Näin koulutat neuroverkkosi

Backpropagation- eli takaisinleviämisalgoritmi

Aaro Salosensaari

Studia Varjomafia V

28.2.2021

Mitä yritän tässä esityksessä selittää

1. **Neuroverkkojen opettaminen**: Jonkinlainen käsitys miten *backpropagation*- eli "takaisinleviämis"-algoritmi teoriassa toimii.

Mitä yritän tässä esityksessä selittää

1. **Neuroverkkojen opettaminen**: Jonkinlainen käsitys miten *backpropagation*- eli "takaisinleviämis"-algoritmi teoriassa toimii.
2. Hieman edellistä Studia Varjomafia-esitystä tarkempi selitys ns "feed-forward" neuroverkkojen toiminnasta.

Mitä yritän tässä esityksessä selittää

1. **Neuroverkkojen opettaminen**: Jonkinlainen käsitys miten *backpropagation*- eli "takaisinleviämis"-algoritmi teoriassa toimii.
2. Hieman edellistä Studia Varjomafia-esitystä tarkempi selitys ns "feed-forward" neuroverkkojen toiminnasta.

Asioita jotka jäävät esityksen ulkopuolelle:

- Muut koneoppimisen muodot.
- Syväoppivien neuroverkkojen monet käytännölliset yksityiskohdat.
- ...

Sisältö

Sisältö

Koneoppiminen ja sen alalajeista

Koneoppivat algoritmit

Koneoppiminen tutkii algoritmeja, jotka **opetusdatan** pohjalta oppivat ratkaisemaan **uusia ongelmia**.

Koneoppiminen ja sen alalajeista

Koneoppivat algoritmit

Koneoppiminen tutkii algoritmeja, jotka **opetusdatan** pohjalta oppivat ratkaisemaan **uusia ongelmia**.

Ohjattu koneoppiminen

Ohjatussa koneoppimisessa (*supervised learning*) algoritmille annetaan opetusdataa, jossa on syötteitä (*input X*) ja vastaavia ulostuloja (*output y*).

Koneoppiminen ja sen alalajeista

Koneoppivat algoritmit

Koneoppiminen tutkii algoritmeja, jotka **opetusdatan** pohjalta oppivat ratkaisemaan **uusia ongelmia**.

Ohjattu koneoppiminen

Ohjatussa koneoppimisessa (*supervised learning*) algoritmille annetaan opetusdataa, jossa on syötteitä (*input X*) ja vastaavia ulostuloja (*output y*).
Halutaan, että algoritmi oppii *yleisesti* yhdistämään syötteet haluttuihin ulostuloihin:

$$X \to y$$

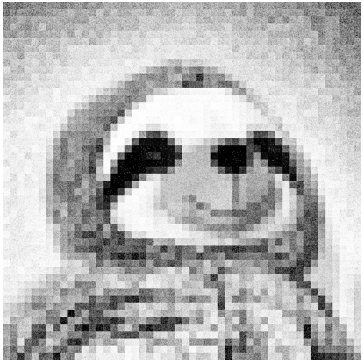
Esimerkki: Kuvasta datamatriisi

X

$\rightarrow$ (algoritmi) $\rightarrow y =$ "LAISKIAINEN"

Muistutus: Kuva on datamatriisi

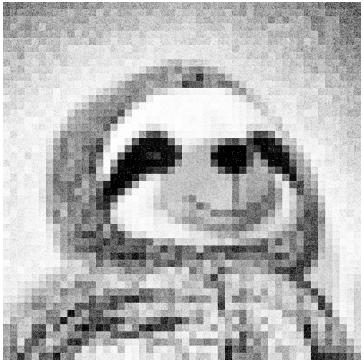
Kuva.



Muistutus: Kuva on datamatriisi

Kuva.

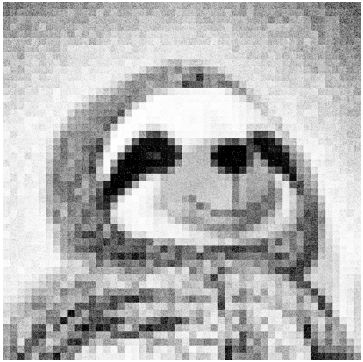
Kuva koostuu
RGB-pikseleistä.



Muistutus: Kuva on datamatriisi

Kuva.

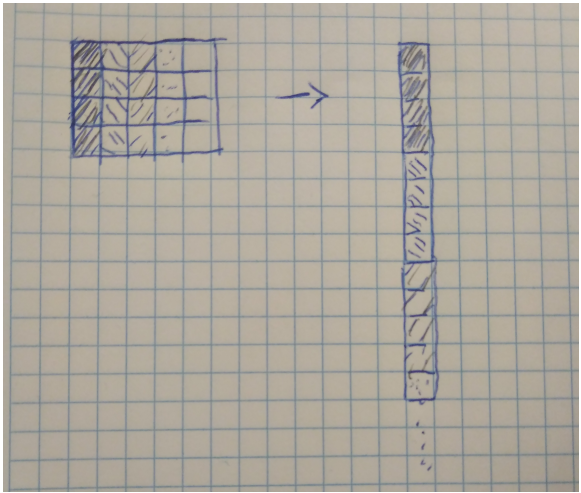
Kuva koostuu RGB-pikseleistä.



Yksinkertaistetaan lisää. Pikseli $=$ yksi numero 0.0 $\ldots$ 1.0.

Esimerkki: Datamatriisista vektoriksi

- Jokainen matriisi voidaan myös ajatella (ja käsitellä) vektorina.



