Aalto-yliopisto
Sähkötekniikan korkeakoulu

ELEC-C1110 Automaatio- ja systeemitekniikan perusteet

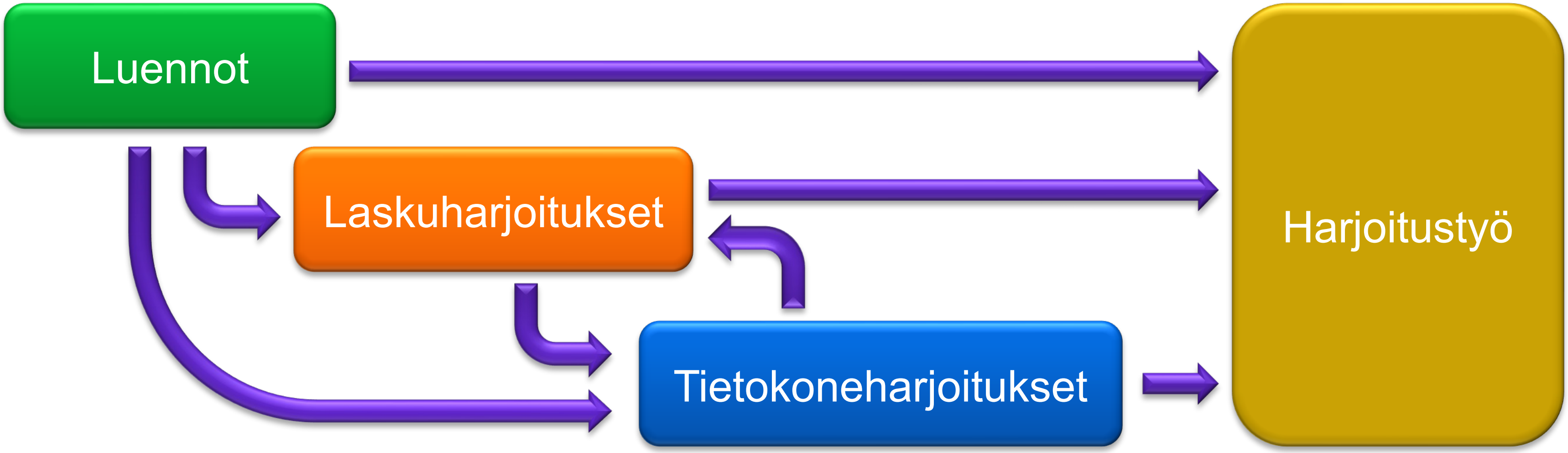
Luento 4

Mallin oppiminen näytteistä

Joni Pajarinen, 6.2.2023

Tämä luento

- Yleisen mallin oppiminen näytteistä



Tämän luennon aiheet

- Viimeksi:
  - Anturit näytteiden mittaamiseen
  - Mallin kalibrointi / mallin parametrien estimointi näytteistä
- Tänään:
  - Mallin oppiminen näytteistä
  - Epälineaarisen mallin oppiminen, gradienttimenetelmä, kantafunktiomalli, neuroverkko
  - Valmiin ja tuntemattoman mallin yhdistäminen

Systeemin mallintaminen

Mitä jos systeemissä on jotain minkä määrittäminen tai mallintaminen on hankalaa?

Voidaanko oppia yleinen malli näytteistä ilman että muodostetaan täydellinen fysikaalinen malli?

Mallin oppiminen

Hukkafunktio (“loss function”)

- Mitataan N näytettä $D = (y(t_1), y(t_2), \dots, y(t_N))$
- Sovitetaan malli $f(x, \theta)$ näytteisiin
- Pienimmän neliövirheen sovitus valitsee parametrit θ, jotka minimoivat virheen

$$L(\theta) = \sum_n \left(f(x_n, \theta) - y(t_n)\right)^2$$

- Muitakin virheitä voidaan minimoida, esim. $\sum_n |f(x_n, \theta) - y(t_n)|$ mutta neliövirhe yleisin

