Perceptron / Neural Networks
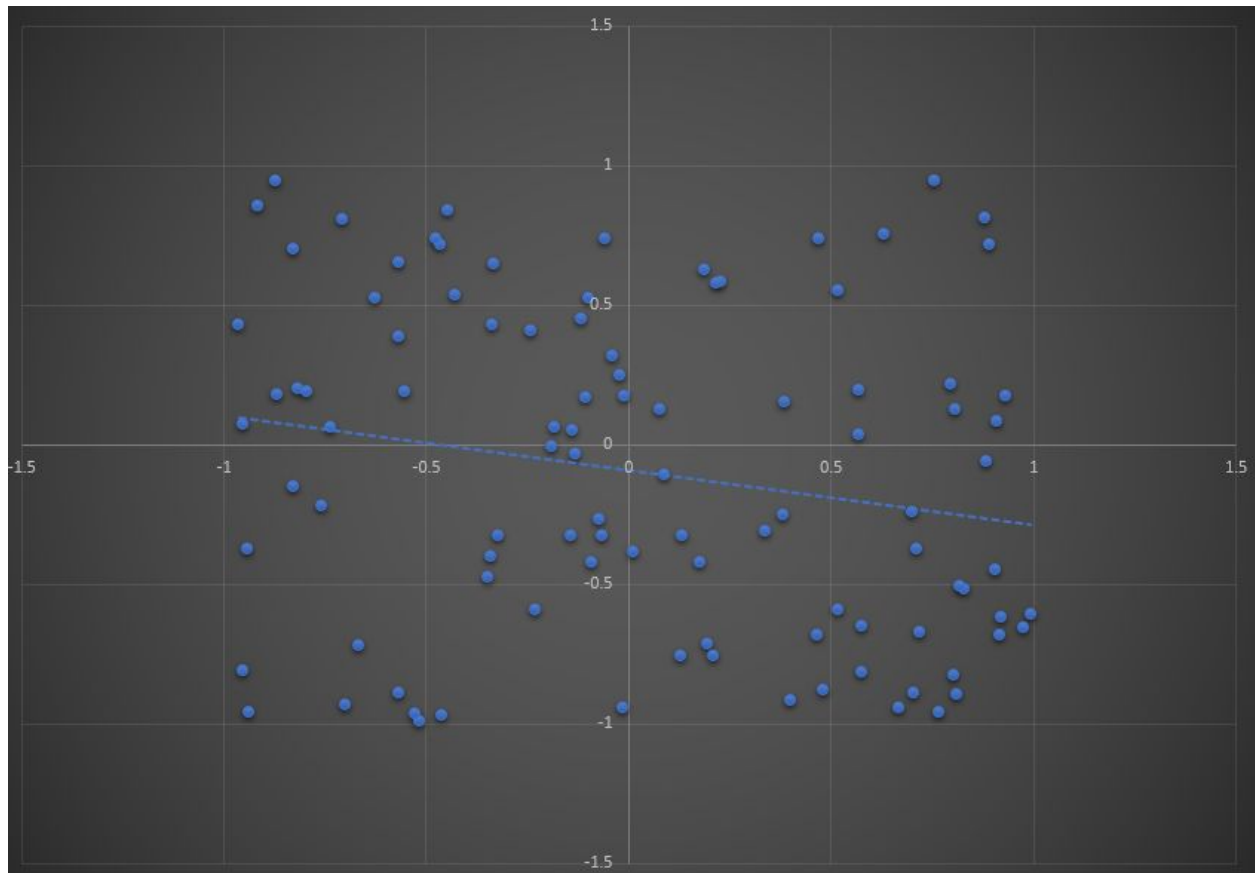
Perceptron Implementation

Formula:
Node 1 Weight: 0.5783828625959256
Node 2 Weight: -0.6011930719462093
Node 0 Weight: 0.7895747249438426

