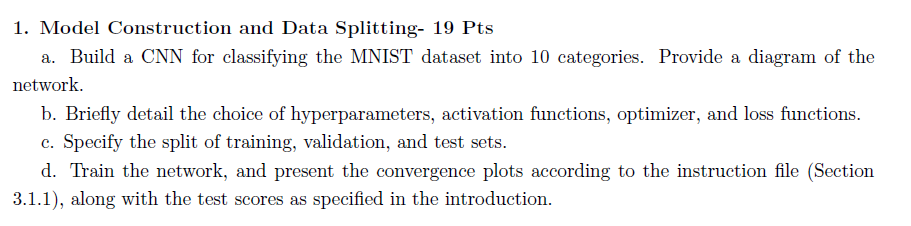
Home Work Assignment 1

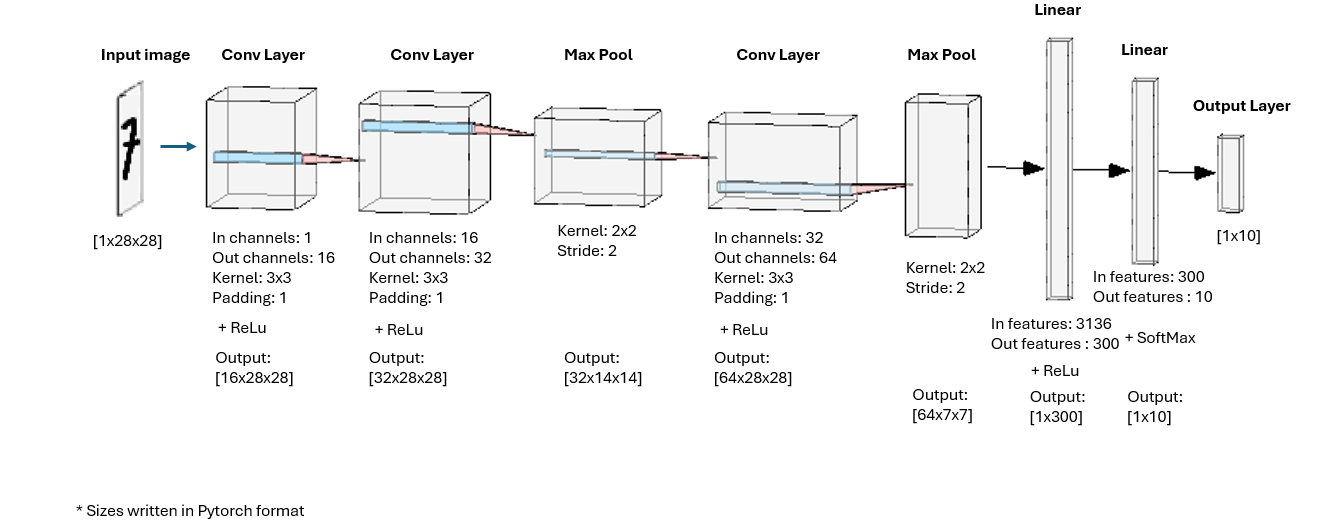
Alon Zeltser - 029644119

Nadav Amir - 308339860

**Question 1**



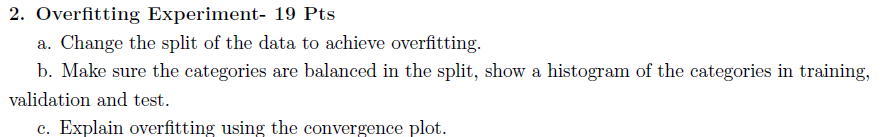
**Answer**



1. We followed a common network architecture that is used for classification problems, starting with CNN and continuing with reduction to fully connected. ReLU was used as the activation function to introduce non-linearity and minimize vanishing gradients phenomena, while SoftMax was applied at the output layer for probabilistic multi-class predictions[[1]](#footnote-1). A mini-batch size of 264 provided a balance between computational efficiency and convergence stability. We chose a learning rate of 0.001 for steady training and ran for 20 epochs to ensure sufficient learning without overfitting. Adam Optimizer was selected for its adaptive capabilities (such as per parameter learning rate and momentum). Cross-entropy loss was used, as it is suitable for multi-class classification problems.
2. We split the Train dataset into 80%/20% split resulting overall - Train: 48,000 images, Validation: 12,000 images, Test: 10,000 images (Given as a separate dataset).
3. See the next page:

|  |  |
| --- | --- |
| Figure 3 – Test Set F1 Score for each category + Macro (Average) Value | Figure 2 – Test Set Confusion Matrix |
| Figure 1 – Training vs. Validation Loss Progress Graph | |

**Question 2**



**Answer**

1. We split the Train dataset into 99.8%/0.2% split resulting overall - Train: 120 images, Validation: 59880 images, Test: 10,000 images (Given as a separate dataset).
2. We used *StratifiedShuddleSplit* to make sure that the data is balanced:

|  |  |  |
| --- | --- | --- |
| Figure 4 – Train Set Class Distribution | Figure 5 – Validation Set Class Distribution | Figure 6 – Test Set Class Distribution |

