

# Assignment #1: Optimizing Neural Networks

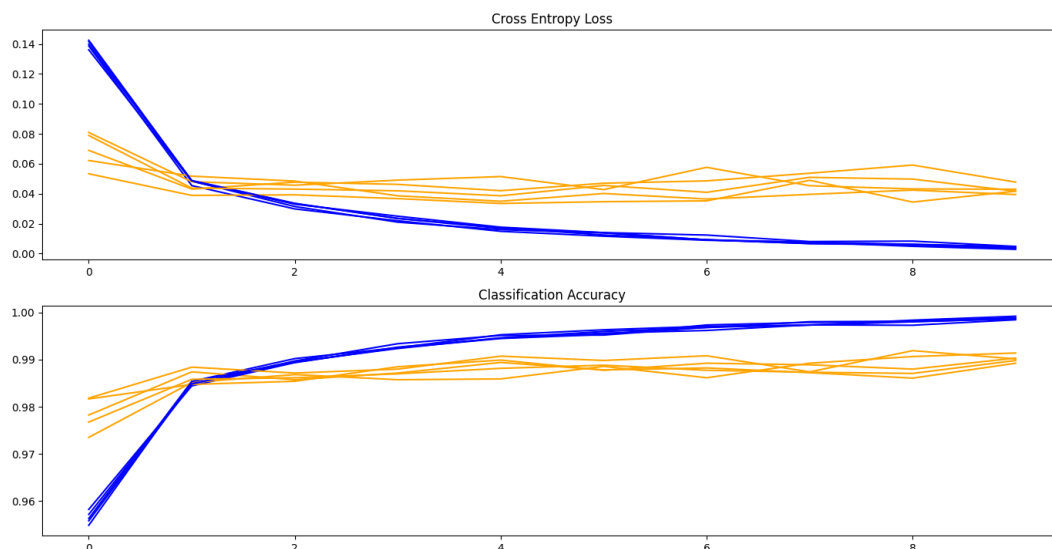
## 1. Introduction and Baseline Model

The goal was to systematically modify the hyperparameters of a baseline CNN model and analyze the impact on performance, generalization, and stability.

The **Baseline Model (Run 1)** used a 3-layer CNN structure (32, 64, and 64 filters) with a 100-neuron Dense layer, trained with SGD (LR = 0.01) for 10 epochs.

Configuration	Mean Accuracy	Std Deviation
Run 1: Baseline	99.020%	0.071

The initial learning curve shows the model still improving slightly at 10 epochs, suggesting potential for improvement by increasing the epoch count.



---

## 2. Impact of Network Structure (Layers and Neurons)

### A. Reduced Complexity (Run 2: Shallow)

The network was reduced to 2 Conv layers and 60 Dense neurons.

Configuration	Layers/Neurons Used	Mean Accuracy	Std Deviation
Run 2: Shallow	2 Conv (32, 64), 1 Dense (60)	98.883%	0.176

**Impact and Justification:** The significant increase in standard deviation ( $\uparrow 0.105$ ) and drop in accuracy confirmed that the simplified structure suffered from **underfitting**, lacking the necessary **model capacity** to fully generalize from the training data.

## B. Increased Complexity (Run 3: Deep)

The complexity was increased by adding a fourth `Conv2D` layer with 128 filters.

Configuration	Layers/Neurons Used	Mean Accuracy	Std Deviation
Run 3: Deep	4 Conv, 1 Dense (100)	98.973%	0.168

Did you notice any degradation of the results with increasing the complexity of the network?

Yes, the model experienced slight performance degradation ( $\downarrow 0.047\%$ ) and reduced stability ( $\uparrow 0.097$  in std) compared to the baseline.

**Justification:** This degradation is due to **over-capacity** and **diminishing returns**; the added complexity was not necessary for the problem and introduced greater instability.

