Perceptron algorithm

- Early machine learning model - looks similar to linear regression but used for classification tasks
- requires data to be linearly seperable (cannot work with XOR)

Model:

$$y = \begin{cases} 1 & \text{if } \omega \cdot x - \theta \geq 0 \\ 0 & \text{if } \omega \cdot x - \theta < 0 \end{cases}$$

where ω are the weights and θ is the threshold

The algorithm

Initialize weights (ω) randomly. Let the true / positive class = 1 and the false / negative class = 0.

- While the algorithm has not converged:
  - pick an x datapoint
  - have the model make a prediction
  - If the prediction is wrong (from cornell notes, this is if $y_i(\omega^T \cdot x) \leq 0$; then:
    - update the weights
    - $\omega_{new} = \omega + yx$
  - If we go through the entire dataset without having to make an update, then the algorithm has converged.

Margin

given by:

$$\gamma = \min|x^T\omega|$$

The perceptron will converge in $\frac{1}{\gamma^2}$ iterations

References

- https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975
- https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html