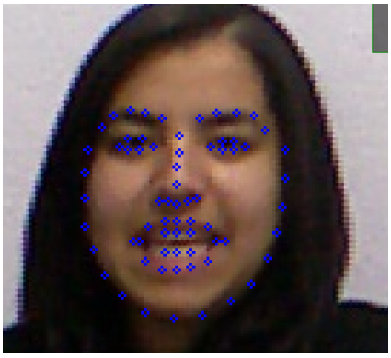
12주차 연구노트

○ 12주차 계획(11주차 피드백 반영)

1. Lip 데이터 전처리
2. MobileNet + LSTM 모델 선정

○ 데이터 전처리

11주차 피드백에서 교수님께서 말씀하신 데이터 전처리과정 수행

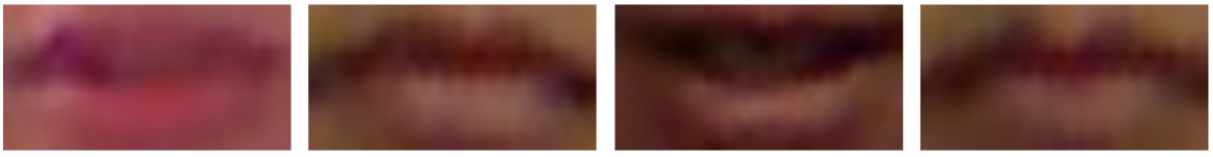


각 단어의 발화영상의 얼굴 부분 Landmark 생성

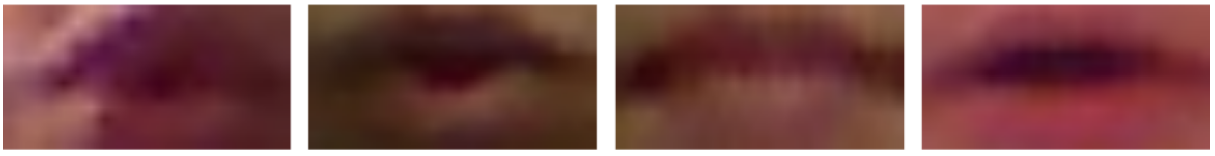
dlib 라이브러리를 활용하여 입술 부분만 추출한뒤 데이터셋 정리

```python
# lip_dataload 함수 정의
def lip_dataload(image):
    face_det = dlib.get_frontal_face_detector()
    landmark_model = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
    image = np.array(image)
    MOUTH_OUTLINE = list(range(48, 61))

    src = image
    grey = cv.cvtColor(src, cv.COLOR_BGR2GRAY)
    faces = face_det(grey)

    for face in faces:
        lm = landmark_model(grey, face)
        lm_point = []
        for p in lm.parts():
            lm_point.append([p.x, p.y])
        lm_point = np.array(lm_point)

        mouth_points = lm_point[MOUTH_OUTLINE]
        min_x = np.min(mouth_points[:, 0])
        max_x = np.max(mouth_points[:, 0])
        min_y = np.min(mouth_points[:, 1])
        max_y = np.max(mouth_points[:, 1])

        mouth = src[min_y:max_y, min_x:max_x]
        mouth = cv.resize(mouth, (120, 60))
        return mouth
    return None
```

ex)



