Assignment 5 - Convolutional Neural Networks
Winter 2024

Atieh Armin
Student ID : 14658308

1 Theory

1. (2pts) Apply kernel K to data X. In other words, what is $X * K$?:

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} K = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

   Solution:

   Since our image size is 3x3 and our kernel size is 3x3, $X * K$ will be 1x1. So, the answer is $\begin{bmatrix} 285 \end{bmatrix}$

2. Given the feature map filter output, $F = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 3 \\ 4 & 5 & 6 & 0 & 0 & 12 \\ 7 & 8 & 9 & 1 & 0 & 4 \\ -100 & -100 & -100 & -100 & -100 & -100 \end{bmatrix}$, what is the output from a pooling layer with width of 2 and stride of 2 if we are using:

   Solution: Our feature map is 4x6 and our pooling layer is 2x2 with stride of 2. So the output of the pooling process should be 2x3.

   - (3pts) Max-Pooling?

     $P = \begin{bmatrix} 5 & 6 & 12 \\ 8 & 9 & 4 \end{bmatrix}$

   - (3pts) Mean-Pooling?

     $P = \begin{bmatrix} 3 & 3.25 & 4 \\ -46.25 & -47.5 & -49 \end{bmatrix}$

3. (2pts) Given an image X, what would the kernel K be that can reproduce the image when convoluted with it? That is, what is K such that $X * K = X$?

Solution: We know that cross-correlation takes a kernel K and applies to an image X as a sliding dot-product. First, if we want to reproduce the image, we should add zero-padding. We can assume that we have odd numbers for the kernel's dimension like 3x3: After adding zero padding, when we start to slide the kernel on the image, to reproduce the exact same image, we will need the center digit of the kernel to be 1 as it covers the pixel in the image that we are going to reproduce, and all other digits to be 0 as we don't want to add it to another value. So each pixel in the output image is the result of multiplying the kernel with the corresponding region of the input image. Because all the surrounding elements of the kernel are 0, the contribution of adjacent pixels would be zero, and the 1 in the center of the kernel ensures that each pixel in the image remains the same.

So, $K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

2 CNN Layers

The first thing you'll need to do is implement some new layers. In particular we'll need:

ConvolutionalLayer Your constructor should take in a kernel width and height to intialize the weights of the kernel accordingly. Note that for simplicity we will have a single kernel (although you may opt to support multiple ones). This layer should have the methods listed below. For further simplicity, we *will not* be passing the gradient back through this layer, so we don't need to put anything in the *backward* or *gradient* methods.

1. forward
2. backward - You can just put *pass* for this
3. gradient - You can just put *pass* for this
4. updateWeights
5. crossCorrelate2D - You will implement your own 2D cross-correlation

MaxPoolLayer This just needs the *forward*, *backward*, and *gradient* methods.

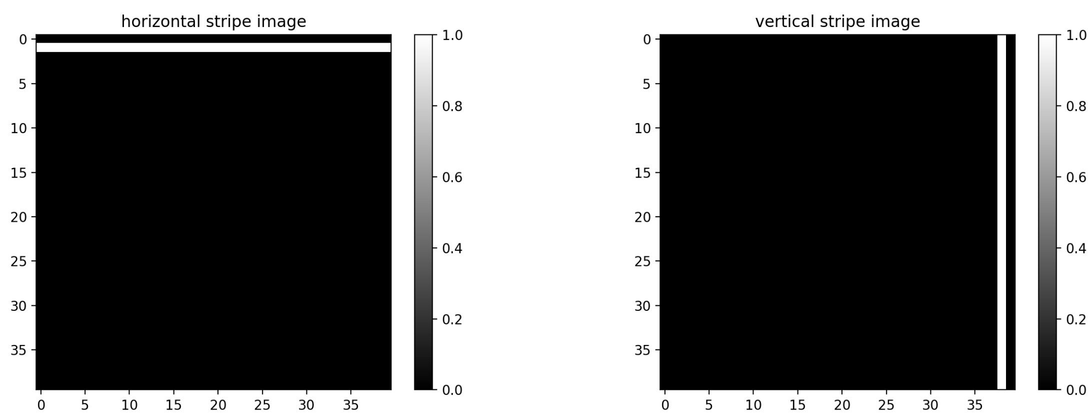
FlatteningLayer This too just needs the *forward*, *backward*, and *gradient* methods.

