Course G: Basic neurosciences 2

Computational neuroscience

Tutorial
Emil Ratko-Dehnert
Summer term 2011

About me

- Studied Mathematics and Psychology (LMU)
- Doctoral student in experimental psychology since 2009, working for Cotesys
- Research Interests:
  - Visual attention and memory
  - Formal modelling and systems theory
  - Philosophy of mind, Epistemology

Organisational Infos

- Room: CIP-Pool (Leopoldstr. 11b)
- Dates: 01.06, 15.06 and 22.06. (WED)
- Time : 16:00 – 18:00
- Slides, Materials and further information on my homepage (→ google „ratko-dehnert“)

Tutorial concept and aims

- Introducing some basic mathematical concepts relevant for computational neuroscience
- Trying to take away the terror of mathematical formalizations and notations
- Gaining hands-on experience with simbrain, an elaborate tool for simulating neural networks and their dynamics

Outline

01.06.: ODEs and Matrix Calculus

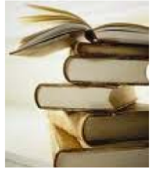
15.06.: Introducing Simbrain, labs on propagation, node rules and vectors in neural networks

22.06: Pattern association, labs on Hebbian Learning and Sensori-motor control; Evaluation

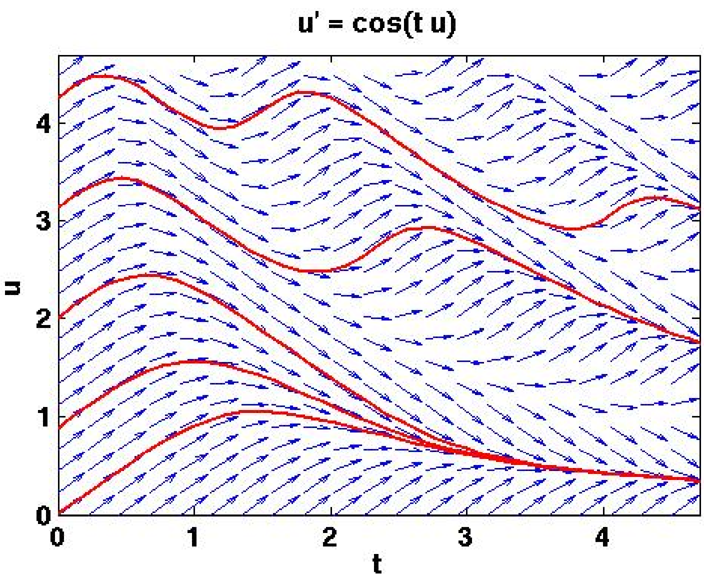
Grading of tutorial

- 1st Session
  - Handout with exercises to be solved in pairs
  - Will be collected by the start of the 2nd Session
- 2nd and 3rd Session
  - Solving simbrain labs in pairs; commenting on handout in pairs

Literature



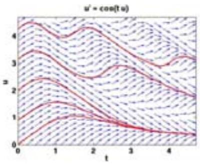
1. Essential Mathematics for Neuroscience *(Fabian Sinz)*
2. Mathematics for Computational Neuroscience & Imaging *(John Porrill)*
3. Simbrain: A visual framework for neural network analysis and education *(Jeff Yoshimi)*



