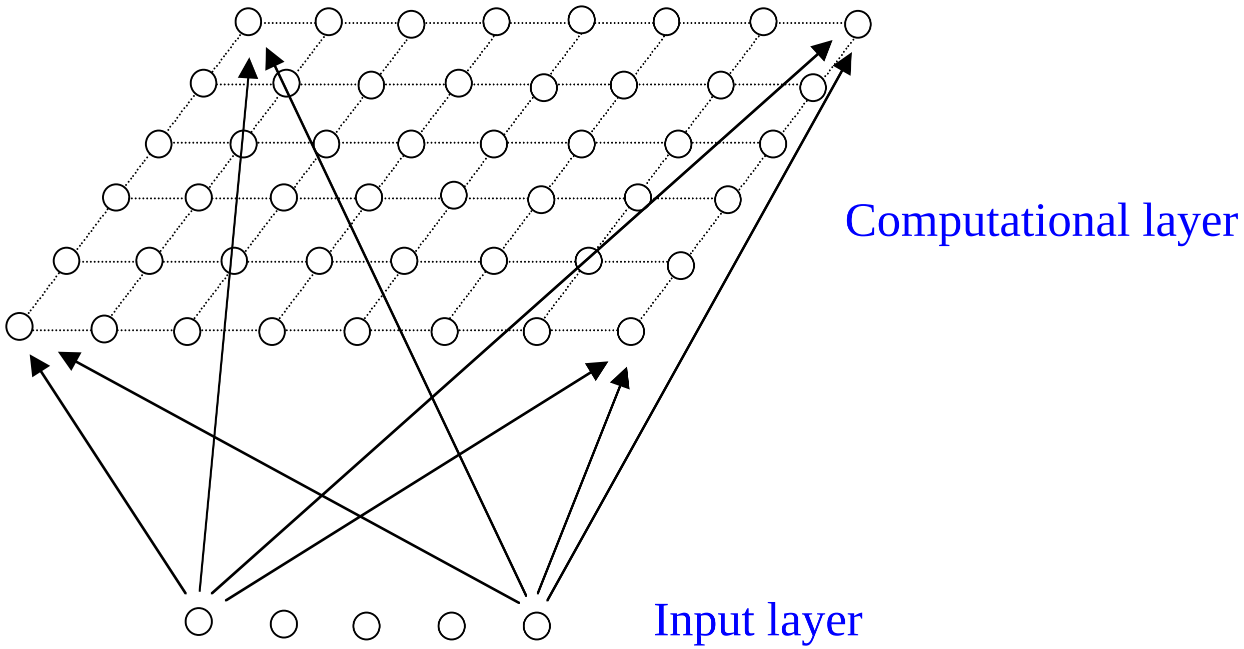
Self Organizing Maps: Algorithms and Applications

Introduction to Neural Networks : Lecture 17

1. The SOM Architecture and Algorithm
2. Properties of the Feature Map
   - Approximation of the Input Space
   - Topological Ordering
   - Density Matching
   - Feature Selection
3. Application: The Phonetic Typewriter

The Architecture a Self Organizing Map

We shall concentrate on the SOM system known as a ***Kohonen Network***. This has a feed-forward structure with a single computational layer of neurons arranged in rows and columns. Each neuron is fully connected to all the source units in the input layer:



A one dimensional map will just have a single row or column in the computational layer.

The SOM Algorithm

The aim is to learn a ***feature map*** from the spatially ***continuous input space***, in which our input vectors live, to the low dimensional spatially ***discrete output space*****,** which is formed by arranging the computational neurons into a grid.

