Project #1 – Neural Network Architectures, Optimization and Regularization Techniques

Team: Ivan Liuliaev, Conor Finnegan, Rose Exeus

This is part 3 - Convolutional NN (Ivan Liuliaev)

Task I – Neural Network Design

3. Convolutional

Description:

- three convolutional layers and one fully connected
- ReLU and Tahn activation functions
- adaptive pooling
- batch normalization
- Forward pass:
    - Convolutions → BatchNorm → Activation (ReLU/Tanh) → Adaptive pooling → Flatten → Fully connected layer → Dropout → Output layer

Design Choices

- Increase number of learneable filters (32 -> 64 -> 128) as we go deeper, to catch more complex higher-level patterns more comprehensively
- Activation functions
    - ReLU for non-linearity and vanishing gradients prevention
    - Tahn to bound (normalize) the outputs
- Adaptive Pooling after the convolution to reduce dimensionality before fully-connected final layer (to reduce overfitting and reduce comp. complexity)

Task II - Techniques for Optimization

3. Convolutional

(1) Parameter initialization strategies

```
---------------------- EFFECTIVE ----------------------
Training single CNN model...
Epoch 1: Train Loss: 0.3377, Train Accuracy: 89.23%
Test Loss: 0.2039, Test Accuracy: 94.52%
Epoch 2: Train Loss: 0.0920, Train Accuracy: 97.33%
Test Loss: 0.1640, Test Accuracy: 95.22%
Epoch 3: Train Loss: 0.0579, Train Accuracy: 98.18%
Test Loss: 0.1643, Test Accuracy: 95.17%
Epoch 4: Train Loss: 0.0446, Train Accuracy: 98.74%
Test Loss: 0.1726, Test Accuracy: 95.37%
Epoch 5: Train Loss: 0.0354, Train Accuracy: 99.05%
Test Loss: 0.1542, Test Accuracy: 96.11%
Epoch 6: Train Loss: 0.0316, Train Accuracy: 99.20%
Test Loss: 0.1475, Test Accuracy: 95.91%
Epoch 7: Train Loss: 0.0275, Train Accuracy: 99.18%
Test Loss: 0.1540, Test Accuracy: 95.91%
Epoch 8: Train Loss: 0.0231, Train Accuracy: 99.36%
Test Loss: 0.1522, Test Accuracy: 96.41%
Epoch 9: Train Loss: 0.0194, Train Accuracy: 99.52%
Test Loss: 0.1542, Test Accuracy: 95.81%
Epoch 10: Train Loss: 0.0172, Train Accuracy: 99.56%
Test Loss: 0.1516, Test Accuracy: 96.31%
Epoch 11: Train Loss: 0.0152, Train Accuracy: 99.73%
Test Loss: 0.1487, Test Accuracy: 96.51%
Epoch 12: Train Loss: 0.0146, Train Accuracy: 99.60%
Test Loss: 0.1508, Test Accuracy: 96.51%
Epoch 13: Train Loss: 0.0133, Train Accuracy: 99.70%