Begin

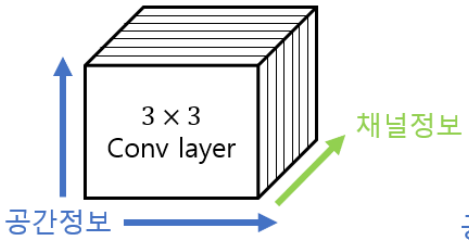


Hello

연속된 이미지기 때문에 시퀀스형태로 데이터셋 구성

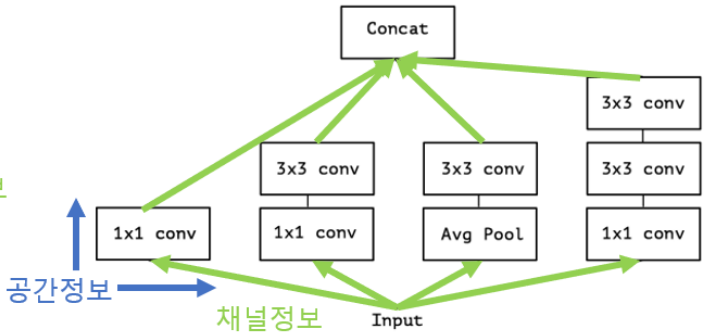
○ MobileNet 모델

⇒ Mobilenet은 효율적인 연산을 위해 Depthwise Separable Convolution을 적절히 활용한 경량화 네트워크이다. Depthwise Separable Convolution은 Google에서 발표한 네트워크 Xception에서 처음 설명된 개념이다. Xception은 GoogLeNet의 Inception 모듈에 대해 고찰한 내용을 발전시켜 설계한 네트워크이다.



일반적인 Convolutional layer

Figure 1. A canonical Inception module (Inception V3).

