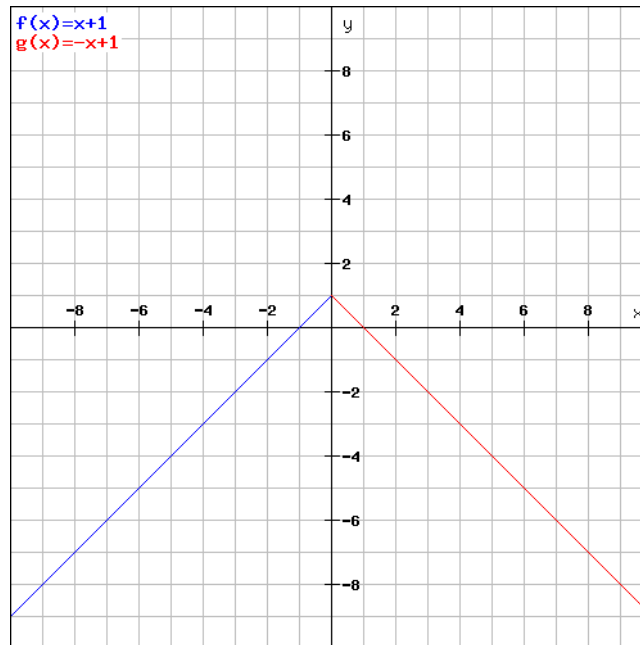
1.3 Aproximácia funkcie RBF sieťou [4b]

1. Nájdite funkčný predpis vašej funkcie $f(x)$ z grafu. Vytvorte množinu bodov definičného oboru X a množinu prislúchajúceho oboru hodnôt Y . Vykreslite funkciu a porovnajte s dodaným grafom. **[0.5b]**

2. Vytvorte a natrénujte RBF sieť. Nastavte cieľovú chybu aproximácie. **[1.5b]**
3. Zistite aký počet skrytých neurónov má sieť po netréňovaní. **[0.5b]**
4. Testujte sieť na ľubovoľnej testovacej vzorke. **[0.5b]**
5. Zobrazte v jednom grafe funkciu $f(x)$ a jej aproximáciu neurónovou sieťou. **[0.5b]**
6. Otestujte aspoň 3 rôzne konfigurácie RBF siete (rôzne voľby cieľovej chyby). **[0.5b]**

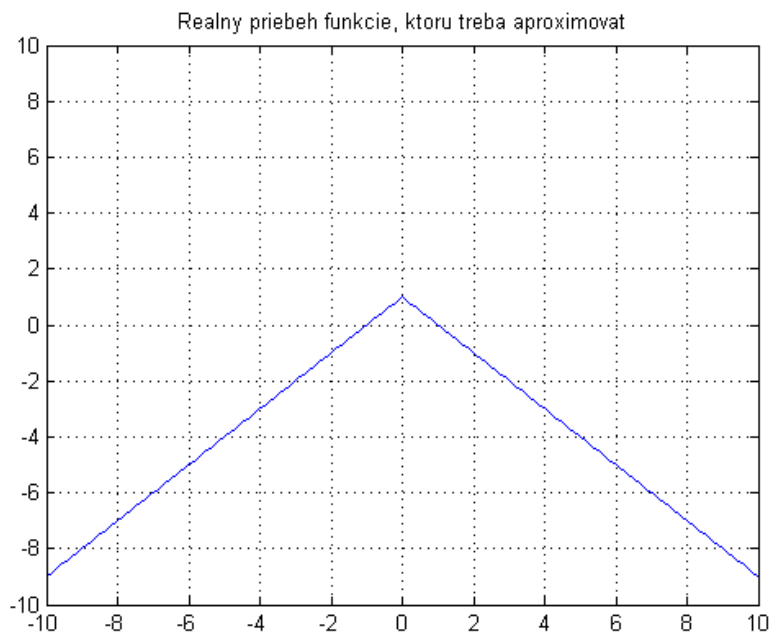
Je **nutné** odovzdať stručnú dokumentáciu s postupom vášho riešenia. Bez odovzdanej dokumentácie sa delia získané body na polovicu. Za kvalitnú dokumentáciu napísanú typografickým systémom \LaTeX môžete získať navyše 0,5 b **ako bonus**. Vaše riešenie spolu s dokumentáciou je potrebné nahráť do AIS.

