Neural Network Design for Behavioral Model Generation with Shape Preserving Properties

Oleg Mikulchenko and Kartikeya Mayaram

Department of ECE

Oregon State University

Corvallis, OR

Outline

- Neural Networks as Approximators
- Model Generation
- Example: MOSFET Model
- Example: Microflow Sensor Model
- Performance Improvement
- Conclusions

Modeling Approaches

- **Physics-based Detailed Models**
  - + accurate
  - – complex
  - – computationally very slow
- **Physical Compact Models**
  - + reasonable accuracy
  - – hard to derive
  - – complex parameter extraction
- **Empirical Models:**
  - + computationally very fast
  - – limited accuracy

Empirical Models

- **Polynomial and Spline Models**
  - cannot handle sharp transitions
  - false oscillations and nonmonotonic behavior
- **Table Lookup Models**
  - limited to low-dimensional problem
- **Neural Network Based Models**
  - false oscillations

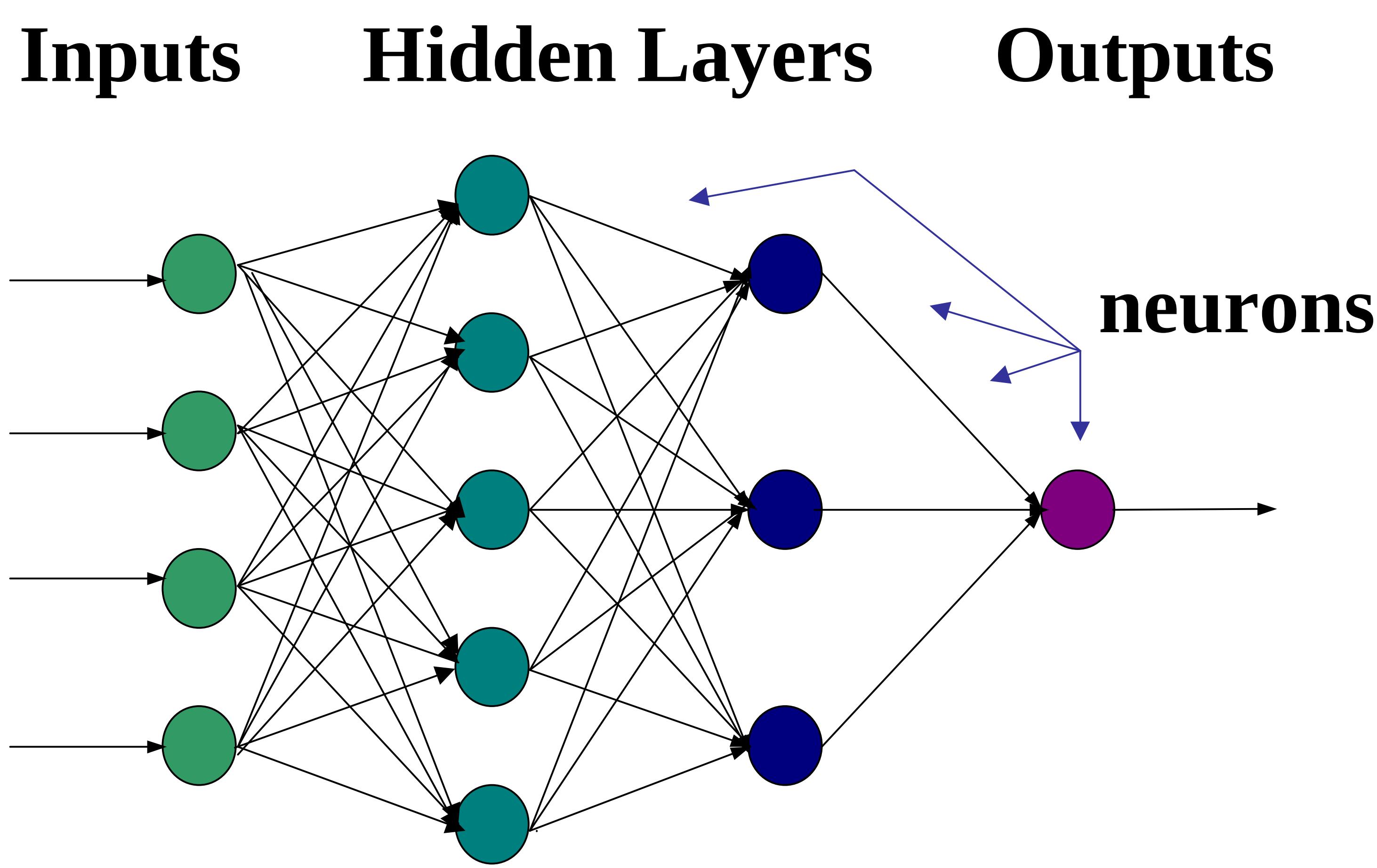
Desired Modeling Approach

- **Arbitrary Number of Variables**
- **Any Nonlinearity**
- **Arbitrary Precision of Approximation**
- **Reasonable Physical Behavior**
  - **monotonic approximation of function and derivatives**
  - **zero of data mapped to zero of model**
  - **high accuracy for wide data ranges**

Solution: Neural Networks with Shape Preserving Properties

- **Regular Structure**
  - multilayer perceptron
- **Modified Design**
  - objective function
  - design of topology
  - design of experiments

