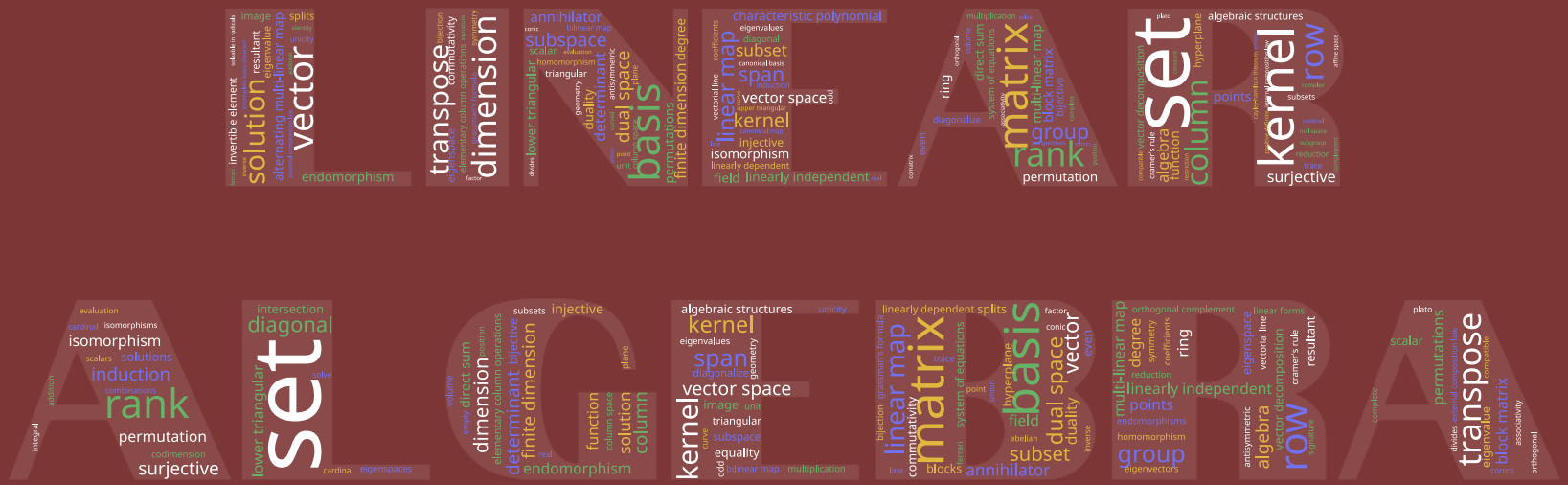


Presentation Topic

MATH214 – Spring 2023

Team member names

1. First section

2. Application of Gradient Descent: MNIST

Application Introduction

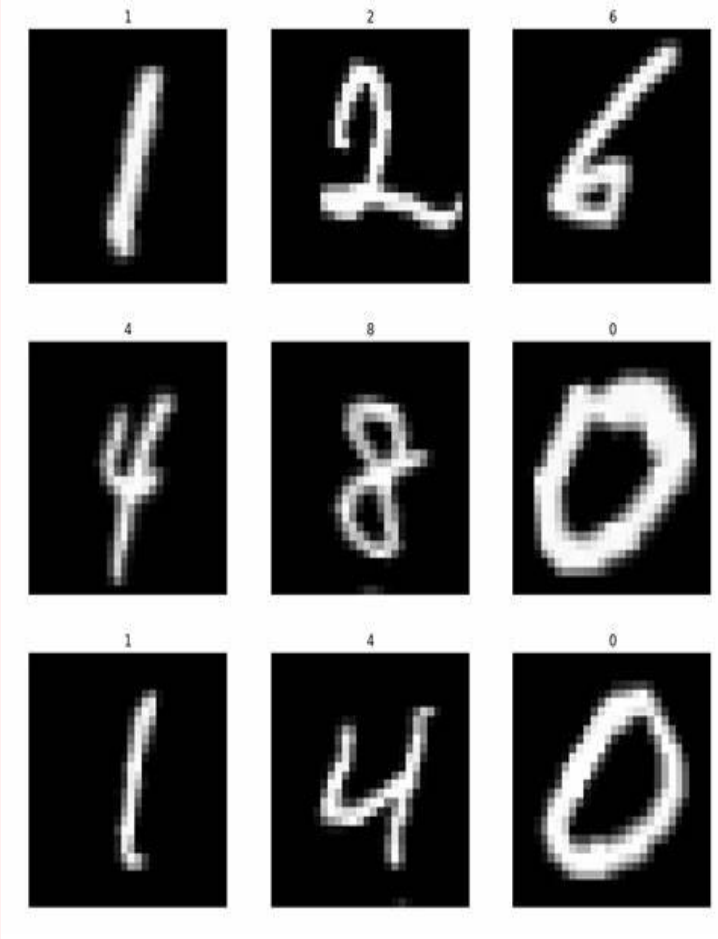


Figure 1

- MNIST is a famous dataset used for image recognition and classification tasks. It consists of 70,000 (60,000 training+10,000 testing) handwritten digits in a 28x28 px grayscale image.
