Machine Learning,ML

Prof. Carl Gustaf Jansson
Prof. Henrik Boström
Prof. Fredrik Kilander
Department of Computer Science and Engineering
KTH Royal Institute of Technology, Sweden

Lecture 14
Artificial Neural Networks

So welcome to the fourth lecture of the third week of this course on machine learning so we will talk about artificial neural networks and their representation. So not official neural network abbreviated ANN is a network of nodes or units commonly called Artificial Neurons connected by edges the corresponding graph is directed and Edges typically have a weight that can be adjusted the weight increases or decreases the strength of the connection. Artificial neurons also have a thresholds that the signals is only sent from the neuron if the aggregate signal from all input edges crosses that threshold. But furthermore before the signal leaves the neuron the state of the neuron is computed from the input it's also transformed by a nonlinear function, so furthermore Artificial Neurons words are aggregated into layers. So typically units are sorted into layers and there are many layers, and the layers may have different functions in the computation and signals travels from the first layer the input layer to the last layer which is normally consider the output possibly after traversing the layers several times. So hidden layer may have loops which is one variant of that is called Recurrent Neural Networks.

So of course the inspiration for this computational model that was online in the last slide is inspired by neuroscience by the way we believe neurons systems work in humans, in animals and so on the difference is this course while the artificial neurons are digital in nature and real neuron is an electrically excitable cell that receives processes and transmits information through a combination of electrical and chemical silent psychic signals. So the real new neuron is a very complex machinery in contrast to the rather simplified artificial Neuron. So a real neuron consists of a cell body called the soma and which is the body although and then you have the input side so you have the input side which called the dendrites. So the dendrites receive the signals from other neurons, and on the output side you have something called an axon and an axon is the output organ so to say from the Neuron. All of these structures are pretty branched and much more complex than in the artificial case and so an axon can also be pretty long so this means that the axon can reach a considerable distance to the next cell to be a neural cell to be affected. But the basic functionality is signals come in via the dendrites are kind of transformed, handle by the soma and then our output via axon. That's the basic neurological model. So in contrast to the artificial that systems that still are a reasonable size the real neuron systems are huge and are in a huge scale processing in parallel so to give you a figure the Human brain is assumed to have in the order of 100 billion neurons and a huge

number of connections so actual typical figure here for the number of connection is 10,000 times the number of the of the neurons itself and then you can you must imagine that the scale of this parallel activity. Let's now be a little more concrete concerning the core components of this kind of representation so let's look again at how we view the architecture of a neural network unit .So the unit is supposed to have a number of inputs actually in the in the real case in the real neural world we talk about synapses and dendrite and so on. In the division case we have simply a number of inputs and these inputs are of course handling signals coming from other units, every such input every such edge in the neural network is given a weight and these weights key parameters for the productivity and the performance of the networks. What happened in the cell body is when receiving the inputs the inputs there is a summation performed where the inputs are the weighted sum of the inputs using the weights of each input channel is performed. Then also a cell body is attributed with a threshold and the negative value the negation of that threshold always sometimes called bias, so essentially what we do is that we sum the negation of the threshold with a weighted sum of the input and then if that value is larger than zero then the unit is ready to what you can say quotation mark "fire" which means to send out an output signal but before that output signal is sent out, the output value is transformed by applying something called an activation function. which in a way modify the output value. And essentially in all these cases we normally talk about numerical values.

So as the functionality of ANN unit is very important for the understanding of the whole representation scheme that is repeated. So if we have an output of unit "i" let's call it "ai", this normally becomes the input to another unit "j". So I is considered the predecessor of "j" and "j" the successor of "I". Each connection is assigned a weight in this case "wij". Each node has an activation treasure the negation of the treasure is termed bias "b" so for node "j" we call it "bj" .In the body of the unit the weighted inputs are summed together with the bias, which means that we say Sum wij times ai plus bj.

$¿\sum wij*ai+bj$

If this sum is greater than 0 the output of unit j is calculated as a j is a function of the sum where the function is a local activations. So this is the criteria for firing which means that outputting a single from the unit and that the weighted sum plus the bias is larger than zero.

$aj=f\left(\sum wij*ai+bj\right)$

So let's turn a minute to the so-called activation function there are a number of those actually and it was you see on this slide a couple of examples essentially obviously that you see that there are two categories here so on the topmost layer here of examples you'll find step functions, you'll find something called a sigmoid function you will find something called a hyperbolic tangent function in all of these cases the output level is maximized somehow. So there is a there is a certain segment and when the output value is controlled by the function but then it reaches a plateau. In the other category the lower part here you see that the output is linear so the larger the original output from what value from the unities the larger that the net output becomes so these are our two of functions. So now let's talk a little about the structure given that we have delved into the functionality of the unit, so also the structures of adjacent neural networks to somehow reflect our reflected in the history of the development of the area so as I mentioned already in the first week of this course one of the key contribution as early as 1957 was the Perceptron but essentially the Perceptron is a very very simple network with only an input layer and an output layer. Still the unit functionality is in the same genre as described here but in this case we only had two layers. And as you also may remember from the first week this kind of simple structure was immediately successful but soon proved very unpractical which meant that there was being a gap here from 1957 to almost 1985 when this area had revival again and actually in 1985 the work introduced what is called hidden layers, so while the pairs of journal only had an input layer and output layer in in the early work of the 1980s Hidden layers were introduced and we will use the word deep learning later, and in a way all neural networks that has more than two layers are considered deep, however in the early work of 1980s and when we maybe had one hidden layer or two hidden layers that term was not in wide use. So the introduction of the layers in early nineties represented a major change and maybe you remember from what we discussed earlier here on Basiyan networks there's a parallel you need an input layer it is when you are in which you have to decode your input data you need an output layer by in a way can harvest the results but you also need intermediate computing elements and this immediate computing elements are the hidden layer. So then of course over time when this technology developed and also computational resources became more available for this purpose, and the possibility reintroduce many more levels ok so what's now talked a lot about and it's become very successful yes what is now called the deep neural network but essentially the word doesn't mean much more than that there is a system with very many levels actually.

