# HW1

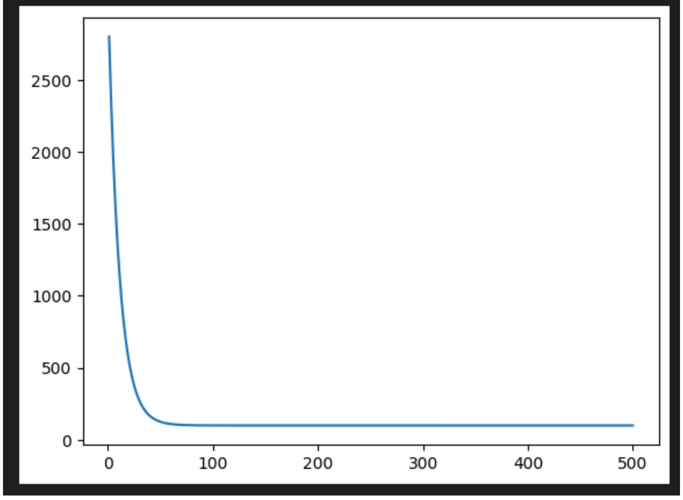
## Part. 1, Coding (60%):

**Linear regression model**

```
#Ans
plt.plot(np.arange(1, len(MSE) + 1), MSE)
print(f'Mean_square_error: {Mean_square_error}')
#y = intercapts + weights * x
print(f'weights: {weights}, intercepts: {intercepts}')
```

✓ 0.1s

```
Mean_square_error: [110.43819255]
weights: [52.74354046], intercepts: [-0.3337589]
```

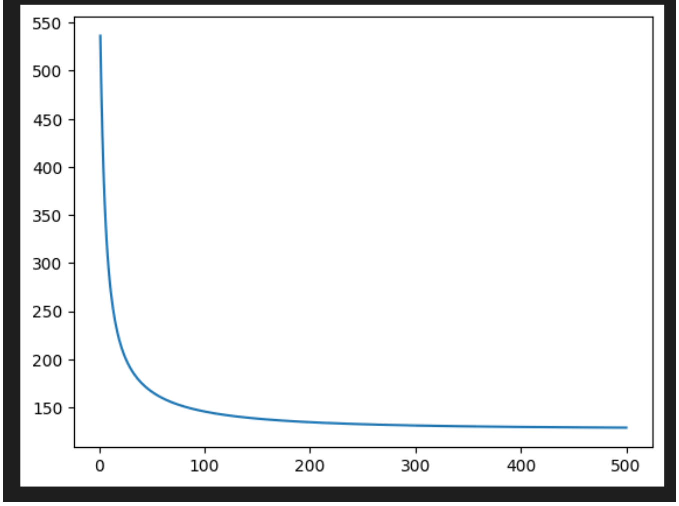



**Logistic regression model**

```
#Ans
plt.plot(np.arange(1, len(CEE) + 1), CEE)
print(f'Cross Entropy Error: {Cross_entropy_error}')
#y = intercapts + weights * x
print(f'weights: {weights}, intercepts: {intercepts}')
```

✓ 0.2s

```
Cross Entropy Error: [45.69575431]
weights: [4.34700661], intercepts: [1.39009528]
```




## Part. 2, Questions (40%):

### 1. What's the difference between Gradient Descent, Mini-Batch Gradient Descent, and Stochastic Gradient Descent?

Gradient Descent: 每次迭代都考慮全部的 training data 。

Mini-Batch Gradient Descent: 每次迭代只隨機從 training data 中選多筆 (batch size) 資料來考慮。

Stochastic Gradient Descent: 每次迭代只隨機從 training data 中選一筆資料來考慮。

當資料太多，(Gradient Descent) 每次迭代都考慮全部的資料會花太多時間，(Stochastic Gradient Descent) 每次迭代只考慮一筆資料的話不會每次都往最優點前進，Mini-Batch Gradient Descent 為折衷方案。

