Nuo (Tony) Chen, chen2886@purdue.edu

Lab2 Report

- Git: https://github.com/Chen2886/CS390-Lab2
- List of resources:
  - https://www.cs.toronto.edu/~kriz/cifar.html
  - https://keras.io/api/datasets/cifar100/
  - https://www.tensorflow.org/datasets/catalog/cifar100
  - https://www.tensorflow.org/tutorials/images/cnn
  - https://stackoverflow.com/questions/49079115/valueerror-negative-dimension-size-caused-by-subtracting-2-from-1-for-max-pool
  - https://medium.com/@birdortyedi_23820/deep-learning-lab-episode-2-cifar-10-631aea84f11e
  -
- Part of lab completed
  - [15] Neural net models.
    - [5] Standard ANN (this can be from lab 1).
    - [10] CNN
  - [55] Accuracy
    - [10] Get an MNIST digit accuracy of 99% or higher.
    - [10] Get an MNIST fashion accuracy of 92% or higher.
    - [11] Get a cifar 10 accuracy of 70% or higher.
    - [12] Get a cifar 100 course accuracy of 50% or higher.
    - [12] Get a cifar 100 fine accuracy of 35% or higher.
    - [+3] EC: Have a cifar 100 fine accuracy of 40% or higher.
    - [+5] EC: Have a cifar 100 fine accuracy of 45% or higher.
  - [10] Pipeline & misc.
    - [2] Code the pipeline to be able to use cifar-10
    - [2] Code the pipeline to be able to use cifar-100 fine
    - [2] Code the pipeline to be able to use cifar-100 course
    - [2] Use matplotlib to create a bar graph showing your accuracy with the standard ANN for each dataset. Save this graph as an image and name the file ANN_Accuracy_Plot.pdf.
    - [2] Use matplotlib to create a bar graph showing your accuracy with the standard CNN for each dataset. Save this graph as an image and name the file CNN_Accuracy_Plot.pdf.
    - [+1] EC: Add the option to preprocess the image using random crops for data augmentation. This option will work best on the cifar datasets.
    - [+1] EC: Code in an option to save your network weights to a file for your CNN and ANN, as well as load them. You should be able to run the network again without training.
  - [20] Report.

- How is a CNN superior to standard ANNs for image processing?
    - CNN is better because it takes account the special relation.
- Why do we sometimes use pooling in CNNs?
    - Pooling is to reduce the runtime of CNN, otherwise it will be a long time to train a CNN.
- Why do you think the cifar datasets are harder than mnist?
    - Cifar images are colored, there are more classes.
- Increases the accuracy of your CNN
    - Dividing the data by 255 increase the accuracy significantly.
    - I modeled my CNN after VGG. Dropout layer increased accuracy
    - Flattened layer and Dense layer at the end increased accuracy
    - Adding batch normalization increased accuracy
- Hyperparamaters:
    - LR: 0.0008
    - Activation function: Mostly relu, softmax at the end
    - Adam optimizer
    - Categorical cross entropy loss function
    - 10 Epochs
    - Drop rate: 0.2

- MNIST F (CNN):
    - Epoch 1/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.6439 - accuracy: 0.7693
    - Epoch 2/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.3409 - accuracy: 0.8730
    - Epoch 3/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2891 - accuracy: 0.8933
    - Epoch 4/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2666 - accuracy: 0.9003
    - Epoch 5/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2514 - accuracy: 0.9076
    - Epoch 6/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2396 - accuracy: 0.9108
    - Epoch 7/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2288 - accuracy: 0.9144
    - Epoch 8/10