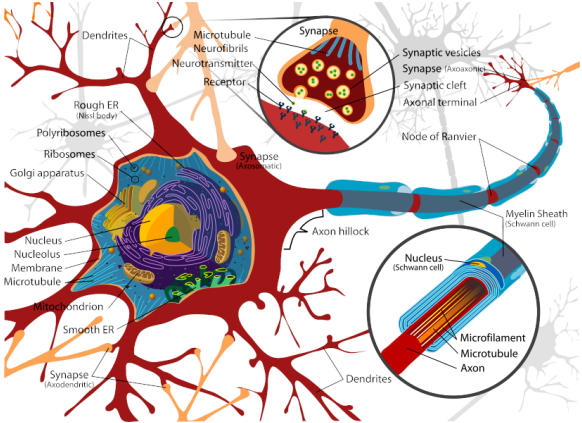
Sisältö

Muistutus!

- Tässä esityksessä puhutaan neuroverkoista ja neuroneista.

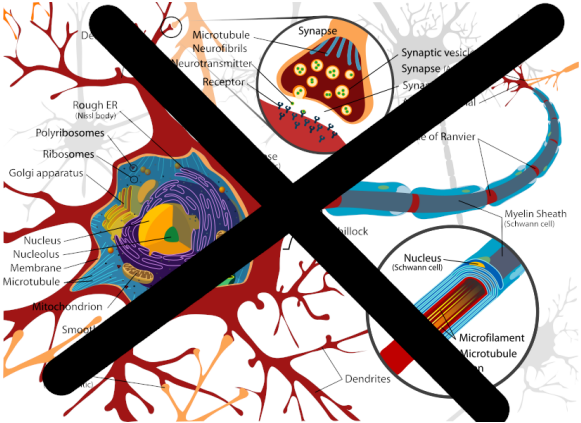
Muistutus!

- Tässä esityksessä puhutaan neuroverkoista ja neuroneista.
- Ne matemaattinen malli, joka on **inspiroitunut** biologisista hermosoluista.



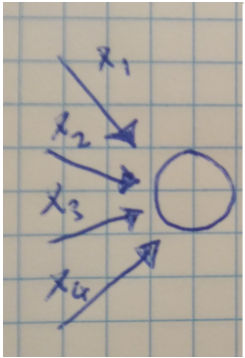
Muistutus!

- Tässä esityksessä puhutaan neuroverkoista ja neuroneista.
- Ne matemaattinen malli, joka on **inspiroitunut** biologisista hermosoluista.
- ...mutta on myös hyvin yksinkertainen, ei malli biologisista hermosoluista.



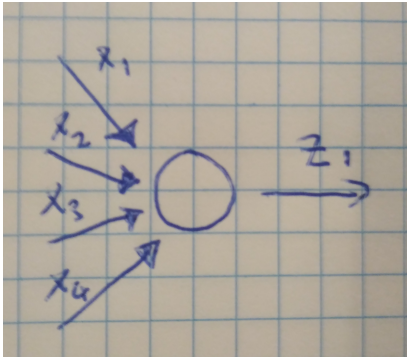
Neuroverkon perusajatus

- Neuroverkko koostuu "neuroneista" (*neural unit*). Neuroni vastaanottaa useita signaaleja $x_1, x_2, \ldots$



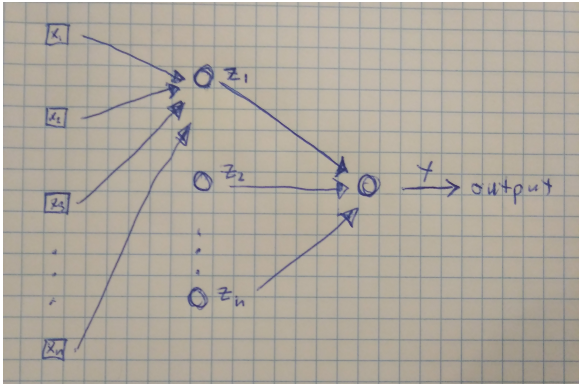
Neuroverkon perusajatus

- Neuroverkko koostuu "neuroneista" (*neural unit*). Neuroni vastaanottaa useita signaaleja $x_1, x_2, \ldots$
- ja yksinkertaisella operaatiolla muodostaa niistä syötteen seuraavalle tasolle z.

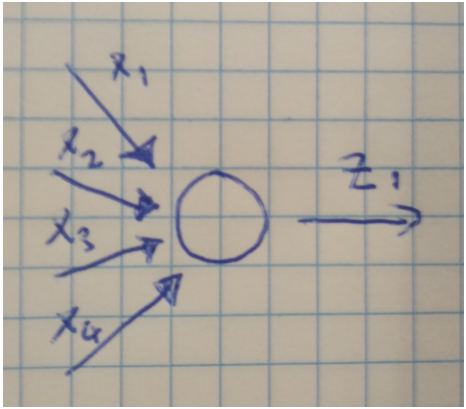


Neuroverkon perusajatus

- Neuroverkko koostuu "neuroneista" (*neural unit*). Neuroni vastaanottaa useita signaaleja $x_1, x_2, \ldots$
- ja yksinkertaisella operaatiolla muodostaa niistä syötteen seuraavalle tasolle z.
- Neuroneista muodostetaan verkko, jossa on useita tasoja. Ensimmäinen taso x syöte, viimeisenä ulostulo y. Yksinkertaisten neuronien iso verkko mahdollistaa hyvin monimutkaista prosessointia.

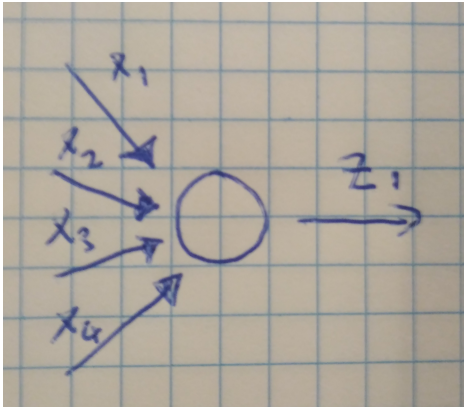


Neuroverkko hieman yksityiskohtaisemmin



Neuroverkko hieman yksityiskohtaisemmin

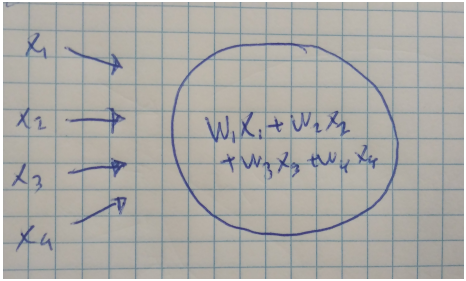
Verkon neuronit toimivat yleensä kahdessa vaiheessa.



