

# 2차 진도 보고서

A+

## 팜테크

팜(Farm)+재테크



소프트웨어융합학과 2018111575

김가을

소프트웨어융합학과 2018111395

박서영



농산물 가격 예측 AI 모델 개발 및 시각화를 통해 가격을 예측하고  
소비자들에게 제철 재료 및 합리적인 가격의 농산물을 제안하는 앱을  
개발한다.

-> 최종적으로 소비자들의 합리적인 소비 지향

수는 한 개로, 신경망 학습이 끝나면 이로부터 전기 판매단가의 예측된 출력 값을 얻는다[10].

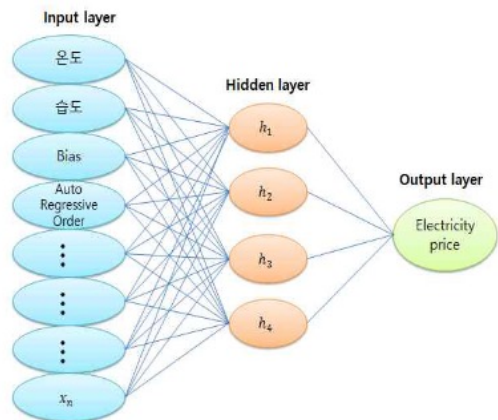


그림 2. 인공신경망을 이용한 날씨별 전기가격 예측 모델  
Fig. 2. An example of the ANN network composed of weather related indexes

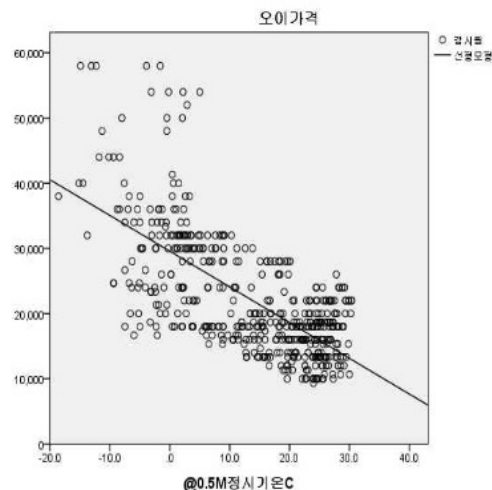


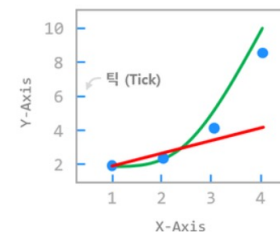
그림 3. 0.5M 정시기온과 오이가격간의 관계  
Fig. 3. Relation of 0.5M temperature with cucumber price

기후와 가격 간의 상관 관계를 분석한 후,  
인공신경망(ANN) 모델에 학습시킨 연구 참고  
(<http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07079511>)

Matplotlib Tutorial - ... / 13. Matplotlib 눈금 표시하기

WikiDocs

### 13. Matplotlib 눈금 표시하기



틱 (Tick)은 그래프의 축에 간격을 구분하기 위해 표시하는 눈금입니다.

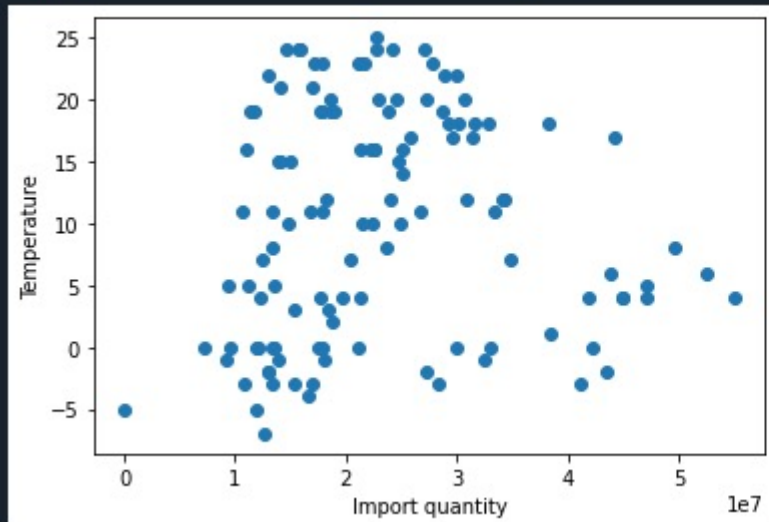
matplotlib.pyplot 모듈의 `xticks()`, `yticks()`, `tick_params()` 함수를 이용해서

그래프에 눈금을 표시하는 방법에 대해 소개합니다.

