



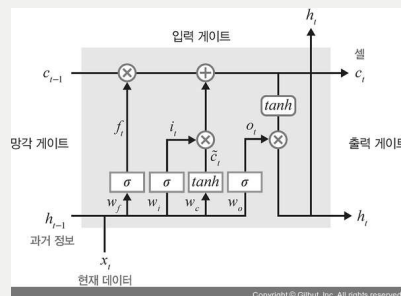
7-5장. LSTM

RNN의 기울기 소멸 문제를 해결하기 위해 LSTM이나 GRU를 이용한다.

- LSTM의 순전파: 망각게이트, 입력 게이트, 출력 게이트

망각 게이트(forget gate): 과거 정보와 현재 데이터를 입력받은 후, 시그모이드를 출력값에 따라 0이면 정보를 버리고 1이면 보존한다.

입력 게이트(input gate): 현재 정보를 기억하기 위해 만들어짐. 시그모이드와 하이퍼볼릭 탄젠트 함수를 기반으로 현재 정보에 대한 보존량을 결정한다.



```

# gates.chunk(4,1)
def forward(self, x, hidden):
    hx, cx = hidden
    x = x.view(-1, x.size(1))

    gates = self.x2h(x) + self.h2h(hx)
    gates = gates.squeeze()
    ingate, forgetgate, cellgate, outgate = gates.chunk(4, 1)

    ingate = F.sigmoid(ingate)
    forgetgate = F.sigmoid(forgetgate)
    cellgate = F.tanh(cellgate)
    outgate = F.sigmoid(outgate)

    cy = torch.mul(cx, forgetgate) + torch.mul(ingate, cellgate)
    hy = torch.mul(outgate, F.tanh(cy))
    return (hy, cy)

```

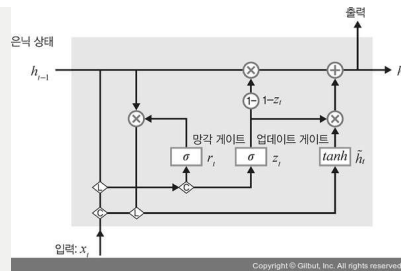
→ 게이트는 `gates.chunk(4,1)`에 의해 네 개로 쪼개져 각각 망각, 입력, 셀, 출력 게이트로 저장된다.

-LSTM의 역전파: 셀을 통한 역전파를 수행(중단 없는 기울기)

7-6장. 게이트 순환 신경망(GRU)

구조: LSTM에서 사용하는 망각 게이트와 입력 게이트를 하나로 합친 것, 별도의 업데이트 게이트로 구성. 출력 게이트가 없어 전체 상태 벡터가 매 단계마다 출력됨.

- 업데이트 게이트: 과거와 현재 정보의 최신화 비율을 결정하는 역할.



```
class GRUCell(nn.Module):
    def __init__(self, input_size, hidden_size, bias=True):
        super(GRUCell, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.bias = bias
        self.x2h = nn.Linear(input_size, 3 * hidden_size, bias=bias)
        self.h2h = nn.Linear(hidden_size, 3 * hidden_size, bias=bias)
        self.reset_parameters()
```

→ LSTM에서는 4를 곱했으나 GRU에서는 세 개의 게이트를 사용하여 3을 곱한다. (망각, 입력, 새로운 게이트)

7-7장. 양방향 RNN

과거 시점과 미래 시점의 데이터를 함께 활용하여 출력 값을 예측한다.

하나의 출력값을 예측하는 데 메모리 셀 두 개를 사용한다.

→ 양방향 LSTM에 적용

```
# bidirectional=True
self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size,
                    num_layers=num_layers, bidirectional=True, batch_first=True)
```

```
# self.num_layer*2
def forward(self, x):
    h_0 = Variable(torch.zeros(self.num_layers*2, x.size(0), self.hidden_size))
    c_0 = Variable(torch.zeros(self.num_layers*2, x.size(0), self.hidden_size))
    out, _ = self.lstm(x, (h_0, c_0))
    out = self.fc(out[:, -1, :])
    out = self.relu(out)
    return out
```