

# Maskinlæring



Even Marius Nordhagen

Universitet i Oslo

*evenmn@fys.uio.no*

March 9, 2020

# Oversikt

- ▶ Motivasjon
- ▶ Teorien bak
- ▶ Implementasjon
- ▶ Dere skal implementere et nevralt nettverk



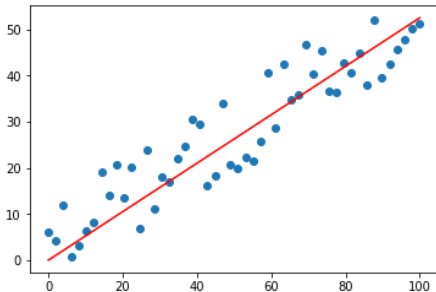
# Motivasjon



Uio : Universitetet i Oslo

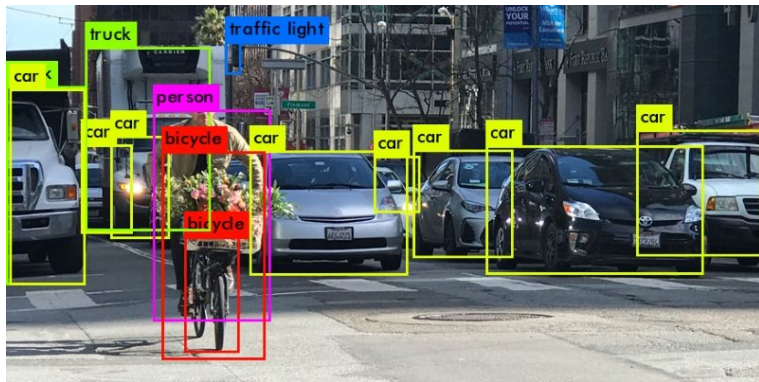
# Regresjon

Regresjon er en enkel form for maskinlæring



# Bildeanalyse

Kjenne igjen hva som er på et bilde og hvor objektene befinner seg



# Generative modeller



UiO : Universitetet i Oslo

# Andre eksempler

- ▶ Stemmegjenkjenning
- ▶ Taktikkspill
- ▶ Autonom teknologi



# Nevrale nettverk





# Hva er et nevralt nettverk?

- Inspirert av biologi og hjerneforskning



# Hva er et nevralt nettverk?

- ▶ Inspirert av biologi og hjerneforskning
- ▶ Fleksible funksjoner → mange parametere



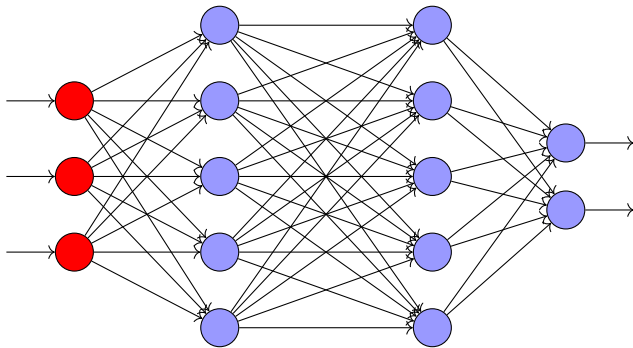
# Hva er et nevralt nettverk?

- ▶ Inspirert av biologi og hjerneforskning
- ▶ Fleksible funksjoner  $\rightarrow$  mange parametere
- ▶ Feedforward nettverk kan representere enhver funksjon



