Optimal Bidding

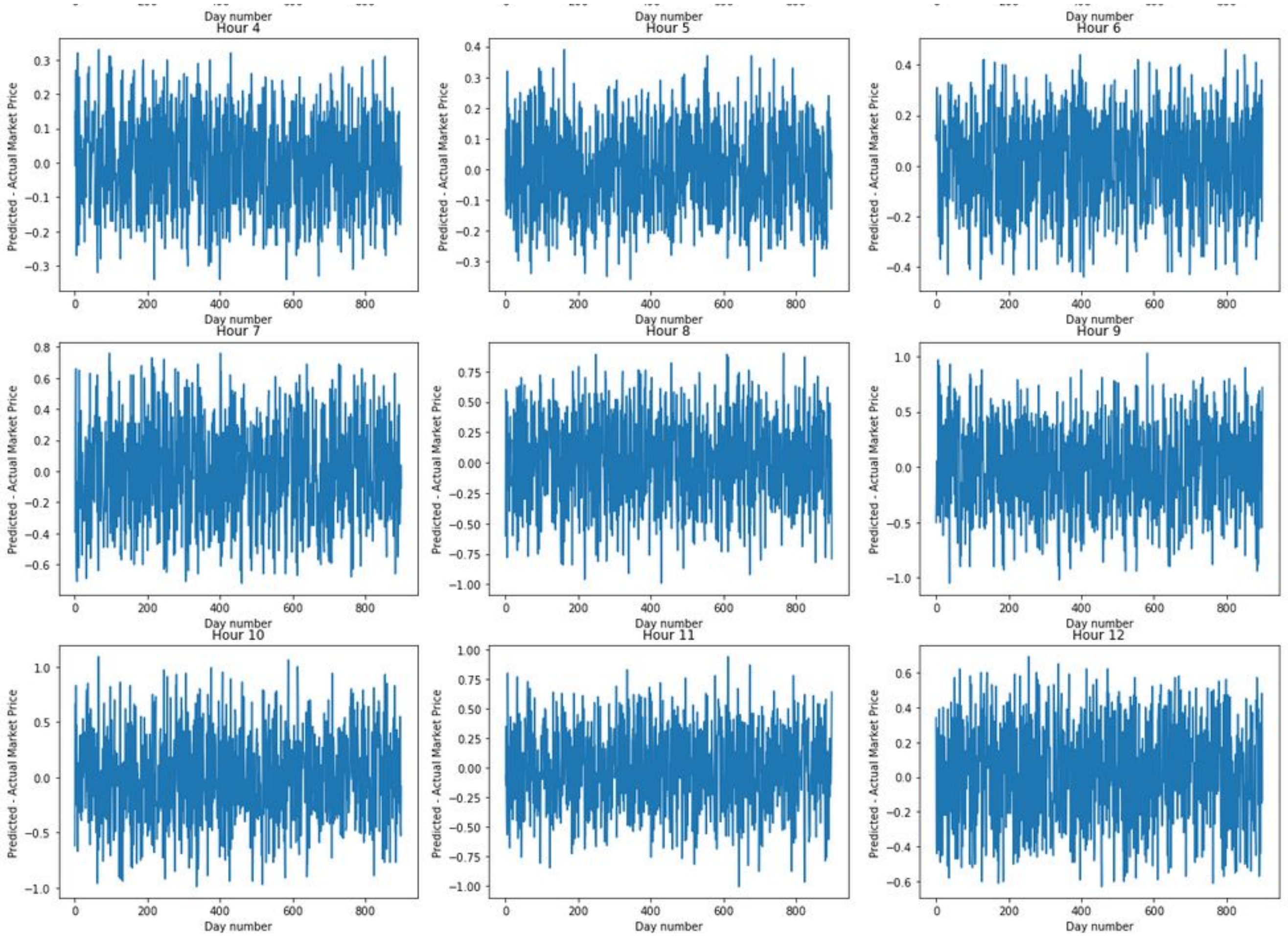
Bid price

Bid price and bill

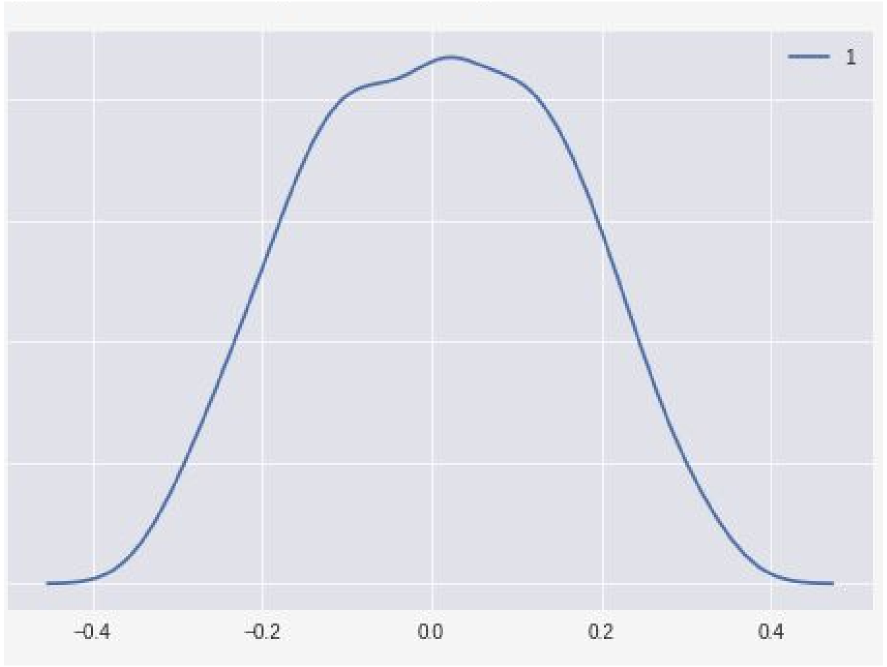
- Losing bid drastically increases bill as DISCOM’s electricity rate is much higher than the market rate.
- To always win bid, our bid price must be greater than actual market price.

Oracle prediction of market price vs Actual market price

Error in oracle’s prediction for market price is symmetric about zero.