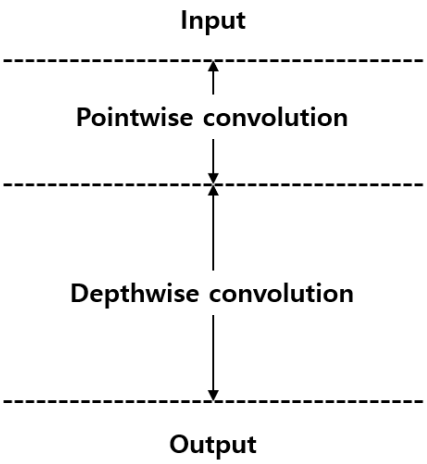


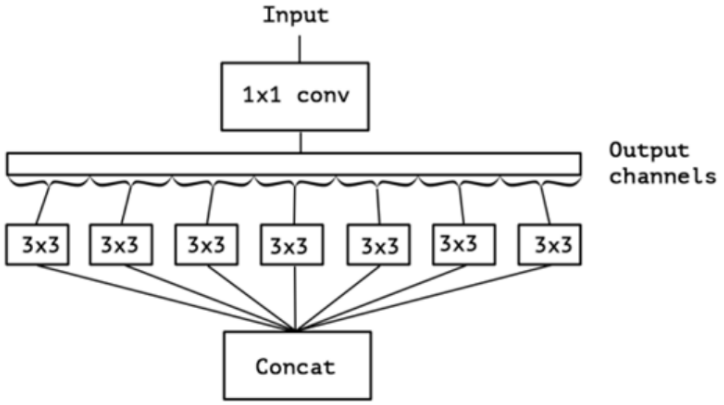
Inception 모듈

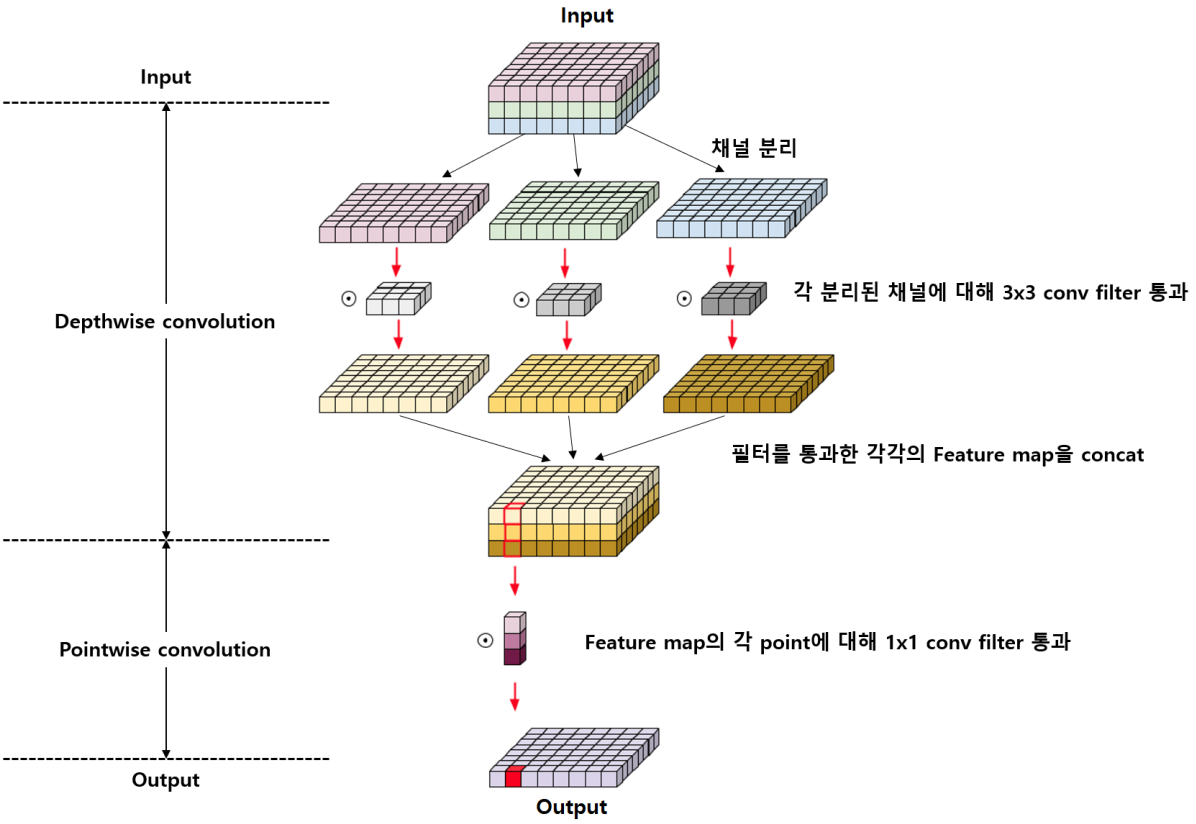
기본적인 Convolution layer는 2차원 공간(width, height)과 1차원 채널로 이뤄진 3차원 구조의 filter이다. 그러므로 Conv layer를 학습할 땐 cross-channel correlation과 cross-spatial correlation, 즉 공간사이의 상관관계 분석과 채널사이의 상관관계 분석이 **동시에** 일어난다.

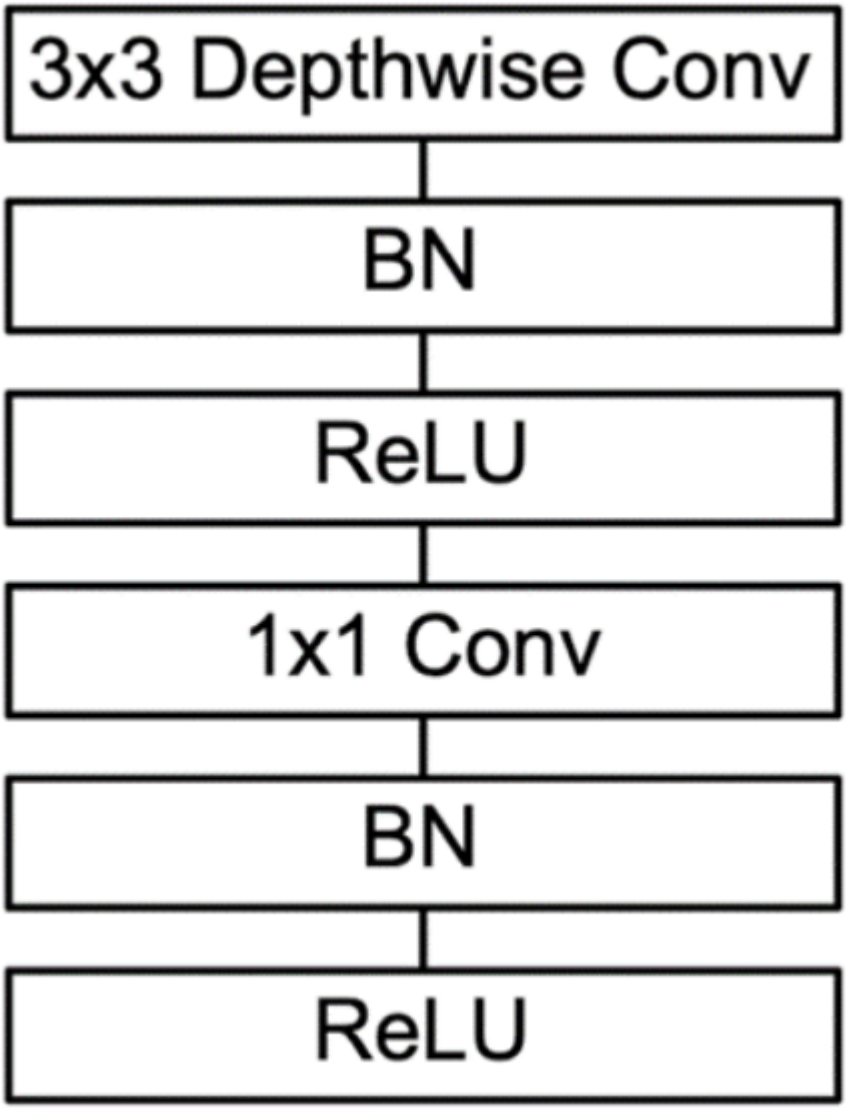
그런데 Inception 모듈은 채널 정보가 4개로 나뉘어 각각의 conv layer에서 공간사이의 상관관계 분석이 이뤄진 후 **마지막으로 채널정보가 합쳐지는** 구조이다.