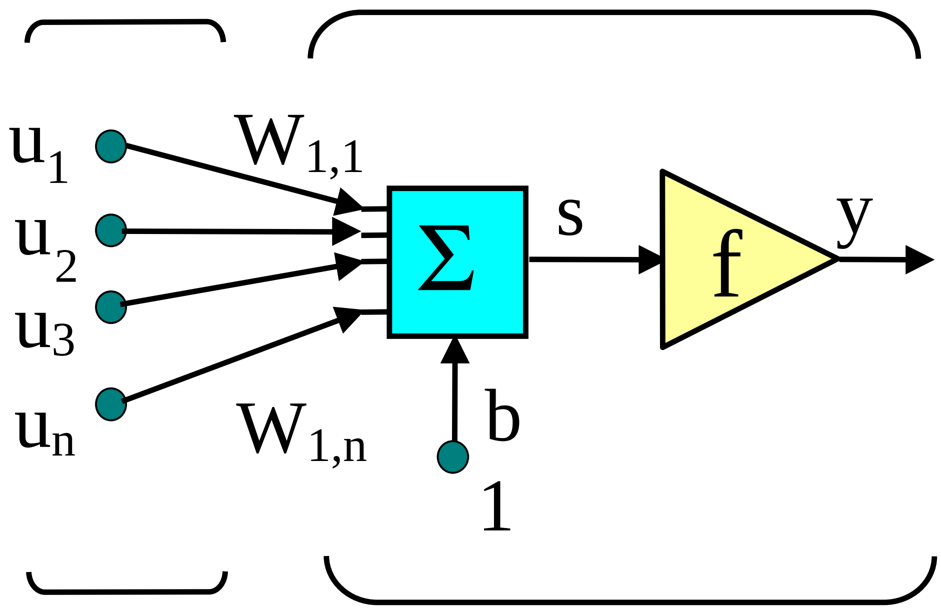
Neural Network (NN) Structure



Complex system of simple elements - neurons

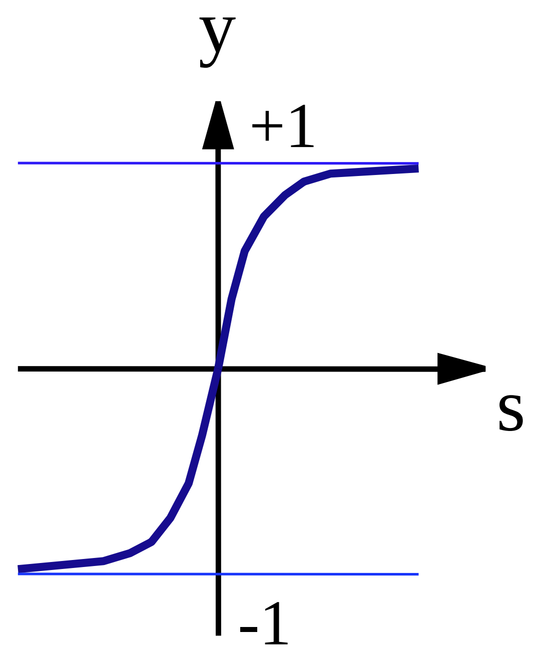
Information Processing by One Neuron

Inputs **Neuron**



$y = f\,(\mathbf{Wu}+b)$

Activation function



$y = f\,(s)$

NN Design: Objective Function

$$f(\mathbf{x},\mathbf{V}) = f_1(\mathbf{x},\mathbf{V}) * f_2(\mathbf{x},\mathbf{V})$$

$\mathbf{x}$ - vector of weights and biases, $\mathbf{V}$ – input data set

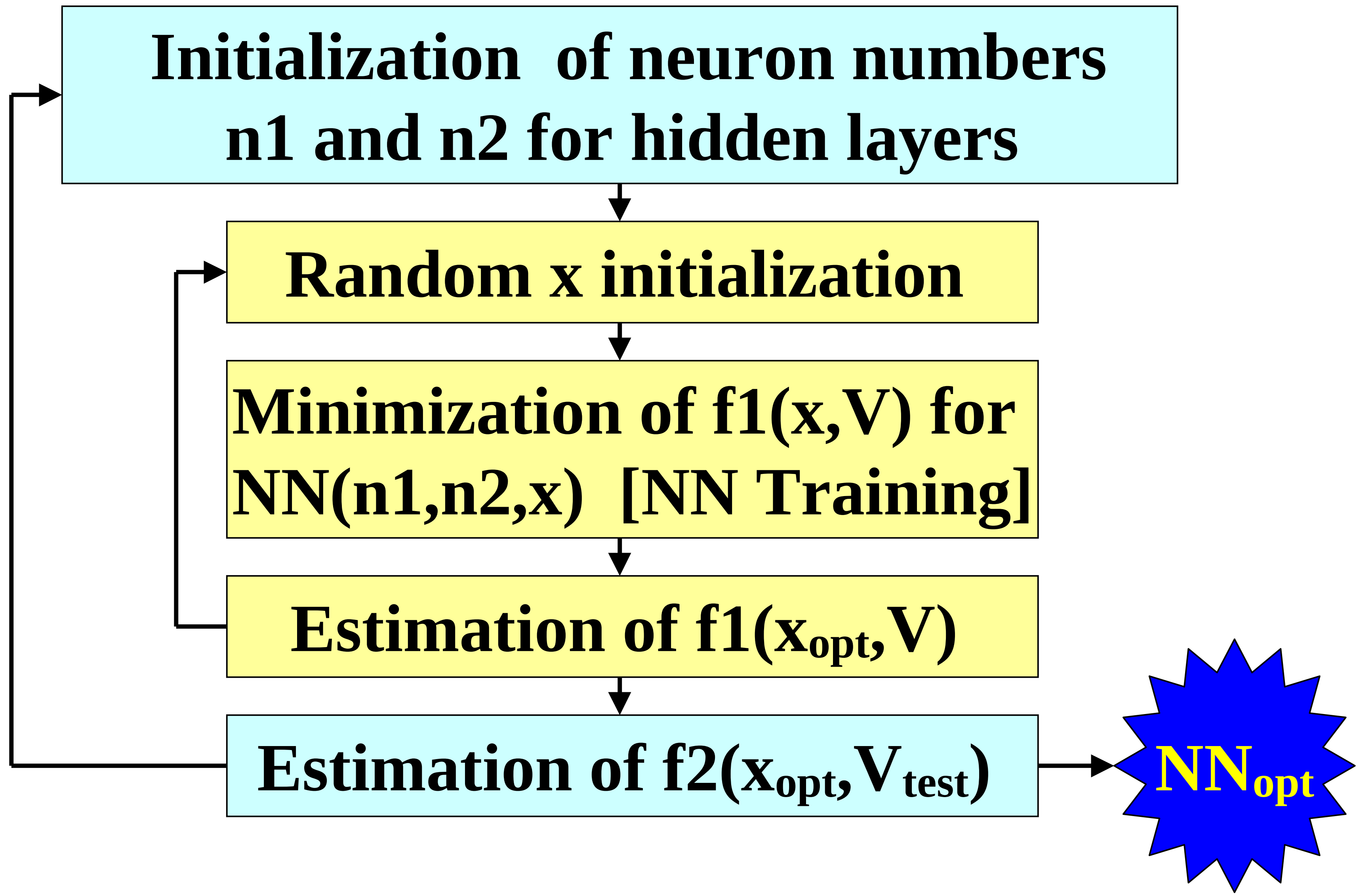
$$f_1(\mathbf{x},\mathbf{V}) = \left[\frac{1}{MN_0}\sum_{i=1}^{N_0}\sum_{j=1}^{M}\left(f_{NN,i}(\mathbf{x},\mathbf{v}_j) - y_{i,j}\right)^2\right]^{1/2}$$

$$f_2(\mathbf{x},\mathbf{V}) = \sum_{k=1}^{N_{inp}} N_{sign,k}(\mathbf{x},\mathbf{V}) + 1$$

