Algorithms for Protein Modelling and Engineering
Professor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 53
Machine Learning to Predict the Secondary Structure from Amino Acid Sequences

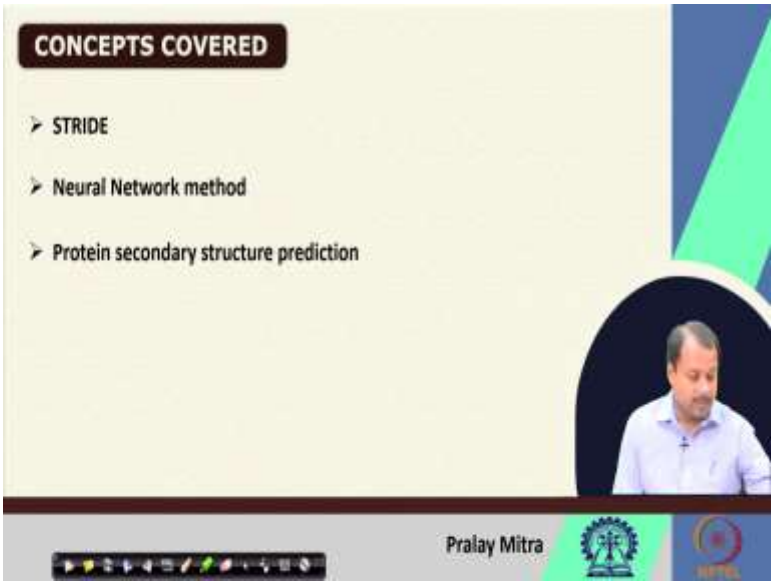
Welcome back. So, we are continuing with this hydrogen bond calculation and what is the role of that hydrogen bond calculation in the secondary structure and from there when the input is the protein atomic level information or the coordinate of the atoms. So, then how do I decide the existence of the hydrogen bond and that we can extend for the checking that whether that particular hydrogen bond leads to formation of some helix, sheet or coil or not that we discussed.

In this context, we also discussed the DSSP Dictionary of Secondary Structure of Proteins that we discussed and we mentioned in the last lecture that one is too meaning that is tried that we will discuss and after that one we will move on to this topic. So, the predicting the secondary structure of from amino acid sequence, or when the protein sequence will be given as an input then how to predict what is the secondary structure.

As you can understand that directly calculating a calculation of the hydrogen bond is not feasible for that one, so we have to come up with some alternative idea and that and in that we will see what is the role of the machine learning or what machine learning can help us to get it. Again, when the hydrogen bond cannot be calculated and we are looking for the secondary structure I mentioned that I will be using a separate term for this that is predicting the secondary structure.

Last time it was assignment of the secondary structure, this time it is predicting the secondary structure.

(Refer Slide Time: 01:55)



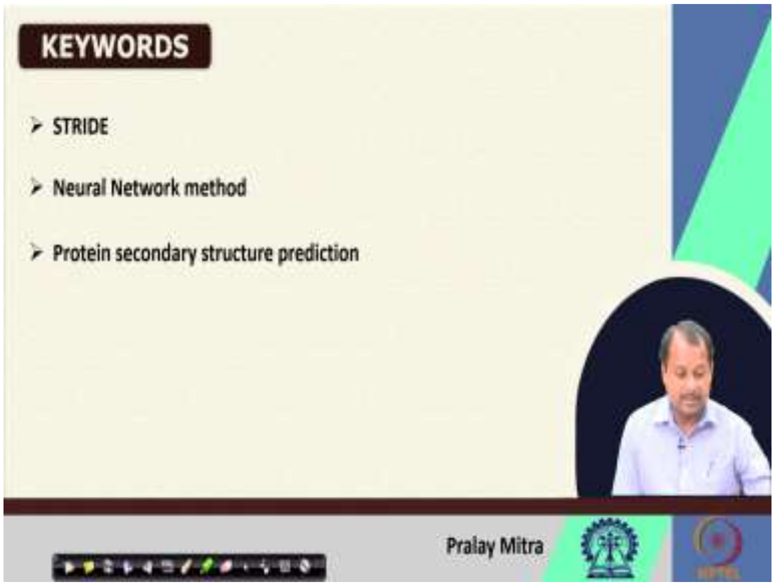
CONCEPTS COVERED

- STRIDE
- Neural Network method
- Protein secondary structure prediction

Pralay Mitra

So, the concept we will continue the stride that is actually assigning the secondary structure from the protein coordinating formation that we are continuing from the last lecture. Next, we will move on to one neural network technique that is one machine learning technique, for predicting the secondary structure from the protein sequence.

