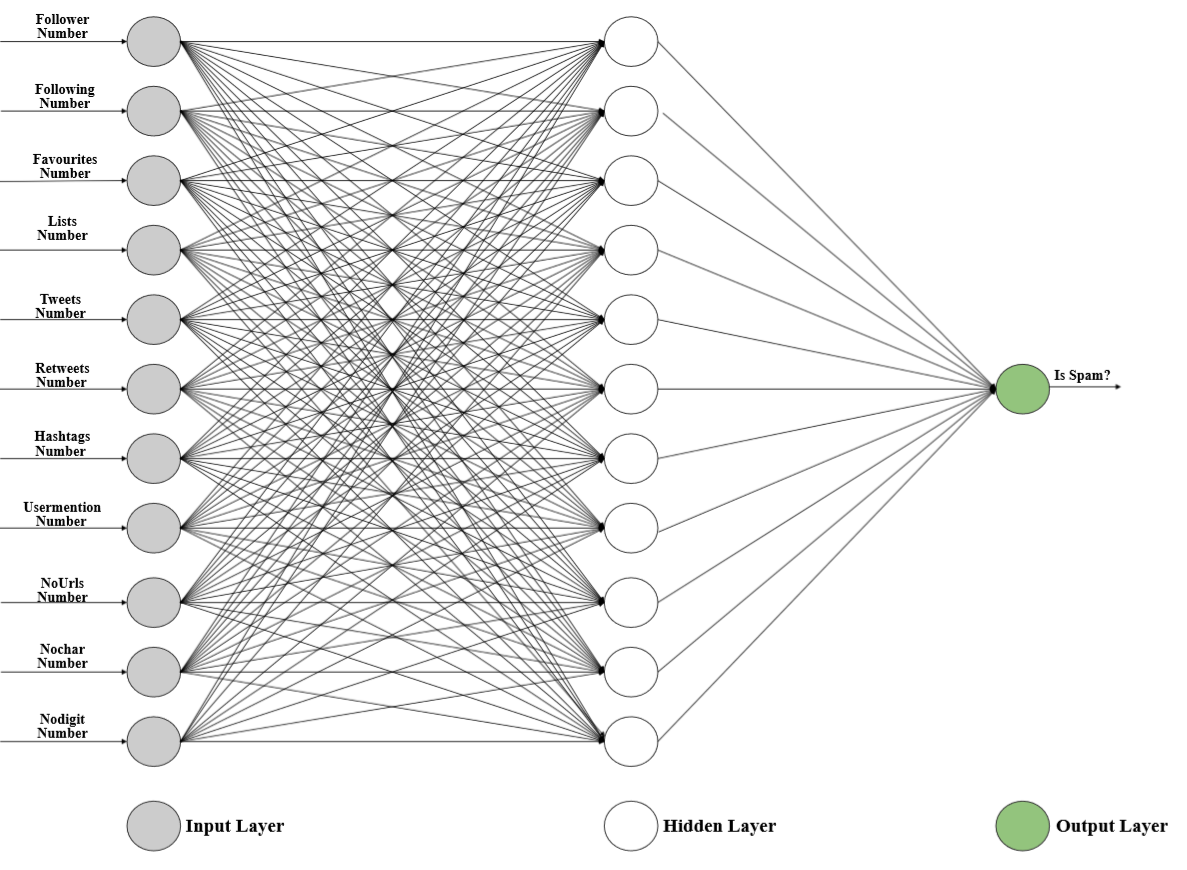
1. **EXAMINATION THE TWITTER SPAMMER ACCOUNTS WITH THE COMBINATION OF THE DIFFERENT ACTIVATION FUNCTIONS AND COMPARED THE EFFICIENCIES DIFFERENT COMBINATIONS**

Like we said in the previous title, we did an examination about finding twitter’s spam accounts using Artificial Neural Networks. Our dataset has eleven input and one output. These inputs are follower number, following number, favourites number, lists number, tweets number, retweets number, hashtags number, usermention number, NoUrls number, nochar number and nodigit number. And the output is spam.

