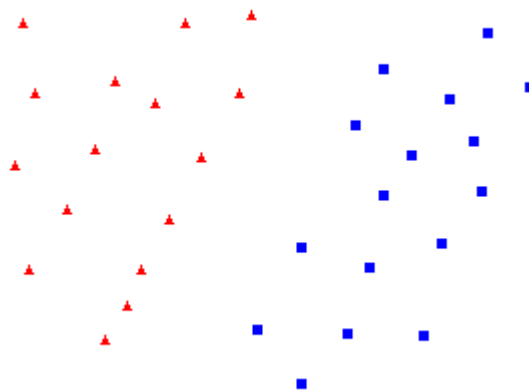


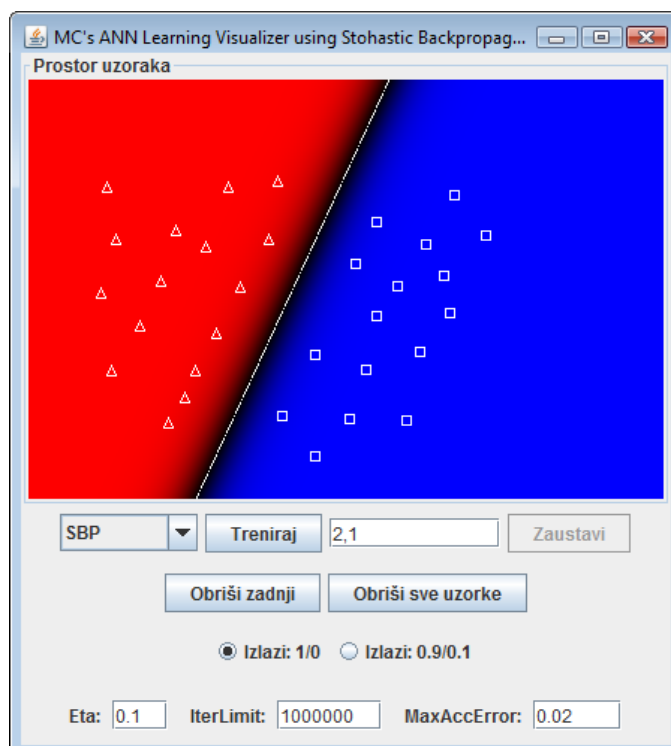
Neuronske mreže i prikladnost arhitekture mreže

U nastavku ovog dokumenta razmotrit ćemo dva zadatka postavljena pred neuronsku mrežu: klasifikacija te funkcijska aproksimacija. Dat ćemo nekoliko primjera uzoraka za učenje te različito ekspresivne neuronske mreže i pogledati kakav je rezultat učenja.



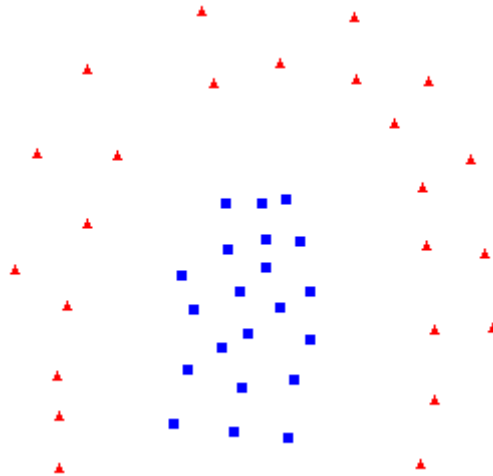
Slika 1. Skup uzoraka za klasifikaciju koji je moguće razdijeliti linearnom decizijskom funkcijom

Skup uzoraka na slici 1 moguće je razdijeliti linearnom decizijskom funkcijom. Neuronska mreža arhitekture 2x1 može obaviti taj posao, što je prikazano na slici 2.



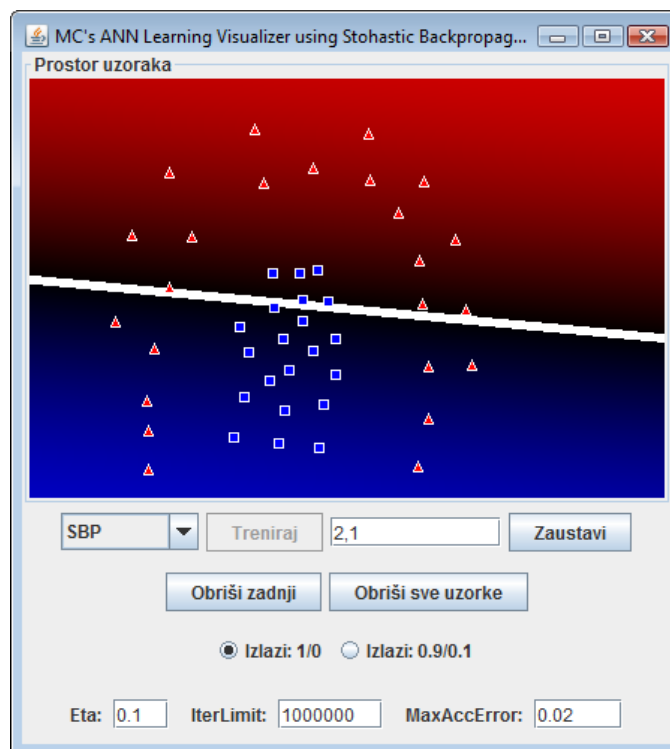
Slika 2. Decizijska granica (bijela linija) naučena mrežom 2x1.

Malo složeniji slučaj prikazan je na slici 3. Prikazane uzorke nije moguće razdijeliti linearnom decizijskom funkcijom.



Slika 3. Linearno neodvojiv skup uzoraka za učenje

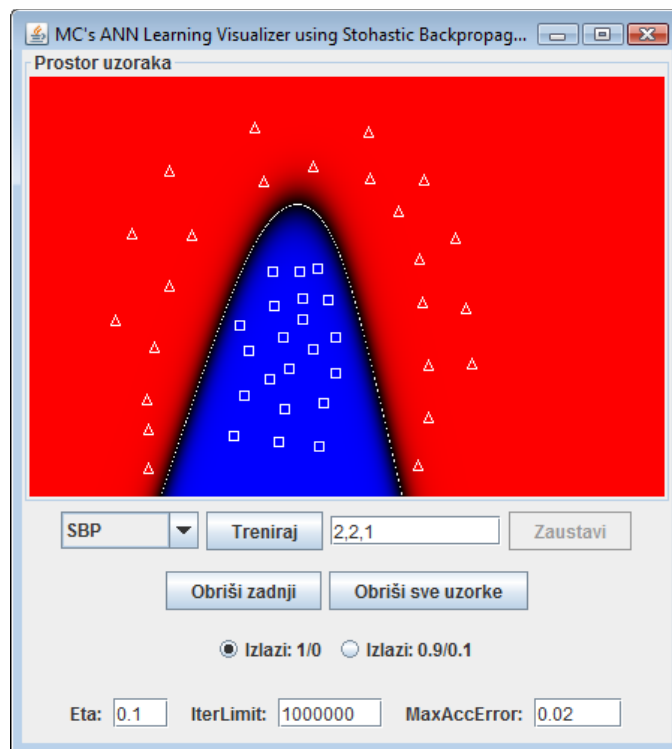
Pokušamo li naučiti mrežu arhitekture 2x1, mreža će naučiti decizijsku funkciju koja minimizira ukupnu pogrešku, ali koja nije sposobna korektno klasificirati sve uzorke (slika 4).