1.4 Vzorový postup riešenia pre MLP sieť



Obr. 1: Vzorová funkcia, ktorú je treba aproximovať

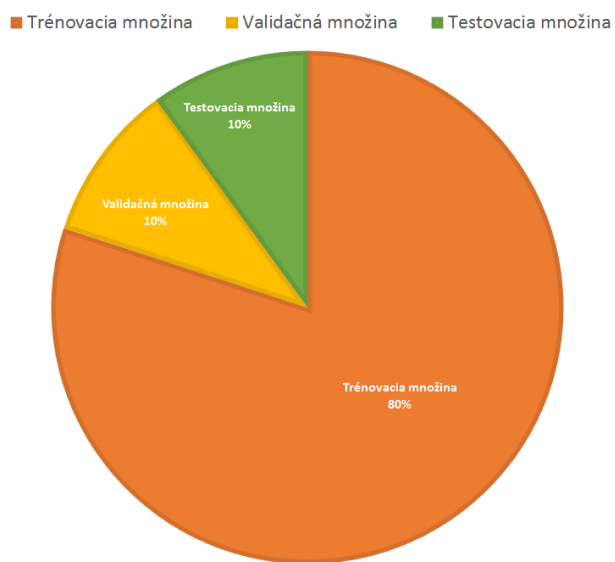
Vizuálnou analýzou zistíme, že graf reprezentuje 2 lineárne funkcie, funkcia vľavo $f(x)$ v intervale $[-10,0]$ a vpravo $g(x)$ v intervale $[0,10]$. Funkcia $f(x)$ má predpis $f(x) = x + 1$ a funkcia $g(x)$ má predpis $g(x) = -x + 1$. Keď poznáme predpis funkcie, vytvoríme vektor bodov X a k nemu prislúchajúci vektor funkčných hodnôt Y . Vykreslíme graf podľa dvojíc (X,Y) .



Obr. 2: Vykreslenie hľadanej funkcie v Matlabe

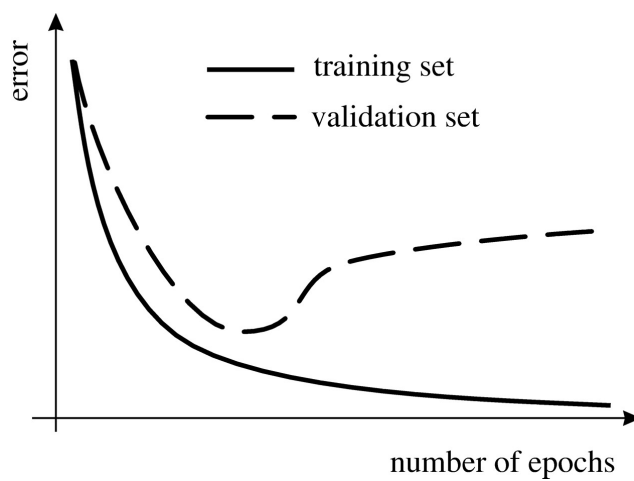
Nasleduje programátorská časť, kde vytvoríme MLP sieť s 1 vrstvou a nastavíme jej počet neurónov.

Množiny X a Y rozdelíme na trénovaciu množinu (80 %), validačnú množinu (10 %) a testovaciu množinu (10 %).

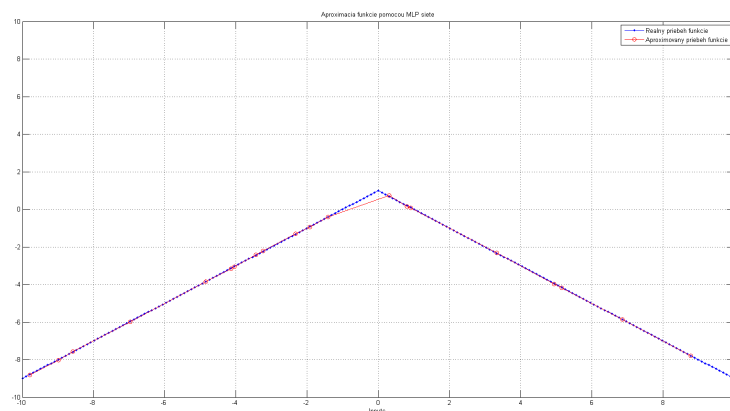


Obr. 3: Rozdelenie údajov pre cross-validáciu

Na trénovacej množine sa sieť učí, teda nastavuje svoje váhy. Na validačnej množine sa počas tréningu kontroluje, či sieť nie je "pretrénovaná". Keď prestane chyba na validačnej množine klesať, je vhodné tréning zastaviť.