- Our project application is to use CNN model predict MNIST handwritten digits. In the application, we discover how Gradient Descent applies in DL.
- Project source code repo: https://github.com/openhe-hub/math214-project.git

Model Introduction

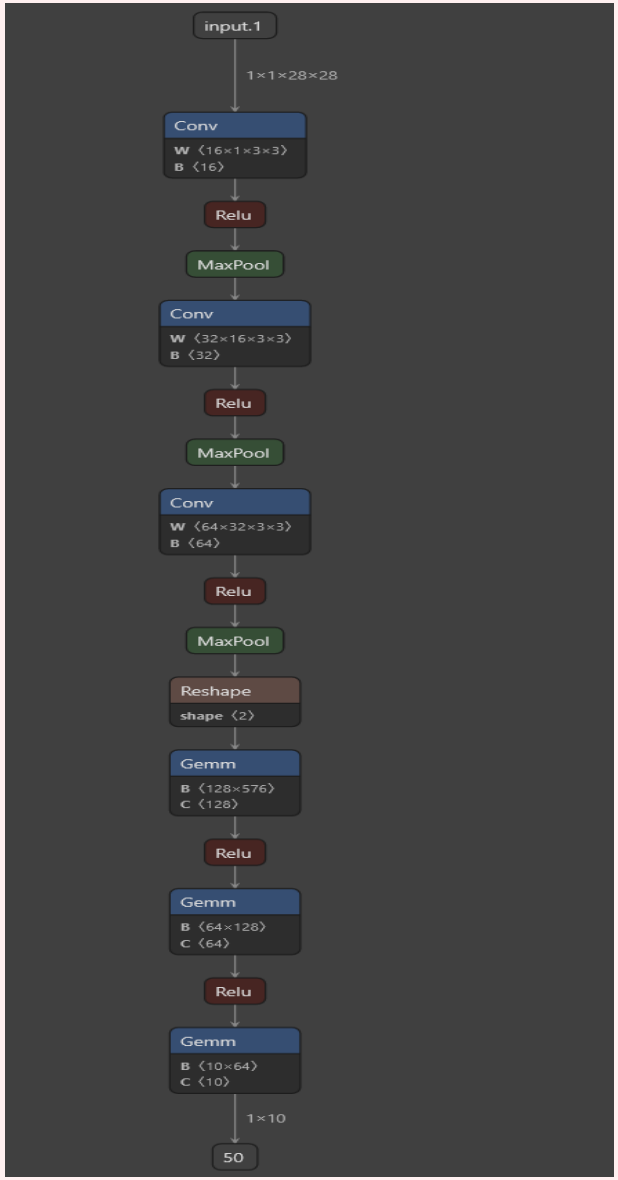


Figure 2 CNN Model (visualized by Netron)

