University of Stuttgart
Germany

Constructive Incremental Learning From Only Local Information

Receptive Field Weighted Regression

Peter
Muschick

Agenda

- Algorithms
  - Receptive Fields Weighted Regression (RFWR)
  - Locally Weighted Projection Regression (LWPR)
- Implementation
- Live Demo

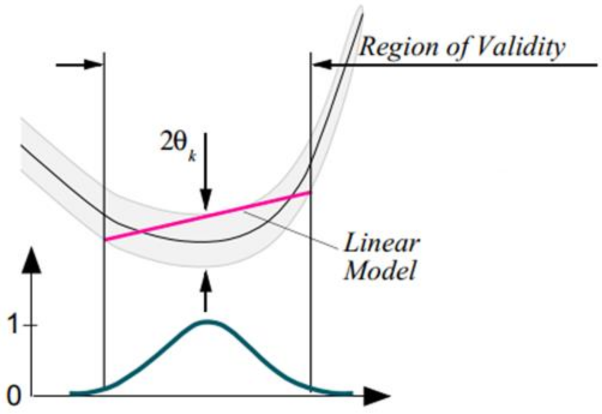
Algorithm

Receptive Fields Weighted Regression (RFWR)

- Constructive Incremental Learning From Only Local Information by Stefan Schaal & Christopher Atkeson - University of Southern California, Los Angeles, 1998
- Models data by means of **spatially localized linear models**
- Learning Scenario:
  - Limited memory
- Two major challenges:
  - Amount of receptive fields (Over and Underfitting)
  - Negative interference (Forgetting of useful knowledge)

Receptive Fields Weighted Regression (RFWR)

- Receptive Fields



- Weighted Regression

$$\beta = (X^T \boldsymbol{W} X)^{-1} X^T \boldsymbol{W} Y$$

Receptive Fields Weighted Regression (RFWR)

Initialize the RFWR with no receptive field (RF);
For every new training sample (**x**,**y**):

a) **For** k=1 to #RF:
   - calculate the activation from (5)
   - update the receptive field parameters according to (13), and (18)

   end;

b) **If** no subnet was activated by more than w_{gen};
   - create a new RF with **c**=**x**, **M**=$\mathbf{M}_{def}$

   end;

c) **If** two RFs are activated more than w_{prune}:
   - erase the RF with the larger det(**D**)

   end;

d) calculate the $m=E\{wMSE\}$ and $std=E\{(wMSE\text{-}m)^2\}^{0.5}$ of all RFs;

e) **For** k=1 to #RF:

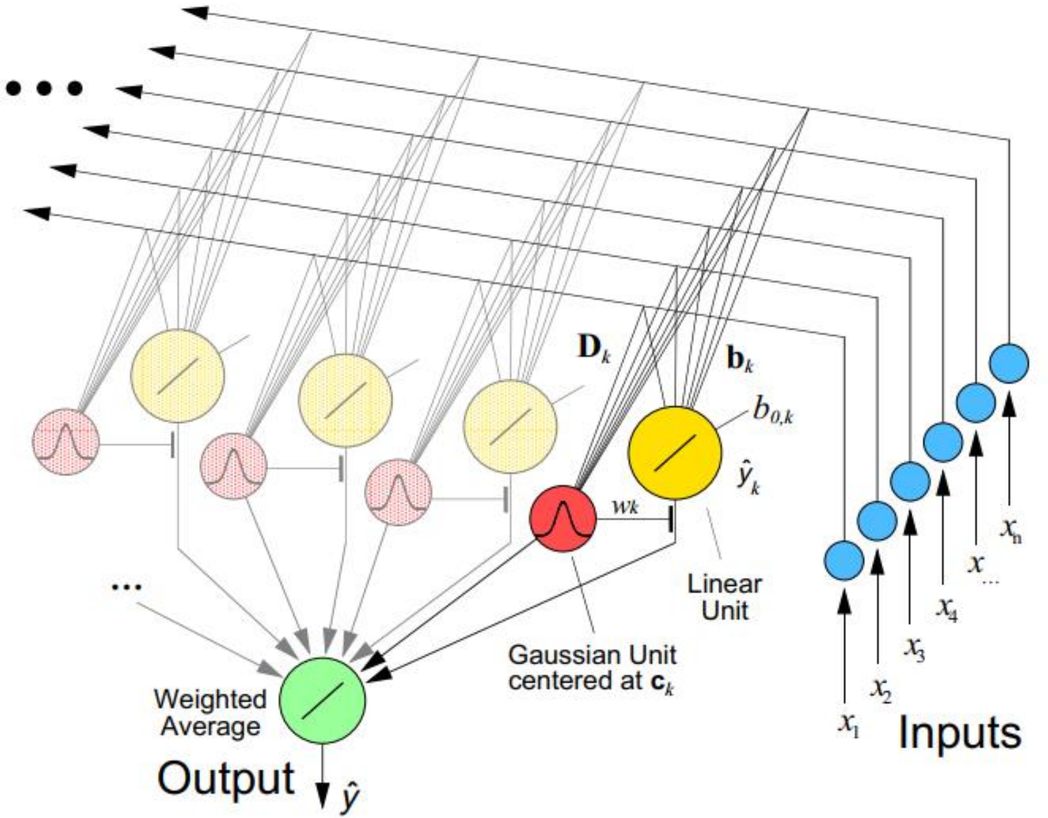
   If $|wMSE - m| > \varphi\, std$,
   - reinitialize receptive field with $\mathbf{M} = \varepsilon\ \mathbf{M}_{def}$

