Minor Smart Devices
Workshop Machine Learning

dr. Wouter Bergmann Tiest

Hogeschool Rotterdam

W.M.Bergmann.Tiest@hr.nl

Machine Learning

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Machine Learning

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Programma

- Hoe werkt een neuraal netwerk?
- Lineaire regressie met een neuraal netwerk.
- Aan de slag met een voorbeeld: slimme thermostaat.

Neuraal netwerk

Neuraal netwerk

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

Neuraal netwerk

Lineaire regressie

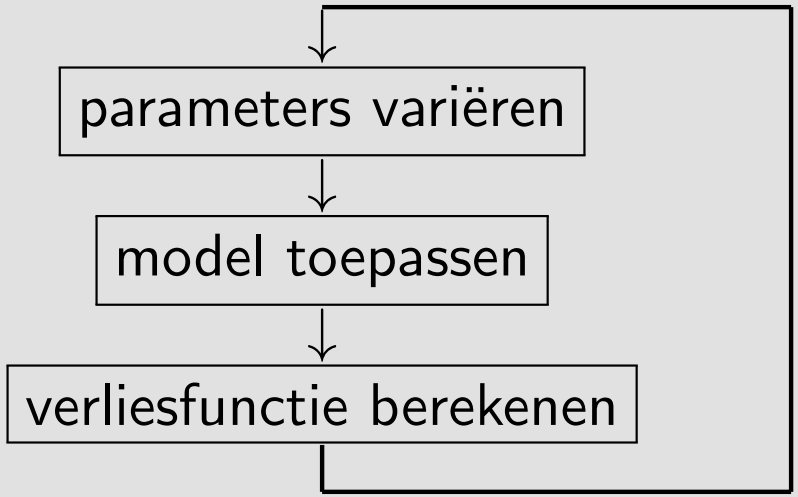
Aan de slag: slimme thermostaat

Hoe werkt een neuraal netwerk?

- Imitatie van biologische hersenen.
- Eenvoudige cellen die met elkaar verbonden zijn en elkaar activeren.
- Herkent patronen.
- Toepassingen: computer vision, spraakherkenning, natural language processing, handschriftherkenning.
- Leren van een groot aantal voorbeelden (probleem & oplossing).
- Door variëren van de parameters van het netwerk, model optimaliseren.
- Vervolgens: voorspellingen doen.

Neuraal netwerk

Hoe werkt een neuraal netwerk?

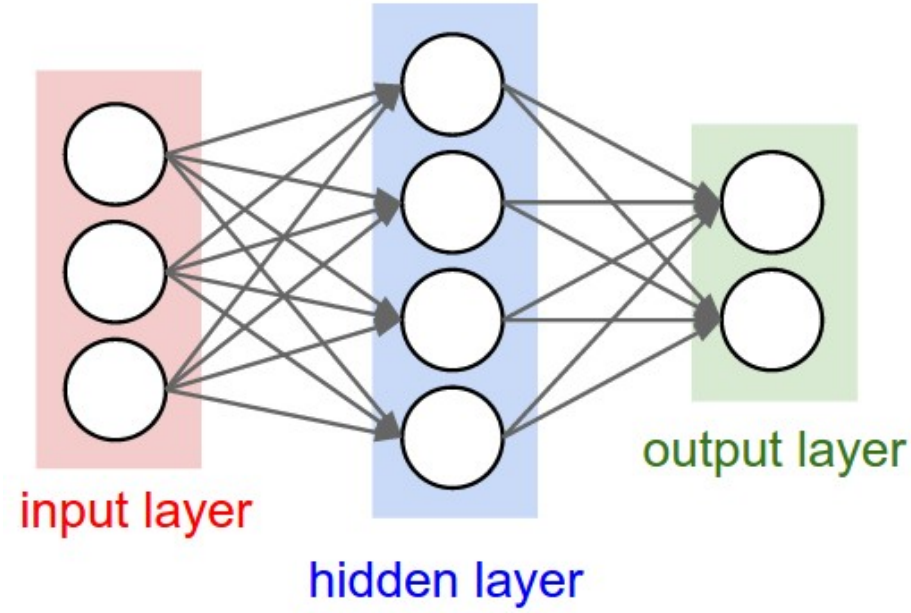


Verliesfunctie (*loss function*): berekent verschil tussen gemodelleerde en 'correcte' oplossing $\rightarrow$ zo klein mogelijk.

