**실무에 바로 쓰는 AI예측/추천 2주차 수업노트**

22.5.18

2주차 배울 내용

* 각종 파라미터 학습 및 튜닝
* 오토인코더
* CNN 알고리즘
* CNN + LSTM

학습 내용

01. 파라미터 튜닝

02. 파라미터 튜닝 연습

03. 오토인코더

04. CNN

05. CNN 실습=Classification

06. CNN-LSTM

07. CNN-LSTM 실습

08. 2주차 숙제

**01. 파라미터 튜닝**

**파라미터**

* 파라미터: Hidden Layer 개수, 배치 크기, 가중치, 반복회수, 편향 등
* Hidden Layer의 수보다 Hidden Node의 숫자가 더 많아야 효과가 큼
* 하지만, Hidden Node 혹은 Layer가 너무 많으면 Overfitting, 적으면 Underfitting

| multi\_step\_model = tf.keras.models.Sequential() multi\_step\_model.add(tf.keras.layers.LSTM(32, return\_sequences=True, input\_shape=x\_train\_multi.shape[-2:])) multi\_step\_model.add(tf.keras.layers.LSTM(16, activation='relu')) multi\_step\_model.add(tf.keras.layers.Dense(72)) |
| --- |

**배치 사이즈**

한 번의 Batch당 수행되는 데이터 샘플의 Size로, 가용 메모리, eprch 수 등을 고려하여 설정

| **BATCH\_SIZE = 256**  BUFFER\_SIZE = 10000  train\_univariate = tf.data.Dataset.from\_tensor\_slices((x\_train\_uni, y\_train\_uni))  train\_univariate = train\_univariate.cache().shuffle(BUFFER\_SIZE).batch(BATCH\_SIZE).repeat() val\_univariate = tf.data.Dataset.from\_tensor\_slices((x\_val\_uni, y\_val\_uni))  val\_univariate = val\_univariate.batch(BATCH\_SIZE).repeat() |
| --- |

**훈련반복회수(Epoch)**

학습의 반복 숫자를 결정하는 변수로, 전체 Dataset을 1회 학습

1 epoch=BatchSize\*Iteration

| **EPOCHS = 10**  simple\_lstm\_model.fit(train\_univariate, **epochs=EPOCHS**, steps\_per\_epoch=EVALUATION\_INTERVAL, validation\_data=val\_univariate, validation\_steps=50) |
| --- |

**활성화 함수**

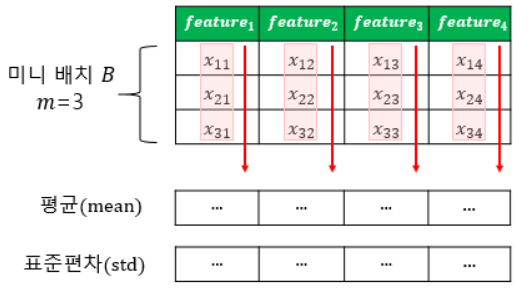
* Sigmoid함수 : (0,1사이, 중간값은 1/2)
* hyperbolic tangent (tanh) 함수→ Sigmoid를 Trnaformation해서 얻어냄
* ReLU함수 (rectified Linear Unit) →최근 가장 많이 사용하는 활성화 함수
* Leaky ReLU 함수

| multi\_step\_model.add(tf.keras.layers.LSTM(16, activation=**'relu'**)) |
| --- |

**배치정규화(Batch Normalization)**

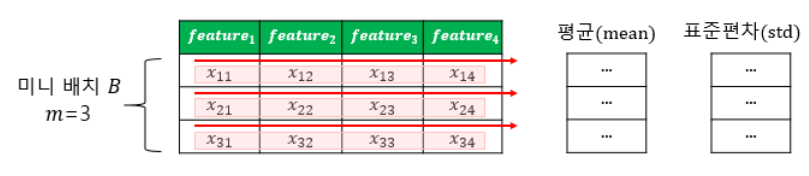
배치정규화: 매순간 Input을 평균0, 표준편차1인 분포로 normalize시켜줌