Neuroverkko hieman yksityiskohtaisemmin

Verkon neuronit toimivat yleensä kahdessa vaiheessa.

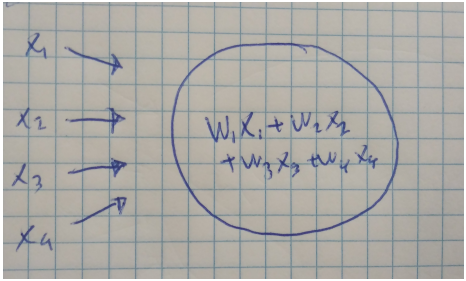
1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$.



Neuroverkko hieman yksityiskohtaisemmin

Verkon neuronit toimivat yleensä kahdessa vaiheessa.

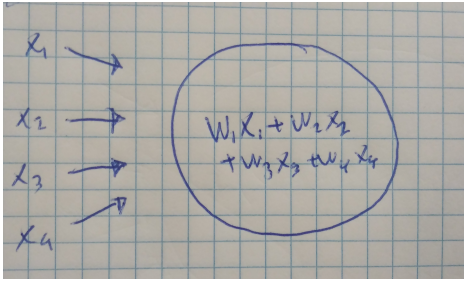
1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$. $w_i \approx$ kuinka tärkeä x_i on.



Neuroverkko hieman yksityiskohtaisemmin

Verkon neuronit toimivat yleensä kahdessa vaiheessa.

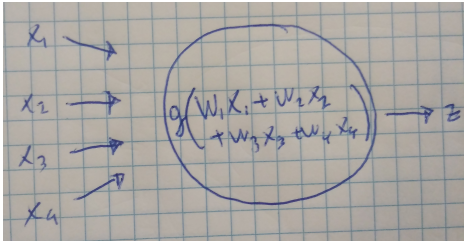
1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$. $w_i \approx$ kuinka tärkeä x_i on.
2. Tärkeys $= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$.



Neuroverkko hieman yksityiskohtaisemmin

Verkon neuronit toimivat yleensä kahdessa vaiheessa.

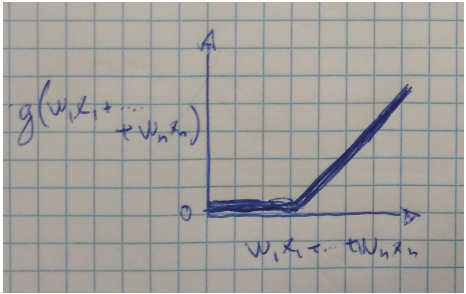
1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$. $w_i \approx$ kuinka tärkeä x_i on.
2. Tärkeys $= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$.
3. Syötteiden summalle tehdään muunnos g, joka antaa neuronin ulostulon: $g(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) = z$.



Neuroverkko hieman yksityiskohtaisemmin

Verkon neuronit toimivat yleensä kahdessa vaiheessa.

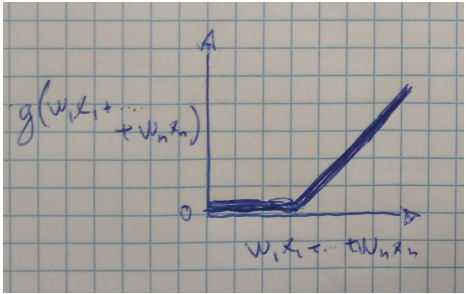
1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$. $w_i \approx$ kuinka tärkeä x_i on.
2. Tärkeys $= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$.
3. Syötteiden summalle tehdään muunnos g, joka antaa neuronin ulostulon: $g(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) = z$. Muutos g on mallintaa neuronin **aktivoitumista**. Latistaa "tärkeyden" nollaan tai korostaa sitä suuresti.



Neuroverkko hieman yksityiskohtaisemmin

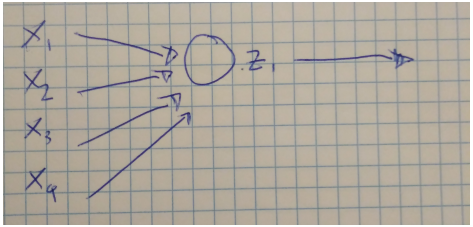
Verkon neuronit toimivat yleensä kahdessa vaiheessa.

1. Neuronilla on oma vektori painoja $[w_1 \dots w_n]$. $w_i \approx$ kuinka tärkeä x_i on.
2. Tärkeys $= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$.
3. Syötteiden summalle tehdään muunnos g, joka antaa neuronin ulostulon: $g(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) = z$. Muutos g on mallintaa neuronin **aktivoitumista**. Latistaa "tärkeyden" nollaan tai korostaa sitä suuresti.
4. Lineaariset summat + epälineaarinen aktivoituminen $\rightarrow$ monimutkaisia asioita.



