From dynamic modeling to neural networks

An informal talk

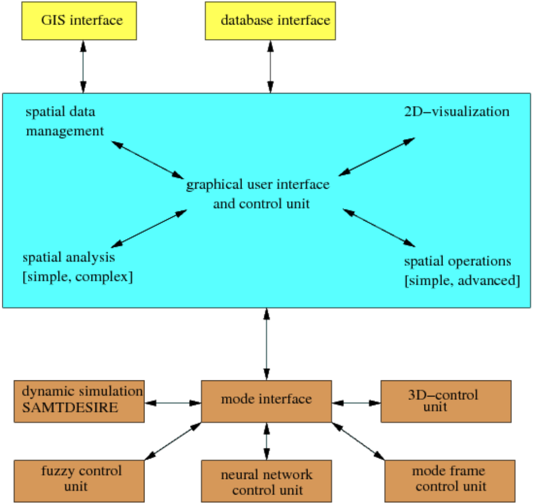
G.A. Korn, X. Holtmann, K. Groth, R. Wieland

ZALF/LSA: rwieland@zalf.de

Spatial Analysis and Modeling Tool (SAMT)

- Interactive
- Combines dynamic and spatial computations
- Allows the integration of fuzzy models and neural networks
- Has its own dynamic model development toolbox "SAMTDESIRE"
- Designed to integrate different particularly models to one "system"

Structure of SAMT



Why are we doing this?

- to understand the dynamics
- teach dynamics
- use SAMTDESIRE as a prototype builder before start a implementation in C++
- direct use of SAMTDESIRE in application (interface between SAMT and SAMTDESIRE)

SAMTDESIRE as a dynamic simulation system

- solves systems of differential equations (20.000 or more)
- includes 16 different solvers for DE
- is splitted in a interpreter part and a compiled dynamic part
- the compiler can use vector of DEs as model replication
- includes a great GUI (Xenia)
- can handle fuzzy models and neuronal networks from SAMT-Fuzzy and SAMT-NN
- interacts with SAMT

Lotka-Volterra system (predator-prey-model):

$$\frac{dx}{dt} = A * x - B * x * y$$
$$\frac{dy}{dt} = C * x * y - D * y$$

with $A, B, C, D > 0$

- Open "lotka.lst"
- Look at the two parts of the simulation: experiment protocol script and the "DYNAMIC" part in the model
- Change "dispt x,y" in "dispxy x,y" (phase plot)
- Play with the parameters

Spread of a epidemic plague

Spread of a rumor through a closed population: x=persons who are ignorant of the rumor; y=persons who are actively spreading the rumor;

$$\frac{dx}{dt} = -u * x * y \text{ with } x(0) = N$$

$$\frac{dy}{dt} = u * (x * y - y * (y - 1) - y * (N + 1 - y - x))$$

with: $N =$ population size; $u =$ control parameter

Evolon

Simple Evolon dynamic, it can be used as part of crop growth models

$$\frac{dx}{dt} = K * x^k * (B^w - x^w)^l$$

It is an enhanced logistic equation $\frac{dx}{dt} = K * x * (B - x)$ with a lot of possibilities of control using the additional parameters k, w, l.

Parameter variation in the Evolon

Problem: investigate in an appropriate set of parameters for a simplified Evolon

$$\text{Vectr}\frac{dx}{dt} = K * x * (B - x)$$

with x and B as arrays; the keyword “Vectr” organizes a model replication.
Remark: prefer model replication whenever it is possible, because it is easy to understand and very fast

Vector Operations of SAMTDESIRE

- Vector y = f(x, z, alpha, ...) compiles into y[i] = f(x[i], z[i], alpha, ...) (i=1,..,n)
- Vectr d/dt x = f(x,y,alpha, ...) compiles into d/dt x[i]=f(x[i], y[i], ...) (1,..,n)
- Vectr delta x = f(x,y,alpha, ...) compiles into x[i] = x[i] + f(x[i],y[i], ...) (1,..,n)
- DOT alpha = u * v compiles into $\sum_i^k u[i] * v[k]$
- Vectr d/dt x = A * x (A is a matrix)

Operations for neural networks

- DOT xnorm=x*x , xxx=1/sqrt(xnorm) , Vector X=xxx*x
- Vector error = TARGET# - v , DELTA W = lrate * error * x
- CLEARN v = W(x) lrate, *crit* < 0:
  $E(x, i) = \sum_{k+1}(W[i, k] - x[k])^2$, the winning v[I] is set to 1, and the template weights W[I,k] are updated to improve template matching:
  W[I,k] = W[I,k] + lrate * (x[k] - W[I,k]) (k = 1,2, ...)
- PLEARN Principal-axes Operations PLEARN y = W * x

Simple backprop neural training for three layers

DYNAMIC
Vector v1 = tanh(W1 * x)
Vector v2 = tanh(W2 * v1)
Vector y = tanh(W3 * v2)
Vector error = Y-y
Vector delta3 = error * $(1 - y^2)$
Vector delta2 = W3% * delta3 * $(1 - v2^2)$
Vector delta1 = W2% * delta2 * $(1 - v1^2)$
DELTA W1 = lrate1 * delta1 * x
DELTA W2 = lrate2 * delta2 * v1
DELTA W3 = lrate3 * delta3 * v2

Radial basis function network

```
DYNAMIC
x=ran()
Vector v=exp(b*(x-X)*(X-x))
DOT y=w*v
target=0.4*cos(4*x)
error=target-y
Vectr delta w=Wgain*error*v
```

