1.1 Error Function

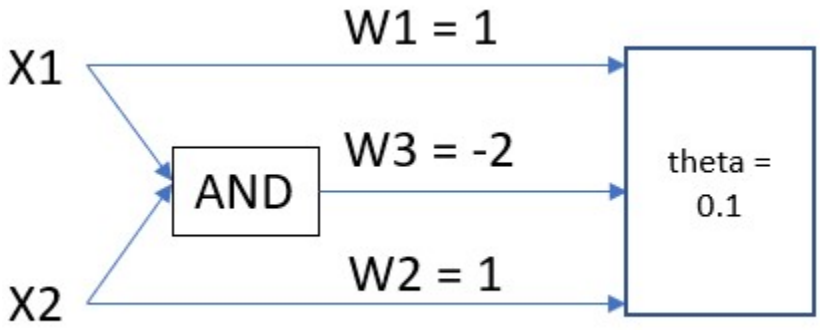
1.2 error function in neural networks

Neural network will try to minimize the error function by optimizing the weight for each neuron. If y becomes probability instead of labels, it is equivalent to a regression problem and techniques used for regression can be used in this case.

2. A and not B, and A XOR B

A and not B, w1 x A + w2 x B > theta, where theta = 0, w1 = 1, w2 = -1, and A = {0, 1} and B = {0,1}

A XOR B:



3. perceptron and gradient descent training rules

Gradient descent training rule:

$$E = 0.5\sum(y - o)^2$$

Take derivative of wi,

$$\frac{\partial E}{\sigma \omega_i} = \frac{\partial}{\partial w_i}\frac{1}{2}\Sigma(y - o)^2 = \sum(y - o)\left(x_i + x_i^2\right)$$

Therefore, we get

$$\Delta\omega_i = \eta(y - o)(x_i + xi^2)$$

Perception training rule:

The perceptron training rule has similar form as gradient descent rule, where o is replaced by $\hat{y}$:

$$\Delta\omega_i = \eta(y - \hat{y})(x_i + xi^2)$$

Where $\hat{y} = y$, if o>=theta, otherwise 0.

4. Use decision tree to perform regression

Decision tree can be used to perform regression, with a different type of loss function that classification problem. If we use squared error, for each split, the algorithm is trying to minimize the following equation:

$$MSE = \frac{1}{N_t}\sum_i (y_i - \hat{y})^2$$

Where Nt is the number of data points under a node, yi is the individual data points, and $\hat{y}$ is the average value of yi:

$$\hat{y} = \frac{1}{N_t}\sum_i y_i$$

When $y_i = \hat{y}$, that would give the minimal MSE (=0), and that is the reason why the expected at any leaf is the mean, given the above error function.

5. Design a lazy decision tree algorithm

A typical lazy decision tree algorithm comprises the following workflow (https://pdfs.semanticscholar.org/38b0/877fa6ac3ebfbb29d74f761fea394ee190f3.pdf):

> Input: A training set T of labeled instances and an unlabeled instance I for classification.

1. if T is pure, i.e., all instances in T have label L, return L
2. otherwise, if all instances in T have the feature values, return the majority class in T.
3. otherwise, select a test X and let x be t he value of the test on the instance I. Assign the set of instances with X=x to T and apply the algorithm to T.

6. Which method to use, decision tree or nearest-neighbor?

I would use decision tree. Decision tree does exactly what the target function does: split and assign one side of the hyper-line(-plane) to one value and the other side a different value. On the other hand, what nearest-neighbor does is a distance-based approximation. Instead of drawing lines, nearest-neighbor tends to draw curves whose shape is determined by the value of data points.

7. VC dimension for a origin-centered circle and origin-centered sphere

Origin-centered circle: 1

If there is one point, this point can be either in or out of the circle.

If there are two points, and their distance to the center is different, we can never have the closer point out of the circle but the farther point in the circle. Therefore the VC dimension is 1.

Origin-centered sphere: 1

If there is one point, this point can be either in or out of the sphere

If there are two points, and their distance to the center is different, we can never have the closer point out of the sphere but the farther point in the sphere. Therefore the VC dimension is 1.