[Music]

Welcome to the number six lecture of the six long six week the course in machine learning. The theme of this lecture is heavy learning an associative memory. So as you can see from the map of the lectures this week we have now left the main stream of work on artificial neural networks the feed-forward networks and recurrent neural networks and turn to another category of systems which can best be characterized by the term associative memory but we will first turn to something more specific which we call Hebbian learning. Hebbian learning theory is a neuroscientific theory claiming that an increase in such synaptic efficacy arises from a presynaptic cells repeated and persistent stimulation of a postsynaptic cell. This theory was introduced by Donald Hebb in his 1949 book the Organizational Behavior. The theories also called the Hebbs rule, Hebbs postulate or cell assembly theory. Hebb expressed himself as follows, let us assume that the persistence or repetition of a reverberator activity tends to induce lasting cellular changes that add to instability. When an axon of cell A is nearly enough to excite a cell B and repeatedly or persistently takes part in firing in some growth process or metabolic change takes place in one or both cells such that the efficiency of A as one of the cells firing B is increased. Actually as it turns out neuron A has to fire slightly before B are not fully in parallel to manifest the causality relation. So this elaboration of Hebbs's work is called spike timing-dependent plasticity. What is Hebbian Learning in an Artificial Neural network? The theory is often summarized as cells that “fire together wire together”. It's an attempt to explain synaptic plasticity the adaption of brain urines during the learning process. In an ANN setting the plasticity is implemented through adaption of weights. So Hebbs law can be represented in the form of two rules. If two neurons on either side of a connection synapse or activated synchronously then the weight of that connection is increased. If two neurons of either side of a connection read synapse are activated asynchronously then the weight of that connection is decreased. So Hebbs law provides the basis for unsupervised learning. Learning here is a local phenomena occurring without any feedback from the environment. Let's turn now to what we call the Hebbian learning algorithm. So first there's always an initialization so the synaptic weights and the threshold is said to small random values in the interval 0 to 1. Then we have this step of activations where we compute the postsynaptic neuron output from the presynaptic input elements, denoted here as Xij from the data item Xj and j here denotes the number of the training instance data item considered, while i is one of the elements of the this input vector. So actually the output from the postsynaptic neuron Yj is 1, if the sum of the input times the weight on the input connection minus T where T is the threshold is larger or equal to zero, if not Yj is zero. So after that that the output value of the postsynaptic neuron is calculated, one can go to step three which is the learning phase and the learning follows that something called the activity product rule which captures the essence of the Hebbian theory. So here we update the weights in the network and the weight correction is turned by this so-called activity product rule, which says that the new weight that is going to apply the in iteration J plus 1 is equal to the old plus a term which is alpha which is a learning rate parameter, as already comment in a kind of damping factor that can be between 0 & 1, multiplied by the new calculated value of the post synaptic neuron times the input value from the presynaptic input element. And after that we start again we use a new data item and the same procedure is repeated.

Let us now look at an example of this algorithm. So we have a neuron for pre synaptic inputs x1 to x4. We have weights on the connections for those inputs, we have an output from the postsynaptic neuron Y, we assume a threshold of 2 we assume a learning rate of 1 and we initiate all the weights to uniformly to all of them to one and then we will now look at two training instances one zero one zero one zero one zero and look at how the calculations are made. As you can see here we have instantiated the inputs to the first elements of the first training instance, we have the predefined weights what we do now is this first installation we call zero and then we have a learning rate of one and we have a threshold of two, so the first thing is we do is we use the first formula for Y and create the sum, so as you see here what we do is a sum the input values with the appropriate weight values and then we subtract with the threshold nearly gives zero but as this fulfills the first criteria, we can output one. So that's the first step. So now we have an output value. So then we are in the position that we can calculate the weight update, so we calculate the Delta weight which is the actually the learning rate times the value new value of Y times the input values on the various input variables. So as you can see this gives us 1 0 1 0 and if we add that to the old weight we get the new weights 2 1 2 1. As the second data item for the second iteration is identical to the first, the calculation looks very very very similar. We still get an output a 1 we update the weights but what we can see and I think it is the only interesting observation on this slide, is that in the cases where we have a synchronous activation, both of the presynaptic and the postsynaptic equal to 1 in this case. Then we further strengthen the weight of that connection, which is actually consistent with the hebbian theory. And I want to say a few words about associative memory. In psychology associative memory is defined as the ability to learn and remember the relationship between unrelated items. This could include for example remembering the name of someone or the aroma of a particular perfume or any other sensory impression. Associated memories declarative memory structure and often episodically based. A normal associative memory tasks involve the testing processes on the recall of pairs of unrelated items such as face-name, pairs but in the realm of human psychology associative memory is obviously a very wide term. When we turn to artificial intelligence and machine learning the term gets more precise. So in these two areas associative memory refers to a broad class of memory structures with mechanisms for storage and recall that can handle general patterns and pattern matching. Theoretically all kinds of structures and data types should be able to be handled in the same system. There is also a clear coupling to the area of content addressable memory so called CAM techniques which is a more classical core area of computer science. So for a associative memory in the computer science setting on the one end of the spectrum there are memorization of specific objective situations, and recall of these based on detail but still partial or noisy descriptions. In the other end of the spectrum there are analogical reasoning where structurally similar but domain unrelated patterns can be recalled. Domain ways you can get something back that is from a different route but it's still in some structural way similar to your query so to say. In the in the middle there are case based reasoning where separate patterns can trigger recall of larger patterns. So central concepts in associative memory are similarity measures, spatial or temporal, another technological aspects of the pattern space like valleys, hills, basins. Optimality criterion aspects of the search page like local minima maxima attractors etc. Ideally want to say that one wants an associative memory system that has as many stable well separated local business as well as memories to store.

So we have two forms of associative memory. One is called a Auto associative memory and in Auto associative memory could also be called auto association memory or auto association networks. It's any type of memory that enables one to retrieve a more complete object descriptions from a partial description. So in more technical terms the input and output factors have exactly the same form so now obviously that Xi and Yi has the same form for the vectors X and Y. So as you see an example to the right we have the partial descriptions of a particular object but there are details missing or a noise, but what you can retrieve is the full description or full picture of that object. So these concrete examples are typical or restoration of imagery like this one or the restoration of speech fragments.

The other category of associative memory is called Hetero associative memory so here on the other hand man can retrieve not only object description on the same form, but potentially also wider range of patterns still satisfying some measure of similarity with respect to a partial description. So in terms of vectors the input vector X and the output vector Y can have very different forms. So as a an example related to the above as you we have as input a key and what we can retrieve is the situation around the application of a key, which is actually part of a door and a lock what were the key is applied. Let me say a few words about some key concepts here. I will talk about three things mainly something called Attractors, something called Basin's and something called Bifurcations so an attractive attractor is a state toward which other states in the region evolve in time similarly each attractor has a basin which is a surrounding region in state space so the all trajectories starting in that region end up in the attractor sooner or later so you can see that illustrated in three ways to the right three different kind of images that illustrate that fact the basses belong to different attractors are separated by a narrow boundary which can have a very irregular shape the appearance of such a boundary separator is called before occasion for initial positions close to the boundary small fluctuations can push the system either into the one or into the other basin and therefore either finally into either the one or the other attractor so close to the boundary the system beams chaotically but inside the bath and it moves predictably towards its attractive ideally as already been said one wants an associative memory system that has many well separated basil's as one have one has memories to store so this was the end of the this lecture thanks for your attention the next lecture six point seven will be on the topic octal networks and Boltzmann machines thank you goodbye