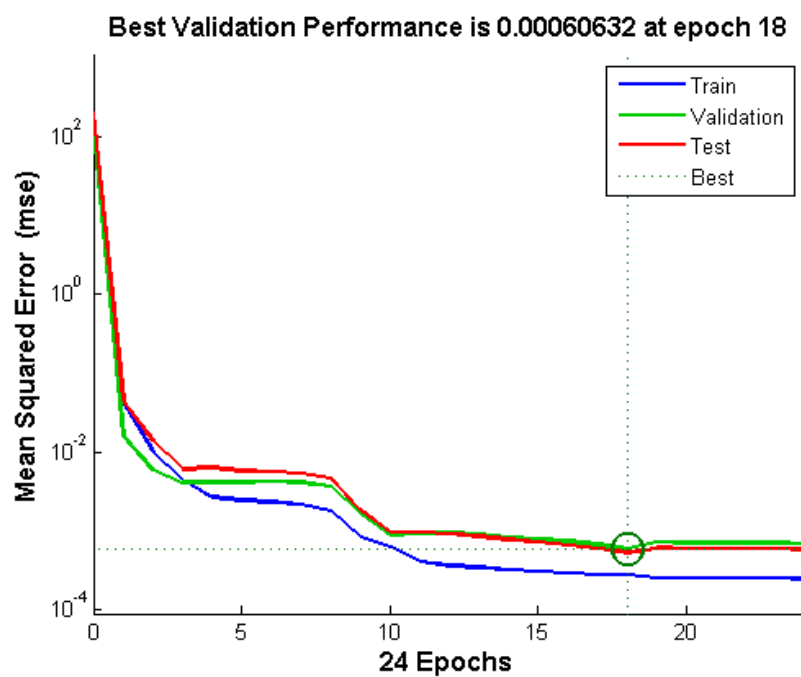


Obr. 4: Ukážka vývoja chyby na trénovacej a validačnej množine



Obr. 5: Ukážka grafu znázorňujúceho originálnu a aproximovanú funkciu

Sieť natrénujeme a zároveň otestujeme na testovacej množine (t.j. údaje, ktoré sieť nepozná). Výstup siete po testovaní použijeme na vizualizáciu aproximovanej funkcie.



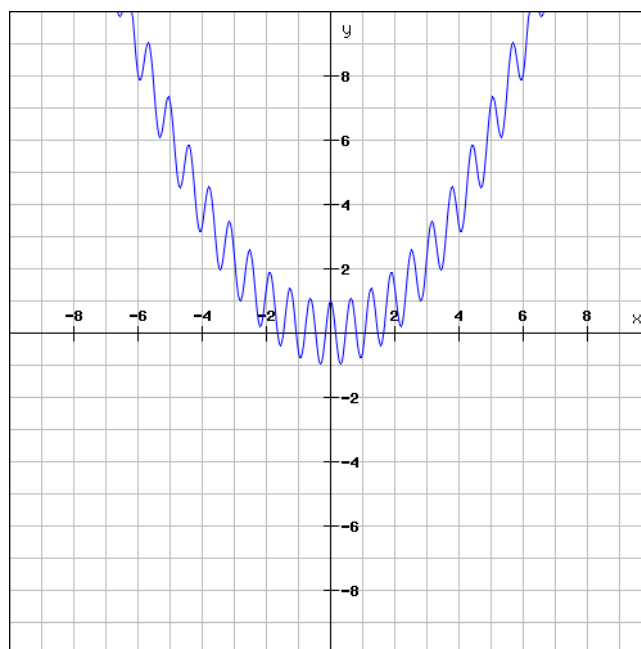
Obr. 6: Vývoj chyby na trénovacej a validačnej množine

Vytvorte aspoň 3 rôzne konfigurácie MLP siete (zvyšujte napr. počet neurónov v skrytej vrstve a porovnajte presnosť aproximácie funkcie so rastúcim počtom neurónov).

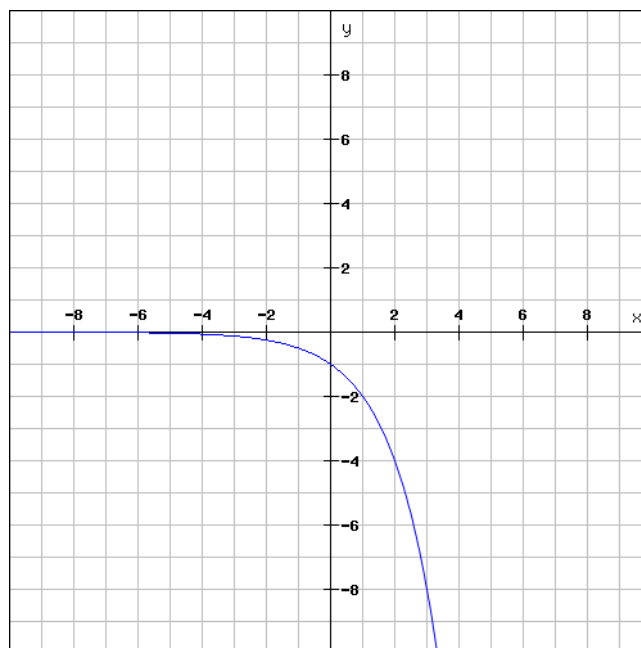
1.5 Ukážka vzorovej dokumentácie k zadaniu

Doplňí sa neskôr.