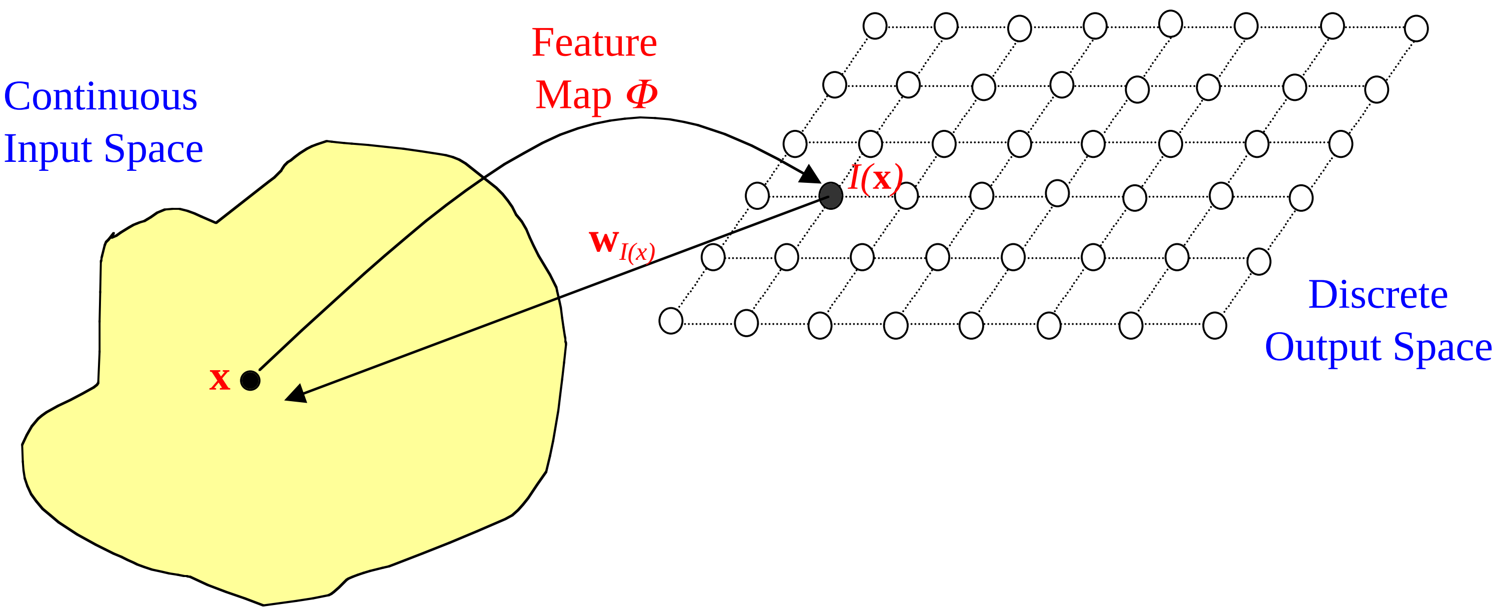
The stages of the SOM algorithm that achieves this can be summarised as follows:

1. ***Initialization*** – Choose random values for the initial weight vectors $\mathbf{w}_j$.
2. ***Sampling*** – Draw a sample training input vector $\mathbf{x}$ from the input space.
3. ***Matching*** – Find the winning neuron $I(\mathbf{x})$ that has weight vector closest to the input vector, i.e. the minimum value of $d_j(\mathbf{x}) = \sum_{i=1}^{D} (x_i - w_{ji})^2$.
4. ***Updating*** – Apply the weight update equation $\Delta w_{ji} = \eta(t)\, T_{j,I(\mathbf{x})}(t)\, (x_i - w_{ji})$ where $T_{j,I(\mathbf{x})}(t)$ is a Gaussian neighbourhood and $\eta(t)$ is the learning rate.
5. ***Continuation*** – keep returning to step 2 until the feature map stops changing.

We shall now explore the properties of the feature map and look at some examples.

Properties of the Feature Map

Once the SOM algorithm has converged, the feature map displays important statistical characteristics of the input space. Given an input vector $\mathbf{x}$, the feature map Φ provides a winning neuron $I(\mathbf{x})$ in the output space, and the weight vector $\mathbf{w}_{I(\mathbf{x})}$ provides the coordinates of the image of that neuron in the input space.



The feature map has four important properties that we shall look at in turn:

Property 1 : Approximation of the Input Space

The feature map Φ represented by the set of weight vectors {w_i} in the output space, provides a good approximation to the input space.

We can state the aim of the SOM as storing a large set of input vectors $\{\mathbf{x}\}$ by finding a smaller set of prototypes $\{\mathbf{w}_i\}$ so as to provide a good approximation to the original input space. The theoretical basis of this idea is rooted in ***vector quantization theory***, the motivation of which is dimensionality reduction or data compression.

In effect, the goodness of the approximation is given by the total squared distance

$$D = \sum_{\mathbf{x}} \left\| \mathbf{x} - \mathbf{w}_{I(\mathbf{x})} \right\|^2$$

which we wish to minimize. If we work through gradient descent style mathematics we do end up with the SOM weight update algorithm, which confirms that it is generating a good approximation to the input space.