- Our CNN Model:
  3 *conv_layers* (Conv+Relu+MaxPool)
  +1 *fc_layer* (Linear+Relu)
- Loss function: *CrossEntropy*
- Gradient Descent: *SGD*
- Other Parameters:
  - $epochs = 30$
  - $batch\_size = 64$
  - $learning\_rate = 0.001$

Gradient Descent

Theorem

SGD Iteration Formula: $w_{t+1} = w_t - \eta \nabla L(w_t)$

Algorithm 1: SGD algorithm for training a CNN

Data: Input data x and labels y

Result: Trained CNN model

Function `Train()`**:**
  for *epoch =1 to epochs* **do**
    for *batch_i* $= (x_i, y_i)$ *in dataset* **do**
      Set gradients of network parameters to zero;
      Forward propagation: $\hat{y}$=forward(x_i);
      Compute loss function: *loss*=$L(\hat{y}, y_i)$;
      Backward propagation: *loss*.backward();
      Update parameters using SGD: $w = w - \eta \nabla L(w)$;
    end
  end

Conclusion

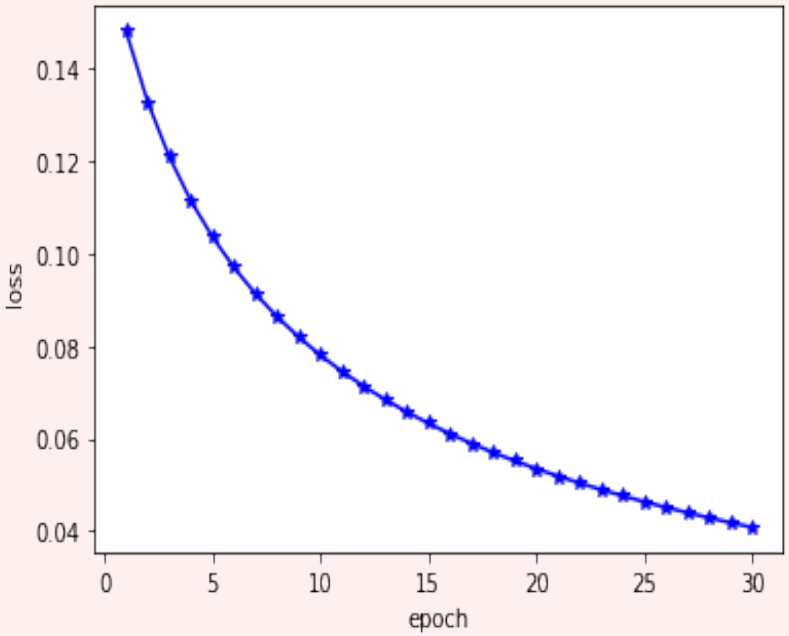


Figure 3 SGD epoch-loss plot

The epoch-loss plot shows how the model's loss function changes over time, as it updates the model parameters using small batches of training data. The plot typically displays a downward trend, with rapid loss reductions in initial epochs, and slower reductions as the algorithm converges.