Lineaarinen malli

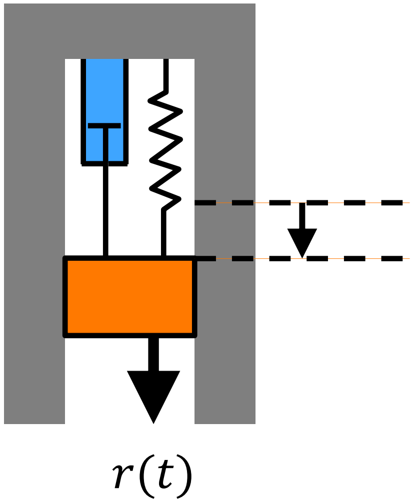
- Lineaarisessa mallissa

$$f(x,\theta) = \sum_i \theta_i \phi_i(x)$$

funktiot $\phi_i(x)$ voivat olla mitä tahansa funktioita

- Esim. jousi-massa-vaimentimessa

$$m\ddot{x}(t_n) + b\dot{x}(t_n) + kx(t_n) = r(t_n) \Rightarrow$$
$$m\phi_1(t_n) + b\phi_2(t_n) + k\phi_3(t_n) = y(t_n)$$
$$\theta_1\phi_1(t_n) + \theta_2\phi_2(t_n) + \theta_3\phi_3(t_n) = y(t_n)$$

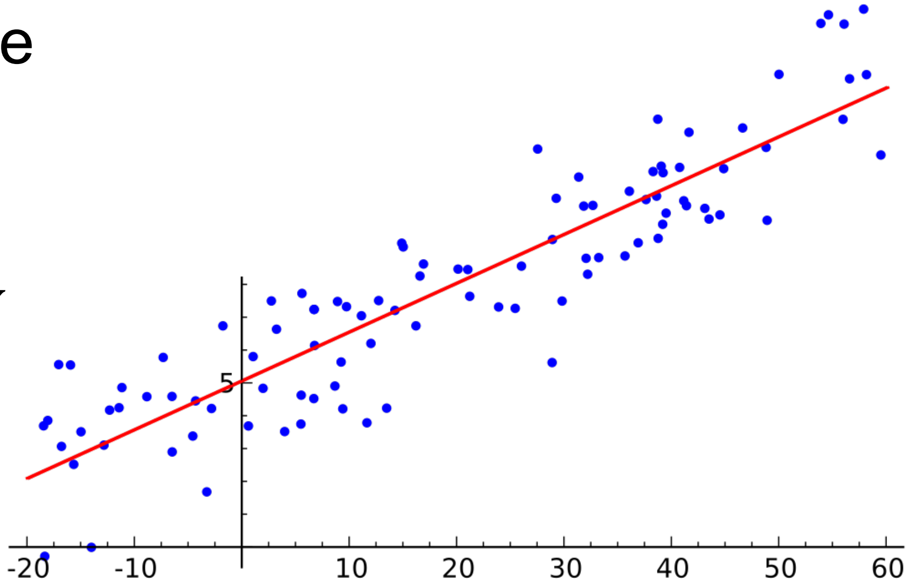


Lineaarisen mallin sovitus näytteisiin

Kahden vektorin pistetulo

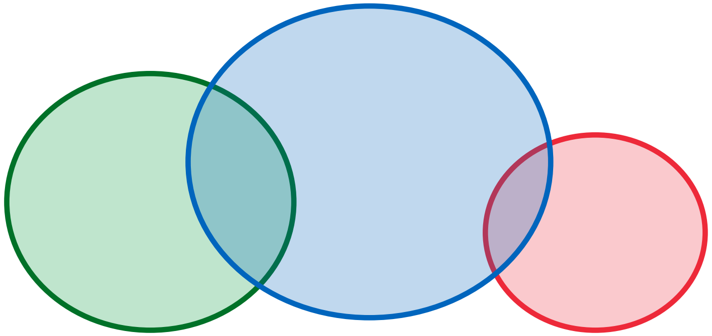
- Lineaarinen malli $f(x,\theta) = \sum_i \theta_i \phi_i(x) = \phi^T(x)\theta$
- Merkitään matriisin Φ elementtejä $\Phi_{\mathrm{n,i}} = \phi_i(x_n)$
- Vektori Y koostuu $y(x_n)$ elementeistä
- Kun minimoidaan neliövirhe

  $\sum_n (\phi^T(x_n)\theta - y(x_n))^2 =$
  $(\Phi\theta - Y)^T(\Phi\theta - Y)$
- saadaan $\theta = (\Phi^T\Phi)^{-1}\Phi^T Y$