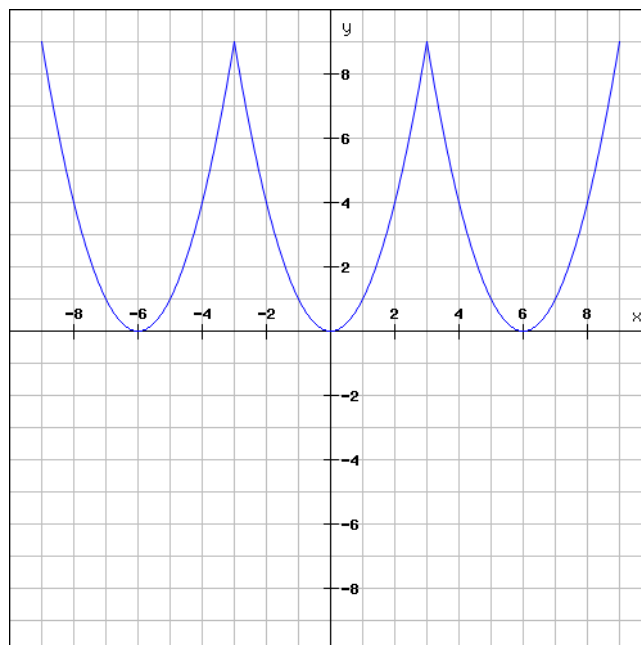
1.6 Funkcie na aproximácii



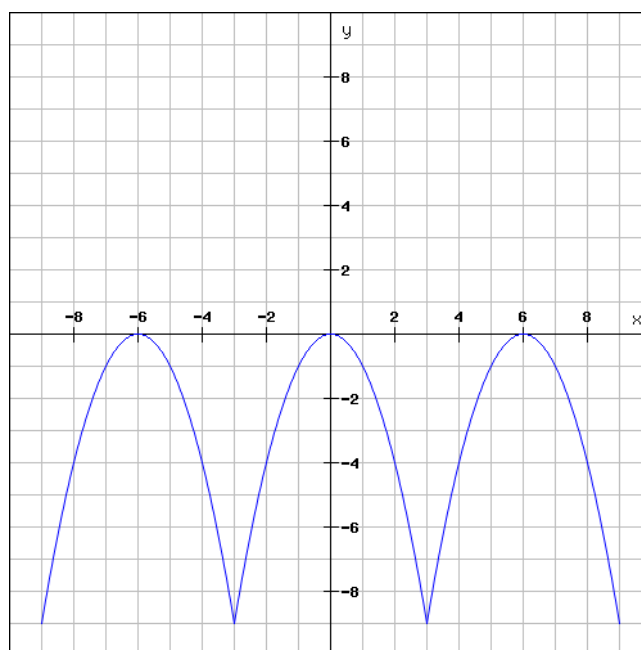
Obr. 7: Funkcia 1



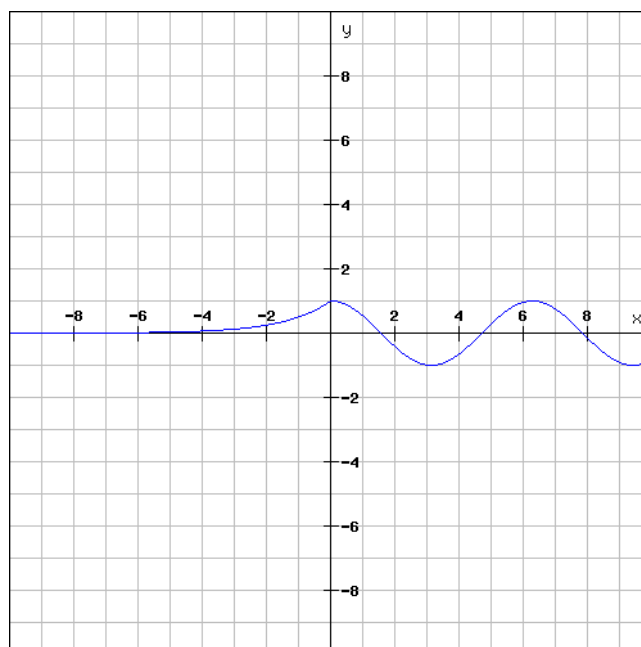
Obr. 8: Funkcia 2



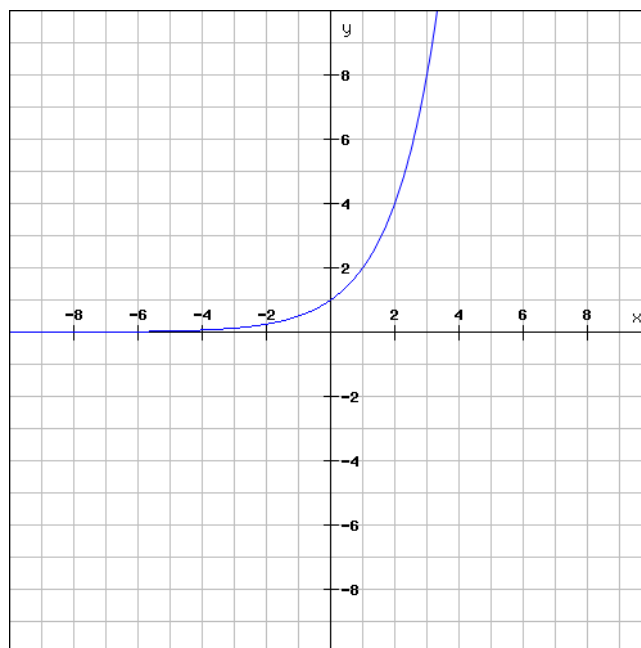
