Problem 1

Problem 1.1

$$\begin{aligned} \min \quad & x_1^2 + 12x_2^2 - 1 \\ \text{s.t.} \quad & 3x_1 + x_2 - 1 = 0 \\ & x_1 - 2x_2 \geq 0 \end{aligned}$$

First of all, we need to change the last constraint equation, thus getting:

$$\begin{aligned} \min \quad & x_1^2 + 12x_2^2 - 1 = f(x_1, x_2) \\ \text{s.t.} \quad & 3x_1 + x_2 - 1 = 0 = g(x_1, x_2) \\ & 2x_2 - x_1 \leq 0 = h(x_1, x_2) \end{aligned}$$

Solution

$$\mathcal{L}(x_1, x_2, \lambda_1, \lambda_2) = x_1^2 + x_2^2 + \lambda_1(x_1 + x_2 - 1) + \lambda_2(2x_2 - x_1)$$

$$\begin{aligned} &\mathcal{L}x_1 = 2x_1 + \lambda_1 - \lambda_2 = 0 \\ &\mathcal{L}x_2 = 2x_2 + \lambda_1 + 2\lambda_2 = 0 \\ &\mathcal{L}\lambda_1 = x_1 + x_2 - 1 = 0 \\ &\mathcal{L}\lambda_2 = 2x_2 + x_1 = 0 \end{aligned}$$

substitute x_1 and x_2

$$\begin{aligned} &x_1 = \tfrac{\lambda_2}{2} - \tfrac{\lambda_1}{2} \\ &x_2 = -\lambda_2 - \tfrac{\lambda_2}{2} \\ &\mathcal{L}\lambda_1 = \tfrac{-\lambda 2 - 2\lambda_1}{2} - 1 = \\ &= \tfrac{-\lambda_2}{2} = 1 + \tfrac{2\lambda_1}{2} = -2 - 2\lambda_1 \\ &\text{substitute } \lambda_1 \\ &\mathcal{L}\lambda_2 = 4 + 4\lambda_1 - \lambda_1 + 1 + \lambda_1 + \tfrac{\lambda_1}{2} = \\ &= 5 + 4\lambda_1 + \tfrac{\lambda_1}{2} = \tfrac{-10}{9} \end{aligned}$$

In the end we have:

$$\lambda_1 = \tfrac{-10}{9}$$

$$\lambda_2 = \tfrac{2}{9}$$

$$x_1 = \tfrac{2}{3}$$

$$x_2 = \tfrac{1}{3}$$

We can verify the limitations by inserting the values of x_1 and x_2:

$$g(x_1, x_2) = \tfrac{2}{3} + \tfrac{1}{3} - 1 = 0$$

$$h(x_1, x_2) = 2\tfrac{1}{3} - (\tfrac{2}{3}) = 0$$

$$f(x_1, x_2) = \tfrac{2}{3}^2 + \tfrac{1}{3}^2 - 1 = \tfrac{4+1-9}{9} = \tfrac{-4}{9}$$

Problem 1.2

Solution We can divide our problem into two parts:

- Get head first
- Get k consecutive tails

Let's start with calculating the expected tosses to get k consecutive tails.
Let e be the number of expected tosses. The probability of getting tail is $p = \frac{1}{2}$,
if we get head first, then the expected number will be $e + 1$,
if we get head then tails, we will have $p = \frac{1}{2}\frac{1}{2} = \frac{1}{4}$ and $e + 2$ and so on until k.
At last, we would need to consider the possibility of getting k tails in k tosses
We can determine the number of tosses needed to get k consecutive f with the following equation:

$$e_k = \frac{1}{2}(e_k + 1) + \frac{1}{4}(e_k + 2) + ... + \frac{1}{2^k}(e_k + k) + \frac{1}{2^k}(k)$$

The above equation can be simplified into:

$$e_k = 2(2^k - 1) = 2^(k + 1) - 2$$

To get the expected number in order to get one head first,
we just need to set $k = 1$ in the previous equation:

$$e_1 = 2^2 - 2 = 2$$

The number of expected tosses in order to get one head and
k tails is simply the sum of the expected tosses for the two cases.

$$e_k + e_1 = 2^{k+1} - 2 + 2 = 2^{k+1}$$

Problem 2

$$\begin{array}{ll}
\min_{w,b,\varepsilon,\hat{\varepsilon_i}} & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\varepsilon_i + \hat{\varepsilon_i}) \\
\text{s.t.} & \quad y_i \leq w^T x_i + b + \epsilon + \varepsilon_i \\
 & \quad y_i \geq w^T x_i + b - \epsilon - \varepsilon_i \\
 & \quad \varepsilon_i \geq 0 \\
 & \quad \hat{\varepsilon_i} \geq 0 \\
 & \quad \forall i = 1...N
\end{array}$$

First of all, we need to change the constraints, thus obtaining:

$$\begin{array}{ll} \min_{w,b,\varepsilon,\hat{\varepsilon}_i} & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\varepsilon_i + \hat{\varepsilon}_i) \\ \text{s.t.} & y_i - w^T x_i - b - \epsilon - \varepsilon_i \leq 0 \\ & -y_i + w^T x_i + b - \epsilon - \varepsilon_i \leq 0 \\ & -\varepsilon_i \leq 0 \\ & -\hat{\varepsilon}_i \leq 0 \\ & \forall i = 1...N \end{array}$$