Slika 4. Naučena linearna decizijska funkcija

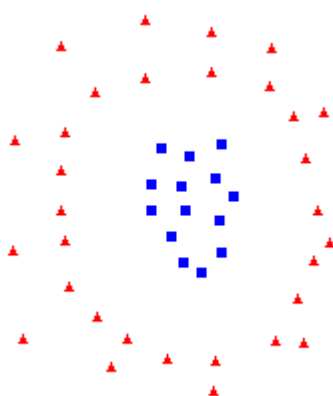
Postupak učenja ovdje nikada neće stati zbog dostizanja zadane minimalno prihvatljive pogreške, već će se zaustaviti kada se dosegne zadan broj epoha (1 000 000).

Povećanje ekspresivnosti mreže će pomoći. Mreža s jednim skrivenom slojem u kojem se nalaze dva neurona može oblikovati decizijsku funkciju koja izgleda "parabolično": za ovaj slučaj to je dovoljno. Decizijska granica koja je naučena mrežom arhitekture 2x2x1 prikazana je na slici 5 danoj u nastavku dokumenta.



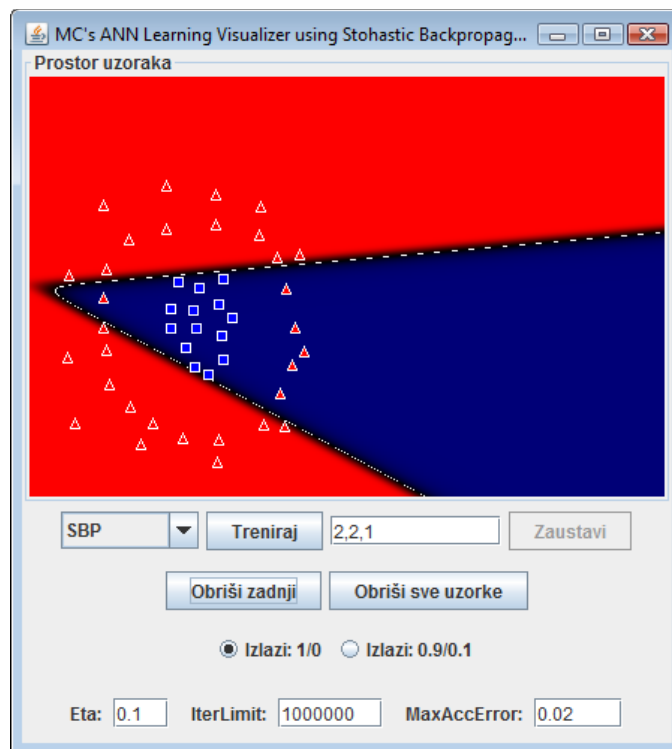
Slika 5. Naučena decizijska funkcije uz mrežu 2x2x1.

Pogledajmo još složeniji skup uzoraka za učenje, koji je prikazan na slici 6.



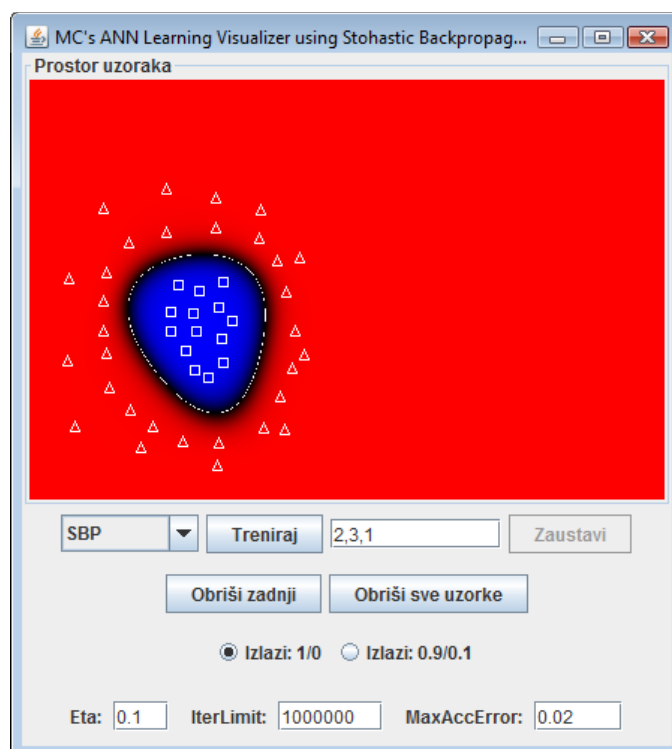
Slika 6. Ugniježdeni razredi

U ovom slučaju, niti mreža 2x2x1 neće moći korektno klasificirati sve uzorke (slika 7).



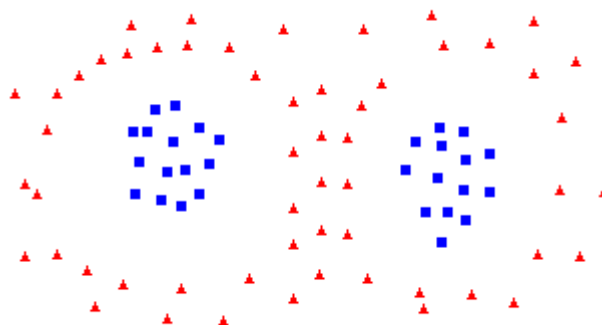
Slika 7. Decizijska funkcija mreže 2x2x1.

Dodavanje još jednog neurona u skriveni sloj (čime se povećava ekspresivnost mreže) može pomoći, što je prikazano na slici 8.



