Function Approximation with Neural Network and Backpropagation

Asif Azad — 1905004

November 11, 2024

Contents

1 Introduction

In this assignment, a Feed-Forward Neural Network (FNN) is implemented from scratch to classify apparel. The goal is to develop a custom neural network model without using any deep learning frameworks. This implementation includes essential components and algorithms required for training, including the backpropagation algorithm, mini-batch gradient descent with Adam optimization, and multi-class classification using Softmax.

2 Basic Components of the Network

The following components form the basic building blocks of the neural network:

- **Dense Layer**: A fully connected layer, where each neuron in the previous layer is connected to every neuron in the current layer. The size of the layer is defined by the input and output dimensions.
- **Normalization**: Batch Normalization is used to normalize the output of each layer to improve training speed and stability, aiding faster convergence.
- **Activation**: ReLU (Rectified Linear Unit) is applied as the activation function to introduce non-linearity and help the network learn complex patterns.
- **Regularization**: Dropout is employed as a regularization technique to prevent overfitting by randomly disabling certain neurons during training, ensuring better generalization.
- **Optimization**: Adam (Adaptive Moment Estimation) is utilized as the optimizer, combining the advantages of both momentum and RMSProp to adapt the learning rate for each parameter during training.
- **Regression**: Softmax is used in the output layer for multi-class classification, converting the raw outputs into a probability distribution over all possible classes.

3 How to Run the Code

To run the code, simply press the `Run All` button in the Jupyter notebook. This will execute all the cells sequentially and complete the model training, evaluation, and testing.

3.1 Required Libraries

The following Python libraries need to be installed in your environment to run the code:

- `numpy`: For numerical operations and matrix manipulations.
- `pickle`: For saving and loading trained models.
- `matplotlib`: For plotting graphs and visualizations.
- `seaborn`: For enhanced data visualization.
- `scikit-learn`: For calculating metrics such as F1 score, confusion matrix, and accuracy, and for data splitting.
- `tqdm`: For displaying progress bars during training.
- `torchvision`: For loading standard datasets like MNIST or FashionMNIST, and for data augmentation.

3.2 Installation Instructions

You can install the required libraries using `pip`. Run the following command in your terminal:

```
pip install numpy pickle5 matplotlib seaborn scikit-learn tqdm torchvision
```

Make sure all these libraries are installed before running the notebook.

4 Model Architectures

4.1 ModelA

Input (784)

↓

Dense (128)

↓

ReLU

↓

Dense (64)

↓

ReLU

↓

Dense (10) (Output)

4.2 ModelB

Input (784)

↓

Dense (324)

↓

ReLU

↓

Dropout

↓

Dense (128)

↓

ReLU

↓

Dropout

↓

Dense (64)

↓

ReLU

↓

Dense (10) (Output)

4.3 ModelC

Input (784)

↓

Dense (784, 324)

↓

BatchNorm (324)

↓

ReLU

↓

Dropout

↓

Dense (324, 128)

↓

BatchNorm (128)

↓

ReLU

↓

Dropout

↓

Dense (128, 64)

↓

BatchNorm (64)

↓

ReLU

↓

Dense (64, 10) (Output)

5 Experiments

5.1 ModelA, LR=0.005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.5899	0.7801	0.4460	0.8372	0.8348
2	0.4038	0.8520	0.3913	0.8613	0.8597
3	0.3678	0.8658	0.3732	0.8682	0.8673
4	0.3462	0.8730	0.3638	0.8682	0.8677
5	0.3300	0.8792	0.3776	0.8685	0.8669
6	0.3196	0.8830	0.3705	0.8720	0.8714
7	0.3054	0.8886	0.3746	0.8675	0.8664
8	0.2987	0.8904	0.3861	0.8742	0.8737
9	0.2883	0.8942	0.3656	0.8773	0.8764
10	0.2858	0.8946	0.3642	0.8747	0.8746
11	0.2792	0.8960	0.3799	0.8775	0.8767
12	0.2729	0.8989	0.3725	0.8801	0.8795
13	0.2681	0.9009	0.3710	0.8782	0.8773
14	0.2617	0.9049	0.3855	0.8730	0.8719
15	0.2569	0.9051	0.3801	0.8781	0.8780
16	0.2580	0.9051	0.4186	0.8722	0.8707
17	0.2486	0.9069	0.3861	0.8782	0.8783
18	0.2443	0.9092	0.3901	0.8785	0.8786
19	0.2433	0.9090	0.3879	0.8795	0.8801
20	0.2414	0.9108	0.3991	0.8778	0.8783

Table 1: Training and Validation Metrics for ModelA with Learning Rate = 0.005

Figure 1: Plots for ModelA and LR=0.005

5.2 ModelA, LR=0.001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.7424	0.7228	0.5510	0.8047	0.7981
2	0.4770	0.8311	0.4600	0.8375	0.8321
3	0.4154	0.8514	0.4157	0.8512	0.8470
4	0.3774	0.8646	0.3886	0.8612	0.8573
5	0.3516	0.8729	0.3677	0.8688	0.8655
6	0.3308	0.8791	0.3559	0.8734	0.8705
7	0.3149	0.8852	0.3455	0.8789	0.8767
8	0.3011	0.8903	0.3394	0.8796	0.8776
9	0.2887	0.8946	0.3361	0.8812	0.8798
10	0.2769	0.8992	0.3332	0.8837	0.8827
11	0.2671	0.9025	0.3348	0.8822	0.8812
12	0.2561	0.9057	0.3391	0.8838	0.8830
13	0.2463	0.9095	0.3337	0.8866	0.8858
14	0.2384	0.9121	0.3344	0.8868	0.8860
15	0.2300	0.9153	0.3348	0.8883	0.8876
16	0.2204	0.9190	0.3418	0.8883	0.8876
17	0.2128	0.9219	0.3457	0.8874	0.8871
18	0.2042	0.9246	0.3577	0.8869	0.8866
19	0.2003	0.9251	0.3603	0.8872	0.8870
20	0.1912	0.9304	0.3666	0.8862	0.8860