(Refer Slide Time: 02:16)



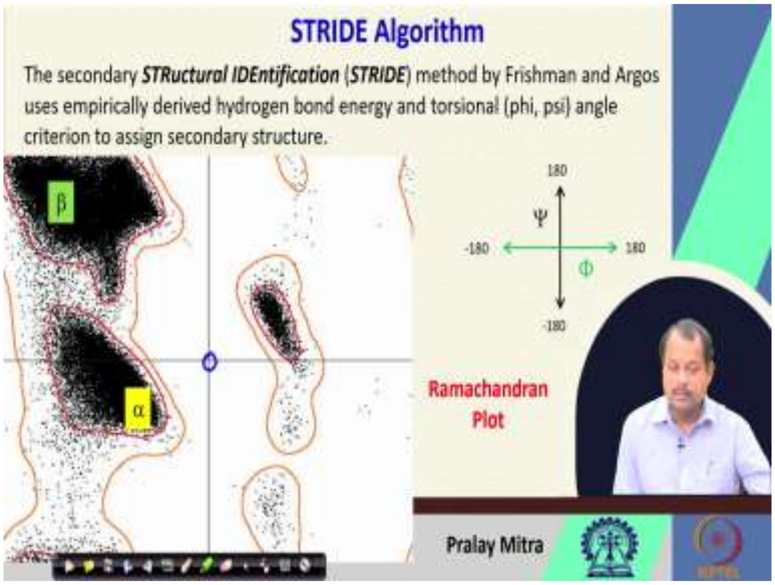
KEYWORDS

- STRIDE
- Neural Network method
- Protein secondary structure prediction

Pralay Mitra

Accordingly the keywords we have picked stride and neural network method, protein secondary structure.

(Refer Slide Time: 02:21)



STRIDE Algorithm

The secondary *STRuctural IDEntification* (*STRIDE*) method by Frishman and Argos uses empirically derived hydrogen bond energy and torsional (phi, psi) angle criterion to assign secondary structure.

Ramachandran Plot

Pralay Mitra

So, stride algorithm what they have observed? So, DSSP is fine and they are exploiting the hydrogen bond formation perfect, no problem, and also they are not considering any hydrogen bond calculation because as you know that when we discussed at the beginning of this week, the hydrogen bond formation the perfect or ideal situation was when there is 180 degree angle, so oxygen, hydrogen and say oxygen, so in the water molecule, so when there is a 180 degree angle.

But theoretically it is 180 degree, practically that may not be 180 degree. So, how much flexibility we can incorporate? So, based upon some model that we have discussed, it says that greater the 90 degree is fine. But that is too much deviation from 180 degree, sometimes people consider, specifically the stride consider that it is greater than 120 degree, and if it is greater than one degree, what is the effect of the distance.

So, whether the hydrogen atom and the oxygen with which it is forming the hydrogen bond they are placed close enough with each other or they are distantly placed. So, the distance and angle based upon that one as you know that this hydrogen bond is not a coherent bond I is basically the non-covalent bonding, so some sort of changes will be there.

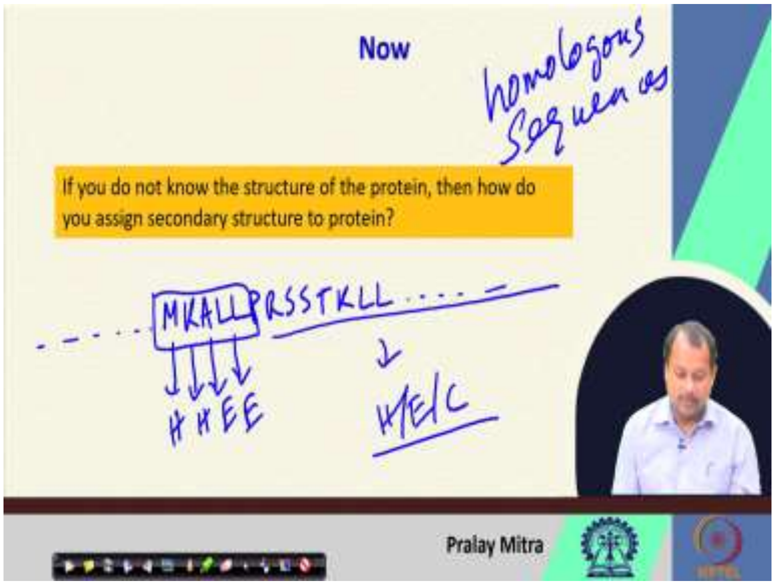
So, DSSP perform some columbic approximation based upon that one they are kind of binary in nature, so they will either consider that is a hydrogen bond or will not consider. But is it possible that instead of 0 and 1 you can incorporate that what is the total contribution, maybe less, maybe more, but what is the contribution I will conclude?

In if I do that one then greater than entity is fine. That is one thing, another thing stride is developed is based upon their observation that if I look at this Ramachandran plot this slide you have seen in the introductory class, on the left hand side you have the Ramachandran plot where along the x axis phi along the y axis psi both varies from minus 180 degree to plus at 180 degrees. So, the origin is her, this is my origin, the blue circle indicates my origin.

So, in this way, what is the region and the region is also marked as alpha and beta. So, is it possible apart from the secondary structure to exploit this alpha and beta also to improve the accuracy? The exactly that thing is done by the stride. So, what they used to do the secondary structure or structural identification in short stride?