$f_1(\mathbf{x},\mathbf{V})$ – RMSE (Root Mean Square Error)

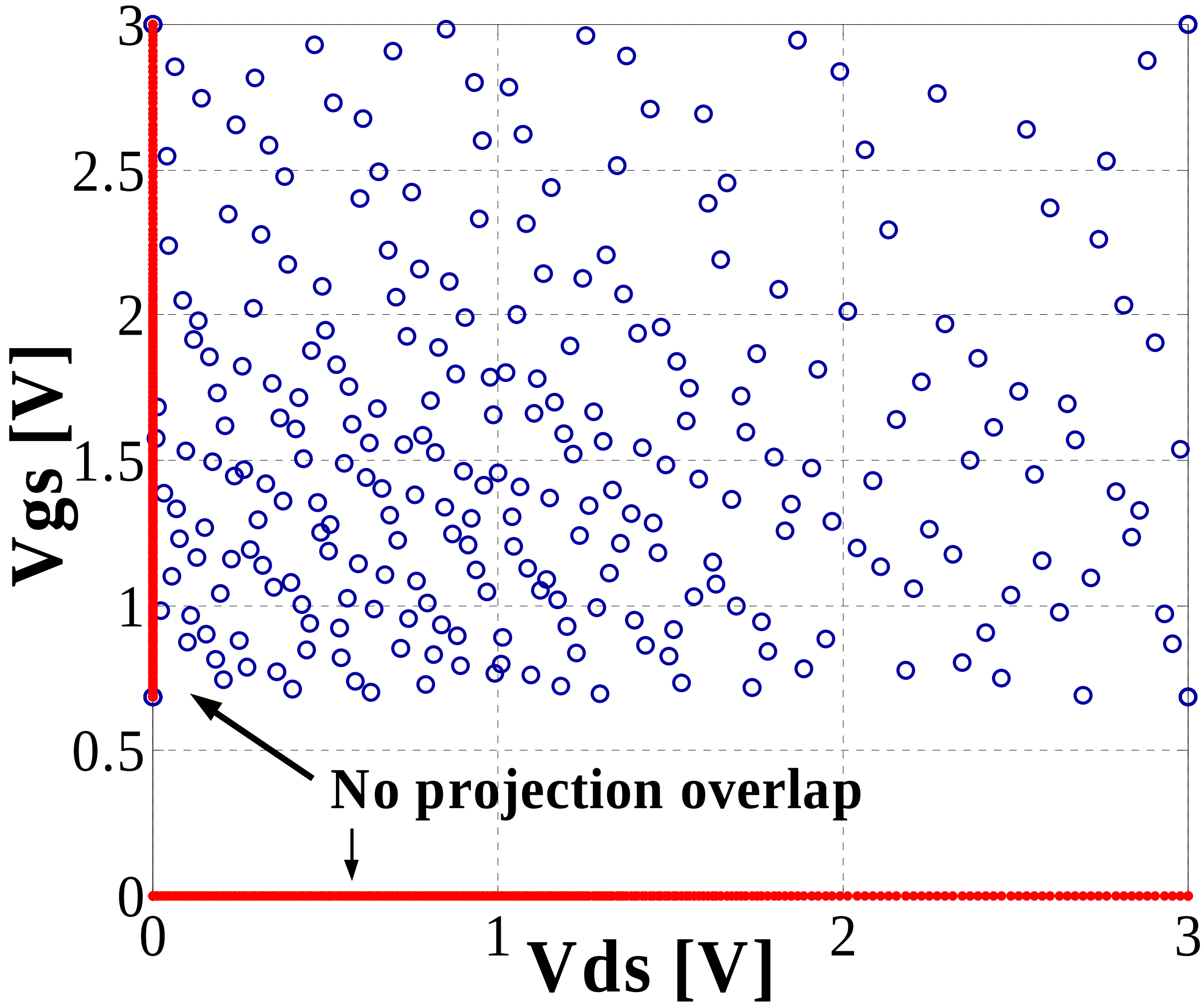
$f_2(\mathbf{x},\mathbf{V})$ – Number of oscillations in $f_{NN}(\mathbf{x},\mathbf{v})$