Table 2: Training and Validation Metrics for ModelA with Learning Rate = 0.001

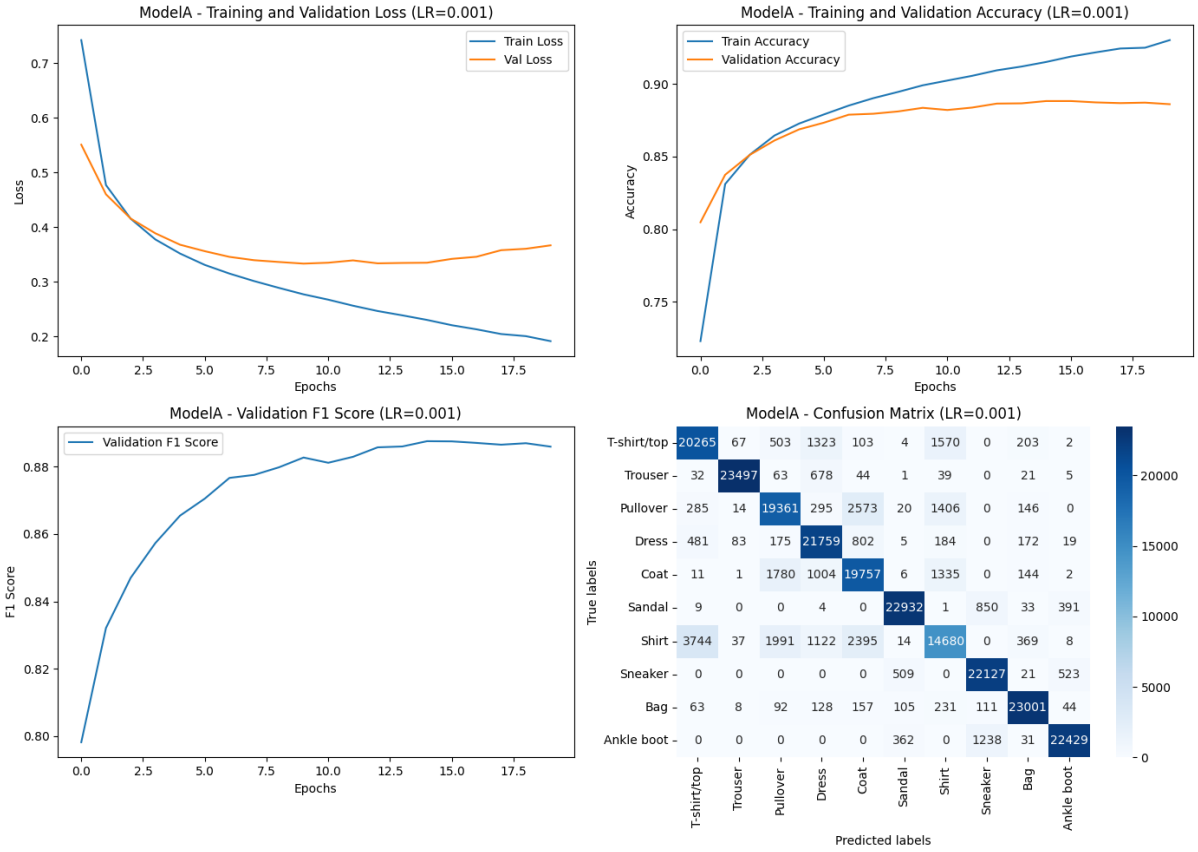

Figure 2: Plots for ModelA and LR=0.001

5.3 ModelA, LR=0.0005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.8640	0.6801	0.6301	0.7718	0.7664
2	0.5551	0.8037	0.5377	0.8133	0.8103
3	0.4803	0.8307	0.4806	0.8304	0.8281
4	0.4382	0.8448	0.4460	0.8426	0.8407
5	0.4098	0.8536	0.4233	0.8529	0.8513
6	0.3889	0.8613	0.4054	0.8587	0.8573
7	0.3718	0.8674	0.3899	0.8639	0.8627
8	0.3569	0.8726	0.3779	0.8681	0.8672
9	0.3438	0.8766	0.3701	0.8720	0.8712
10	0.3316	0.8802	0.3617	0.8755	0.8749
11	0.3204	0.8838	0.3540	0.8771	0.8766
12	0.3102	0.8874	0.3490	0.8777	0.8772
13	0.3006	0.8909	0.3465	0.8788	0.8784
14	0.2916	0.8940	0.3448	0.8803	0.8801
15	0.2837	0.8970	0.3418	0.8811	0.8808
16	0.2762	0.8997	0.3395	0.8823	0.8822
17	0.2689	0.9024	0.3395	0.8832	0.8831
18	0.2620	0.9048	0.3382	0.8847	0.8845
19	0.2554	0.9070	0.3389	0.8848	0.8847
20	0.2494	0.9089	0.3397	0.8862	0.8862

Table 3: Training and Validation Metrics for ModelA with Learning Rate = 0.0005

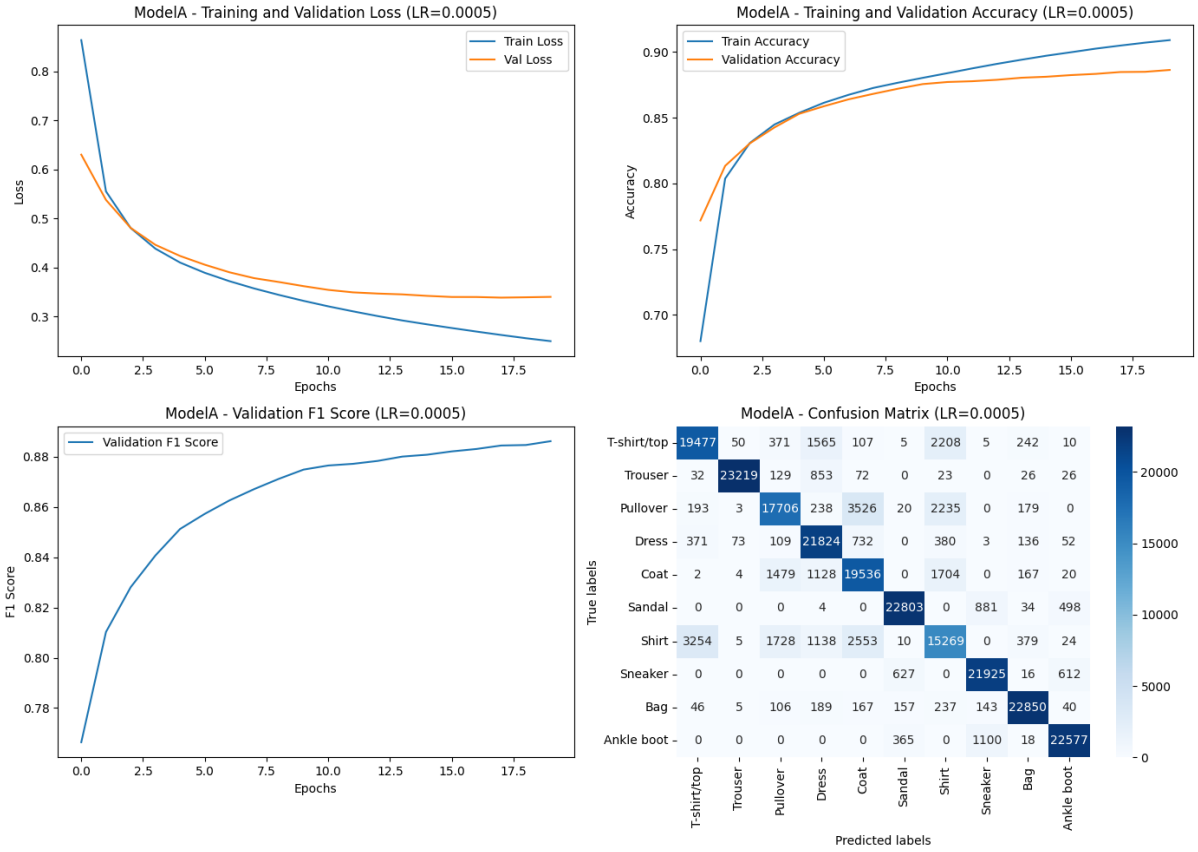

Figure 3: Plots for ModelA and LR=0.0005

5.4 ModelA, LR=0.0001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	1.2686	0.5538	0.8562	0.6711	0.6415
2	0.7677	0.7088	0.7150	0.7385	0.7288
3	0.6645	0.7581	0.6436	0.7678	0.7628
4	0.6093	0.7819	0.6002	0.7900	0.7876
5	0.5733	0.7985	0.5692	0.8032	0.8018
6	0.5468	0.8106	0.5460	0.8132	0.8124
7	0.5257	0.8182	0.5266	0.8180	0.8174
8	0.5076	0.8246	0.5100	0.8220	0.8215
9	0.4918	0.8307	0.4956	0.8261	0.8257
10	0.4782	0.8348	0.4833	0.8284	0.8282
11	0.4663	0.8385	0.4726	0.8334	0.8333
12	0.4558	0.8418	0.4632	0.8372	0.8370
13	0.4466	0.8446	0.4553	0.8411	0.8409
14	0.4385	0.8470	0.4483	0.8434	0.8433
15	0.4312	0.8500	0.4421	0.8451	0.8449
16	0.4245	0.8522	0.4365	0.8465	0.8464
17	0.4184	0.8539	0.4314	0.8478	0.8477
18	0.4128	0.8559	0.4266	0.8493	0.8493
19	0.4075	0.8579	0.4222	0.8512	0.8512
20	0.4025	0.8595	0.4182	0.8522	0.8522