The name of the algorithm method by Frishman and Argos uses empirically derived hydrogen bond energy and torsional angle criteria to assign secondary structure. So, this is the Ramachandran plot, there result indicates that for the same dataset probably a DSSP used for validation of their algorithm their performance is marginally good and for better us, so you can use also the stride instead of DSSP.

(Refer Slide Time: 05:42)



Now

If you do not know the structure of the protein, then how do you assign secondary structure to protein?

Pralay Mitra

Now, if you do not know the structure of the protein then how to do how do you assign secondary structure to protein. So, that is what we are going to discuss. So, when you do not know the structure of the protein which means that you are given only the sequence say then how can I do that one? So, first of all although I mentioned and definitely we will go for some machine learning technique like neural network method, but what is our observation regarding the problem.

So, the regarding the problem our observation is that we do not have anything, it is just a sequence MKALLPQPRSS, so corresponding to that our intention is that whether M will be helix, helix, sheet, sheet, so basically this sequence this sequence we need to map to either H, or E, or C corresponding to each amino acid I have to mention whether it is H, or E, or C.

So, for 20 different amino acids I have to map it to three of the states, either H, or E, or C that is my problem actually. Now, in this problem statement the challenge or issue is that we do not have any information, then what we can do? If I asked you that okay, so this say when I say as the secondary structure then I am specifically concerning about a small stretch, and its environment in the structure.

If say, there exists one structure corresponding to this sequence, then in (())(08:06) of space, it has some environment, and also it has some sequential environment. So, sequential and structural environment is there, as per this sequence, now how do I encode that information for my own purpose? For that the observation also indicates that for this sequence let us see how many homologous sequences are there.

Similar, to the situation we discussed in our protein design, where input was a protein structure and in order to know at each position what are the probability of the mutation of the 19 different other amino acid, then we took help of the homologous structures and from there we identify, so these are the homologous structures and from these homologous structures at say ith position, so these are the possibilities.

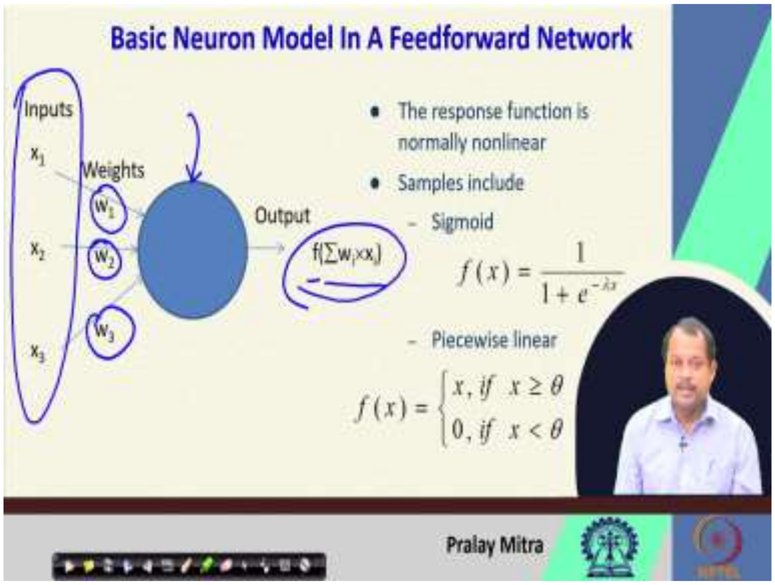
So, these seduce can occur, so based upon that basically we infer that what are what is the probability and how do I it. So, similarly here is sequences and input, so what we can do, we can look for homologous sequences. Now, from that homologous sequences, since the homologous sequences are there, then we know that correspond and that homologous sequences are there, and if we extract that homologous sequence from say protein databank, then corresponding to that homologous sequence, there is a structure.

And corresponding to that structure secondary structure is also reported there, either using DSSP or stride or using any other method. And you will agree that assigning the secondary structure is easier compared to predicting the secondary structure. I mean given the atomic level coordinate determining what is that secondary structure is easier compared to this given the sequence you need to predict the secondary structure.

So, first thing we will do, we will identify the homologous sequences, from that protein databank. So, we know what are the secondary structure for all those sequences, in one side there homologous with this particular given sequence in another side we know their structure, so based upon that information is it possible for us to infer that what will be the secondary structure for our own purpose. So, that thing we need to do.

So, in order to do that one, let us briefly discuss about the neural network method. So, this is there is no scope of going into details of the neural network just briefly we will touch upon, if you know this then it will be referenced for you, if you do not know then this much information will be enough for you to go and implement the algorithm we will be discussing for predicting the secondary structure from the protein sequence.