Node 1 Weight * Input value (X) + Node 2 Weight * Input Value (Y) + Node 0 Weight
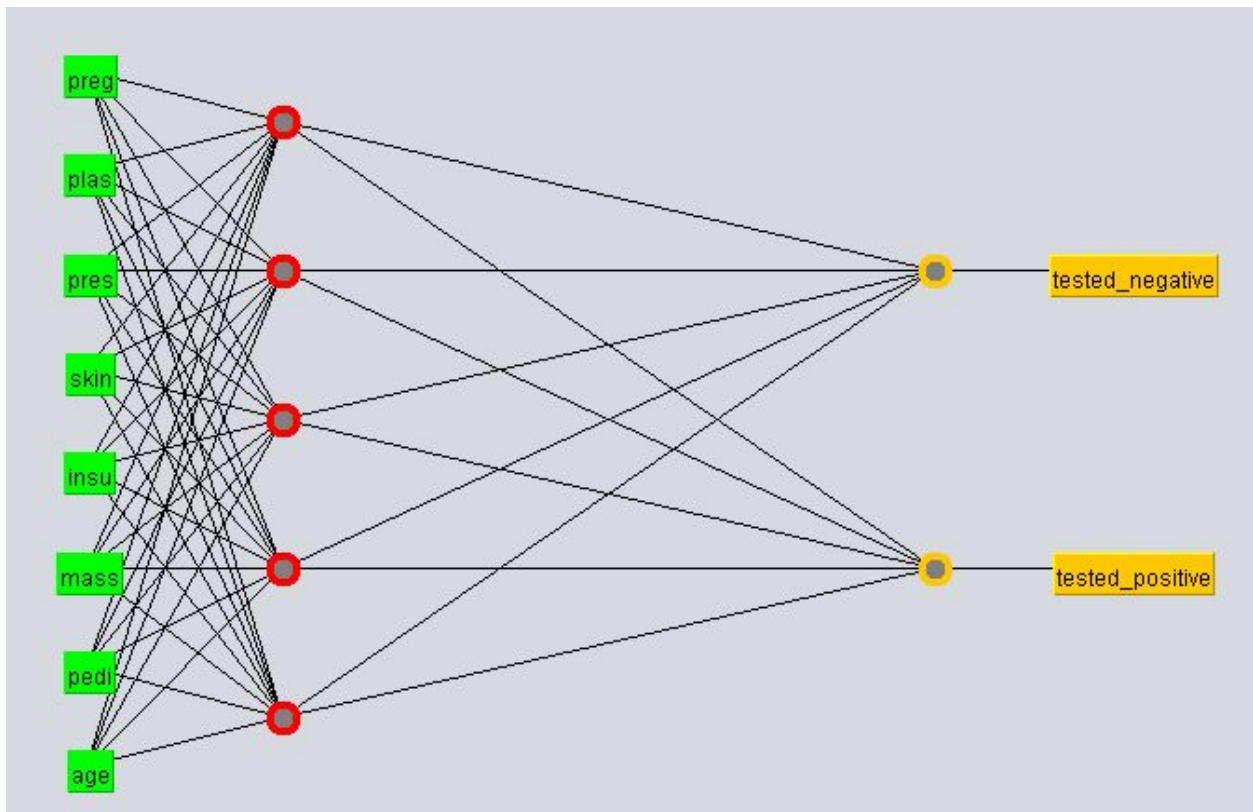
Multi-layer perceptron

- **Exercise:**
  - **Parameters:**
    - Inputs nodes: 9
    - Hidden nodes: 1
    - Outputs nodes: 2
    - Learning rate: 0.3
    - Momentum: 0.2
    - Instances:768
  - **Neural Network:**

  - **Input Nodes:**

No.		Name
1		preg
2		plas
3		pres
4		skin
5		insu
6		mass
7		pedi
8		age
9		class

```
=== Summary ===

Correctly Classified Instances         500               65.1042 %
Incorrectly Classified Instances       268               34.8958 %
Kappa statistic                          0
Mean absolute error                      0.4545
Root mean squared error                  0.4766
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  MCC     ROC Area  PRC Area  Class
                 1.000    1.000    0.651      1.000   0.789      0.000   0.497     0.650     tested_negative
                 0.000    0.000    0.000      0.000   0.000      0.000   0.497     0.348     tested_positive
Weighted Avg.    0.651    0.651    0.424      0.651   0.513      0.000   0.497     0.544

=== Confusion Matrix ===

   a    b    <-- classified as
 500    0 |  a = tested_negative
 268    0 |  b = tested_positive
```

- ○ **Reflection:** The ANN converges slow because it has a low learning rate, that means that if the learning rate is small, the convergence will be slow and if it is big, the convergence will be fast. Also in the Weka, the weight of the thresholds of each sigmoid node have a big difference between them.

- **Explanations as to what are ANNs good for.**

  It is good for problems with a large datasets or problems with prior knowledge

- **Where would you use them?**

  ANN is used in: medical, military, industrial, software, anomaly detection, etc.

- **Are they worth the effort implementing or not?**

  There is not a correct answer because ANN can be the best option in a problems but in other problem can be a bad option

- **What kinds of problems do they not solve?**

  One example that ANN can't solve is a person's recognition in photographs because they can analyze it but if you input a photo with a lot of noise or with a different rotation, the recognition can fail