Test Loss: 0.1550, Test Accuracy: 96.21%
Epoch 14: Train Loss: 0.0117, Train Accuracy: 99.77%
Test Loss: 0.1573, Test Accuracy: 96.21%
Epoch 15: Train Loss: 0.0135, Train Accuracy: 99.71%
Test Loss: 0.1577, Test Accuracy: 96.36%
Epoch 16: Train Loss: 0.0099, Train Accuracy: 99.81%
Test Loss: 0.1639, Test Accuracy: 96.36%
Epoch 17: Train Loss: 0.0101, Train Accuracy: 99.82%
Test Loss: 0.1654, Test Accuracy: 96.31%
```

```
Epoch 18: Train Loss: 0.0095, Train Accuracy: 99.81%
Test Loss: 0.1568, Test Accuracy: 96.36%
Epoch 19: Train Loss: 0.0068, Train Accuracy: 99.93%
Test Loss: 0.1642, Test Accuracy: 96.21%
Epoch 20: Train Loss: 0.0081, Train Accuracy: 99.82%
Test Loss: 0.1583, Test Accuracy: 96.51%


—---------------------- SLOW —----------------------
Training single CNN model...
Epoch 1: Train Loss: 1.1149, Train Accuracy: 61.57%
Test Loss: 0.3147, Test Accuracy: 90.98%
Epoch 2: Train Loss: 0.1418, Train Accuracy: 95.82%
Test Loss: 0.2591, Test Accuracy: 92.87%
Epoch 3: Train Loss: 0.0918, Train Accuracy: 97.35%
Test Loss: 0.1908, Test Accuracy: 94.67%
Epoch 4: Train Loss: 0.0715, Train Accuracy: 97.93%
Test Loss: 0.1668, Test Accuracy: 95.12%
Epoch 5: Train Loss: 0.0562, Train Accuracy: 98.27%
Test Loss: 0.1688, Test Accuracy: 95.57%
Epoch 6: Train Loss: 0.0489, Train Accuracy: 98.46%
Test Loss: 0.1578, Test Accuracy: 95.67%
Epoch 7: Train Loss: 0.0438, Train Accuracy: 98.63%
Test Loss: 0.1624, Test Accuracy: 95.52%
Epoch 8: Train Loss: 0.0390, Train Accuracy: 98.67%
Test Loss: 0.1549, Test Accuracy: 96.06%
Epoch 9: Train Loss: 0.0299, Train Accuracy: 99.01%
Test Loss: 0.1730, Test Accuracy: 95.17%
Epoch 10: Train Loss: 0.0260, Train Accuracy: 99.18%
Test Loss: 0.1481, Test Accuracy: 96.16%
Epoch 11: Train Loss: 0.0286, Train Accuracy: 99.16%
Test Loss: 0.1618, Test Accuracy: 95.81%
Epoch 12: Train Loss: 0.0239, Train Accuracy: 99.29%
Test Loss: 0.1628, Test Accuracy: 95.86%
Epoch 13: Train Loss: 0.0170, Train Accuracy: 99.62%
Test Loss: 0.1510, Test Accuracy: 96.01%
Epoch 14: Train Loss: 0.0189, Train Accuracy: 99.48%
Test Loss: 0.1667, Test Accuracy: 95.71%
Epoch 15: Train Loss: 0.0194, Train Accuracy: 99.45%
Test Loss: 0.1626, Test Accuracy: 96.26%
Epoch 16: Train Loss: 0.0169, Train Accuracy: 99.55%
Test Loss: 0.1556, Test Accuracy: 95.96%
Epoch 17: Train Loss: 0.0148, Train Accuracy: 99.55%