(Refer Slide Time: 11:14)



Basic Neuron Model In A Feedforward Network

Inputs

x_1

x_2

x_3

Weights

w_1

w_2

w_3

Output

$f(\sum w_i x x_i)$

- The response function is normally nonlinear
- Samples include
  - Sigmoid

$$f(x) = \frac{1}{1+e^{-\lambda x}}$$

  - Piecewise linear

$$f(x) = \begin{cases} x, \text{if } x \geq \theta \\ 0, \text{if } x < \theta \end{cases}$$

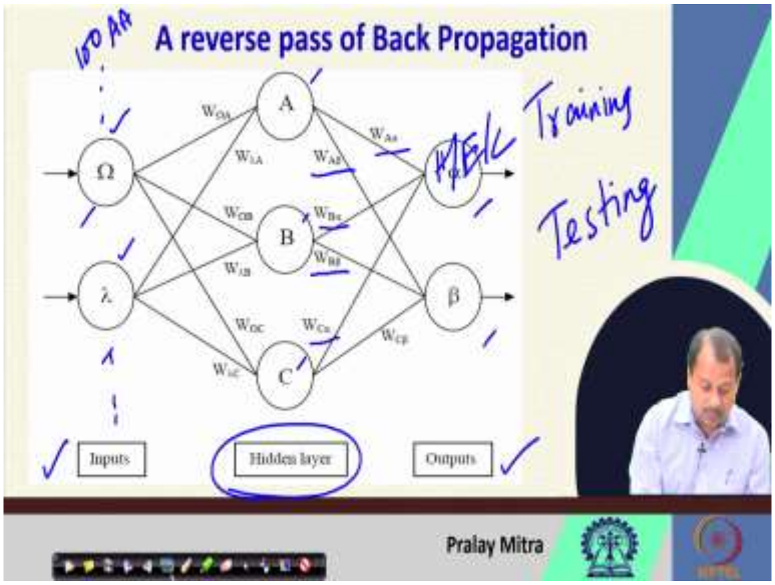
Pralay Mitra

So, this is a basic neural model in a feed forward network. So, what is the input? So, this is my input. So, there are three inputs, X1, X2 and X3 and they are ease and weight corresponding to each input W1, W2, W3, then when it will be feed to some network, it will output a function f which is a summation over Wi multiplied with Xi.

So, each input will be weighted by the weight value assigned there, and taking the sum or the linear combination you will get the output function. Now, regarding this output function, there are few choices. So, the response function is normally nonlinear in nature. So, it can be either sigmoid or piecewise linear.

So, usually the sigmoid function which is a fx equals to 1 upon 1 divided by 1 plus e to the power minus lambda x, is basically more popular compared to the piecewise linear. Where you need to determine or decide on threshold say theta and if your x value is greater than theta or greater than equal to theta, then x will go directly, if it is not then it is 0, which means it is piecewise linear or kind of a thresholding. But sigmoid is most widely used.

(Refer Slide Time: 12:51)



A reverse pass of Back Propagation

So, here is the reverse pass of back propagation and actually this back propagation is going to be useful for our whole purpose. So, what are the different components? So, previously in the basic neuron, we observed that there are inputs, there is a weight value attached with the input and then that input along with the weight value is map to some output function. In this case, one extra layer has been incorporated that is called as the hidden layer.

So, there is definitely input and there are outputs. So, previously there was only one function in this case I can assume that there are two functions say alpha and beta, and input is sigma and lambda. Now, in the hidden layer I will have three different nodes A, B, C. Now, why is three how to decide about the three that basically depends upon the problem you are trying to solve.

In this case, we are considering that three nodes are there at the hidden layer, of also how many hidden layers will be there, so that is also problem statement specific or depends upon that problem statement. Now, if I assume that there is only one hidden layer and there are three nodes like A, B and C, then following the concept of the basic neuron model that I have shown you on the last slide, so this input can be weighted and will go to one node like this A.

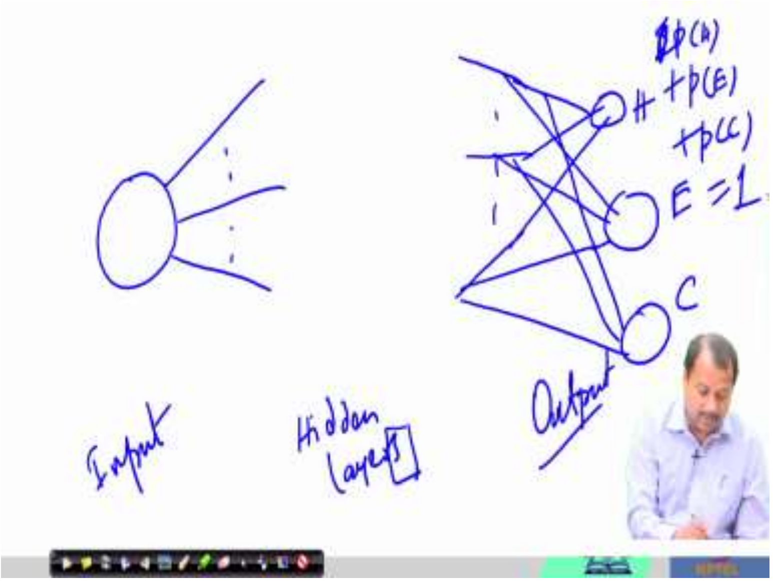
So, let us assume since it is going to be going from omega to lambda, so it is W omega A, then it is omega B, then it is omega C, like that where it is lambda, so lambda A going to A, lambda B weight going to B, and lambda C going to C. Now, if I assume that so this sigma then this lambda and sigma is not there, you suppress that input, you assume that hidden layer as it is my input.

