Mathematical Modeling Course
End-of-Semester Project: *Linear Classification of Pneumonia in Chest X-rays*

April 7, 2025

The goal of this exercise is to use mathematical modeling to solve a practical problem: Diagnosing Pneumonia using chest X-ray images. You should:

- Formulate mathematical models, which solve the task
- Improve the model by incorporating concepts such as regularization and feature extraction.
- Implement your models in a computer program
- Use the implemented models for the analysis of data
- Report the results from the analysis and discuss the validity of the model

The report for this exercise must be 5 pages at most, including graphs, tables, and images (excluding the front page and appendices). In this exercise, several questions are asked. The report, however, should not be a list of answers, but instead be a coherent documentation and discussion of the analysis performed.

Project Overview

In this project, you will implement a linear classification pipeline to detect pneumonia from chest X-ray images using the **MedMNIST dataset** [1] . Your goal is to investigate how different feature extraction strategies impact the performance of a linear classifier, especially under constraints such as a **small dataset** and **variations in image intensity**.

You will start with implementing a classifier based on the raw pixel values of the X-rays as the input feature to your classifier. In the next steps, you implement more robust features like gradients and histogram representations. For each of these steps, you will analyze the effects of **regularization** and examine the interpretability of your models.

Detecting Pneumonia from X-ray Images

Pneumonia is a serious lung infection that inflames the air sacs in one or both lungs of a patient. It can usually be diagnosed using a radiological imaging technique using X-rays, which is a widely used diagnostic tool in medical practice that captures images of the internal structures of the body using electromagnetic radiation. In chest radiography, X-rays are used to visualize the lungs, heart, and surrounding tissues. Because different tissues absorb X-rays to varying degrees, pneumonia

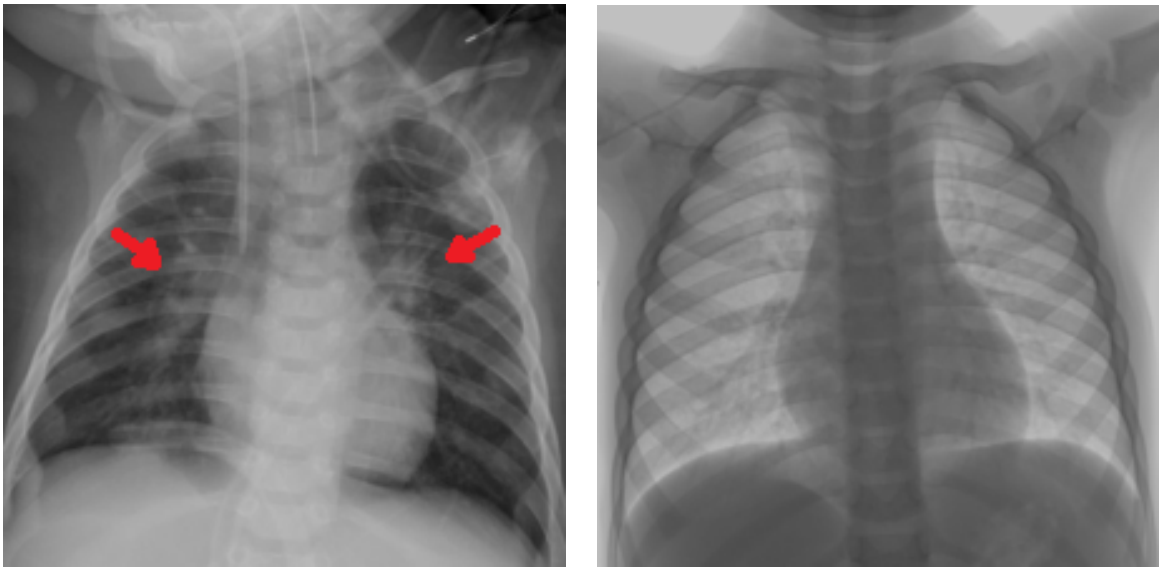

Figure 1: Example X-ray test images from the MedMNIST dataset. With (left), and without Pneumonia (right). Pneumonia is typically visible on chest X-rays as regions of increased opacity or haziness in the lungs, which may appear as cloud-like patterns or localized consolidation due to fluid or inflammation. Note that some images in the database have inverted intensities (dark regions are mapped to bright intensities and vice-versa).