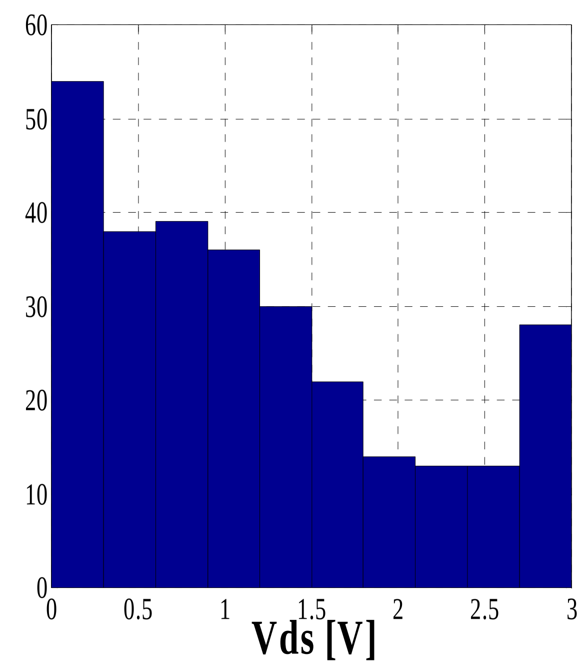
NN Design: Topology Optimization



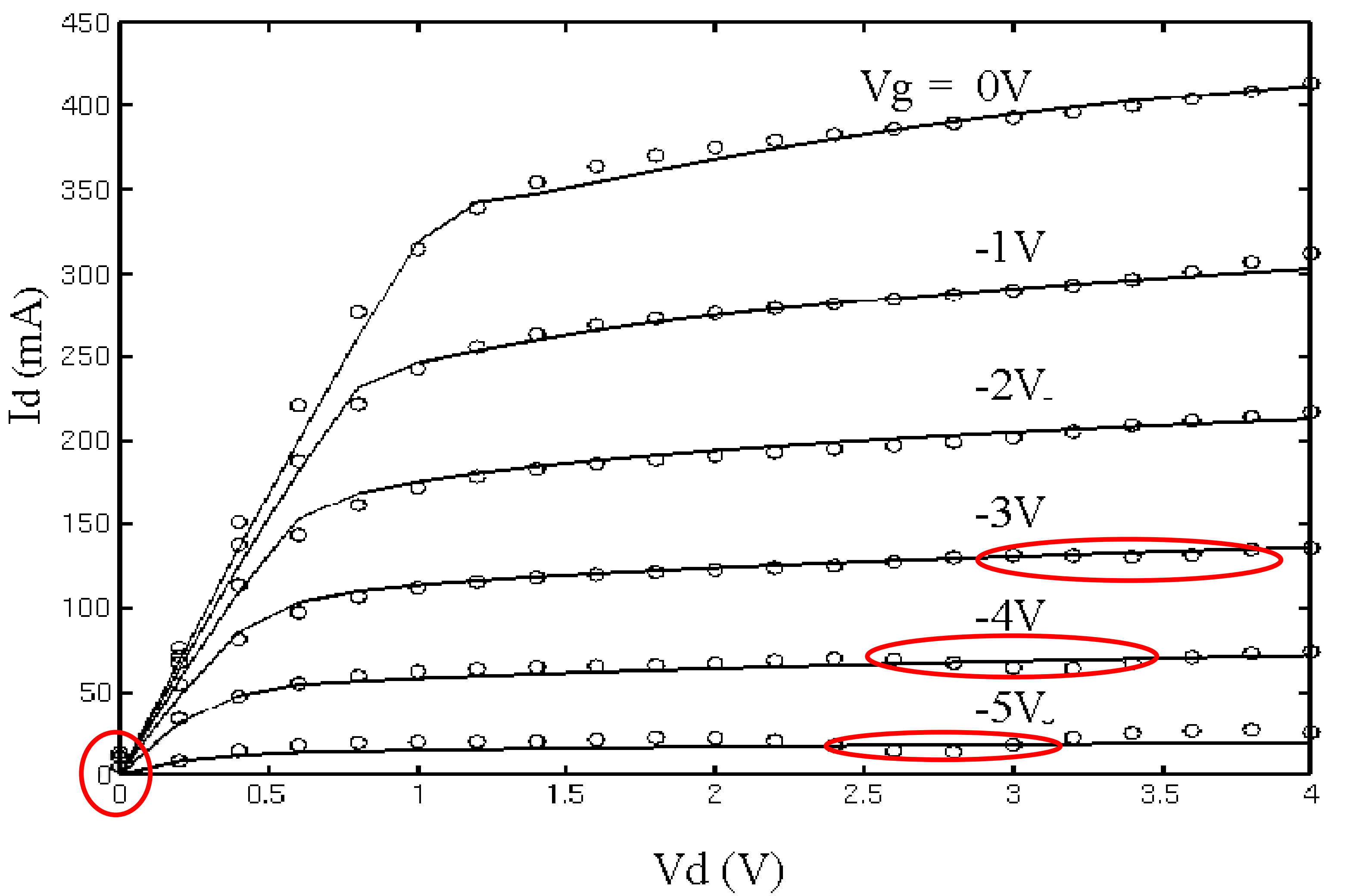
NN Design: Input Data Distribution

Select data points such that projections on x and y axes do not overlap



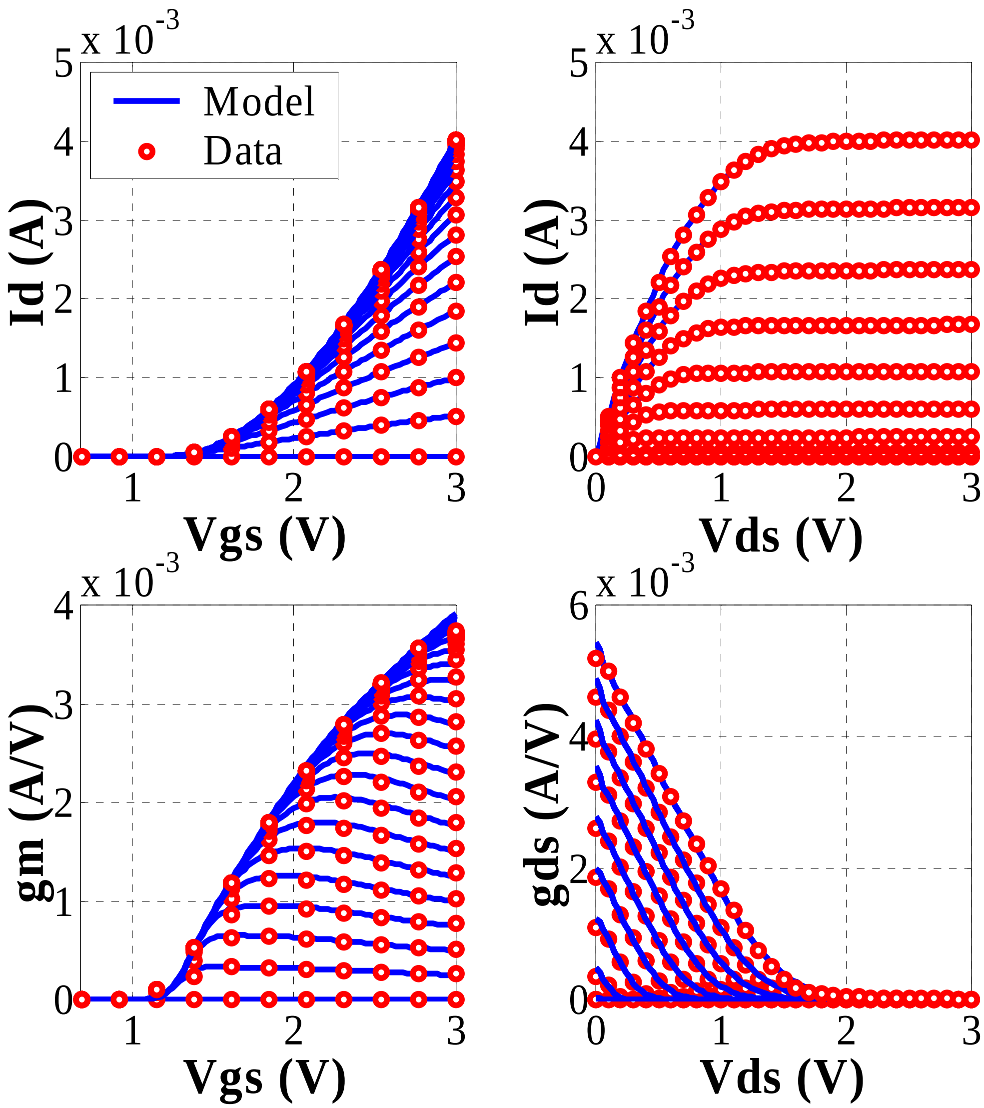


Device Modeling by Regular NN



MOSFET Modeling by Improved NN