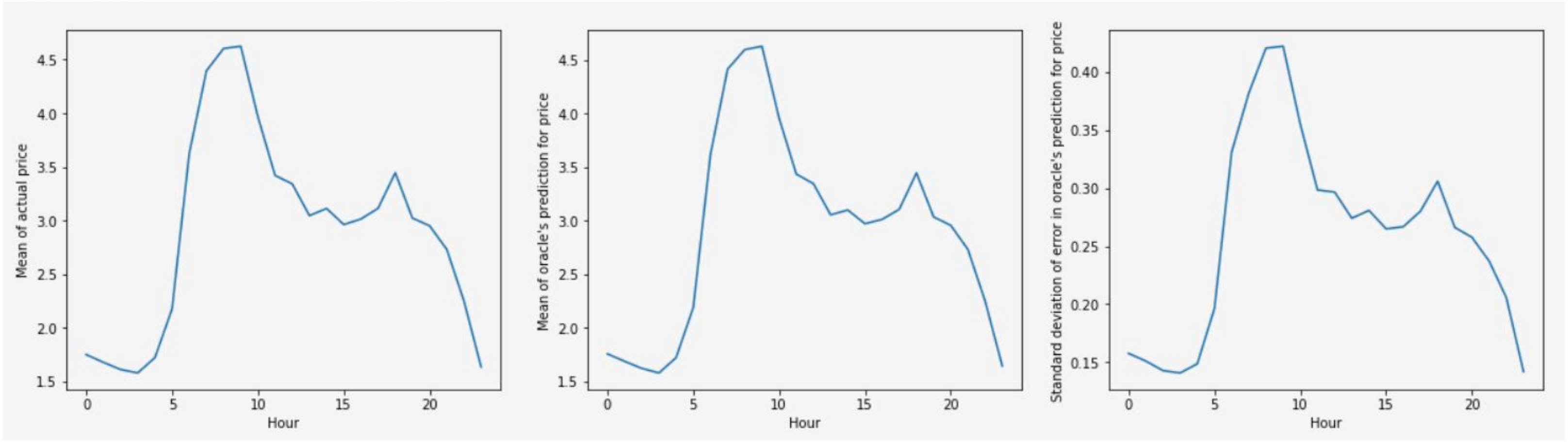

Day number
Hour 4
Hour 5
Hour 6
Hour 7
Hour 8
Hour 9
Hour 10
Hour 11
Hour 12
Predicted - Actual Market Price
Day number
0
200
400
600
800

Error in oracle’s prediction for market price is normally distributed.



Density plot of error of oracle’s prediction for market price

Variance of error distribution is directly proportional to the price.