can often be identified on chest X-rays as areas of increased opacity or consolidation. Radiologists examine these images to detect abnormalities such as fluid buildup or infiltrates that are indicative of pneumonia. Automatic classification of pneumonia from chest X-rays can assist in early diagnosis, especially in settings where expert interpretation is not immediately available. However, this task presents several challenges:

- **High-dimensional input:** Images consist of thousands of pixels, but the dataset has relatively small number of samples (400 training samples). This leads to an underdetermined problem prone to overfitting.
- **Intensity Variations:** Some images have flipped intensity measurements (i.e., img $= 1 -$ img), adding extra noise to the detection algorithm.
- **Need for Interpretability:** Certain level of transparency is required to verify the basis for the detection of a model. Linear classifiers offer insight via visualized weights that show which regions influence predictions.

You aim to develop a model that balances accuracy, robustness, and interpretability. Starting with simple features and improving them incrementally will illustrate the modeling tradeoffs and benefits.

Linear Classifiers and Logistic Regression

You will use a linear classifier for detecting Pneumonia label from the X-ray images. Here, we describe the logic behind the Logistic regression method as a candidate tool. Note that there are other alternatives to Linear classifier than the Logistic regression. See for example The Elements

of Statistical Learning, Python scikit package, and Matlab ClassificationLinear class. **Whatever you use, you should understand it!**

Given input $\mathbf{x} \in \mathbb{R}^d$ and label $y \in \{0, 1\}$, a linear model predicts the probability of the positive class via logistic regression:

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic function that maps an arbitrary scalar value in between the probability range $(0, 1)$.

The parameters of this distribution are optimized using the binary cross-entropy objective:

$$\mathcal{L}(\mathbf{w}, b) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i + b)) + (1 - y_i)\log(1 - \sigma(\mathbf{w}^T\mathbf{x}_i + b))\right],$$

which is minimized using gradient descent. The gradients are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{1}{N}\sum_{i=1}^{N}(\sigma(\mathbf{w}^T\mathbf{x}_i + b) - y_i)\mathbf{x}_i,$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{N}\sum_{i=1}^{N}(\sigma(\mathbf{w}^T\mathbf{x}_i + b) - y_i).$$

Regularization

As many problems could suffer from overfitting, the model is regularized to have small weights. This reduces the influence of input features that do not consistently contribute to the model's decision across all samples and helps prevent the model from memorizing individual examples.

One approach to regularization is to apply the **L2 regularization**:

$$\mathcal{L}_{\text{reg}}(\mathbf{w}, b) = \mathcal{L}(\mathbf{w}, b) + \lambda\|\mathbf{w}\|^2$$

where $\lambda > 0$ is the regularization strength. This penalizes large weights, promoting simpler, more generalizable models. Regularized optimization can also be optimized via gradient descent.

You will test multiple λ values and analyze their effect on model performance.

Project Workflow

You must implement three modeling approaches in the following order, beginning with the simplest representation and progressing toward more robust and structured features.

Part 1

In the first approach, you will use the raw pixel intensities of the input X-ray images as input features to your model. This involves flattening each image into a one-dimensional vector, where each element corresponds to a grayscale intensity value. Using these raw pixel vectors, you will train a linear classification model to classify images as pneumonia-positive or negative. Because the dataset is relatively small and high-dimensional, you must train the model using several values of the regularization strength λ, and select the one that gives the best performance on the test set. For each value, evaluate and report the test accuracy. In addition, you must visualize the learned weights of the linear model by reshaping them back into image space. Because the weights indicate

which input features contribute more to the final decision of the model, visualizing the absolute value of the weight would show which features are mostly influencing the model. In other words, this helps interpret which regions of the image the classifier relies on to make predictions.