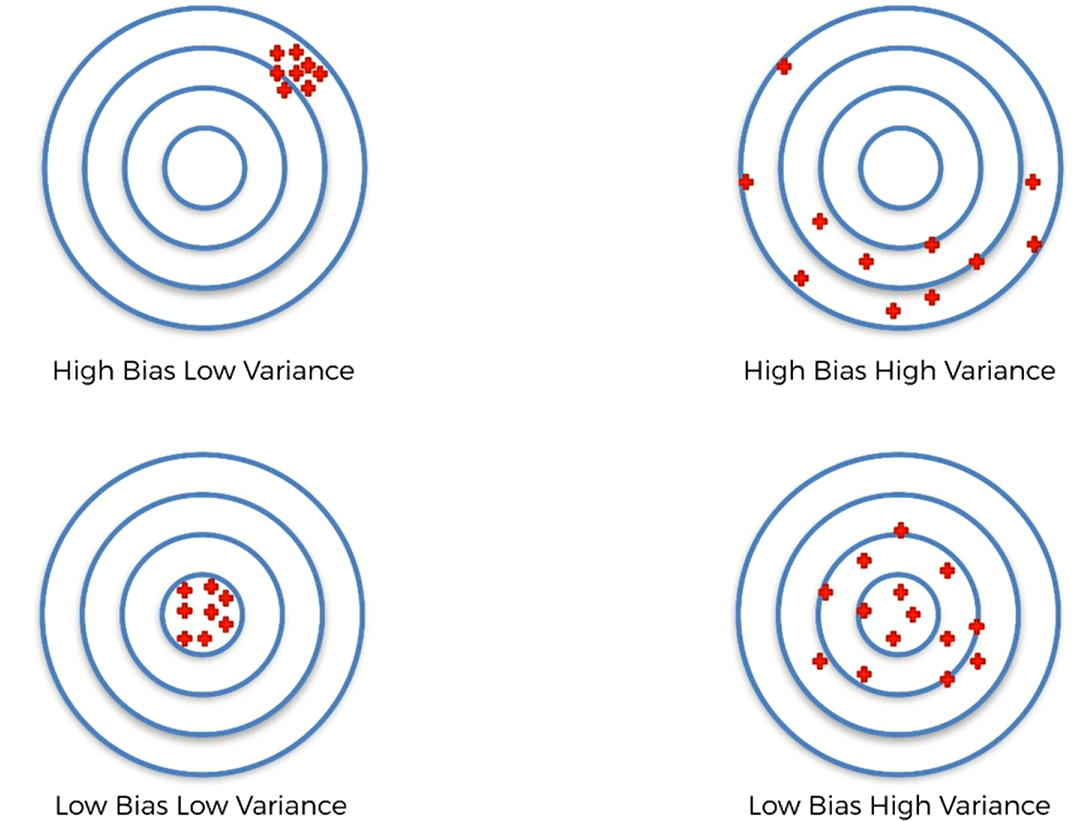
If spam, equal to ‘1’ that means this user is a spam or else spam is equal to ‘0’ that means this user is not a spam. Our output values will be between 0 and 1 after ANN. You can see our artifcial neural network in the below.



*Figure 2.1 Neural Network for Find the Twitter Spam Accounts*

**2.1 Evaluating the ANN**

When we trained our artificial neural network twice, we notice the second time we obtained a lower accuracy both on the training set and the test set than the first time. The reason of this is The Bias-Variance Tradeoff.



