

고혈압 분석 모델

2021-07-27

컴퓨터과학과 황승현

저번주 한 일

- 2021-07-16
 - 김혜림 박사님과 미팅
 - 질병관리청 교육 수강
- 2021-07-17
 - 데이터 CSV 파일을 일부 수정함
 - 변수 HYPERTENSION 추가
 - 일부 변수 수정
 - Pandas를 이용하여 CSV 파일 다룸
- 2021-07-19
 - Keras 당뇨병 발병 데이터 실습

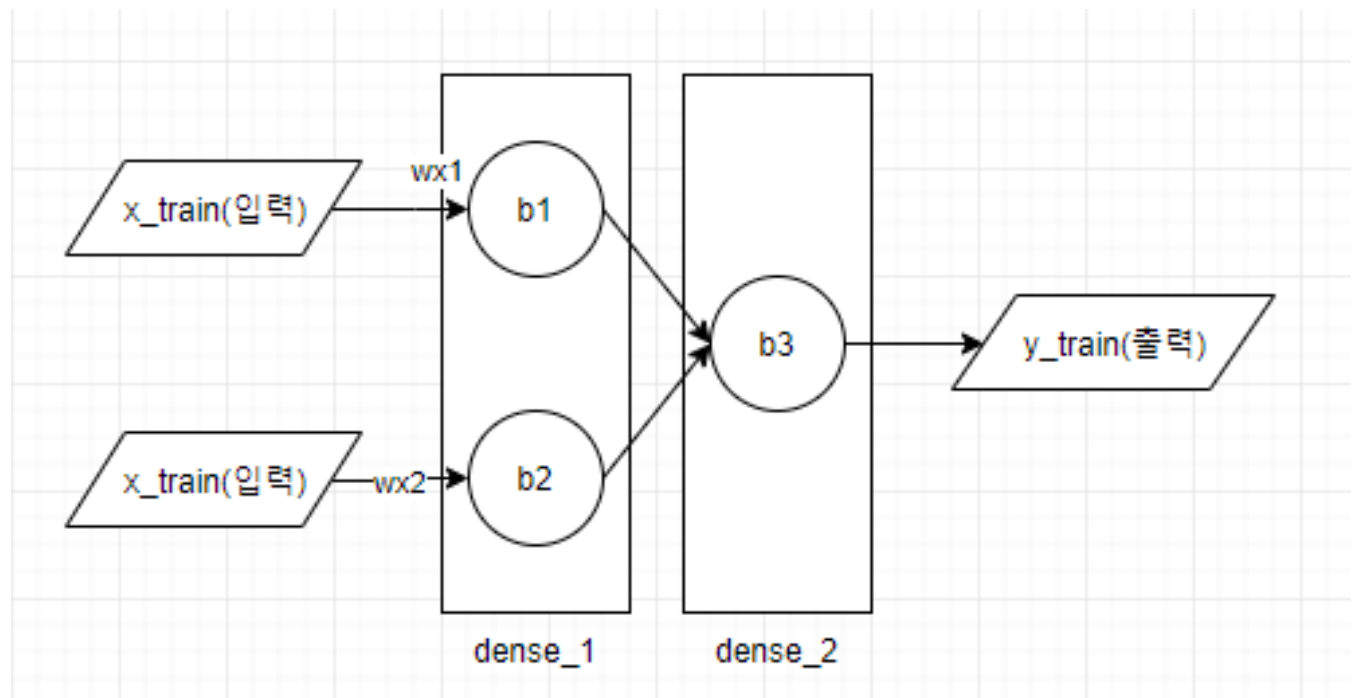
이번주 한 일

- 2021-07-21
 - Keras 당뇨병 실습
 - 새로운 모델을 적용해봄
 - 정확도 80% ~ 90%
- 2021-07-22
 - 당뇨병 모델을 고혈압에 맞게 수정하여 적용
 - 정확도 70% ~ 80%
- 2021-07-23
 - 식품영양학과 김혜림 박사님과 미팅
 - 고혈압 모델 시현
 - 개발 계획 수립

Keras 당뇨병 실습

- <https://partrita.github.io/posts/ML-by-Keras/>
 - 참고 블로그
- `model.add(Dense(15, activation='relu'))`
 - Sequential 모델에 layer 추가
- Dense class
 - $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$
 - NN 레이어

Dense



- Deep Learning에서의 은닉층
- <https://saenggja.tistory.com/356>

Dense

unit: 출력 차원

activation: 활성화 함수

```
tf.keras.layers.Dense(  
    units,  
    activation=None,  
    use_bias=True,  
    kernel_initializer="glorot_uniform",  
    bias_initializer="zeros",  
    kernel_regularizer=None,  
    bias_regularizer=None,  
    activity_regularizer=None,  
    kernel_constraint=None,  
    bias_constraint=None,  
    **kwargs  
)
```

Google Colaboratory

- 온라인에서 사용하는 Jupyter Notebook
 - IPython 작성
- GPU를 포함한 컴퓨팅 리소스를 무료로 제공
- IPython
 - 향상된 파이썬 인터프리터
 - 데이터 과학에서 주로 사용.
- 개발 시간 단축 및 개발 환경에 구애받지 않기 위해 사용
 - 컴퓨터에 TensorFlow 없어도 실행 가능

