

# Konstruksjon av høydimensjonale nevralt nettverk-potensialer for molekylærdynamikk

John-Anders Stende

Fysisk institutt  
Universitetet i Oslo

Masterpresentasjon, oktober 2017

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

NNP for Si

Konklusjon og fremtidig arbeid

# Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

NNP for Si

Konklusjon og fremtidig arbeid

## Hva er molekylærdynamikk?

- ▶ Numerisk metode for å simulere atomers og molekylers bevegelser i gasser, væsker og faste stoffer.
- ▶ Virtuelt eksperiment.

## Dynamikk

- ▶ Partiklenes interaksjoner styrer dynamikken.
- ▶ Interaksjonene bestemmes av et kraftfelt  $\mathbf{F}$ :

$$\mathbf{F} = -\nabla V(\mathbf{r})$$

- ▶ Potensiell energiflate / potensial):

$$V(\mathbf{r}), \quad \mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$$

- ▶  $V(\mathbf{r})$  inneholder fysikken.

## Ab initio molekylærdynamikk

Løse Schrödinger-likningen ved hvert tidssteg.

## Klassisk molekylærdynamikk

Bruke en predefinert analytisk funksjon.

## Klassisk potensial

$$V(\mathbf{r}) \approx \sum_i^N V_1(\mathbf{r}_i) + \sum_{i,j}^N V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k}^N V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots$$

1. Hvor mange ledd bør tas med?
2. Hvordan bør leddene se ut?

Eksperiementer / kvantemekanikk

## Empirisk potensial:

1. Starte med en funksjonsform med noen parametre.
2. Bestemme parametre fra eksperimentelle data.

## Kvantemekanisk potensial:

1. Produsere kvantemekaniske data.
2. Tilpasse en generell funksjonsform til datasettet.
  - ▶ Fordeler: Ab initio nøyaktighet, ingen bias, overførbart.
  - ▶ Ulemper: Rent matematisk uttrykk, all relevant data må inkluderes.