    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2202 - accuracy: 0.9187
    - Epoch 9/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2001 - accuracy: 0.9235
    - Epoch 10/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2014 - accuracy: 0.9232
    - Testing TF_CNN.
    - Classifier algorithm: tf_conv
    - Classifier accuracy: 91.940000%
- MNIST D (CNN):
    - Epoch 1/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.2987 - accuracy: 0.9062
    - Epoch 2/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0586 - accuracy: 0.9819
    - Epoch 3/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0456 - accuracy: 0.9866
    - Epoch 4/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0401 - accuracy: 0.9876
    - Epoch 5/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0351 - accuracy: 0.9897
    - Epoch 6/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0285 - accuracy: 0.9910
    - Epoch 7/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0270 - accuracy: 0.9915
    - Epoch 8/10
    - 1875/1875 [==============================] - 15s 8ms/step - loss: 0.0251 - accuracy: 0.9917
    - Epoch 9/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0247 - accuracy: 0.9921
    - Epoch 10/10
    - 1875/1875 [==============================] - 14s 7ms/step - loss: 0.0235 - accuracy: 0.9927
    - Testing TF_CNN.
    - Classifier algorithm: tf_conv

    - Classifier accuracy: 99.320000%
- Cifar 10 output (CNN):
    - Epoch 1/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 1.6969 - accuracy: 0.4087
    - Epoch 2/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 1.1289 - accuracy: 0.5992
    - Epoch 3/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.9683 - accuracy: 0.6585
    - Epoch 4/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.8595 - accuracy: 0.6987
    - Epoch 5/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.7974 - accuracy: 0.7234
    - Epoch 6/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.7396 - accuracy: 0.7382
    - Epoch 7/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.7094 - accuracy: 0.7484
    - Epoch 8/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.6622 - accuracy: 0.7675
    - Epoch 9/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.6363 - accuracy: 0.7776
    - Epoch 10/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 0.6144 - accuracy: 0.7842
    - Testing TF_CNN.
    - Classifier algorithm: tf_conv
    - Classifier accuracy: 72.170000%
- Cifar 100 Fine output (CNN):
    - Epoch 1/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 4.1747 - accuracy: 0.0887
    - Epoch 2/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 3.1656 - accuracy: 0.2321
    - Epoch 3/10

    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.7770 - accuracy: 0.2985
    - Epoch 4/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.5599 - accuracy: 0.3417
    - Epoch 5/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.4305 - accuracy: 0.3657
    - Epoch 6/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.3290 - accuracy: 0.3912
    - Epoch 7/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.2447 - accuracy: 0.4034
    - Epoch 8/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.1698 - accuracy: 0.4246
    - Epoch 9/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.1255 - accuracy: 0.4359
    - Epoch 10/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.0479 - accuracy: 0.4504
    - Testing TF_CNN.
    - Classifier algorithm: tf_conv
    - Classifier accuracy: 45.460000%
- Cifar 100 Coarse output (CNN):
    - Epoch 1/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.6582 - accuracy: 0.2115
    - Epoch 2/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 2.0492 - accuracy: 0.3703
    - Epoch 3/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 1.8291 - accuracy: 0.4301
    - Epoch 4/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 1.7122 - accuracy: 0.4608
    - Epoch 5/10
    - 1563/1563 [==============================] - 16s 10ms/step - loss: 1.6042 - accuracy: 0.4952
    - Epoch 6/10

- 1563/1563 [==============================] - 16s 10ms/step - loss: 1.5359 - accuracy: 0.5145
- Epoch 7/10
- 1563/1563 [==============================] - 16s 10ms/step - loss: 1.4711 - accuracy: 0.5327
- Epoch 8/10
- 1563/1563 [==============================] - 16s 10ms/step - loss: 1.4239 - accuracy: 0.5495
- Epoch 9/10
- 1563/1563 [==============================] - 16s 10ms/step - loss: 1.3842 - accuracy: 0.5625
- Epoch 10/10
- 1563/1563 [==============================] - 16s 10ms/step - loss: 1.3377 - accuracy: 0.5745
- Testing TF_CNN.
- Classifier algorithm: tf_conv
- Classifier accuracy: 53.770000%