고혈압 모델

```
[ ] model = Sequential()
model.add(Dense(12, input_dim=52, activation='relu')) # input layer requires input_dim param
model.add(Dense(15, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid')) # sigmoid instead of relu for final probability between 0 and 1

# compile the model, adam gradient descent (optimized)
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=['accuracy'])

# call the function to fit to the data (training the network)
# verbose=0 는 프로그램을 숨깁니다.
history = model.fit(X_train, y_train, epochs=100, batch_size=10, verbose=0)
# print(history.history)

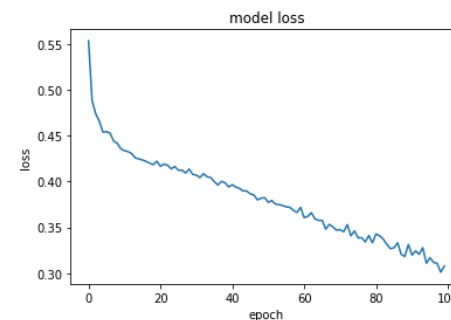
# 모델의 정확도를 계산합니다.
scores = model.evaluate(X_test, y_test)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1] * 100))

4/4 [=====] - 0s 3ms/step - loss: 0.6595 - accuracy: 0.7455
accuracy: 74.55%
```

```
[ ] fig, ax1 = plt.subplots()

# 모델의 오차를 그래프입니다.
ax1.plot(history.history['loss'])
ax1.set(title='model loss', xlabel='epoch', ylabel='loss')
```

[Text(0, 0.5, 'loss'), Text(0.5, 0, 'epoch'), Text(0.5, 1.0, 'model loss')]

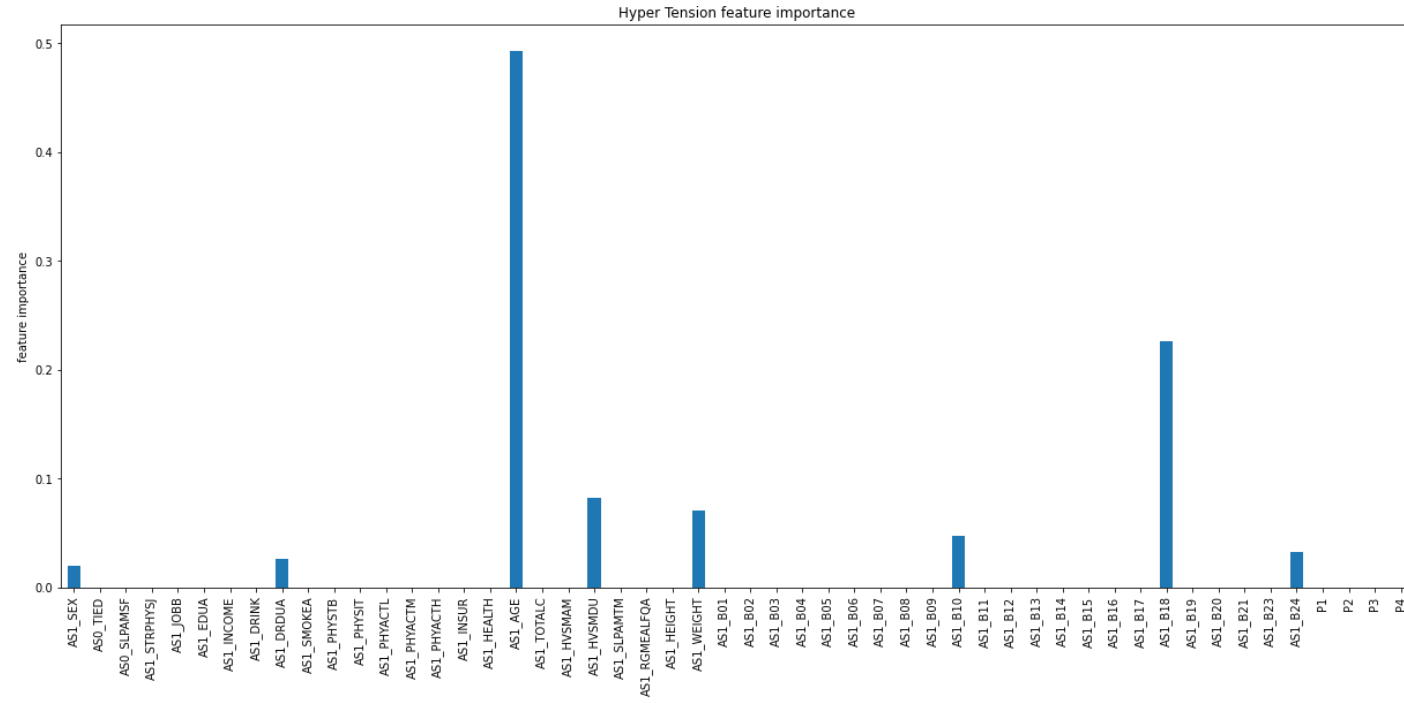


개발 계획

- 모델 정확도 향상
 - Layer 가중치 수정, decision tree
 - 목표: 90% 이상
- Decision tree
 - 변수 속아내기, 결과에 영향을 주지않는 변수 제거
 - 모델 정확도 향상
- 기반 년도, 2차, 5차, 8차 연구에서의 고혈압 발병 유무 비교
 - 향후 10년 안에 고혈압에 걸릴 확률 분석
 - 추후 연구로 진행

다음주 할 일

- 모델 튜닝
 - 하이퍼파라미터 조정
 - GridSearch, RandomSearch
- Decision tree 설계
 - Decision tree 개념 공부
 - Decision tree 예제 실습
 - Decision tree 고혈압 모델에 적용
- 김혜림 박사와 미팅



Feature Importance

- 나이, 비타민A1 섭취량과 상관관계 있음
- 결과의 속 뜻은 연구해봐야함