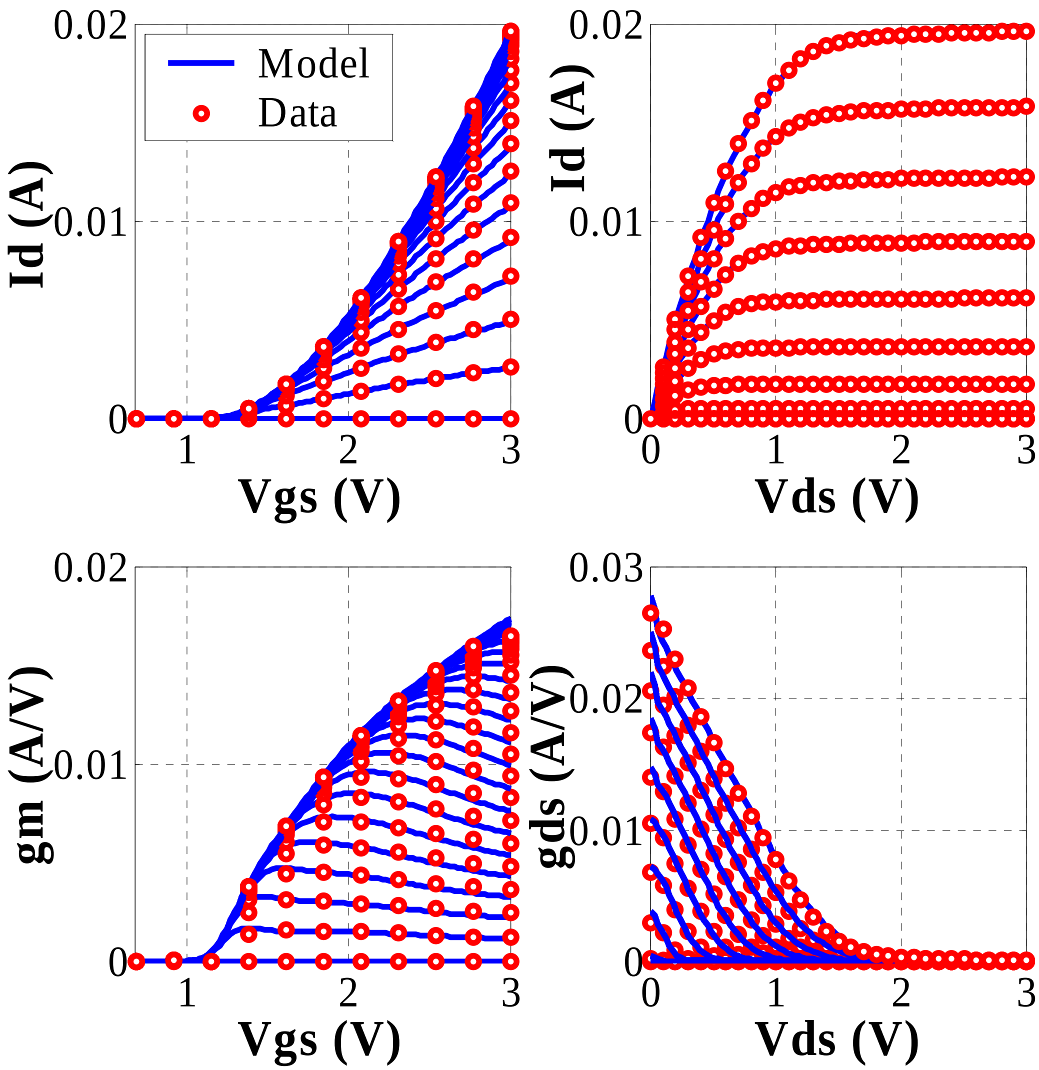
- **Comparison for data not used in training set**



W=45 μm
L=2 μm
Vsb=2 V
L=[1,100] μm
W=[1,100] μm
Vds=[0,3] V
Vgs=[0,3] V
Vsb=[0,3] V

MOSFET Modeling by Improved NN

- Comparison for data not used in training set



W=100 μm
L=1 μm
Vsb=2 V
L=[1,100] μm
W=[1,100] μm
Vds=[0,3] V
Vgs=[0,3] V
Vsb=[0,3] V