Classification Result

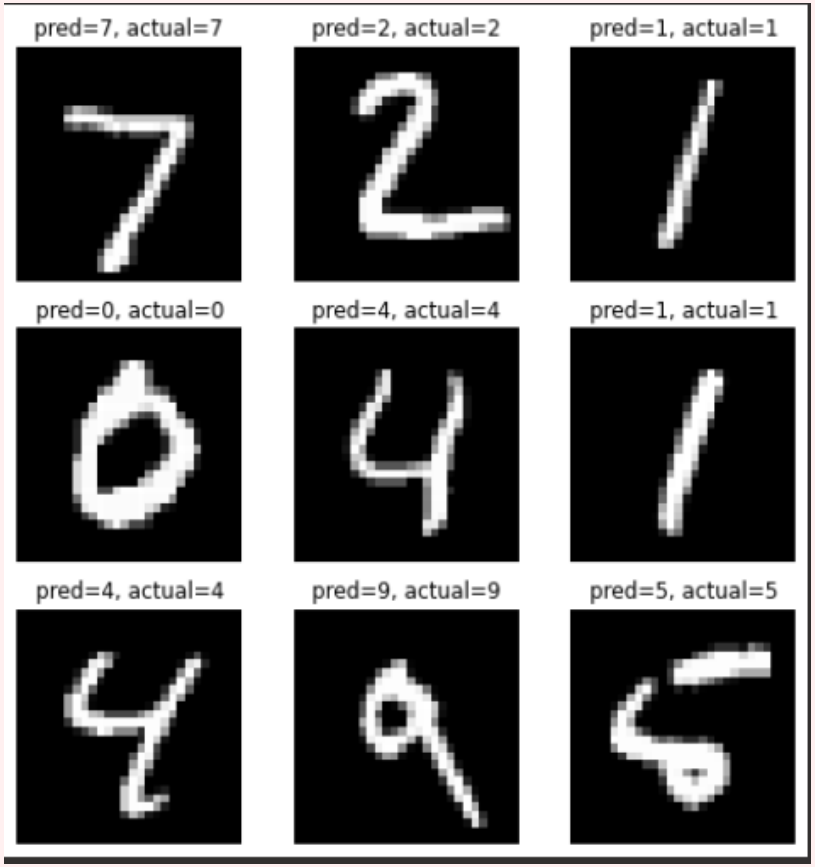
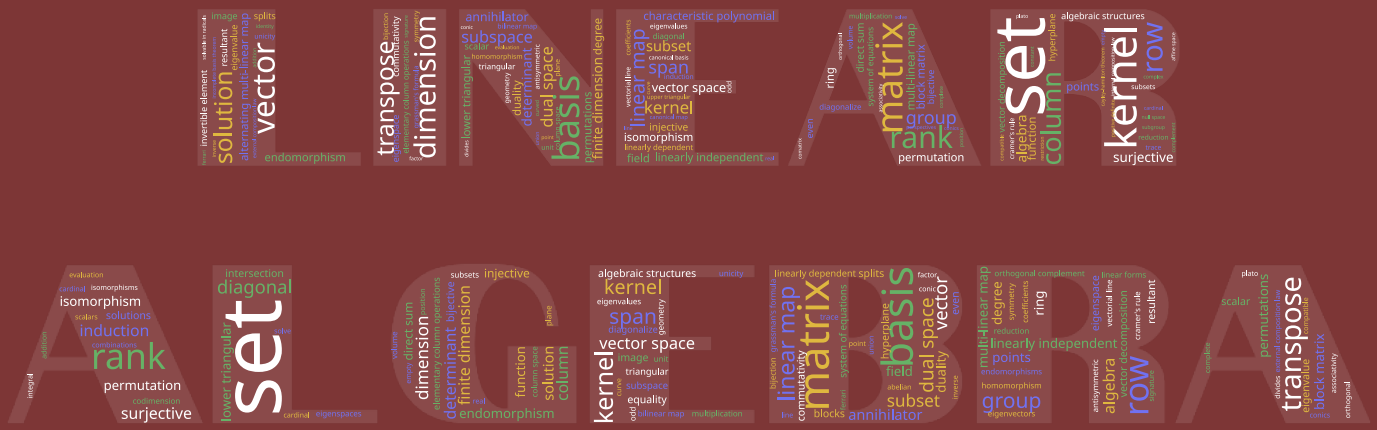


Figure 4 Testing set prediction result

- Result: The total loss is about 0.0413, and the accuracy is about 98.67%. Algorithms (SGD) based on Gradient Descent works well in our CNN model.
- Improvement: Consider adjusting parameters like *learning_rate*, *batch_size*. Also, we may change different GD algorithms like *Adam*.

3. Third section

4. Fourth section



Thank you!