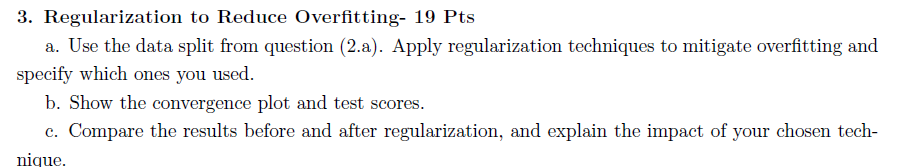
1. According to Fig 7, around epoch 12 the validation loss detaches from the training loss and begins to rise slightly, while the training loss continues converging toward zero. This suggests the model is starting to "memorize" the training set rather than generalizing.

A graph of a train vs. vallation loss

AI-generated content may be incorrect.

Figure 7 – Training vs. Validation Loss Progress Graph – Over Fitting Experiment

**Question 3**



**Answer**

1. We Applied three regularization techniques separately:
   1. Early stopping with patience of 2 epochs
   2. Dropout – we applied dropout with probability 0.05 after each layer in the model (can see specification in the attached code)
   3. L2 regularization
2. Graphs are attached for each method:

**Early stopping with patience of 2 epochs**

|  |  |
| --- | --- |
| Figure 8 – Test Set F1 Score for each category + Macro (Average) Value – Over Fitting + Early Stopping | Figure 9 – Test Set Confusion Matrix – Over Fitting + Early Stopping |
| Figure 10 – Training vs. Validation Loss Progress Graph – OverFitting + Early Stopping | |

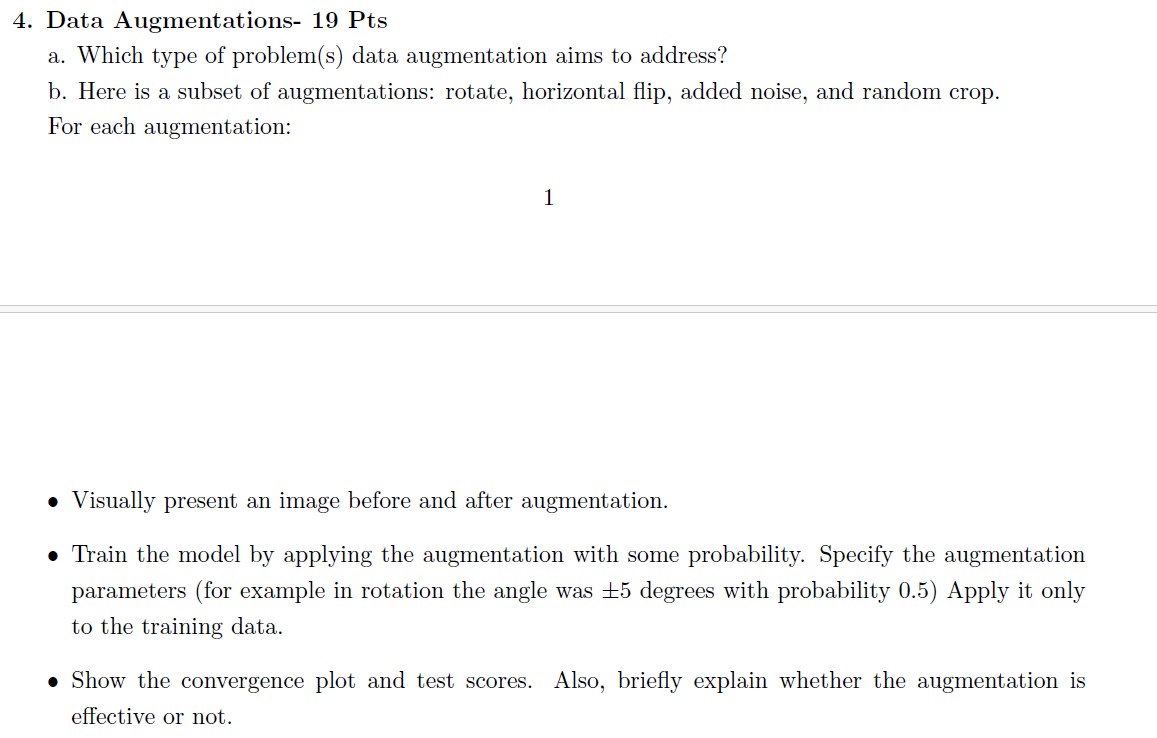
**Dropout**

|  |  |
| --- | --- |
| Figure 11 – Test Set F1 Score for each category + Macro (Average) Value – Over Fitting + Dropout | Figure 12 – Test Set Confusion Matrix – Over Fitting + Dropout |
| Figure 13 – Test Set F1 Score for Each Category + Macro (Average) Value – Over Fitting + Dropout  **L2 Regularization** | |
| Figure 14 – Test Set F1 Score for each category + Macro (Average) Value – Over Fitting + l2 regularization | Figure 15 – Test Set Confusion Matrix – Over Fitting + l2 regularization |
| Figure 16 – Test Set F1 Score for each category + Macro (Average) Value – Over Fitting + l2 regularization | |

