

## 12.3.7 모델 구성 요소에 기반한 손실과 지표 (add\_loss, add\_metric))

**Note.** 본 자료는 오렐리앙 제롱의 핸즈온 머신러닝 (2판)을 참고하여 작성되었습니다.

일반적으로 손실이나 지표는 실제  $y$  값 (정답; 입력된 정답)과 예측  $\hat{y}$  값 (추정; 출력된 예측)으로부터 얻은 오차에 기반하여 정의된다. 실제  $y$  값과 예측  $\hat{y}$  값 간의 차, 즉 오차를 계산하는 방식을 정의한 함수를 손실함수라고 한다. 손실함수는 `mean_squared_error`, `categorical_crossentropy` 등 다양한 형태로 존재한다. 여기서 주의할 점은 우리가 보통 손실을 정의할 때 실제 값과 예측 값의 차를 활용한다는 것이다.

그러나 때로는 실제 값 (입력 값)과 예측 값 (출력 값)이 아닌 중간 값 (은닉층의 가중치 혹은 중간 활성화 층의 반환 값) 등 모델의 구성 요소를 기반으로 손실을 정의해야 할 때가 있다. 이렇게 중간 값을 활용한 보조 손실을 정의하면 규제 및 모델의 내부 상황 모니터링에 유용하다.

그럼 이렇게 중간 값을 활용한, 즉 모델 구성 요소에 기반한 손실을 정의한 신경망 모델을 구축해보자. 예를 들어 다섯 개의 은닉층과 출력층으로 구성된 회귀분석용 MLP 모델의 코드는 아래와 같다.

```
class ReconstructionRegressor(keras.Model):
    def __init__(self, output_dim, **kwargs):
        super().__init__(**kwargs)
        self.hidden = [keras.layers.Dense(30, activation = "selu",
kerel_initialization = "lecun_normal") for _ in range(5)] # Dense() 객체를 5
회 호출하여 리스트 형태로 담은뒤 self.hidden 인스턴스 속성으로 정의한다.
        self.out = keras.layers.Dense(output_dim) # output_dim을 Dense() 객
체의 n_neurons에 해당하는 매개변수로 전달하여 self.out 인스턴스 속성으로 정의한다.

    # 보조 손실 (재구성 손실) 정의
    ## 모델 구성 요소 중 '완전 연결층'을 활용해 재구성 손실을 계산할 수 있다. 재구성 손실
을 계산하기 위해 구축된 층을 재구성 층이라고 한다.
    ## 재구성 손실은 어느 한 완전 연결층이 반환하는 잠재행렬 (represented matrix) Z와
입력 가s 가s 오차를 의미한다.
    ## 재구성 손실을 계산하기 위해선 재구성 층의 반환 가s과 입력 가s을 비교해야 하므로 재
구성 층의 뉴런 개수는 입력 가s 변수 개수 (차원크기)와 가s게 설정한다.
    ## 입력배치의 차원크기가 몇 개가 될지 알 수 없으며 이는 재구성 층이 뉴런 개수 역시 알
수 없다는 뜻이다.
    ## 차원크기의 자유도를 보장하기 위해 재구성 층은 `build()` 메서드에
self.reconstruct 인스턴스 속성으로 정의한다.
    def build(self, batch_input_shape):
        n_inputs = batch_input_shape[-1]
        self.reconstruct = keras.layers.Dense(n_inputs)
        super().build(batch_input_shape) # 부모 클래스의 인스턴스 속성인
self.built에 True를 전달한다.

    # 본 클래스의 `build()` 메서드가 부모
클래스 `build()` 메서드를 오버라이딩 하면서 self.reconstruct의 존재가 부모 클래스에 알
려진다.

    def call(self, inputs):
        Z = inputs
        for layer in self.hidden:
```

```

        Z = layer(Z)
        reconstruction = self.reconstruct(Z)

        # 재구성 손실함수 정의
        recon_loss = tf.reduce_mean(tf.square(reconstruction - inputs))

        # `add_loss(x)` 메서드는 모델 실행시 손실을 저장하는 리스트에 x가스를 추가한다.
        self.add_loss(0.05 * recon_loss)
        return self.out(Z)

```

위 코드에서 주목할 만한 사실들은 다음과 같다.

- 보조 손실함수 `recon_loss`의 변수인 `reconstruction`은 `self.reconstruct()` 층이 반환한 잠재행렬이다.
- 재구축 손실은 신경망을 통과한 입력값의 표현 (representation), 즉 잠재행렬이 얼마나 입력 값을 유사하게 "재구축"하였는지를 수치화 한 값이다. 다시 말해 재구축 손실은 잠재행렬과 입력 값 간 오차로 정의 되므로 잠재행렬의 차원크기와 입력배치의 변수 개수는 같아야 한다.
- `self.reconstruct()` 인스턴스 속성이 `build()` 메서드에서 정의되는데 그 이유는 차원의 크기가 입력배치의 변수개수(차원크기)와 같아야 하는데 `build()` 메서드가 호출되기 전에는 입력배치의 차원크기를 알 수 없으므로 `batch_input_shape` 라는 클래스 속성을 활용하기 위함이다.
- `self.add_loss()` 를 통해 손실을 리스트 자료형으로 저장할 수 있으며 저장된 손실은 아래와 같은 방법으로 조회 가능하다.

```

RR = ReconstructRegressor(output_dim)
y = RR(x)

print(RR.losses)

```

- 다만 `add_loss()` 는 계속 덧씌워지며 `losses` 메서드가 반환하는 손실 리스트는 마지막 순전파시 손실이 저장된다는 점을 유의해야 한다.
  - `add_loss()` 가 활용된 모델을 학습시키는 방법은 <https://www.kaggle.com/kurianbenoy/tensorflow-2-0-keras-crash-course> 에서 확인할 수 있다.
- `add_metric()` 역시 `add_loss()` 와 같은 방식으로 활용할 수 있다.