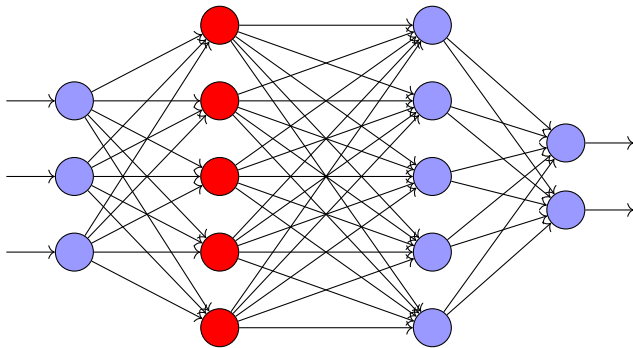
# Nevralt nettverk: Feedforward

Fase: Fremover



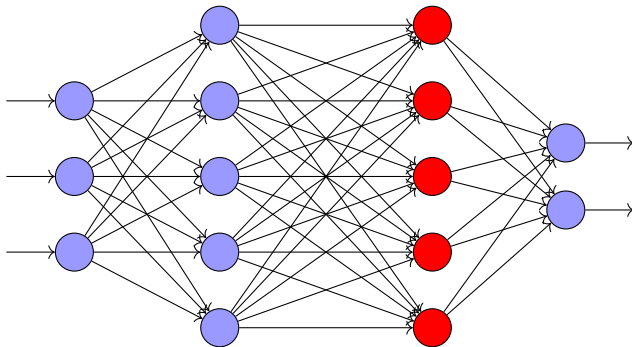
# Nevralt nettverk: Feedforward

Fase: Fremover



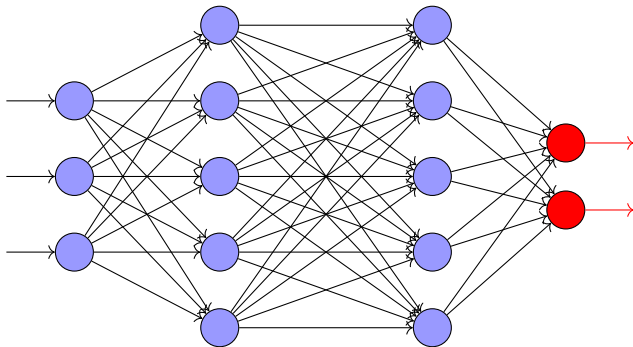
# Nevralt nettverk: Feedforward

Fase: Fremover



# Nevralt nettverk: Feedforward

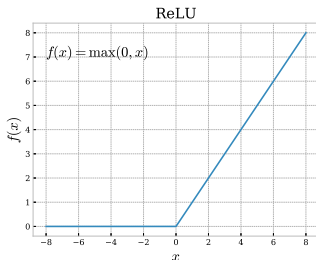
Fase: Fremover



# Nevralt nettverk: Feedforward

## Aktiveringsfunksjon

En aktiveringsfunksjon brukes for å *aktivere* inputen til hvert lag. Rectified Linear Unit (ReLU) er en vanlig funksjon:





# Nevralt nettverk: Feedforward

## Feilestimat

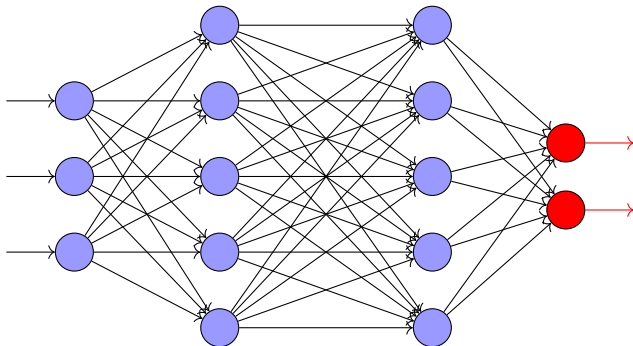
En mye brukt kostfunksjon er minste kvadraters metode:

$$\mathcal{C} = \frac{1}{2} (y - \tilde{y})^2$$



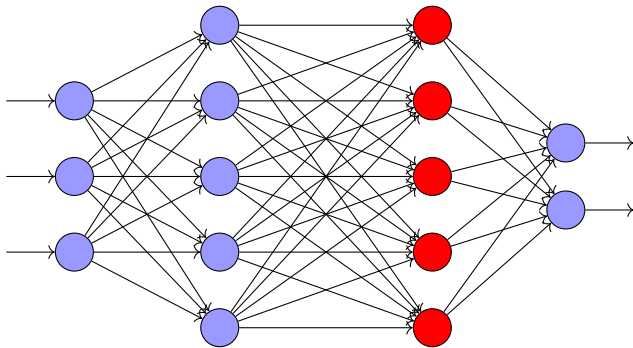
# Nevralt nettverk: Feedforward

Fase: Bakover



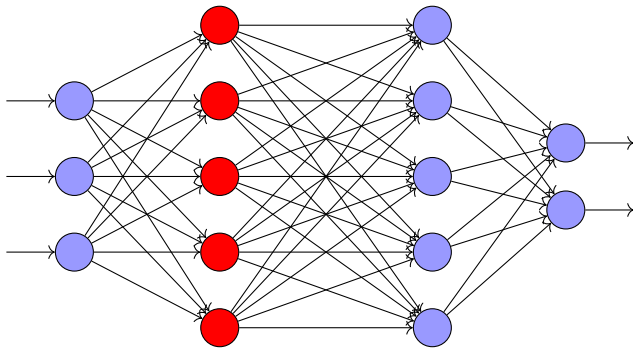
# Nevralt nettverk: Feedforward

Fase: Bakover



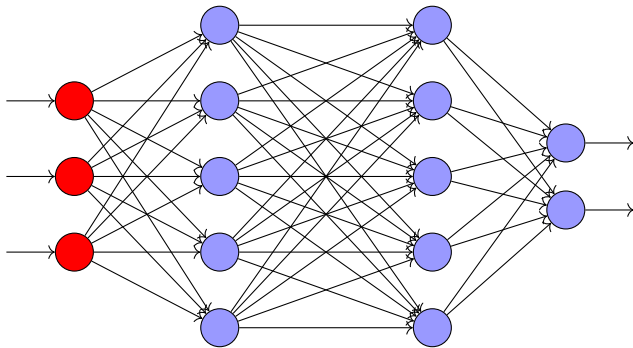
# Nevralt nettverk: Feedforward

Fase: Bakover



# Nevralt nettverk: Feedforward

Fase: Bakover



# Nevralt nettverk for snake



# Treningsdata for nettverket

Nettverket trenger informasjon for å avgjøre hvilken vei slangen skal bevege seg. Det vi velger å sende inn er:



# Treningsdata for nettverket

Nettverket trenger informasjon for å avgjøre hvilken vei slangen skal bevege seg. Det vi velger å sende inn er:

- ▶ I hvilken retning er maten?
- ▶ Hva befinner seg foran slangen?
- ▶ Hvilken vei valgte slangen gå i den situasjonen?
- ▶ En evaluering av valget



# Treningsdata for nettverket

## Evaluering

Vi implementerer noen veldig enkle regler for å evaluere en avgjørelse:

- ▶ Dårlig (-1)
- ▶ Middels (0)
- ▶ Bra (+1)

Det som kjennetegner en dårlig avgjørelse er at slangen kræsjer. Det som kjennetegner en god avgjørelse, er at den nærmer seg maten. Alt annet er middels.

# Nevralt nettverk i Pytorch

## Installasjon

Ved hjelp av Anaconda:

```
conda install pytorch torchvision -c pytorch
```

Oversikt over installasjonsmetoder: <https://pytorch.org/>

# Nevralt nettverk i Pytorch

## Bestemme arkitektur

I Pytorch kan man lage en liste med moduler som spesifiserer arkitekturen til det nevrale nettverket.

Start med å importere torch:

```
import torch  
import torch.nn as nn
```

Definer skjult lag med 25 noder og 5 noder til venstre:

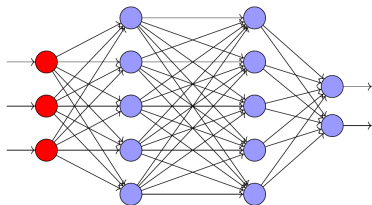
```
modul1 = nn.Linear(5, 25)
```

Definer aktiveringsfunksjon (ReLU):

```
modul2 = nn.ReLU()
```

# Nevralt nettverk i Pytorch

## Bestemme arkitektur



```
modules = []  
modules.append(nn.Linear(3, 5))  
modules.append(nn.ReLU())  
modules.append(nn.Linear(5, 5))  
modules.append(nn.ReLU())  
modules.append(nn.Linear(5, 2))  
model = nn.Sequential(*modules)
```