Mean of actual price
Hour
Mean of oracle's prediction for price
Hour
Standard deviation of error in oracle's prediction for price
Hour

Bid price observation

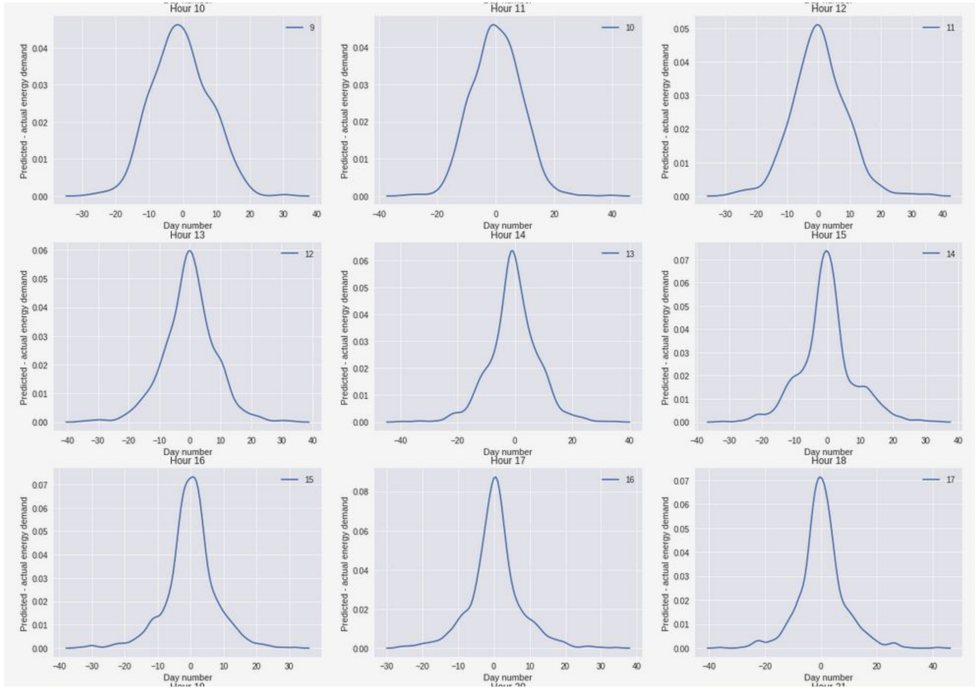
- If we exactly bid at oracle’s predicted market price, we will lose bid half of the times.
- If we lose, we will have to take electricity from DISCOM at much higher rate.
- To make sure we always win, we will add some value to oracle’s prediction for market price.
- As the error in predicted price is directly proportional to the price, the value we add will be directly proportional to the price.
- bid_price = oracle_price_prediction + c * oracle_price_prediction
- bid_price = k * oracle_price_prediction
- We will find the value of k so as to minimise the bill.

Bid quantity

Bid quantity and bill

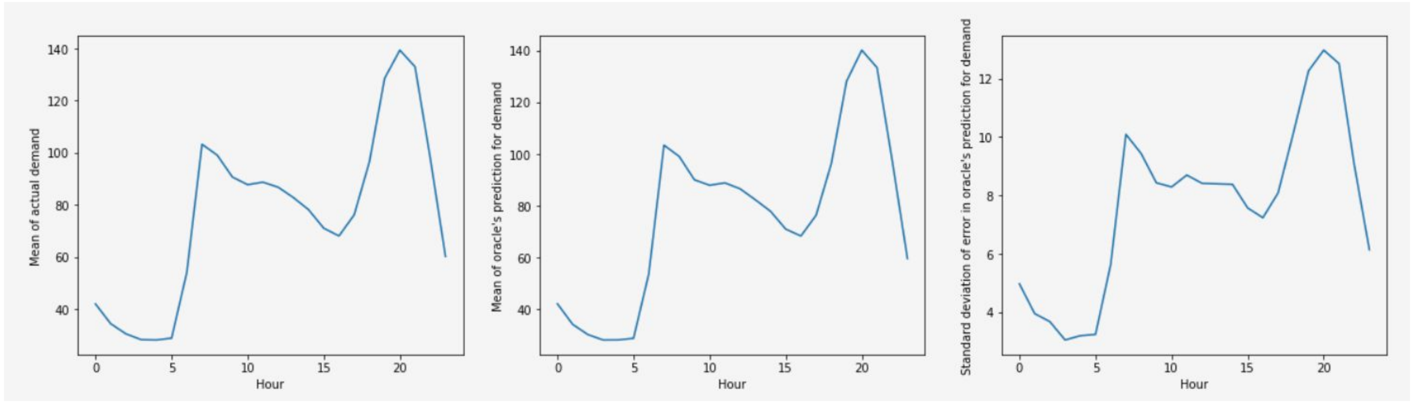
- We cannot beforehand assume if we want to bid more or less than the predicted difference of demand and solar output as there is battery involved.
- The battery acts as a buffer. It can store energy when there is extra energy. It can used when there is shortage of electricity.
- The most optimal way to bid is to bid more when price is low and store energy in battery and bid less when price is high and used energy stored in battery.

