Neural networks

Architectures and training tips

Who? Sebastian Björkqvist

From? IPRally Technologies

When? 09.01.2019

Why neural networks?

- Can approximate any function [Hornik, 1991]
- May learn to respond to unexpected patterns
- Useful especially when the amount of data is large
- Less need for feature engineering compared to traditional ML methods

Mitä on informaatio?

Algoritmien ollessa kyseessä **Churchin-Turingin teesi** kertoo minkälaiset asiat ovat laskettavissa.

Mitä on informaatio?

Algoritmien ollessa kyseessä **Churchin-Turingin teesi** kertoo minkälaiset asiat ovat laskettavissa.

Millä tavalla voidaan vastaavasti määritellä informaatio?

Merkkijonon informaatiomäärä

Haluamme määritellä jonon informaatiomäärän ja verrata sitä jonon pituuteen.

Informaatiomäärä on siis merkkijono.

Merkkijonon lyhin kuvaus

Olkoon x merkkijono binäärimuodossa. Jonon x **lyhin kuvaus** $d(x)$ on lyhin merkkijono $\langle M, w \rangle$, jolle pätee että Turingin kone M palauttaa merkkijonon x syötteellä w.

Määrittelemme jonon $\langle M, w \rangle$ konkatenoituna merkkijonona Mw.

Kolmogorov-kompleksiteetti

Jonon x **Kolmogorov-kompleksiteetti** $K(x)$ on jonon lyhimmän kuvauksen pituus, toisin sanoen

$$K(x) = |d(x)|$$

Kolmogorov-kompleksiteetin perustuloksia

On olemassa vakiot c_1 ja c_2, joille pätee

1. $K(x) \leq |x| + c_1$
2. $K(xx) \leq K(x) + c_2$

kaikilla merkkijonoilla x.

Konkatenoidun jonon kompleksiteetti

Kahden mielivaltaisen jonon konkatenaation kompleksiteetille voimme helposti todistaa seuraavan ylärajan:

On olemassa vakio c, jolle pätee

- $K(xy) \leq 2 \cdot K(x) + K(y) + c$

kaikilla merkkijonoilla x ja y.

Konkatenoidun jonon kompleksiteetti

Todistuksen idea: Tuplataan kuvauksen $d(x)$ bitit, ja konkatenoidaan perään kuvaus $d(y)$. Väliin lisätään erotin 01.

Konkatenoidun jonon kompleksiteetti

Todistuksen idea: Tuplataan kuvauksen $d(x)$ bitit, ja konkatenoidaan perään kuvaus $d(y)$. Väliin lisätään erotin 01.

Esimerkiksi jonoilla $d(x) = 0110110$ ja $d(y) = 1110001$ saadaan

Konkatenoidun jonon kompleksiteetti

Todistuksen idea: Tuplataan kuvauksen $d(x)$ bitit, ja konkatenoidaan perään kuvaus $d(y)$. Väliin lisätään erotin 01.

Esimerkiksi jonoilla $d(x) = 0110110$ ja $d(y) = 1110001$ saadaan

$$\underbrace{00111100111100}_{\text{d(x) tuplattuna}}01\underbrace{1110001}_{\text{d(y)}}$$

Konkatenoidun jonon kompleksiteetti

Voimme saavuttaa myös ylärajan

- $K(xy) \leq 2 \cdot log_2(K(x)) + K(x) + K(y) + c,$

Konkatenoidun jonon kompleksiteetti

Voimme saavuttaa myös ylärajan

- $K(xy) \leq 2 \cdot log_2(K(x)) + K(x) + K(y) + c$,

mutta osoittautuu mahdottomaksi saavuttaa raja

- $K(xy) \leq K(x) + K(y) + c$.

Kolmogorov-kompleksiteetin optimaalisuus

Turingin koneen vaihtaminen joksikin muuksi funktioksi määritelmässä ei olennaisesti muuta merkkijonon kompleksiteettia.

Tämä johtuu siitä, että voimme simuloida kaikkia algoritmeja Turingin koneella.

Tiivistyvät merkkijonot

Merkkijono x on **c-tiivistyvä**, jos

$$K(x) \leq |x| - c.$$

Jos merkkijono ei ole 1-tiivistyvä, on se **tiivistymätön**.

Tiivistymätön merkkijono?

Onko olemassa tiivistymättömiä merkkijonoja?

Tiivistymätön merkkijono?

Onko olemassa tiivistymättömiä merkkijonoja?

Pituutta n olevia merkkijonoja on 2^n kappaletta, mutta tätä lyhyempiä merkkijonoja on vain $2^n - 1$ kappaletta.

Ainakin yksi pituutta n oleva jono on siis tiivistymätön.

Kompleksiteetin määrittäminen

Brute force-algoritmi ei toimi, sillä saatamme ajautua ikuiseen silmukkaan.

Kompleksiteetin määrittäminen

Brute force-algoritmi ei toimi, sillä saatamme ajautua ikuiseen silmukkaan.

Yleistä algoritmia kompleksiteetin määrittämiselle ei ole.

Ongelma on itseasiassa Turing-yhtäpitävä pysähtymisongelman kanssa.

Tiivistymättömän jonon ominaisuuksia

Yleisesti ei ole mahdollista selvittää, onko jokin jono tiivistymätön.

Tiivistymättömillä jonoilla on paljon samoja ominaisuuksia kuin satunnaisilla jonoilla.

Jonon lyhin kuvaus $d(x)$ on lähes tiivistymätön.

Yhteenveto

- Kolmogorov-kompleksiteetti kertoo merkkijonon informaatiosisällön
- Kompleksiteetti ei olennaisesti riipu käytetystä laskentamallista
- Tiivistymättömät jonot ovat satunnaisjonojen kaltaisia
- Kompleksiteettia ei yleensä voida määrittää

References

Nielsen, Michael A. *Neural Networks And Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com/

Hornik, Kurt. *Approximation Capabilities of Multilayer Feedforward Networks*. Neural Networks, 4(2), 251–257, 1991.