---

## 3. Impact of Learning Rate (LR)

### A. Low Learning Rate (Run 4: $LR = 0.001$ )

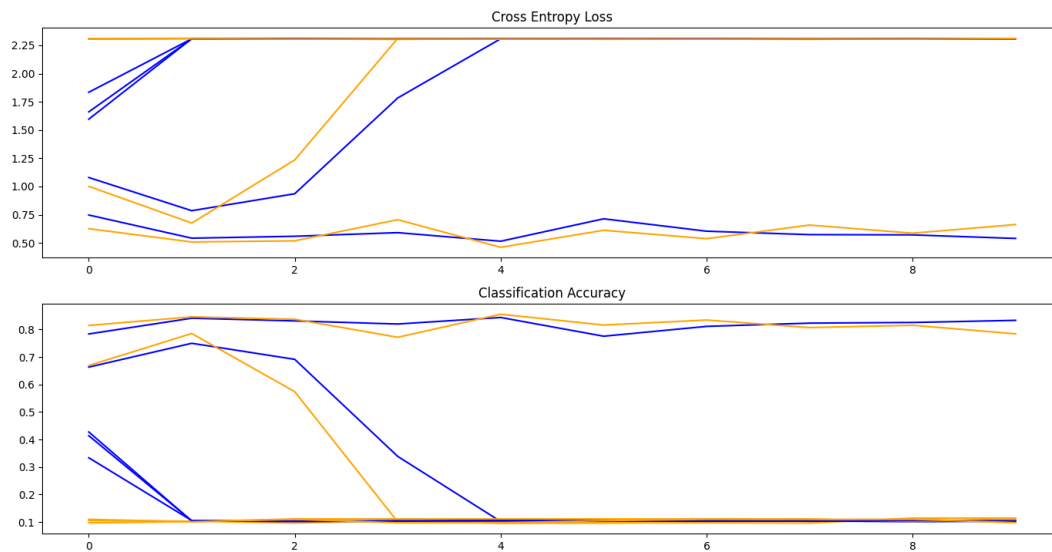
Configuration	Learning Rate	Mean Accuracy
Run 4: Low LR	0.001	98.607%

**Impact:** The model performed poorly because the small steps led to **under-training** and failure to converge within 10 epochs.

### B. High Learning Rate (Run 5: $LR = 0.1$ )

Configuration	Learning Rate	Mean Accuracy	Std Deviation
Run 5: High LR	0.1	24.112%	27.169

**Impact:** The model exhibited **catastrophic divergence** and extreme instability, confirming that the large steps caused the optimizer to overshoot the minimum loss, failing to learn.



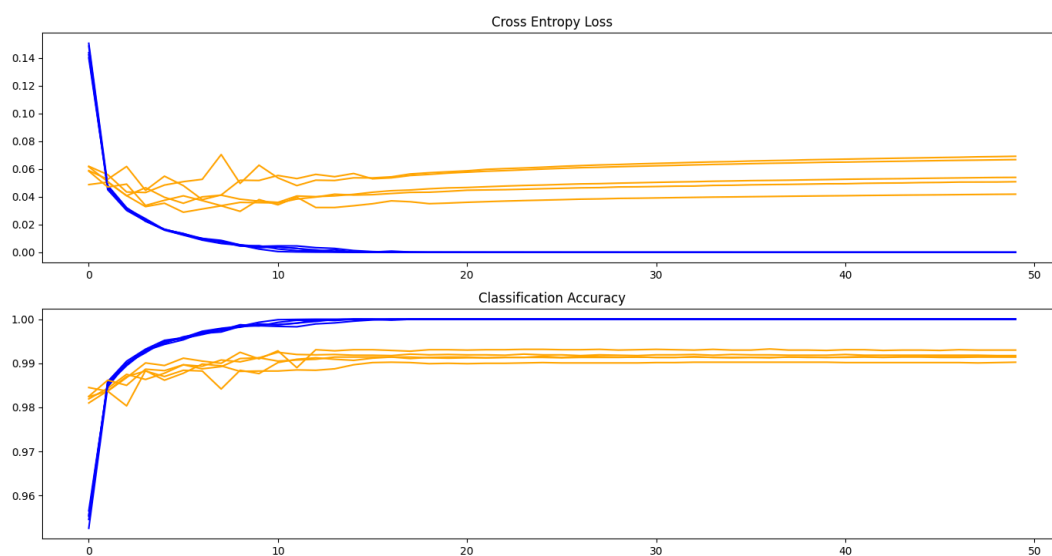
## 4. Impact of Training Epochs

### A. Increased Epochs (Run 6: 50 Epochs)

Configuration	Epochs	Mean Accuracy
Run 6: Long	50	99.158%

#### When do you get overfitting?

Overfitting is clearly visible in the loss plot for the 50-epoch run. The Validation Loss (orange lines) begins to plateau and subsequently rise after approximately Epoch 10 to 15, while the Training Loss (blue lines) continues to decrease toward zero.

