



Paskaitos
Turinys

ATGALINIO SKLIDIMO TINKLAS

1. Elementarus atgalinio sklidimo tinklas

- Apibrėžimas
- Struktūra

2. Atgalinio sklidimo tinklo mokymo algoritmas

3. Atgalinio sklidimo tinklo atsakas

4. Atgalinio sklidimo tinklo savybės

- Konvergencija
- Privalumai
- Trūkumai
- Naudojimas

5. Atgalinio sklidimo tinklo optimizavimas

- Mokymas su inercija
- Adaptyvusis mokymo greitis
- Kitos optimizavimo kryptys



Pagrindinė
Literatūra

1. Simpson, P.K. (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*. Pergamon Press, pp. 112-119.
2. Rumelhart, D.E. & McClelland, J.L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. The MIT Press. Vol. 1.
3. MATLAB ver. 4.0 (1994). *Neural Networks: Toolbox*. Math Corp., 5 chapter.

Skaidrės Nr.

1



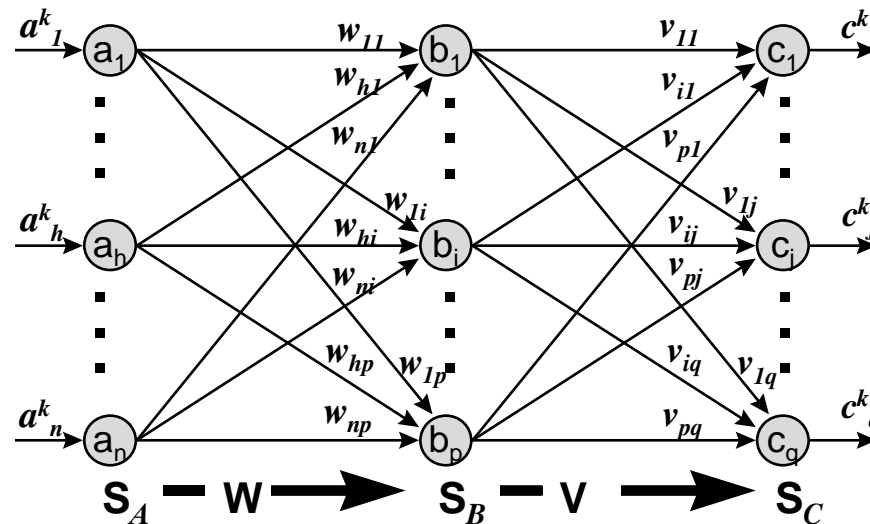
Elementarus atgalinio sklaidimo tinklas

Elementarus atgalinio sklaidimo tinklas yra trijų sluoksnių tinklas su nuosekliais ryšiais nuo įėjimo iki paslėpto sluoksnio bei nuo paslėpto sluoksnio iki išėjimo sluoksnio. Yra įmanoma naudoti keletą paslėptųjų sluoksnių, grįžtamuosius bei kryžminius ryšius. Nors šios šiuolaikinės struktūros yra svarbios, tačiau jos užgožia šio algoritmo paprastumą. Todėl toliau remsimės elementaraus tinklo struktūra.

Apibrėžimas

Elementarus atgalinio sklaidimo tinklas (angl.k.- Backpropagation) yra heteroasociatyvusis įrenginys įsimenantis analogines pavyzdžių poras (A^k, C^k) , $k=1, 2, \dots, m$, naudodamasis daugiasluoksne gradientinio nusileidimo klaidos korekcijos procedūra, kur k -toji pavyzdžių pora pateikiama vektoriumi $A^k=(a_1^k, \dots, a_n^k)$ ir $C^k=(c_1^k, \dots, c_m^k)$. DNT mokomas nerealiame laike, veikia diskrečiame laike ir yra atvaizduojamas trijų sluoksnių nuoseklių ryšių struktūra.

Struktūra



Pav. 8.1



Atgalinio sklidimo tinklo mokymo algoritmas

A. Atsitiktiniu būdu parenkamos mažo dydžio ryšių ir slenksčių svorių reikšmės.

B. Įėjimo pavyzdį (analoginių reikšmių vektorių $A^k = (a^k_1, \dots, a^k_n)$) pateikiame tinklo įėjimui, o norimą atsako pavyzdį (analoginių ar dvejetainių reikšmių vektorių $C^k = (c^k_1, \dots, c^k_p)$) pateikiame DNT išėjimo sluoksniui.

Kiekvieno ciklo metu gali būti pateikiami vis nauji pavyzdžiai arba pavyzdžiai turi būti pateikiami cikliškaiki iki visiškų ryšių svorių nusistovėjimo.

C. Skaičiuojamas tikrasis tinklo atsakas:

$$b_i = F\left(\sum_{h=1}^n a_h \cdot w_{hi} + Q_i\right), \quad c_j = F\left(\sum_{i=1}^n b_i \cdot v_{ij} + Z_j\right)$$

visiems $i=1, \dots, p$ ir $j=1, \dots, q$.

D. Adaptuojami ryšių ir slenksčių svoriai.

Naudojamas rekursinis algoritmas.