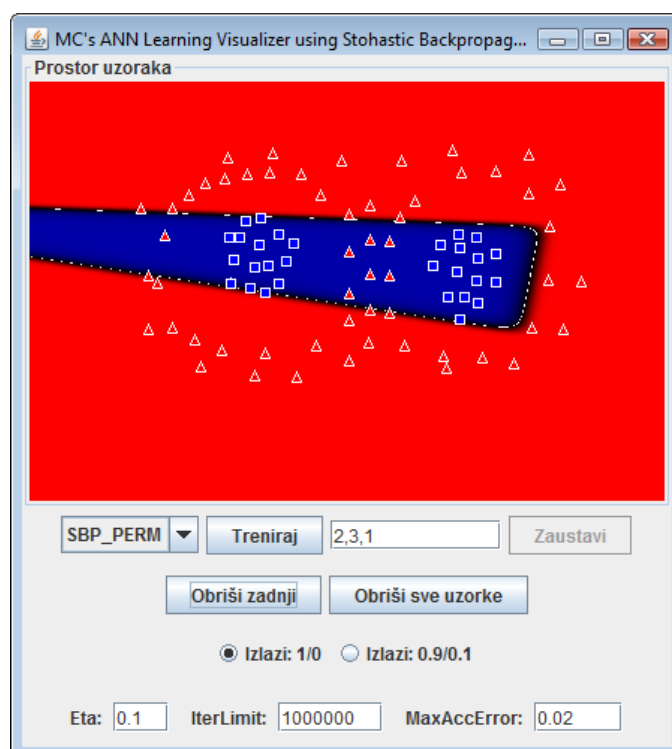
Slika 8. Korektna klasifikacija postiže se uz mrežu arhitekture 2x3x1.

Razmotrimo sada još malo kompleksniji skup uzoraka za učenje, koji je prikazan na slici 9.



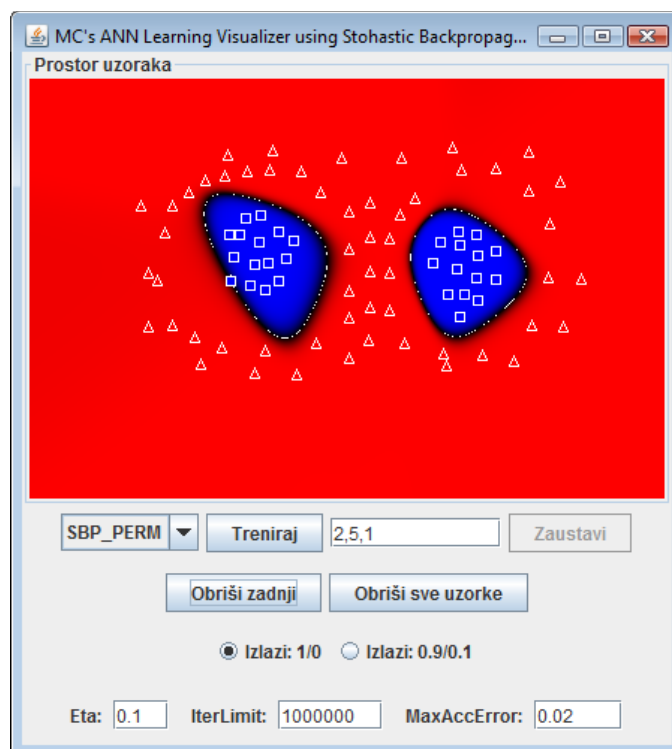
Slika 9. Još složeniji skup uzoraka za učenje

Mreža arhitekture 2x3x1 ne može naučiti korektnu klasifikaciju jer nije dovoljno ekspresivna (slika 10).



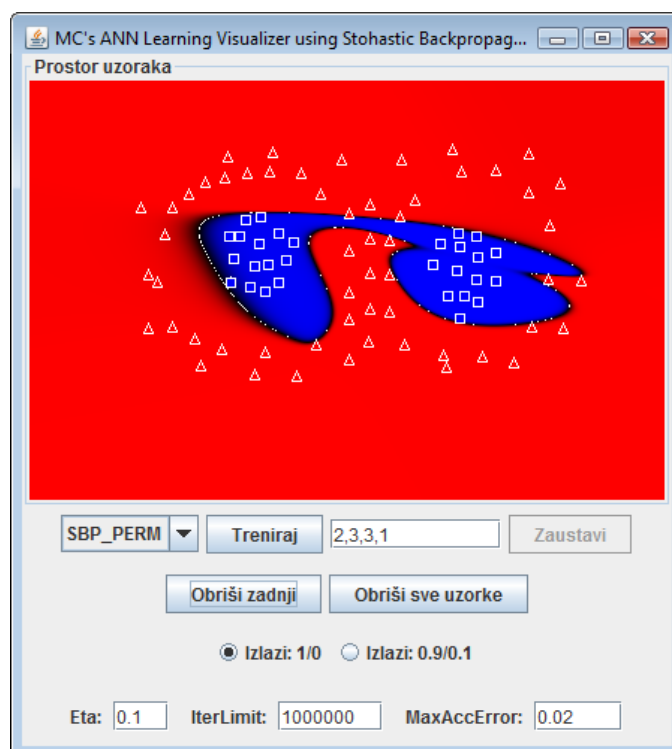
Slika 10. Decizijska granica naučena kod mreže 2x3x1.

Stoga ćemo pokušati s nešto ekspresivnijim mrežama. Najprije ćemo u skriveni sloj dodati još dva neurona čime ćemo dobiti mrežu arhitekture 2x5x1. Ova mreža može naučiti obavljati klasifikaciju bez pogreške, što je prikazano na slici 11.

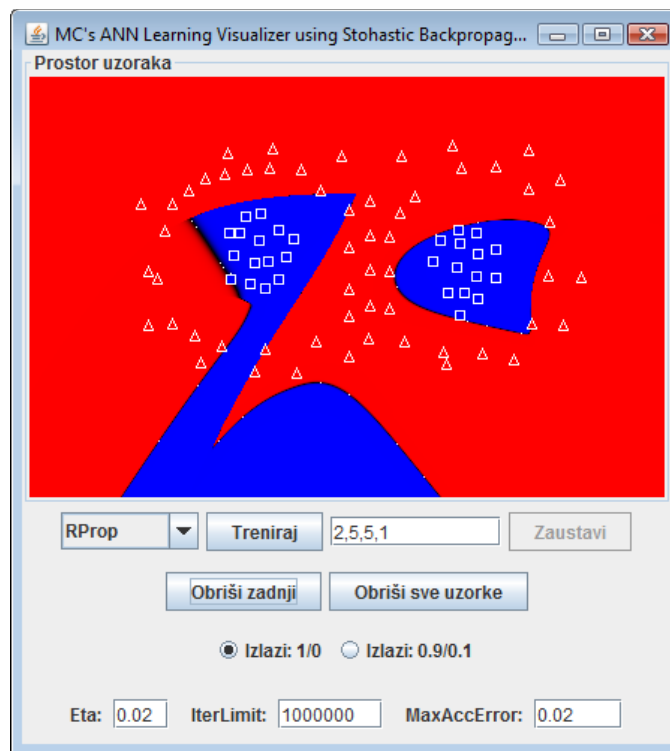


Slika 11. Mreža 2x5x1 može modelirati ovakvu distribuciju.