Performance Improvement

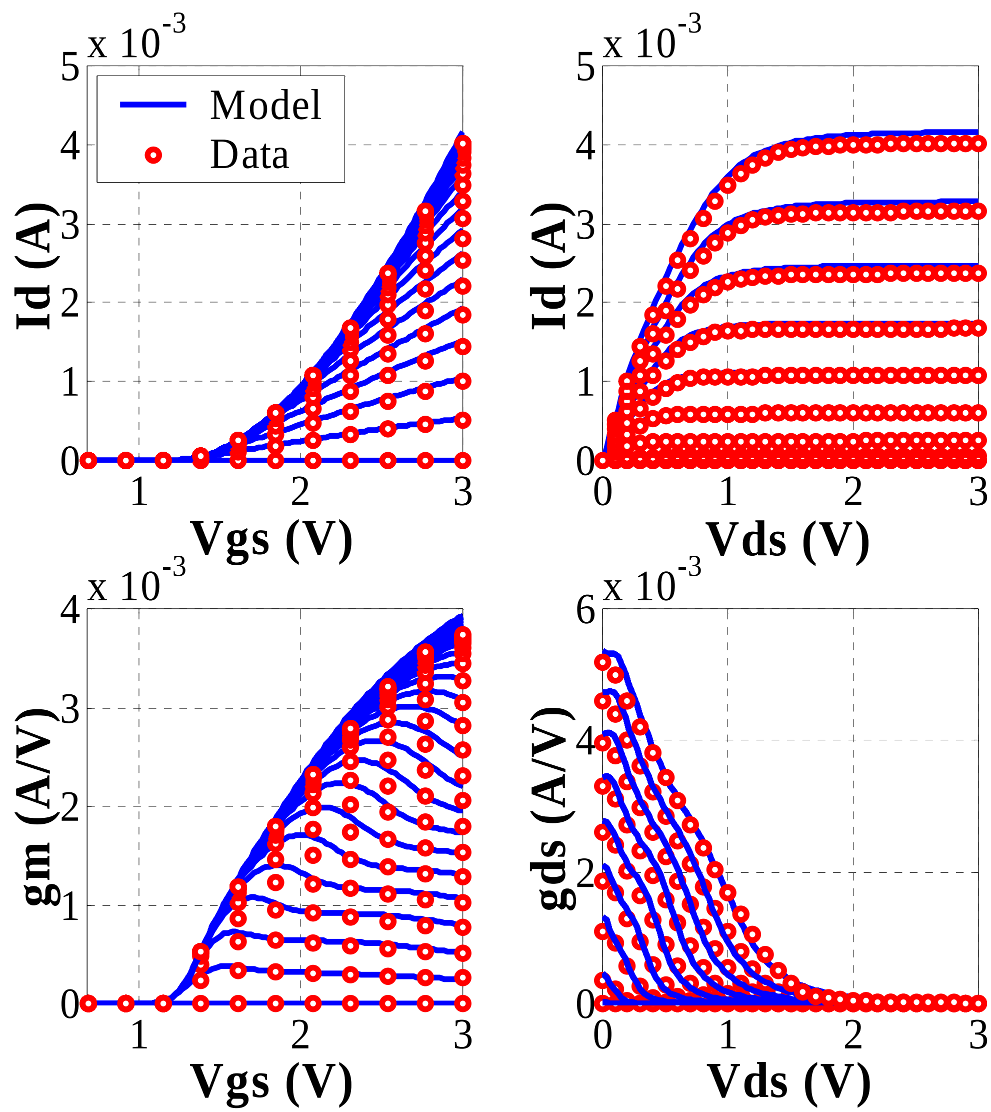
- **Efficient Activation Functions**
  - **$y=s/(1+|s|)$ - continuous first derivative**
  - **$y=(s+s*|s|)/(1+|s|+s^2)$ - continuous till 4th derivative**
- **Restricted Data Set**

  W=[40, 50] μm, L=2 μm, Vds=[0, 3] V,

  Vgs=[0, 3] V, Vsb=[0, 2] V

Performance Improvement

- Comparison for data not used in training set



W=45 μm, L=2 μm

Vds=[0, 3] V

Vgs=[0, 3] V

Vsb=[0, 2] V

Number of neurons:

8 on 1st hidden layer

4 on 2nd hidden layer

Original NN:

15 on 1st hidden layer

10 on 2nd hidden layer

Performance Evaluation

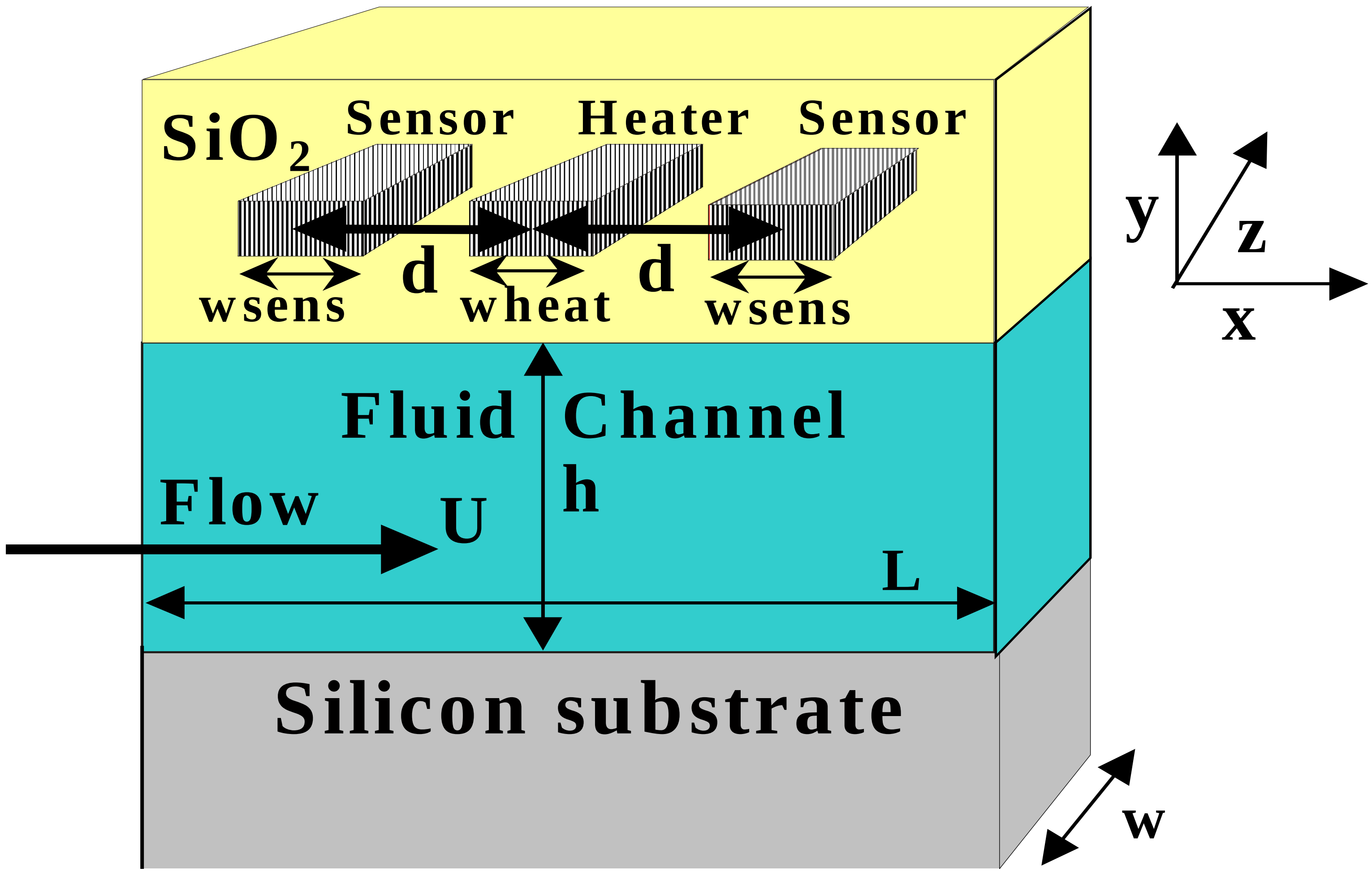
- **100 MOSFETs, 20000 steps, Sun ULTRA10, Solaris 5.8, SPICE3f5**