Obr. 9: Funkcia 3



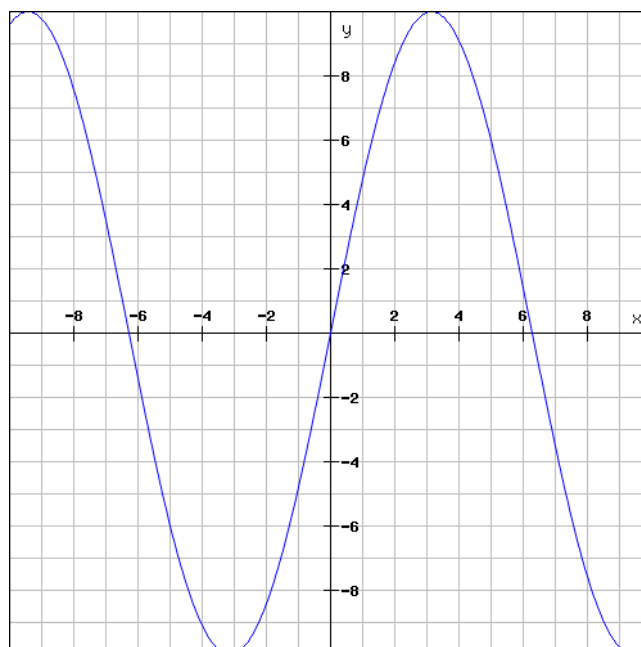
Obr. 10: Funkcia 4



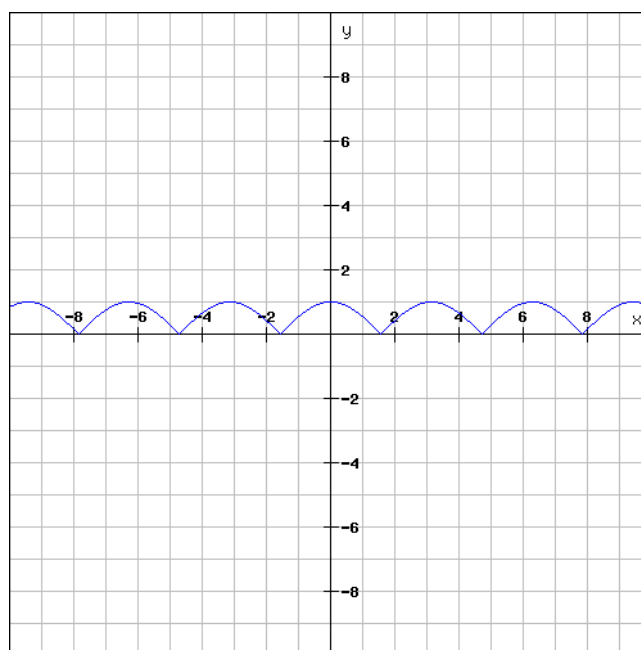
Obr. 11: Funkcia 5



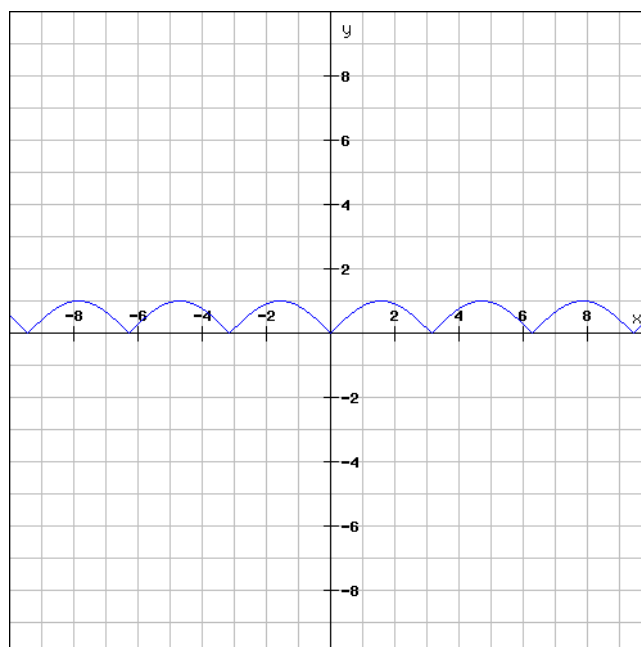
Obr. 12: Funkcia 6



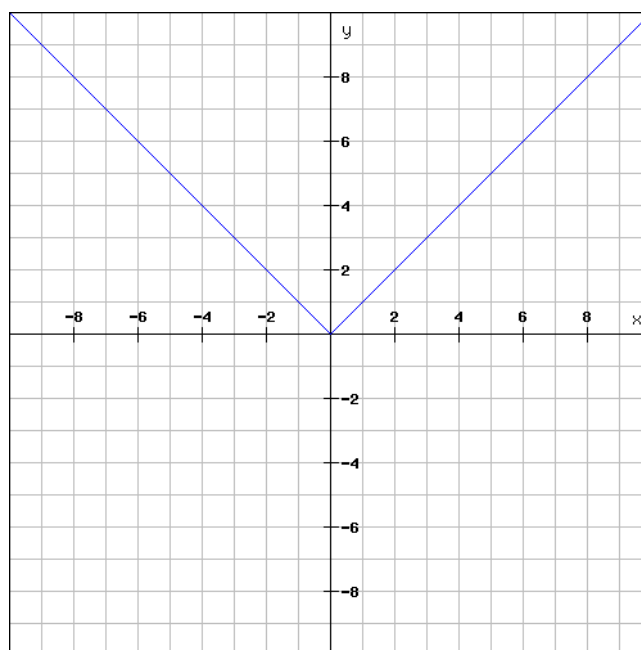