Error in oracle’s prediction for demand is normally distributed.



Density plot of error of oracle’s prediction for demand

Variance of error distribution is directly proportional to the demand.

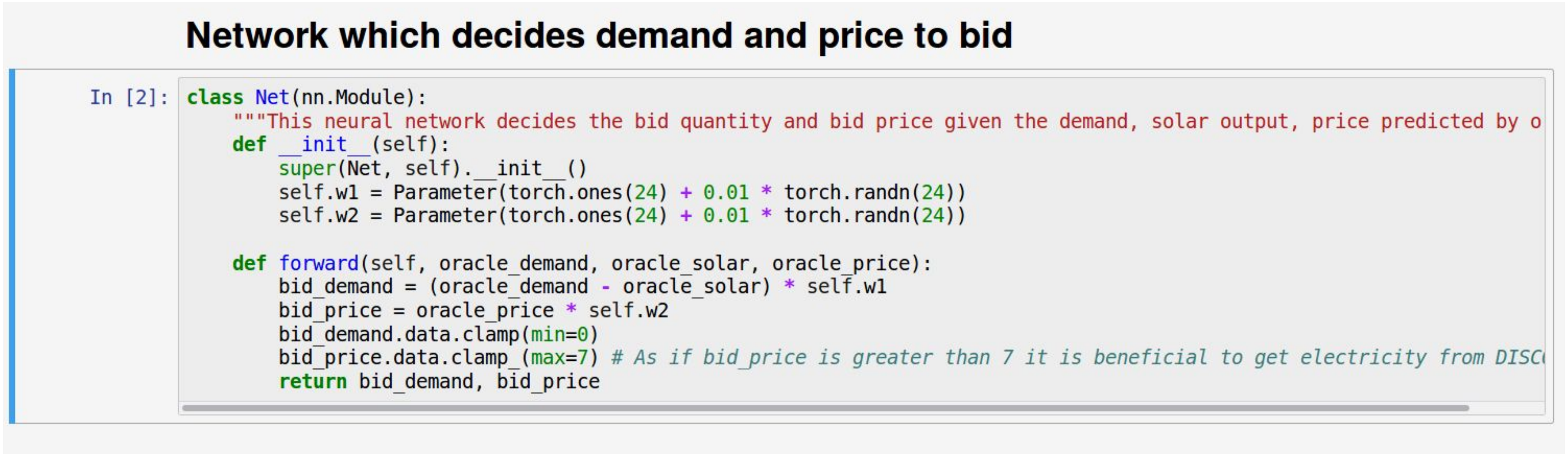

Mean of actual demand
Hour
Mean of oracle's prediction for demand
Hour
Standard deviation of error in oracle's prediction for demand
Hour

Bid quantity observation

- When the market price is low, we need to bid more and vice versa for optimal use of battery.
- As the error in predicted residual demand is directly proportional to the residual demand, the value we add will be directly proportional to the residual demand.
- bid_quantity = oracle_demand_prediction + c * oracle_demand_prediction
- bid_quantity = k * oracle_demand_prediction
- We will find the value of k so as to minimise the bill.

Model

- For each hour, we multiply the market price and the residual demand (demand - solar_output) with different values to generate bid price and bid quantity.
- This is because market price, demand, solar output are different for each hour, therefore each hour needs to be considered separately.



