

Konvoluutioneuroverkot

Teemu Sarapisto

February 6, 2018

Viimeisen hieman yli kymmenen vuoden aikana voidaan sanoa keinotekoisien neuroverkkojen ja syväoppimisen tehneen läpimurron. Syväoppimisen voidaan katsoa syntyneen jo 40-luvulla, mutta laajamittaiseen sovelluskäyttöön se on tullut vasta viime vuosina, kun sekä riittävä määrä luokiteltua dataa, että riittävästi prosessointitehoa on tullut helposti saataville. Myös algoritmipuolella tapahtuneet edistykset ovat edesauttaneet läpimurtoa. Aikaisemmin koneoppimisen alalla haasteelliseksi osoittautuneissa sovelluskohteissa kuten kuvien sekä puheen sisällön tunnistamisessa keinotekoiset neuroverkot ovat osoittautuneet toistaiseksi ylivoimaisesti parhaiten toimiviksi ratkaisuiksi.

1 Neuroverkkojen rakenne

1.1 Keinotekoinen neuroni

Biologisista vaikuttimistaan huolimatta keinotekoiset neuronit ovat käytännössä Kaavan 1 muotoisia matemaattisia funktioita.

$$\Sigma w_i x_i \mapsto f(\Sigma w_i x_i + b) \quad (1)$$

Yleisessä muodossaan neuroni ottaa vastaan yhden tai useampia syötteitä x_1, x_2, \dots, x_n , joista kullekin on asetettu jokin painoarvo w_i . Syötteiden ja painotuksien tulojen summa $\Sigma x_i w_i$ annetaan parametrina aktivaatiofunktiolle f ja tämän funktion arvo toimii neuronin lopullisena ulostuloarvona.

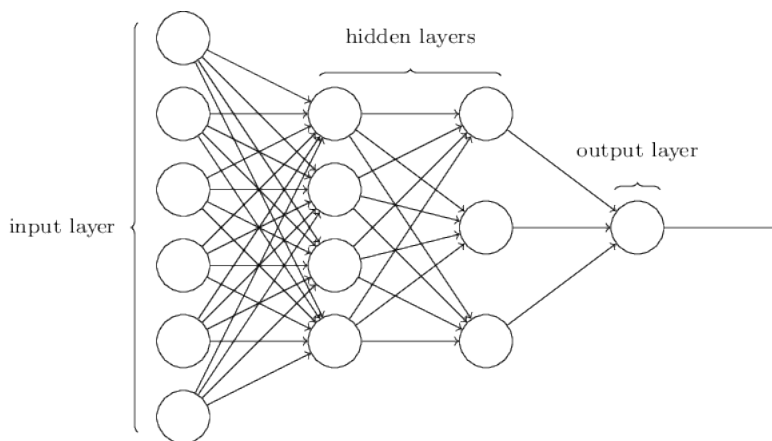
Toisinaan käytetään myös taipumusvakiota (bias) b , joka lisätään syötteiden ja painotuksien tulojen summaan.

Ensimmäinen tällainen neuroni, perseptroni, kehitettiin 50-luvulla. Sen syötteet ja ulostuloarvot ovat binäärisiä ja aktivaatiofunktiona toimii Kaavan 2 mukainen funktio.

$$\text{ulostuloarvo} \begin{cases} 0 \text{ jos } \Sigma x_i w_i + b \leq 0 \\ 1 \text{ jos } \Sigma x_i w_i + b > 0 \end{cases} \quad (2)$$

Yksittäisen neuronin tasolla neuronien oppiminen tapahtuu syötteiden painotuksien ja taipumusarvon muuttumisen kautta. Perseptroneja käytettäessä törmätään kuitenkin usein ongelmaan, jossa yksi pieni muutos painotuksissa tai taipumusarvossa johtaa ulostuloarvon vaihtumiseen, joka saattaa aiheuttaa suuria muutoksia ulostuloarvoissa myös koko neuroverkon tasolla. Usein halutaan hienovaraisempia muutoksia ja siksi usein käytetään neuroneita joiden syöte- ja paluuarvot voivat olla myös mitä vain reaalityyppisiä nollan ja yhden väliltä. Esimerkiksi yksi tällainen laajalti käytössä oleva neuroni on sigmoidinen neuroni, jonka aktivaatiofunktiona toimii sigmoidinen funktio.

1.2 Keinotekoisien neuroverkkojen rakenne



Kuva 1: <http://neuralnetworksanddeeplearning.com/chap1.html>

Yksinkertaisimman verkkorakenteen omaavat eteenpäinsyöttävät neuroverkot muodostetaan tasoittain, jossa jokaisen verkon tason neuronit saavat syötteenään niitä edeltävän tason neuroneiden ulostuloarvot. Poikkeuksena ensimmäinen taso (kuvassa vasemmanpuoleisimpana), joihin verkon syöte koodataan. Esimerkiksi haluttaessa syöttää 64x64 kuva neuroverkolle, voidaan syötekerroksena käyttää 64x64 neuronin kerrosta, johon kuvan pikselien väriarvot koodataan.

Vaikka syväoppimista voidaan harjoittaa myös muutoin kuin keinotekoisilla neuroverkoilla, neuroverkkojen tapauksessa termillä viitataan neuroverkkojen piilokerroksiin ja niiden määrään. Kasvattamalla neuroverkkotasojen sekä tasoissa olevien neuronien määrää, neuroverkoilla voidaan mallintaa entistä monimutkaisempia funktioita.

2 Neuroverkkojen oppiminen / opettaminen

Neuroverkkoa opettaessa tavoitteena on minimoida neuroverkon tekemä virhe sen approksimoidessa jotakin funktiota. Tämän virheen määrää arvioidaan virhefunktion (error function) avulla. Neuroverkon laskiessa ulostuloarvon jollekin syötteelle, tapahtuu eteenpäinkulkeutumista (forward-propagation). Taaksepäinkulkeutumisiksi (back-propagation) kutsutaan algoritmiä jonka avulla jollekin neuroverkon painoille määritellylle virhefunktiolle lasketaan gradientti, jonka perusteella neuroverkon approksimoinnin virhettä voidaan lähteä pienentämään gradientin laskeutumismenetelmää käyttäen.

Suuri haaste neuroverkkojen opetuksessa on ylisovitus (overfitting) jossa neuroverkon virhefunktion arvo on harjoitusdatalla saatu erittäin pieneksi, mutta uuden datan kanssa virhefunktion antaa suuria arvoja. Tällöin neuroverkon oppima malli vastaa harjoitusdataa liian tarkkaan, eikä enää suoriudu yleisestä tapauksesta toivotulla tavalla. Ylisovitusta korjaamaan on kehitetty metodeja kuten esimerkiksi neuroniyksikköjen pudotus (dropout) jossa opetusvaiheessa yksittäisiä neuroneita poistetaan käytöstä, joka estää yksittäisiä neuroneita naapureineen erikoistumatta tiettyyn datan ominaisuuteen liian tarkasti.

Neuroverkkojen opetuksessa olennaisessa osassa on myös verkon alkuperäisten painotuksien valitseminen sopivalla tavalla.

Lähteet