Assignment 3 Presentation

Agenda

Data preprocessing and experimental set-up

Deep learning models

Training algorithm

Results

Data preprocessing and experimental set-up

Data Prep

Siamese Network

1. Randomly Apply (50%)
    a. Random rotation (45 degrees)
    b. Random affine
        i. Translate / Shift
        ii. Scale / Zoom
        iii. Shear / Italicize
    c. Random flip (50%)
2. Normalize
3. Pairs as input

Generative Adversarial Network + CNN

1. Normalize (mean=0.5, std=0.5)
2. GAN used to generate more training samples

Tools: pytorch*

Data selection

Images Background

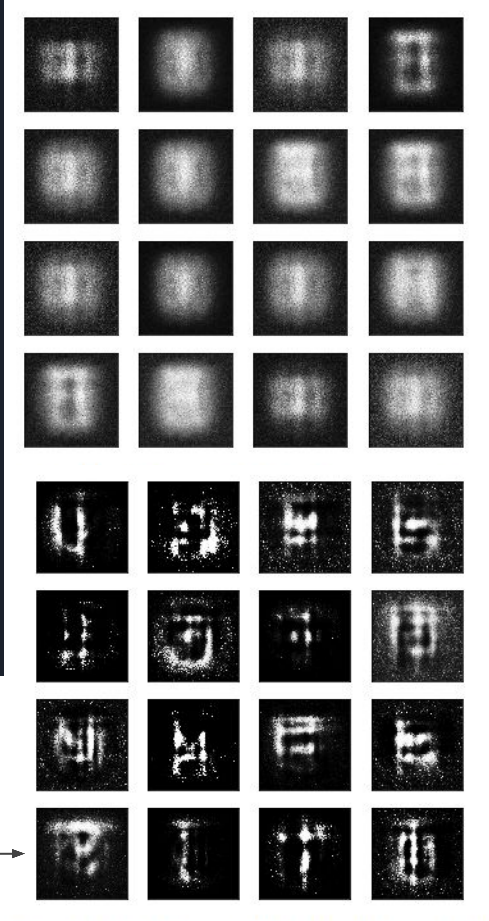
Images Evaluation

Images Background Small 1

GAN generated training samples

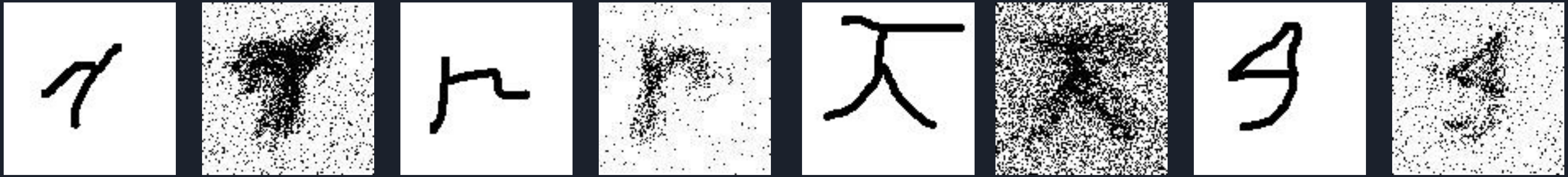
Initial GAN

Good results but no way to tell which class is which.



Look a dog

Using GAN to extend Data



GAN is run for each character:

1. Noisy.. but is that good?
2. Takes a really long time

Model GAN

Discriminator:

Linear LeakyReLU Dropout

Linear LeakyReLU Dropout(0.3),

Linear LeakyReLU Dropout(0.3),

Linear Sigmoid

Generator:

