# Learning in IR (2)

In last week’s class, we reviewed the notion of learning in IR and looked at the application of one approach, namely evolutionary computation as a search for solutions in information retrieval.

The advantages were that we have solutions that could be analysed. However, the usefulness of the solution found is dependent on the usefulness of the primitive features chosen to extract from queries and the documents collection.

Dominant learning approach in the last few years has been the *neural* approach.

These can be seen as being applied directly to:

i) The information retrieval tasks itself

ii) To other related problems/areas that can feed into the IR process

iii) Related problems in IR (e.g. query suggestion)

Approaches in the domain have been both supervised and unsupervised.

One of the first approaches to adopt a Neural network model can be traced back to the 1980s.

Effectively a three layer network:

Documents Terms Query

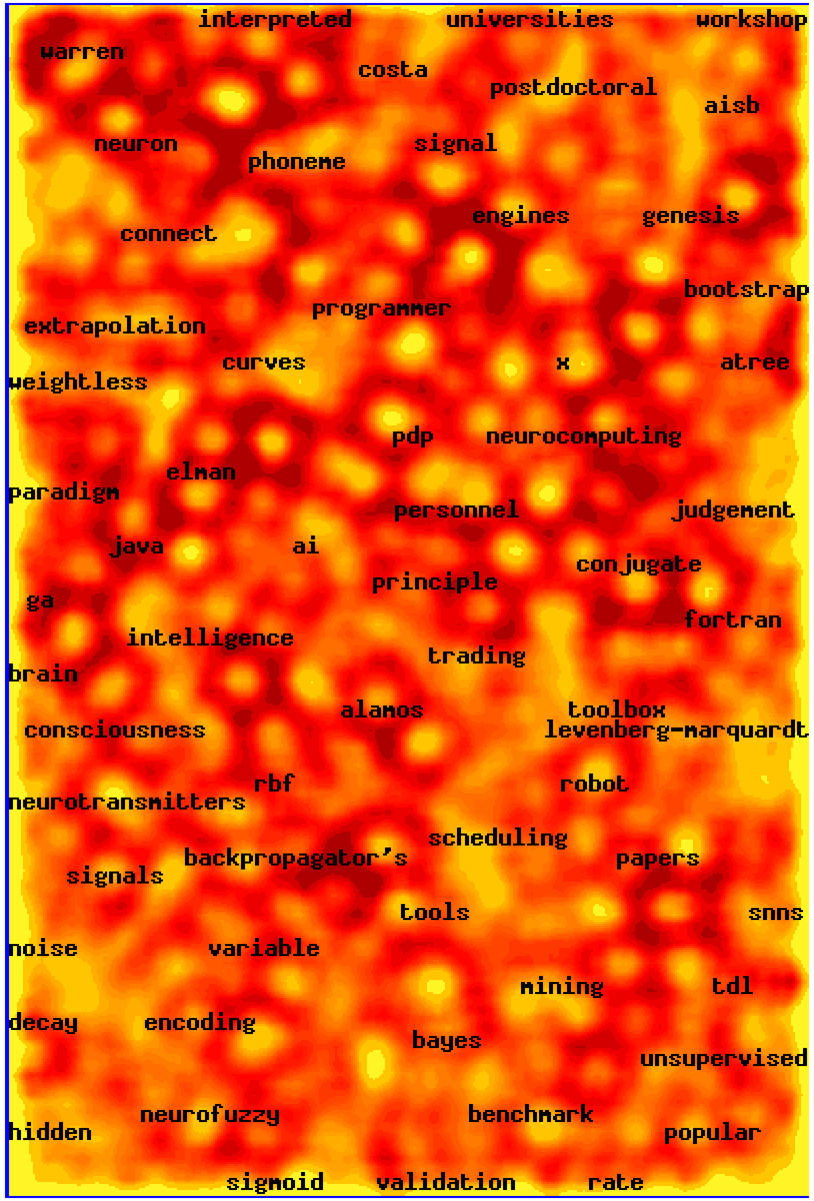
Spreading activation method used; query nodes are highlights; these propagate to terms which in term highlight certain documents.
Captured some interesting notions.