*Figure 2.1.1 The Bias-Variance Tradeoff.*

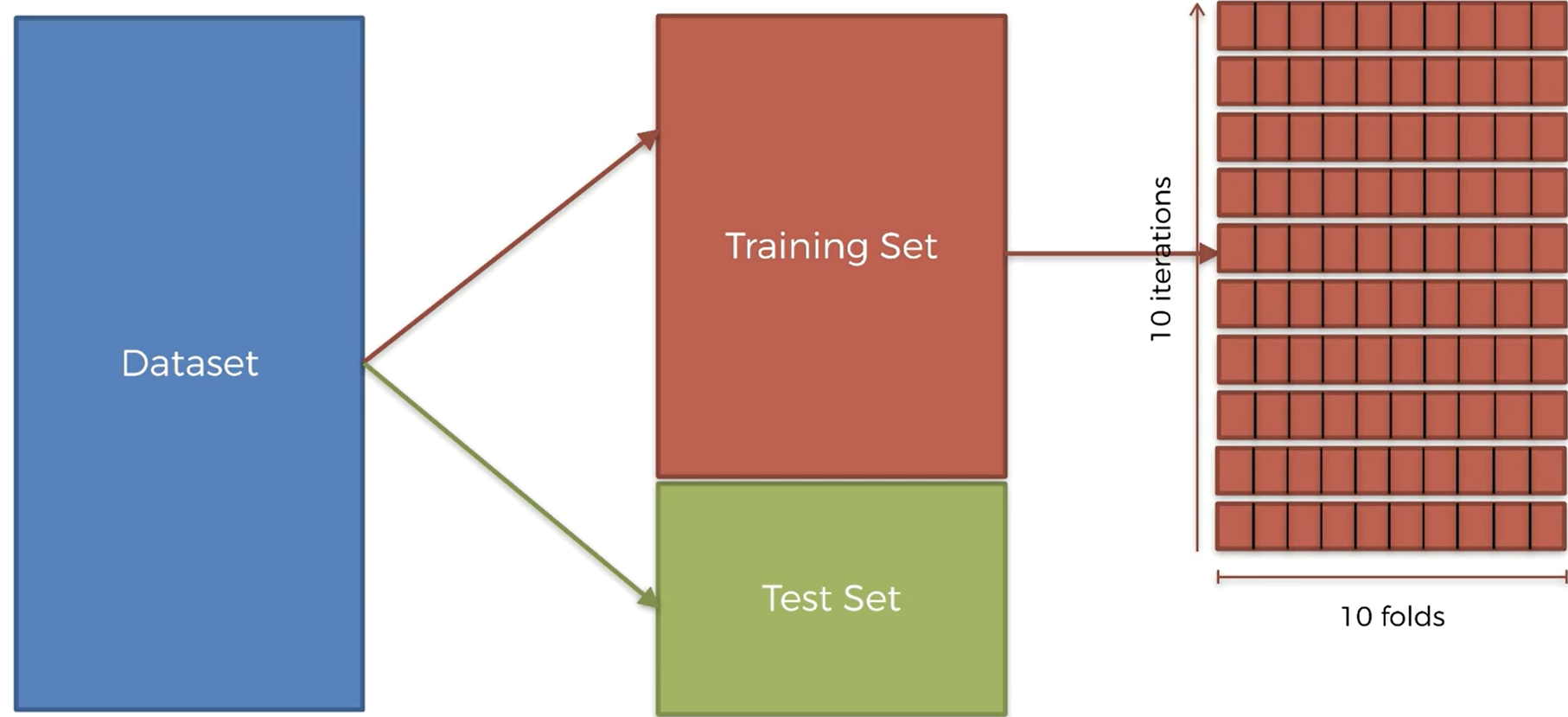
The Bias-Variance Tradeoff is the fact that we are trying to train the model that will not only accurate but also that should not have too much variance of accuracy, when we trained several times. And this happened our Artificial Neural Network when we trained twice we obtained two different accuracy. We can not be sure which one of these two accuracy we should take to evaluate models performances.

We need to optimized our weight to evaluatethe models. We split our dataset between the training set and the test set. We trained our model on the training set and we tested it performance on the test set.

That’s the correct way to evaluate the model performance but that’s not the best one. Because, we actually have the variance problem. The variance problem can be explain the fact that when we get the accuracy on the test set if we run the model again and test again performance on the another test set we can get very diffirent accuracy. So, judging our model performance only on one accuracy on one test set is actually not relevant. This is not the most relevant way to evaluate the model performance. For the fix that variance problem we used this tactic called k-Fold Cross Validation.