Esimerkkejä lineaarisista malleista

- Lineaarinen malli yleisesti $f(x,\theta) = \sum_i \theta_i \phi_i(x)$
- Polynomipohjainen $f(x,\theta) = \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M$
  - Polynomisessa mallissa x on skalaari arvo, ei vektori
- Kantafunktiopohjainen $f(x,\theta) = \sum_i \theta_i e^{-\sigma_\mathrm{i}(x-\mu_i)^T(x-\mu_i)}$
  - μ_i vektori määrittää kantafunktion i keskuspisteen
  - σ_i määrittää kantafunktion i leveyden
  - Funktio $f(x,\theta)$ lineaarinen suhteessa θ_i parametreihin



Epälineaarinen malli

- Esimerkkinä epälineaarinen kantafunktiopohjainen malli

$$f(x,\theta) = f\left(x,\hat{\theta},\mu_1,\dots,\mu_{M,}\Sigma_1,\dots,\Sigma_{\mathrm{M}}\right) = \sum_i \hat{\theta}_i e^{-\sigma_i (x-\mu_i)^T (x-\mu_i)}$$

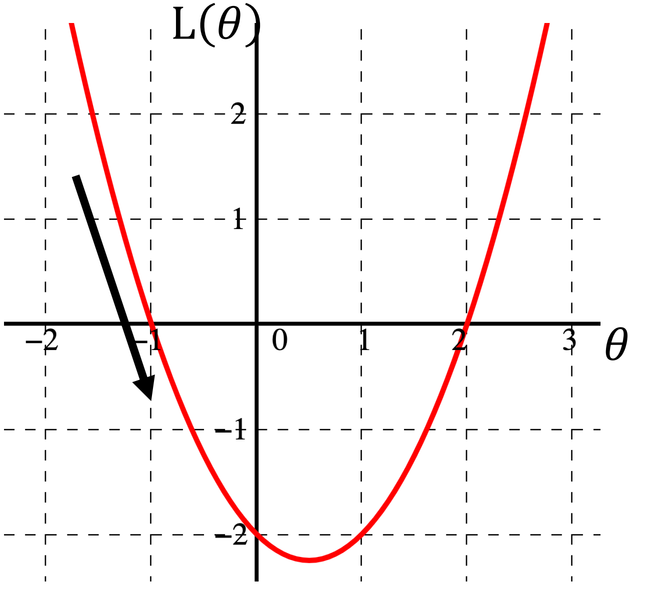
  - μ_i vektorit ja σ_i arvot osana parametreja θ. Malli on epälineaarinen *suhteessa parametreihin* μ_i ja σ_i
- Miten sovitetaan parametrit θ kun malli $f(x,\theta)$ ei ole lineaarinen?
- Minimoidaan neliövirhe $\sum_n \left(f(x_n,\theta) - y(t_n)\right)^2$

Epälineaarinen malli

- Minimoidaan neliövirhe $L(\theta) = \sum_n (f(x_n, \theta) - y(t_n))^2$
- Gradienttimenetelmä seuraa gradienttia $\frac{\delta L(\theta)}{\delta \theta}$
- Valitaan θ_0, vektori joka sisältää parametrialkuarvot
- Gradienttimenetelmässä parametrit päivitetään kaavalla