⇒ 여기서 알 수 있는 가정은, Inception 모듈이 채널 사이의 상관관계와 공간사이의 상관관계를 분리해서 처리할 수 있다는 점이다.


Input
Pointwise convolution
Depthwise convolution
Output


Input
1x1 conv
Output channels
3x3
3x3
3x3
3x3
3x3
3x3
3x3
Concat


Input
Depthwise convolution
Pointwise convolution
Output
Input
채널 분리
각 분리된 채널에 대해 3x3 conv filter 통과
필터를 통과한 각각의 Feature map을 concat
Feature map의 각 point에 대해 1x1 conv filter 통과
Output



※ MobileNet + LSTM 네트워크 모델로 선정 ⇒ 기존 네트워크들 (VGG+LSTM,CNN+LSTM)과 다르게 연산량을 압도적으로 낮추면서도 Pooling layer을 사용하지 않아 해상도 유지 가능.

출처 : https://velog.io/@woojinn8/LightWeight-Deep-Learning-5.-MobileNet

○ LSTM

ㅁ LSTM(Long Short-Term Memory)은 RNN(Recurrent Neural Network)의 일종으로, 긴 시퀀스의 데이터를 처리하는 데 적합하다. RNN은 시퀀스 데이터(예: 시간 시리즈, 텍스트 데이터 등)에서 중요한 정보를 캡처할 수 있는 모델이지만, 시퀀스가 길어질수록 학습

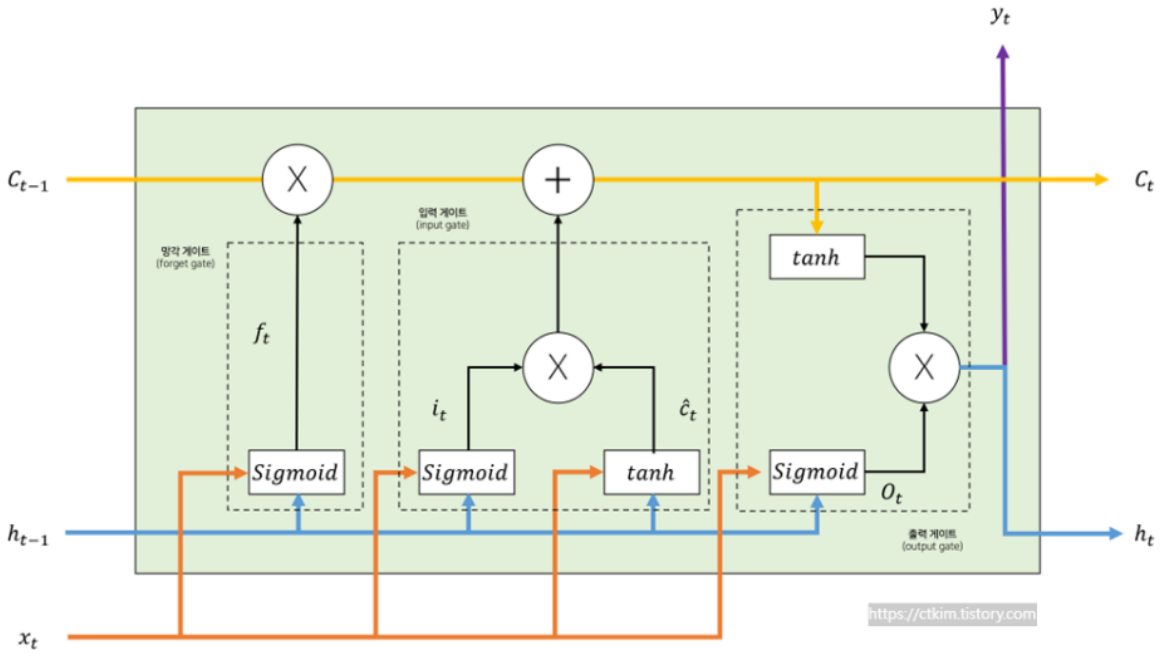
이 어려워지는 문제(기울기 소실 문제)가 있다. LSTM은 이러한 문제를 해결하기 위해 개발되었다.