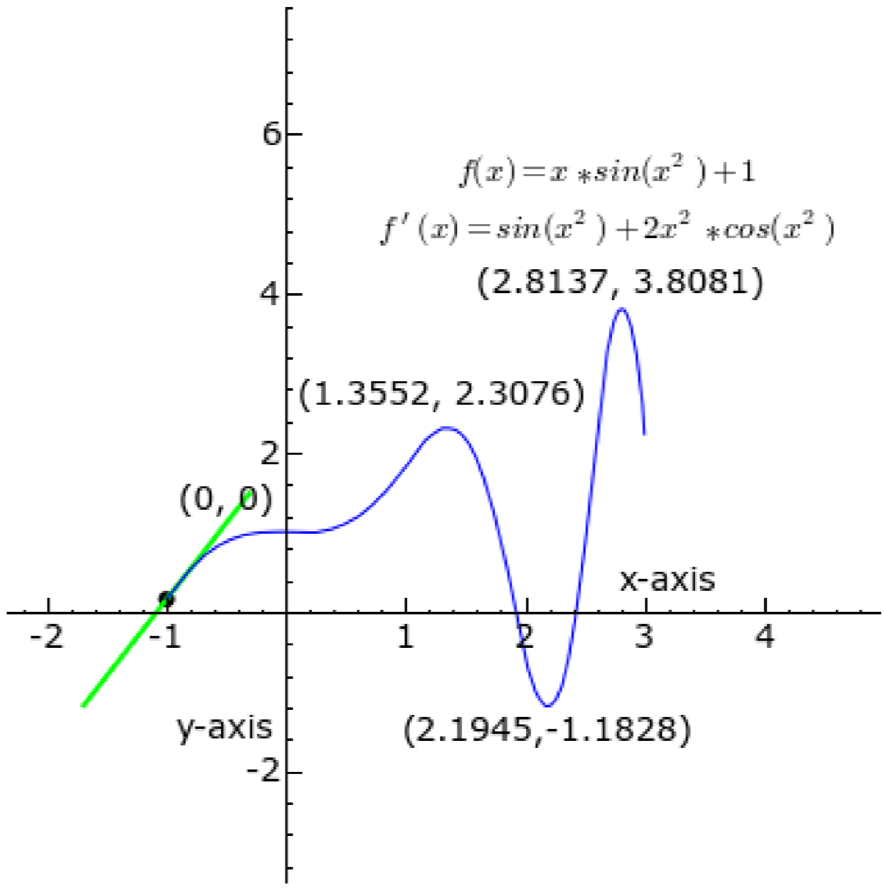
Session 1.1

DIFFERENTIAL EQUATIONS

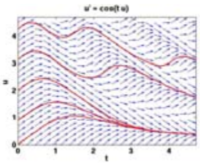
Calculus



- In Calculus one is interested in
  - solutions of functions
  - derivatives of functions
  - min/ max points
  - limits of series
  - integrals of functions, etc.