   end;

   end;

Receptive Fields Weighted Regression (RFWR)

- Receptive Fields
  - Input: x_n
  - Output: $\hat{y}$
  - Gaussian Unit centered: c_k
  - Gaussian Kernel: w_k



Receptive Fields Weighted Regression (RFWR)

- Gaussian Kernel: w_k
- $w_k = \exp\left(-\frac{1}{2}(\mathrm{x} - c_k)^T D_k (\mathrm{x} - c_k)\right), D_k = M_k^T M_k$

Query point

Gauss center

Positive definite distance metric

Upper triangular distance matrix

$$\begin{pmatrix} d_{x,x_{0,0}} & \cdots & d_{x,x_{0,j}} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{x,x_{i,j}} \end{pmatrix}$$

$$\mathbf{M}^{n+1} = \mathbf{M}^n - \alpha \frac{\partial J}{\partial \mathbf{M}}$$

Receptive Fields Weighted Regression (RFWR)

- Gaussian Kernel: w_k
- $w_k = \exp\left(-\frac{1}{2}(\mathrm{x} - c_k)^T D_k (\mathrm{x} - c_k)\right), D_k = M_k^T M_k$
- $W = \begin{pmatrix} w_{0,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{i,i} \end{pmatrix} \longleftarrow \boxed{\text{Diagonal weights matrix}}$

Receptive Fields Weighted Regression (RFWR)

- Gaussian Kernel: w_k
- $w_k = \exp\left(-\frac{1}{2}(\mathrm{x} - c_k)^T D_k (\mathrm{x} - c_k)\right), D_k = M_k^T M_k$
- $W = \begin{pmatrix} w_{0,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{i,i} \end{pmatrix}$
- $\beta = (X^T X)^{-1} X^T Y$ ← Linear Regression

Receptive Fields Weighted Regression (RFWR)

- Gaussian Kernel: w_k
- $w_k = \exp\left(-\frac{1}{2}(\mathrm{x} - c_k)^T D_k (\mathrm{x} - c_k)\right), D_k = M_k^T M_k$
- $W = \begin{pmatrix} w_{0,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{i,i} \end{pmatrix}$
- $\cancel{\beta = (X^T X)^{-1} X^T Y}$
- $\beta = (X^T \boldsymbol{W} X)^{-1} X^T \boldsymbol{W} Y \leftarrow \boxed{\text{Weighted Regression}}$