Sada ćemo pogledati još malo ekspresivnije mreže: arhitekturu 2x3x3x1 (što je neuronska mreža koja ima dva skrivena sloja, i u oba ima po tri neurona; slika 12) te arhitekturu 2x5x5x1 (što je neuronska mreža koja ima dva skrivena sloja, i u oba ima po pet neurona; slika 13). U oba slučaja izgled naučene decizijske funkcije razlikovst će se od pokretanja do pokretanja algoritma učenja. Ponekad, decizijska granica izgledat će kao na slikama 12 i 13: mreža će savršeno raditi klasifikaciju ali će uslijed prevelike ekspresivnosti vrlo loše generalizirati. Pogledajte primjerice sliku 12: u središnjem području za koje bismo bez razmišljanja rekli da su tamo svi uzorci crveni, mreža će za neke dijelove uzorke proglašavati plavima. Naime, u uskupu uzoraka za učenje nema uzoraka koji bi to opovrgnuli (mreža je savršeno naučila), ali kako ima veliku ekspresivnu moć, ona je “zamislila” da je tamo područje plavih uzoraka.



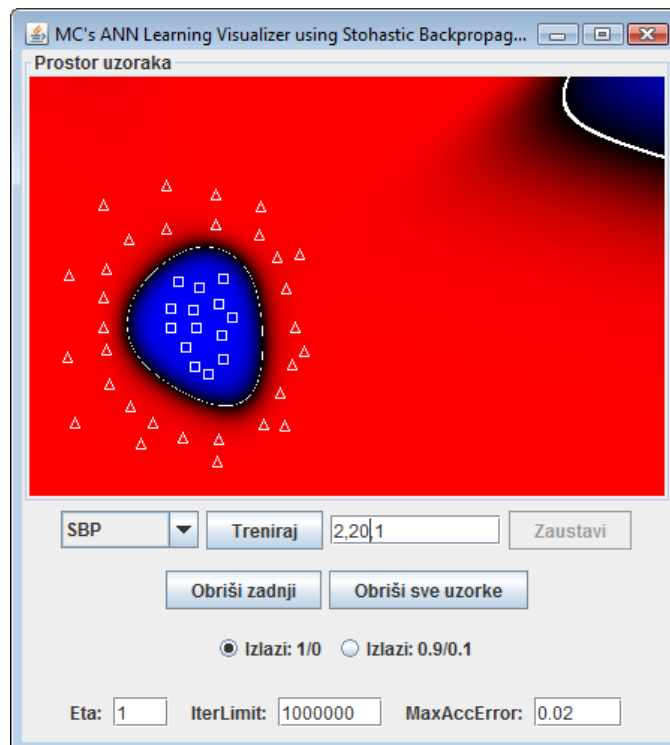
Slika 12. Mreža 2x3x3x1 može korektno klasificirati prikazani skup uzoraka



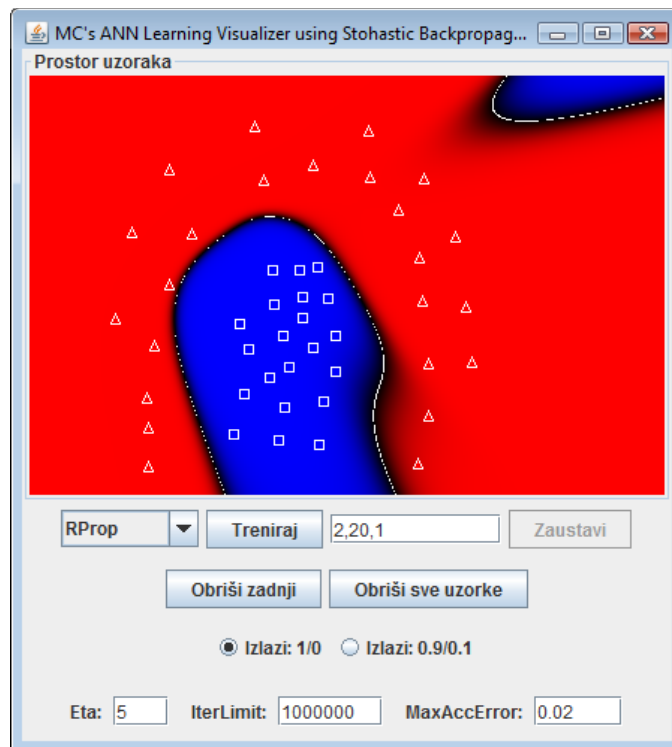
Slika 13. Mreža 2x5x5x1 može korektno klasificirati prikazani skup uzoraka

Slika 13 ovo ilustrira još bolje: završili smo s naučenom decizijskom funkcijom koja u donjem dijelu prostora velik dio klasificira kao područje plavih uzoraka, iako u tom području nije bilo niti jednog plavog elementa na temelju kojega bi se donio takav zaključak.

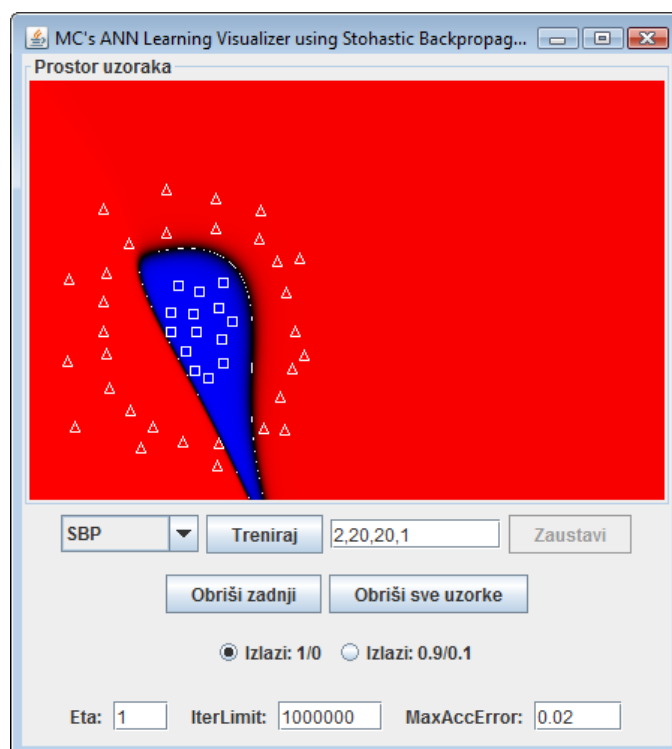
Slične probleme sa sposobnošću generalizacije vidimo i na slikama 14 do 16.