Skaičiavimas vyksta pasluoksniui pradedant nuo galinio (išėjimo) sluoksnio ir baigiant pirmuoju paslėptuoju sluoksniu.

► **Skaičiuojamas ryšių svorių pokytis:**

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot \delta_j x'_i, \text{ kur}$$

η - konstantė,

δ_j - klaida,

x'_i - i-tojo neurono įėjimo reikšmė.

► **Jei j-tasis neuronas yra išėjimo sluoksnyje, tai δ_j skaičiuojamas sekančiais:**

$$\delta_j = y'_j \cdot (1 - y'_j) \cdot \sum_k (\delta_k \cdot w_{kj}), \text{ kur}$$

y'_j - norimas j-tojo neurono atsakas.

► **Jei j-tasis neuronas yra paslėptajame sluoksnyje, tai δ_j skaičiuojamas taip:**

$$\delta_j = x'_j \cdot (1 - x'_j) \cdot \sum_k (\delta_k \cdot w_{jk}), \text{ kai } k \text{ kinta nuo } 1 \text{ iki visų neuronų esančių po } j\text{-tojo neurono skaičiaus.}$$

Slenksčiai adaptuojami analogiškai kaip ir ryšių svoriai, tarus, kad jų įėjimo signalai lygūs vienetui.

E. Grįžti prie B žingsnio.



Atgalinio sklidimo tinklo atsakas. Savybės

Atgalinio sklidimo tinklo atsakas

Atgalinio sklidimo DNT atsakas formuojamas įėjimo pavyzdį perduodant paeiliui per W ir V matricas, t.y.:

$$b_i = F\left(\sum_{h=1}^n a_h \cdot w_{hi} + Q_i\right), \text{ visiems } i = 1, 2, \dots, p.$$

$$c_j = F\left(\sum_{i=1}^n b_i \cdot v_{ij} + Z_j\right), \text{ visiems } j = 1, 2, \dots, q.$$

Konvergencija

Yra įrodoma, kad atgalinio sklidimo DNT visada konverguos, tačiau nėra garantuojama, kad mokymo metu DNT suras globalų klaidos minimumą.

Privalumai

- gali išmokti žymiai daugiau pavyzdžių negu yra neuronų įėjimo sluoksnyje
- gali išmokti bet kokio sudėtingumo netiesinius sąryšius

Trūkumai

- labai ilgai trunka mokymas
- nesugeba mokytis realiame laike
- žinant, kad kiekvienam uždaviniui egzistuoja tikslaus sąryšio mokymo procedūra, neaišku, kaip ją realizuoti

Naudojimas

- vaizdų apdorojime
- kalbos apdorojime
- trumpalaikiame apdorojime
- nuspėjime ir optimizavime
- diagnostikoje
- raidžių atpažinime
- žinių apdorojime
- teksto ir sakinių apdorojime
- signalų apdorojime
- ir t.t.



Atgalinio sklidimo tinklo optimizavimas

Mokymas su inercija

Inercija į mokymo algoritmą įvedama naudojant šią išraišką:

$$\Delta W(i, j) = m \cdot c \cdot \Delta W(i, j) + (1 - m \cdot c) \cdot lr \cdot D(i) \cdot P(j),$$

t.y. ryšių svorių pokytis yra lygus prieš tai buvusio ryšių svorių pokyčio daliai ir naujam pokyčiui apskreistam atgalinio sklidimo algoritmo. Inercijos konstantė m lemia kokią dalį prieš tai buvusio ryšių svorių pokyčio pridėti.

Kai $m=0$, tai inercijos sumos narys išnyksta ir gauname įprastą atgalinio sklidimo tinklo mokymo išraišką.

Kai $m=1$, tai antrasis sumos narys išnyksta ir gauname įprastą perceptronų tinklo mokymą.

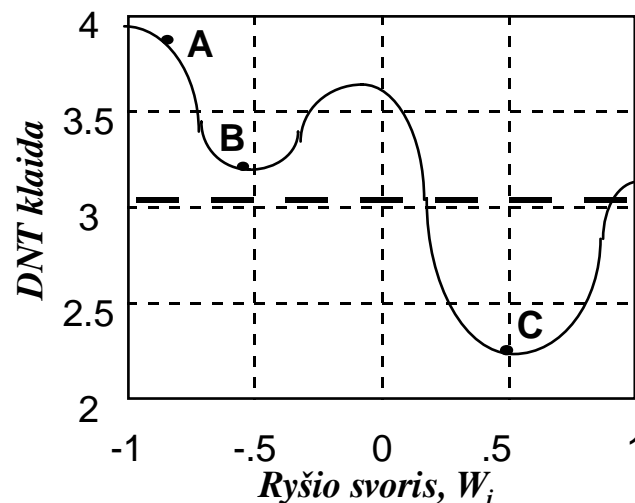
Mokant, inercija įgalina atgalinio sklidimo tinklą ne tik sekti vietinio gradiento kryptimi, bet ir reaguoti į paskutiniuosius pokyčius klaidos paviršiuje.