## Sample approach:  Case study

Self organizing maps: unsupervised learning

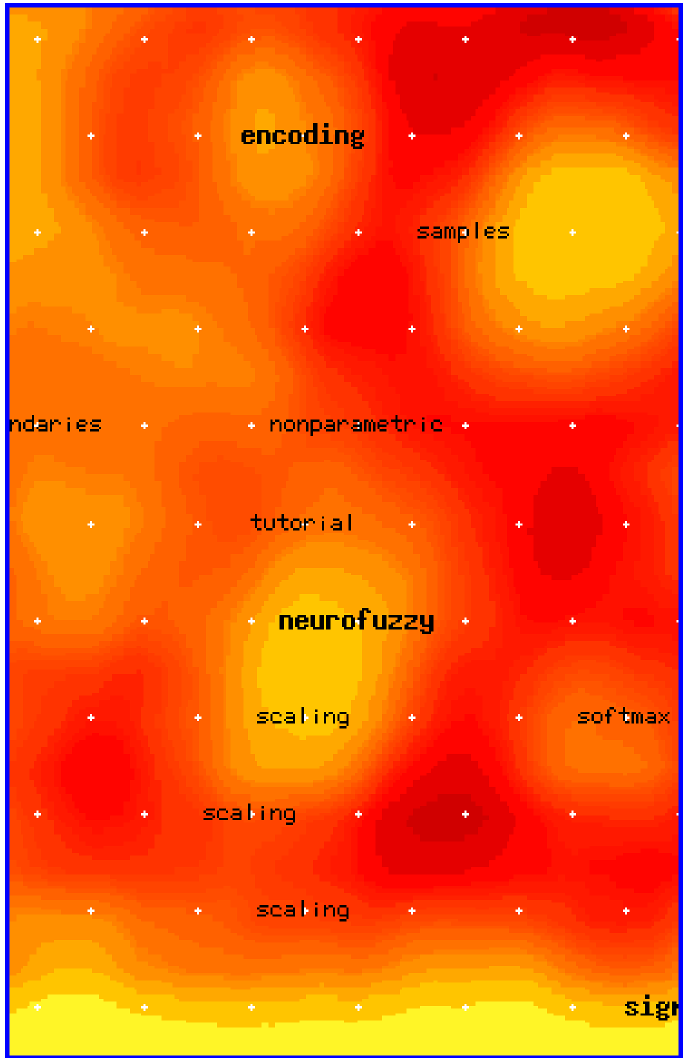
map documents to 2D space; dense areas indicate clusters hierarchically

Kohonen: 2D space; each region characterized/represented by terms



SOM created over a collection of AI related documents.

Users can traverse the collection by clicking on area of map of interest



Finally, users arrive at a list of papers/articles that have been clustered together.

# Self-Organising Maps (Kohonen)

Represents a sub-symbolic, neural approach to clustering.

The algorithm takes a set of n-dimensional vectors and attempts to map them onto a two dimensional grid.

The grid comprises a set of nodes, each of which is assigned an N-dimensional vector. These vectors contain randomly assigned weights.

**Algorithm:**

```
Repeat until weights on grid converge
    Select an input vector randomly.
    Identify grid node which is `closest' to input vector. (the `winning node')
    Adjust weights on winning node so that it is closer to the input vector.
    Adjust weights on nodes near the winning node so that they are closer to the winning node.
```

Notes:

Rate of modification of weights decreases over time.