Slika 14. Mreža 2x20x1 i loša generalizacija

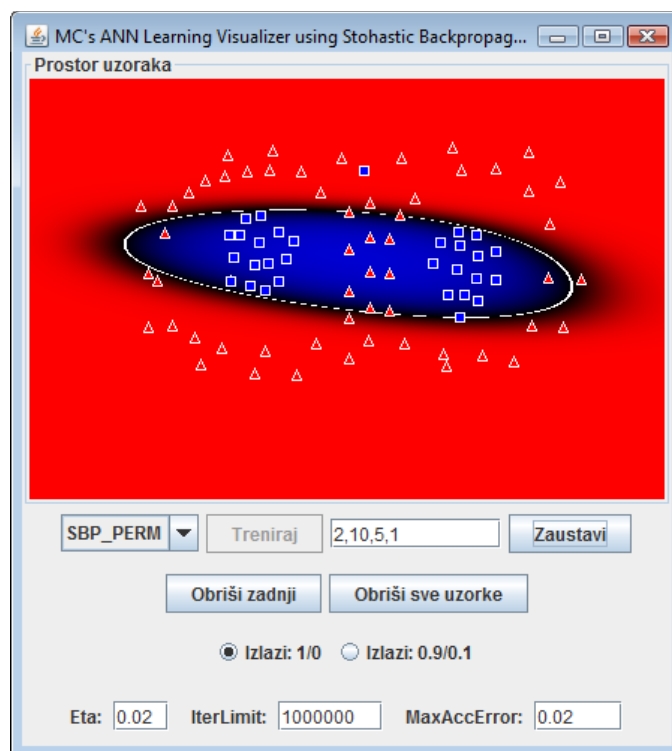


Slika 15. Mreža 2x20x1 i loša generalizacija



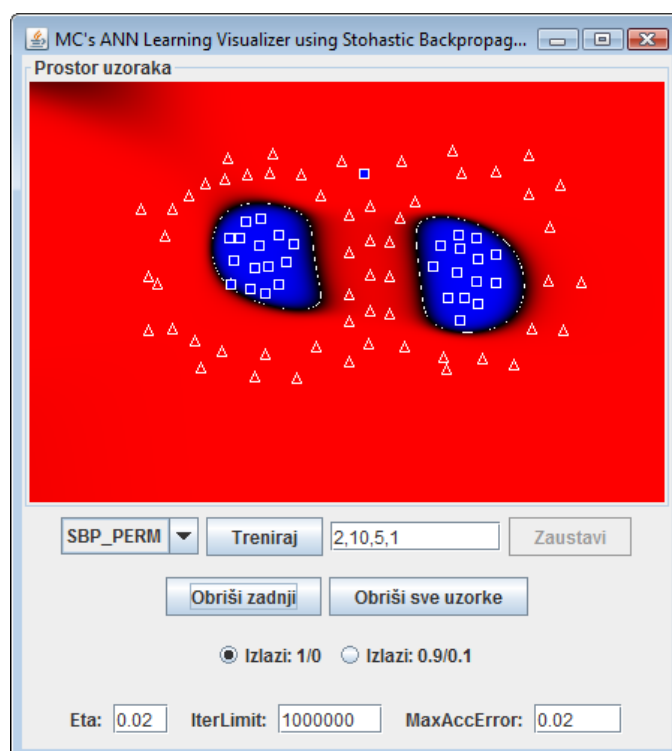
Slika 16. Mreža 2x20x20x1 i loša generalizacija

Pretpostavimo sada da smo u uzorke za učenje dodali jedan pogrešan primjer (engl. *Outlier*); modeliramo pogrešku u prikupljanju podataka (vidi sliku 17). Pokrenimo učenje neuronske mreže 1x10x5x1. Nakon određenog broja epoha, postupak učenja će doći do decizijske funkcije koja je prikazana na slici 17. Jasno se vidi da je decizijska funkcija loša, ali mreža već poprilično dobro raspoznaje vanjski dio crvenih elemenata.



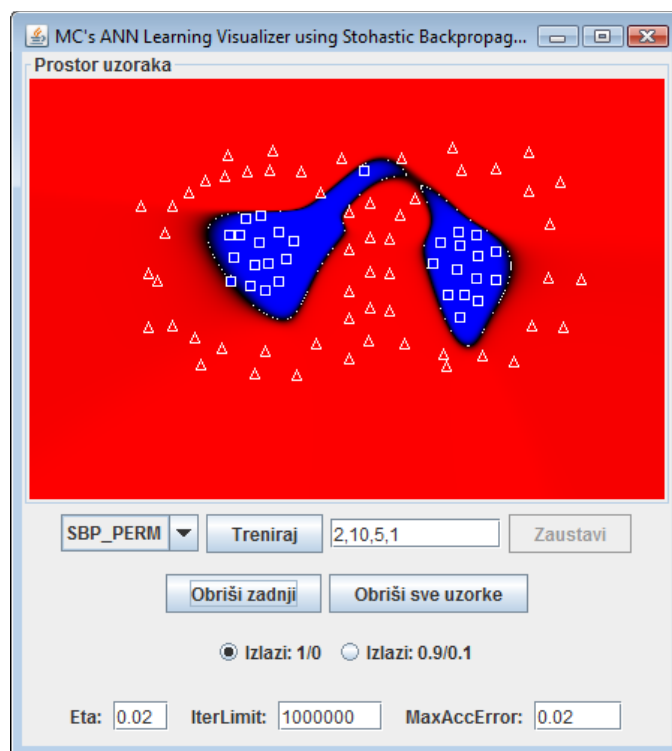
Slika 17. Mreža 2x10x5x1: stanje u ranijoj fazi učenja

Nastavkom treniranja, mreža će doći do decizijske funkcije koja je prikazana na slici 18. Mreža jasno raspoznaje dva velika područja u kojima se nalaze plavi elementi. Međutim, strogo se oslanjajući na informaciju koju daje funkcija pogreška, ova mreža još uvijek griješi: postoji jedan plavi element (u gornjem dijelu slike) koji mreža “pogrešno” klasificira kao crveni.