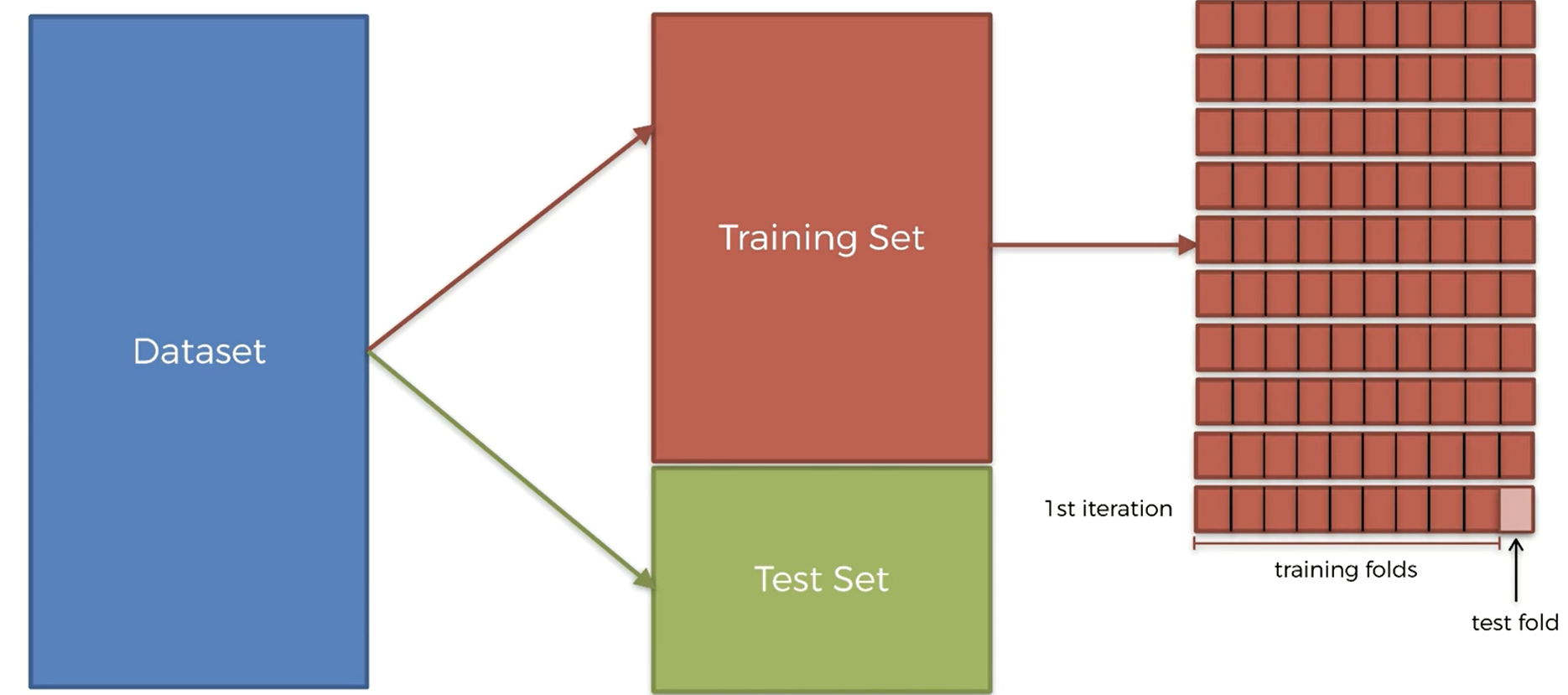
**2.2 k-Fold Cross Validation**

For the fix this variance problem we use k-Fold Cross Validation. It will fix it by splitting the training set into ten fold when k equals to ten and most at a time k equal ten.



*Figure 2.2.1 k-Fold Cross Validation*

We trained our model on nine folds and we tested on the last remaining fold. Since ten folds, we can make ten different combinations of nine folds trained the model an one fold the test it.