Obr. 13: Funkcia 7



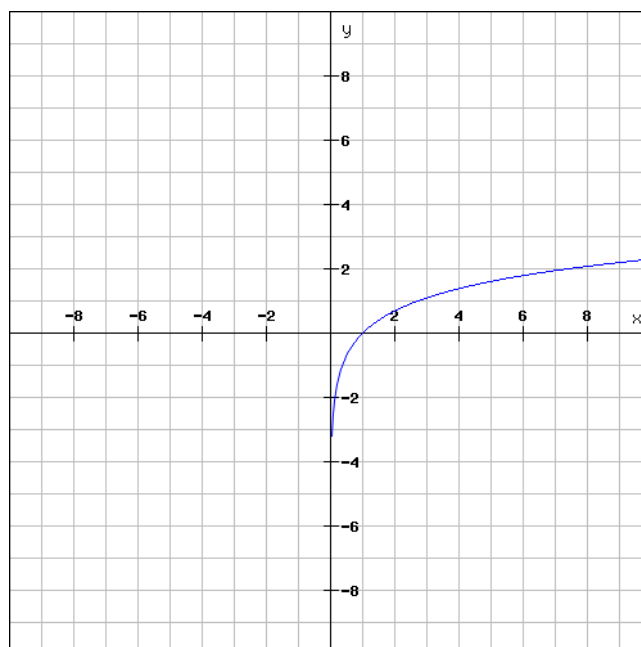
Obr. 14: Funkcia 8



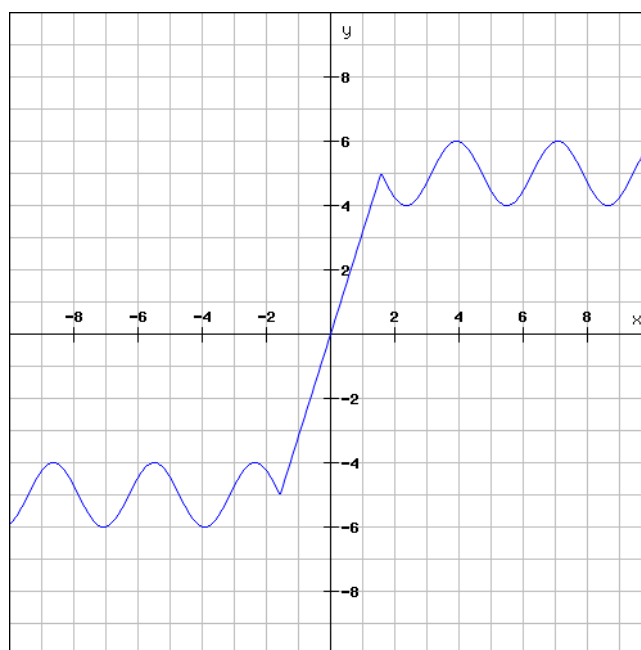
Obr. 15: Funkcia 9



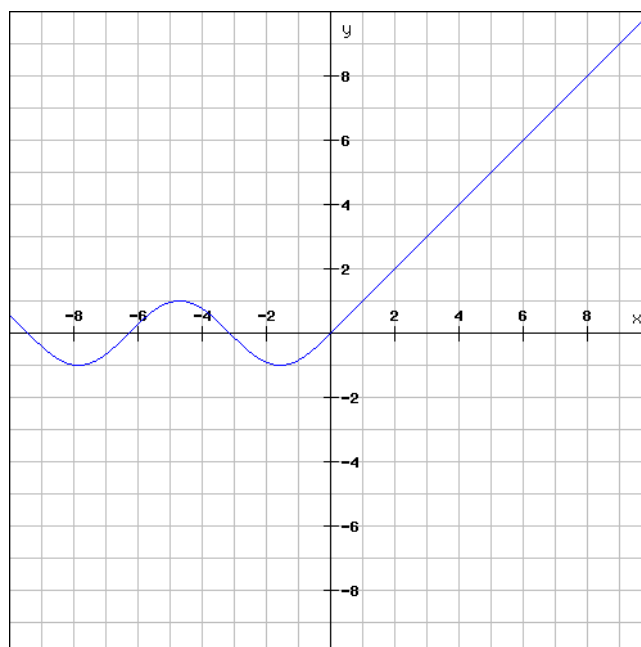
Obr. 16: Funkcia 10



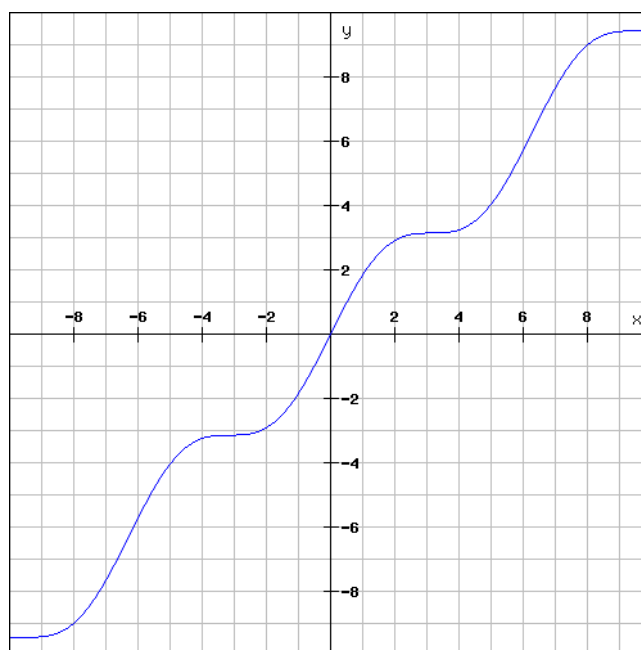