Linear LeakyReLU

Linear LeakyReLU

Linear LeakyReLU

Linear Tanh

Tools: pytorch*

Tools: pytorch*

Model CNN

Feature / Conv

Conv2d BatchNorm2d ReLU

Conv2d BatchNorm2d ReLU MaxPool2d

Conv2d BatchNorm2d ReLU

Conv2d BatchNorm2d ReLU MaxPool2d

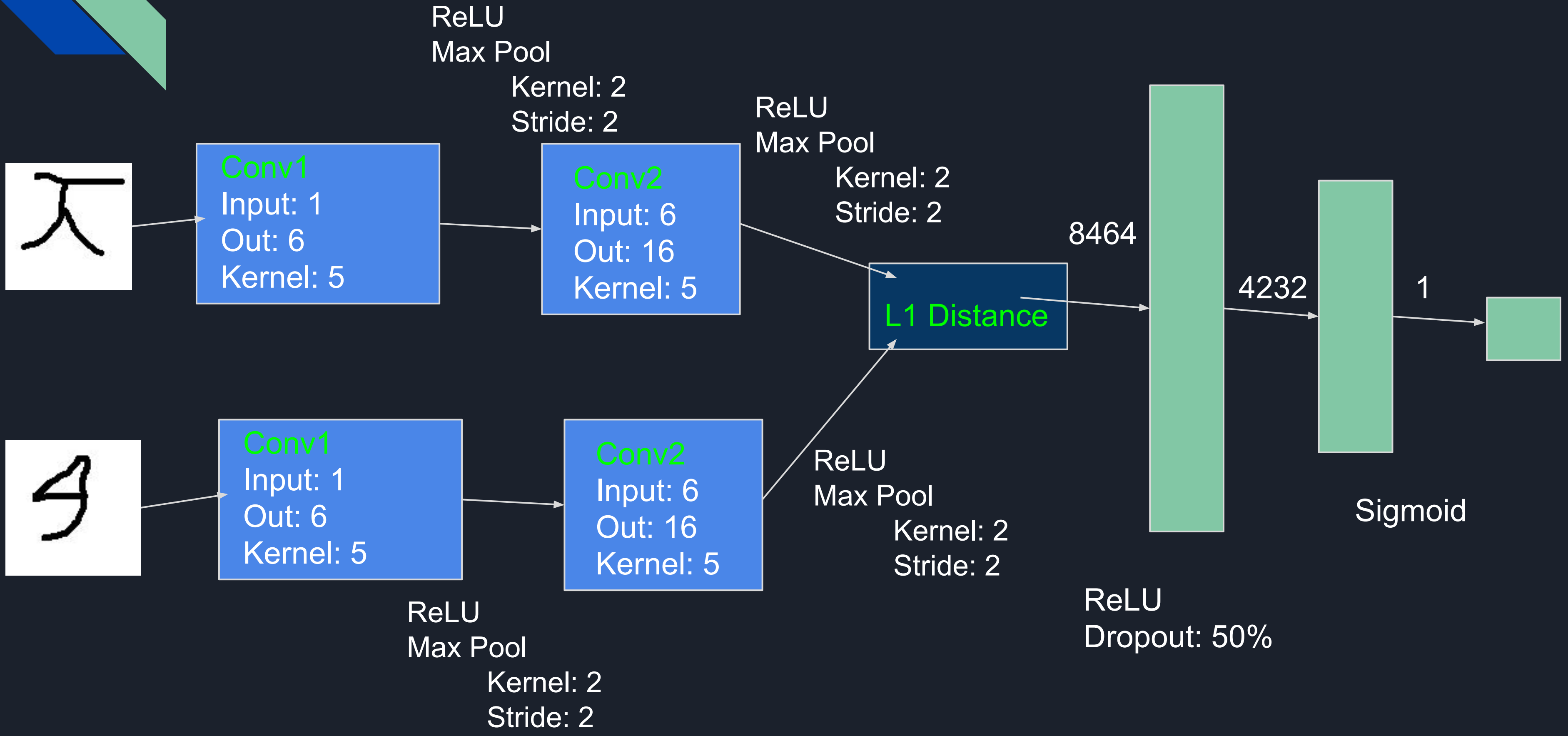
Classifier

Dropout Linear BatchNorm1d ReLU

Dropout Linear BatchNorm1d ReLU

Dropout Linear

Siamese Network Model



Training:

GAN was run over each character and after a certain number of epochs the output was saved to the dataset.

Once all the data was generated 2 CNN models of the same structure were run on the generated and original data.

Batch size of 100 was used. Learning rate of 0.0001.

Adam optimizer
Cross-Entropy as loss function

Results:

Extending the dataset with GAN output could be a valid way of reducing generalization error. As can be seen with an accuracy difference of ~7%.

Discuss performance:

Roadblock:

Thank you!