Slika 18. Mreža 2x10x5x1: stanje u srednjoj fazi učenja

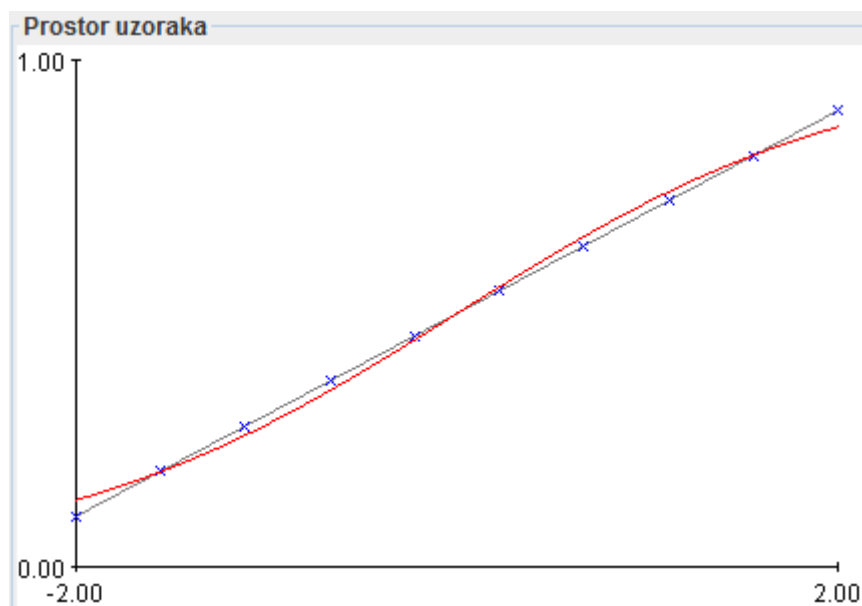
Ako je mreža dovoljno ekspresivna, ona će pronaći načina da si i taj element “objasni” (odnosno da ga naštreba, ma koliko to bilo besmisleno). Konačna decizijska funkcija koja prikazuje upravo jedan takav ishod prikazana je na slici 19. S obzirom da je mreža dovoljno ekspresivna, uspjela je modelirati decizijsku funkciju koja vrlo loše generalizira videne primjere ali fantastično reproducira skup uzoraka za učenje. To je nešto što u praksi često smatramo kao loše svojstvo.



Slika 19. Mreža 2x10x5x1 i pretreniranost

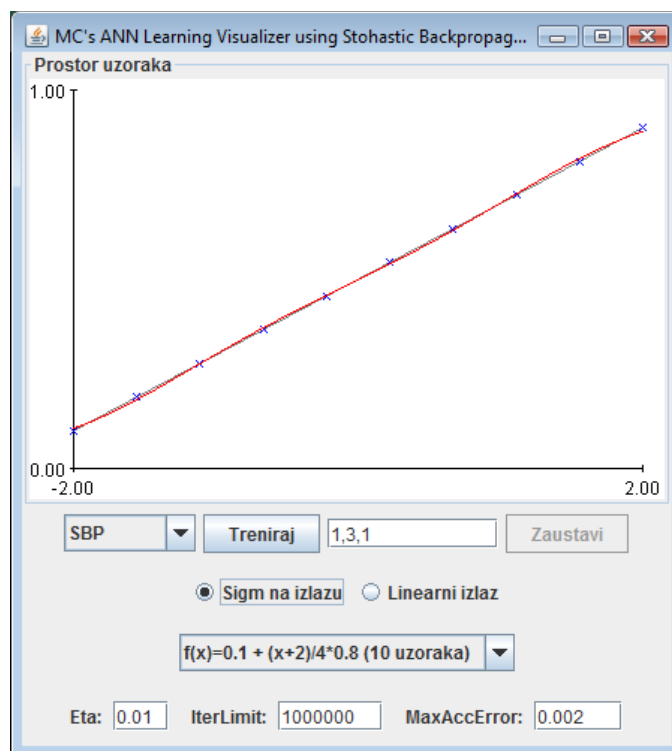
Pogledajmo sada isto na primjeru neuronske mreže koju učimo aproksimirati zadanu funkciju na temelju uzoraka za učenje. Na slikama koje slijede, originalna funkcija iz koje su uzorci izvučeni prikazana je sivom bojom, uzorci su prikazani križićima a funkcija koju je naučila neuronska mreža crvenom bojom.

Slika 20 pokazuje prvi slučaj: izvukli smo 10 uzoraka uniformno iz funkcije $f(x)=0.1+0.8*(x+2)/4$ i učili neuronsku mrežu 1x1 (jedan ulaz, jedan sigmoidalni neuron na izlazu). Kako je njegova prijenosna funkcija upravo sigmoida, na slici je lijepo vidi da postoje manja odstupanja od zadanih uzoraka, ali mreža je lijepo uhvatila "trend" u podacima. Uočite da su vrijednosti funkcije koju učimo ograničene u interval $[0, 1]$ jer sigmoidalni neuron na izlazu ne može dati vrijednosti koje su izvan tog intervala.



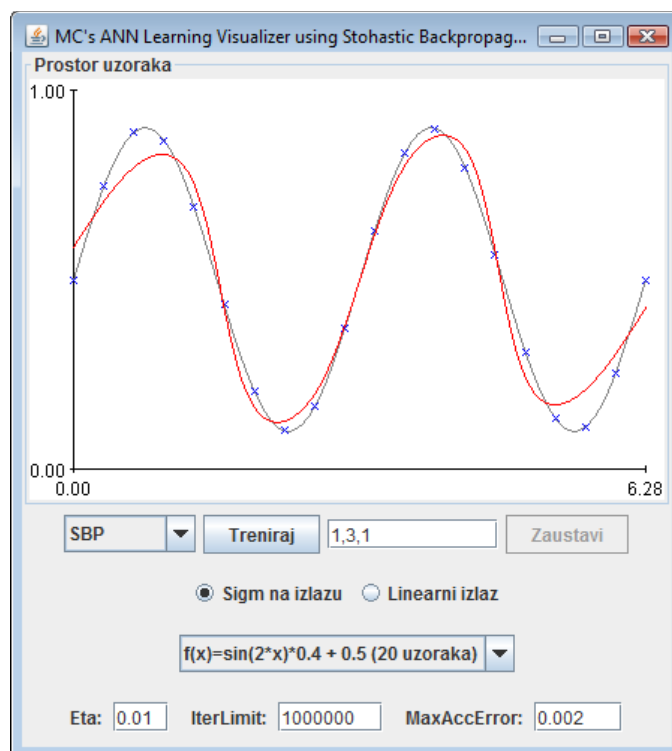
Slika 20. Mreža 1x1 i modeliranje linearne funkcije na temelju 10 uzoraka

Uzmemo li mrežu koja je nešto kompleksnija, dobit ćemo bolji rezultat (slika 21).