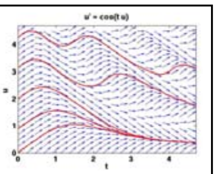
Differential Equations



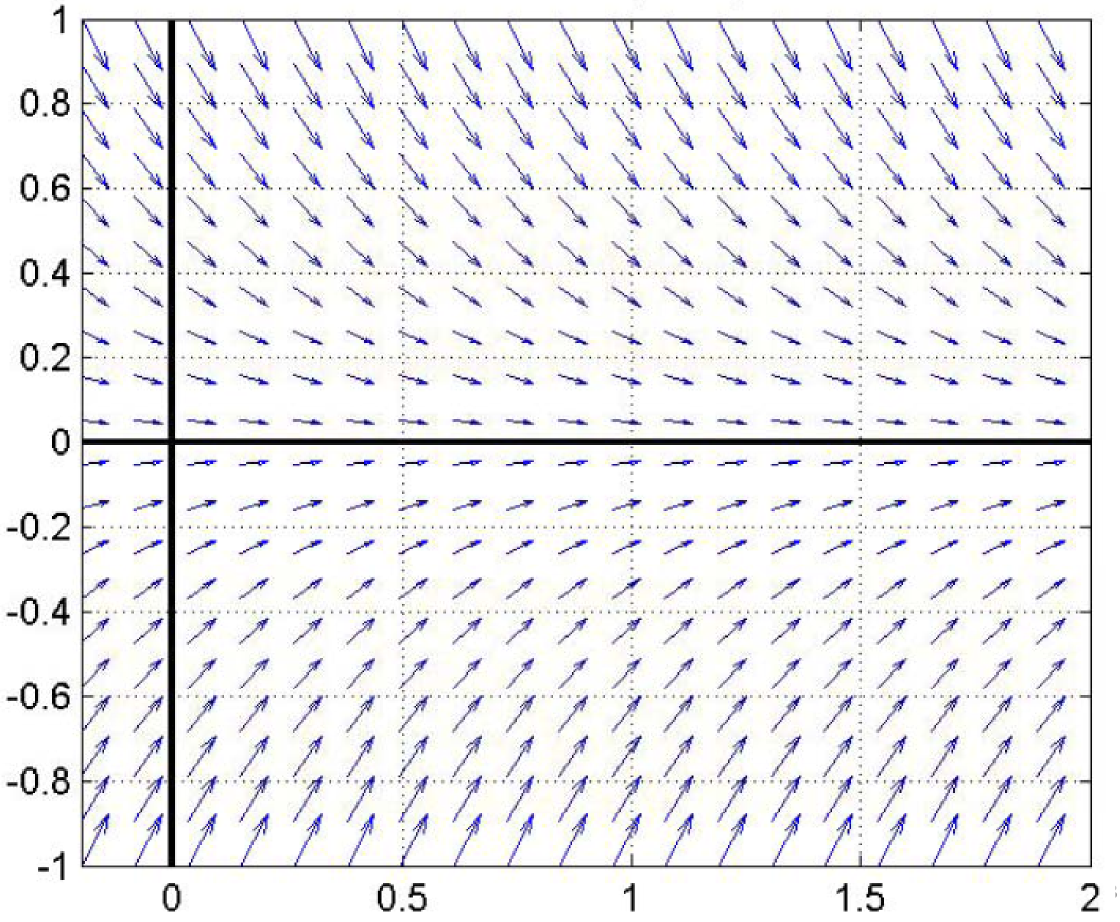
- A differential equation is a mathematical equation for an unknown function of one or several variables that relates the values of the function itself and its derivatives of various orders, e.g.

$$\frac{du}{dx} = cu + x^2 \Rightarrow u(x)?$$

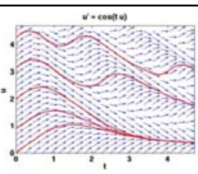
Applications



- Which function's slope matches the direction of the flow field (defined by the ODE) at every point?
- Has furthered most parts of physics
  - classical mechanics
  - statistical mechanics
  - dynamical systems
- But also applied fields like financial, biological and neuro sciences



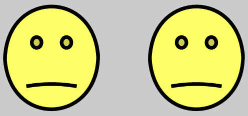
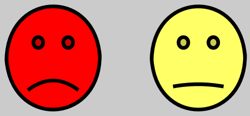
Classification

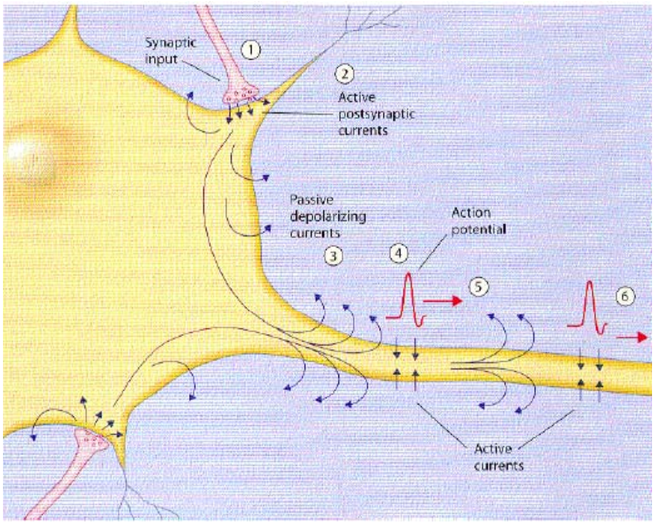


- Ordinary Differential Equations (ODEs)
  - Growth processes (linear, 1st order, constant coefficients)
  - Newtons 2nd law of motion (linear, 2nd order, constant coeff.)
- Partial Differential Equations (PDEs)
  - Heat Equation (linear, 2nd order) →
  - Wave Equation (linear, 2nd order)