Veikdama kaip žemų dažnių filtras, inercija leidžia DNT nekreipti dėmesio į nedidelius netolygumus klaidos paviršiuje, ir tokiu būtu išvengti įstrigimo vietiniuose minimumuose.

Pavyzdys

Tarkime, kad norime pasiekti ne didesnę kaip 3.05 DNT klaidą.

Naudojant įprastą atgalinio sklidimo algoritmą, W_i reikšmė bus -0.5, o DNT klaida atžvilgiu šio svorio taške B bus (3.2), t.y. reikiamo rezultato nepasieksime, nes klaida įstrigs vietiniame minimume.



Įvedus inerciją į mokymą, W_i reikšmė bus >0.2 ir DNT klaida net galės pasiekti savo globalų minimumą W_i svorio atžvilgiu, t.y. C tašką.

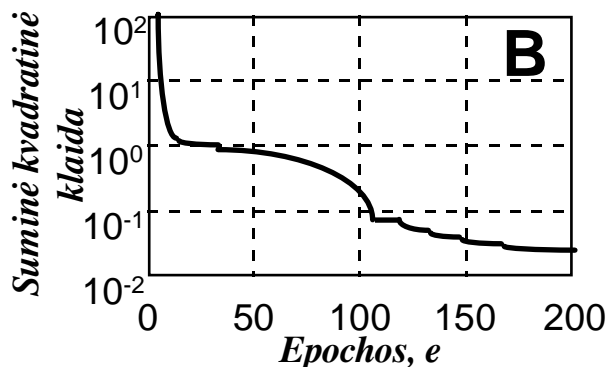
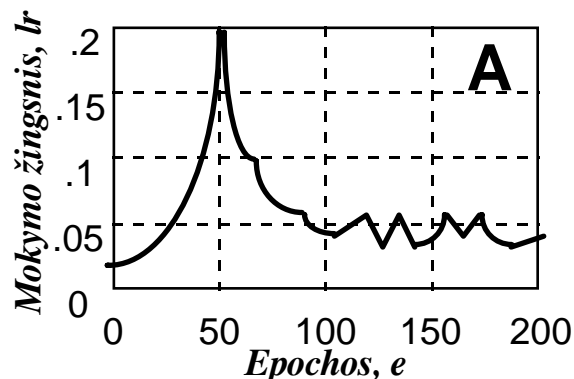


Atgalinio sklaidimo tinklo optimizavimas (tęsinys)

Adaptyvusis mokymo greitis

Kiekvienos epochos metu skaičiuojamos naujos ryšių ir slenksčių svorių reikšmės. Tačiau, jei tinklo klaida padidėja, tai naujos reikšmės atmetamos, o mokymo žingsnis sumažinamas. Jei tinklo klaida sumažėja, tai naujos reikšmės paliekamos, o mokymo žingsnis padidinamas. Kai mokymo žingsnis tampa per didelis, kad užtikrintų DNT klaidos mažėjimą, jis mažinamas iki atsinaujina stabilus mokymasis.

✎ Naudojant adaptyvių mokymo greitį žymiai sutrumpiname atgalinio sklaidimo tinklo mokymo laiką.



Kitos optimizavimo kryptys

- dirbtinių neuronų, esančių paslėptuosiuose sluoksniuose, skaičiaus optimizavimas
- naujų efektyvesnių struktūrų paieška
- normavimo, apibendrinimo ir atsparumo klaidoms sąvybių analizė
- aukštesniųjų koreliacijų bei naujų neuronų aktyvavimo funkcijų naudojimo analizė



Pagrindiniai teiginiai

- ✎ *Elementarus atgalinio sklidimo tinklas yra trijų sluoksnių nuoseklių ryšių heteroasociatyvusis įrenginys išimenantis analogines pavyzdžių poras, naudodamasis daugiasluoksne gradientinio nusileidimo klaidos korekcijos procedūra.*
- ✎ *Gradientinio nusileidimo klaidos korekcijos procedūra yra skirta daugiasluoksnių nuoseklių ryšių struktūrų DNT mokyti.*
- ✎ *Atgalinio sklidimo DNT mokymo procedūros esmė: iš pradžių skaičiuojama išėjimo sluoksnio klaida ir pagal ją šis sluoksnis mokomas, o tik po to nuosekliai pereinama prie paslėtųjų sluoksnių ir tik pabaigoje mokomas įėjimo sluoksnis.*
- ✎ *Atgalinio sklidimo tinklas atsaką formuoja tiesiogiai, perfiltruodamas įėjimo pavyzdį nuosekliai per visus DNT sluoksnius.*
- ✎ *Didžiausias atgalinio sklidimo DNT trūkumas - negarantuota konvergencija prie globalaus klaidos minimumo, didžiausias privalumas - tinklas gali įsisąvinti reikiamu tikslumu bet kokio sudėtingumo funkcijas.*
- ✎ *Atgalinio sklidimo DNT labiausiai šiuo metu naudojamas tinklas.*
- ✎ *Atgalinio sklidimo DNT naudotinas tose situacijose, kai reikalaujama įsisąvinti sudėtingą netiesinį sąryšį, tačiau nėra būtina tai atlikti realiame laiko mastelyje.*