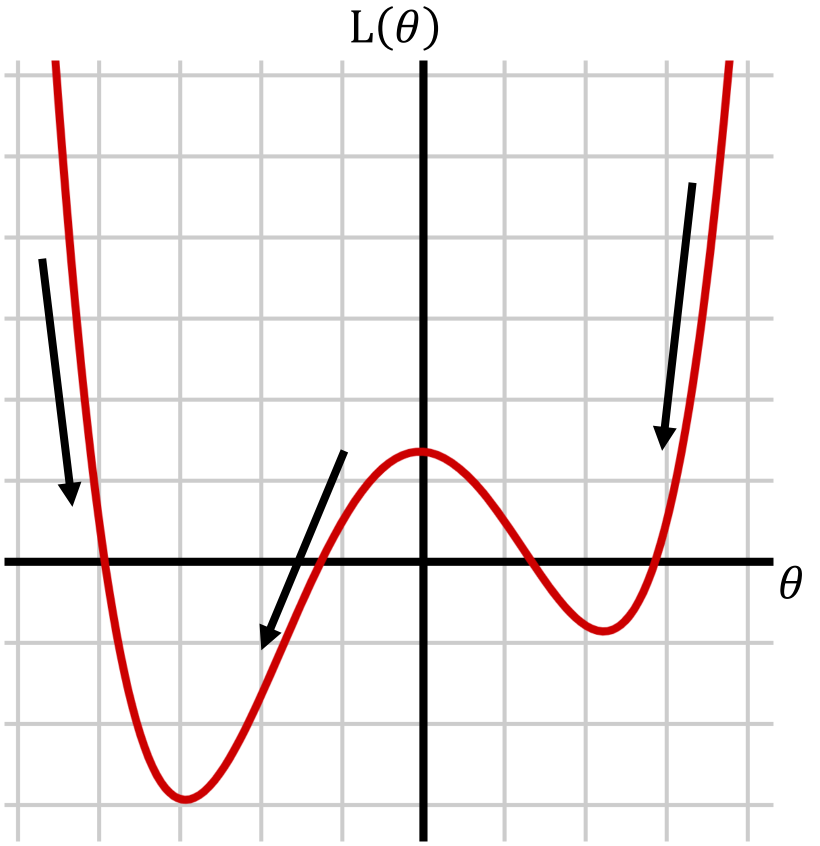
$$\theta^{k+1} = \theta^k - \alpha \frac{\delta L(\theta)}{\delta \theta}$$

- Kun $f(x, \theta)$ on lineaarinen, löydetään globaali minimi hukkafunktiolle $L(\theta)$



Epälineaarinen malli: lokaalit minimit

- Kun $f(x, \theta)$ on epälineaarinen, löydetään lokaali minimi
- Lokaaleja minimejä voi olla monta, riippuen mallista
- Se mihin minimiin päädytään riippuu alkuarvosta
- Miten löydetään pieni minimi?