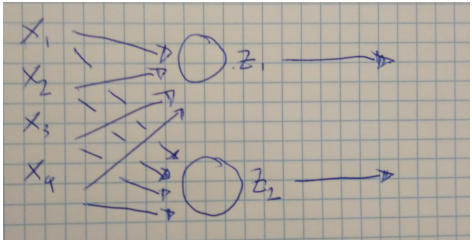
Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}\mathbf{x_1} + w_{1,2}\mathbf{x_2} + \cdots + w_{1,n}\mathbf{x_n}) = z_1$



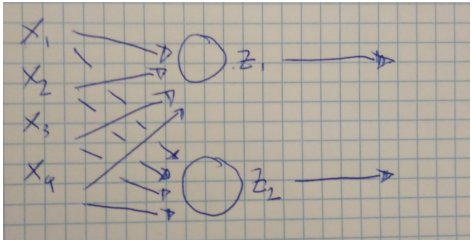
Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}\mathbf{x_1} + w_{1,2}\mathbf{x_2} + \cdots + w_{1,n}\mathbf{x_n}) = z_1$
- $g(w_{2,1}\mathbf{x_1} + w_{2,2}\mathbf{x_2} + \cdots + w_{2,n}\mathbf{x_n}) = z_2$



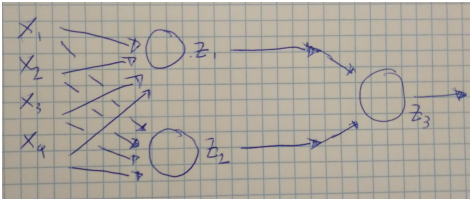
Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$



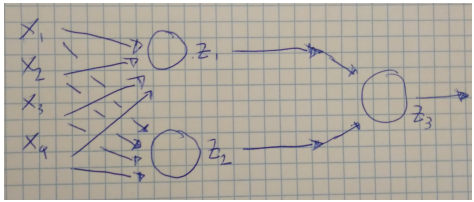
Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$
- $g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) = \mathbf{z_3}$
- jne.



Neuroverkko on verkko myös matemaattisesti

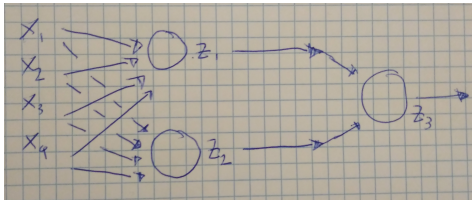
- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$
- $g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) = \mathbf{z_3}$
- jne.



- Viimeisen ulostulon y voisi kirjoittaa isona yhtälönä:

Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$
- $g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) = \mathbf{z_3}$
- jne.

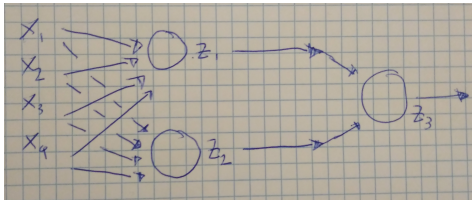


- Viimeisen ulostulon y voisi kirjoittaa isona yhtälönä:

$$y = g(w_i\mathbf{z_3} + \dots) = \quad = g(w_i g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) + \dots)$$

Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$
- $g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) = \mathbf{z_3}$
- jne.



- Viimeisen ulostulon y voisi kirjoittaa isona yhtälönä:

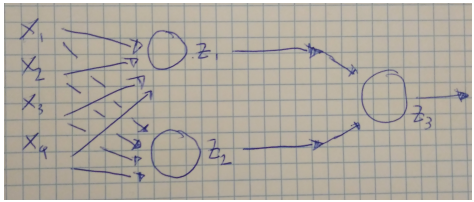
$$y = g(w_i\mathbf{z_3} + \dots) = \quad = g(w_i g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) + \dots)$$

yleisesti $y = M(x)$ missä $x = [x_1 \dots, x_n]$

tai

Neuroverkko on verkko myös matemaattisesti

- $g(w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,n}x_n) = \mathbf{z_1}$
- $g(w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,n}x_n) = \mathbf{z_2}$
- $g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) = \mathbf{z_3}$
- jne.



- Viimeisen ulostulon y voisi kirjoittaa isona yhtälönä:

$$y = g(w_i\mathbf{z_3} + \dots) = \quad = g(w_i g(w_{3,1}\mathbf{z_1} + w_{3,2}\mathbf{z_2}) + \dots)$$

$$\text{yleisesti} \quad y = M(x) \qquad \text{missä } x = [x_1 \dots, x_n]$$

tai

$$y = M(x) = g(W_L g(W_{L-1} g(\dots g(W_1 x)))),$$

missä W_L matriisi kaikista tason L vektoreista w_l.

...

Hengähdystauko!

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

- Jokaisella neuronilla on
  - vektori painoja $[w_1 \ldots w_n]$, jotka kertovat mitä mieltä neuroni on syötteistä $x_1, \ldots x_n$
  - ja aktivointifunktio g joko korostaa tai latistaa neuronin mielipidettä.

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

- Jokaisella neuronilla on
  - vektori painoja $[w_1 \ldots w_n]$, jotka kertovat mitä mieltä neuroni on syötteistä $x_1, \ldots x_n$
  - ja aktivointifunktio g joko korostaa tai latistaa neuronin mielipidettä.

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

- Jokaisella neuronilla on
  - vektori painoja $[w_1 \ldots w_n]$, jotka kertovat mitä mieltä neuroni on syötteistä $x_1, \ldots x_n$
  - ja aktivointifunktio g joko korostaa tai latistaa neuronin mielipidettä.
- Neuronit muodostavat verkon M. (Merkintä: W_l = tason l painot matriisina.)

$$M(x) = g(W_L g(W_{L-1} g(\ldots g(W_1 x))))$$

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

- Jokaisella neuronilla on
  - vektori painoja $[w_1 \ldots w_n]$, jotka kertovat mitä mieltä neuroni on syötteistä $x_1, \ldots x_n$
  - ja aktivointifunktio g joko korostaa tai latistaa neuronin mielipidettä.
- Neuronit muodostavat verkon M. (Merkintä: W_l = tason l painot matriisina.)

$$M(x) = g(W_L g(W_{L-1} g(\ldots g(W_1 x))))$$

- Jos parametrit $W_1, \ldots W_L$ on valittu oikein, niin viimeinen ulostulo $M(x) = y =$ "kissa" kun ensimmäiselle tasolle annetaan kuva jossa on kissa.