Property 2 : Topological Ordering

The feature map Φ computed by the SOM algorithm is topologically ordered in the sense that the spatial location of a neuron in the output lattice/grid corresponds to a particular domain or feature of the input patterns.

The topological ordering property is a direct consequence of the weight update equation that forces the weight vector $\mathbf{w}_{I(\mathbf{x})}$ of the winning neuron $I(\mathbf{x})$ to move toward the input vector $\mathbf{x}$. The crucial factor is that the weight updates also move the weight vectors $\mathbf{w}_j$ of the closest neighbouring neurons j along with the winning neuron $I(\mathbf{x})$. Together these weight changes cause the whole output space to become appropriately ordered.

We can visualise the feature map Φ as an ***elastic or virtual net*** with a grid like topology. Each output node can be represented in the input space at coordinates given by their weights. Then if the neighbouring nodes in output space have their corresponding points in input space connected together, the resulting image of the output grid reveals directly the topological ordering at each stage of the network training.

Property 3 : Density Matching

The feature map Φ reflects variations in the statistics of the input distribution: regions in the input space from which the sample training vectors* x *are drawn with high probability of occurrence are mapped onto larger domains of the output space, and therefore with better resolution than regions of input space from which training vectors are drawn with low probability.

We need to relate the input vector probability distribution $p(\mathbf{x})$ to the ***magnification factor*** $m(\mathbf{x})$ of the feature map. Generally, for two dimensional feature maps the relation cannot be expressed as a simple function, but in one dimension we can show that

$$m(\mathbf{x}) \propto p^{2/3}(\mathbf{x})$$

So the SOM algorithm doesn't match the input density exactly, because of the power of 2/3 rather than 1. Computer simulations indicate similar approximate density matching in general, always with the low input density regions slightly over-represented.

Property 4 : Feature Selection

Given data from an input space with a non-linear distribution, the self organizing map is able to select a set of best features for approximating the underlying distribution.

This property is a natural culmination of properties 1 through 3.

Remember how Principal Component Analysis (PCA) is able to compute the input dimensions which carry the most variance in the training data. It does this by computing the eigenvector associated with the largest eigenvalue of the correlation matrix.

PCA is fine if the data really does form a line or plane in input space, but if the data forms a curved line or surface (e.g. a semi-circle), linear PCA is no good, but a SOM will overcome the approximation problem by virtue of its topological ordering property.

The SOM provides a discrete approximation of finding so-called ***principal curves*** or ***principal surfaces***, and may therefore be viewed as a non-linear generalization of PCA.

Application: The Phonetic Typewriter

One of the earliest and well known applications of the SOM is the phonetic typewriter of Kohonen. It is set in the field of speech recognition, and the problem is to classify phonemes in real time so that they could be used to drive a typewriter from dictation.

The real speech signals obviously needed pre-processing before being applied to the SOM. A combination of filtering and Fourier transforming of data sampled every 9.83 ms from spoken words provided a set of 16 dimensional spectral vectors. These formed the input space of the SOM, and the output space was an 8 by 12 grid of nodes.

Though the network was effectively trained on time-sliced speech waveforms, the output nodes became sensitised to phonemes and the relations between, because the network inputs were real speech signals which are naturally clustered around ideal phonemes. As a spoken word is processed, a path through output space maps out a phonetic transcription of the word. Some post-processing was required because phonemes are typically 40-400 ms long and span many time slices, but the system was surprisingly good at producing sensible strings of phonemes from real speech.

Overview and Reading

1. We began by reviewing the SOM architecture and algorithm.
2. We then looked at the important properties of the feature map: its ability to approximate the input space, the topological ordering that emerges, the matching of the input space probability densities, and the ability to select the best set of features for approximating the underlying input distribution.
3. We ended with a simple application – the phonetic typewriter.

Reading

1. Haykin: Sections 9.3, 9.4, 9.5, 9.6
2. Beale & Jackson: Sections 5.2, 5.3, 5.4, 5.5, 5.7
3. Gurney: Sections 8.1, 8.2, 8.3
4. Hertz, Krogh & Palmer: Sections 9.4, 9.5
5. Callan: Sections 3.1, 3.2, 3.3, 3.4, 3.5