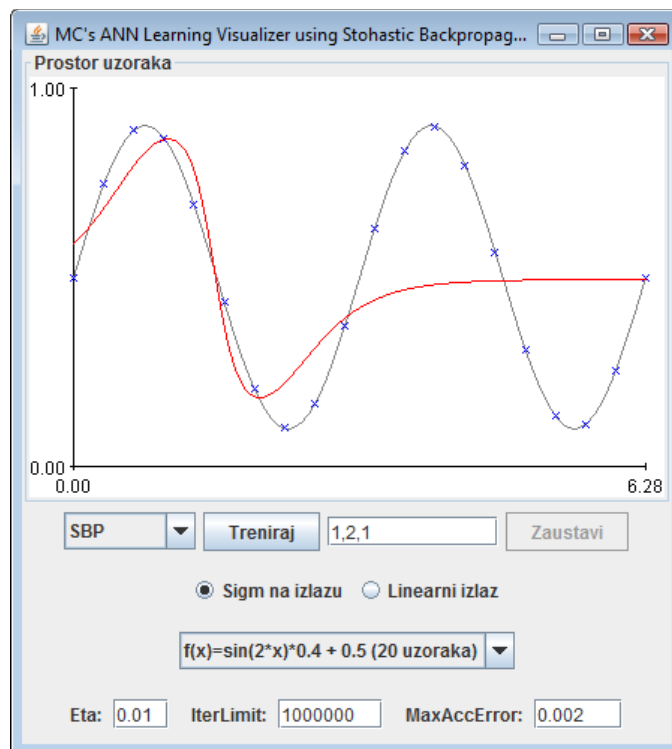
Slika 21. Mreža 1x3x1 i modeliranje linearne funkcije na temelju 10 uzoraka

Mrežu 1x3x1 možemo isprobati i na složenijem primjeru: modeliranje dvije periode sinusoidalne funkcije (skalirane i translaticirane u interval [0, 1]) na temelju 20 uzoraka. Rezultat je prikazan na slici 22.



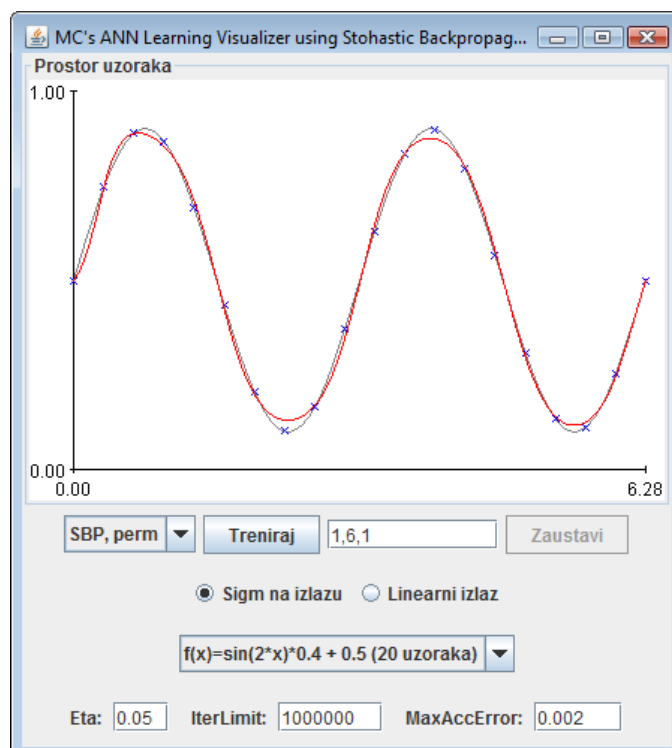
Slika 22. Mreža 1x3x1 i modeliranje dvije periode sinusoidalne funkcije na temelju 20 uzoraka

Kao što vidimo, mreža ima dosta problema pri modeliranju iako pogreška i nije prevelika. Jednostavnija neuronska mreža od ove naprosto nije dovoljno ekspresivna kako bi modelirala dane podatke. Rezultat učenja mreže 1x2x1 na istim podacima prikazan je na slici 23.



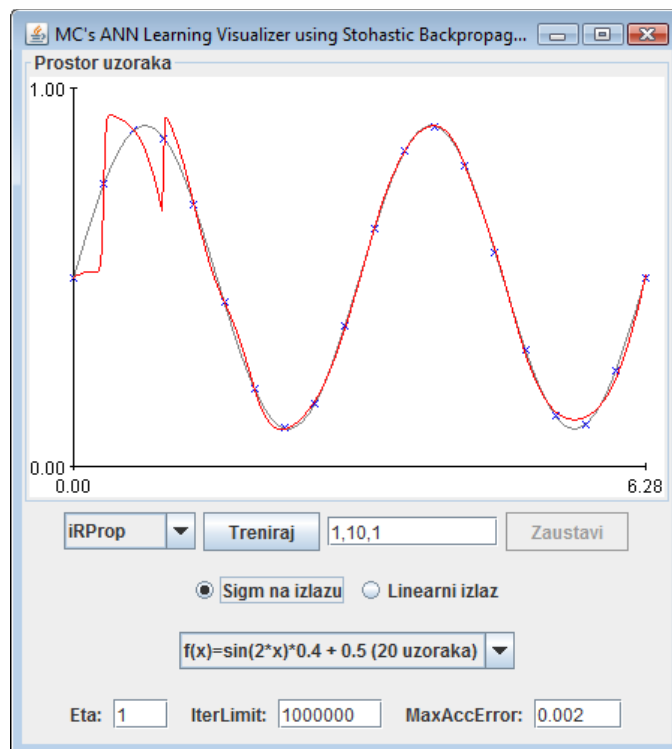
Slika 23. Mreža 1x2x1 i modeliranje dvije periode sinusoidalne funkcije na temelju 20 uzoraka

Povećamo li ekspresivnost mreže, dobit ćemo bolje rezultate. Rezultat učenja mreže 1x6x1 na istim podacima prikazan je na slici 24. Uočite da je rezultat bolji od onoga dobivenog mrežom 1x3x1.



Slika 24. Mreža 1x6x1 i modeliranje dvije periode sinusoidalne funkcije na temelju 20 uzoraka

Sljedeći eksperiment koji ćemo napraviti jest dodatno povećanje ekspresivnosti mreže. Razmotrit ćemo mrežu 1x10x1 na istom aproksimacijskom problemu. Jedan mogući rezultat učenja prikazan je na slici 25. Kao što vidimo, mreža je preekspresivna u odnosu na skup uzoraka za učenje i ima problema s generalizacijom, što je lijepo vidljivo iz naučene funkcije koja je prikazana crvenom linijom.



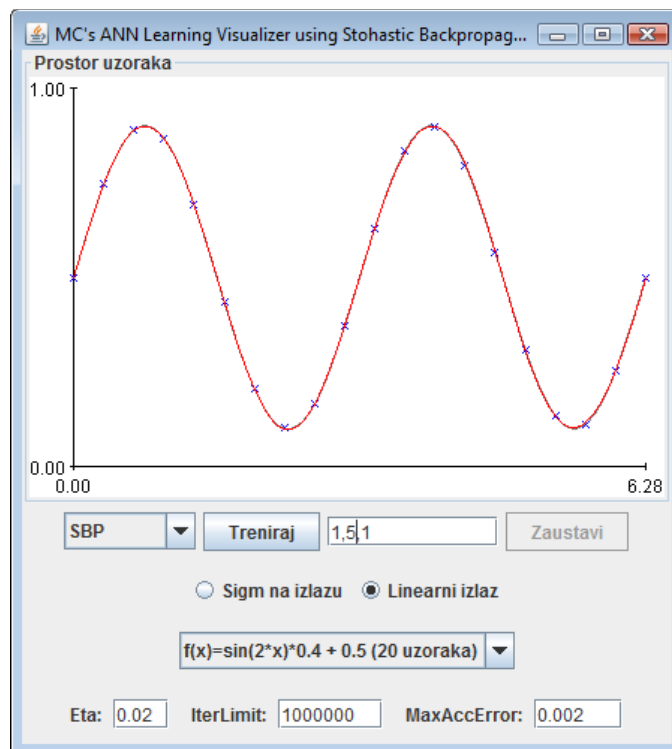
Slika 25. Mreža 1x10x1 i modeliranje dvije periode sinusoidalne funkcije na temelju 20 uzoraka

