Report: Assignment 4

---

1) A Basic CNN

Architecture Description:

The basic CNN architecture consists of three convolutional layers followed by max-pooling layers and two fully connected layers. The convolutional layers have 16, 32, and 64 filters respectively, with each filter having a kernel size of 3x3 and padding of 1 to maintain the spatial dimensions. Max-pooling layers with a kernel size of 2x2 and stride of 2 are applied after each convolutional layer to reduce spatial dimensions. The output of the convolutional layers is flattened and fed into two fully connected layers with 512 and the number of classes neurons respectively. ReLU activation function is used after each layer except the final fully connected layer, which uses softmax for multi-class classification.

Name	Type	Params
conv1	Conv2d	448
conv2	Conv2d	4.6 K
conv3	Conv2d	18.5 K
pool	MaxPool2d	0
fc1	Linear	2.1 M
fc2	Linear	5.1 K

Training Procedure:

The model is trained using the Adam optimizer with a learning rate of 1e-3. The training procedure includes early stopping with a patience of 5 epochs to prevent overfitting. During training, the cross-entropy loss is minimized. Training is conducted until convergence.

Results:

- **Training Accuracy:** The final training accuracy achieved is 83.44%.
- **Training Loss:** The final training loss achieved is 0.274.
- **Validation Accuracy:** The validation accuracy achieved is 64.41%.
- **Validation Loss:** The validation loss achieved is 1.164.

Test Metrics:

- **Test Accuracy:** The test accuracy achieved on the unseen test dataset is 62.67%.
- **Test Loss:** The test loss obtained is 1.27.

Conclusion:

The basic CNN architecture achieved decent performance on the Imagenette dataset. The model shows some degree of overfitting, as evidenced by the difference in performance between the training and test datasets. Further experimentation with regularization techniques or more complex architectures may improve generalization performance.

2) All Convolutional Network

Architecture

The All Convolutional Network (AllConvNet) architecture consists of several convolutional layers followed by max pooling layers. Unlike traditional CNN architectures, AllConvNet does not include any fully connected layers, making it a fully convolutional network. The absence of fully connected layers allows AllConvNet to process input images of variable sizes.

Layers:

1. **Convolutional Layers**:
   - `conv1` : 2D convolutional layer with 3 input channels, 32 output channels, kernel size 3x3, and padding 1.
   - `conv2` : 2D convolutional layer with 32 input channels, 64 output channels, kernel size 3x3, and padding 1.
   - `conv3` : 2D convolutional layer with 64 input channels, 128 output channels, kernel size 3x3, and padding 1.
   - `conv4` : Output convolutional layer with 128 input channels and `num_classes` output channels (determined by the number of classes in the dataset), kernel size 3x3, and padding 1.
2. **Pooling Layer**:
   - `pool` : Max pooling layer with kernel size 2x2 and stride 2.
3. **Activation Function**:
   - ReLU activation function is applied after each convolutional layer.
4. **Global Average Pooling**:
   - After the final convolutional layer, global average pooling is applied to convert the spatial features into a 1D vector.

Name	Type	Params
conv1	Conv2d	896
conv2	Conv2d	18.5 K
conv3	Conv2d	73.9 K
pool	MaxPool2d	0
conv4	Conv2d	11.5 K

- 104 K Trainable params
- 0 Non-trainable params
- 104 K Total params
- 0.419 Total estimated model params size (MB)

Results

Train and Validation Metrics

- **Train Accuracy:** 33.36%
- **Train Loss:** 1.87
- **Validation Accuracy:** 33.47%
- **Validation Loss:** 1.86

Test Metrics

- **Test Accuracy:** 30.16%
- **Test Loss:** 1.94

3) Regularization: Data Augmentation

Regularization is a set of techniques used to prevent overfitting in machine learning models. Overfitting occurs when a model learns to memorize the training data instead of generalizing patterns, leading to poor performance on unseen data. One common regularization technique is data augmentation.

Data Augmentation

Data augmentation involves applying various transformations to the training data to create new, slightly modified versions of the original samples. By introducing these variations, data augmentation helps the model learn more robust and invariant features, thus improving its ability to generalize to unseen data.

Regularization has been applied on the model 2 (All Convolutional Network):

- **Random Resized Crop (64x64):** Randomly crops and resizes images to a size of 64x64 pixels, providing different views of the input images.
- **Random Horizontal Flip:** Randomly flips images horizontally, increasing the diversity of the training samples.

- **Color Jitter:** Randomly adjusts the brightness, contrast, saturation, and hue of images, making the model more robust to changes in lighting conditions and color variations.
- **Random Rotation (up to 15 degrees):** Randomly rotates images up to 15 degrees, simulating different orientations of objects in the images.

