Q2.3

To calculate gradient for $J(\theta) = \frac{1}{2}\sum_j w^{(j)}(y^{(j)} - \theta^T x^{(j)})^2$

$$\begin{aligned}
\frac{\partial}{\partial\theta_i} J(\theta) &= \sum_j \frac{\partial}{\partial\theta_i}\frac{1}{2} w^{(j)}(y^{(j)} - \theta^T x^{(j)})^2 \\
&= \sum_j 0 + w^{(j)} \cdot \frac{1}{2} \times 2 (y^{(j)} - \theta^T x^{(j)}) \cdot \frac{\partial}{\partial\theta_i}(y^{(j)} - \theta^T x^{(j)}) \\
&= \sum_j w^{(j)} \cdot (y^{(j)} - \theta^T x^{(j)}) \cdot \frac{\partial}{\partial\theta_i}(y^{(j)} - (\theta_0 x_0^{(j)} + \cdots \theta_n x_n^{(j)})) \\
&= -\sum_j w^{(j)} \cdot (y^{(j)} - \theta^T x^{(j)}) \cdot x_i^{(j)} \\
&= \sum_j w^{(j)} \cdot (\theta^T x^{(j)} - y^{(j)}) \cdot x_i^{(j)}
\end{aligned}$$

An algorithm for calculating θ by batch gradient decent for locally weighted linear regression.

Step 1: For each input x in input vector X, define weighting function $w^{(j)}$:

$$w^{(j)} = \exp\left(- \frac{(x - x^{(j)})^T \cdot (x - x^{(j)})}{2\tau^2}\right)$$

Step 2: Update θ based on batch gradient decent.

$$\theta_i := \theta_i - \alpha \underbrace{\sum_{j=1}^{m} w^{(j)}(y^{(j)} - \theta^T x^{(j)}) \cdot x_i^{(j)}}_{\frac{\partial}{\partial\theta_i} J(\theta)}$$

Step 3: Repeat Step 2 till convergence.

Step 4: Output $\theta^T x$

Step 5: Repeat Step 1 - Step 4 for each x in input vector X

Q2.3-2

Locally weighted linear regression is a non-parametrix algorith, The model doesn't learn a fixed set of parameters as is done in ordinary linear regression. Rather parameters θ are computed for each query point separately.