Konačno, još se moramo osvrnuti na dva detalja (koja se oba rješavaju na isti način). Prvi detalj je činjenica da sigmoidalna prijenosna funkcija samo asimptotski teži prema vrijednostima 0 i 1; da bismo natjerali takav neuron da na izlazu da vrijednost koja je bliska nuli ili jedan, algoritam učenja mora težine postaviti na vrlo velike vrijednosti, što za algoritam često predstavlja problem. To najbolje možete vidjeti na slici 24, gdje je mreža imala manjih problema s modeliranjem minimalnih i maksimalnih vrijednosti funkcije.

Drugi detalj je činjenica da smo sve uzorke na kojima smo do sada ilustrirali učenje neuronske mreže uzimali iz funkcija koje čija je kodomena bila interval $[0, 1]$ (ili njegov podskup). Znači li to da neuronskom mrežom ne možemo aproksimirati funkciju koja daje vrijednosti iz intervala $[-10, 10]$?

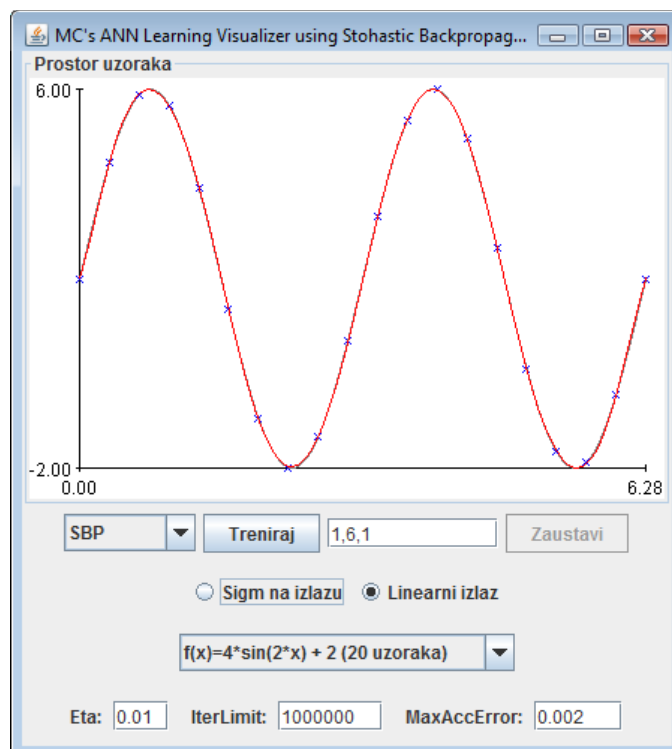
Odgovor je, dakako, ne. Neuronskom mrežom možemo na jednostavan način modelirati i takve funkcije. Dovoljno je u izlaznom neuronu (i samo njemu!) sigmoidalnu prijenosnu funkciju zamijeniti funkcijom identiteta, čime imamo mrežu koja u skrivenim slojevima ima sigmoidalne neurone a na izlazu ADALINE neurone (čiji je izlaz doslovno izračunati *net*, odnosno linearna kombinacija izlaza neurona iz posljednjeg skrivenog sloja).

Uporabom ovakvog neurona na izlazu mreža može jednostavnije modelirati zadanu funkciju, čak i kada je ona iz intervala $[0, 1]$. Primjer je prikazan na slici 26, gdje smo mrežom 1x5x1 dobili kvalitetniju aproksimaciju no što smo uspjeli dobiti mrežom 1x6x1 u kojoj su svi neuroni (uključivo i izlazni) bili sigmoidalni (slika 24).



Slika 26. Mreža 1x5x1 i modeliranje dvije periode sinusoidalne funkcije na temelju 20 uzoraka, linearni izlaz

Pokušamo li samo sigmoidalnim neuronima modelirati funkciju čija je kodomena šira od $[0,1]$, rezultat neće biti dobar – isprobajte to uživo. Mi ćemo u nastavku dati primjer mreže koja aproksimira funkciju $f(x)=4*\sin(2x)+2$, čija je kodomena $[-2, 6]$. Rezultat je prikazan na slici 27.



Slika 27. Mreža 1x6x1 i modeliranje dvije periode skalirane i translahirane sinusoidalne funkcije na temelju 20 uzoraka, linearni izlaz

Ovakve mreže također možemo učiti algoritmom Backpropagation. Kako je izlazna pogreška kod tog algoritma definirana kao derivacija prijenosne funkcije puta razlika između očekivane vrijednosti i dobivene vrijednosti, kod linearnog elementa pogreška će biti doslovno razlika između očekivane vrijednosti i dobivene vrijednosti jer je derivacije funkcije identiteta konstanta 1.

Na temelju prikazanih primjera trebali biste imati generalnu ideju kako ekspresivnost modela utječe na rezultat učenja.

- Što se događa kada je model preslab? Može li ukupna pogreška pasti na nulu?
- Što se događa kada je model “taman”? Može li ukupna pogreška pasti na nulu?
- Što se događa kada je model preekspresivan? Može li ukupna pogreška pasti na nulu? Što se događa sa sposobnošću generalizacije?

Ostala pitanja koja nismo razmatrali su:

- Koja je optimalna arhitektura neuronske mreže?
- Kako se povećati ekspresivnost mreže: da li u skrivene slojeve dodati nove neurone ili dodati nove slojeve?

Jasnih odgovora na ova pitanja danas još uvijek nema. Generalno znamo da dodavanje novih slojeva može poprilično povećati ekspresivnost mreže ali i smanjiti efikasnost učenja. Klasični algoritmi učenja (u koje ubrajamo i Backpropagation) naprosto nisu primjenjivi na učenje neuronskih mreža koje imaju puno slojeva. Za takve mreže danas se razvijaju novi načini učenja u okviru vrlo “vrućeg” područja *dubokih neuronskih mreža* (engl. *Deep Neural Networks*).

Svi demonstracijski programi koji su korišteni u ovom dokumentu dostupni su na adresi:
<http://java.zemris.fer.hr/nastava/ui/ann/>
u paketu `ffmlp-ann.zip` pa Vas potičemo da primjere i sami isprobate.