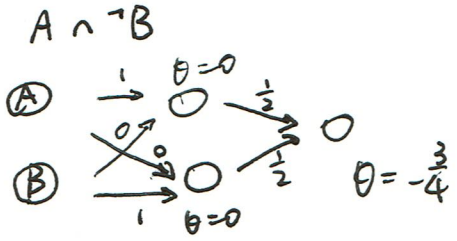
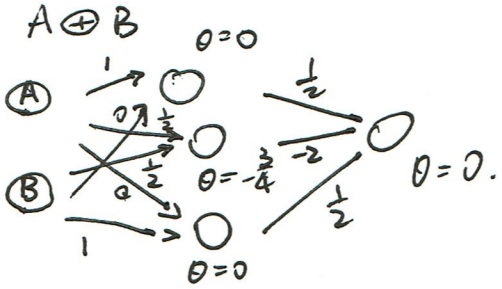
Question 2

And could be achieved by putting 0.5 for the weight and -0.7 as theta. Negative could be put as -1 and theta as 0.7.

For XOR, it's 'or' – 'and'*2. So, the second layer weight for A and B is 0.5 and for the 'and' is -2 and theta is 0.





Question 3

Gradient boosting works better for training data that's are not linearly separatable and perform more robust.

perceptron Training

$w_i := w_i + \Delta w_i$

$\Delta w_i = \eta (y - \hat{y}) x_i$

Gradient boosting :

$E(W) = \frac{1}{2} \sum (y-a)^2$

$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \frac{1}{2} \sum (y-a)^2 = \sum (y-a) \frac{\partial}{\partial w_i} -\sum x_i w_i = \sum (y-a) x_i$

Question 4

Instead of using gini index or entropy, it selects the feature to split by minimizing the variance of the target variable. At the leaf node, it takes the mean of the target variable as the prediction for the sample.

```
In [9]: import pandas as pd

        from sklearn.model_selection import cross_val_score

        from sklearn.model_selection import train_test_split

        from sklearn.pipeline import Pipeline
        from sklearn.preprocessing import StandardScaler,MinMaxScaler  # doctest: +SKIP
        from sklearn.model_selection import GridSearchCV


        from sklearn.model_selection import ShuffleSplit
        from matplotlib import pyplot as plt
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.model_selection import learning_curve
        %matplotlib inline

In [10]: X_train, X_test, y_train, y_test = train_test_split(df.data, df.target, test_size=0.2, random_state=42)

In [13]: X_train.shape
Out[13]: (404, 13)

In [20]: regressor = DecisionTreeRegressor(random_state=88,max_depth=3,min_samples_leaf=5)
         regressor.fit(X_train,y_train)
         regressor.score(X_test,y_test)

Out[20]: 0.7175975744790715
```

Let k be the value that minimize the error function, Y is the target variable.

$E(y-k)^2 = E(y^2 - 2yk + k^2)$

$= \int y^2 dy - 2k \int y f(y) dy + k^2 \int f(y) dy$

$= \int y^2 dy - 2k u + k^2 \Rightarrow$ to minimize $E(y-k)^2 \Rightarrow \frac{\partial E}{\partial k} = 0 \Rightarrow -2u + 2k = 0$

$\therefore u = k.$

Question 5

My lazy algorithm will split randomly on features. The good thing about this algorithm is it saves the calculation time to select the best feature to split, whether it's Gini index or information entropy. And you don't need to define the metric for 'best feature' to begin with.

Another benefit is the randomness can help to avoid the overfitting with the training data.

The disadvantage is the algorithm might converge slower than the eager algorithm and the tree might be deeper to the algorithm.

Question 6

Decision tree should perform better than the nearest neighborhood.

KNN would suffer for the points lying on the or around the lines.

On the other hand, with two features and a theta, decision tree can describe the exact line on the plane.

Question 7

1. Circle should have 3 VC dimensions. Since 3 points confirms a circle, then if the 4^{th} point is within the triangle, there's no way to exclude the point
2. Sphere should have 4 VC dimensions. 4 points confirms a sphere, then if the 5^{th} point is with the sphere, there's no way to exclude the point