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

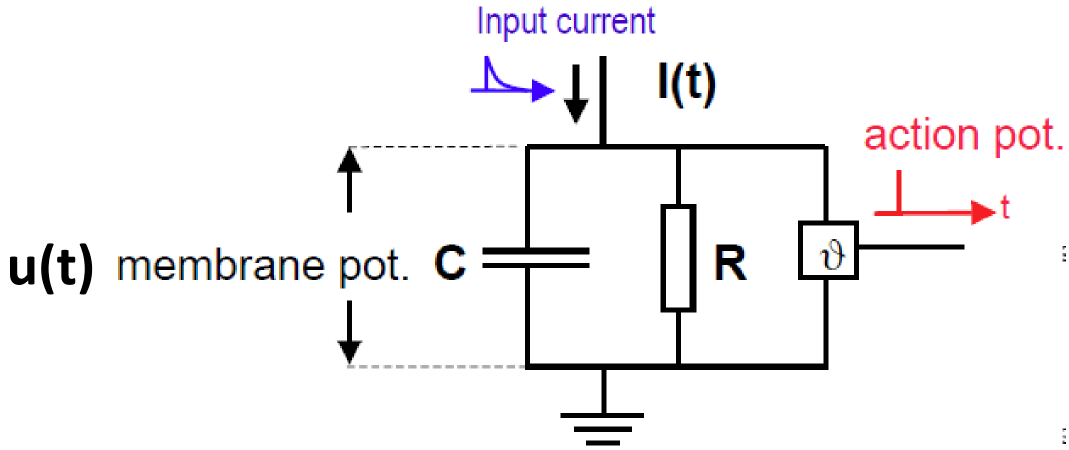
- Stochastic Differential Equations (SDEs)
  - Black-Scholes Formula (linear and non-linear, 2nd order)

	ODEs	PDEs	SDEs
Solutions?			
Unique?			
Analytic?/ Explicit			
Stability/ Linearity?			



THE HODGKIN-HUXLEY MODEL

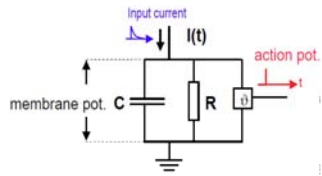
Hodgkin-Huxley‘s model of a neuron



$$\frac{du}{dt} = -k \cdot u(t) + m \cdot I(t)$$

$$\frac{du}{dt} = -\frac{1}{RC} \cdot u(t) + \frac{1}{C} \cdot I(t)$$

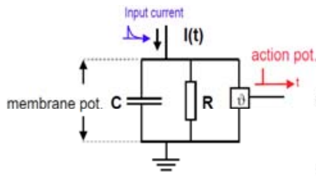
Deriving the model



- This is well-described by a very simple differential equation called the *leaky integrator equation*
- Because every system obeying this equation can be seen as a model for this, one can use a more concrete example to illustrate the derivation and system behaviour...

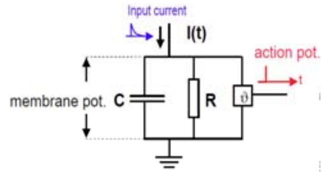
Filling a bath, when the plughole is open

Flow in



- **u(t)** = volume of water at time t
- **I(t)** = rate of flow in (volume/time)

Flow out

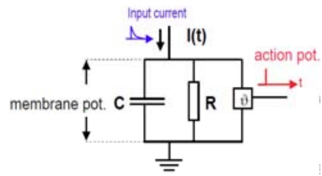


- Now more physics know-how is needed

$$\text{flow out} \propto \text{pressure} \propto \text{depth} \propto \text{volume}$$

- flow out = **k*u(t)**
- **k** constant scalar parameter
- **k>0**: water flowin out (not in)