Counter prop network

```
ARRAY x[1],v[n],h[n],y[1],error[1]
ARRAY W[n,1],U[1,n]
fill W with rand values and U with zeros
DYNAMIC
x[1]=ran()
target=cos(5*x[1])+noise*ran()
CLEARN v=W(x)lratex,crit
Delta h=v
Vector y=U*v
Vector error=target-y
DELTA U=lratef*error*v
```

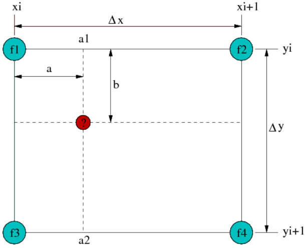
Connection between SAMT and SAMTDESIRE

Problem: SAMT can handle some million of grid cells; SAMTDESIRE can solve only up to 20.000 differential equations

- Values from SAMT are often discrete
- Continuous values can be often discretized without an significant error

For example if we have a range between [0..1], often we can calculate the values on $\{0.0, 0.1, 0.2, 0.3, ..., 0.9, 1.0\}$ and interpolate between these values.

Bilinear interpolation



$$
\begin{aligned}
a1 &= f1 + a/\Delta x * (f2 - f1) \\
a2 &= f3 + a/\Delta x * (f4 - f3) \\
q &= a1 + b/\Delta y * (a2 - a1)
\end{aligned}
$$

Remark: this method can be easily expanded to three dimensions

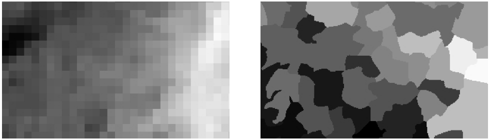
Example SAMT and SAMTDESIRE

Challenge: crop growth modeling of agricultural plants

- A map of soil quality index (B) and a map of temperature distribution (K) are given
- A simplified Evolon is used as dynamic model: $\frac{dx}{dt} = K * x * (B - x)$
- K can vary from [0.02 .. 0.04] and B can vary from [0.8 .. 1.0]
- K and B where discretized in 30 equal spaced steps each
- First simulation runs to the 150th day and in a second run to the 200th day from the sowing

Remark: see “evolon_samt.lst” and “interpol.config” for this example

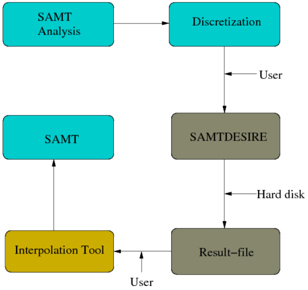
Example: spatial dynamics using SAMTDESIRE



K-parameter and soil quality index (the brighter the higher)

Biomass after 150 days and biomass after 200 days

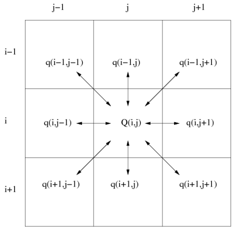
Interaction between SAMT and SAMTDESIRE



- The procedure is clear but not very elegant
- The user interactions should be avoided
- The storage to hard disk is slow
- The interaction is O.K. if there is no spatial fluxes

Spatial fluxes

Problem: lateral fluxes



$$\frac{dQ(i,j)}{dt} = f(Q(i,j), q(i-1,j-1), q(i-1,j), \ldots q(i,j-1))$$

Such a complicated system needs a lot of interaction between spatial and dynamic simulation

Dynamic spatial simulation

- Build a simulation model in C++ including spatial and dynamic parts
- Use shared memory for a fast interaction between spatial simulation and SAMTDESIRE

Summary

- SAMTDESIRE is very useful for teaching and research of dynamic systems
- SAMTDESIRE can open the mind to a new general view to neural networks as dynamic systems
- Open Source of SAMTDESIRE allows the integration of new methods easily (fuzzy models in SAMTDESIRE)
- SAMTDESIRE is used to check new methods before there are implemented in C++

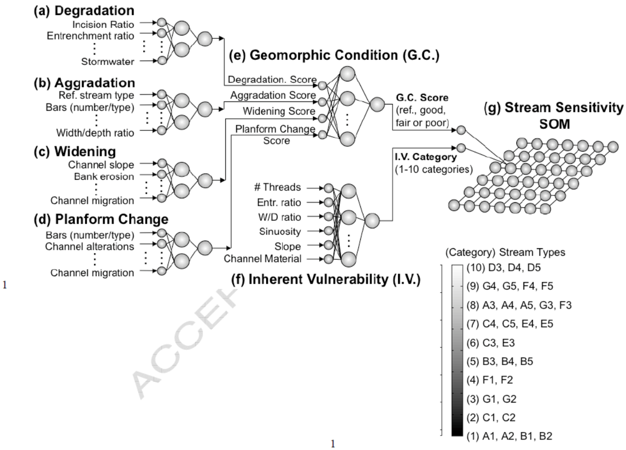
SADATO - SAMT Data Tool

- tool ready for the end user
- nice graphical user interface
- all data are store in a database engine (sqlite3 or mysql)
- simple import of CSV files
- includes a basic statistic toolbox for dataanalysis
- Open source and based on open source software (gsl, apophenia, fann, libsvm, qt, ...)
- was developed using SAMTDESIRE as development system (counterprop, RBF)

Regression example with a noisy yield estimation

- always start with statistics
- Can a neural network do a better job?
- What about a SVM?

Classification problem from Lance



Summary SADATO

SADATO is under development:

- new methods like (CP and RBF will be included)
- fuzzy training will be included this year
- trained nn or SVM will be available in SAMT
- documentation has to be written

You are invited to join the development!