## Interpolere datasett

- ▶ Spliner
- ▶ Minste kvadraters metode
- ▶ Kunstige nevrle nettverk

# Table of Contents

Molekylærdynamikk

**Nevrale nettverk**

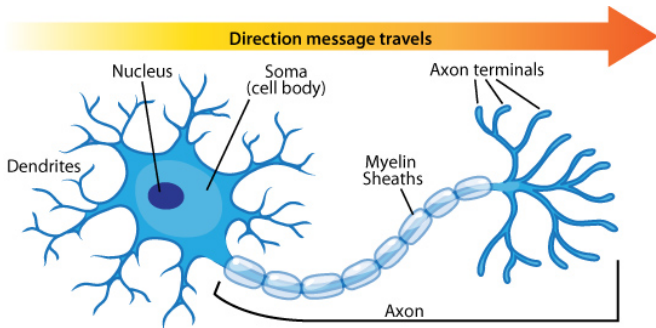
Nevralt nettverk-potensial

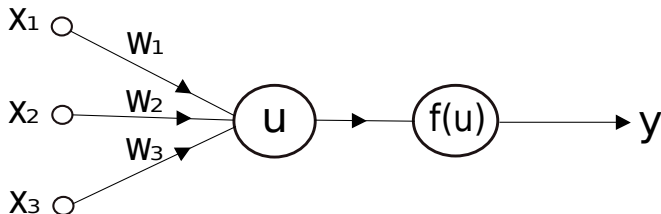
NNP for Si

Konklusjon og fremtidig arbeid

## Kunstige nevrale nettverk

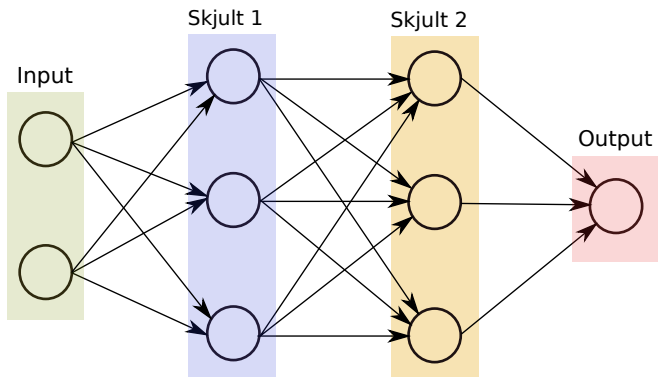
- ▶ Maskinlæringsalgoritme.
- ▶ Etterlikner biologiske nevrale nettverk.
- ▶ Kunstige nevroner sender signaler i form av matematiske funksjoner.





$$y = f \left( \sum_{i=1}^n w_i x_i + b_i \right) = f(u)$$

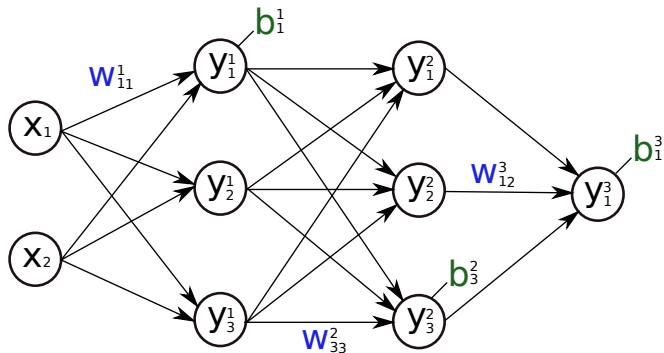
- ▶  $x_i$ : Inputsignaler.
- ▶  $w_i$ : Vekter - forsterkning/forminksning.
- ▶  $u$ : Sum - signalmiksing i soma.
- ▶  $f(u)$ : Akteriveringsfunksjon - aksjonspotensial.



## Fully-connected feed-forward nettverk.

- ▶ Hvert nevron i et lag er koblet til alle nevroner i neste lag.
- ▶ Informasjon propageres kun fremover fra input til output.

## Detaljert beskrivelse



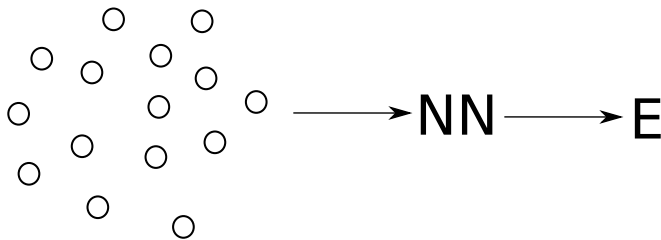
- ▶ Hver forbindelse/pil har en vekt  $w$ .
- ▶ Hver node har en bias  $b$ .
- ▶ Inneholder 25 parametre  $\{w, b\}$

# Analytisk uttrykk

$$\begin{aligned} y_1^3 &= f_3 \left[ \sum_{j=1}^3 w_{1j}^3 f_2 \left( \sum_{k=1}^3 w_{jk}^2 f_1 \left( \sum_{m=1}^2 w_{km}^1 x_m + b_k^1 \right) + b_j^2 \right) + b_1^3 \right] \\ &= f_3(x_1, x_2) \end{aligned}$$

- ▶ Mapping:  $(x_1, x_2) \in \mathbb{R}^2 \rightarrow y_1^3 \in \mathbb{R}$ .
- ▶ Ved å justere de 25 parameterne får uttrykket stor fleksibilitet.
- ▶ Dimensjonene til nettet (inputs og outputs) må stemme overens med funksjonen som skal tilpasses.
- ▶ Et nettverk med ett skjult lag kan approksimere enhver kontinuerlig funksjon.