Table 4: Training and Validation Metrics for ModelA with Learning Rate = 0.0001

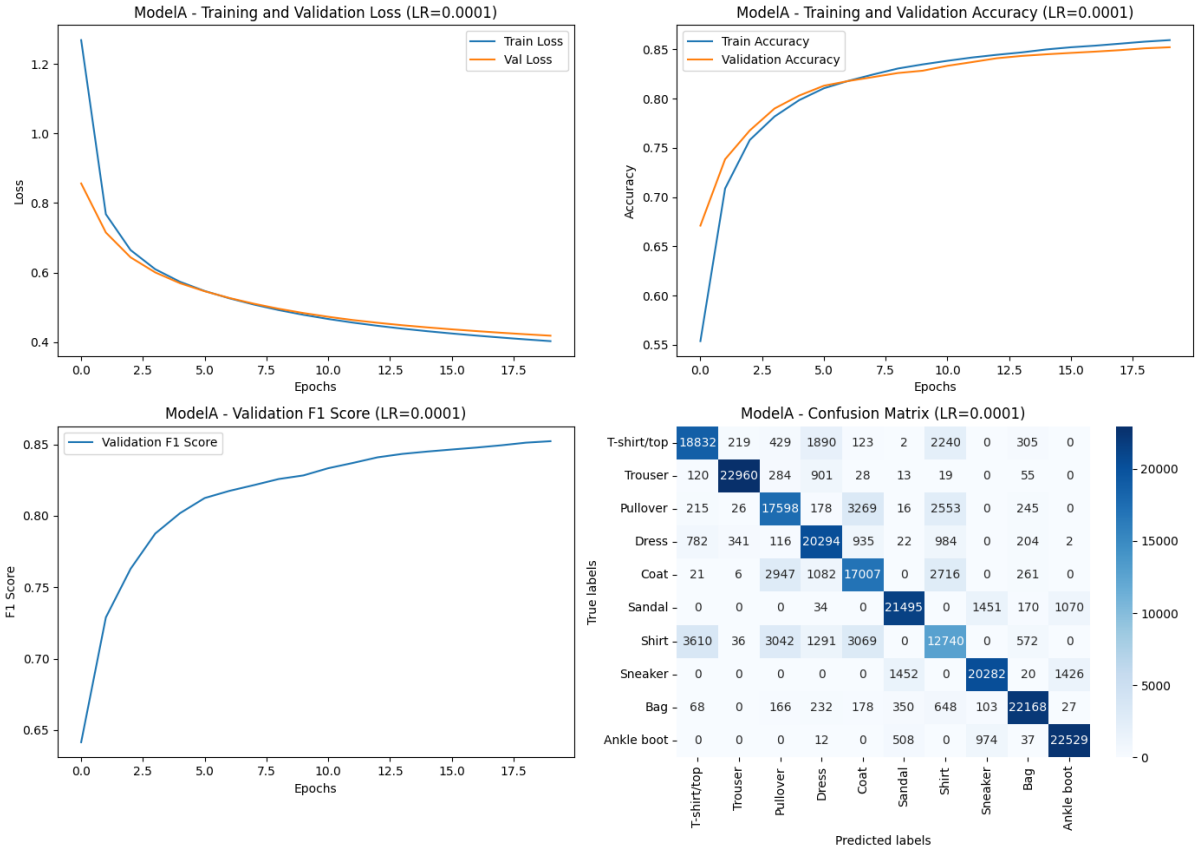

Figure 4: Plots for ModelA and LR=0.0001

5.5 ModelB, LR=0.005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.8480	0.6695	0.6801	0.7554	0.7533
2	0.6458	0.7656	0.6255	0.7759	0.7758
3	0.6063	0.7809	0.6089	0.7863	0.7828
4	0.5739	0.7947	0.6109	0.7886	0.7872
5	0.5642	0.8010	0.5765	0.8017	0.7984
6	0.5438	0.8065	0.5767	0.8000	0.7960
7	0.5396	0.8073	0.5526	0.8098	0.8091
8	0.5344	0.8102	0.5414	0.8109	0.8102
9	0.5259	0.8141	0.5718	0.7999	0.7965
10	0.5165	0.8154	0.5476	0.8103	0.8083
11	0.5213	0.8155	0.5494	0.8147	0.8146
12	0.5201	0.8167	0.5362	0.8138	0.8119
13	0.5055	0.8214	0.5350	0.8133	0.8090
14	0.5093	0.8213	0.5480	0.8113	0.8112
15	0.5088	0.8196	0.5372	0.8157	0.8136
16	0.5028	0.8219	0.5329	0.8152	0.8168
17	0.5045	0.8210	0.5490	0.8133	0.8136
18	0.4995	0.8251	0.5378	0.8200	0.8192
19	0.5011	0.8249	0.5317	0.8197	0.8196
20	0.4965	0.8241	0.5374	0.8148	0.8144

Table 5: Training and Validation Metrics for ModelB with Learning Rate = 0.005

Figure 5: Plots for ModelA and LR=0.005

5.6 ModelB, LR=0.001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.9251	0.6303	0.6405	0.7655	0.7514
2	0.5660	0.7893	0.5454	0.8048	0.8022
3	0.4928	0.8223	0.4757	0.8351	0.8344
4	0.4587	0.8340	0.4552	0.8391	0.8385
5	0.4372	0.8431	0.4398	0.8469	0.8464
6	0.4208	0.8487	0.4390	0.8458	0.8463
7	0.4082	0.8533	0.4344	0.8454	0.8449
8	0.3970	0.8563	0.4315	0.8466	0.8479
9	0.3860	0.8606	0.4133	0.8542	0.8548
10	0.3783	0.8627	0.4056	0.8567	0.8567
11	0.3736	0.8656	0.3879	0.8622	0.8621
12	0.3687	0.8672	0.4125	0.8555	0.8551
13	0.3613	0.8702	0.4177	0.8542	0.8537
14	0.3558	0.8709	0.4053	0.8574	0.8571
15	0.3529	0.8742	0.3939	0.8635	0.8632
16	0.3484	0.8742	0.4099	0.8573	0.8573
17	0.3427	0.8770	0.3968	0.8612	0.8614
18	0.3404	0.8752	0.3863	0.8642	0.8649
19	0.3383	0.8775	0.3895	0.8621	0.8608
20	0.3356	0.8780	0.3918	0.8607	0.8604