```

```
Test Loss: 0.1777, Test Accuracy: 95.76%
Epoch 18: Train Loss: 0.0155, Train Accuracy: 99.55%
Test Loss: 0.1601, Test Accuracy: 96.11%
Epoch 19: Train Loss: 0.0123, Train Accuracy: 99.67%
Test Loss: 0.1757, Test Accuracy: 95.86%
Epoch 20: Train Loss: 0.0134, Train Accuracy: 99.68%
Test Loss: 0.1615, Test Accuracy: 96.06%




—---------------------- TOO FAST —----------------------
Training single CNN model...
Epoch 1: Train Loss: 115.1631, Train Accuracy: 81.37%
Test Loss: 37.9726, Test Accuracy: 93.92%
Epoch 2: Train Loss: 23.0041, Train Accuracy: 94.32%
Test Loss: 33.6575, Test Accuracy: 93.27%
Epoch 3: Train Loss: 18.3046, Train Accuracy: 95.53%
Test Loss: 30.4274, Test Accuracy: 94.82%
Epoch 4: Train Loss: 12.1260, Train Accuracy: 96.50%
Test Loss: 27.4602, Test Accuracy: 95.71%
Epoch 5: Train Loss: 9.5002, Train Accuracy: 97.28%
Test Loss: 26.2504, Test Accuracy: 95.32%
Epoch 6: Train Loss: 8.6785, Train Accuracy: 97.35%
Test Loss: 28.1030, Test Accuracy: 95.12%
Epoch 7: Train Loss: 6.2175, Train Accuracy: 97.61%
Test Loss: 29.6387, Test Accuracy: 94.77%
Epoch 8: Train Loss: 5.8134, Train Accuracy: 98.01%
Test Loss: 23.2463, Test Accuracy: 96.06%
Epoch 9: Train Loss: 5.9186, Train Accuracy: 98.13%
Test Loss: 25.7301, Test Accuracy: 96.21%
Epoch 10: Train Loss: 4.0533, Train Accuracy: 98.38%
Test Loss: 27.7220, Test Accuracy: 95.27%
Epoch 11: Train Loss: 4.4310, Train Accuracy: 98.30%
Test Loss: 24.4619, Test Accuracy: 95.76%
Epoch 12: Train Loss: 4.1031, Train Accuracy: 98.59%
Test Loss: 24.4672, Test Accuracy: 95.71%
Epoch 13: Train Loss: 2.6407, Train Accuracy: 98.83%
Test Loss: 22.6899, Test Accuracy: 96.56%
Epoch 14: Train Loss: 3.1188, Train Accuracy: 98.88%
Test Loss: 21.2305, Test Accuracy: 96.41%
Epoch 15: Train Loss: 2.5810, Train Accuracy: 98.90%
Test Loss: 25.1566, Test Accuracy: 95.86%
```

```
Epoch 16: Train Loss: 2.2932, Train Accuracy: 98.94%
Test Loss: 23.1154, Test Accuracy: 95.96%
Epoch 17: Train Loss: 2.6031, Train Accuracy: 98.99%
Test Loss: 21.6009, Test Accuracy: 96.21%
Epoch 18: Train Loss: 2.6621, Train Accuracy: 99.03%
Test Loss: 24.5975, Test Accuracy: 95.86%
Epoch 19: Train Loss: 1.9132, Train Accuracy: 99.20%
Test Loss: 24.4067, Test Accuracy: 95.91%
Epoch 20: Train Loss: 2.3963, Train Accuracy: 99.07%
Test Loss: 24.6544, Test Accuracy: 95.57%
```