Evaluate change rate

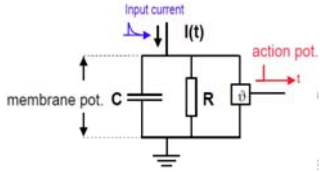


$$\frac{du}{dt} = \text{flow in} - \text{flow out} = I(t) - k \cdot u(t)$$

Now we know:

- Dynamical system (evolves aroundt **t**)
- Differential equation (relates **du** to **u**)
- Linear in **u**
- First order (highest derivative is **du**)
- Constant coefficients (**1, K**)

Going back to Hodgkin-Huxley

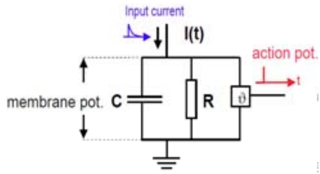


$$\frac{du}{dt} = I(t) - k \cdot u(t) \qquad \text{↲} \cdot \frac{1}{C}$$

$$\frac{du}{dt} = \frac{1}{C} I(t) - \frac{k}{C} \cdot u(t) \qquad \text{↲} \; k = \frac{1}{R}$$

$$\frac{du}{dt} = \frac{1}{C} I(t) - \frac{1}{RC} \cdot u(t) \qquad \square$$

Exact integrator



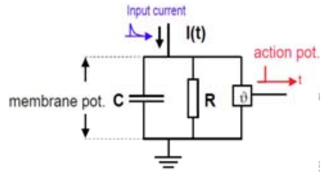
- Special case: **k = 0** (i.e. no leakage)

$$\frac{du}{dt} = I(t) \quad \xrightarrow{\text{FTC}} \quad u(t) = \int_0^t I(\tau)\, d\tau + u_0$$

$$u(t) = u_0 + \int_0^t I_{\max}\, d\tau = u_0 + \left[I_{\max} \right]_0^t$$

$$= u_0 + \left[I_{\max} \cdot t - I_{\max} \cdot 0 \right] = u_0 + I_{\max} \cdot t$$

Sanity checks

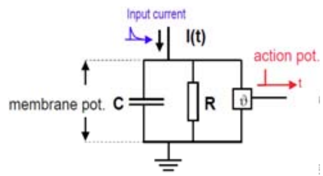


- When will the bath be full?

$$u_0 + I_{\max} \cdot t = u_{\max} \ ?$$

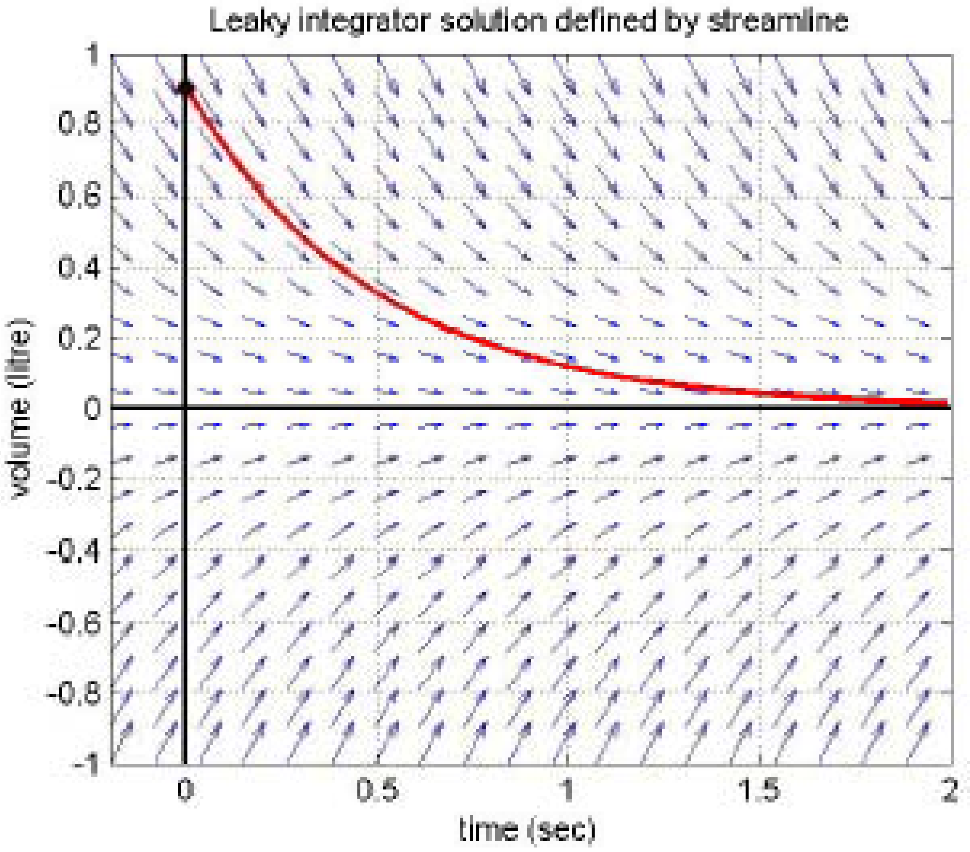
$$t = t_{full} = \frac{u_{\max} - u_0}{I_{\max}}$$