ㅁ LSTM의 장점

장기 의존성 문제 해결 + 시퀀스 데이터를 처리하는데 적합함 + 기울기 소실 문제 해결 가능

ㅁ LSTM의 구조

LSTM은 기본적으로 셀 상태(cell state)와 여러 게이트(gate)로 구성되어 있다. 셀 상태는 정보의 흐름을 조절하며, 게이트는 정보를 추가하거나 제거하여 셀 상태를 업데이트한다.



1. 셀 상태(Cell State)

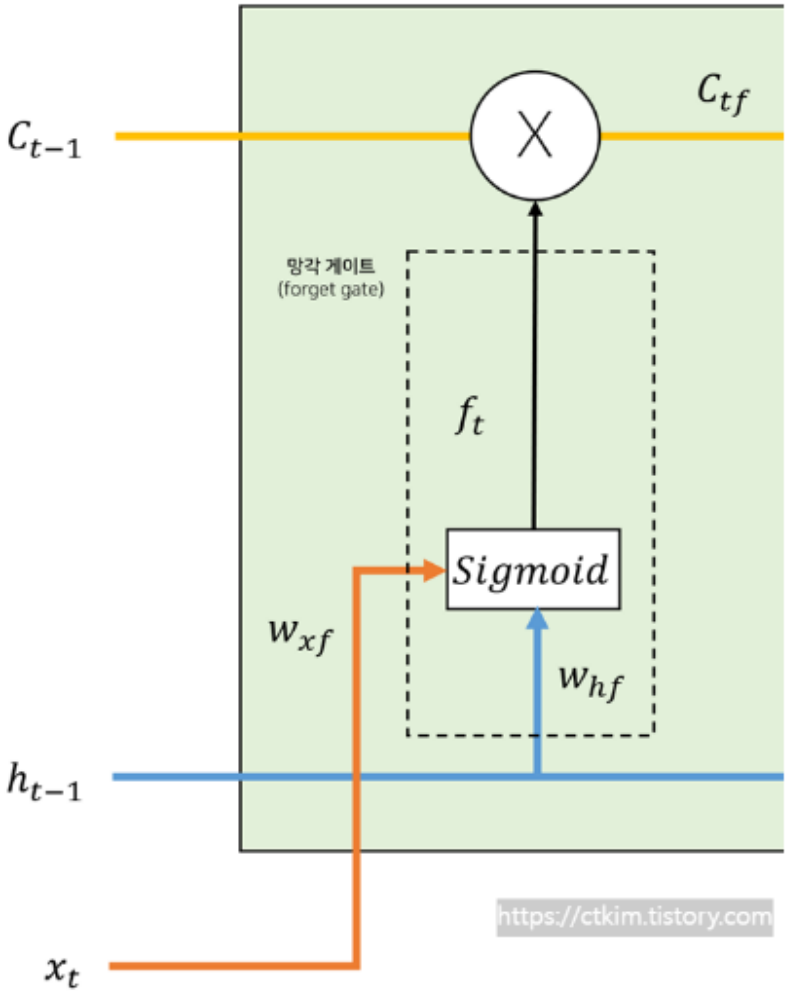
이전 상태에서 현재 상태까지 유지되는 정보의 흐름을 나타낸다. 이를 통해, LSTM은 오래된 정보를 기억하고 새로운 정보를 적절히 갱신할 수 있다. 셀 상단에 Cell State가 있으며 자기 자신에게 피드백된다. 망각 게이트와 입력 게이트를 거치고 나온 결과가 다음 입력이 된다.