Neuroverkon tiivistelmä

Tiivistelmä: Jos neuronin i sisääntulo on $[x_1, \ldots, x_n]$, sen ulostulo z_i on

$$g(w_{i,1}x_1 + w_{i,2}x_2 + \ldots w_{i,n}x_n) = g(w_i \cdot x), w_i = [w_{i,1}, \ldots, w_{i,n}].$$

- Jokaisella neuronilla on
  - vektori painoja $[w_1 \ldots w_n]$, jotka kertovat mitä mieltä neuroni on syötteistä $x_1, \ldots x_n$
  - ja aktivointifunktio g joko korostaa tai latistaa neuronin mielipidettä.
- Neuronit muodostavat verkon M. (Merkintä: W_l = tason l painot matriisina.)

$$M(x) = g(W_L g(W_{L-1} g(\ldots g(W_1 x))))$$

- Jos parametrit $W_1, \ldots W_L$ on valittu oikein, niin viimeinen ulostulo $M(x) = y =$ "kissa" kun ensimmäiselle tasolle annetaan kuva jossa on kissa.

Eli miten niiden painojen oppiminen tapahtuu?

Sisältö

Neuroverkon ulostulo

- Muistutus: neuroverkko M syö sisääntulot x, tekee asioita (kullakin tasolla $l = 1 \ldots L$ parametrit W_l), ja sylkee ulostulon $M(x) = y =$ "kissa".

Neuroverkon ulostulo

- Muistutus: neuroverkko M syö sisääntulot x, tekee asioita (kullakin tasolla $l = 1 \dots L$ parametrit W_l), ja sylkee ulostulon $M(x) = y =$ "kissa".
- Oikeastaan y on vektori todennäköisyyksiä $y = [0.99, 0.01, 0.0, \dots]$.

Neuroverkon ulostulo

- Muistutus: neuroverkko M syö sisääntulot x, tekee asioita (kullakin tasolla $l = 1 \dots L$ parametrit W_l), ja sylkee ulostulon $M(x) = y =$ "kissa".
- Oikeastaan y on vektori todennäköisyyksiä $y = [0.99, 0.01, 0.0, \dots]$.

  ... joiden merkitys on sovittu:

$$\rightarrow \quad M_w(x) = y$$

	y
kissa	0.99
koira	0.01
laiskiainen	0.0
...	..

Neuroverkon ulostulo ja virhe

- . . . siis jos painot W oikein. Jos painot eivät ole oikein vastaus on väärin:

Neuroverkon ulostulo ja virhe

- ...siis jos painot W oikein. Jos painot eivät ole oikein vastaus on väärin:

$$\rightarrow \quad M_W(x) = y$$

	y
kissa	0.0865
koira	0.11234
laiskiainen	0.349
...	(kohinaa)

Neuroverkon ulostulo ja virhe

- ...siis jos painot W oikein. Jos painot eivät ole oikein vastaus on väärin:

$$\rightarrow \quad M_W(x) = y$$

	y
kissa	0.0865
koira	0.11234
laiskiainen	0.349
...	(kohinaa)

Neuroverkon ulostulo ja virhe (ii)

Kertaus:

- Sarja kuvia eläimistä joille tiedämme oikeat vektorit $=$ opetusdata $= y_{\text{opetus}}$.

Neuroverkon ulostulo ja virhe (ii)

Kertaus:

- Sarja kuvia eläimistä joille tiedämme oikeat vektorit = opetusdata = y_{opetus}.
- Valitaan jokin sopiva virhefunktio L jolle syötetään sekä neuroverkon ennuste y ja y_{opetus}. Jos $y = y_{\text{opetus}}$ niin $L = 0$.

Neuroverkon ulostulo ja virhe (ii)

Kertaus:

- Sarja kuvia eläimistä joille tiedämme oikeat vektorit = opetusdata = y_{opetus}.
- Valitaan jokin sopiva virhefunktio L jolle syötetään sekä neuroverkon ennuste y ja y_{opetus}. Jos $y = y_{\text{opetus}}$ niin $L = 0$.
- Kuva $x \rightarrow$ verkko antaa ulostulon $M_W(x) = y \rightarrow$ lasketaan $L(y, y_{\text{opetus}})$.

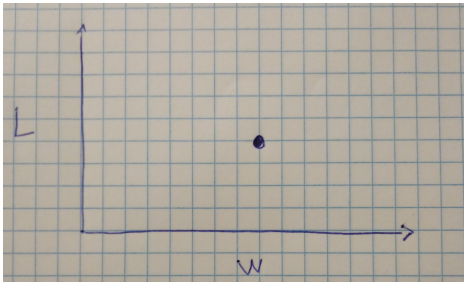
Neuroverkon ulostulo ja virhe (ii)

Kertaus:

- Sarja kuvia eläimistä joille tiedämme oikeat vektorit = opetusdata = y_{opetus}.
- Valitaan jokin sopiva virhefunktio L jolle syötetään sekä neuroverkon ennuste y ja y_{opetus}. Jos $y = y_{\text{opetus}}$ niin $L = 0$.
- Kuva $x \rightarrow$ verkko antaa ulostulon $M_W(x) = y \rightarrow$ lasketaan $L(y, y_{\text{opetus}})$.
- Haluamme löytää parametrit $W = [w_{1,1}, \ldots, w_{h,k}]$ joilla virhe on pieni kaikille kuville.