3 CNN for Classification of Synthetic Data

Let's start attempting to differentiate between your two synthetic images using a simple CNN architecture. The CNN architecture is as follows. All hyperparameter choice are up to you:

1. A **single** 9×9 convolutional kernel.
2. A max-pool layer with $width = 4$ and $stride = 4$.
3. A flattened layer
4. A Fully Connected Layer
5. A Logistic Sigmoid activation function
6. A Log Loss objective function.

Synthetic Data:

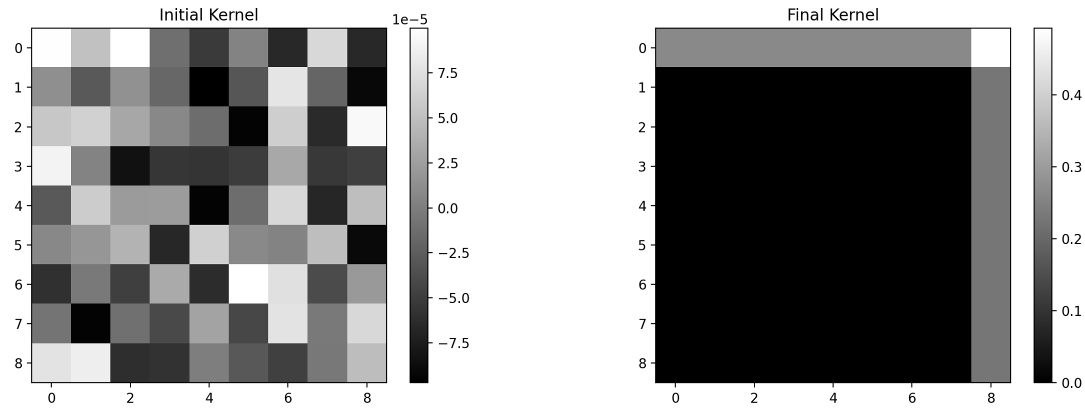


(a) Generated image with horizontal stripe (b) Generated image with vertical stripe

Figure 1: 40 x 40 Synthetic Data

What you will need for your report

1. Image representations of your initial and final kernels.



(a) Image representation of initial kernel (b) Image representation of final kernel