## Regresjon med nevralt nettverk

- ▶ Mål: Interpolere datasett av konfigurasjoner  $X$  og energier  $Y$  slik at  $NN : X \in \mathbb{R}^n \rightarrow Y \in \mathbb{R}$ .
- ▶ Referansenergiene  $Y$  regnes ut kvantemekanisk.
- ▶ Trening: Iterativt justere parameterne slik at nettets output matcher referanseenergiene.
- ▶ Én enkelt konfigurasjon kalles et *treningseksempel*, og datasettet  $(X, Y)$  kalles et *treningssett*.

Feilen til nettet defineres ved en cost-funksjon:

Gjennomsnittlig kvadratisk feil:

$$\Gamma = \frac{1}{2N} \sum_{i=1}^N (Y_i - y_i)^2$$

Cost-funksjonen minimeres med gradient descent:

$$\theta_{k+1} = \theta_k - \gamma \nabla_{\theta_k} \Gamma(\theta)$$

- ▶  $\theta_k$ : Vektor av parametre ved iterasjon  $k$ .
- ▶  $\gamma$ : Steglengde/læringsrate.

# Hvordan finne gradienten av et nevralt nettverk?

$$y_1^3 = f_3 \left[ \sum_{j=1}^3 w_{1j}^3 f_2 \left( \sum_{k=1}^3 w_{jk}^2 f_1 \left( \sum_{m=1}^2 w_{km}^1 x_m + b_k^1 \right) + b_j^2 \right) + b_1^3 \right]$$

## Backpropagation

- ▶ Verdien av cost-funksjonen (feilen) propageres *bakover* fra output til input.
- ▶ Alle deriverte innhentes ved å propagere feilen én gang.
- ▶ En anvendelse av kjerneregelen.

# Table of Contents

Molekylærdynamikk

Nevrale nettverk

**Nevralt nettverk-potensial**

NNP for Si

Konklusjon og fremtidig arbeid

## Hvor mange nettverk?

- ▶ Ett nettverk: Uhåndterlig størrelse.
- ▶ Atomære nettverk,

$$E = \sum_{i=1}^N E_i$$

- ▶ Behler-Parrinello: Hver atomenergi  $E_i$  avhenger kun av naboatomer innenfor en cutoff-radius  $r_c$ .

## Høydimensjonalt potensial - utfordringer

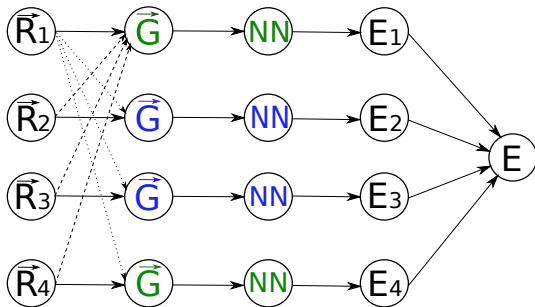
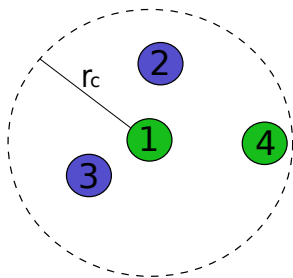
1. Varierende antall naboer.
2. Translasjonell og rotasjonell invarians.
3. Rekkefølgen på naboer.

## Behler-Parrinello-metoden: Symmetrifunksjoner

1. Et sett av såkalte symmetrifunksjoner transformerer de kartesiske koordinatene til alle naboer til en vektor av symmetriverdier.
2. Symmetrifunksjonene er atomsentrerte.
3. Symmetrifunksjonene er definert som summer over naboer.

# Behler-Parrinello eksempel

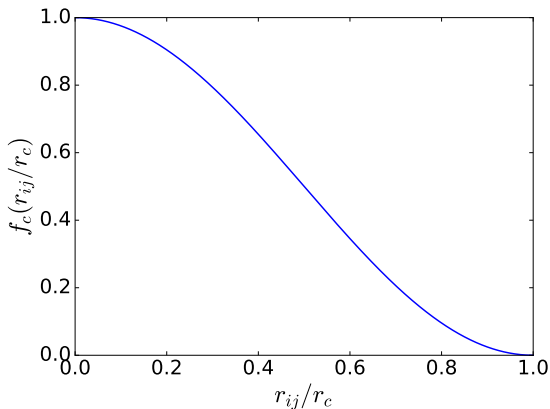
- ▶ Symmetrifunksjone beskriver hvert atoms kjemiske omgivelser.
- ▶ Hvert atom har både et eget nettverk og et eget sett av symmetrifunksjoner.
- ▶ Alle atomer av samme kjemiske element har identiske nettverk og symmetrifunksjonssett.