Obr. 17: Funkcia 11



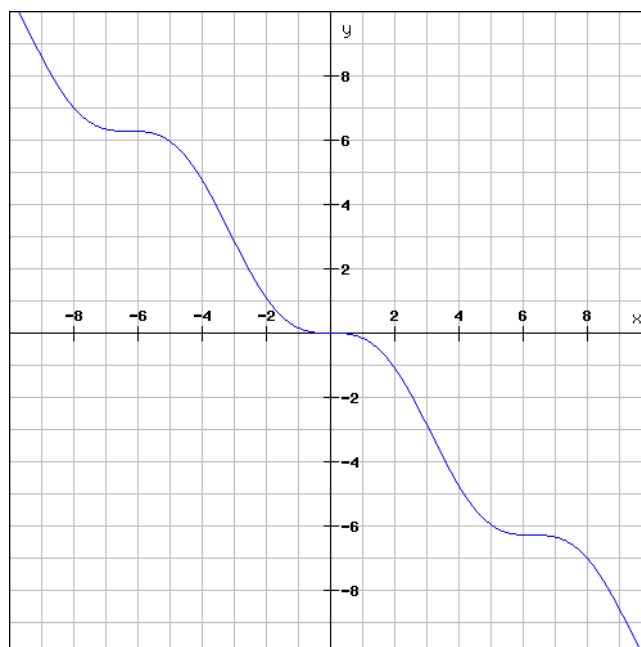
Obr. 18: Funkcia 12



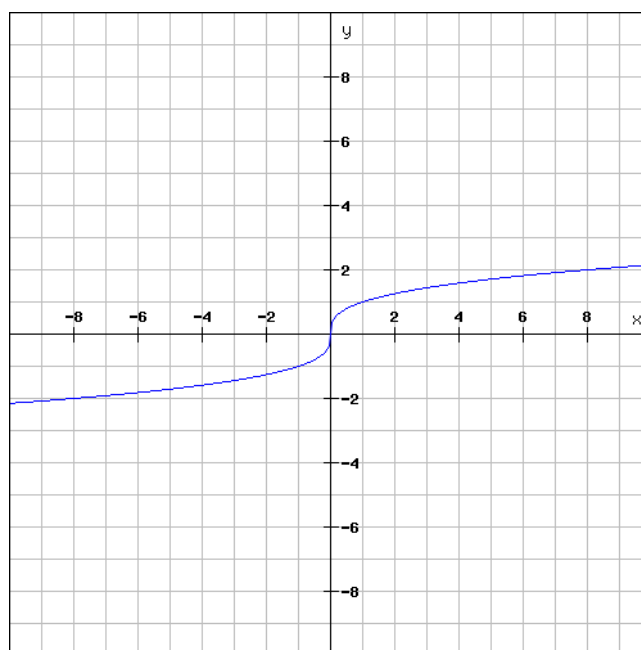
Obr. 19: Funkcia 13



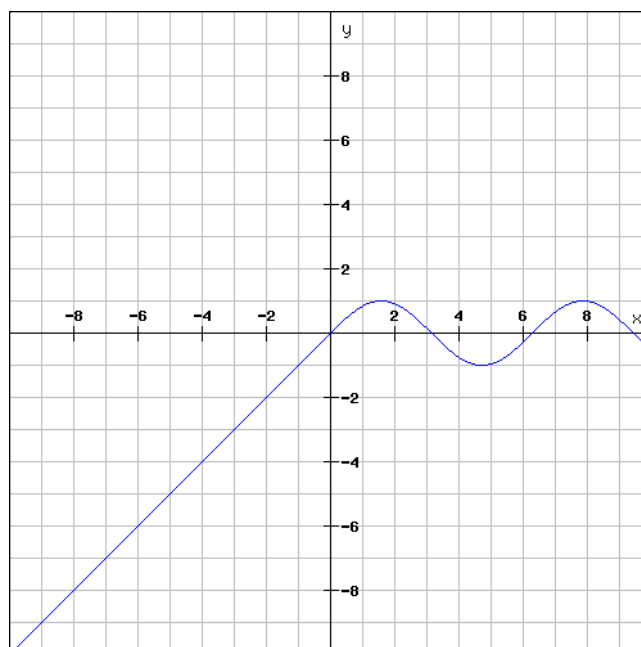
Obr. 20: Funkcia 14