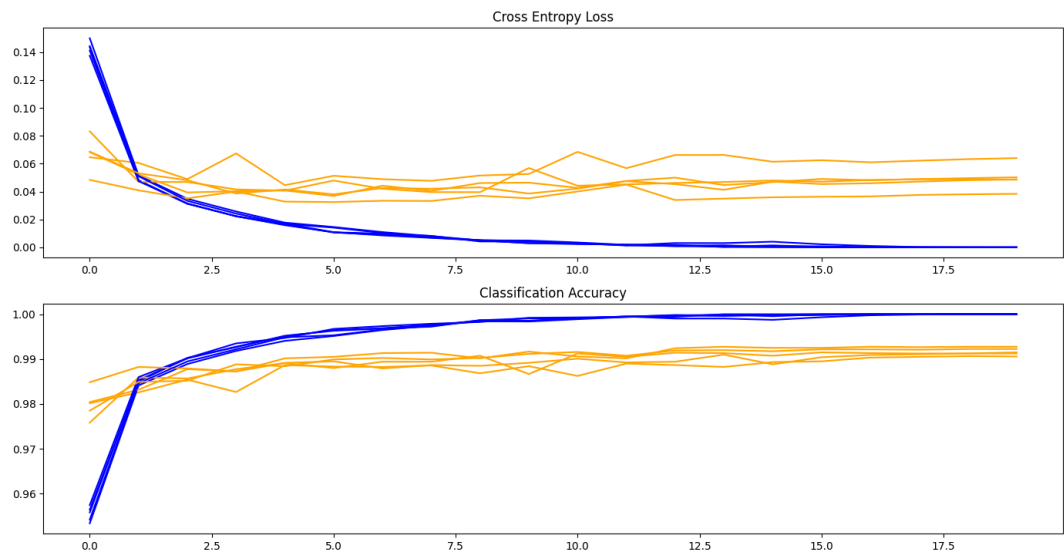


## B. Optimal Epochs (Run 7: 20 Epochs)

The final optimal run was set to 20 epochs, balancing convergence and overfitting risk.

Configuration	Epochs	Mean Accuracy	Std Deviation
Run 7: Optimal	20	99.167%	0.076

This run achieved the best combination of high accuracy and stability.



---

## 5. Optimum Parameters and Conclusion

**Are there any optimum parameters?**

Yes. The final optimum parameters, which delivered the highest stable performance ( 99.167% mean accuracy), were determined to be:

Parameter Category	Optimum Value
Network Structure	Baseline (3 Conv layers, 100 Dense neurons)
Learning Rate	0.01 (SGD Optimizer)
Training Epochs	20 epochs

The investigation demonstrated that increasing the training epochs from the initial 10 to 20 was the most effective adjustment, showing that **more epochs** did lead to **better performance** up to the point where overfitting began around 15 epochs.

## Report by Mustafa Emir Kaynar - 222010020068

<b>Github Repository</b>	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1</a>
--------------------------	---

<b>Screenshots &amp; Outputs</b>	<b>Link</b>
Run 1 (Baseline)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_1_baseline_config">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_1_baseline_config</a>
Run 2 (Shallow)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_2_shallow_config">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_2_shallow_config</a>
Run 3 (Deep)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_3_deep_config">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_3_deep_config</a>
Run 4 (Low LR)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_4_low_learning_rate">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_4_low_learning_rate</a>
Run 5 (High LR)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_5_high_learning_rate">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_5_high_learning_rate</a>
Run 6 (Increased Epochs)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_6_increased_epochs">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_6_increased_epochs</a>
Run 7 (Optimal)	<a href="https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_7_best_config">https://github.com/theEMA-dev/KFAU-Deep-Learning/tree/main/assignment-1/results/run_7_best_config</a>