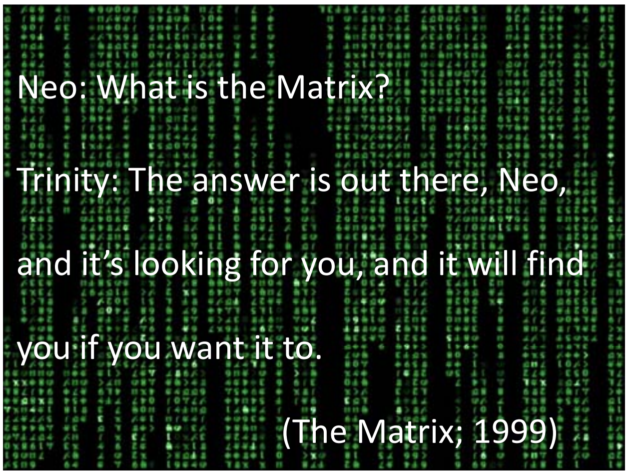
Transient behaviour



- Special case: setting **I(t) = 0**

$$\frac{du}{dt} = -k \cdot u(t) \quad \xrightarrow{\text{Intuition}} \quad u(t) = A \cdot \exp^{-kt}$$





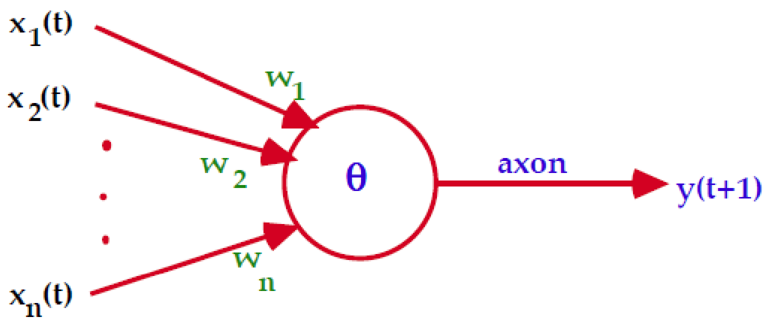
Session 1.2

MATRIX CALCULUS

What will we do

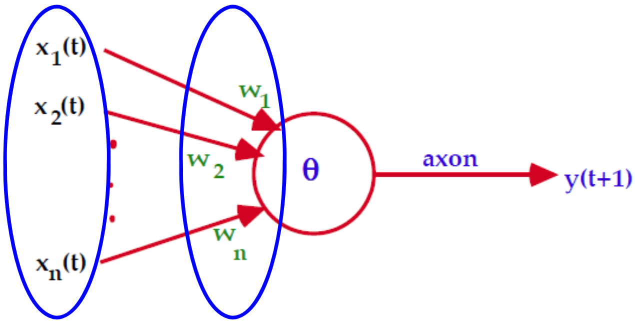
- Examples where vector and matrix calculus plays a role
- Reprise matrices and their operations
- Apply matrix notation to neural networks