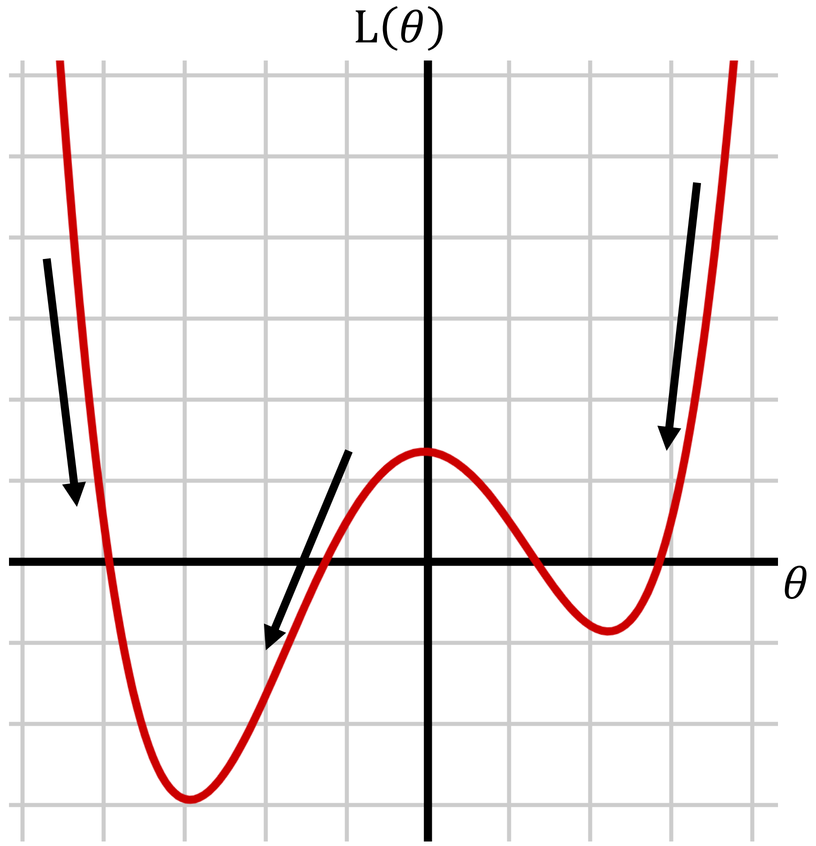


Epälineaarinen malli: mallin sovittaminen

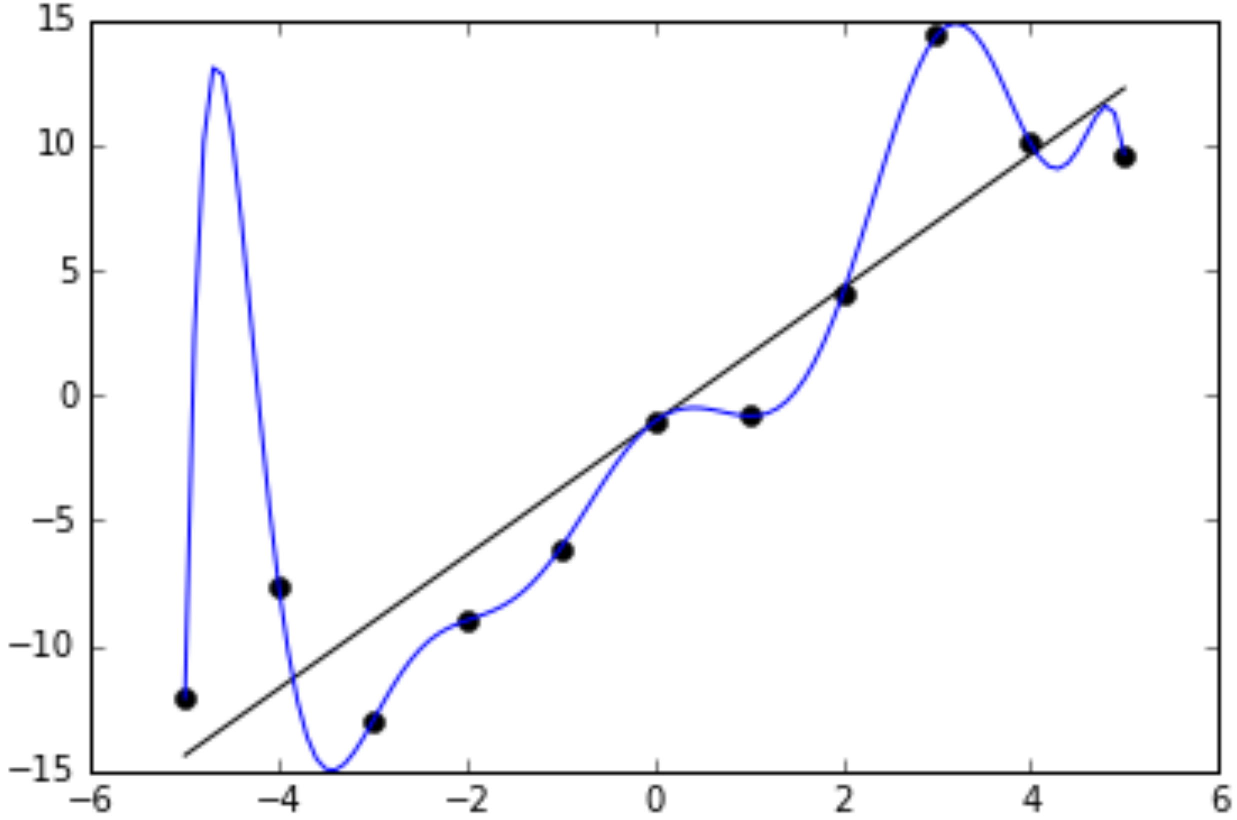
- Kun mallissa on monta parametria θ_i

1. Gradientilla on useampi muuttuja jota muuttamalla voi pienentää $L(\theta)$:ää
2. Lokaaleja minimejä on useampi

- $\rightarrow$ Helpompi löytää pieni minimi
- $\rightarrow$ Malli voi ylisovittaa (”overfit”) näytteisiin

