|  |  |
| --- | --- |
| **ANN** | **CNN** |
| The maximum accuracy achieved by the Artificial Neural Network (ANN) is associated with three hidden layers. The first two layers utilize the Parametric ReLU activation function, while the last layer employs Softmax activation. The optimizer chosen is Adadelta with a learning rate of 1.0. The accuracy achieved is **99.48%** in the training set and **98.07%** in the test set, respectively. | The CNN achieved its maximum accuracy with a configuration comprising a convolutional layer followed by two max-pooling layers. In the 1st layer, there are 64 kernels with a kernel size of 4 x 4 and a max-pooling layer with a pool size of 3 x 3. The 2nd layer consists of 32 kernels, along with a max-pooling layer with a pool size of 2 x 2. Padding is applied in the next step, with the padding type set to 'same,' and ReLU is utilized as the activation function. The optimizer chosen is Adadelta. The obtained accuracy is **99.72**% in the training set and **99.11%** in the testing set, respectively. |
| The computational time taken by the model to train is 30.2 seconds. | The computational time taken by the model to train is 4 minutes and 14 seconds. |
| **Confusion Matrix:**  [[584 0 0 0 0 0 0 0 1 2]  [ 0 628 1 0 0 0 1 0 0 0]  [ 1 1 597 1 0 0 0 0 0 0]  [ 0 0 0 625 0 0 0 0 1 1]  [ 0 3 0 0 587 0 0 1 0 4]  [ 0 1 0 8 0 533 4 0 3 0]  [ 2 0 0 0 0 0 569 0 0 0]  [ 2 3 0 0 2 0 0 658 0 3]  [ 3 0 0 0 0 1 0 0 593 0]  [ 1 0 0 3 4 0 0 0 2 566]] | **Confusion Matrix:**  [[578 0 4 0 0 0 1 1 1 2]  [ 0 623 1 1 0 0 3 1 1 0]  [ 2 2 593 0 0 0 0 1 2 0]  [ 1 0 3 614 0 3 0 0 6 0]  [ 1 4 1 0 580 0 0 2 0 7]  [ 0 0 2 9 0 523 9 0 2 4]  [ 1 0 0 0 0 0 569 0 1 0]  [ 0 2 0 1 1 0 0 661 0 3]  [ 0 2 1 1 2 0 1 1 587 2]  [ 0 2 1 2 5 4 0 3 1 558]] |
| **Heat Map:** | **Heat Map:** |
| Trained with 10 Epochs | Trained with 10 Epochs |
| **ANN Summary**  Model: "sequential\_14"  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Layer (type) Output Shape Param #  =================================================================  flatten\_11 (Flatten) (None, 784) 0    dense\_49 (Dense) (None, 100) 78600    dense\_50 (Dense) (None, 50) 5100    dense\_51 (Dense) (None, 10) 510    =================================================================  Total params: 84210 (328.95 KB)  Trainable params: 84210 (328.95 KB)  Non-trainable params: 0 (0.00 Byte) | **CNN Summary**  Model: "sequential\_15"  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Layer (type) Output Shape Param #  =================================================================  conv2d\_17 (Conv2D) (None, 28, 28, 64) 1088    max\_pooling2d\_17 (MaxPooli (None, 9, 9, 64) 0  ng2D)    conv2d\_18 (Conv2D) (None, 9, 9, 32) 18464    max\_pooling2d\_18 (MaxPooli (None, 4, 4, 32) 0  ng2D)    flatten\_12 (Flatten) (None, 512) 0    dense\_52 (Dense) (None, 100) 51300    dense\_53 (Dense) (None, 50) 5050    dense\_54 (Dense) (None, 25) 1275    dense\_55 (Dense) (None, 10) 260    =================================================================  Total params: 77437 (302.49 KB)  Trainable params: 77437 (302.49 KB)  Non-trainable params: 0 (0.00 Byte) |
| **Advantages**   1. Parametric ReLU Activation: The use of Parametric ReLU activation allows the network to learn the optimal values for the alpha parameters, introducing flexibility in the activation function. 2. Sequential Model: The Sequential API in Keras simplifies the construction of neural networks, making it easy to define a linear stack of layers. 3. Custom Optimizer: The code specifies a custom Adadelta optimizer with a learning rate of 1.0, providing control over the optimization process. 4. Sparse Categorical Crossentropy: The choice of sparse categorical crossentropy as the loss function is suitable for integer-encoded target labels, common in classification tasks. 5. Training Routine: The model is trained on the flattened input data for 10 epochs, facilitating quick experimentation and model evaluation | **Advantages**   1. Convolutional Neural Network (CNN): The use of Conv2D and MaxPooling2D layers allows the model to capture spatial hierarchies and patterns in image data, making it suitable for image classification tasks. 2. Padding and Pooling: The inclusion of padding='same' in convolutional layers and max-pooling layers helps maintain spatial dimensions and reduce information loss at the borders. 3. Sequential Model: The Sequential API simplifies the model architecture, making it easy to understand and modify. It follows a sequential flow of layers, enhancing readability. 4. Custom Optimizer: The code employs a custom Adadelta optimizer with a learning rate of 1.0, offering control over the optimization process during training. 5. Sparse Categorical Crossentropy: The choice of sparse categorical crossentropy as the loss function is suitable for integer-encoded target labels, common in classification tasks. |
| **Disadvantages**   1. Parametric ReLU Overhead: Using Parametric ReLU may introduce additional parameters to learn, potentially leading to longer training times and increased memory requirements. 2. Fixed Learning Rate: The Adadelta optimizer is configured with a fixed learning rate of 1.0, which might not be optimal for all datasets or converge efficiently in certain cases. 3. Limited Model Complexity: The model architecture is relatively simple with only dense layers and lacks convolutional or recurrent layers, which may limit its ability to capture complex hierarchical patterns. | **Disadvantages**   1. Fixed Learning Rate: The Adadelta optimizer is configured with a fixed learning rate of 1.0, which might not be optimal for all datasets or converge efficiently in certain cases. 2. Limited Comments: The code lacks detailed comments, making it less clear for someone else to understand the purpose and choices made during model construction and training. 3. No Validation Set: The training routine does not include a separate validation set for monitoring model performance on unseen data during training. 4. Limited Model Complexity: The model architecture is relatively simple with only dense layers, which may limit its ability to capture complex hierarchical patterns in certain datasets. 5. Epochs and Data: The number of epochs (10) and the training data (X\_train, y\_train) are fixed, and adapting these parameters may be necessary for optimal model training, depending on the dataset and task. |