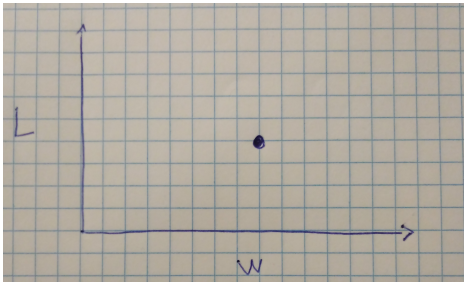
Haluamme löytää parametrit joilla virhe on pieni

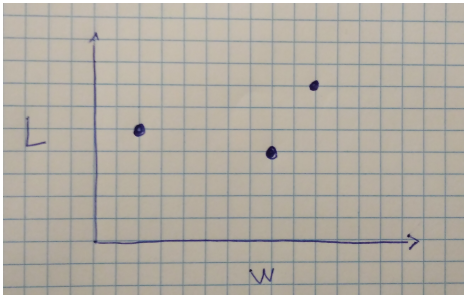
- Arvataan?



Haluamme löytää parametrit joilla virhe on pieni

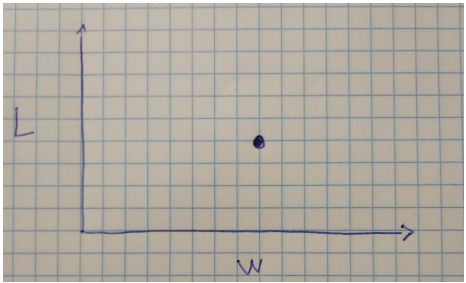
- Arvataan?

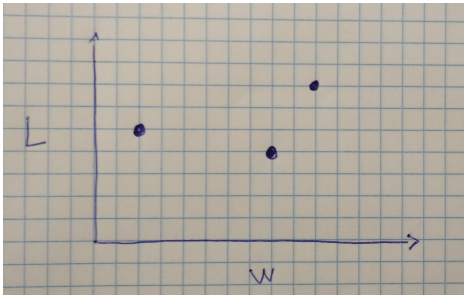




Haluamme löytää parametrit joilla virhe on pieni

- Arvataan?

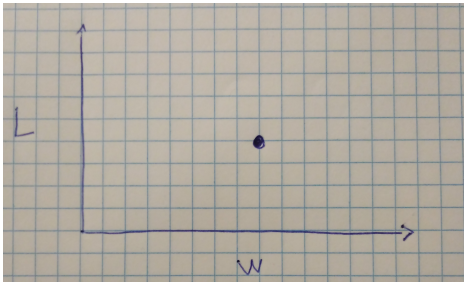


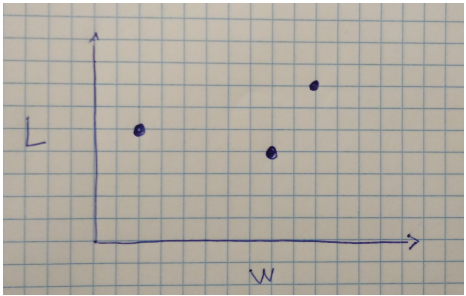


- Neuroverkoissa parametreja $\approx h \times k$,

Haluamme löytää parametrit joilla virhe on pieni

- Arvataan?





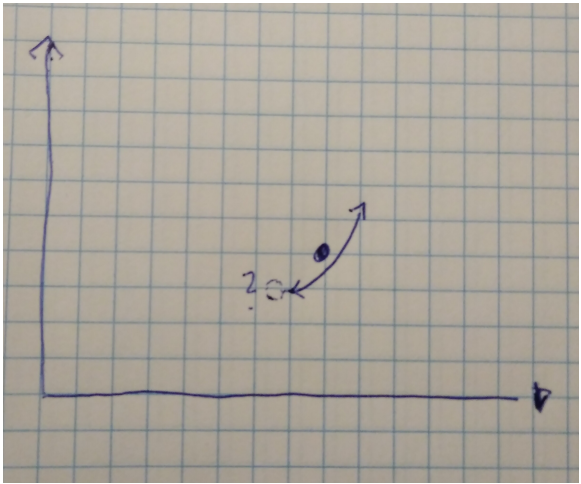
- Neuroverkoissa parametreja $\approx h \times k$,
  - $h =$ neuroverkon syvyys,
  - $k =$ alemman tason neuronit. $h \times k =$ **paljon**.

Hengähdystauko.

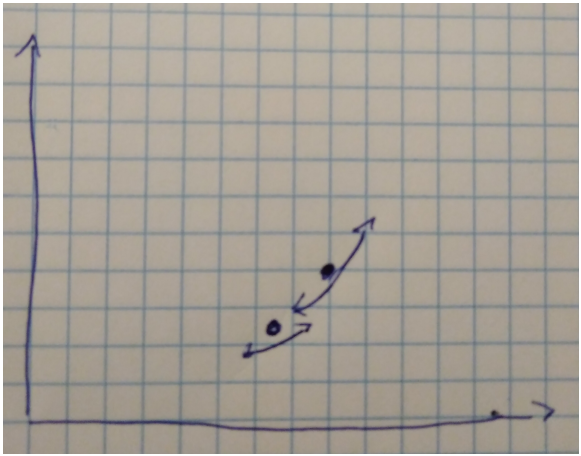
Hengähdystauko.

DERIVAATTA.

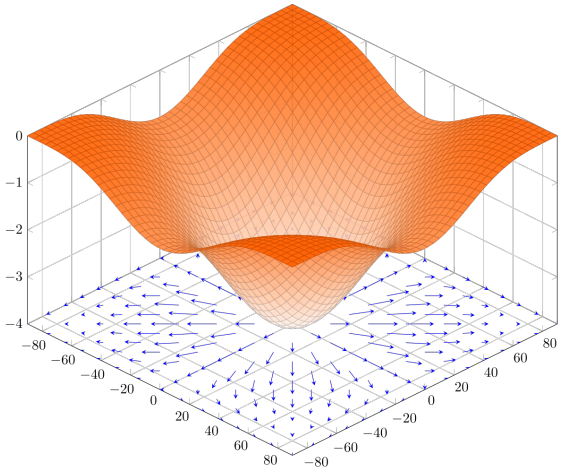
Yksiulotteinen virhemaasto ja derivaatta



Yksiulotteinen virhemaasto ja derivaatta



Kaksiulotteinen virhemaasto ja gradientti



Neuroverkon virhemaasto ja gradientti

- Virhefunktio

$$L = L(y, y_{\text{opetus}}) = L(M(x_{\text{opetus}}), y_{\text{opetus}})$$

Neuroverkon virhemaasto ja gradientti

- Virhefunktio

$$L = L(y, y_{\text{opetus}}) = L(M(x_{\text{opetus}}), y_{\text{opetus}})$$

- Derivaatta $\frac{dL}{dw_{h,i}}$. Gradientti $\left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right] = \nabla_{w_h} L$.