* 학습의 효율을 높이기 위해 도입
* 활성화 함수의 활성화 값 혹은 출력값을 정규화 하는 과정



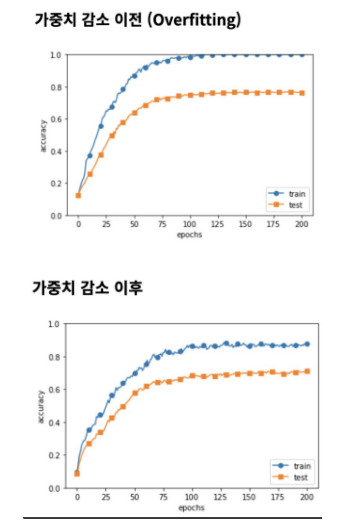
**층 정규화(Layer Normalization)**

층 정규화: 층 단위의 정규화



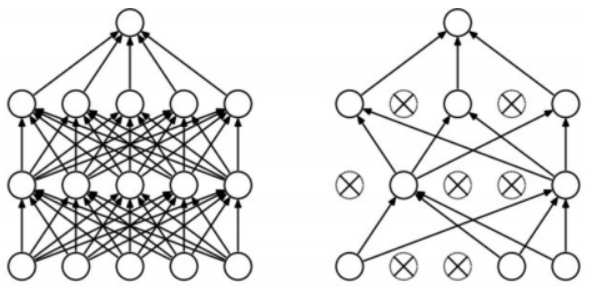
**가중치 규제**

학습중 가중치가 큰 것에 대하여 일종의 패널티를 부과해서 과적합 위험을 줄임 ⇒ 모델이 저장할 수 있는 정보의 양과 종류에 제약을 부과하는 방법



**드롭 아웃(Drop Out)**

신경망 모델이 복잡해질때 neuron의 연결을 임의로 삭제해 신호를 전달 못하도록 설정

****

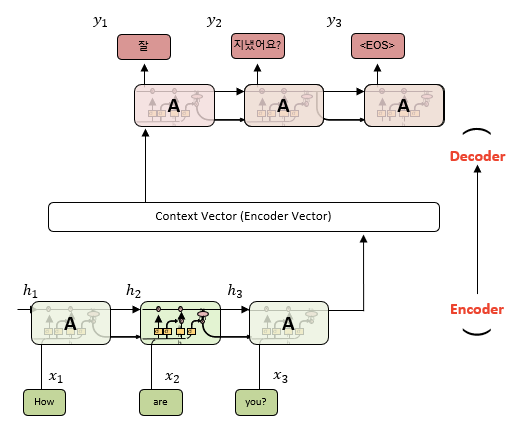
**03. 오토인코더**

오토인코더란 입력데이터의 주어진 input과 output이 유사하도록 학습하는 인공 신경망

* 인코딩: 주어진 input에 대해 원래 차원보다 적은차원으로 표현
* 디코딩: 줄어든 hidden layer값을 다시 원래 차원으로 만들어주는 과정

