1 – Overview lecture, intro to ML (Elin)

ML is an area of AI which enables machines to find models describing the data and correlations. It can for example: predict, classify, identify or decide how to act to get the best reward.

The computer doesn’t know what it is looking for and what features are interesting. Can easily be confident in its identifications even in white noise. It is important to have all different background and surroundings in the images in the training set, otherwise it may actually use background as features.

**Baeysian (probabilistic) methods**.

2 – Python and basic maths (Jacek)

# Python

**Python version is Python 3.6.1**

Jupyter – Literate programming, combined for reading and programming/compiling

You may run single cells and compile cell by cell.

If you want to **import all functions in module**:

from math import \*

If you do this you don’t need to write math.foo() but only foo()

Lists can mix types of elements, [bool, int, String]

Dictionaries are useful:

Params = { “param1” : 1.0,

“param2” : 2.0 }

Constructor: def \_\_init\_\_(self)

toString: def \_\_str\_\_(self)

Need to add: self, in beginning of functions

To declare a specific type you can write **dtype=type**

## Data

CSV, comma separated values, is one common data shape.

Colon can be a “wildcard” for getting for example a full row or col. Ex. M[1, :]

# Basic maths

## Tensors

* Scalar 0D
* Vectors 1D
* Arrays/matrices 2D
* 3D tensors
* 4D tensors
* …

Usually data is represented in tensors. In this course a 2D tensor but could be way more.

The concepts of **neighbors** is really important, think of interpreting an image. This is why we don’t want to store an image in a vector but in an array instead.

## Tensor operations

The arrays are ordered as **array(row, col)**.

Array addition, multiplications and so on gives **scalar element wise operations.**

We can **convert** arrays to matrices in order to perform **tensor(matrix) wise operations.**

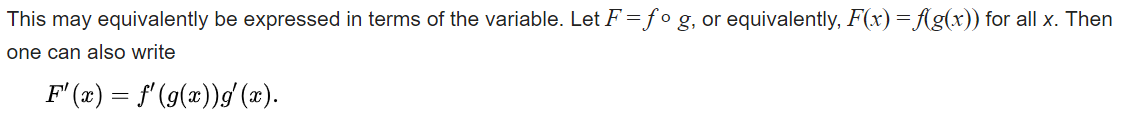
You **need to know** which space you are operating in. Is it **scalar or matrix** space?

Conversion in Python **M = matrix(A)**

## Chaining rule for derivation



This may equivalently be expressed in terms of the variable. Let *F* = *f* ∘ *g*, or equivalently, *F*(*x*) = *f*(*g*(*x*)) for all *x*. Then one can also write:



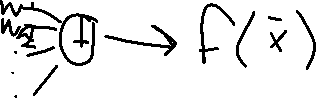
## Gradient descent

Tweak parameter array until you found the local extreme.

# Keras in chapter 2.1

**Activation=’relu’** is specifying the activation function.

When overfitting the training model has a significantly better



accuracy than the test data.