Neuraal netwerk

Lagen

- Invoer-laag (input layer)
- Uitvoer-laag (output layer)
- Eventueel één of meer verborgen lagen (hidden layers)



Neuraal netwerk

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

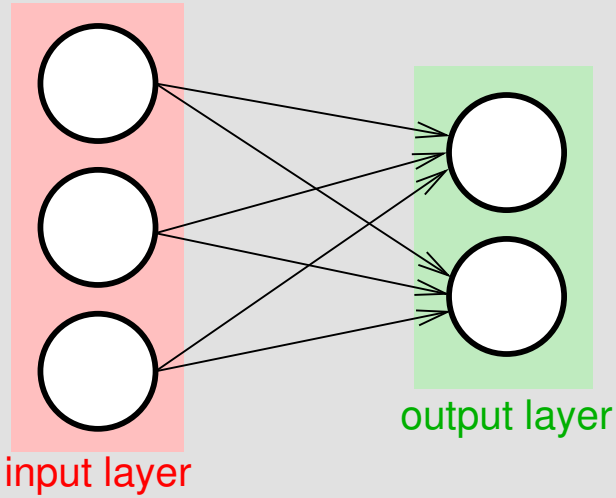
Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Lagen

- Eenvoudig geval: alleen invoer- en uitvoerlaag.



- Te schrijven als een matrixvermenigvuldiging:

$$\begin{pmatrix} \cdot \\ \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

Lineaire regressie

Lineaire regressie

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Lineaire regressie met een neuraal netwerk

- *Nodes* verbonden door *links*.
- Iedere *link* heeft een *gewicht*.
- De *inference*-functie rekent het netwerk door.
- De *loss*-functie bepaalt hoe goed het model klopt.
- *Squared error loss*: het kwadraat van het verschil tussen model en goede antwoord.
- Door middel van *gradient descent* worden de gewichten aangepast.
- Extra input: bias.

Voorbeeld: linear_regression.py

Lineaire regressie

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

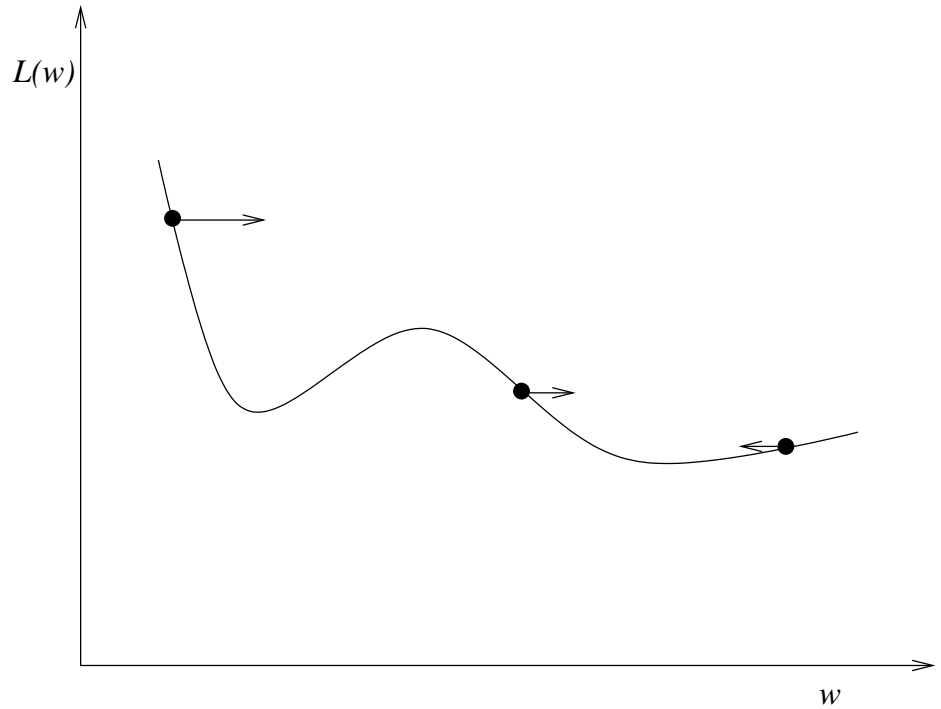
Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Gradient descent