**LSTM 오토인코더**

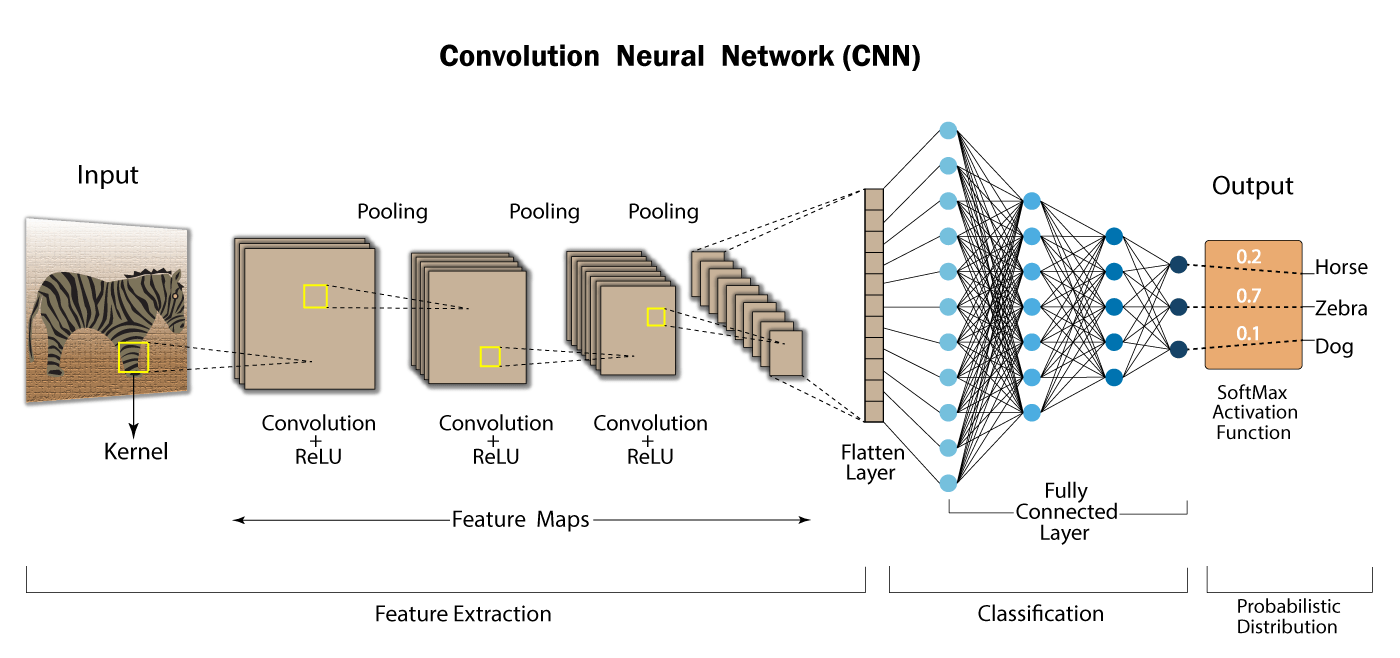
Sequence데이터에 인코더-디코더 LSTM 구조를 적용



**04. CNN**

CNN이란 입력된 이미지를 여러개의 신경층을 거쳐 이미지의 부분적 "특징"을 추출

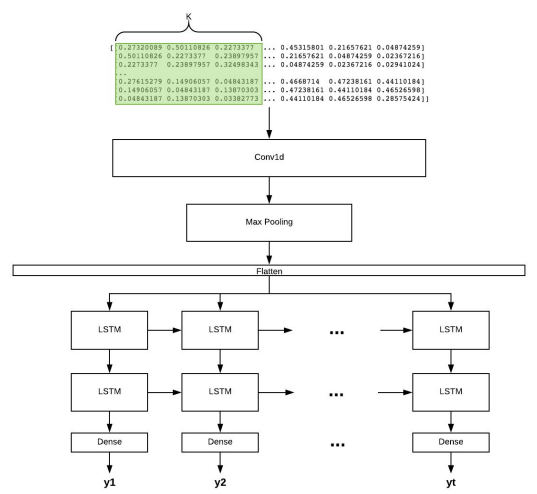
개, 고양이 등 이미지 모델을 인식하는 모델



**06. CNN-LSTM**

CNN-LSTM 모델은 이미지 분류 문제 뿐만 아니라 모든형태의 "Feature"추출 데이터에 사용 가능 (텍스트, 시계열등)

CNN으로 시계열 input data의 Feature를 추출하여 LSTM으로 학습하는 과정



**2주차 수강후기**

개, 고양이를 판단하는데 가장 많이 이용되는 CNN 이론이라고 한다. 하지만 정작 사람은 3~4살이 되면 개와 고양이를 구분할 줄 안다.

인간의 뇌 구조를 연구하는데 이렇게 많은 노력이 따르는데, 정작 80~100년 가까이 되는 한 사람의 인생을 연구하는 데는 얼마만큼의 노력이 필요할까?

열길 물속은 보여도 한길 사람속은 못본다는데, 사람의 마음을 파악하는 인공지능 알고리즘을 개발하는 것은 얼마나 어려울까 라는 생각을 해 본다.

그나저나 인공지능 예측 알고리즘 공부는 너무 어렵다