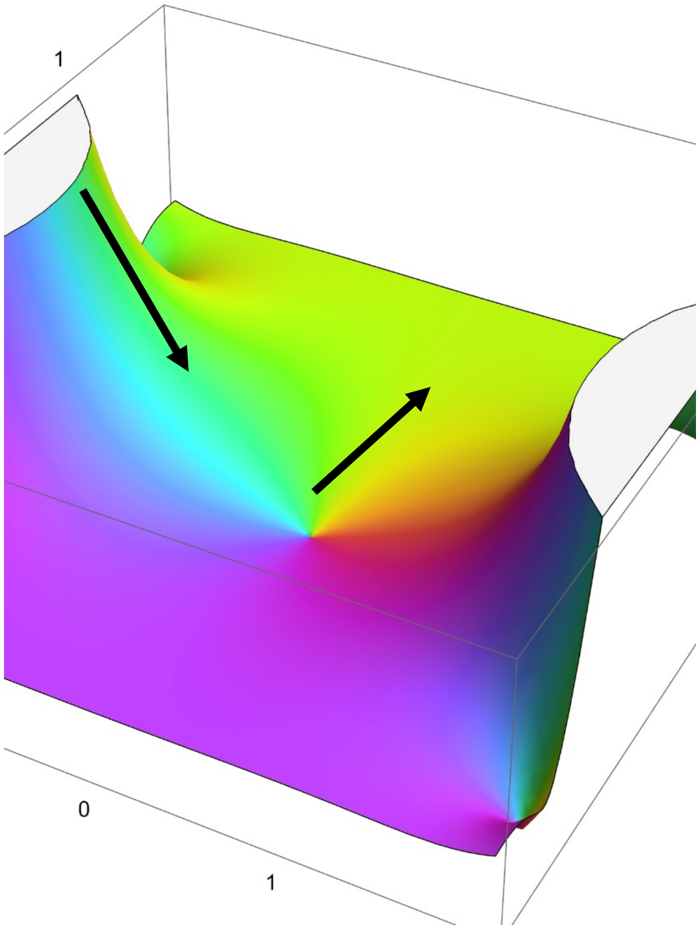


Epälineaarinen malli: ylisovittaminen



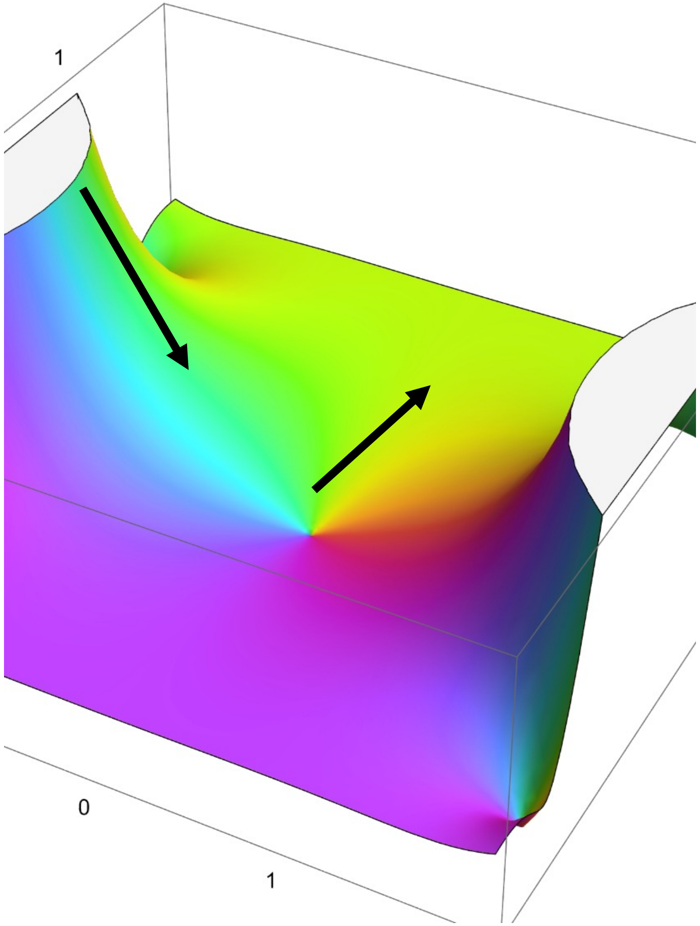
Epälineaarinen malli: askelpituus

- Gradienttimenetelmässä parametrit päivitetään kaavalla $\theta^{k+1} = \theta^k - \alpha \frac{\delta L(\theta)}{\delta \theta}$
- Voidaan haluta pieniä muutoksia ”pudotuksissa” ja suuria muutoksia laaksoissa
- Askelpituus α_i voidaan valita tilanteen mukaan jokaiselle parametrille θ_i^k adaptiivisesti, jotta ei jäädä jumiin