If it is so, then from here to the output you can map again with some weight value, so A to alpha you are going, so WA alpha, A to beta you are going so WA beta, B to alpha you are going so WB alpha, B to beta you are going so WB beta, C to alpha you are going WC alpha, C to beta you are going so WC beta. Now, we would job will be to determine these weight values all W.

Because you know the input, you need to know all the W. While I am doing that one, so there will be two steps, one is called as the training, another testing. Now, you try to understand this situation that in all the cases input is known to you, so our intention is to say given one protein sequence we need to predict that what will be the secondary structure.

So, for us corresponding to each amino acid you can consider that as if there is all there are say this is extended in both the sides and when it is extended then say 100 amino acids are there. So, corresponding to each some node is there. So, this may not be one node, so I will go to that detail, but for the timing if I assume corresponding to each amino acid there is only one node. So, sigma is representing one amino acid, lambda is representing one amino acid. Now, on the right hand side corresponding to each node actually this alpha will be either H, or E, or C.

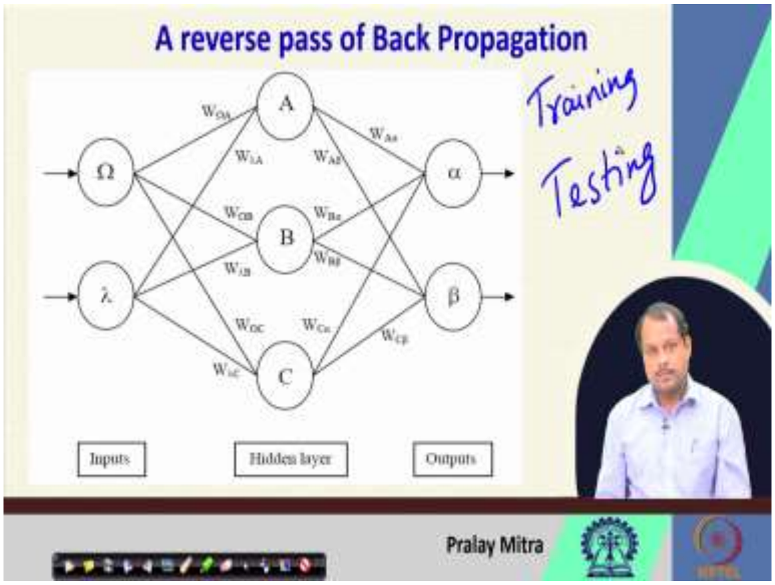
(Refer Slide Time: 17:13)



Now, alternatively what I can think is that basically one amino acid is given to you, definitely there will be some hidden layers, this is my input and this is my output, I am not really discussing about how many hidden layers are there, how many hidden nodes are there, but my point is when it will go then it will go to either H, or E, or C.

Or else what I can think that, it will map to all three of them say H, E and C with some probability value, where the summation of the probability say p of H plus, so some summation p of H plus p of E plus p of C is going to be 1, that is also fine, that is the use of that network that we are discussing for our problem, detail I will go again later.

(Refer Slide Time: 18:39)



A reverse pass of Back Propagation

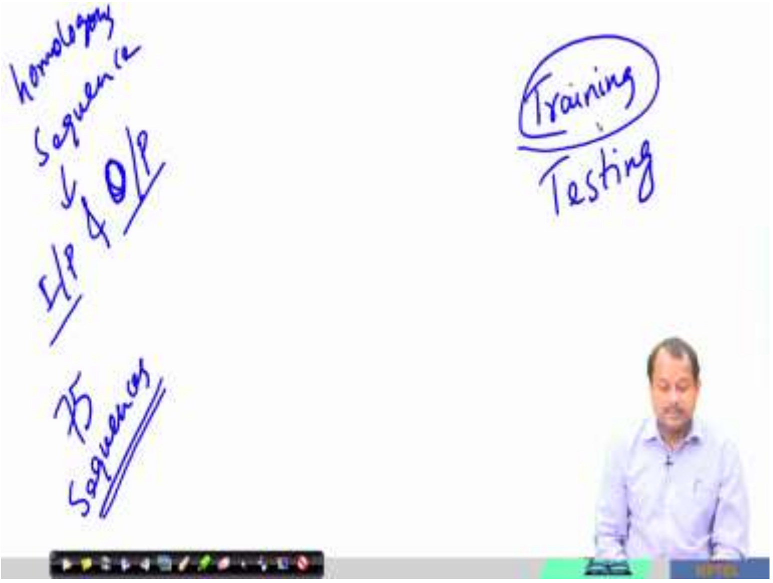
Training

Testing

Now, if this is given to you, then as I was talking about that there are two things, one is called testing and training. Now, while I discussed regarding the homologous sequences, then what I said? Input is one protein sequence, after having your input protein sequence, what you did? So, you perform some operation which will give me homologous sequences from the protein structure protein databank which contains the structure as well as the sequence.