2. 게이트(Gate)

게이트는 Sigmoid 함수(값이 0에서 1 사이인 출력)를 사용하여 정보를 선택적으로 통과시키는 역할한다.

- **망각 게이트(Forget Gate)**: 셀 상태에서 이전 셀 상태의 정보를 얼마나 잃어버릴지(즉, 잊어버릴지) 결정합니다. Sigmoid 함수를 사용해, 값이 0이면 셀 상태정보는 0이되어 사라지고 1이면 그대로 전달됨.

$$f_t = \sigma(W_{xf}h_{t-1} + W_{xf}x_t)$$



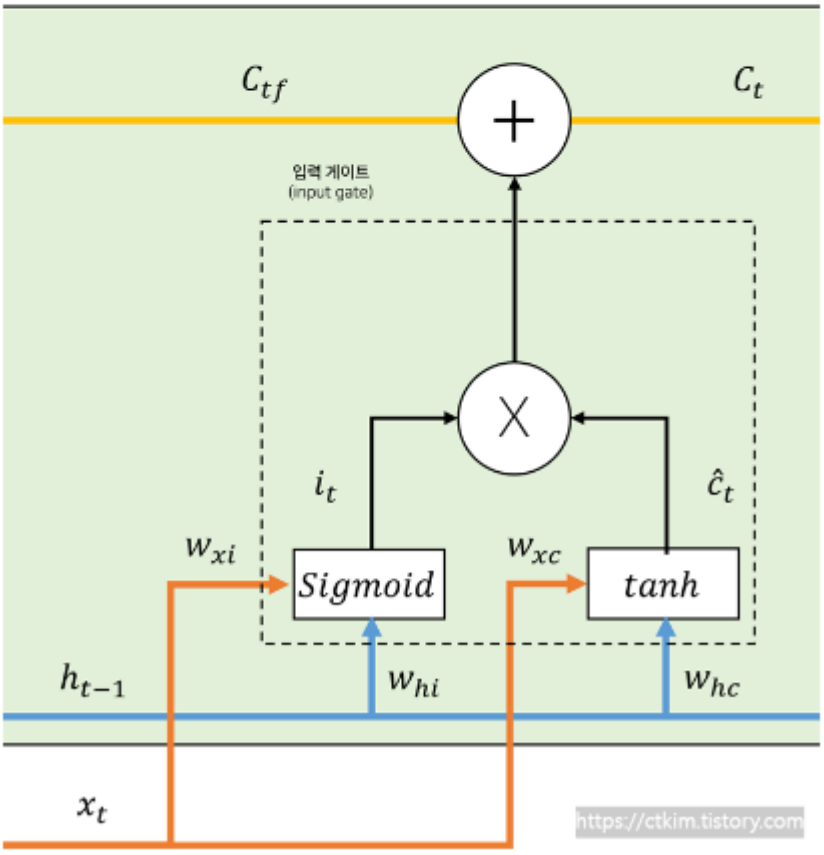
- **입력 게이트(Input Gate)**: 새로운 정보를 셀 상태에 어떻게 반영할 것이지 결정. 이를 통해 LSTM은 기존 정보와 새로운 정보를 적절하게 조합하여 예측 수행이 가능해진다.