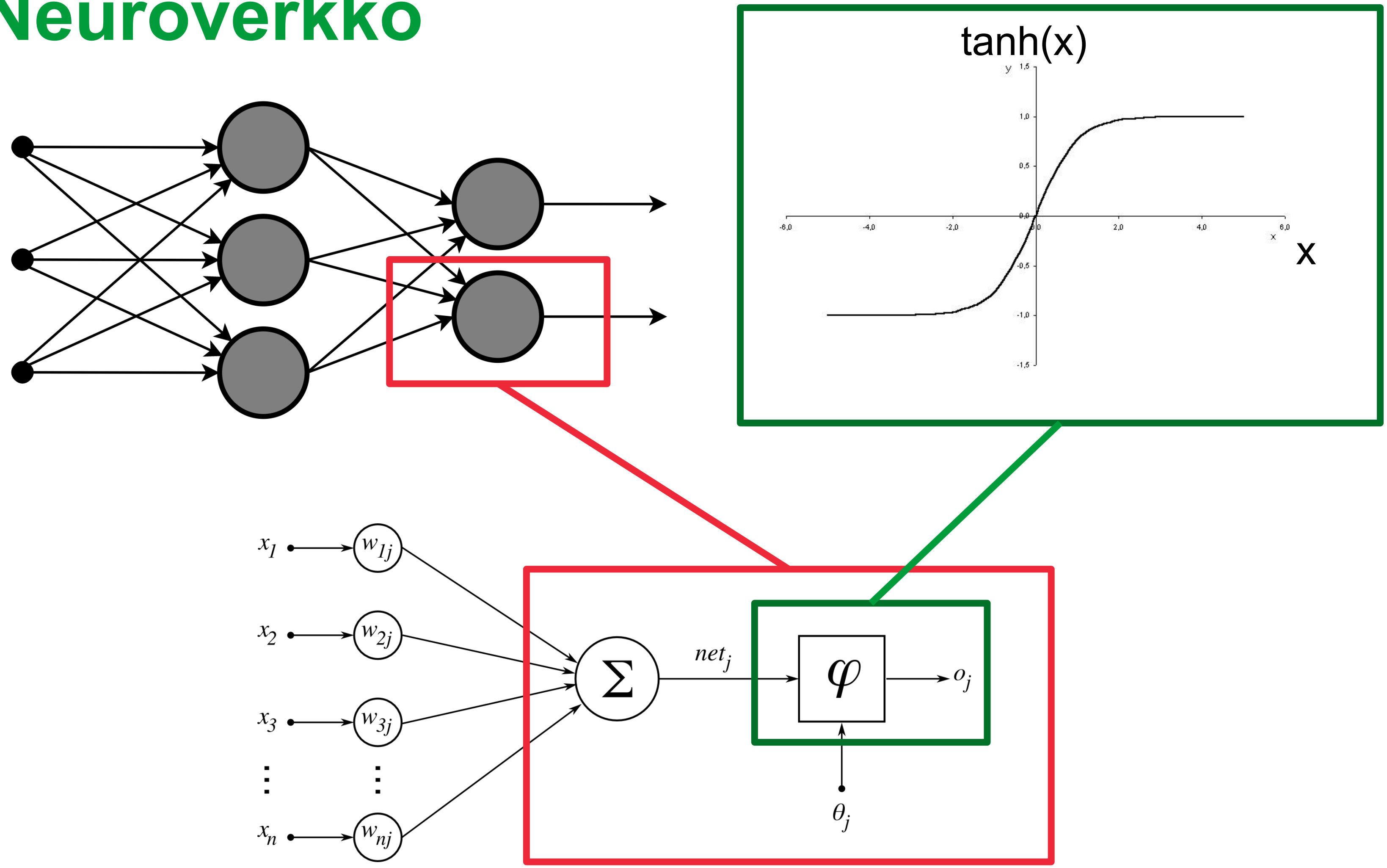
Epälineaarinen malli käytännössä

- Valitaan malli, joka pystyy kuvaamaan prosessia riittävällä tarkkuudella
- Kerätään tarpeeksi näytteitä
- Käytetään valmista python funktiota sovittamaan mallin parametrit näytteisiin