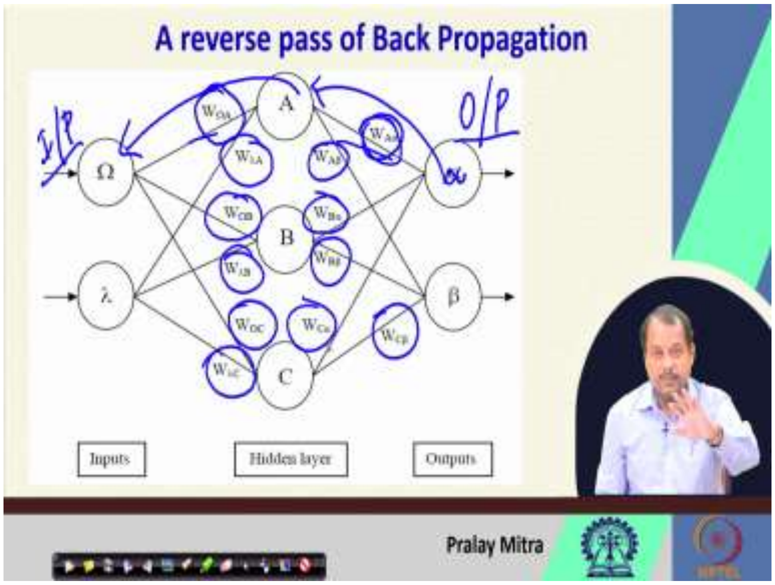
So, with the sequence I will perform the say multiple pairwise sequence alignment using the dynamic programming that we discussed, and then we compute what is the score value based upon that we will decide whether considers that as a homologous sequence or not. If I consider that as a homologous sequence then corresponding to one homologous sequence, I have input and output both.

(Refer Slide Time: 19:51)



So, for homologous sequence I have input and output both. Now, corresponding to each such homologous sequence I will have input and output. Now, for one protein sequence if I searched the total protein databank and without any loss of generality let us assume that I got say 75 such sequences, now I can use that sequence to train my system, which system? Now you come back.

(Refer Slide Time: 20:45)


A reverse pass of Back Propagation

So, train my system, what I will do? So, that 75 homologous sequences for that I have input and I have output. Now, from this particular network what is unknown to you is the weight functions or weight values. Now, if you know input and output, so first of all, so assuming that what is my alpha, then I can write an equation which will give me A and from here I can go here.

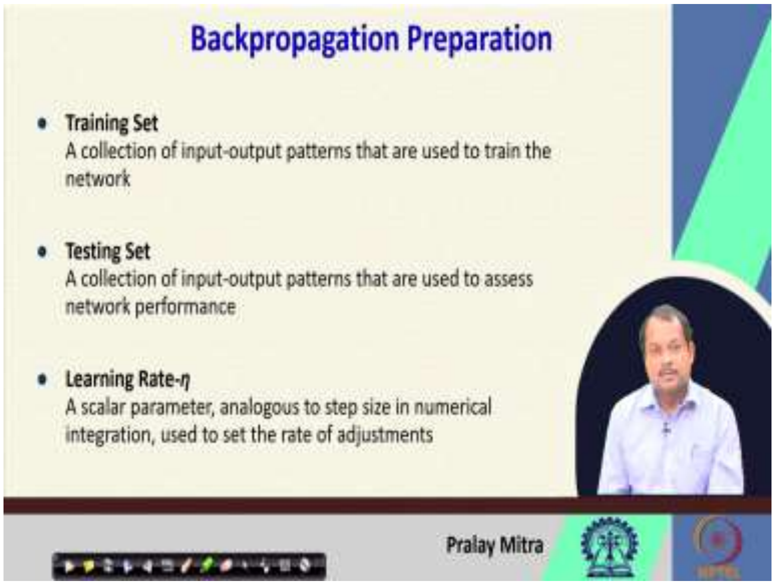
Now, if I have input and output then I will have from the set of equations using that set 75 homologous sequences I can solve them and what I can have, I can have one weight function corresponding to this, once I will have the wait function then you see your life is easy. Why it is easy? Now, so I have the known protein sequences for which secondary structures are also known with say 100 percent accuracy, it will not be 100 percent but let us assume with very high accuracy.

Now, if it is so, then all the weight functions are known. Now, you see if you are given one input then from that input which is lambda and sigma from there using that weight function you can go to A, B, C, then from A, B, C using other weight functions you can map to alpha and beta that is what is required. So, given one input sequence you move to A, B, C the hidden layer then from there you go to alpha beta, because weight functions are known to you.