Table 6: Training and Validation Metrics for ModelB with Learning Rate = 0.001

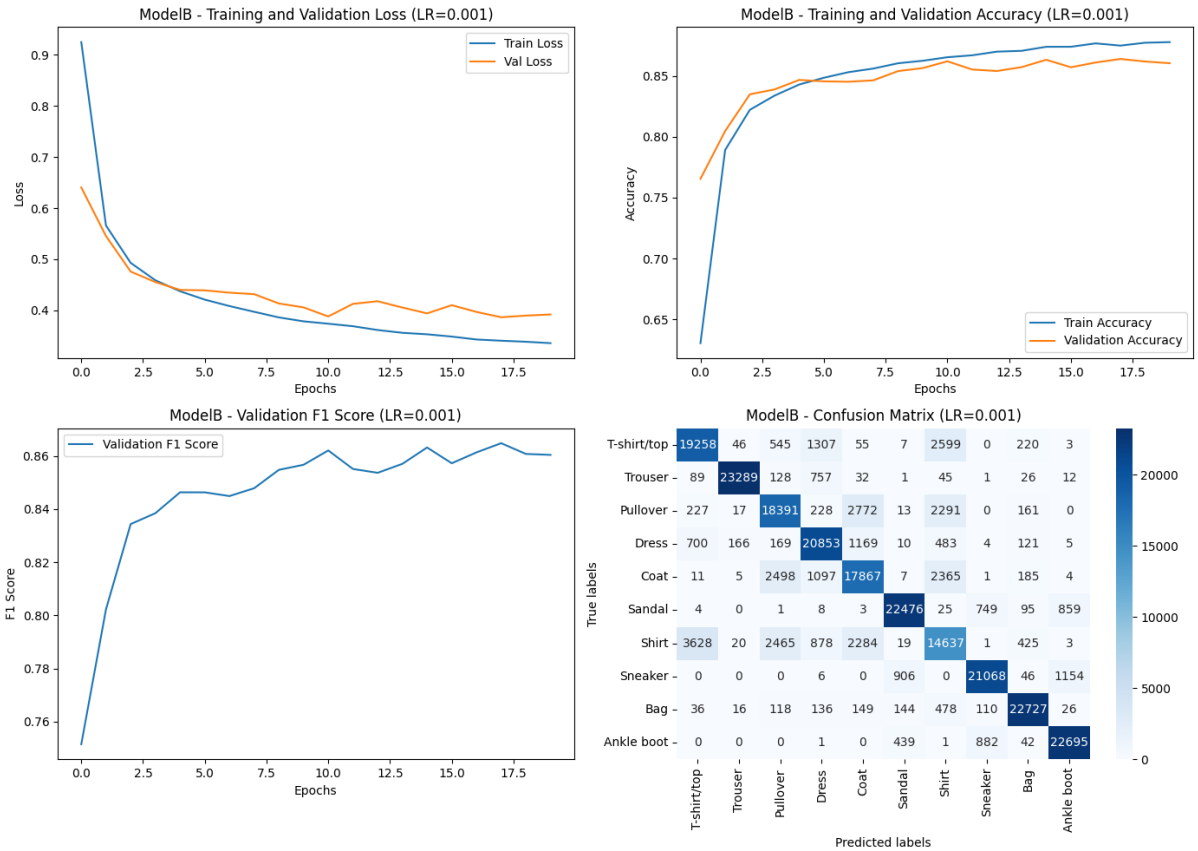

Figure 6: Plots for ModelA and LR=0.001

5.7 ModelB, LR=0.0005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	1.1012	0.5475	0.8270	0.6683	0.6559
2	0.7048	0.7323	0.6155	0.7771	0.7712
3	0.5662	0.7906	0.5424	0.8040	0.8012
4	0.5053	0.8176	0.5094	0.8205	0.8197
5	0.4704	0.8323	0.4790	0.8308	0.8325
6	0.4430	0.8415	0.4606	0.8372	0.8376
7	0.4229	0.8497	0.4456	0.8400	0.8412
8	0.4110	0.8542	0.4266	0.8469	0.8475
9	0.3951	0.8576	0.4214	0.8486	0.8494
10	0.3870	0.8618	0.4169	0.8553	0.8564
11	0.3742	0.8656	0.4097	0.8558	0.8559
12	0.3681	0.8666	0.4016	0.8582	0.8592
13	0.3614	0.8693	0.3994	0.8597	0.8598
14	0.3534	0.8733	0.4003	0.8570	0.8583
15	0.3533	0.8734	0.3984	0.8617	0.8617
16	0.3434	0.8761	0.3898	0.8594	0.8599
17	0.3418	0.8749	0.3957	0.8609	0.8610
18	0.3325	0.8787	0.3832	0.8618	0.8608
19	0.3287	0.8799	0.3869	0.8652	0.8649
20	0.3260	0.8815	0.3908	0.8652	0.8658