Figure 2: 9 x 9 Kernel

2. A plot of the log loss as a function of the number of iterations.

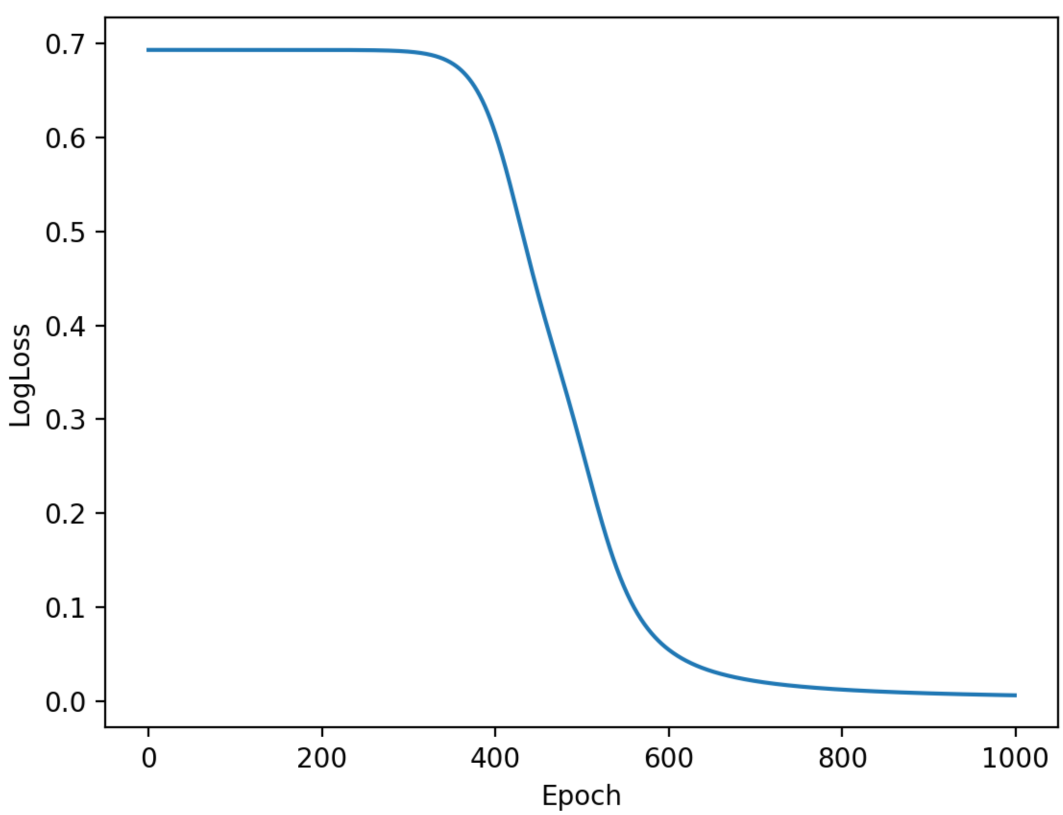


Figure 3: Log Loss vs. Epochs

3. Your hyperparameter choices.

   I tried different amounts for each hyperparameter, and the best and most sensible are the following:

   (a) Learning rate $\eta = 10^{-2}$
   (b) Number of epochs $= 1000$
   (c) Weights randomly initialized to $\pm 10^{-4}$

4 CNN For Image Classification

Let's see if this works on real data! We'll attempt to classify people in the Yale Faces dataset. Since training a CNN can be very time consuming, we'll so the following:

- Resize the images to be 40×40
- Just use one image for each person for training.

As a result, your training data will be a $14 \times 40 \times 40$ tensor.
The CNN architecture is as follows. All hyperparameter choice are up to you:

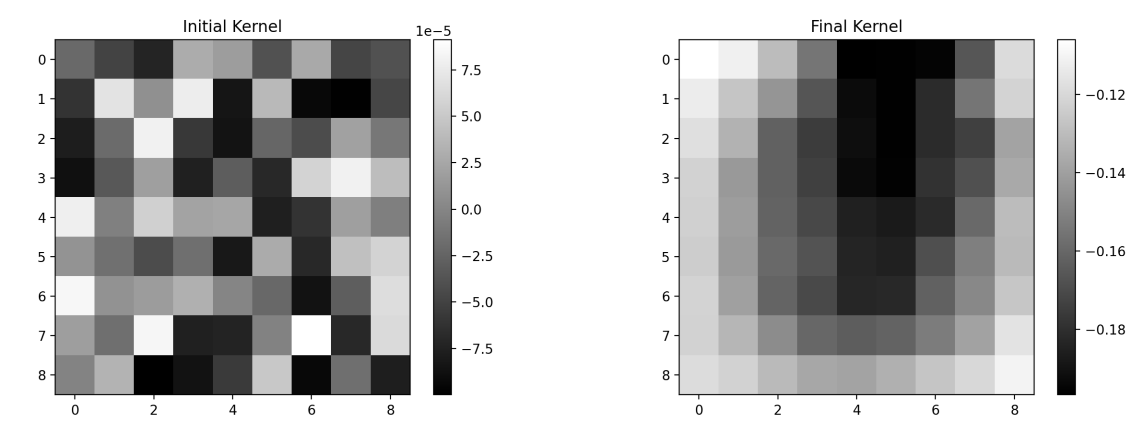
1. A **single** 9×9 convolutional kernel.
2. A max-pool layer with $width = 4$ and $stride = 4$.
3. A flattened layer
4. A Fully Connected Layer
5. A Softmax activation function
6. A Cross Entropy Loss objective function.

Additional Implementation Details

- One-hot encode your targets (beware that the first ID is not 0)
- Make sure to either z-score your features or divide them by 255.