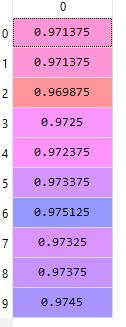


*Figure 2.2.2 k-Fold Cross Validation nine folds for train an one fold for test*

That means we can train the model and test the model on ten combinations of training and test sets. This given us a much better idea of model performance because we took an average of the ten different accuracies and also compute this ten of deviations to have look at the variance. Eventually our analysis became much more relevant.

* 1. **Performance of the Sigmoid as an Activation Function in the Input Layer, Hidden Layer and the Output Layer (Sigmoid - Sigmoid - Sigmoid )**

We implement the sigmoid function as an activation function to the input layer, hidden layer and the output layer. You can see the performance of the sigmoid function. Accuracies and the mean of accuracies like in the *Figure 2.3.1.* Also you can see the variance in the below. Variance is too low for this neural network and the mean of accuracies are high enough.

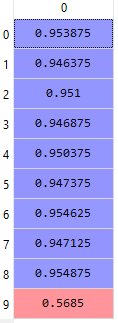




*Figure 2.3.1 Performance of the Sigmoid Function;**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

**2.4. Performance of the Hyperbolic Tangent as an Activation Function in the Input Layer, Hidden Layer and the Output Layer (tanh-tanh-tanh)**

We implement the Hyperbolic Tangentfunction as an activation function to the input layer, hidden layer and the output layer. You can see the performance of the Hyperbolic Tangentfunction. Accuracies and the mean of accuracies like in the *Figure 2.4.1.* Also you can see the variance in the below. Hyperbolic Tangent’s results not as good as the results of sigmoid function, the variance is too high and the mean of accuracies are too low.

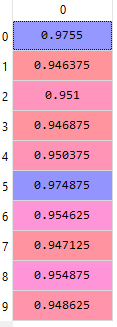


C:\Users\oguzh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Ekran Görüntüsü (31).png

*Figure 2.4.1. Performance of the Hyperbolic Tangent*;*Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

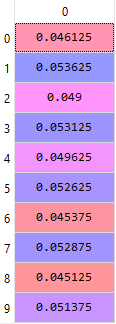
**2.5. Performance of the Rectifier Function as an Activation Function in the Input Layer, Hidden Layer and the Output Layer (relu-relu-relu)**

We implement the Rectifierfunction as an activation function to the input layer, hidden layer and the output layer. You can see the performance of the Rectifierfunction. Accuracies and the mean of accuracies like in the *Figure 2.5.1.* Rectifier Function’s results are better than Hyperbolic Tangent function’s results but still not good as Sigmoid Function’s results.



*Figure 2.5.1. Performance of the Rectifier Function*;*Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

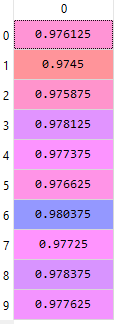
**2.6. Performance of the Softmax Function as an Activation Function in the Input Layer, Hidden Layer and the Output Layer (softmax-softmax-softmax)**

We implement the Softmaxfunction as an activation function to the input layer, hidden layer and the output layer. You can see the performance of the Softmaxfunction. Accuracies and the mean of accuracies like in the *Figure 2.6.1.* Softmax Function has the worst results. Because softmax function also used for the output layer. As an output layer softmax function is not suitable.

*Figure 2.6.1. Performance of the Softmax Function*;*Ten Accuracies*

**2.7. Performance of the Softmax Function and the Sigmoid Function as a Combination on the ANN (softmax - softmax - sigmoid)**

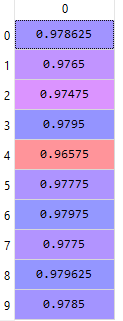
We implement the Softmax function as an activation function to the input layer and the hidden layer. For the activation of the output layer we used Sigmoid Function. You can see the performances of the Sigmoid function and the Softmaxfunction. Accuracies and the mean of accuracies like in the *Figure 2.7.1*. This combination has the best mean of the accuracies. Results are very close to combination of the sigmoid-sigmoid-sigmoid. Variance is too low which is very good.