Network which decides demand and price to bid

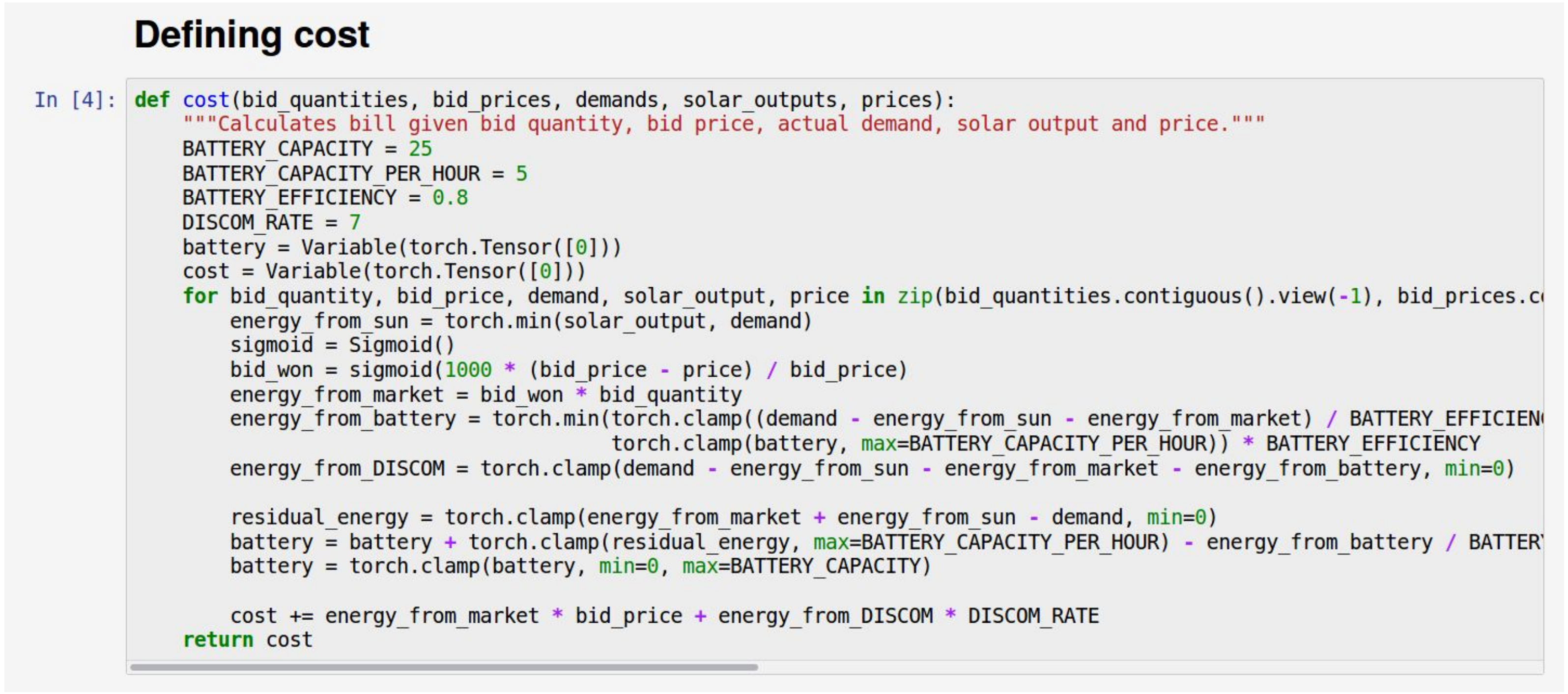
```
In [2]: class Net(nn.Module):
            """This neural network decides the bid quantity and bid price given the demand, solar output, price predicted by o
            def __init__(self):
                super(Net, self).__init__()
                self.w1 = Parameter(torch.ones(24) + 0.01 * torch.randn(24))
                self.w2 = Parameter(torch.ones(24) + 0.01 * torch.randn(24))

            def forward(self, oracle_demand, oracle_solar, oracle_price):
                bid_demand = (oracle_demand - oracle_solar) * self.w1
                bid_price = oracle_price * self.w2
                bid_demand.data.clamp(min=0)
                bid_price.data.clamp_(max=7) # As if bid_price is greater than 7 it is beneficial to get electricity from DISC
                return bid_demand, bid_price
```

Network to decide bid quantity and bid price for minimising bill.

How to find these parameters to minimise bill?

Finding parameters for optimal bidding

- We write a neural network which decides bid quantity and bid price.
- We train this network by minimising cost function. The cost function is the bill for that period of time.



