<u>Techniques Used in Literatures</u>

- Amino Acid Composition (AAC)
- Dipeptide Composition
- Composition of k-spaced amino acid and pairs (CKSAAP)
- One-hot encoding
- KNN

<u>Amino Acid Composition:</u> In protein sequence there are 20 types of amino acids. If we want to describe a fraction of amino acid from the protein sequence, we have to use amino acid composition technique.

Formula: $f(r) = N_r /N$; $r = 1,2,3,\ldots\ldots..,20$

Here, N_r is the number of amino acid type. R and N is the length of the sequence.

- ACWYKAYWX window size: 9

A	C	D	E	....	W	X	Y
$\frac{2}{9}$	$\frac{1}{9}$	$\frac{0}{9}$	$\frac{0}{9}$	....	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{2}{9}$

We can use this technique. But it has few problems. For example, If I switch 2nd "W" with 3rd last "Y" It will give the same result. Another problem is it does not ensure the position of the amino acids.

Dipeptide Composition: Dipeptide composition will give 400-dimensional descriptor. If we count “X” the 441 dimensional (21*21).

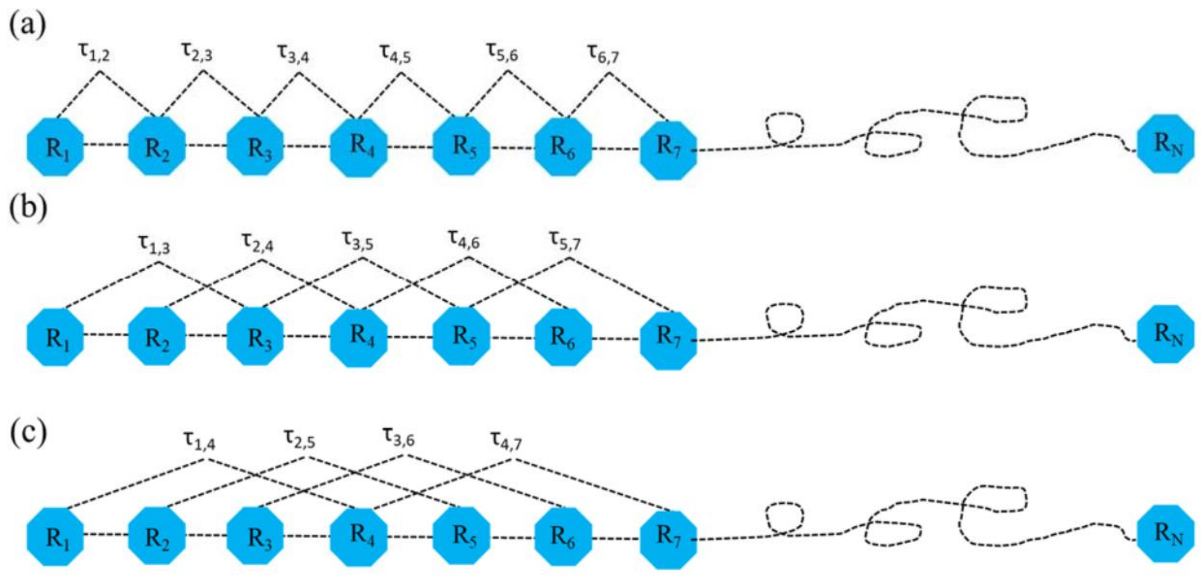
ACWYKAYWX

AA	AC	AD	AE	………..	YW	YX	YY
$\frac{0}{8}$	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{0}{8}$	………..	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{0}{8}$

Now, even if I switch the amino acids it will not give the same results.

Composition of k-spaced amino acid and pairs (CKSAAP): If we want to calculate the amino acid pair frequency separated by k residues(any) then 'Amino acid composition or, Dipeptide composition' won't be much of a help. To help CKSAAP comes handy.

Here, (K = 3) 0, 1 , 2


(a)
$\tau_{1,2}$ $\tau_{2,3}$ $\tau_{3,4}$ $\tau_{4,5}$ $\tau_{5,6}$ $\tau_{6,7}$
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_N
(b)
$\tau_{1,3}$ $\tau_{2,4}$ $\tau_{3,5}$ $\tau_{4,6}$ $\tau_{5,7}$
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_N
(c)
$\tau_{1,4}$ $\tau_{2,5}$ $\tau_{3,6}$ $\tau_{4,7}$
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_N

One Hot Vector: With one hot vector encoding we can convert the categorical variables into numerical values. For each amino acid 21 unique pattern is needed.

A = {1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
K = {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
D = {0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
.
.
.
.
Rest……….

One might ask why don’t we use {1,1,0,………….} instead of {1,0,0,…………}. For that to answer we have to think about the algorithm. If we use {1,1, 0,……….}, then it will put more weight than {1,0,0,…………}. We don’t want that.

KNN: K-nearest neighbor is a very simple Machine Learning algorithm. It can be used in both classification and regression problems. By using Euclidean distance, it calculates distance between points and selects specified number of examples (“K”) closest to that point. If the problem is classification problem it votes for frequent results. If it is regression problem; then it averages the results.

Some Basics

Weights & Bias: These two are learnable parameters.