So, this is the basic structure of reverse pass of back propagation algorithm, only thing you need to decide that what will be the size of the input, what will be the size of the output, how many hidden layers, and each layer how many hidden nodes will be there. Accordingly, how many weight functions will be there you can decide.

Now, you think that, so corresponding to one amino acid sequence I say identified the homologous sequences and then I computed the weight functions that is kind of I train the model and one model I computed and deposited. Next, when a new sequence will come then the problem will (())(23:29) to compute or to prepare the input from here and then multiply with the weight functions in order to map to the final output function. So, it is very simple and easy.

(Refer Slide Time: 23:44)



Backpropagation Preparation

- **Training Set**
  A collection of input-output patterns that are used to train the network
- **Testing Set**
  A collection of input-output patterns that are used to assess network performance
- **Learning Rate-η**
  A scalar parameter, analogous to step size in numerical integration, used to set the rate of adjustments

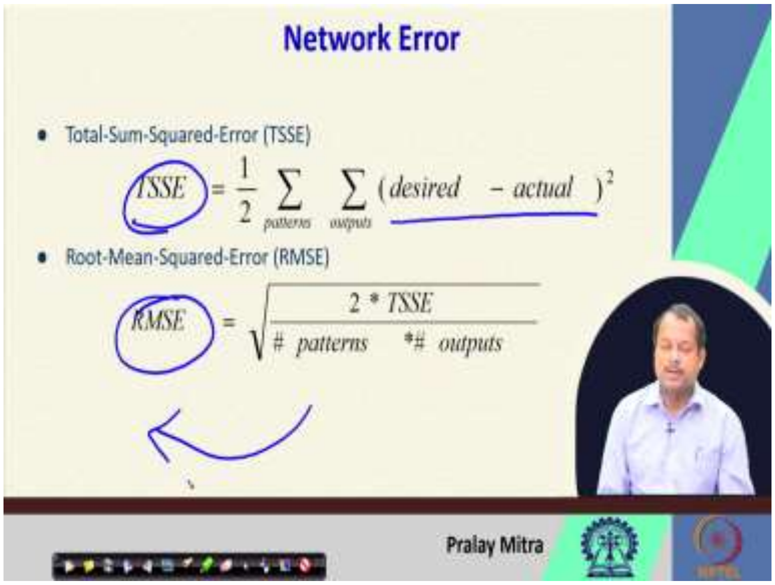
Pralay Mitra

Now, what is required for your back propagation? So, training set, a collection of input output patterns that are used to train the network I mentioned that it is required to compute your weight function. Next, testing set, a collection of input output patterns that are used to assess network performance. So, that assessment is also required.

So, if in your selection of the sequence indicates some bias, so it may be possible that you computed some weight value and so for a subset of sequences, the weight value is giving good result, but for other subset of sequences, it is not giving that good value. So, these differences may bias your system and may not allow you to go for a generic usage.

So, you should not use that. So, that is why testing is also done, and while doing that testing, some error may come. So, that error you can consider also as a feedback in your network so that by changing the learning rate, it is a scalar parameter allow us to step size in numerical integration used to set the date of adjustment, basically you adjust the weight values and you tune it. So, those are the things required.

(Refer Slide Time: 25:03)



Network Error

- Total-Sum-Squared-Error (TSSE)

$$TSSE = \frac{1}{2}\sum_{patterns}\sum_{outputs}(desired\ - actual\ )^2$$

- Root-Mean-Squared-Error (RMSE)

$$RMSE = \sqrt{\frac{2 * TSSE}{\#\ patterns\ \ *\#\ outputs}}$$

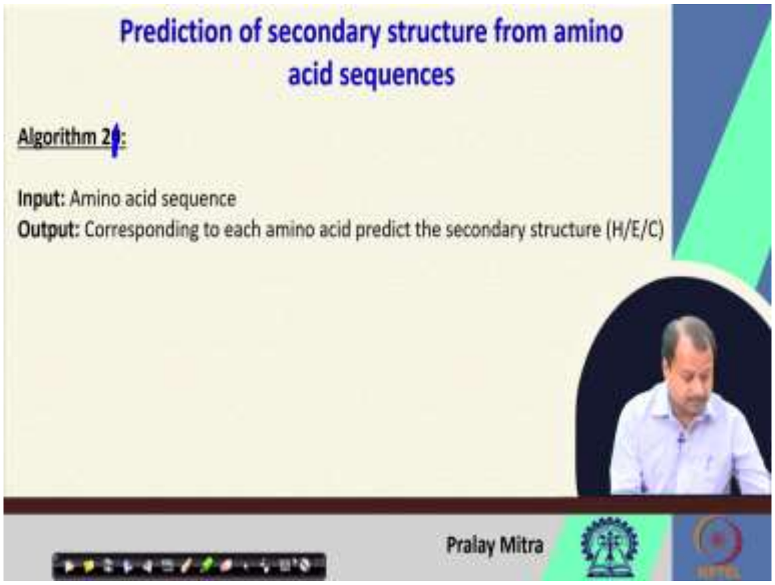
So, among several network errors, so total sum squared error which is given here and the root mean squared error based upon that total sum squared error is computed here. Basically, what it does the desired and actual the different square is taken as the error measure. So, when say you are having say 75 homologous sequences that I have mentioned, corresponding to each amino acid you have one prediction helix, sheet or coil, now you do not have any weight function.