Neuroverkon virhemaasto ja gradientti

- Virhefunktio

$$L = L(y, y_{\text{opetus}}) = L(M(x_{\text{opetus}}), y_{\text{opetus}})$$

- Derivaatta $\frac{dL}{d_{w_{h,i}}}$. Gradientti $\left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right] = \nabla_{w_h} L$.
- Neuroverkon funktio muoto tunnetaan:

$$y = M(x) = g(W_L g(W_{L-1} g(\ldots g(W_1 x))))$$

Neuroverkon virhemaasto ja gradientti

- Virhefunktio
$$L = L(y, y_{\text{opetus}}) = L(M(x_{\text{opetus}}), y_{\text{opetus}})$$
- Derivaatta $\frac{dL}{d_{w_{h,i}}}$. Gradientti $\left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right] = \nabla_{w_h} L$.
- Neuroverkon funktio muoto tunnetaan:
$$y = M(x) = g(W_L g(W_{L-1} g(\ldots g(W_1 x))))$$
- joten $\nabla_{w_h} L$ voidaan laskea jokaiselle tasolle h.

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y)$$

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y) = \frac{d}{dw_{ij}} L(g(W_l g(\dots))), y)$$

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y) = \frac{d}{dw_{ij}} L(g(W_l g(\dots))), y)$$

$$= L'(g(W_l g(\dots), y) \frac{d}{dw_{ij}} g(W_l g(\dots))$$

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y) = \frac{d}{dw_{ij}} L(g(W_l g(\dots))), y)$$

$$= L'(g(W_l g(\dots), y) \frac{d}{dw_{ij}} g(W_l g(\dots))$$

$$= L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots) \frac{d}{dw_{ij}} W_l g(\dots)\right)$$

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y) = \frac{d}{dw_{ij}} L(g(W_l g(\dots))), y)$$

$$= L'(g(W_l g(\dots), y) \frac{d}{dw_{ij}} g(W_l g(\dots))$$

$$= L'(g(W_l g(\dots), y)\, g' \left(W_l g(\dots) \frac{d}{dw_{ij}} W_l g(\dots) \right)$$

$$= L'(g(W_l g(\dots), y)\, g' \left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots) \right)$$

Virheen takaisinleviämisalgoritmi

- Backpropagation eli "takaisinleviäminen" on tehokas tapa laskea laskea gradientit. Miten?
- Intuitio (lukiolaisille): Ketjusääntö! $\frac{d}{dx} f(h(x)) = f'(h(x))h'(x)$.
- Sovellettuna neuroverkon virheeseen $L(M(x), y)$:

$$\frac{d}{dw_{ij}} L(M(x), y) = \frac{d}{dw_{ij}} L(g(W_l g(\dots))), y)$$

$$= L'(g(W_l g(\dots), y) \frac{d}{dw_{ij}} g(W_l g(\dots))$$

$$= L'(g(W_l g(\dots), y)\, g' \left(W_l g(\dots) \frac{d}{dw_{ij}} W_l g(\dots) \right)$$

$$= L'(g(W_l g(\dots), y)\, g' \left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots) \right)$$

ja jatkuu ja jatkuu

Mitäh?

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$
- Sisältää
  1. termejä jotka on jo laskettu kun mentiin verkossa eteenpäin,

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$
- Sisältää
  1. termejä jotka on jo laskettu kun mentiin verkossa eteenpäin,
  2. funktioiden L ja g derivaattoja,

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$
- Sisältää
  1. termejä jotka on jo laskettu kun mentiin verkossa eteenpäin,
  2. funktioiden L ja g derivaattoja,
  3. toistuvaa rakennetta, jossa tason l derivaatat riippuvat ainoastaan edessä olevien tasojen $l+1, l+2, \dots, L$ termeistä.

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$
- Sisältää
  1. termejä jotka on jo laskettu kun mentiin verkossa eteenpäin,
  2. funktioiden L ja g derivaattoja,
  3. toistuvaa rakennetta, jossa tason l derivaatat riippuvat ainoastaan edessä olevien tasojen $l+1, l+2, \dots, L$ termeistä.
- . . . ??? PROFIT:

Mitäh?

- Tämä *on* esityksen vaikein asia.
- $L'(g(W_l g(\dots), y)\, g'\left(W_l g(\dots)\, W_l \frac{d}{dw_{ij}} g(\dots)\right)$
- Sisältää
  1. termejä jotka on jo laskettu kun mentiin verkossa eteenpäin,
  2. funktioiden L ja g derivaattoja,
  3. toistuvaa rakennetta, jossa tason l derivaatat riippuvat ainoastaan edessä olevien tasojen $l+1, l+2, \dots, L$ termeistä.
- . . . ??? PROFIT:
- Lasku voidaan tehdä tehokkaasti menemällä *taaksepäin*.