- **BSIM3 in SPICE3f5**
  - **total analysis time: 501 s**
  - **load time: 437 s**

- **Original NN**
  - **3x slower than BSIM3**

- **Improved NN**
  + **total analysis time: 135 s**
  + **load time: 98 s (4.5x faster than BSIM3)**

Microflow Sensor Modeling



Microflow Sensor Modeling

Basic equations for T(x, y) :

Silicon dioxide: $k_{sio2} div(\nabla T)=0$

Substrate: $k_{si} div(\nabla T)=0$

Sensors: $k_p div(\nabla T)+\chi_1=0$

Heater: $k_p div(\nabla T)+\chi_2=0$

Fluid channel: $k_{fluid} div(\nabla T)+\mu F(y)=\rho c_p u(y)\partial T/\partial x$,

$F(y)=(\partial u/\partial y)^2$, $u(y)=4Uy(h-y)/h^2$

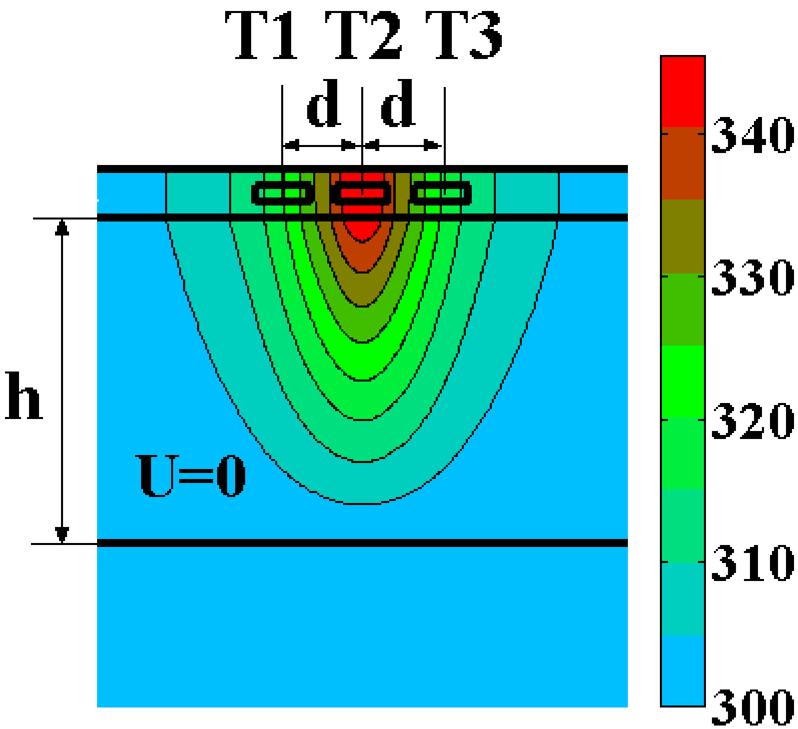
Boundary conditions:

Upper surface : $k_{sio2}\partial T/\partial y=\eta(T-T_\infty)$

Left, right, and bottom edges: $T(x=0)=T(x=L)=T(y=0)=T_\infty=constant.$

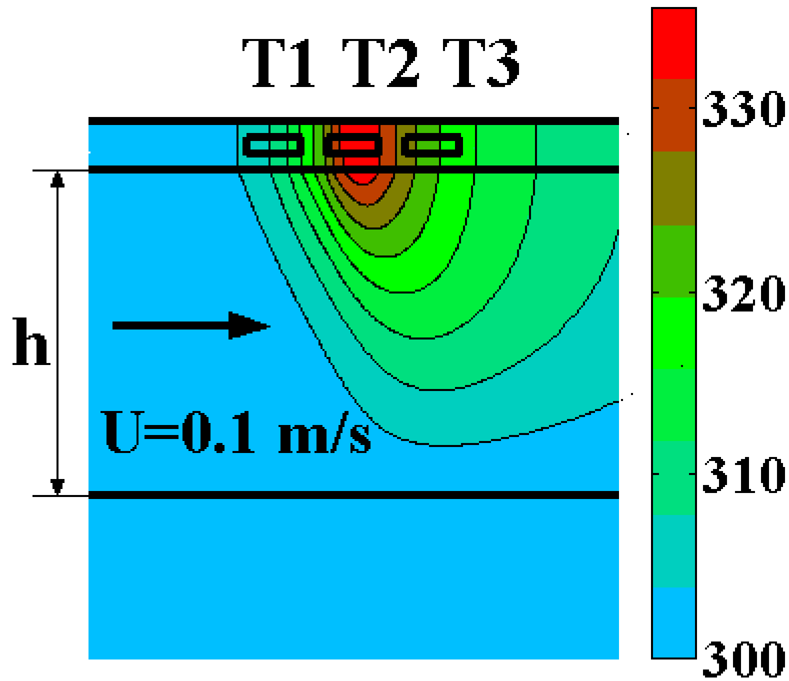
Microflow Sensor PDE Solutions

Channel height h = 20 μm, sensor separation d = 30 μm



No flow

ΔT=T3-T1=0 ºC

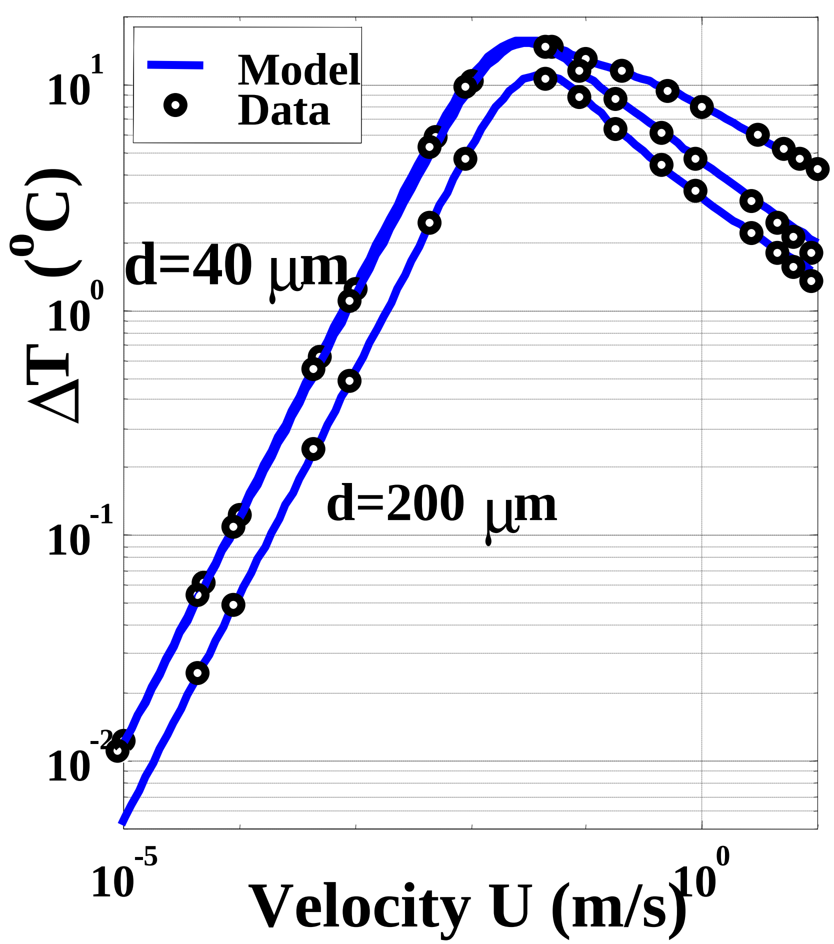


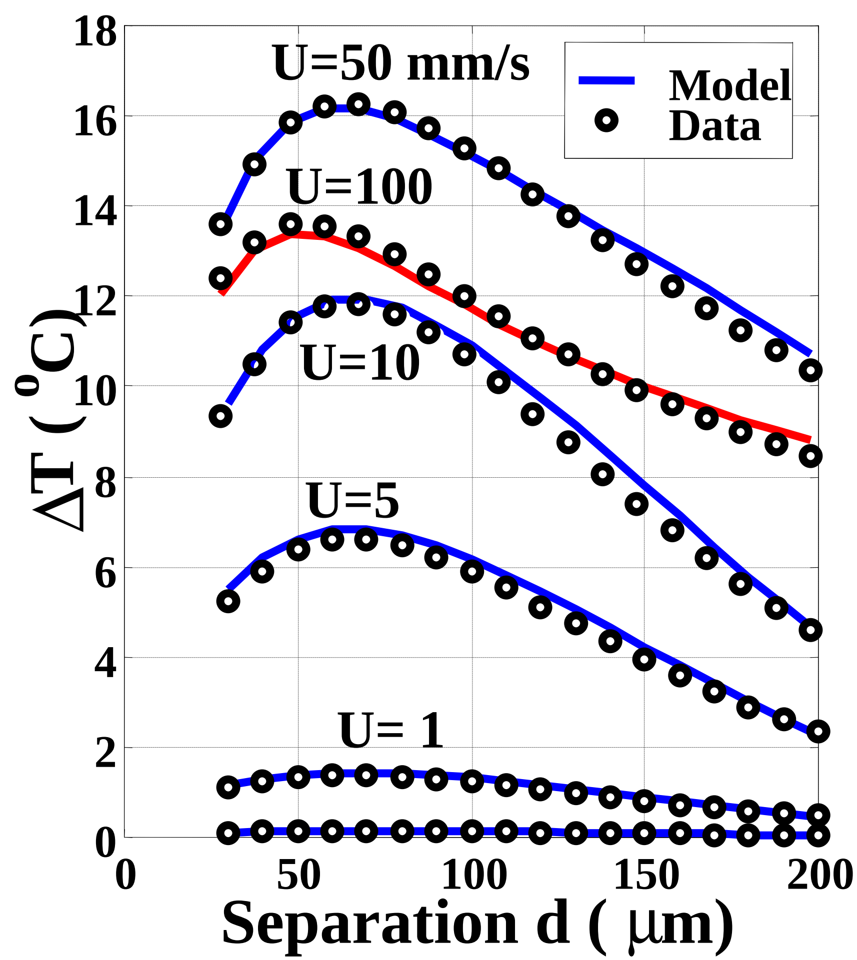
With flow

ΔT=T3-T1=11.8 ºC

Microflow Sensor Modeling by NN

h = 50 μm   Training set h={2, 20, 100, 200} μm





The channel height of 50 μm was not used in the training set

Conclusions

Shape Preserving Neural Networks are

- **simple**
- **accurate**
- **fast**
- **have reasonable physical behavior**
- **applicable to a wide range of parameters**
- **can handle trade-off: accuracy versus efficiency**