## Cutoff-funksjon

Monotonisk minkende del av en cosinus-funksjon.

$$f_c(r_{ij}) = \begin{cases} 0.5[\cos(\pi r_{ij}/r_c) + 1], & r_{ij} \leq r_c \\ 0, & r_{ij} > r_c \end{cases}$$

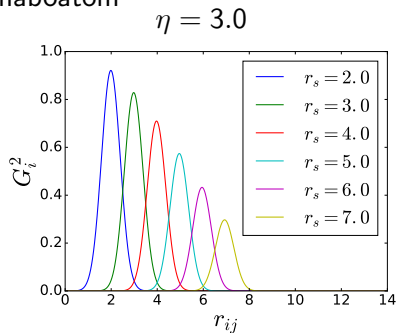
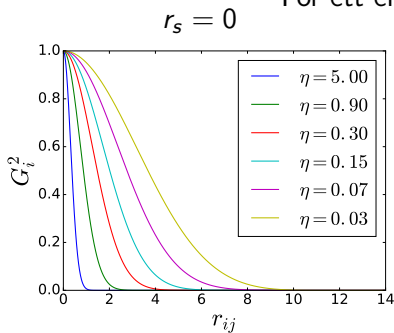




# Radiell symmetrifunksjon

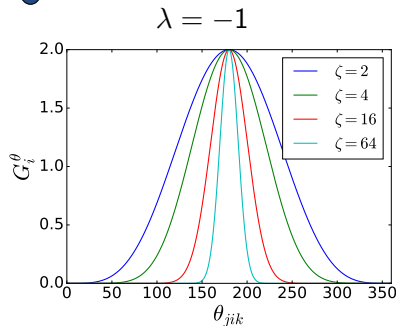
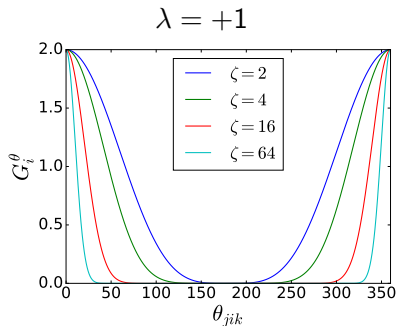
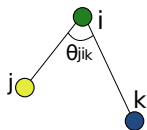
$$G_i^2 = \sum_{j=1}^N \exp[-\eta(r_{ij} - r_s)^2] f_c(r_{ij})$$

For ett enkelt naboatom



# Angulær symmetrifunksjon

$$G_i^5 = 2^{1-\zeta} \sum_{j \neq i} \sum_{k > j} \left[ (1 + \lambda \cos \theta_{jik})^\zeta \exp[-\eta(r_{ij}^2 + r_{ik}^2)] f_c(r_{ij}) f_c(r_{ik}) \right]$$



## Krefter for et grunnstoff

$$\mathbf{F}_i = -\nabla_i E$$

$$F_{i,x} = -\frac{\partial E}{\partial x_i} = -\sum_{j=1}^{N_i+1} \frac{\partial E_j}{\partial x_i} = -\sum_{j=1}^{N_i+1} \sum_{s=1}^M \frac{\partial E_j}{\partial G_s} \frac{\partial G_s}{\partial x_i}$$

$$E_j = \text{NN}[\mathbf{G}(\mathbf{r}_{jk})]$$

# Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

**NNP for Si**

Konklusjon og fremtidig arbeid

# Simuleringspakker

## LAMMPS

- ▶ Pakke for klassisk molekylærdynamikk utviklet ved Sandia National Laboratories.
- ▶ Kjøres ved inputscripts med egen syntaks.
- ▶ Vi har utvidet med samplingsalgoritme og nevral nettverk-potensial.