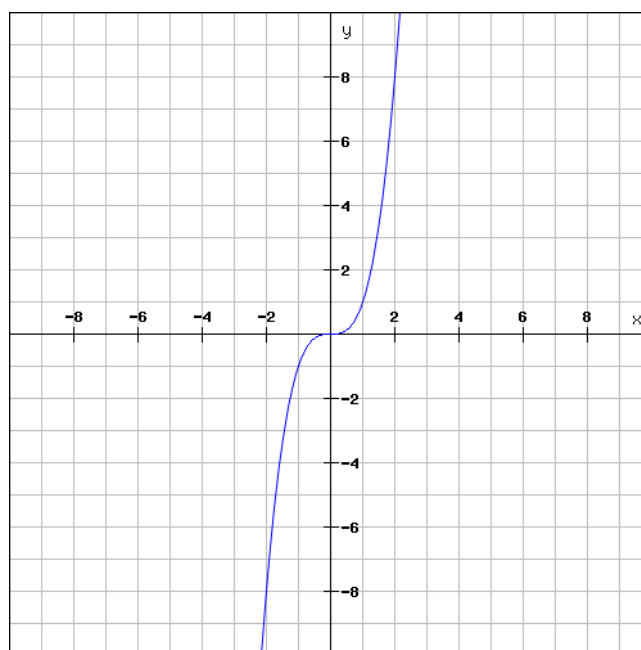
Obr. 21: Funkcia 15



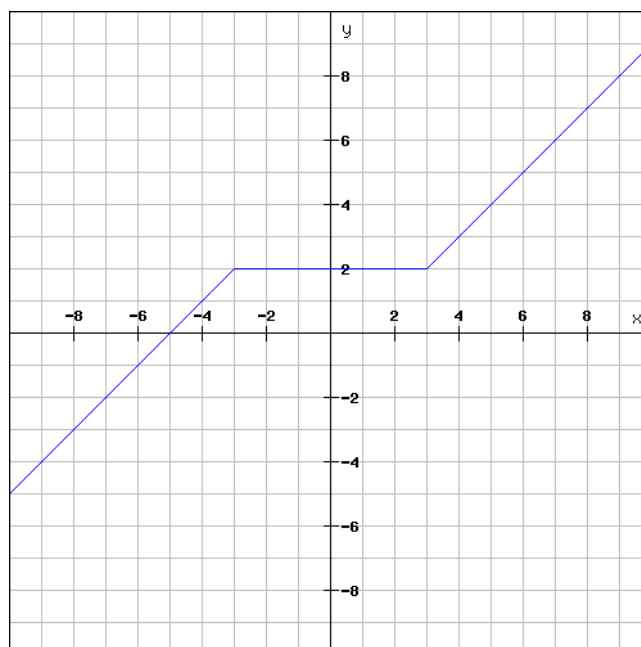
Obr. 22: Funkcia 16



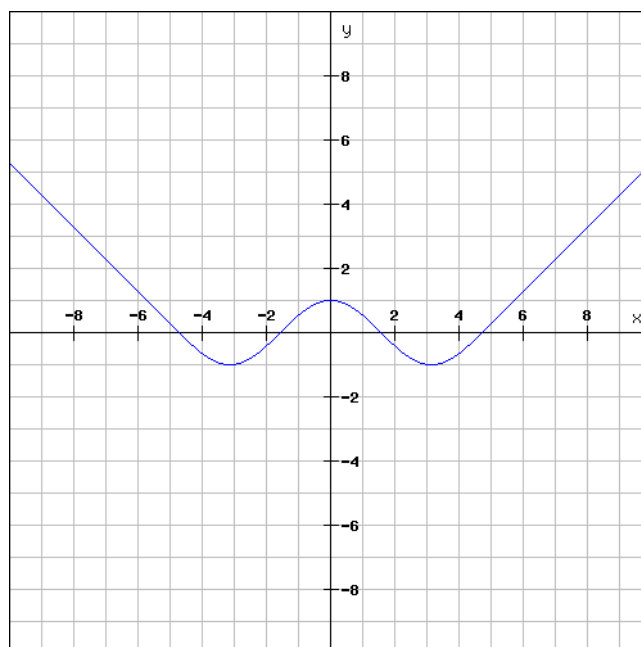
Obr. 23: Funkcia 17



Obr. 24: Funkcia 18



Obr. 25: Funkcia 19



Obr. 26: Funkcia 20