$Y = WX + b$

Bias is presumption of a model. And weight is the gradient. When we change the value of "X"; "Y" changes with it. And "Y" changes big.

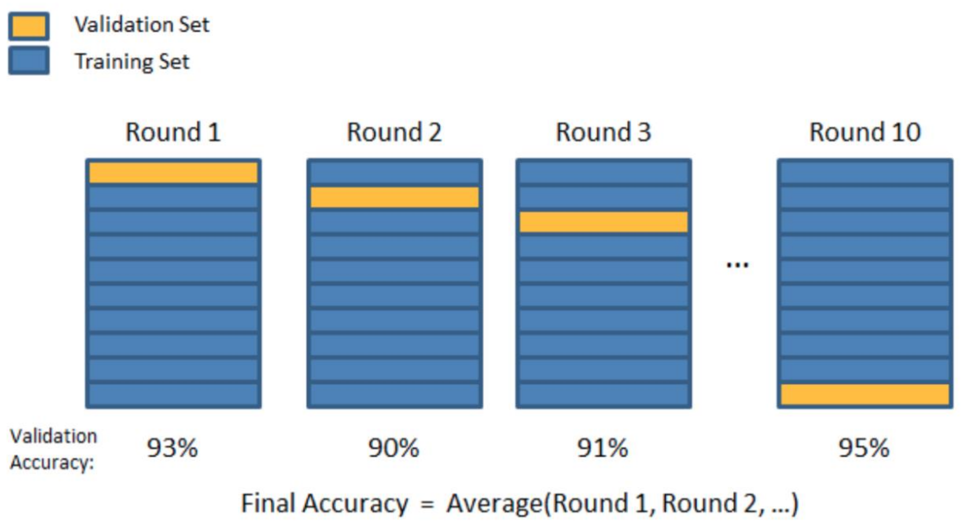
Overfitting: Overfitting means when our model learns too well. Why it happens you ask? It happens when the model is too complex. So, why it is not good? Because it can't perform well with unknown data. To solve this issue we can drop out some neurons from the layers or, we can use L1 or, L2 regularization to reduce the loss function.

Underfitting: It happens when the model does not fit with the data. It happens when the model is too simple or, we are using liner model to the data that are not linear. To avoid this issue we can use non linear model or, we can use more data to train the model.

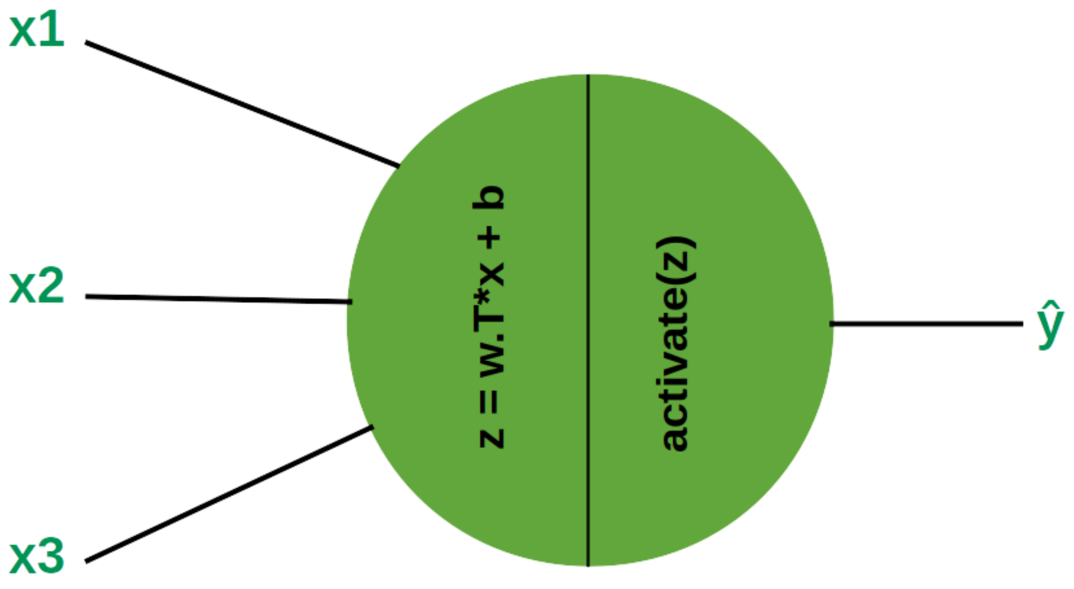
K Fold Cross Validation: I have read this term many times when reviewing papers. It is used to check the skill of my model has learned. It makes prediction on how my model will perform on unknown data.

Process:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    - Take the group as a hold out or test data set
    - Take the remaining groups as a training data set
    - Fit a model on the training set and evaluate it on the test set
    - Keep the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores.



<u>Activation Function:</u> Activation function is a mathematical “gate” in between the input feeding the current neuron and its output going to the next layer. They basically decide whether the neuron should be activated or not.



Without activation function weights and bias will transform only do linear transformation. No matter how many layers you put in the model. NN uses non linear activation functions like Relu, Sigmoid & tanh that helps the network to learn from complex data.