Takaisin leviäminen

- Neuroverkon ulostulo $M(x)$ ja virhe $L(M(x), y)$ lasketaan ensin taso tasolta **eteenpäin** (*forward pass*)

Takaisin leviäminen

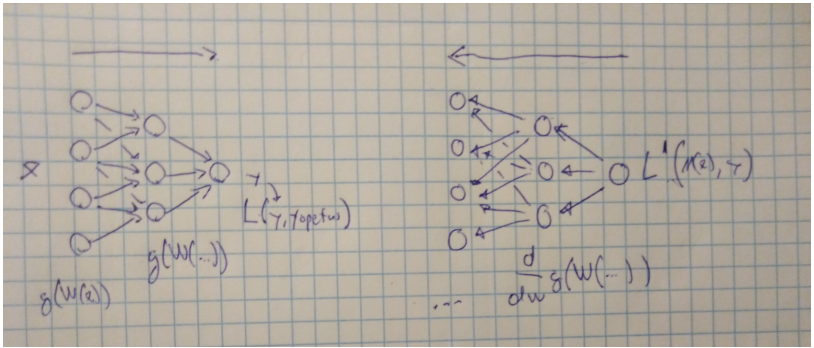
- Neuroverkon ulostulo $M(x)$ ja virhe $L(M(x), y)$ lasketaan ensin taso tasolta **eteenpäin** (*forward pass*)
- ... jos laskujen välivaiheet tallennetaan, *niin* jokainen $\frac{d}{dw_{ij}} L(\dots)$ saadaan menemällä lähes samat askeleet **taaksepäin** (*backpropagation*, "takaisin leviäminen")

Takaisin leviäminen

- Neuroverkon ulostulo $M(x)$ ja virhe $L(M(x), y)$ lasketaan ensin taso tasolta **eteenpäin** (*forward pass*)
- ... jos laskujen välivaiheet tallennetaan, *niin* jokainen $\frac{d}{dw_{ij}} L(\dots)$ saadaan menemällä lähes samat askeleet **taaksepäin** (*backpropagation*, "takaisin leviäminen")
  - (ja tekemällä muutamia yksinkertaisia derivointilaskuja g' ja L').

Takaisin leviäminen

- Neuroverkon ulostulo $M(x)$ ja virhe $L(M(x), y)$ lasketaan ensin taso tasolta **eteenpäin** (*forward pass*)
- . . . jos laskujen välivaiheet tallennetaan, *niin* jokainen $\frac{d}{dw_{ij}} L(\dots)$ saadaan menemällä lähes samat askeleet **taaksepäin** (*backpropagation*, "takaisin leviäminen")
  - (ja tekemällä muutamia yksinkertaisia derivointilaskuja g' ja L').



Mitäh??!?!?!?

Mitäh??!?!?!?

Uusi hengähdystauko.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \rightarrow$ suuri virhe $L(M(x), y_{\text{opetus}})$.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \rightarrow$ suuri virhe $L(M(x), y_{\text{opetus}})$.
3. Gradientti $\nabla L_{w_h} = \left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right]$ kertoo parhaan arvauksen mistä etsiä uusia parametrin arvoja tason h parametreille $w_{h,j}$.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \to$ suuri virhe $L(M(x), y_{\text{opetus}})$.
3. Gradientti $\nabla L_{w_h} = \left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right]$ kertoo parhaan arvauksen mistä etsiä uusia parametrin arvoja tason h parametreille $w_{h,j}$.
4. Backpropagation-algoritmi on tehokas tapa laskea gradientti kaikille $w_{i,j}$ etenemällä taso kerrallaan takaisinpäin.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \rightarrow$ suuri virhe $L(M(x), y_{\text{opetus}})$.
3. Gradientti $\nabla L_{w_h} = \left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right]$ kertoo parhaan arvauksen mistä etsiä uusia parametrin arvoja tason h parametreille $w_{h,j}$.
4. Backpropagation-algoritmi on tehokas tapa laskea gradientti kaikille $w_{i,j}$ etenemällä taso kerrallaan takaisinpäin.
5. Kun gradientti on laskettu, parametreja W siirretään pieni askel ∇L:n osoittamaan suuntaan.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \rightarrow$ suuri virhe $L(M(x), y_{\text{opetus}})$.
3. Gradientti $\nabla L_{w_h} = \left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right]$ kertoo parhaan arvauksen mistä etsiä uusia parametrin arvoja tason h parametreille $w_{h,j}$.
4. Backpropagation-algoritmi on tehokas tapa laskea gradientti kaikille $w_{i,j}$ etenemällä taso kerrallaan takaisinpäin.
5. Kun gradientti on laskettu, parametreja W siirretään pieni askel ∇L:n osoittamaan suuntaan.
6. Algoritmi toistetaan uudestaan kunnes virhe pieni.

Tiivistelmä:

1. Neuroverkko koostuu tasoista neuroneita.
2. Huonoilla parametreilla neuroverkon ulostulot $M(x) \to$ suuri virhe $L(M(x), y_{\text{opetus}})$.
3. Gradientti $\nabla L_{w_h} = \left[\frac{\partial L}{\partial w_{h,1}}, \ldots, \frac{\partial L}{\partial w_{h,k}}\right]$ kertoo parhaan arvauksen mistä etsiä uusia parametrin arvoja tason h parametreille $w_{h,j}$.
4. Backpropagation-algoritmi on tehokas tapa laskea gradientti kaikille $w_{i,j}$ etenemällä taso kerrallaan takaisinpäin.
5. Kun gradientti on laskettu, parametreja W siirretään pieni askel ∇L:n osoittamaan suuntaan.
6. Algoritmi toistetaan uudestaan kunnes virhe pieni.

TLDR: Neuroverko opetetaan laittamalla tietokone laskemaan paljon isoja derivaattalaskuja.

Sisältö

Yksityiskohtia joita tässä ei käsitelty

- Paljon teknisiä yksityiskohtia: millainen on verkon rakenne, kuinka pitkiääaskelia otetaan, jne.
- Jyrkimmän pudotuksen suunnan valitseminen ei välttämättä johda parhaimpaan mahdolliseen lopputulokseen. Satunnaisuutta syntyy vaihtelemalla harjoituskuvia askeleiden välissä.
- Neuroverkon harjoittaminen ei muutenkaan ole käytännössä suoraviivaista.
- ...

...