*PAPRASTO (“BACK-PROPAGATION”) NEURONINIO TINKLO FUNKCIJOS ATPAŽINIMAS GENERUOJANT ATSITIKTINIUS TAŠKUS*

*Laimonas Sutkus*

*2017-05-19*

# Hipotezė

Ar neuroninis tinklas gali atpažinti funkcijas (tiesines, kvadratines, logaritmines), generuojant atsitiktinius taškus tam tikrame poerdvyje ir klausiant: „ar šis taškas yra aukščiau, ar žemiau tau nežinomos funkcijos ?“

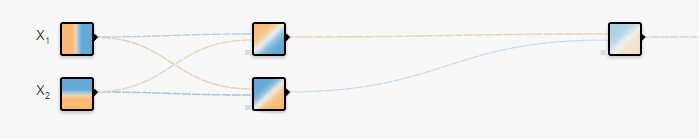
# Realizacijos idėja

1. Sugalvokime funkciją *f(x).*
2. Sugeneruokime atsitiktinį tašką leistinoje dvimatėje srityje. Tašo kordinatės susidteda iš *x* ir *y*.
3. Įstatykime x į mūsų sugalvotą funkciją *f(x)* ir gautą rezultatą palyginkime su mūsų atsitiktinai sugeneruotu y t.y. (*y > f(x) ar y <= f(x)*). Jei *y > f(x),* mūsų sugeneruotasis taškas yra aukščiau duotosios funkcijos, priešingu atveju jei *y < f(x),* mūsų sugeneruotas taškas yra žemiau duotosios funkcijos.
4. Paduokime neuroniniam tinklui x ir y koorinates ir prašykime jo atsakymo „aukščiau ar žemiau ?“
5. Palyginkime jo atsakymą su mums jau žinomu atsakymu ir pasakykime neuroniniam tinklui, ar prognozė buvo teisinga. (Mokymasi su mokytoju metodas).
6. Apmokykime tinklą su dabartiniu tašku.
7. Kartojame procesą nuo 2 punkto. (1000, 5000, 10000 kartų)

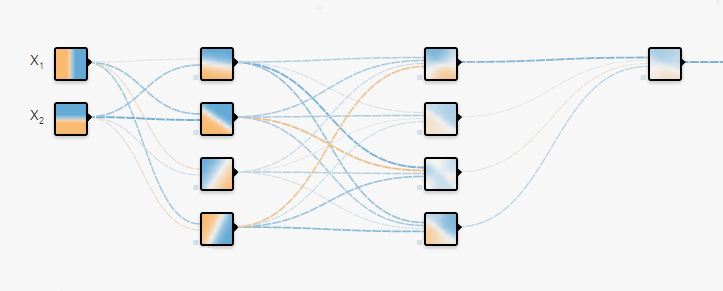
# Neuroninio tinklo architektūra

Naudojami trys neuroniniai tinklai:

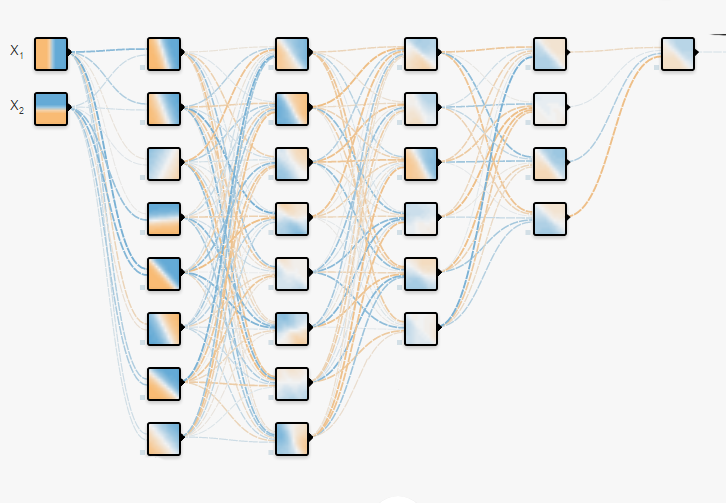
*2x2x1:*

**

*2x4x4x1:*

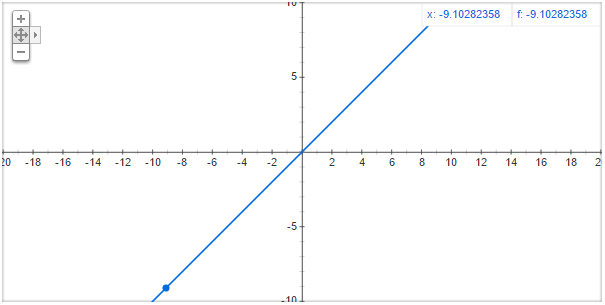
**

*2x8x8x6x4x1:*

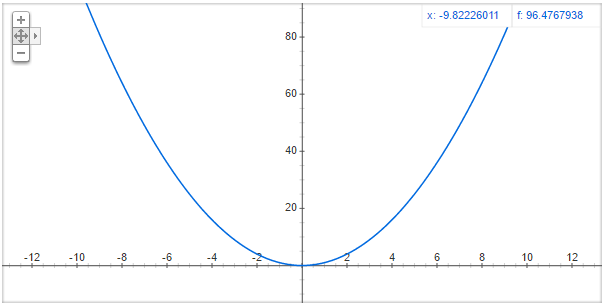
**

# Testinės funkcijos

*Lineari:*

**

*Kvadratinė:*

**

# Rezultatai

*Lineariai funkcijai:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tinklas* | *1000* | *5000* | *10000* | *Vizualizacija (5000)* |
| *2x2x1* | *Laikas* - 3.977s  *MSE* – 0.2233 | *Laikas* – 16.35s  *MSE* – 0.125 | *Laikas* – 33.26s  *MSE* – 0.114 |  |
| *2x4x4x1* | *Laikas* – 4.48s  *MSE* – 0.44 | *Laikas* – 17.65s  *MSE* – 0.08 | *Laikas* – 37.24s  *MSE* – 0.049 |  |
| *2x8x8x6x4x1* | *Laikas* – 4.83s  *MSE* – 0.44 | *Laikas* – 21.9  *MSE* – 0.394 | *Laikas* – 47s  *MSE* – 0.43 |  |

*Kvadratinei funkcijai:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tinklas* | *5000* | *10000* | *50000* | *Vizualizacija (50000)* |
| *2x2x1* | *Laikas* – 16.5s  *MSE* – 0.249 | *Laikas* – 33.6s  *MSE* – 0.27 | *Laikas* – 182.5s  *MSE* – 0.313 |  |
| *2x4x4x1* | *Laikas* – 18.9s  *MSE* – 0.25 | *Laikas* – 37.3s  *MSE* – 0.168 | *Laikas* – 198.45s  *MSE* – 0.0638 |  |
| *2x8x8x6x4x1* | *Laikas* – 20.7s  *MSE* – 0.5 | *Laikas* – 48.35s  *MSE* – 0.48 | *Laikas* – 214s  *MSE* – 0.507 |  |

# Išvados

Treniruojant patį paprasčiausią neuroninį tinklą (*2x2x1*) pastebėta, kad jis tinka identifikuoti tik linearioms funkcijoms. Pritaikius šį neuroninį tinklą kvadratinei funkcijai gauname linearius prognozavimus. Treniruojant didelį ir sudėtingą neuroninį tinklą (*2x8x8x6x4x1*) pastebėta, kad tinklas beveik neišsimokina visiškai. Taip yra todėl, kad tokiam dideliam tinklui reikia ir žymiai daugiau laiko ir duomenų, kad apmokyti. Pats praktiškiausias tinklas buvo (*2x4x4x1*), kuris tiko atpažinti tiek linearioms tiek kvadratinėms funkcijoms. Tinklo dydis nėra nei per mažas (kad daryti lineariius prognozavimus) nei per didelis (kad reikalautų daug laiko ir duomenų), todėl net prie 10‘000 duomenų jau galėjome pastebėti, kad jis gan gerai identifikuoja kvadratinę funkciją su 0.168 MSE tikslumu.