1. Comparison. Attached is a table with macro f1 score for each regularization scenarios and for the non-regularized one.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Without regularization** | **Early stooping** | **Dropout** | **L2** |
| **Macro F1** | 0.7645 | 0.7701 | 0.8023 | 0.7865 |

Overall, the regularization techniques produced mixed results, with some training sessions showing only slight improvements over the non-regularized model. The following results are from sessions that outperformed the non-regularized model. Early Stopping: This method stops training when the validation performance stops improving. Dropout: Randomly deactivating a small subset of neurons during training, forcing the network to learn more robust features. We examined several dropout configurations, and the selected setup-applying a small dropout after each layer, appeared to provide adequate results. L2 Regularization: Adding a penalty for large weights to the loss function, L2 regularization encourages simpler models that generalize better.



A screenshot of a computer

AI-generated content may be incorrect.

**Answer**

1. Augmentation aims to address several problems when training a model
   1. **Bias Reduction**: When the data set is, for example 1) too small, or 2) unbalanced, Augmentation can be used to expand it, adding more trainable data to the network.
   2. **Generalization and Prevention of Overfitting**: Augmentation makes models more robust and reliable by mimicking real world variations (shifts, flips, rotations, etc). This approach reduces the risk of overfitting, as the model is less likely to memorize specific patterns from the training data. For example, if all training images are captured during the day, augmentation can help the model adapt to various lighting conditions that may be encountered in a real-world environment.
2. Visual representation (left to right), as shown in Figure 17:

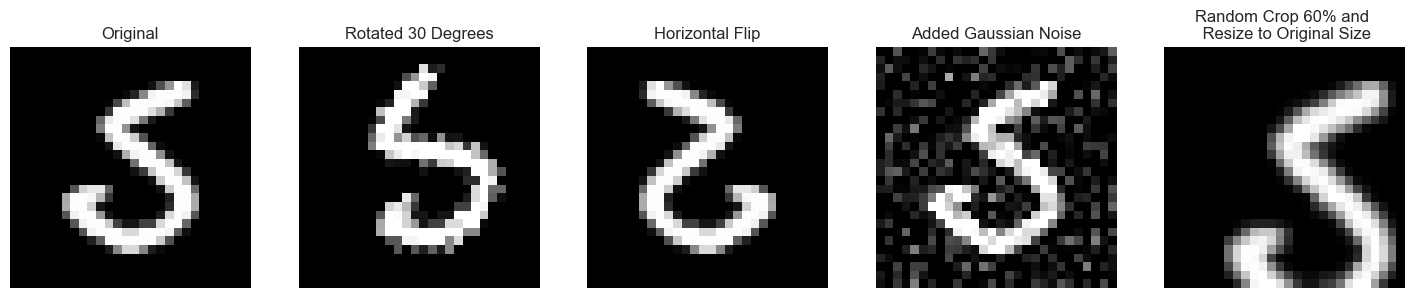
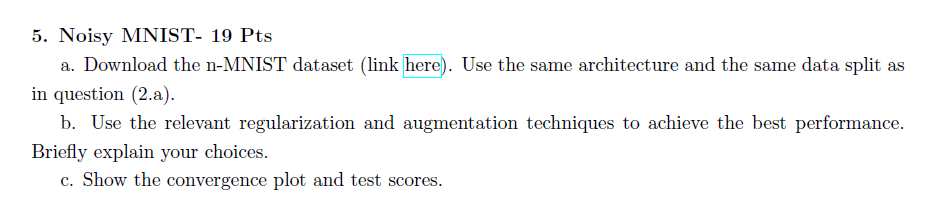


Figure 17 – Visual Representation of Augmentations

1. We ran the augmentation on Section 1 data (80%/20%). We used:
   1. Rotation: -20 degrees to +20 degrees with probability of 0.25.
   2. Horizontal flip with probability of 0.25.
   3. Gaussian noise with amplitude of 0.2 with probability of 0.25.
   4. Random crop of between 50 to 100% of the image with probability of 0.25.
2. In total, the original network achieved almost perfect results. We observed that the augmentation of the training data resulted in slightly worse results. We assume that the reason for this is that some of the augmentation techniques used are not relevant for the test data (for example: horizontal flip does not represent any test examples)

|  |  |
| --- | --- |
| Figure 17 – Test Set F1 Score for each category + Macro (Average) Value – Augmentation | Figure 18 – Test Set Confusion Matrix – Augmentation |
| Figure 19 – Test Set F1 Score for each category + Macro (Average) Value – Augmentation | |



1. Although softmax is implied for multi-class classification, we did not apply it explicitly, as PyTorch’s CrossEntropyLoss includes it internally. [↑](#footnote-ref-1)