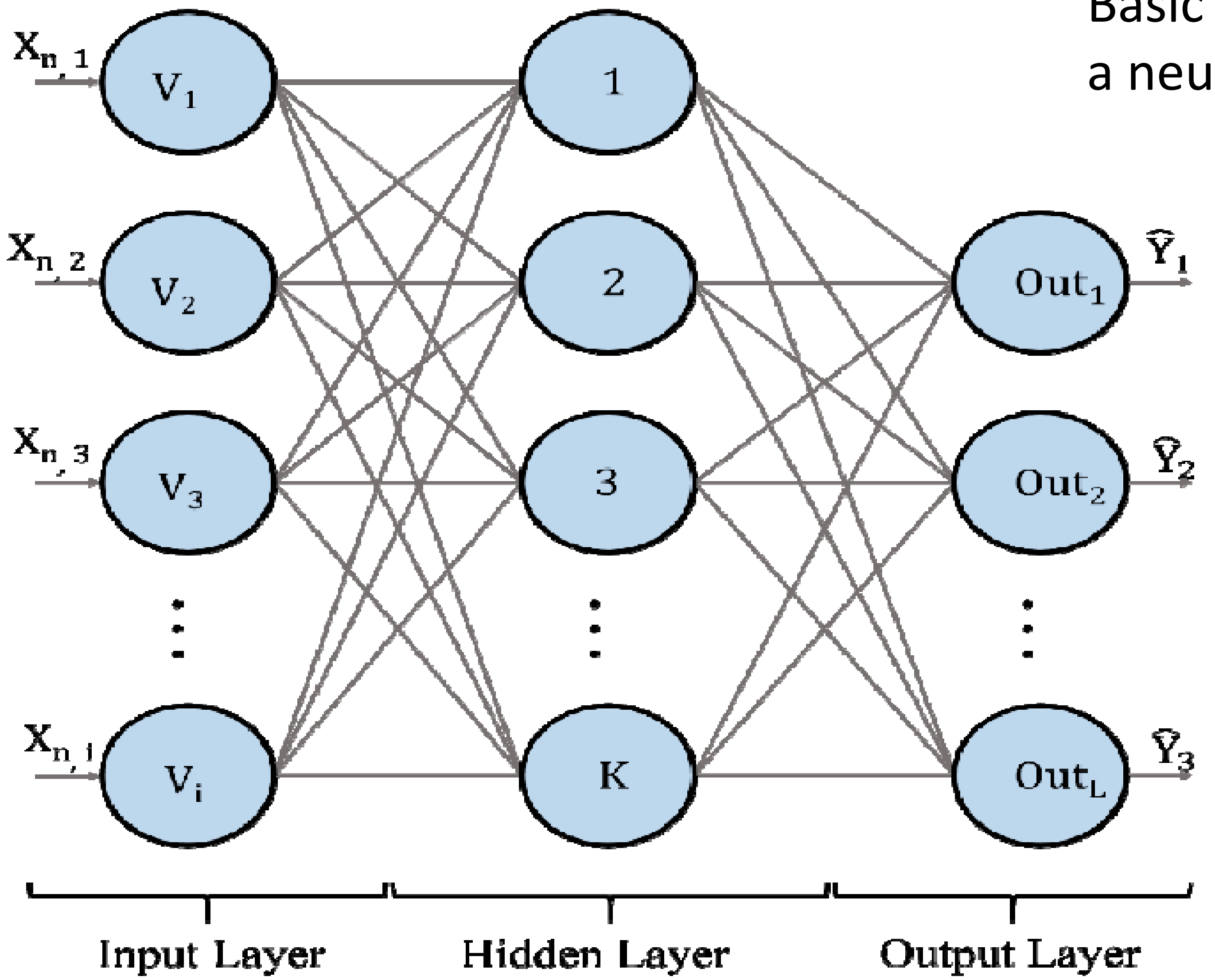
Size of the neighbourhood (near winning node) affected decreases over time.

The resulting clustering of tuples maintains the distance relationship between the input data.

# More recent neural approaches

**Overview NN**

Basic architecture of a neural network



Huge interest in the application of NN models to IR in the past few years; several breakthroughs due to the use of neural networks with multiple layers (so called deep architectures) and the availability of large datasets and computing power.

Proposed neural models learn representations of language from raw text that can bridge the gap between query and document vocabulary

Neural IR  is the application of shallow or deep NN to IR tasks.

Neural models for IR use vector representations of text and usually contain a large number of parameters that need to be tuned.

ML models with large set of parameters benefits from large quantities of training data.

Unlike traditional “Learning to rank” approaches that train over a set of hand crafted features, more recent NN accept the raw text as input.

Some main steps in classical IR:

Generating a representation of the user’s query

Generating a representation of the document that captures the ‘content’ of the document

Generating a ‘similarity’ score (comparison)

All neural approaches can be classified as to whether they affect:

i) the representation of the query

ii)the document

iii) the comparison

By inspecting only the query terms the IR mode ignores all evidence of the context provided in the rest of the document…only occurrences of a word are included and not other terms that captures the same meaning or the same topic

Traditional IR models have also used dense vector representations of terms and documents  (e.g. Deerwester).

Many neural representation have commonalities with these traditional approaches.

# Query representation

Types of vector representations – one hot representations (akin to what we have seen thus far in that each term is represented by one value)

Distributed representation – typically a real valued vector which attempts to better capture the meaning of the terms

NNs often used to learn this ‘embedding’ – this representation of terms

# Distributed representation

The distributional hypothesis states that terms that occur in similar contexts tend to be semantically similar

Am embedding is a representation of items in a new space such that the properties of, and the relationships between the items are preserved from the original representation.

Many algorithms – *word2vec* etc

Generate representation for queries and for documents

Compare query and document in this embedding space

Document and query that are similar should be similar in this embedding space

Text and language is typically represented as sequence

For analysing questions and sentences, we need to learn/model sequences:

Recurrent Neural Network – neuron’s output is a function of the current state of the neuron and the input vector. Very successful in capturing/learning sequential relationships.

(e.g LSTM, GRU)

Large set of architectures used – different topologies.

Convolutional (most often associated with images) also used to learn relationships between terms

Sequential processing has been used in query understanding, retrieval, expansion etc.

# Summary (neural approaches)

Powerful approach

Typically more computationally expensive than traditional approaches.

Good performance

Explainability - issues