Explanation of modified points

```
return (1.0 / (1.0 + exp(-1.0 * (x))));
```

1つの極を持つシグモイド関数を定義。

```
double sum = 0.0;
for(size_t i = 0 ; i < length ; i++){
    sum += v1[i] * v2[i];
}
return sum;
```

v1とv2の内積を計算する。各点同士の積を足し合わせる。

```
mlp->output_errors[i] = diff * ((1.0 - output) * output);
```

シグモイドの微分をdiffにかける。

```
weights[i][j] += learning_rate * errors[i] * inputs[j];
```

各点の重さの計算。

Discussion on the results of your program.

どの点が0と1のどちら側に移動したか

Add the answer for the algorithm assignment.

- Effective input
  適切に処理、フォーマット化された入力データ。
- Desired output
  特定の入力を達成することを目指すために期待される結果。
- Actual output
  生成された実際の結果。
- Error signal
  Desired outputとActual outputの差。

Based on slides 19-22, summarize the “error back propagation” algorithm for a three layer neural network.

Step1: 重みを初期化する。

Step2: 合計誤差をリセットする。

Step3: 各トレーニング例で隠れ層と出力層のニューロンの出力を計算し、合計誤差を更新する。

Step4: 目標出力と実際の出力の差に基づいて、出力ニューロンの誤差信号を計算する。

Step5: 誤差信号とニューロンの出力を使用して、出力層と隠れ層の両方の重みを更新する。

Step6: すべてのトレーニング例を使用したか確認する。していない場合、Step3に戻る。

Step7: 合計誤差が所定の値より小さいか確認する。そうでない場合、誤差をリセットしてStep2に戻る。そうであれば終了する。