- tanh function
- Sigmoid function : 후보 값을 얼마나 전달할지 정함

$$i_t = \sigma(W_{hi}h_{t-1} + W_{xi}x_t)$$

$$\hat{C}_t = tanh(W_{hc}h_{t-1} + W_{xc}x_t)$$

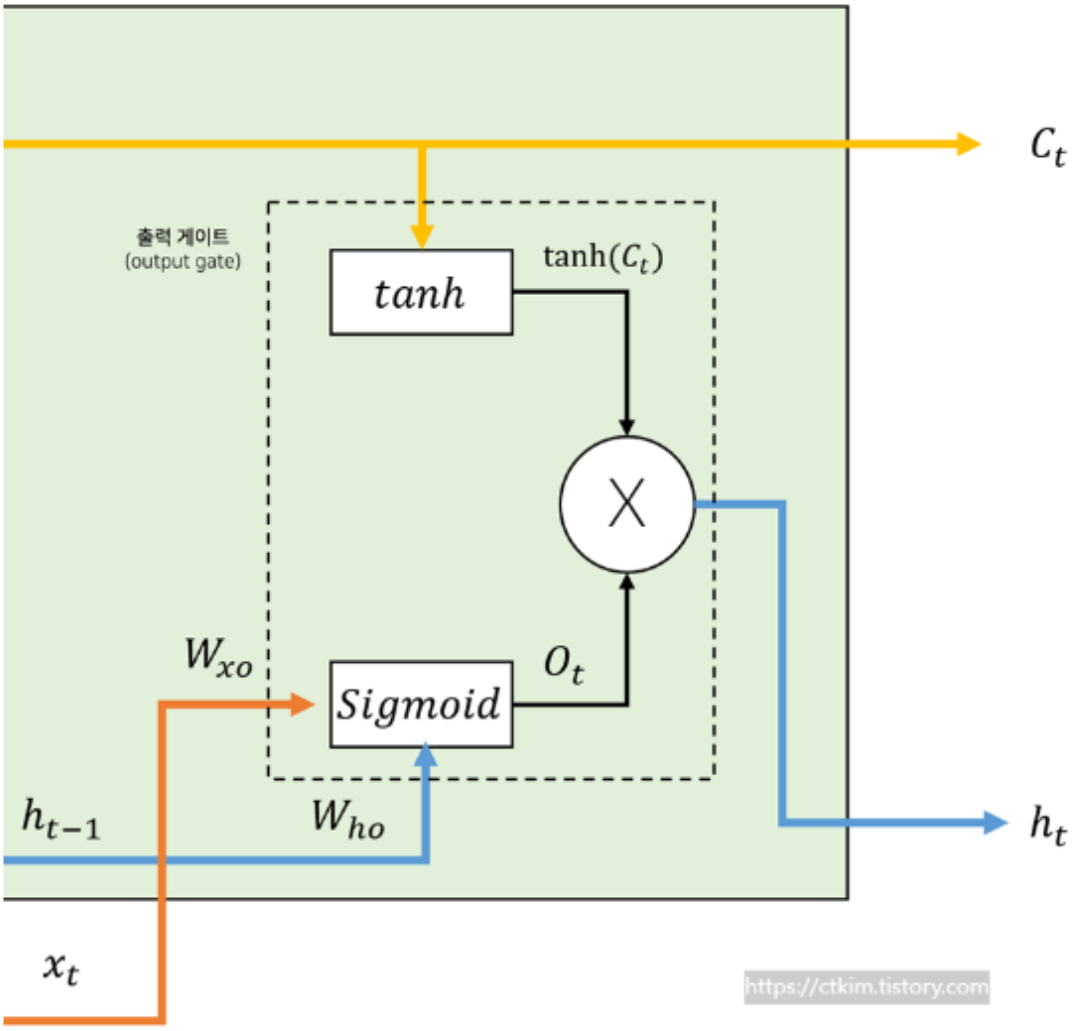
$$C_t = c_{tf} \oplus i_t \otimes \hat{c}_t$$



- **출력 게이트(Output Gate)**: 셀 상태에서 어떤 정보를 출력할지 결정합니다. 값을 그대로 사용하지는 않고, 출력의 중요도에 따라서 Sigmoid 함수를 적용해 0~1 사이 값으로 만든 후 출력하고자 하는 신호에 곱해서 출력한다.

$$O_t = \sigma(W_{ho}h_{t-1} + W_{xo}x_t)$$

$$h_t = Ot \otimes \tanh(c_t)$$



출처 : https://ctkim.tistory.com/entry/LSTMLong-short-time-memory-기초-이해

ㅇ 데이터 셋 수정

1. 영어 LipReading - Miracle-VC1(10개 단어/문장)

2. 한국어 LipReading - 직접 추출
3. Emotions

- AIHub '한국인 감정 인식을 위한 복합 영상' - Sample data(4가지 감정)
- Kaggle 'fer2013'

  https://www.kaggle.com/datasets/deadskull7/fer2013/code?-datasetId=28577&searchQuery=Mobilenet