### 2. Will different values of learning rate affect the convergence of optimization? Please explain in detail.

learning rate 控制我們根據 Gradient Descent 來調整 weights 的程度大小。太小的 learning rate 一次調整的比例太小，會收斂的很慢，需要迭代更多的次數才能達到理想的成果；相反太高的 learning rate 一次調整的比例太大，則會導致無法收斂至最優值，而且超大的 learning rate 可能會導權重一直跟著最新的 Gradient Descent ，結果發散而非收斂。

### 3. Show that the logistic sigmoid function (eq. 1) satisfies the property σ(−a) = 1 − σ(a) and that its inverse is given by σ−1(y) = ln {y/(1 − y)}.

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \qquad \text{(eq. 1)}$$

$1-\sigma(a)=\frac{1+\exp(-a)}{1+\exp(-a)}-\frac{1}{1+\exp(-a)}=\frac{\exp(-a)}{1+\exp(-a)}=\frac{1}{\frac{1}{\exp(-a)}+1}=\frac{1}{exp(-a)^{-1}+1}=\frac{1}{exp(a)+1}=\sigma(-a)$

Proven $\sigma(-a)=1-\sigma(a)$.

$y=\sigma(x)=\frac{1}{1+exp(-x)}$

$1+exp(-x)=1/y$

$exp(-x)=1/y-1$

$exp(-x)=1/y-y/y$

$exp(-x)=(1-y)/y$

$ln(exp(-x))=ln((1-y)/y)$

$-x=ln((1-y)/y)$

$x=-ln((1-y)/y)$

$\sigma^{-1}(y)=x=ln(y/(1-y))$

Proven $\sigma^{-1}(y)=ln(y/(1-y))$

### 4. Show that the gradients of the cross-entropy error (eq. 2) are given by (eq. 3).

$$E(\mathbf{w}_1,\ldots,\mathbf{w}_K)=-\ln p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K)=-\sum_{n=1}^{N}\sum_{k=1}^{K}t_{nk}\ln y_{nk} \qquad \text{(eq. 2)}$$

$$\nabla_{\mathbf{w}_j}E(\mathbf{w}_1,\ldots,\mathbf{w}_K)=\sum_{n=1}^{N}(y_{nj}-t_{nj})\,\boldsymbol{\phi}_n \qquad \text{(eq. 3 )}$$

Hints:

$$a_k=\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}. \qquad \text{(eq. 4)}$$

$$\frac{\partial y_k}{\partial a_j}=y_k(I_{kj}-y_j) \qquad \text{(eq. 5)}$$

According to eq. 2, we have $\frac{\partial E}{\partial y_{nk}}=-\frac{t_{nk}}{y_{nk}}$. (eq. 6)

According to eq. 4, we have $\bigtriangledown_{wj}a_{nj}=\phi_n$. (eq. 7)

Given eq.5 and eq. 6, we can compute

$\frac{\partial E}{\partial a_{nj}}=\Sigma_{k=1}^{K}\frac{\partial E}{\partial y_{nk}}\frac{\partial y_{nk}}{\partial a_{nj}}=-\Sigma_{k=1}^{K}\frac{t_{nk}}{y_{nk}}y_{nk}(I_{kj}-y_{nj})=-\Sigma_{k=1}^{K}t_{nk}(I_{kj}-y_{nj})=-t_{nj}+\Sigma_{k=1}^{K}t_{nk}y_{nj}=y_{nj}-t_{nj}$. (eq. 8)

Given eq. 7 and eq. 8, we can compute

$\bigtriangledown_{wj}E(w_1,\ldots,w_k)=\Sigma_{n=1}^{N}\frac{\partial E}{\partial a_{nj}}\bigtriangledown_{wj}a_{nj}=\Sigma_{n=1}^{N}(y_{nj}-t_{nj})\phi_n$

Proven the gradients of the cross-entropy error (eq. 2) are given by (eq. 3).