Table 7: Training and Validation Metrics for ModelB with Learning Rate = 0.0005

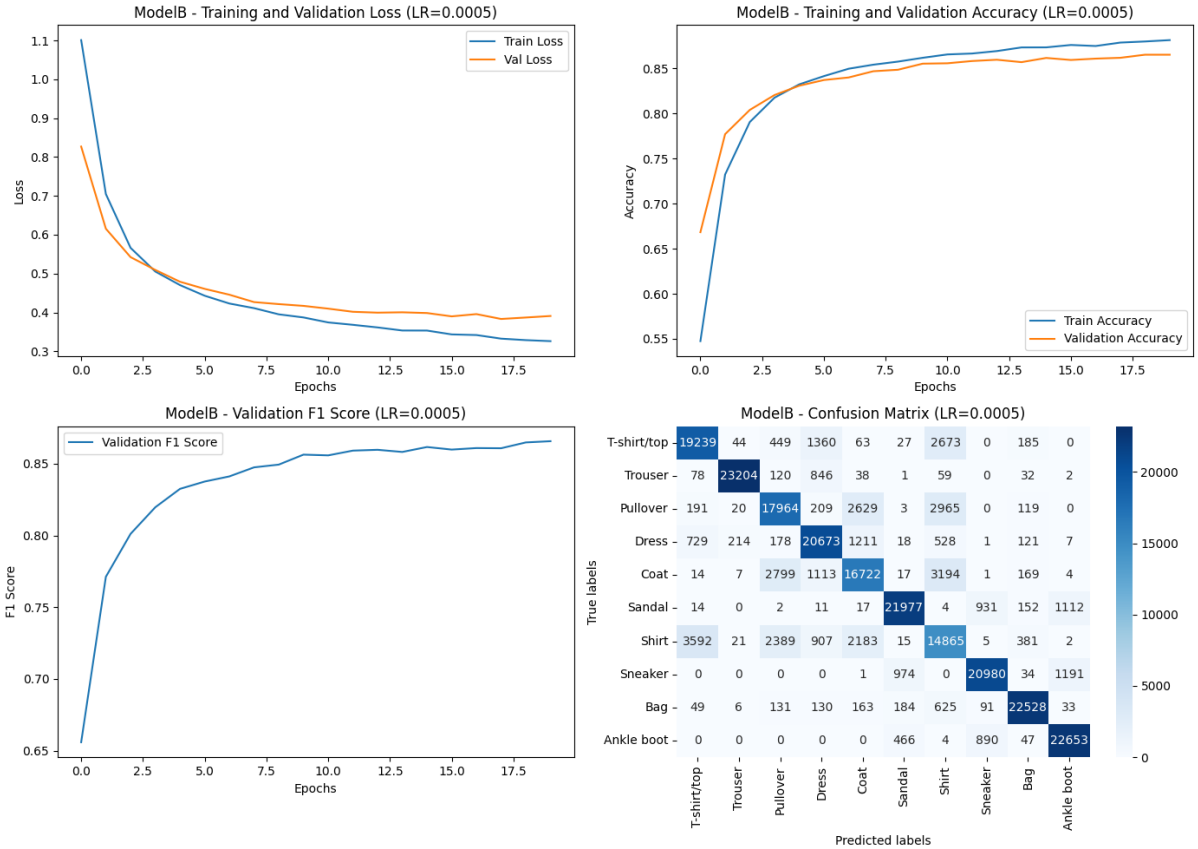

Figure 7: Plots for ModelA and LR=0.0005

5.8 ModelB, LR=0.0001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	1.3804	0.4309	1.0769	0.5603	0.5137
2	0.9794	0.6030	0.9339	0.6285	0.5821
3	0.8884	0.6422	0.8694	0.6603	0.6261
4	0.8316	0.6735	0.8182	0.6955	0.6723
5	0.7770	0.7095	0.7585	0.7244	0.7056
6	0.7174	0.7389	0.7088	0.7448	0.7319
7	0.6749	0.7566	0.6727	0.7583	0.7455
8	0.6438	0.7672	0.6384	0.7691	0.7582
9	0.6164	0.7785	0.6169	0.7757	0.7666
10	0.5928	0.7851	0.6001	0.7843	0.7792
11	0.5715	0.7924	0.5777	0.7903	0.7855
12	0.5511	0.8000	0.5617	0.7972	0.7935
13	0.5312	0.8085	0.5388	0.8071	0.8051
14	0.5160	0.8171	0.5245	0.8196	0.8181
15	0.4992	0.8233	0.5154	0.8187	0.8171
16	0.4852	0.8301	0.4998	0.8231	0.8222
17	0.4764	0.8347	0.4875	0.8316	0.8311
18	0.4636	0.8387	0.4833	0.8274	0.8257
19	0.4556	0.8415	0.4684	0.8375	0.8372
20	0.4445	0.8450	0.4616	0.8373	0.8366

Table 8: Training and Validation Metrics for ModelB with Learning Rate = 0.0001

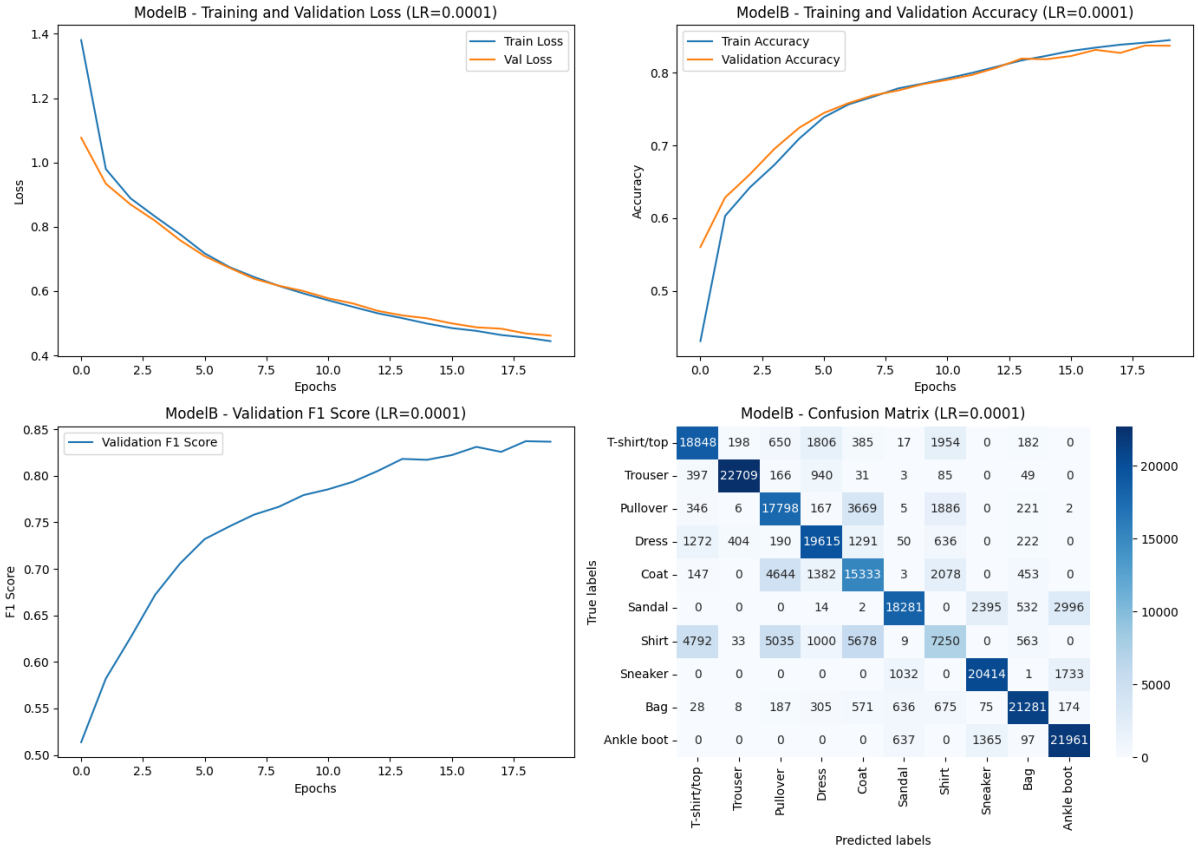

Figure 8: Plots for ModelA and LR=0.0001

5.9 ModelC, LR=0.005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.6570	0.7648	0.5010	0.8194	0.8192
2	0.4954	0.8237	0.4590	0.8346	0.8339
3	0.4587	0.8377	0.4298	0.8432	0.8431
4	0.4351	0.8450	0.4163	0.8502	0.8494
5	0.4169	0.8505	0.4133	0.8523	0.8523
6	0.4068	0.8559	0.4078	0.8546	0.8541
7	0.3952	0.8582	0.4021	0.8563	0.8561
8	0.3806	0.8622	0.3965	0.8568	0.8566
9	0.3753	0.8639	0.3900	0.8598	0.8593
10	0.3674	0.8674	0.3983	0.8586	0.8586
11	0.3653	0.8682	0.3833	0.8632	0.8628
12	0.3578	0.8703	0.3909	0.8627	0.8626
13	0.3488	0.8740	0.3895	0.8594	0.8593
14	0.3494	0.8744	0.3811	0.8632	0.8630
15	0.3407	0.8750	0.3787	0.8682	0.8680
16	0.3358	0.8796	0.3788	0.8643	0.8641
17	0.3348	0.8792	0.3847	0.8651	0.8646
18	0.3271	0.8826	0.3870	0.8603	0.8603
19	0.3229	0.8836	0.3889	0.8648	0.8648
20	0.3185	0.8840	0.3843	0.8639	0.8637