Part 2

In the second approach, you will move beyond the raw pixel values and instead extract gradient-based features from the X-ray images. You can use the techniques you learned in the first project to implement image gradient calculation. You may use the magnitude of the pixel gradients, which are more robust to global intensity changes and can highlight edges and structural details that are more relevant for detecting abnormalities like pneumonia. Once computed, the gradient values (e.g., the magnitude or the separate horizontal and vertical components) should be flattened and used as input features for a logistic regression model. As in Part 1, you will train the model using a range of regularization strengths λ, evaluate performance on the test set, and identify the best-performing model. Visualize the corresponding learned weights in the image space and analyze whether gradient-based features offer better interpretability or robustness compared to the raw pixel features.

Part 3

The third and final approach involves designing a more robust and compact feature representation that addresses the limitations of both raw pixels and direct gradient features. A suggested method is to divide each X-ray image into local patches—for example, non-overlapping regions of size 32×32 and compute a histogram of orientations for gradients within each patch. This is detailed in the following Algorithm 1.

This type of representation captures the distribution of edge directions while discarding exact pixel locations, making it more invariant to small shifts or intensity inversions. The concatenation of these local histograms forms a new feature vector that is both lower-dimensional and more robust to noise and irrelevant details.

Alternatively, you are encouraged to experiment with other feature representations (e.g., low-pass filtered patches, PCA projections, etc.), as long as the method aims for two characteristics: (1) it is more robust to image variations, such as color inversions or structural noise, and (2) it reduces the dimensionality of the input, improving generalization. Not reaching optimal results is not criticized, but you should have a justification for your design choices.

As with previous approaches, train your linear classifier with various levels of regularization λ, evaluate the performance of the model, and visualize the learned weights. You should compare this method both quantitatively and qualitatively with the earlier two approaches and reflect on how the feature representation impacts classification accuracy and interpretability.

Data Details and Structure

You are given a dataset of images containing two folders: *Train* and *Test*. Each containing images, where the name indicates whether the image label is positive (neumonia) or negative (healthy). All images are grayscale of size 224×224.

Algorithm 1 Computing the histogram of orientations at local patches of image gradients

1: **Compute Image Gradients:**
E.g. using basic gradients or Sobel filter to get vertical and horizontal components G_x, G_y.

2: **Calculate Gradient Magnitude and Orientation:**
At each pixel, compute the gradient magnitude as $M = \sqrt{G_x^2 + G_y^2}$.
Compute the gradient orientation as $\theta = \arctan 2(G_y, G_x)$.
Orientations can be expressed in the signed range ($-180°$ to $180°$) or unsigned ($0°$ to $180°$).

3: **Divide the Image into Patches:**
Split the image into non-overlapping cells, e.g., 32×32 pixel patches.

4: **Create Orientation Bins:** Choose a fixed number of orientation bins (e.g., 18 bins). Each bin represents an equal range of angles (e.g., $10°$ per bin if $0°$ to $180°$ is used).

5: **Compute Histogram for Each Patch:**
For each pixel within a patch:
I. Identify the gradient orientation and magnitude.
II. Add the pixel's magnitude to the corresponding orientation bin.
III. Optionally, apply bilinear interpolation between bins for smoother histograms.

6: **Normalize Histograms (Optional):**
Normalize each patch histogram to have unit length.
This will improve robustness to local contrast variations.

7: **Concatenate Histograms:**
Combine all patch histograms into a single feature vector.
Use this vector as the input to your linear classifier.

Report Guidelines

Max 5 pages, including all figures and tables (excluding front page and appendices). Avoid answering questions as a list. Instead, provide a coherent narrative that explains your analysis.

Include:

- Introduction and problem motivation
- Description of data and experiment
- Description of the mathematical model - how and why?
- Feature extraction steps
- Visualizations (sample images, features, weights)
- Accuracy and performance comparisons
- Regularization discussion
- Final conclusions; what is the contribution of the analysis

References

[1] Jiancheng Yang, Rui Shi, Bingbing Ni. *MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis.* In *Proceedings of the IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021.