

정확도 측정

```
var fitParam = { epochs: 100 } // 몇번 반복할 것인가
model.fit(원인, 결과, fitParam)
```

```
<!DOCTYPE html>
<html>

<head>
  <title>TensorFlow.js Tutorial - lemon</title>

  <!-- Import TensorFlow.js -->
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0/dist/tf.min.js"
></script>

</head>

<body>
  <script>
    // 1. 과거의 데이터를 준비합니다.
    var 온도 = [20, 21, 22, 23];
    var 판매량 = [40, 42, 44, 46];
    var 원인 = tf.tensor(온도);
    var 결과 = tf.tensor(판매량);

    // 2. 모델의 모양을 만듭니다.
    var X = tf.input({ shape: [1] });
    var Y = tf.layers.dense({ units: 1 }).apply(X);
    var model = tf.model({ inputs: X, outputs: Y });
    var compileParam = { optimizer: tf.train.adam(), loss:
tf.losses.meanSquaredError }
    model.compile(compileParam);

    // 3. 데이터로 모델을 학습시킵니다.
    //           var fitParam = {epochs: 100}
    var fitParam = {
      epochs: 100,
      callbacks: {
        onEpochEnd: // 콜백 함수 얼마나 실행되었는가를 볼 수 있다
          function (epoch, logs) { // epoch, logs
```

```

        console.log('epoch', epoch, logs, 'RMSE=>',
Math.sqrt(logs.loss)); // loss : 얼마나 잘 학습되었는냐 0에 가까울 수록 학습이 잘 되
있는지 알 수 있다.
    }
}
} // loss 추가 예제
model.fit(원인, 결과, fitParam).then(function (result) {

    // 4. 모델을 이용합니다.
    // 4.1 기존의 데이터를 이용
    var 예측한결과 = model.predict(원인);
    예측한결과.print();

});

// 4.2 새로운 데이터를 이용
// var 다음주온도 = [15,16,17,18,19]
// var 다음주원인 = tf.tensor(다음주온도);
// var 다음주결과 = model.predict(다음주원인);
// 다음주결과.print();
</script>
</body>

</html>

```

원인	결과	모델의 예측값	차이 예측값 - 실제값	제곱 차이 ²
1	10	7	-3	9
2	10	10	0	0
3	10	13	3	9

$$\frac{-3 + 0 + 3 (0)}{3}$$

Mean Squared Error (MSE)
평균 제곱 오차

$$\frac{9 + 0 + 9 (18)}{3} = 6$$

Root Mean Squared Error (RMSE)
평균 제곱근 오차

$$\sqrt{\frac{9 + 0 + 9 (18)}{3}} (6) = 2.449$$

54.204442430003394	
epoch 93 ▶ {loss: 2935.7158203125} RMSE=> 54.18224635720173	6.html:34
epoch 94 ▶ {loss: 2933.3115234375} RMSE=> 54.16005468458742	6.html:34
epoch 95 ▶ {loss: 2930.90869140625} RMSE=> 54.13786744420443	6.html:34
epoch 96 ▶ {loss: 2928.507568359375} RMSE=> 54.11568689723318	6.html:34
epoch 97 ▶ {loss: 2926.10791015625} RMSE=> 54.0935107952539	6.html:34
epoch 98 ▶ {loss: 2923.70947265625} RMSE=> 54.0713368861567	6.html:34
epoch 99 ▶ {loss: 2921.3125} RMSE=> 54.049167431145506	6.html:34
Tensor	array_ops.ts:1067
$\begin{bmatrix} -10.1826859 \\ -10.696785 \\ -11.210885 \\ -11.7249851 \end{bmatrix}$	

RMSE : 값 값만큼 오차가 있다

epoch 값이 클수록 학습이 많이되고 RMSE값이 줄어 들게 된다!