Table 9: Training and Validation Metrics for ModelC with Learning Rate = 0.005

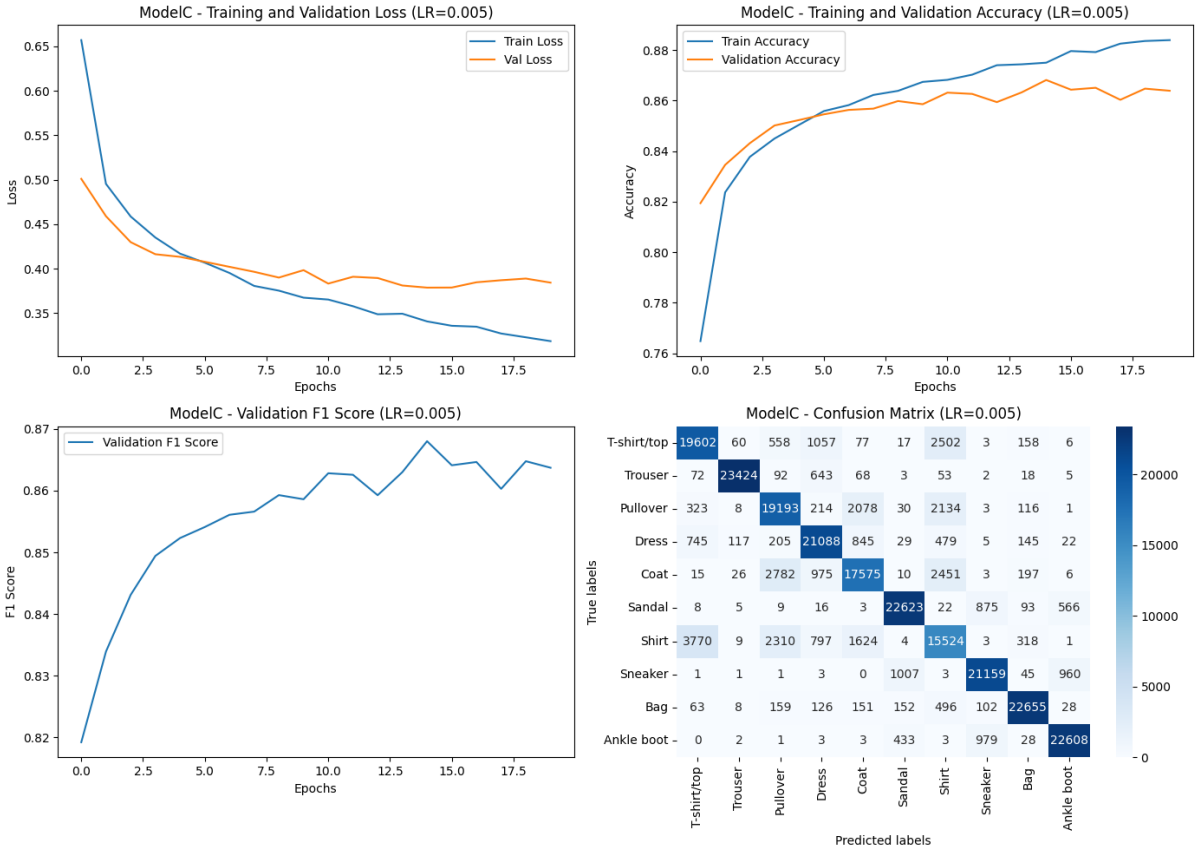

Figure 9: Plots for ModelA and LR=0.005

5.10 ModelC, LR=0.001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.7595	0.7583	0.5081	0.8241	0.8232
2	0.5001	0.8252	0.4546	0.8396	0.8384
3	0.4557	0.8411	0.4335	0.8420	0.8417
4	0.4300	0.8494	0.4200	0.8495	0.8487
5	0.4147	0.8540	0.4087	0.8544	0.8541
6	0.4027	0.8565	0.4010	0.8576	0.8571
7	0.3895	0.8610	0.4033	0.8580	0.8574
8	0.3769	0.8652	0.3956	0.8561	0.8556
9	0.3692	0.8691	0.3943	0.8599	0.8597
10	0.3646	0.8696	0.3906	0.8585	0.8580
11	0.3578	0.8725	0.3834	0.8639	0.8637
12	0.3511	0.8739	0.3878	0.8601	0.8595
13	0.3435	0.8768	0.3838	0.8615	0.8615
14	0.3367	0.8771	0.3781	0.8630	0.8627
15	0.3349	0.8786	0.3733	0.8664	0.8660
16	0.3308	0.8808	0.3753	0.8678	0.8676
17	0.3234	0.8832	0.3767	0.8642	0.8636
18	0.3203	0.8849	0.3724	0.8705	0.8704
19	0.3189	0.8855	0.3728	0.8657	0.8659
20	0.3110	0.8867	0.3711	0.8707	0.8708