We need to introduce a Lagrange multiplier for each constraint we have: α, β, γ, δ.
The lagrangian function is defined as follows:

$$\mathcal{L}(w, b, \varepsilon, \hat{\varepsilon}, \alpha, \beta, \gamma, \delta) =$$

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\varepsilon_i + \hat{\varepsilon}_i) + \sum_{i=1}^{N}\alpha_i(y_i - w^T x_i - b - \epsilon - \varepsilon_i)$$

$$+ \sum_{i=1}^{n}\beta_i(-y_i + w^T x_i + b - \epsilon - \hat{\varepsilon}_i) + \sum_{i=1}^{N}\gamma_i(-\varepsilon_i) + \sum_{i=1}^{N}\delta_i(-\hat{\varepsilon}_i)$$

After this we need to calculate the gradient $\nabla\mathcal{L}(w, b, \varepsilon, \hat{\varepsilon}, \alpha, \beta, \gamma, \delta)$,
and in order to do so, we need to compute the partial derivate for each variable.

- $\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{i=1}^{N}\alpha_i x_i + \sum_{i=1}^{N}\beta_i x_i$
- $\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N}\alpha_i x_i + \sum_{i=1}^{N}\beta_i x_i$
- $\frac{\partial \mathcal{L}}{\partial \varepsilon} = C - \alpha - \gamma$
- $\frac{\partial \mathcal{L}}{\partial \hat{\varepsilon}} = C - \beta - \delta$

Setting $\nabla\mathcal{L}(w, b, \varepsilon, \hat{\varepsilon}, \alpha, \beta, \gamma, \delta) = 0$:

- $w = \sum_{i=1}^{N}\alpha_i x_i + \sum_{i=1}^{N}\beta_i x_i$
- $\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{N}\alpha_i x_i = \sum_{i=1}^{N}\beta_i x_i$
- $\frac{\partial \mathcal{L}}{\partial \varepsilon} = C = \alpha + \gamma$
- $\frac{\partial \mathcal{L}}{\partial \hat{\varepsilon}} = C = \beta + \delta$

If we substitute the newly found equations in the original one, we get:

$$-\frac{1}{2}\sum_{i=1}^{N}(\alpha_i - \beta_i)\sum_{j=1}^{N}(\alpha_j - \beta_j)x_i^T x_j + \sum_{i=1}^{N}y_i(\alpha_i - \beta_i) - \sum_{i=1}^{N}\epsilon(\alpha_i + \beta_i)$$

Finally, we can declare the dual problem with the new KKT constraints for soft SVM:

$$\begin{array}{ll} \max_{\alpha,\beta} & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\beta_i)(\alpha_j-\beta_j)x_i^T x_j + \sum_{i=1}^{N} y_i(\alpha_i-\beta_i) - \sum_{i=1}^{N}\epsilon(\alpha_i+\beta_i) \\ \text{s.t.} & 0 \geq \alpha_i \leq C, \forall i \\ & 0 \geq \beta_i \leq C, \forall i \\ & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \sum_{i=1}^{N}\beta_i y_i = 0 \end{array}$$

Problem 3

Several tests have been ran in order to get a possible best run configuration for the spiral set.

- **features**: features have been added one by one to observe the behavior of the network.
  I have observed that with the spiral database, the *sin* function is the most adept.
  The chosen features are: X1, X2, sin(X1), sin(X2)
- **neurons per layer** : 5 neurons per layer
- **hidden layers** : after multiple test runs, I found out that with 2 hidden layers,
  the network was able to keep the loss extremely low and allowed faster convergence speed.
  Using 3 layers actually made the loss rate more unstable
- **learning rate** : the fastest convergence speed has been achieved selecting a learning rate of 0.1
- **activation function**: the function that gives the fastest convergence speed is *tanh*
- **regularization**: introducing regularization actually worsened the error values,
  so in the end, I chose to go without regularization

Problem 4

$$\max_{w} \mathcal{L}(w)$$

We know that the iterative formula is defined as:

$$w_{t+1} = w_t - H^{-1}\nabla_w \mathcal{L}(w_t)$$

First of all, we need to find w, such as $\nabla_w \mathcal{L}(w) = 0$ and the Hessian matrix

- $\nabla_w \mathcal{L}(w) = X(y-\mu)$
- $H = -XRX^T$

Note that $\mu_i = \psi(w_t^T x_i) = \frac{1}{1+exp(w_t^T x_i)}$ and $\nabla_w \mu_i = \mu_i(1-\mu_i) = R_{ii}$
By substituting the above equations into the iterative formula we get:

$$w_{t+1} = w_t - (XRX^T)^{-1}X(y-\mu)$$

$$w_{t+1} = (XRX^T)^{-1}(XRX^T w_t - X(y-\mu))$$

We can apply the same procedure for the L2-norm regularized logistic regression.
Note that we already have the gradient of $\mathcal{L}(w)$

$$\max_w -\frac{\lambda}{2}||w||_2^2 + \mathcal{L}(w)$$

- $\nabla_w - \frac{\lambda}{2}||w||_2^2 + \nabla_w \mathcal{L}(w) = -\lambda w + X(y-\mu)$
- $H = \nabla_w^2 - \frac{\lambda}{2}||w||_2^2 + \nabla_w \mathcal{L}(w) = -\lambda - XRX^T$

So the new update equation is defined as:

$$w_{t+1} = w_t + (\lambda + XRX^T)^{-1}(-\lambda w_t + X(y-\mu))$$

This problem couldn't be solved due to an error during operation between matrix
, thus preventing further experiments