*Figure 2.7.1. Performance of the Softmax Function and the Sigmoid Function as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

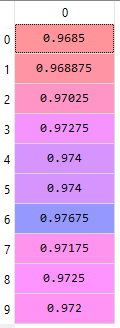
1. **EXTRAS**

**3.1 Performance of the Hyperbolic Tangent and the Sigmoid Function as a Combination on the ANN (tanh - tanh - sigmoid)**



*Figure 3.1.1. Performance of the Hyperbolic Tangent**and the Sigmoid Function as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

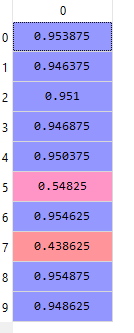
**3.2 Performance of the Rectifier Function and the Sigmoid Function as a Combination on the ANN (relu - relu - sigmoid)**



*Figure 3.2.1. Performance of the Rectifier Function**and the Sigmoid Function as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

**3.3 Performance of the Rectifier Function and the Hyperbolic Tangent as a Combination on the ANN (relu - relu - tanh)**

Input layer and hidden layer activated by the Rectifier Function. Output layer activated by the Hyperbolic Tangent.

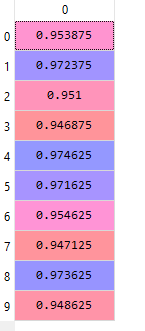


C:\Users\oguzh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Ekran Görüntüsü (38).png

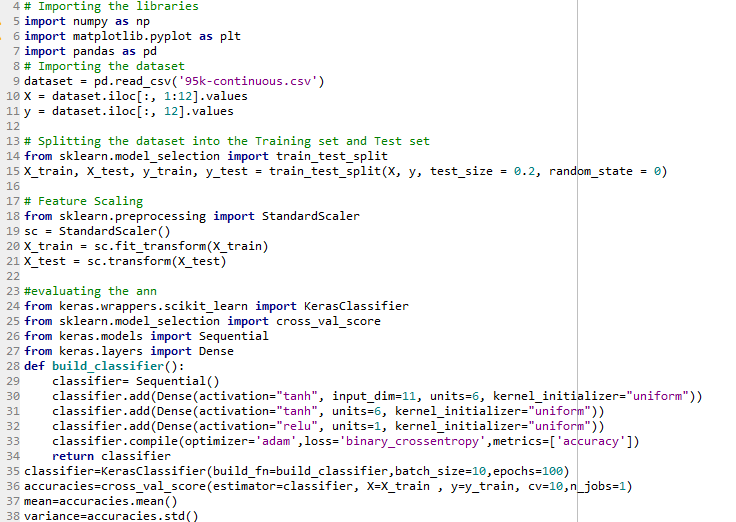
*Figure 3.3.1. Performance of the Rectifier Function**and the Hyperbolic Tangent**as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

**3.4 Performance of the Rectifier Function and the Hyperbolic Tangent as a Combination on the ANN (tanh - tanh - relu)**

Input layer and hidden layer activated by the Hyperbolic Tangent. Output layer activated by the Rectifier Function.

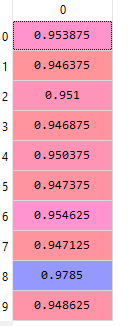


*Figure 3.4.1. Performance of the Rectifier Function**and the Hyperbolic Tangent**as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*



*Figure 3.4.2. Source code for creating an ANN. Input layer and hidden layer activated by the Hyperbolic Tangent. Output layer activated by the Rectifier Function*

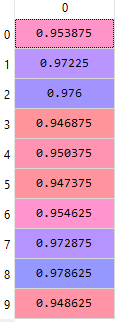
**3.5 Performance of the Softmax Function and the Hyperbolic Tangent as a Combination on the ANN (softmax - softmax - tanh)**

****



*Figure 3.5.1. Performance of the Softmax Function**and the Hyperbolic Tangent**as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*

**3.6 Performance of the Softmax Function and the Rectifier Function as a Combination on the ANN (softmax - softmax - relu)**



*Figure 3.6.1. Performance of the Softmax Function**and the Rectifier Function**as a Combination**Ten Accuracies on the Left, Mean of Accuracies and the Variance on the Right*