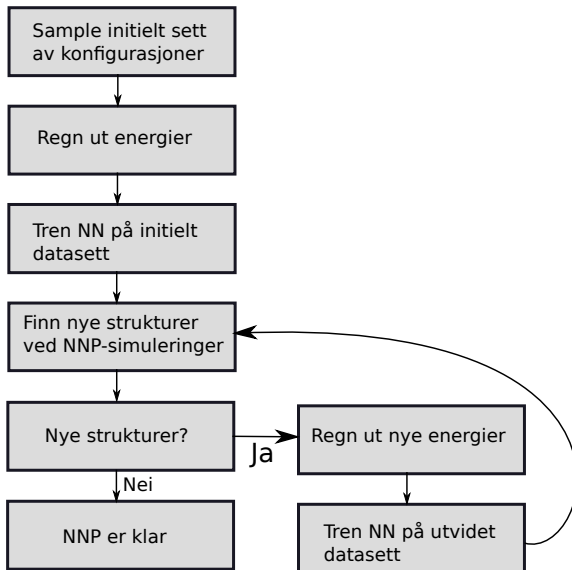
## TensorFlow

- ▶ Maskinlæringspakke utviklet av Google.
- ▶ Vi har brukt TensorFlow Python API til å utvikle en modell for regresjon med nevrale nettverk.

## Konstruksjon av et nevralt nettverk-potensial (NNP)

1. Generere treningsdata som er relevant for anvendelsen av potensialet, dvs. identifisere konfigurasjonsrommet.
2. Trene et nevralt nettverk for å tilpasse en funksjon til dataene.
3. Bruke det trente nettverket som et analytisk potensial i molekylærdynamikksimuleringer.

# Sample konfigurasjonsrommet vha. MD-simuleringer



# Konstruere et potensial for Si

## 1. Initiell sampling

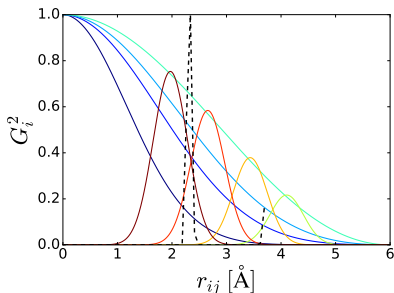
- ▶ Reprodusere Stillinger-Weber.
- ▶  $T \in [0, 500]$  K.
- ▶ Konfigurasjoner og energies samples med vår samplingsalgoritme.



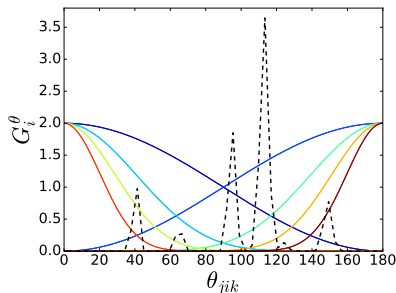
## 2. Foreløpig potensial

### Initielt symmetrifunksjonssett

#### Radielle



#### Angulære



- ▶ Bør dekke konfigurasjonsrommet på en noenlunde uniform måte.
- ▶ Unngå symmetrifunksjoner som er null for alle konfigurasjoner.

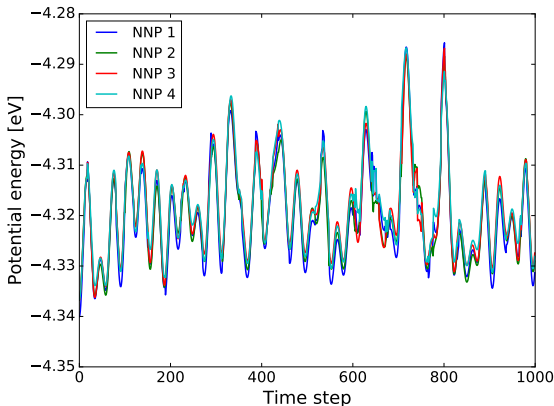
Trener et nevralt nettverk for å konstruere et foreløpig potensial.