So, you start with some random assignment of the weight function, and you check that what is the prediction, whether it is the helix or not. If helix, then actual desired is helix, actual is also helix. If say it is not then desired is say helix and actual is actually sheet. So, there is a difference, so you compute the difference, take the square of that, you compute the TSSE, you compute the RMSE, and you send back this RMSE, it will allow to change the weight value, and how many runs will be done that way depend upon the learning rate.

So, that way you keep on learning changing the weight value little bit and check what is the effect, what is the effect, whether it is improving or not that way you keep on changing and finally, you have the model. So, you understand that training or building the model or say identifying the weight values is time consuming process, but it will be done only once.

That way although significant amount of time may be required for the training but we should not bother much, I mean say much, which means if it is very high means several days or weeks then definitely it is a concern, otherwise it is not much, because once the model is developed I mean the weight functions are with me, then it will be just simple mathematical calculation in order to predict the output given one particular input.

(Refer Slide Time: 27:20)



Prediction of secondary structure from amino acid sequences

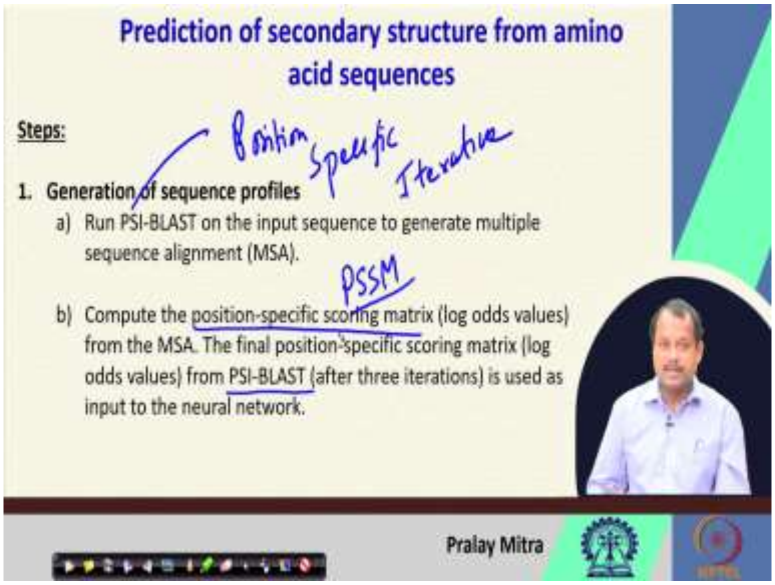
Algorithm 21:

Input: Amino acid sequence

Output: Corresponding to each amino acid predict the secondary structure (H/E/C)

Now, the prediction of secondary structure from amino acid sequence, we will use the same core idea for our own purpose. Input is going to be the amino acid sequence, and the output is going to be the corresponding to each amino acid predictor secondary structure H, R, or C, sorry as per the serial number that correct will be say 21, that we are going to discuss.

(Refer Slide Time: 27:52)



Prediction of secondary structure from amino acid sequences

Position Specific Iterative

Steps:

1. Generation of sequence profiles
   a) Run PSI-BLAST on the input sequence to generate multiple sequence alignment (MSA).

   PSSM

   b) Compute the position-specific scoring matrix (log odds values) from the MSA. The final position-specific scoring matrix (log odds values) from PSI-BLAST (after three iterations) is used as input to the neural network.

Pralay Mitra

So, regarding the steps, so first step is generation of sequence profile. So, run psi-blast. So, blast is most widely used sequence alignment program, so it is a web server is available or the source code is available to download and run in your system locally. So, run psi-blast, so psi is the short form of the position specific iterative blast, on the input sequence to generate multiple sequence alignment.

Now, compute the position specific scoring matrix log odd values from MSA, that multiple sequence alignment, the final position specific scoring matrix log odd values from psi-blast after three iterations is used as input to the neural network. So, right now, we know that what is going to be the input to our neural network method.

So, it is the output of the psi-blast technique, so given one protein sequence, so I will compute, so basically I will check or identify the homologous sequences, once the homologous sequences I will get then I will compute the position specific scoring metrics, so PSSM, you remember this PSSM we discussed in detail, we computed (())(29:35) weight and we discussed in the context of protein design also.

So, in this context also this position specific scoring matrix or PSSM will be useful. So, we will compute the position specific scoring matrix after obtaining the homologous sequences or performing the multiple sequence alignment of the homologous sequences corresponding to the input sequence. So, let us stop here in this lecture we will continue this discussion in the next lecture. Thank you for your attention.