Comment:

We see how the slow one does not decrease the loss to 0.17 and less until epoch 4, while effective strategy reaches this loss by the 2nd epoch. Too fast initialization fails to stabilize around the local minima due to weights starting from excessively large values leading to instability in gradient descent.

(2) Learning rate

```
—----------------------TOO SLOW —----------------------
LEARNING_RATE = 0.0001
Training single CNN model...
Epoch 1: Train Loss: 1.8831, Train Accuracy: 36.58%
Test Loss: 1.2508, Test Accuracy: 62.03%
Epoch 2: Train Loss: 1.1267, Train Accuracy: 65.85%
Test Loss: 0.8929, Test Accuracy: 78.03%
Epoch 3: Train Loss: 0.8341, Train Accuracy: 77.08%
Test Loss: 0.7155, Test Accuracy: 82.66%
Epoch 4: Train Loss: 0.6872, Train Accuracy: 81.05%
Test Loss: 0.6124, Test Accuracy: 85.25%
Epoch 5: Train Loss: 0.5850, Train Accuracy: 84.39%
Test Loss: 0.5475, Test Accuracy: 86.50%
Epoch 6: Train Loss: 0.5152, Train Accuracy: 86.48%
Test Loss: 0.5012, Test Accuracy: 87.39%
*Interrupted*


—---------------------- EFFECTIVE —----------------------
LEARNING_RATE = 0.03
Training single CNN model...
Epoch 1: Train Loss: 0.2536, Train Accuracy: 91.94%
```

```
Test Loss: 0.1967, Test Accuracy: 95.47%
Epoch 2: Train Loss: 0.0670, Train Accuracy: 97.96%
Test Loss: 0.1832, Test Accuracy: 95.57%
Epoch 3: Train Loss: 0.0435, Train Accuracy: 98.72%
Test Loss: 0.1723, Test Accuracy: 96.06%
Epoch 4: Train Loss: 0.0371, Train Accuracy: 98.90%
Test Loss: 0.1922, Test Accuracy: 95.67%
*Interrupted*


—---------------------- TOO FAST —----------------------
LEARNING_RATE = 0.10
Training single CNN model...
Epoch 1: Train Loss: 1.3306, Train Accuracy: 87.29%
Test Loss: 5.1437, Test Accuracy: 87.10%
Epoch 2: Train Loss: 872.5229, Train Accuracy: 80.94%
Test Loss: 8975.7860, Test Accuracy: 74.34%
Epoch 3: Train Loss: nan, Train Accuracy: 18.23%
Test Loss: nan, Test Accuracy: 17.89%
*Interrupted*
```

Comment:

We can see effective LR decreases the loss quite fast yet the decrease is steady.
LR too small is steady but decreases the loss very slowly.
LR too large sends error to the stratosphere. One possible explanation is it is bouncing off of the “walls” of the local minima.

(3) How the batch size would impact the batch normalization for your convolutional neural network. Then demonstrate an effective batch size and an ineffective batch size on the convolutional neural network you have.

The bigger the batch - the more stable statistics used for normalizing will be, due to statistical significance and standard sampling error. Hence - better normalization.
Effective Batch Size here is a 64 or larger. Ineffective is like a 4.

```
BATCH_SIZE = 64
Training single CNN model...
Epoch 1: Train Loss: 0.2536, Train Accuracy: 91.94%
Test Loss: 0.1967, Test Accuracy: 95.47%
Epoch 2: Train Loss: 0.0670, Train Accuracy: 97.96%
```

```
Test Loss: 0.1832, Test Accuracy: 95.57%
Epoch 3: Train Loss: 0.0435, Train Accuracy: 98.72%
Test Loss: 0.1723, Test Accuracy: 96.06%
Epoch 4: Train Loss: 0.0371, Train Accuracy: 98.90%
Test Loss: 0.1922, Test Accuracy: 95.67%
*Interrupted*


BATCH_SIZE = 4
Training single CNN model...
Epoch 1: Train Loss: nan, Train Accuracy: 16.51%
Test Loss: nan, Test Accuracy: 17.89%
Epoch 2: Train Loss: nan, Train Accuracy: 16.38%
Test Loss: nan, Test Accuracy: 17.89%
```