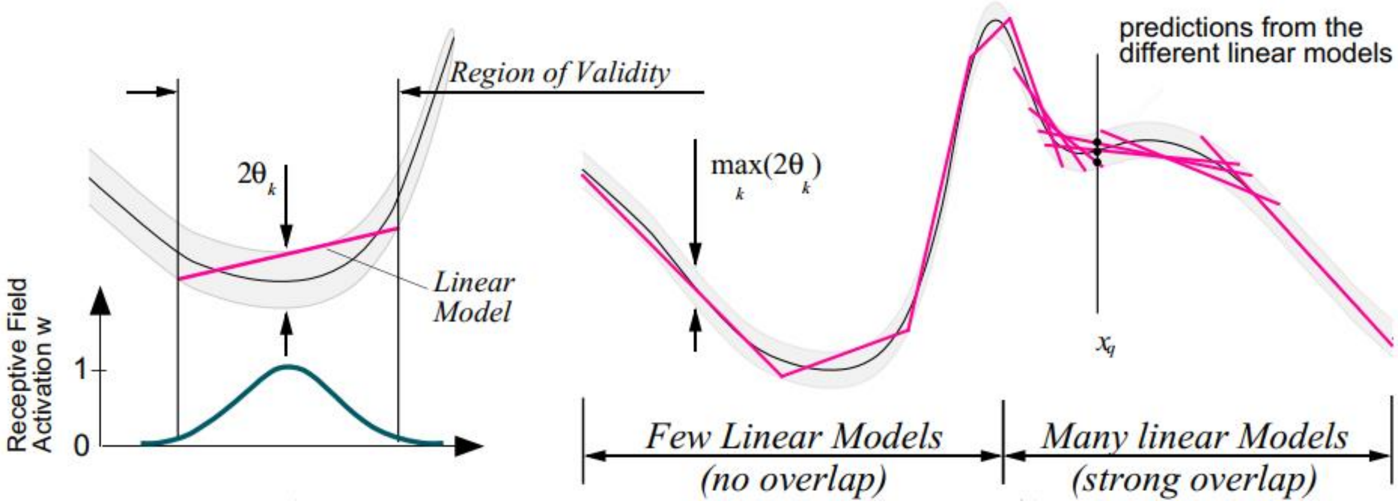
Receptive Fields Weighted Regression (RFWR)

- Gaussian Kernel: w_k
- $w_k = \exp\left(-\frac{1}{2}(\mathrm{x} - \boxed{c_k})^T D_k (\mathrm{x} - c_k)\right), D_k = M_k^T \boxed{M_k}$
- $W = \begin{pmatrix} w_{0,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{i,i} \end{pmatrix}$
- $\cancel{\beta = (X^T X)^{-1} X^T Y}$
- $\beta = (X^T \boldsymbol{W} X)^{-1} X^T \boldsymbol{W} Y$

For each receptive field *independently*, without any information about the other receptive fields

Receptive Fields Weighted Regression (RFWR)

- **Receptive Fields**
- Weighted Regression



Receptive Fields Weighted Regression (RFWR)

Adjusting Receptive Fields

- **Shape, Size,** Amount
- $J = \frac{1}{W}\sum_{i=1}^{p} w_i \|y_i - \hat{y}_i\|^2 \;\; where\ W = \sum_{i=1}^{p} w_i$
  - Minimizing J → overfitting
- $J = \frac{1}{W}\sum_{i=1}^{p} w_i \left\|y_i - \hat{y}_{i,-i}\right\|^2$
  - Minimizing J → leave-one-out cross validation (is expensive)
- $J = \frac{1}{W}\sum_{i=1}^{p} w_i \left\|y_i - \hat{y}_{i,-i}\right\|^2 = \frac{1}{W}\sum_{i=1}^{p} \frac{w_i\left\|y_i - \hat{y}_{i,-i}\right\|^2}{(1 - w_i \tilde{x}_i^T P \tilde{x}_i)^2}$
  - Minimizing J → receptive fields will shrink to a very small size
- $J = \frac{1}{W}\sum_{i=1}^{p} \frac{w_i\left\|y_i - \hat{y}_{i,-i}\right\|^2}{(1 - w_i \tilde{x}_i^T P \tilde{x}_i)^2} + \gamma \sum_{i,j=1}^{n} D_{ij}^2$

Receptive Fields Weighted Regression (RFWR)

Adjusting Receptive Fields

- Shape, Size, **Amount**
- Increase
  - A new receptive field is created if a training sample ($\mathbf{x}$,$\mathbf{y}$) does not activate any of the existing receptive field by more than a threshold
- Decrease
  - Overlaping fields, do not degrade approximation quality
  - Computational cost

Receptive Fields Weighted Regression (RFWR)

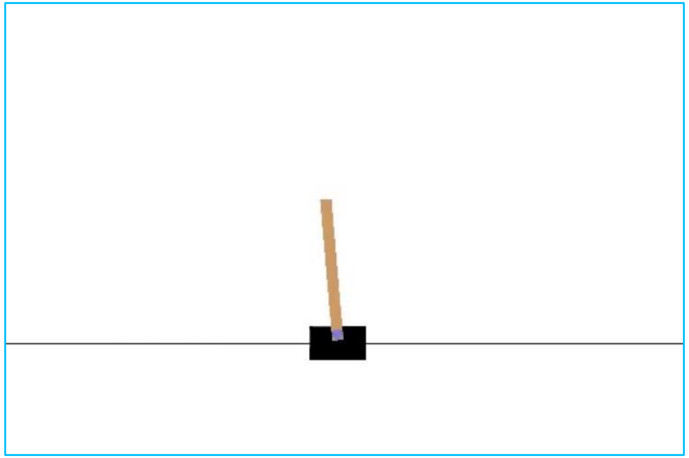
- RFWR vs. Locally Weighted Projection Regression (LWPR)
- LWPR downsizes actively the dimension in the receptive fields
  - Partial Least Squares
  - Principal Component Regression
  - Covariant Projection Regression