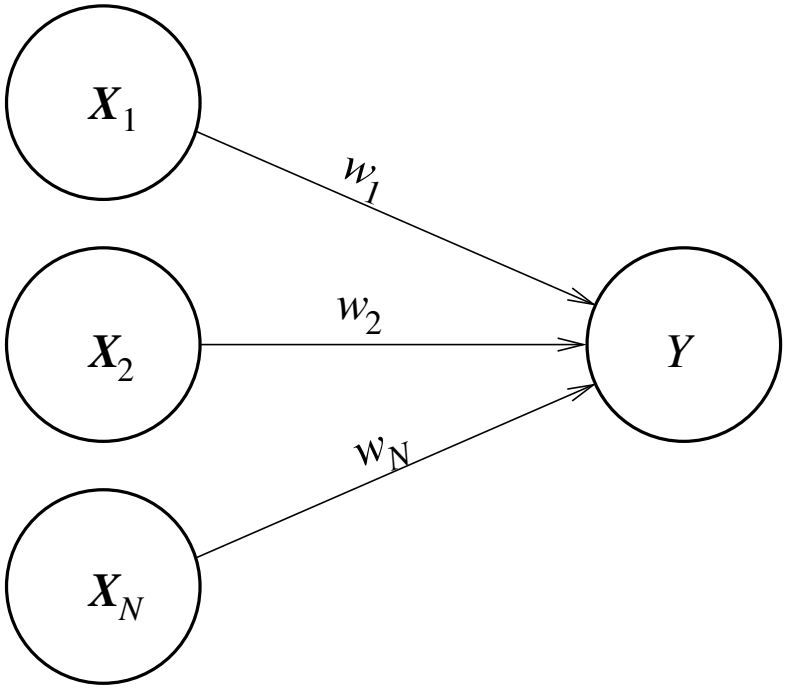
- Neem de afgeleide van de loss functie naar alle gewichten.
- Pas het gewicht aan met de afgeleide maal de *learning rate* (in tegengestelde richting).



Lineaire regressie

Meerdimensionele lineaire regressie

- Meerdere inputs.
- Netwerk leert wat belangrijk is en wat niet.



Voorbeeld: multiple_linear_regression.py

Aan de slag: slimme thermostaat

Aan de slag: slimme thermostaat

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

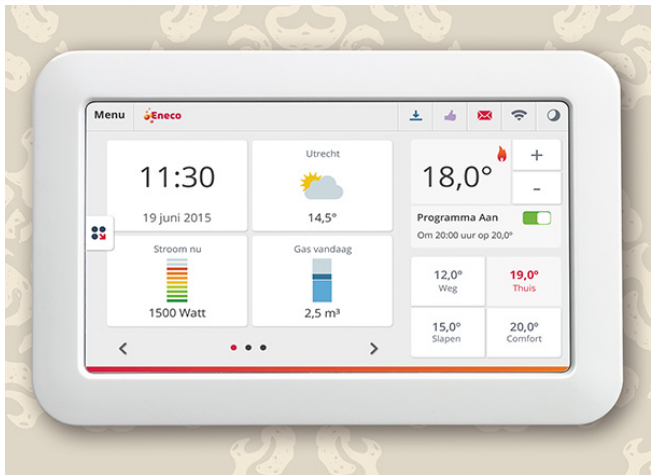
Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat

Probleemstelling

- Een gebruiker stelt de thermostaat in op basis van
  - tijd van de dag
  - gedragen kleding → buitentemperatuur
  - gevoelstemperatuur → luchtvochtigheid
  - ...
- Deze gegevens kunnen we *meten*.
- Kan de thermostaat de gewenste instelling *leren*?



Aan de slag: slimme thermostaat

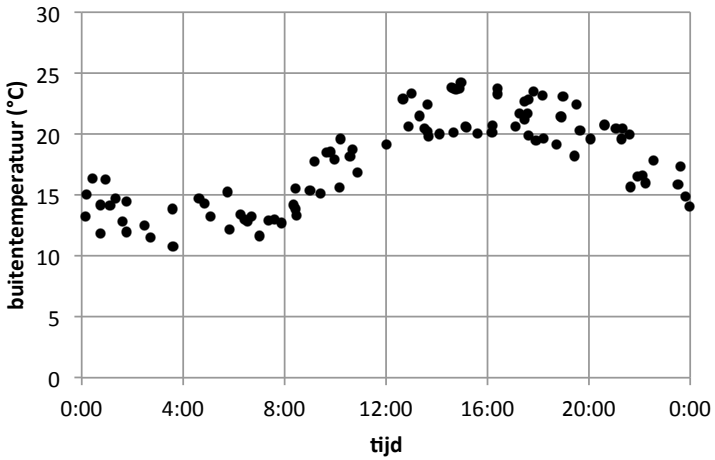
Minor Smart Devices Workshop Machine Learning