Table 10: Training and Validation Metrics for ModelC with Learning Rate = 0.001

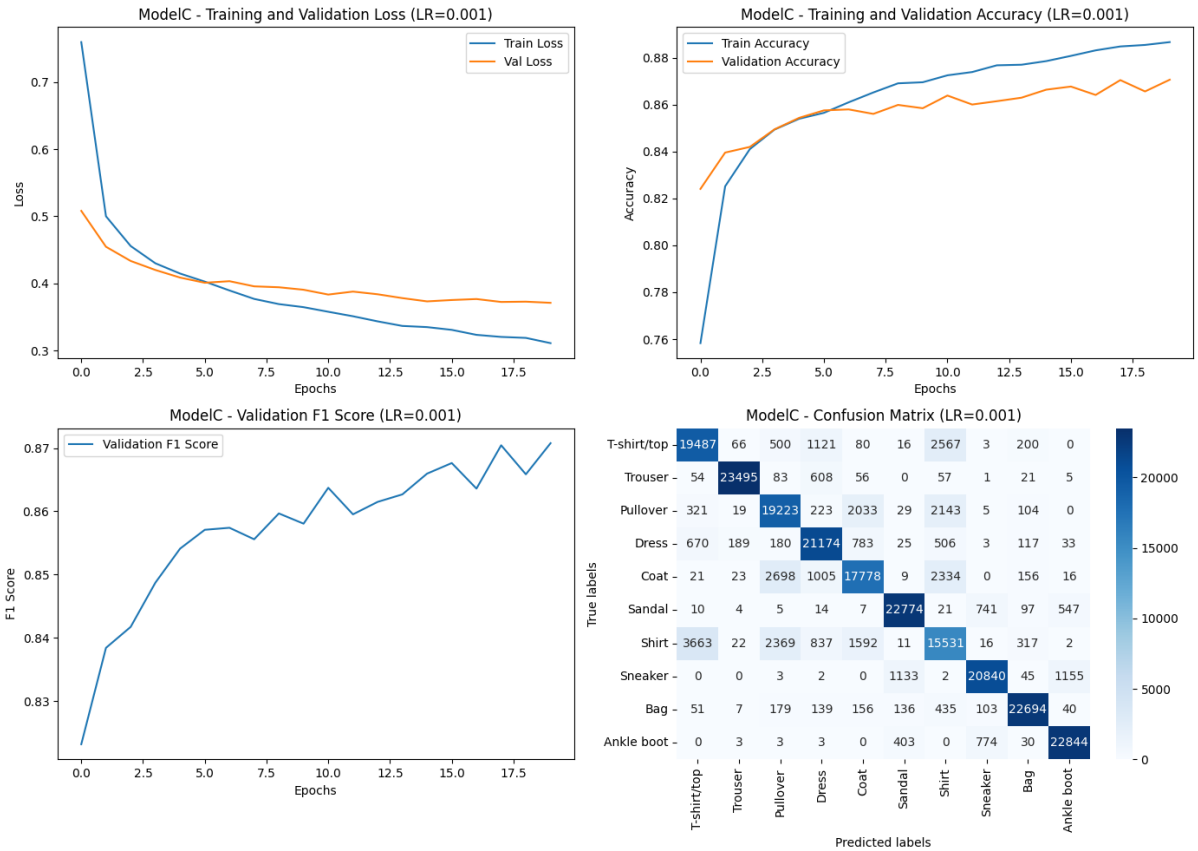

Figure 10: Plots for ModelA and LR=0.001

5.11 ModelC, LR=0.0005

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	0.8630	0.7535	0.5318	0.8257	0.8230
2	0.5100	0.8271	0.4659	0.8341	0.8331
3	0.4561	0.8423	0.4274	0.8479	0.8470
4	0.4320	0.8494	0.4210	0.8498	0.8490
5	0.4123	0.8549	0.4082	0.8555	0.8553
6	0.3980	0.8585	0.4042	0.8543	0.8537
7	0.3885	0.8631	0.4019	0.8588	0.8582
8	0.3755	0.8667	0.4011	0.8585	0.8577
9	0.3697	0.8695	0.3907	0.8596	0.8591
10	0.3605	0.8728	0.3880	0.8614	0.8611
11	0.3572	0.8728	0.3809	0.8631	0.8629
12	0.3482	0.8772	0.3908	0.8619	0.8618
13	0.3429	0.8781	0.3823	0.8644	0.8644
14	0.3377	0.8788	0.3770	0.8686	0.8682
15	0.3315	0.8813	0.3815	0.8636	0.8635
16	0.3255	0.8829	0.3768	0.8641	0.8638
17	0.3249	0.8839	0.3861	0.8633	0.8631
18	0.3224	0.8836	0.3763	0.8697	0.8693
19	0.3124	0.8874	0.3733	0.8698	0.8696
20	0.3114	0.8892	0.3810	0.8691	0.8695

Table 11: Training and Validation Metrics for ModelC with Learning Rate = 0.0005

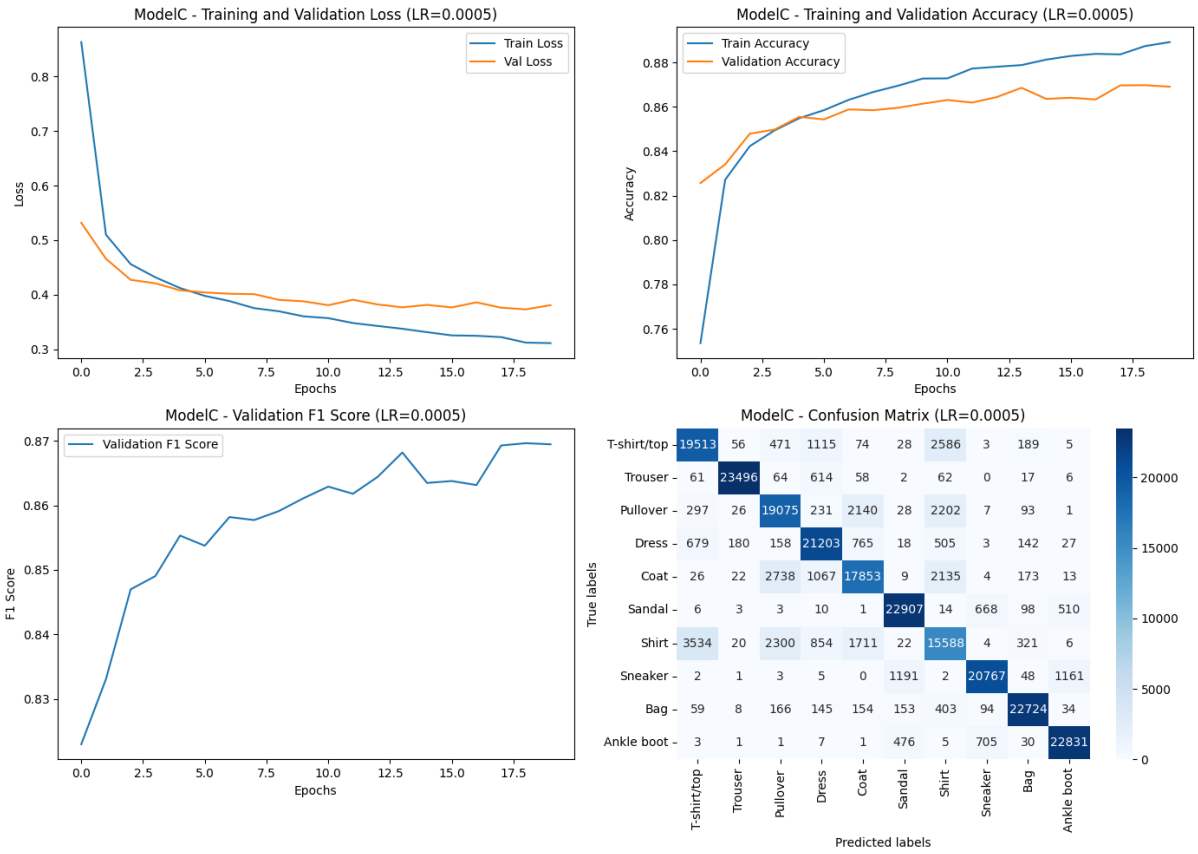

Figure 11: Plots for ModelA and LR=0.0005

5.12 ModelC, LR=0.0001

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Val F1
1	1.4579	0.6916	0.9736	0.7593	0.7350
2	0.8277	0.7697	0.6899	0.8033	0.7949
3	0.6320	0.8161	0.5548	0.8334	0.8303
4	0.5309	0.8370	0.4961	0.8403	0.8385
5	0.4801	0.8459	0.4608	0.8489	0.8476
6	0.4476	0.8538	0.4434	0.8525	0.8515
7	0.4248	0.8581	0.4206	0.8542	0.8533
8	0.4085	0.8630	0.4186	0.8559	0.8556
9	0.3937	0.8659	0.4066	0.8564	0.8561
10	0.3859	0.8676	0.3999	0.8603	0.8594
11	0.3757	0.8706	0.3989	0.8609	0.8602
12	0.3687	0.8705	0.3983	0.8625	0.8626
13	0.3632	0.8741	0.3915	0.8645	0.8639
14	0.3521	0.8771	0.3846	0.8628	0.8623
15	0.3500	0.8770	0.3740	0.8682	0.8674
16	0.3444	0.8800	0.3825	0.8655	0.8649
17	0.3411	0.8826	0.3776	0.8660	0.8659
18	0.3354	0.8822	0.3839	0.8636	0.8631
19	0.3316	0.8821	0.3736	0.8686	0.8684
20	0.3292	0.8852	0.3770	0.8661	0.8660