What you will need for your report

1. Image representations of your initial and final kernels.



(a) Image representation of initial kernel (b) Image representation of final kernel

Figure 4: 9 x 9 Kernel

2. A plot of the cross entropy loss as a function of the number of iterations.

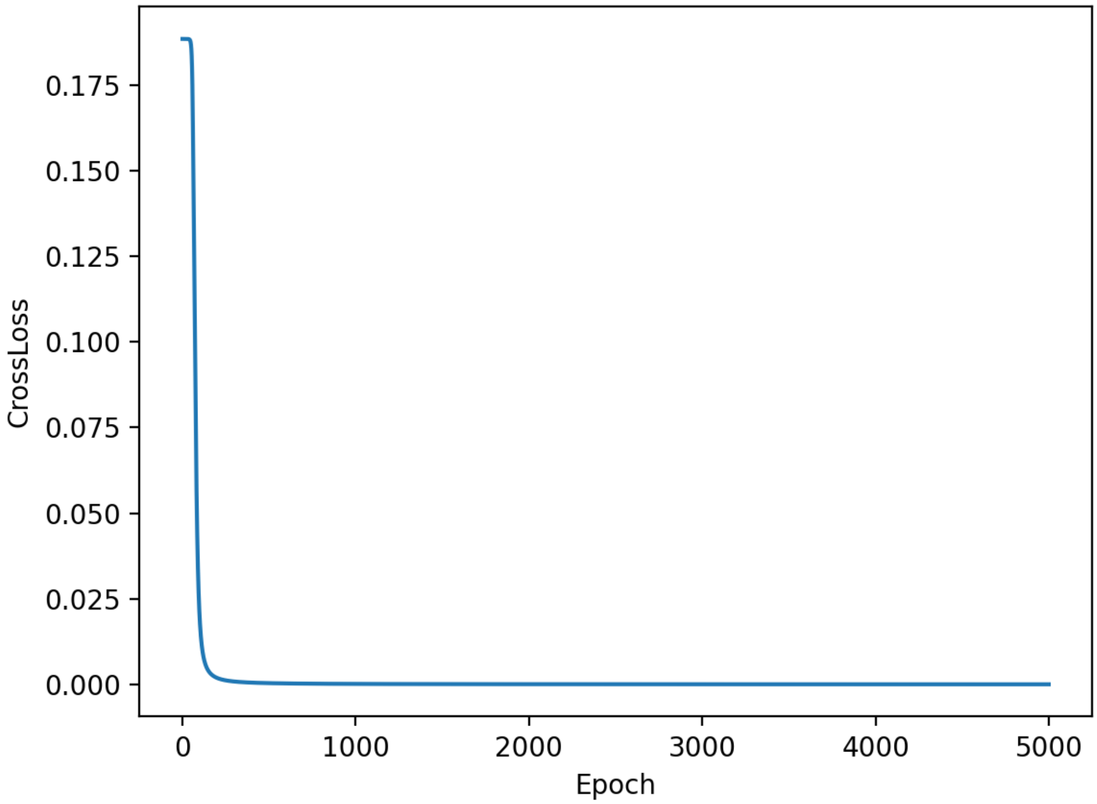


Figure 5: Cross Entropy Loss vs. Epochs

3. Your hyperparameter choices.

   I tried different amounts for each hyperparameter, and the best and most sensible are the following:

   (a) Learning rate $\eta = 10^{-2}$

   (b) Number of epochs = 5000

   (c) Weights randomly initialized to $\pm 10^{-4}$

   So the accuracy of this model is as follows:

   Train Accuracy: 1.0

   Test Accuracy: 0.9311224489795918

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1:
   (a) Your solutions to the theory question
2. Part 2: Nothing
3. Part 3:
   (a) Image representations of your initial and final kernels.
   (b) A plot of the log loss as a function of the number of iterations.
   (c) Your hyperparameter choices.
4. Part 4:
   (a) Image representations of your initial and final kernels.
   (b) A plot of the cross entropy loss as a function of the number of iterations.
   (c) Your hyperparameter choices.