Implementation

- Open Gym Cartpole (python3)
- LWPR Module (python2)

Observation

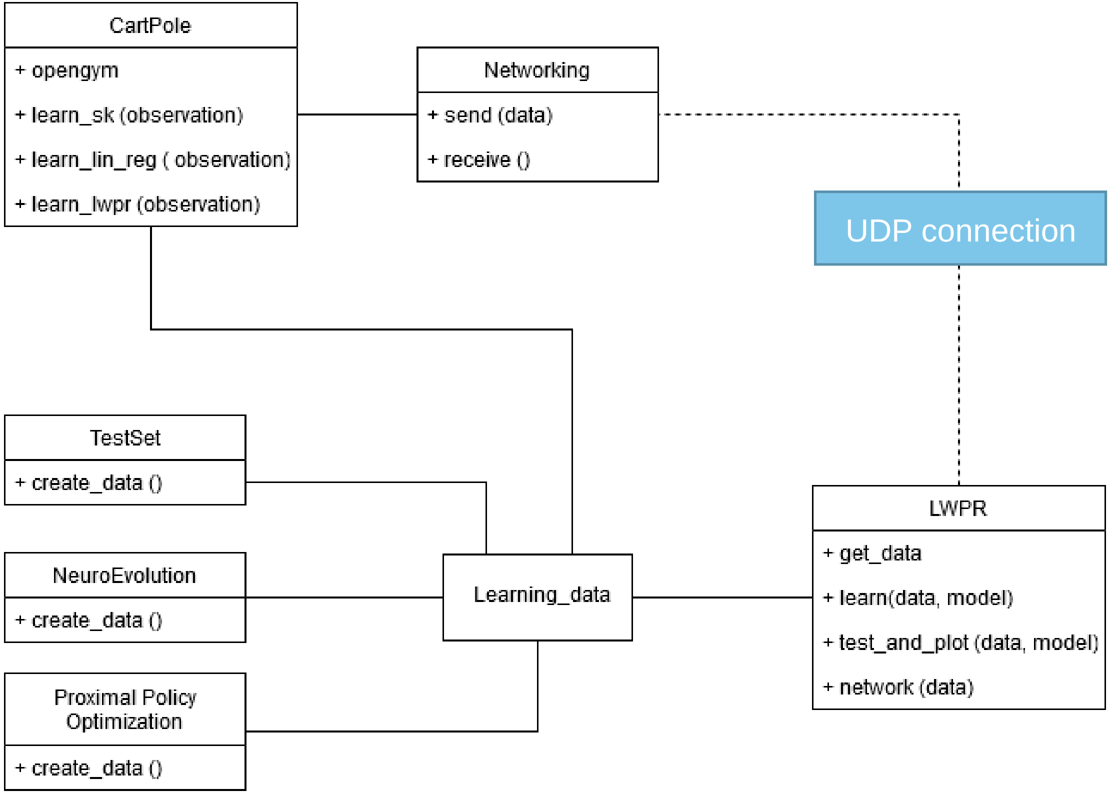
Num	Observation	Min	Max
0	Cart Position	-4.8	4.8
1	Cart Velocity	-Inf	Inf
2	Pole Angle	-24 deg	24 deg
3	Pole Velocity At Tip	-Inf	Inf



Action

Num	Action
0	Push cart to the left
1	Push cart to the right

Implementation



Jupyter notebook demo

University of Stuttgart

Thank you!

Peter Muschick

e-mail

phone +49 (0) 711 685-

www.

University of Stuttgart

Outlock

- Comparision

Implementation

- Open Gym Cartpole (python3)
- LWPR Module (python2)
  - Client-Server application via UDP
  - Generating Training data

Implementation

- Open Gym Cartpole (python3)
  - Started implementation
  - Unclear parts
  - Found matlab file
    - Very complex
  - Found module for the algorithm
    - Python2
    - Couldn’t execute it
    - Specific version of python, gcc and packages
- LWPR Module (python2)

Implementation