(4) Momentum

```
MOMENTUM = 0.5
Training single CNN model...
Epoch 1: Train Loss: 0.3022, Train Accuracy: 90.95%
Test Loss: 0.2159, Test Accuracy: 93.57%
Epoch 2: Train Loss: 0.1057, Train Accuracy: 97.08%
Test Loss: 0.1744, Test Accuracy: 95.07%
Epoch 3: Train Loss: 0.0735, Train Accuracy: 98.07%
Test Loss: 0.1568, Test Accuracy: 95.37%
Epoch 4: Train Loss: 0.0592, Train Accuracy: 98.29%
Test Loss: 0.1448, Test Accuracy: 95.76%
Epoch 5: Train Loss: 0.0483, Train Accuracy: 98.63%
Test Loss: 0.1406, Test Accuracy: 95.86%
*Interrupted*


MOMENTUM = 0.9
Training single CNN model...
Epoch 1: Train Loss: 0.2536, Train Accuracy: 91.94%
Test Loss: 0.1967, Test Accuracy: 95.47%
Epoch 2: Train Loss: 0.0670, Train Accuracy: 97.96%
Test Loss: 0.1832, Test Accuracy: 95.57%
Epoch 3: Train Loss: 0.0435, Train Accuracy: 98.72%
Test Loss: 0.1723, Test Accuracy: 96.06%
```

```
Epoch 4: Train Loss: 0.0371, Train Accuracy: 98.90%
Test Loss: 0.1922, Test Accuracy: 95.67%
*Interrupted*


MOMENTUM = 0.99
Training single CNN model...
Epoch 1: Train Loss: 1.3857, Train Accuracy: 87.77%
Test Loss: 5.5538, Test Accuracy: 89.99%
Epoch 2: Train Loss: 61.1896, Train Accuracy: 88.23%
Test Loss: 862.7408, Test Accuracy: 73.39%
Epoch 3: Train Loss: 30162.5725, Train Accuracy: 84.05%
Test Loss: 411018.6796, Test Accuracy: 71.25%
Epoch 4: Train Loss: 76407678326.3459, Train Accuracy: 57.50%
Test Loss: 1900891257663.6492, Test Accuracy: 13.15%
*Interrupted*
```

Comment:

Momentum = 0.5
Slow but Stable
Gradients are not accumulated aggressively, making updates more independent.

Momentum = 0.9
Balance closer to Optimal
It smooths updates without overshooting, leading to faster convergence and stable generalization.

Momentum = 0.99
Training Collapse
Gradient updates accumulate excessively, leading to exploding loss values.

Task III - Techniques for Improving Generalization

3. Convolutional

Use an ensemble to improve the generalization performance

For the ensemble, I am using three NNs of the same architecture and parameres which proved to be the best. They will still introduce some diversity due to the weight initialized randomly in each, as well as due to dropout, which is also random.

```
Training ensemble of 3 CNN models...

Training model 1 of the ensemble:
Epoch 1: Train Loss: 0.2563, Train Accuracy: 91.77%
Epoch 2: Train Loss: 0.0877, Train Accuracy: 97.56%
Epoch 3: Train Loss: 0.0541, Train Accuracy: 98.29%
Epoch 4: Train Loss: 0.0423, Train Accuracy: 98.61%
Epoch 5: Train Loss: 0.0301, Train Accuracy: 98.99%
Epoch 6: Train Loss: 0.0282, Train Accuracy: 99.16%
Epoch 7: Train Loss: 0.0209, Train Accuracy: 99.40%

Training model 2 of the ensemble:
Epoch 1: Train Loss: 0.2599, Train Accuracy: 91.70%
Epoch 2: Train Loss: 0.0721, Train Accuracy: 97.74%
Epoch 3: Train Loss: 0.0495, Train Accuracy: 98.57%
Epoch 4: Train Loss: 0.0414, Train Accuracy: 98.68%
Epoch 5: Train Loss: 0.0327, Train Accuracy: 98.88%
Epoch 6: Train Loss: 0.0240, Train Accuracy: 99.30%
Epoch 7: Train Loss: 0.0198, Train Accuracy: 99.51%

Training model 3 of the ensemble:
Epoch 1: Train Loss: 0.2423, Train Accuracy: 92.10%
Epoch 2: Train Loss: 0.0779, Train Accuracy: 97.61%
Epoch 3: Train Loss: 0.0466, Train Accuracy: 98.52%
Epoch 4: Train Loss: 0.0391, Train Accuracy: 98.96%
Epoch 5: Train Loss: 0.0290, Train Accuracy: 99.04%
Epoch 6: Train Loss: 0.0217, Train Accuracy: 99.31%
Epoch 7: Train Loss: 0.0212, Train Accuracy: 99.36%

Ensemble Test Accuracy: 96.36%
```

Comment:

Test Accuracy is on par with with a single NN of the same structure, which is expected since the difference due to the random factor here should’ve been too large here. Also, a single NN was already not overfitted and had a decent accuracy.