Defining cost

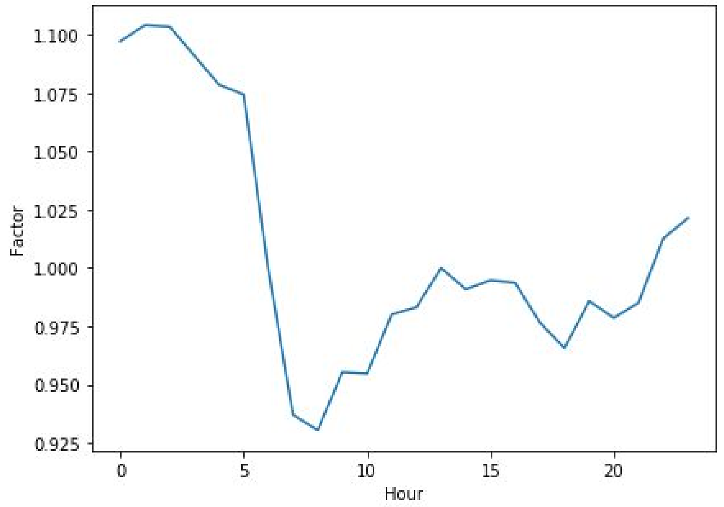
```
In [4]: def cost(bid_quantities, bid_prices, demands, solar_outputs, prices):
            """Calculates bill given bid quantity, bid price, actual demand, solar output and price."""
            BATTERY_CAPACITY = 25
            BATTERY_CAPACITY_PER_HOUR = 5
            BATTERY_EFFICIENCY = 0.8
            DISCOM_RATE = 7
            battery = Variable(torch.Tensor([0]))
            cost = Variable(torch.Tensor([0]))
            for bid_quantity, bid_price, demand, solar_output, price in zip(bid_quantities.contiguous().view(-1), bid_prices.c
                energy_from_sun = torch.min(solar_output, demand)
                sigmoid = Sigmoid()
                bid_won = sigmoid(1000 * (bid_price - price) / bid_price)
                energy_from_market = bid_won * bid_quantity
                energy_from_battery = torch.min(torch.clamp((demand - energy_from_sun - energy_from_market) / BATTERY_EFFICIEN
                                                torch.clamp(battery, max=BATTERY_CAPACITY_PER_HOUR)) * BATTERY_EFFICIENCY
                energy_from_DISCOM = torch.clamp(demand - energy_from_sun - energy_from_market - energy_from_battery, min=0)

                residual_energy = torch.clamp(energy_from_market + energy_from_sun - demand, min=0)
                battery = battery + torch.clamp(residual_energy, max=BATTERY_CAPACITY_PER_HOUR) - energy_from_battery / BATTER
                battery = torch.clamp(battery, min=0, max=BATTERY_CAPACITY)

                cost += energy_from_market * bid_price + energy_from_DISCOM * DISCOM_RATE
            return cost
```

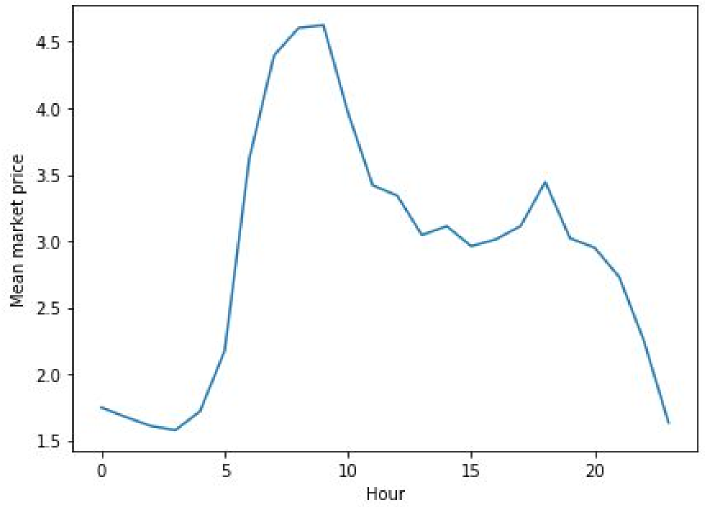
Cost function

Analysis

Learned parameters

- Demand Coefficients: [1.09723032 1.10422564 1.10357499 1.0911119 1.07863581 1.07440257 0.99930233 0.93699789 0.93034673 0.95528203 0.95467639 0.98006302 0.98314083 0.99996793 0.99084544 0.99470854 0.99364167 0.9765951 0.96554035 0.98577815 0.97856957 0.98506409 1.01262844 1.0213747],
  Price Coefficients: [1.18369961 1.18209887 1.17976892 1.17646837 1.18019557 1.17607522 1.17634344 1.1575532 1.16714895 1.18082559 1.17663801 1.17372763 1.17879426 1.18214524 1.17882335 1.17696249 1.17962289 1.1733129 1.1747092 1.19124782 1.16645944 1.17419469 1.17935419 1.17650652]





Factor of demand vs Mean market price

Observations

- Model bids more when price is low and bids less when price is high making optimal use of battery.
- As expected model makes bid price such that bid is always won to prevent getting electricity from DISCOM at higher rates.