Model Performance:

Metric	Without Regularization	With Regularization
Train Accuracy	33.36%	60.81%
Train Loss	1.87	1.22
Test Accuracy	30.16%	67.92%
Test Loss	1.94	1.12

Train Accuracy:

With regularization, the train accuracy increased from 33.36% to 60.81%, indicating that the model's performance on the training data significantly improved due to the regularization techniques applied during training.

Train Loss:

The train loss with regularization decreased from 1.87 to 1.22, suggesting that the regularization techniques helped in reducing the model's training loss, indicating better convergence during training.

Validation Loss:

The validation loss with regularization decreased from 1.86 to 1.09, suggesting that the regularization techniques helped in reducing the model's training loss, indicating better convergence during training.

Test Accuracy:

The test accuracy with regularization increased from 30.16% to 67.92%, showing a significant improvement in performance on unseen data, likely due to the regularization techniques preventing overfitting on the training data.

Test Loss:

With regularization, the test loss decreased from 1.94 to 1.12, indicating that the regularization techniques significantly reduced generalization error, leading to a lower loss on the test data compared to the model without regularization.

Transfer Learning

Selected a pretrained model from previous runs - Regularization_Model_data_aug.pth

1) Trained from Scratch:

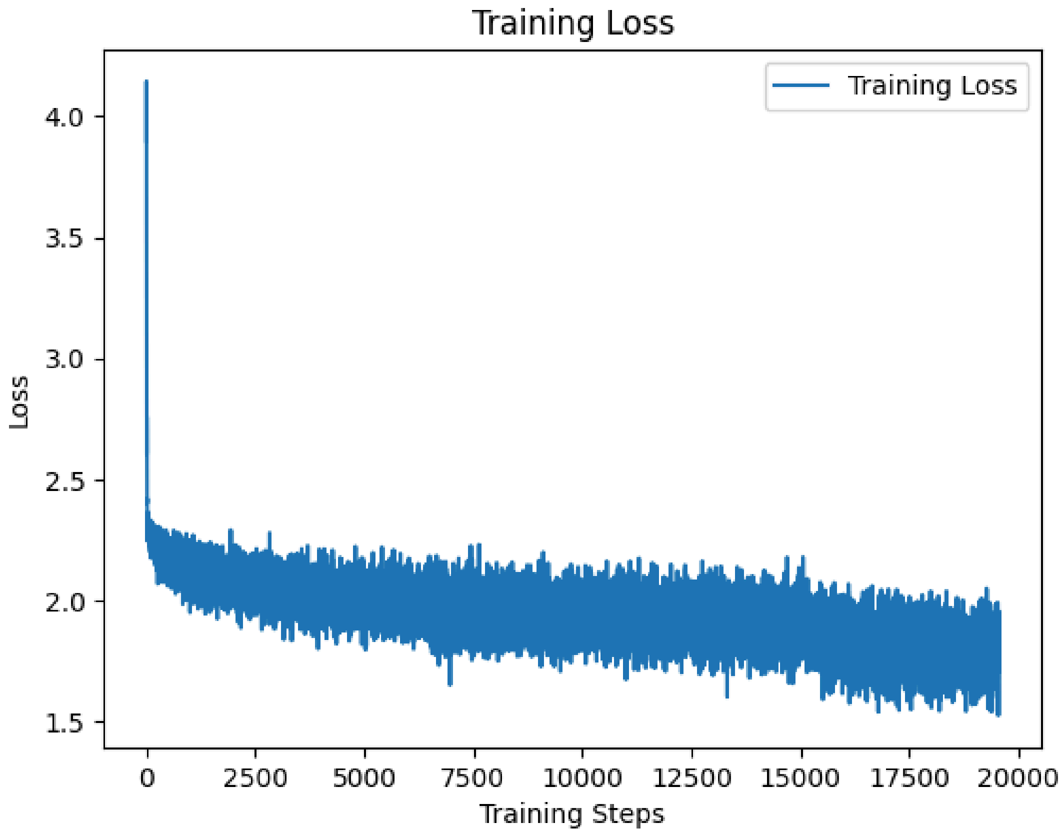
Evaluation:

- **Train Accuracy:**13.59%
- **Train Loss:**2.1176
- **Test Accuracy:**7.12%
- **Test Loss:**2.3030

2) Transfer Learning: Finetuning the pretrained model (Imagenette) from Regularization: Data Augmentation on the CIFAR10 dataset

Evaluation:

- - **Train Accuracy:**14.09%
  - **Train Loss:**1.9016
  - **Test Accuracy:**10%
  - **Test Loss:**2.3926
- **train loss:**



A Basic CNN with fully connected is found to be performing good when compared to other models

Basic_CNN_Model.pth is model file (in zip file)

```
In [ ]:
```

In [ ]:

```
```

In [ ]:

```
```