1 Project 1 — Machine Learning — Classification

Create a model to classify the images in the MINST data set. This is a dataset containing grayscale images of handwritten digits, commonly used to train image processing systems. The dataset is briefly described in https://en.wikipedia.org/wiki/MNIST_database. This dataset is available in `sklearn` and can be loaded with the following code:

```
from sklearn.datasets import load_digits
digits = load_digits()
```

To display one of the images of the dataset, we can use the following code:

```
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[25])
```

Your task in this project is to create a neural network based classifier for this dataset. Experiment with the configuration of the neural network with the goal of obtaining a good classifier.

2 Project 2 — Machine Learning — Regression

Create a model to make predictions for the California Housing Dataset, which can be loaded with the following code:

```
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
```

Your task for this project is to create a neural network regression model to predict the median house value for house blocks. Notice that the 'target' array in this case represents a random variable taking floating point values, not classes as in the Iris or MINST examples. Use the class `MLPRegressor` class, documented at https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html to define the model.

Notice that this is a more complex data set, and may require some preprocessing. One aspect that has to be addressed is the different scales present in the data, which usually is solved by normalizing the data to an appropriate range.

3 Project 3 — Partial Differential Equations — The Heat Equation

We start with a simple one-dimensional model. Suppose we have a rod of heat-conducting material. We assume that the rod is well-insulated, except at it's endpoints, where heat can be exchanged with the environment. Let L be the length of the rod. We use the variable x to identify the position of a point on the rod, and let $u(x,t)$ denote the temperature of the rod at position x and time t. The evolution of $u(x,t)$ is then derived according to the following equation:

$$u_t = \alpha u_{xx} + f(x)$$

Let's start by finding the equilibrium solution corresponding to this equation, obtaining by setting $u_t = 0$, which corresponds to the stable solution of the heat equation obtained as $t \to \infty$. That is, the equation we will solve is:

$$\alpha u_{xx} + f(x) = 0$$

which can be written:

Notice that now $u(x)$ is a function of one variable, denoting the stationary temperature distribution.

Besides the equation above, which models heat flow in the rod, we need to set boundary conditions that specify how heat is exchanged at the ends of the rod. In our first example, we will assume that the ends of the rod are kept at constant temperature, represented by the equations:

$$u(0) = a, \quad u(L) = b$$

To solve this equation approximately, we use the method of finite differences. We discretize space using the same formulas for numerical differentiation from Lesson 13.

To discretize space, we choose an integer N and subdivide the interval $[0, L]$ in N subintervals, by defining the grid points:

$$x_i = ih, \quad (0 \le i \le N),$$

where $h = 1/N$. We want to approximate the solution $u(x)$ at the grid points x_i. To be precise, we want to compute values $u_0, \dots u_N$ such that $u_i \approx u(x_i)$.

To construct the approximation, we use a central difference approximation for the second derivative:

$$u_{xx}(x_i) \approx \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

We use this formula to define the discretized version of the heat equation:

$$\alpha \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = -f(x_i), \quad (1 \le i \le N - 1),$$

which we rewrite as:

$$u_{i+1} - 2u_i + u_{i-1} = -\frac{h^2}{\alpha} f(x_i), \quad (1 \le i \le N - 1)$$

This formula is only valid at the grid points. At the boundary points we set:

$$u_0 = a, \quad u_N = b.$$

In order to compute a numerical approximatio, we reformulate the problem in terms of matrices. We define a vector:

$$\mathbf{u} = \begin{bmatrix} u_0 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}$$

We also define a matrix $A = (a_{ij})_{0 \le i,j \le N}$ by:

$$\begin{cases} a_{ii} = -2 & \text{if } 1 \le i \le N-1 \\ a_{i,i+1} = a_{i,i-1} = 1 & \text{if } 1 \le i \le N-1 \\ a_{00} = a_{NN} = 1 & \\ a_{ij} = 0 & \text{for all other values of } i,\, j \end{cases}$$

Finally, we define a vector $\mathbf{r} = (r_i)_{0 \le i \le N}$ by:

$$r_i = \begin{cases} -\frac{h^2}{\alpha} f(x_i) & \text{if } 1 \le i \le N-1 \\ a & \text{if } i = 0 \\ b & \text{if } i = N \end{cases}$$

With this notation, we can write the discretized heat equation as the linear system:

$$A\mathbf{u} = \mathbf{r}$$

1. Solve the following version of the steady-state heat equation, for a rod of length $L = 1$:

$$0.5u_{xx} = -\sin(\pi x)$$

with boundary conditions:

$$u(0) = 1, \quad u(\pi) = -2$$

To solve this, you need to define, with the notation above, the matrix A, the vector $\mathbf{r}$, and then find the vector $\mathbf{u}$ that solves $A\mathbf{u} = \mathbf{r}$. Create a plot of the solution.

2. To verify the accuracy of the numerical solution, solve the equation exactly:

- Integrate the equation $0.5\ u_{xx} = -\sin(\pi x)$ twice to obtain the general solution $u(x)$. Your solution will have two constants of integration.
- Find the values of the two constants of integration by plugging in the boundary conditions into the general solution.

Then, compute the exact solution at the gridpoints and compare the results with the numerical solution. Study the accuracy of the solution for different numbers of gridpoints.

3. Modify the equation to add a drift term:

$$\alpha u_{xx} + \beta u_x = f - (x)$$

The parameter β can be positive or negative. The drift represents a bias towards the right or the left in the heat distribution. For example, in a model for insect dispersion, the drift term might represent the effect of the wind. Create a finite difference model for this equation with the same boundary conditions as in the previous example.

Do one of the two following items (or do both for extra credit):

4. Solve the two-dimensional steady-state heat equation on a rectangle::

$$\alpha_1 u_{xx} + \alpha_2 u_{yy} = -f(x,y), (0 < x < L, 0 < y < M)$$

with the boundary conditions:

$$u(x,0) = a, u(x,L) = b, u(y,0) = c, u(y,M) = d$$

Create two graphical representations of the solution: a 3-d plot of the solution surface, and a contour plot. You can also try to make a "heatmap", using different colors to represent temperature variation. See the `matplotlib` documentation for 3-d plots.

5. Solve the time-dependent version of the one-dimensional heat equation:

$$u_t = \alpha u_{xx} + f(x)$$

Use the same boundary conditions as before. Now, you will also need an initial condition:

$$u(x,0) = u_0(x)$$

where $u_0(x)$ is a given function.

The strategy for this problem is the following:

- Discretize time as explained above. Do not discretize time.
- Write the equations in discretized time. This will give you a system or ordinary differential equations for the vector $\mathbf{u}$. The boundary conditions are needed at this step.
- Use a numerical ODE solver to solve the system. You will need the initial condition to set up the system.

Create a three-dimensional display of the heat evolution. See the `matplotlib` documentation for 3-d plots.