CHAPTER 2: MATHEMATICAL BUILDING BLOCK OF NN

**A FIRST LOOK AT NN**

1. **NIST (MNIST or National Institute of Standards and Technology):** like hello world of DL => verify if algorithm works as intended.

from keras.datasets import mnist

(training\_set), (test\_set) = mist.load\_data()

1. In classification: category = class, datapoints = samples, class w/ specific sample = label

Ex: Images will be encoded in Numpy arrays, label will be a array of digits (0 to 9) => handwriting problem

1. Workflow:
   1. Feed NN training data
   2. NN will learn to associate images and labels
   3. Use it on test data
2. The core building block of NN is the layer, a data-processing module like a filter for data
3. **Layers** extract representations out of the data fed into them

Combine simple layers => form of progressive **data distillation**

1. A DL model is like a sieve for data processing, made of succession of increasingly refinded data filter- the layers

from keras import models

from keras import layers

network = models.Sequential()

network.add(layers.Dense(512, activation='relu', input\_shape=(28 \* 28,)))

network.add(layers.Dense(10, activation='softmax'))

1. **2 Dense** layers (densely/fully connected), the 2nd layers is 10-way softmax layer => return an **array** of 10 probability scores (summing to 1). Each score => probability that current digit image belongs to.
2. To make NN ready for training: 3 more things: **loss function/optimizer/metric to monitor process**