Table 12: Training and Validation Metrics for ModelC with Learning Rate = 0.0001

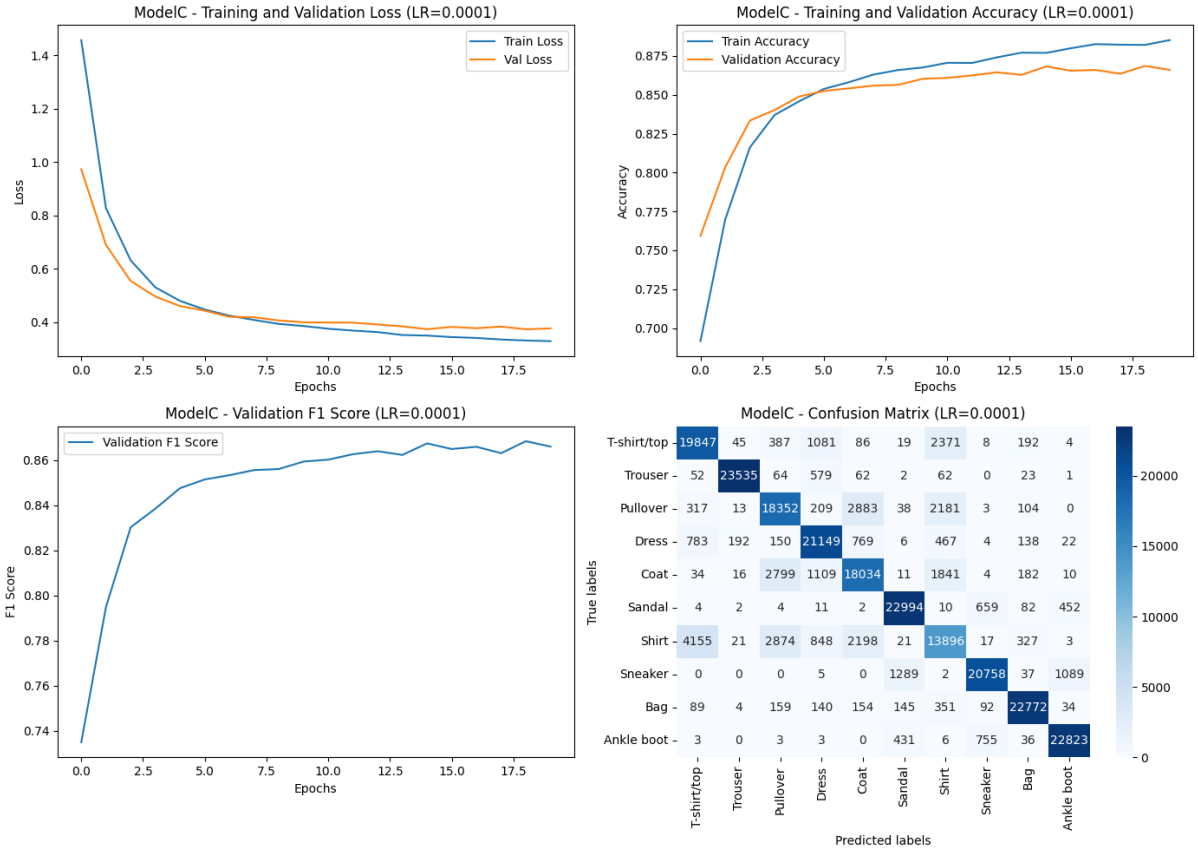

Figure 12: Plots for ModelA and LR=0.0001

6 Test Results of Best Model

Best Scored Model is ModelA with LR = 0.001

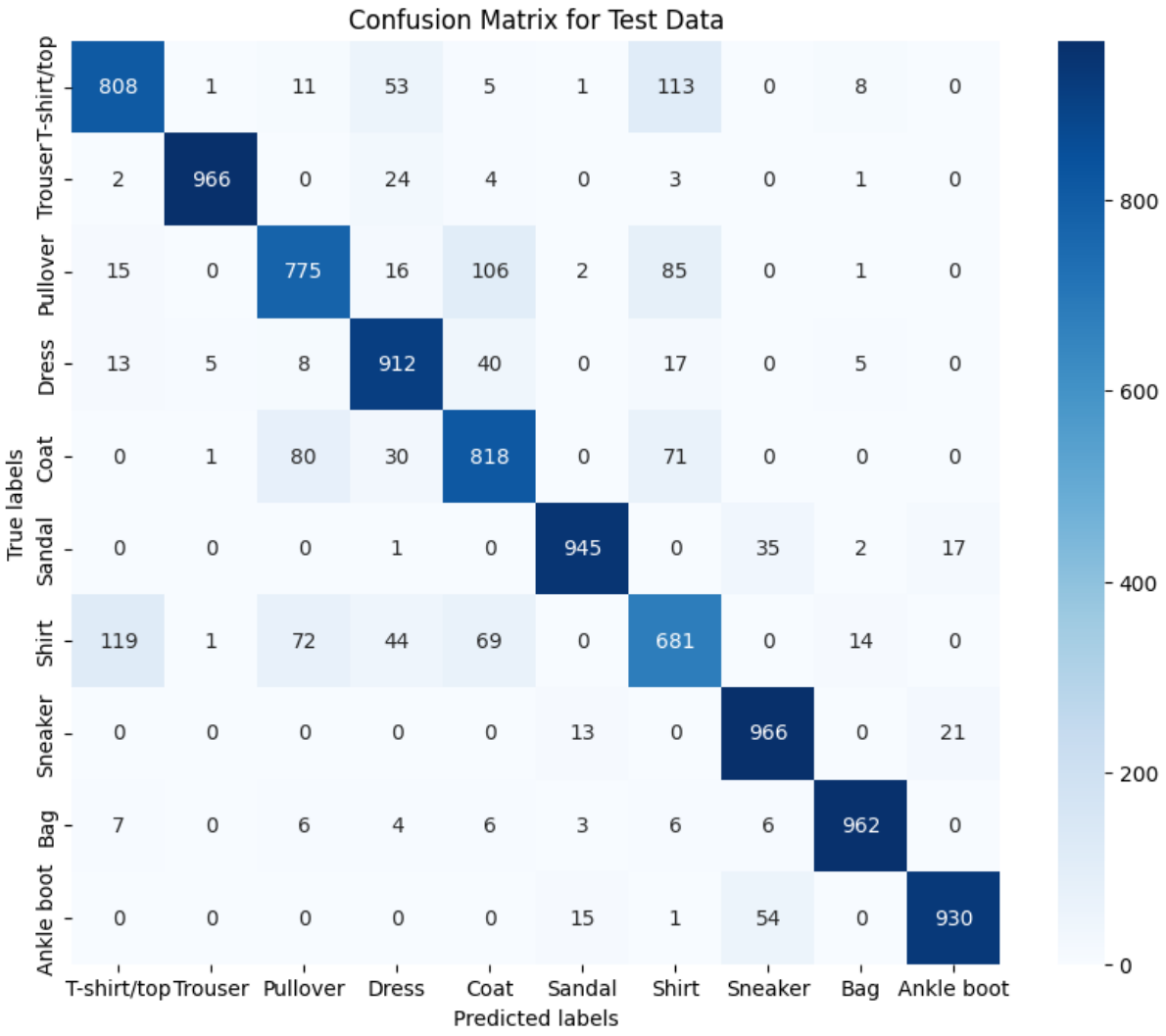

Figure 13: Plots for best model: ModelA and LR=0.001