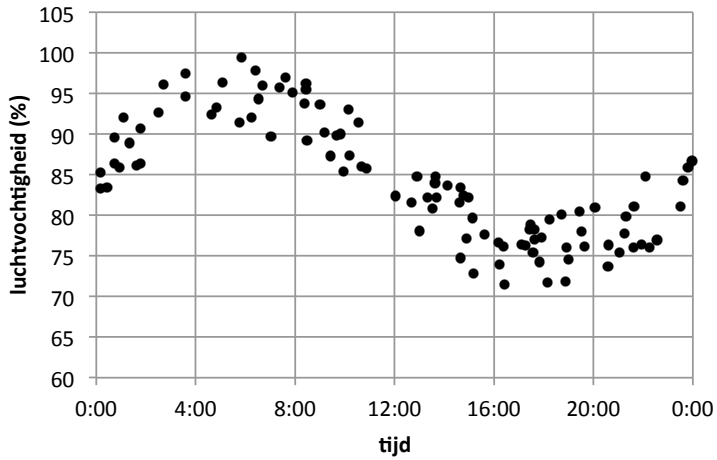
dr. Wouter Bergmann Tiest

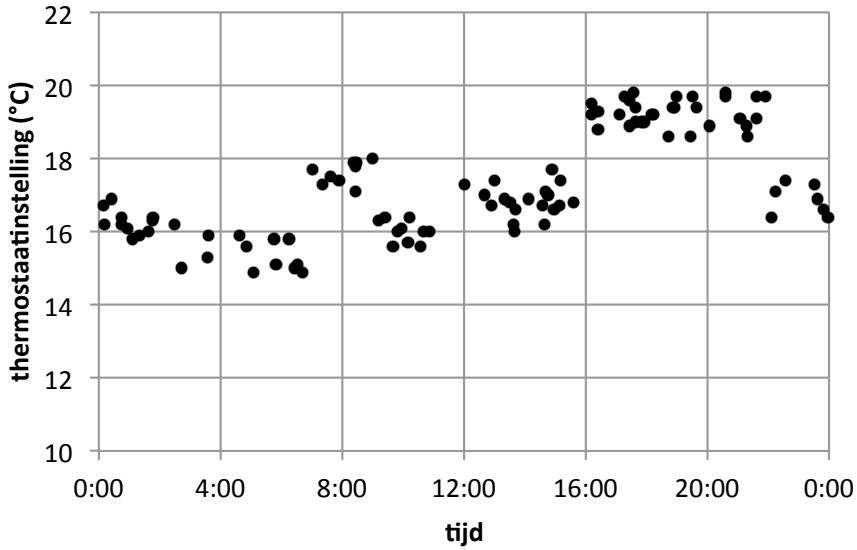
Neuraal netwerk

Lineaire regressie

Aan de slag: slimme thermostaat







Aan de slag: slimme thermostaat

Minor Smart Devices Workshop Machine Learning

dr. Wouter Bergmann Tiest

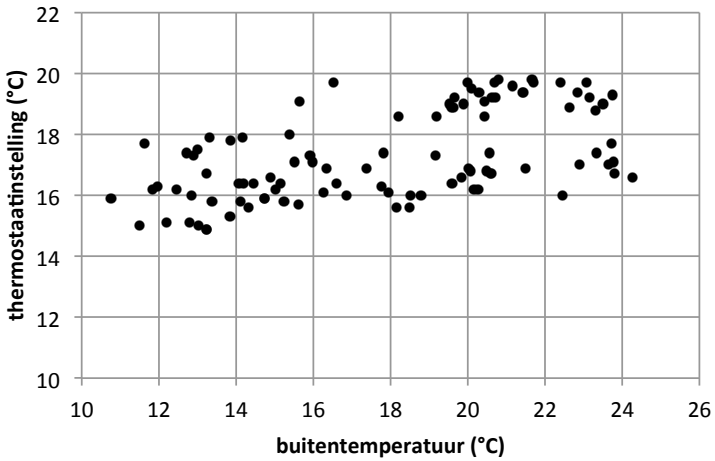
Neuraal netwerk

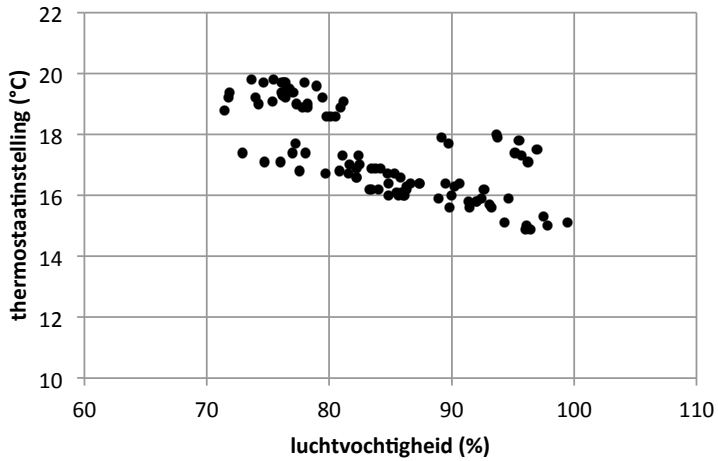
Lineaire regressie

Aan de slag: slimme thermostaat

Beschikbare data

- tijd (hh:mm:ss)
- buitentemperatuur (°C)
- luchtvochtigheid (%)
- thermostaatinstelling (°C)





Aan de slag: slimme thermostaat

Opdracht

- Bouw een neuraal netwerk dat leert welke thermostaatinstelling gewenst is.
  - Definieer de input nodes en voeg een bias toe.
  - Lees data in uit een CSV-bestand. Converteer de hh:mm:ss timestamp naar een enkel getal.
  - Stop de data-items één voor één in de input nodes.
  - Bereken de gemiddelde squared error loss.
- Train je netwerk met de data uit het bestand `thermostaat.csv`.
- Evalueer de kwaliteit met een paar testcases.