McCulloch-Pitts neuron



$$y(t+1) = \begin{cases} 1, & if \sum_i w_i x_i(t) \geq \theta \\ 0, & if \sum_i w_i x_i(t) < \theta \end{cases}$$

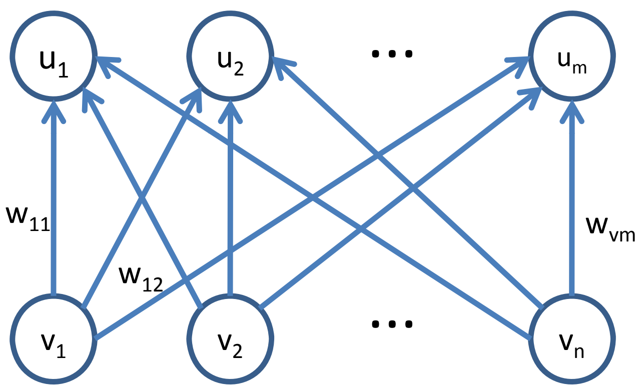
Inner product of vectors



$$\langle x, w \rangle = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_i \\ \vdots \\ w_n \end{pmatrix} = \sum_i w_i x_i$$

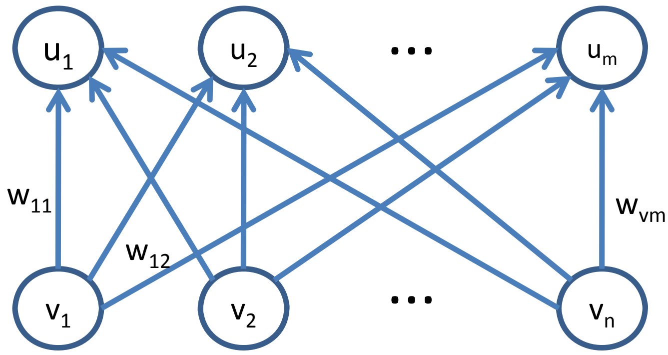
Linear associator

- Here one would have to compute inner products for **n*m** pairs of inputs and outputs
- Alternatively ...



Matrix interpretation

- Regard the weights as a coefficient matrix and the inputs and outputs as vectors



- But what does this mean and how does one compute that?

$$u = W \cdot v$$

What is a matrix?

- Def: A matrix ***A = (a***$_{i,j}$***)*** is an array of numbers or variables
- It has ***m*** rows and ***n*** columns (***dim = (m, n)***)

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & & & & \vdots \\ \vdots & & & & & \vdots \\ a_{m,1} & \cdots & \cdots & \cdots & \cdots & a_{m,n} \end{bmatrix}$$

m

n

Connection to vectors

- Trivially, every vector can be interpreted as a matrix
- e.g. x is an (n, 1)-matrix
- Thus all the following statements hold true for them as well

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix}$$

Equality of matrices

- Two matrices are equal, if and only if (iff) the have the same dimension (m, n) and all the elements are identical

$$A = B \iff A_{i,j} = B_{i,j}, \ \forall i, j$$

How are matrices added up?