Keyword: `plt.xticks()`, `plt.yticks()`, `plt.tick_params()`, tick, 눈금 스타일

데이터 시각화를 위한 Matplotlib 모듈 활용에 대한  
공부

```
In [5]: runfile('C:/Users/8305-01/.spyder-py3/visualization.py', wdir='C:/Users/8305-01/.spyder-py3')
```



기온과 반입량의 관계를 시각화

이처럼, 각각의 데이터를 서로 비교해서 특징 벡터로 선택한 근거를 마련함

## 선형 회귀를 활용한 가격 예측 AI 개발 중에 있음

```
####
xy = np.array(data)
print(xy)

x_data = xy[:, 0:-1] ## A,B 날짜, 반입량.
y_data = xy[:, [-1]] ## C 가격.

tf.compat.v1.disable_eager_execution()
X = tf.compat.v1.placeholder(tf.float32, shape=[None, 2])
Y = tf.compat.v1.placeholder(tf.float32, shape=[None, 1])

# 가중치 값 초기화
W = tf.Variable(tf.compat.v1.random_normal([2, 1]), name="weight")

# bias 값 초기화
b = tf.Variable(tf.compat.v1.random_normal([1]), name="bias")

# 선형 회귀의 경우 행렬의 곱 연산을 이용하여 결과식을 세울 수 있다
hypothesis = tf.matmul(X, W) + b

# 비용함수 : tensorflow에서 기본적으로 제공해주는 reduce_mean 사용
cost = tf.reduce_mean(tf.square(hypothesis - Y))

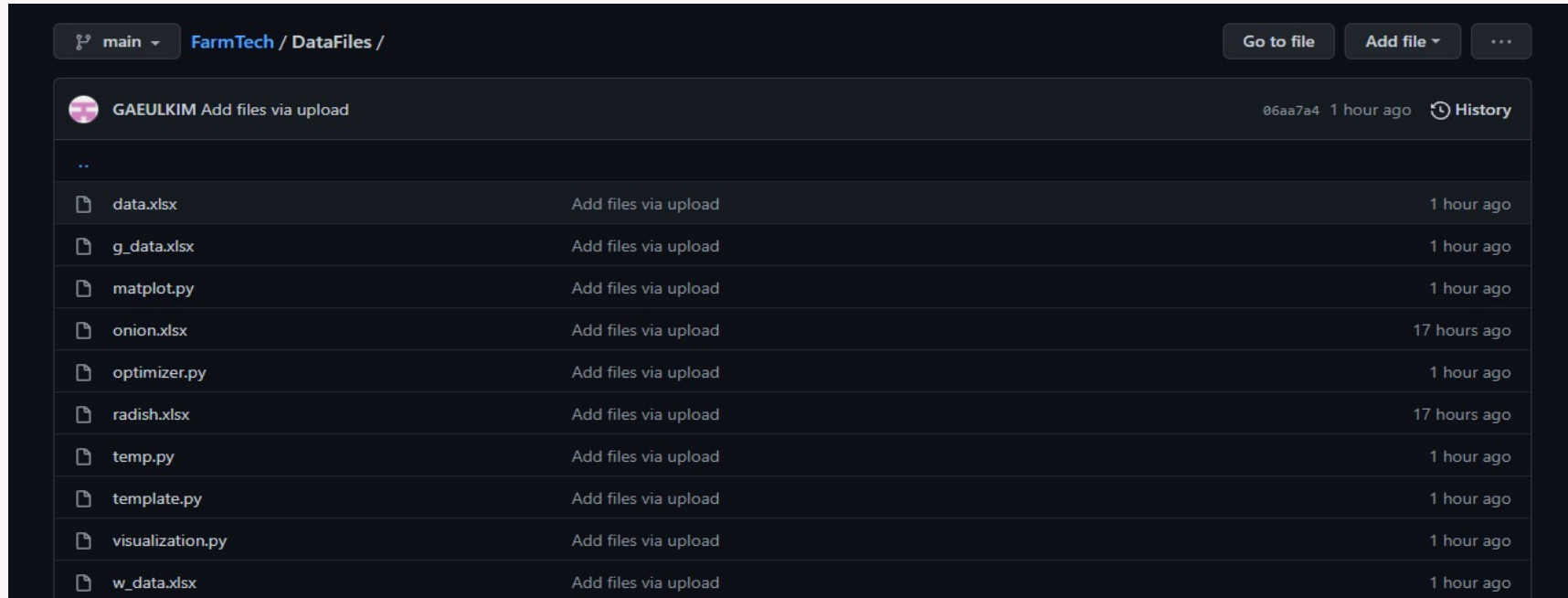
# 최적화 함수 : 경사하강법 : 학습률 0.000005로 세팅
optimizer = tf.compat.v1.train.GradientDescentOptimizer(learning_rate = 0.000005)

# cost를 최소화하는 최적값 계산
train = optimizer.minimize(cost)

# 세션 생성 및 초기화
sess = tf.compat.v1.Session()
init = tf.compat.v1.global_variables_initializer()
sess.run(init)

# 1만번 트레이닝 진행
for step in range(10001):
    hypo2, cost2, third = sess.run([hypothesis, cost, train], feed_dict={X: x_data, Y: y_data})
    if step % 500 == 0:
        print("#", step, "손실 비용: ", cost)
        print("- 양파 가격: ", hypo2[0])
```

```
# 0 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [-7569625.5]
# 500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 1000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 1500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 2000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 2500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 3000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 3500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 4000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 4500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 5000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 5500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 6000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 6500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 7000 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
# 7500 손실 비용: Tensor("Mean_2:0", shape=(), dtype=float32)
- 양파 가격: [nan]
```



수집한 데이터, 계획, 데이터 전처리 기준 등을 깃허브에 정리해  
서 공유 중

1. 5대 품목 위주의 데이터로 신경망 학습을 시켜 가격 예측 모델 만들어보기
2. 선형 회귀를 활용한 가격 예측 모델 완성하기
3. AI 모델의 특징 벡터 3가지 이상 넣어서 학습시키기
4. 모든 결과 데이터 깃허브에 공유하기