### 3. Iterativ sampling

#### Ekstrapolerende konfigurasjoner

Bruke max og min for hver symmetrifunksjon anvendt under trening.

#### Interpolerende konfigurasjoner - multiple-NN-metoden



## 4. Tilpasse endelig treningssett

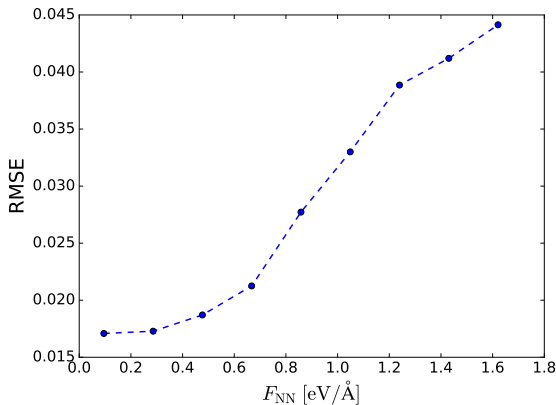
### Hyperparametre

- ▶ Parametre som ikke optimeres automatisk.
- ▶ Eksempler:
  - ▶ Antall lag.
  - ▶ Antall noder i hvert lag.
  - ▶ Antall treningsiterasjoner.
  - ▶ Læringsrate.
- ▶ Bestemmes fra kvalitative argumenter eller ved en form for gridsøk.

Tilpasse endelig treningssett

RMSE: 0.864 meV.

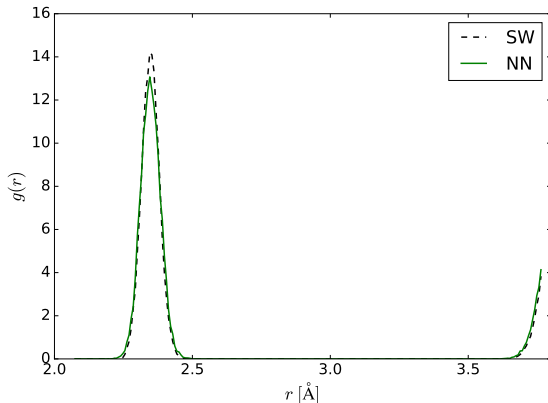
RMSE krefter



RMSE: 41.2 meV

## Radiell distribusjonsfunksjon $g(r)$

Sammenlikner tidsmidlet SW og NN.



## Mekaniske egenskaper

	NNP	Analytic SW	Relative error
Bulk modulus	103.0	101.4	1.58 %
Shear modulus	53.6	56.4	5.22 %
Poisson ratio	0.348	0.335	3.88 %

# Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

NNP for Si

Konklusjon og fremtidig arbeid

## Det ideelle potensial

- ▶ Potensialet bør være nøyaktig.
- ▶ Burde finnes måter å systematisk forbedre potencialet.
- ▶ Potensialet bør være generelt og anvendbart på alle typer systemer.
- ▶ Potensialet bør kunne modellere faseoverganger.
- ▶ Potensialet bør være høydimensjonalt, dvs. avhenge av alle frihetsgrader.



## Det ideelle potensial fortsetter

- ▶ Konstruksjonen av potensialet bør være så automatisert som mulig.
- ▶ Potensialet bør være prediktivt.
- ▶ Potensialet bør være raskt å evaluere.
- ▶ Konstruksjonen bør ikke ta for mye tid.
- ▶ Analytisk derivert bør være tilgjengelig.

## Fremtidig arbeid

- ▶ *Ab initio* data.
- ▶ Mer nøyaktige krefter.
- ▶ Andre systemer.
- ▶ Optimering.