- Addition of two (2, 2)-matrices ***A, B*** performed component-wise:

$$\begin{bmatrix} 1 & 4 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 1 & -1 \end{bmatrix}$$

A **B** **A+B**

- Note that „+“ is commutative, i.e. A+B = B+A

Scalar Multiplication

- Scalar Multiplication of a (2, 2)-matrix ***A*** with a scalar ***c***

$$2$$

c

- Again commutativity, i.e. ***c*A = A*c***

Matrix multiplication

- Matrix multiplication of matrices ***C (2-by-3)*** and ***D (3-by-2)*** to ***E (2-by-2)***:

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

C **D** **E**

$$E_{11} = 1 \cdot 3 + 0 \cdot 2 + 2 \cdot 1 = 5$$

Falk-Schema

$$A \cdot B = C$$

			b_{11}	b_{12}	b_{13}
			b_{21}	b_{22}	b_{23}
			b_{31}	b_{32}	b_{33}
a_{11}	a_{12}	a_{13}	c_{11}	c_{12}	c_{13}
a_{21}	a_{22}	a_{23}	c_{21}	c_{22}	c_{23}
a_{31}	a_{32}	a_{33}	c_{31}	c_{32}	c_{33}

Matrix specifics

!Warning!

One can only multiply matrices if their **dimensions** correspond, i.e. (m,n) * (n, k) → (m, k)

- And generally: if ***A*B*** exists, ***B*A*** need not
- Furthermore: if ***A*B***, ***B*A*** exists, they need not be equal!

Transposition

- Transposition of a 2-by-3 matrix ***A*** → $\boldsymbol{A}^T$

$$\underbrace{\begin{bmatrix} 1 & 2 & 4 \\ 0 & -6 & 9 \end{bmatrix}^T}_{\mathbf{A}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 4 & 9 \end{bmatrix}}_{\mathbf{A}^T}$$

- It holds, that $A^{TT} = A$.

Matrix inversion

Square case, i.e. **dim = (n, n)**

- If **A** is regular (determinant is non-zero), then $\mathbf{A}^{-1}$ exists, with $\mathbf{A} * \mathbf{A}^{-1} = \mathbf{A}^{-1} * \mathbf{A} = \mathbf{I}_n$

Non-square matrices (**dim = (n, m)**)

- **A** with **dim = (n, m)** has a right inverse **B** with **dim (m, n)**, if the rank of **A** is **n** and a left inverse **B'** with **dim (n, m),** if the rank of **A** is **m.**
- It holds that $\mathbf{A} * \mathbf{B} = \mathbf{I}_n$ and $\mathbf{B'} * \mathbf{A} = \mathbf{I}_m$

Methods of inversion

- Gauss-Jordan elimination algorithm
- Cramer's Rule
- For **dim = (2, 2)** there exists a short and explicit formula

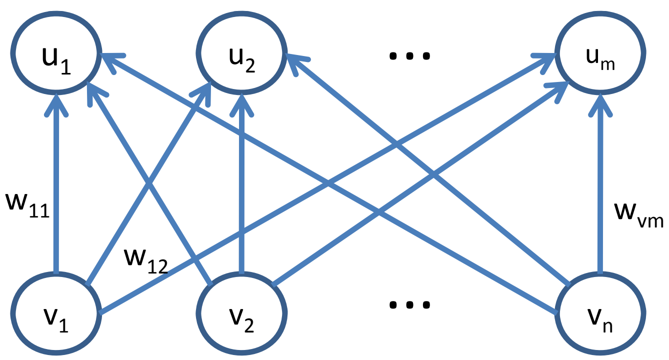
$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \underbrace{\frac{1}{ad-bc}}_{1/\det(A)} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Significance of matrices

- Matrix calculus is relevant for
  - *Algebra*: Solving systems linear equations (**Ax = b**)
  - *Statistics*: LLS, covariance matrices of random variables
  - *Calculus*: differentiation of multidimensional functions
  - *Physics*: mechanics, linear combinations of quantum states and many more

Back to linear associators

$$u = W \cdot v$$



Back to linear associators

- We need different operations to address issues like
  - What will be the output **u** of a given input **v**, when we know the configuration of the weights **W**? --> **$u=W*v$**
  - For a given output **u** and weight matrix **W**, what was the input **v**? --> **$W^{-1}*u = v$**
  - Compute the weight matrix **W** for desired input/output associations **v/u**. --> **$u*v^{-1} = W$**