And so finally and there is a slide here on different versions on neural network you can see the you can see the Perceptron, you can see the feed-forward maker which had a Hidden layer but not many hidden layers. What is interesting is in the middle of this figure because there is a problem with networks that only are direct in a forward manner is that it's very difficult to represent sequences of states and also to design a memory function so this means that we one need to introduce loose (16:33) in the network to handle that.

So let's now look at how problem-solving is carried out for this kind of representation so you hopefully now understood the basic functionality of the unit you have also basically understood the structural properties of this representation. So when we have a problem we - as always define a set of features to express our example as you learn away and then when we have expressed all our examples in the certain kind of features we have to if want to use a artificial neural networks representation we have to map the features onto the units in the input level and depending on exactly how that looks it becomes more or less complicated because we should understand that that the neural networks are purely numerical so the but is an input level is a discrete set of numerical items so whatever input we have we have to map that into this discrete set of numerical values. And as I said if we want to handle sequences and not only a single level of input that has to be handled by a special kind of networks with internal loops but we will come back to that later. Yeah in the same way we also have to make the same kind of modeling for the output layer because of course in the output layer is where we want to harvest our results and so the output from the output layers have to be decoded back in into the original feature representations. And all the amount goes for any kind of problems on it doesn't matter whether we try to do a classification of something, we have a regression task or if the network are supposed to generate output actions in some system. Also all neural network computation schemes have many so called hyper parameters that control the detail behaviour and I mean just taking examples selecting exactly which output function you should have been you know and so there are many of these like internal parameters that guides the artificial neural network machinery, that typically need to be adjusted to facilitate the problem at hand but then also typically the basic the basic flow of computation is forward feeding machinery coming and going from starting from input layer going through the hidden layers and ending up in the last output layer.

So on the following three slides I just show you a few examples these examples are pretty standard ones there are no examples with the Perceptron, they're all examples from on the use of networks with you with one hidden layer as it depicted in these cases and I think the important message for all these examples is simply that the basic analysis around the domain and the features you have to consider is not different when using an artificial neural network ,you still have to analyze domain, you still have to choose the relevant feature set and this goes for any learning approach. What you have to do here is you have to find of encoding features in any reasonable way into the input layer and of course the simplest case is that all your features are all you know the zero one and then you can just have one input node for each of these or you can have a numerical value simply a numerical value share something more complicated there is another mapping process.

So the last example and I want to show you has a different character so essentially this is an example where and the input is not a well-ordered digital feature set but the input are images so this means that if you want to give such a task to this kind of network it's not it has to solve two kinds of tasks first of all it have to solve the original task the one where the input is is well-engineered in a digital form and when you can do a classification task you can do regression tasks as well, in this case you the network have to do that but also to analyse, transform the image into the digital form in a sensible way. So for this kind of network the network not only do learning in the form we have discussed upto now but also that it has to perform an image recognition task. So this means typically that if you want a network to do all these things it have to be much deeper they have to be much more room for internal computation in this kind of network and also for the first image recognition part the network typically have to be specially engineered and a special property structural properties to manage that task but that we will come back to in a later week. So essentially in this example illustrates the case where your input these images but you still want down out get something semantically meaningful out of the system which is this case is the name of a lady of the image.

So let's turn to now what learning means in this kind of representation yeah so as for the Bayesian networks there are one can say two cases ,so the low-hanging fruits here than very natural learning mechanisms are the updating of the weights of the edges with all the key parameter of this kind of system but also for example the thresholds of nodes that could be other small things too but the weights of the edges and the channels of the nodes are the key parts the key parameters that can be updated and where learning can take place. We will not delve into the learning mechanism this point but one of the most earlier well-known approaches is an approach where the outcome the result the output from the network is reviewed externally and after that review feedback into the system and then the learning machinery is such that the feedback that would be fed back into the system is analyzed in such a way that credit and blame can be given to specific connections and the ways of these connections can be increased or decreased depending on who was to blame or who was to give be giving credit. So this is one example of how learning can take place but there are many options here and we will come back to those, if we leave that is of course also possible to change the network in more dramatic ways so while in the first case the whole structure or the network is supposed to be static and only the parameters are supposed to be chang, while we also can consider the case where we can dynamically or update the network with change the connections in the network take away connections, add connection but also introduce new nodes and new levels which is of course the most advanced. Finally which is not mentioned in the slide but I will say now that also various attempts also to actually learn the kind of the feature sets that are actively used so learning the selection of features is would say also a possibility within category 2 here. So this is the end of this lecture thank you for your attention, the next lecture this week will be on the topic genetic algorithms thank you bye