Neuroverkko

Neuroverkko



Neuroverkko

- Neuroverkko koostuu useasta kerroksesta
- Jokaisessa kerroksessa i useampi noodi j. Noodiin tulee sisään useampi signaali $x_{i-1,j}$ kerroksesta $i-1$
- Noodin ulostulo lasketaan $\mathrm{x}_{i,j} = \psi\left(\sum_k w_{i,j,k}\, x_{i-1,k} + b_{i,j}\right)$, missä $\psi()$ on epälineaarinen funktio, esim. $\tanh()$
- Noodilla on painokertoimet $w_{i,j,k}$ ja paino $b_{i,j}$ parametreina
- Tavallisella kaksikerroksisella neuroverkolla voi olla parametreja yli N^2, missä N on noodien määrä kerroksessa

Neuroverkko

- Vahvuudet:
  - Pystyy mallintamaan monimutkaisia prosesseja
  - Pystyy periaatteessa extrapoloimaan
  - Tehokas python koodi erilaisten neuroverkkojen opettamiseen on valmiina
- Heikkoudet:
  - Neuroverkon toimintaa hankala ymmärtää. Mitä neuroverkko antaa vasteeksi tietylle syötteelle?
  - Opetus voi vaatia paljon näytteitä

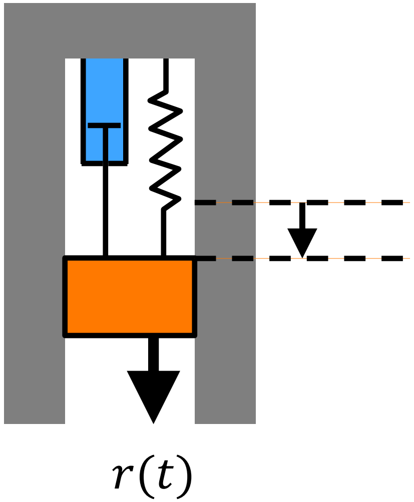
Valmiin ja tuntemattoman mallin yhdistäminen

Jousi-massa-vaimennin: epätarkka malli

- Jousi-massa-vaimentimelle on valmis malli

$$m\ddot{x}(t_n) + b\dot{x}(t_n) + kx(t_n) = r(t_n)$$
$$\mathrm{f}(\mathrm{x}, \theta) = m\ddot{x}(t_n) + b\dot{x}(t_n) + kx(t_n)$$

- Mutta entä kitka ja muut tuntemattomat asiat mitkä voivat vaikuttaa prosessiin?
- Jos $\mathrm{f}(\mathrm{x}, \theta)$ ei ole riittävän tarkka, pitäisikö oppia yleinen malli näytteistä valmiin mallin tilalle?
- Koko prosessin mallintaminen voi vaatia paljon näytteitä

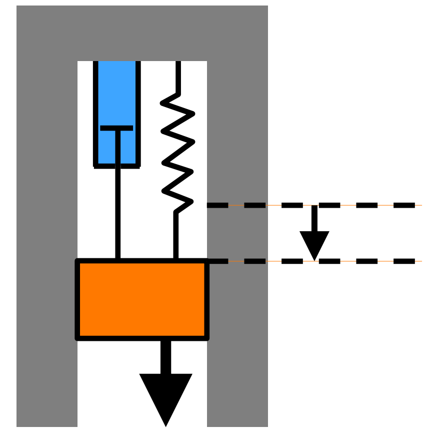


Yhdistetään valmis ja tuntematon malli

- Opitaan yleinen malli $\mathrm{g}(\mathrm{x}, \theta_g)$ vain tuntemattomalle osalle
- Kokonaismalli on $\hat{\mathrm{f}}(\mathrm{x}, \theta) = \mathrm{f}(\mathrm{x}, \theta_f) + \mathrm{g}(\mathrm{x}, \theta_g)$, missä θ koostuu valmiin mallin parametreistä θ_f ja yleisen mallin parametreistä θ_g
- θ_g löydetään minimoimalla neliövirhe

$$L(\theta_g) = \sum_n \left(\mathrm{f}(\mathrm{x_n}, \theta_f) + g(x_n, \theta_g) - y(t_n)\right)^2$$

- Minimointiin voidaan käyttää gradienttimenetelmää



$y(t) = r(t)$

θ_f pidetään vakiona

Yhteenveto

- Systeemin mallintaminen näytteistä
- Lineaarimallit
- Epälineaarimallin sovitus
- Neuroverkko
- Valmiin ja tuntemattoman mallin yhdistäminen