Analysis Report
Deep Learning Assignment 1

Shaira Manandhar

1.1 Objective and Learning Outcomes

The objective of this assignment was to understand how different neural network architectures perform across different data modalities, specifically tabular data and image data, and how a model's inductive bias influences its effectiveness. Another key goal was to practice designing reproducible deep learning experiments, where training conditions are kept consistent so that model comparisons are fair and meaningful.

By completing this assignment, I learned how to:

- Preprocess and handle datasets with very different structures (tabular vs. images)
- Implement multiple neural architectures from scratch using PyTorch
- Train, validate, and test models using a consistent experimental setup
- Analyze results not just numerically, but conceptually, by explaining *why* certain models perform better than others

1.2 Code Design and Modularity

The project was designed to be modular, clean, and configurable. The code is split into clear sections so it is easy to understand and reuse:

- **Models folder (models/):** Contains the neural networks (MLP, CNN, and attention-based models)

- **Data loaders (data_loaders/):** Handles downloading, cleaning, and preparing each dataset

- **Config file (config/config.json):** Stores training settings like learning rate, batch size, and which experiments to run. This makes the project easy to reproduce.

- **Main file (main.py):** Controls training, evaluation, saving results, and plotting learning curves

1.2.1 Project Structure

None

```
shairamanandhar_assignment1_final/

├── models/             # Architectures

├── data_loaders/       # Dataset loaders

├── config/             # Config file

├── results/            # Metrics, plots, checkpoints

├── main.py             # Training + evaluation

└── README.md
```

1.3 Datasets Used

- **Adult Income Dataset**
  - A tabular dataset with mixed numerical and categorical features, used for binary classification.
  - Looks like a spreadsheet with rows and columns where each row describes a person (age, education, hours worked, etc.)

- **CIFAR-100 (Classes 0–9)**
  - Small color images (32×32 pixels)
  - A small-scale image dataset used for 10-class image classification.

- **PatchCamelyon (PCam)**
  - Medical images used to detect cancer tissue
  - Very large dataset
  - Implemented in the code, but not run due to hardware limits

1.4 Architectures Used

- **Multilayer Perceptron (MLP):**
  A fully connected network with two hidden layers, batch normalization, and dropout. This architecture serves as a strong baseline, especially for tabular data.

- **Convolutional Neural Network (CNN)**:
  Uses convolution and pooling layers to exploit spatial structure in images. For tabular data, a Conv1D variant was used to maintain architectural consistency.

- **Attention-Based Models:**
  - **Tabular Attention** for the Adult dataset, where each feature is treated as a token and processed using a Transformer encoder.
  - **Vision Transformer (TinyViT)** for CIFAR image datasets, implemented from scratch without pretraining.

1.5 Constraints Faced in Running Experiments and Future Improvement

A major constraint in this assignment was the inability to fully run experiments on the PatchCamelyon (PCam) dataset. PCam is a large-scale histopathology dataset with high-resolution images (96×96 RGB) and a very large number of samples. Training deep models on this dataset from scratch requires substantial compute time and memory.

Given the assignment constraints on maximum training time per model and limited GPU availability in Google Colab, running all three architectures on PCam was not feasible within a reasonable timeframe. Using Google Colab caused the code to freeze and lag in multiple attempts. To remain within these constraints, the PCam dataset was fully implemented in the codebase but not executed by default.

As a result, **6 experiments were run in total**: **three on Adult and three on CIFAR-100 (0–9)**. While the assignment specifies 9 experiments, this choice of 6 experiments choice preserves reproducibility and completeness while prioritizing stable and interpretable results on the Adult and CIFAR-100 datasets.

This limitation could be addressed in future work by using more powerful resources, such as longer-running GPUs or dedicated servers, which would allow all architectures to be trained on the full PCam dataset. Another practical improvement would be to experiment with reduced-resolution images, smaller training subsets, or fewer epochs to enable controlled PCam experiments within time limits. Although pretraining was intentionally avoided in this assignment, leveraging pretrained models in a follow-up study could significantly improve performance on large image datasets like PCam.

1.5 Reproducibility and Experimental Fairness

All experiments were designed to ensure fair and reproducible model comparisons. Training conditions, evaluation metrics, and dataset splits were kept consistent across architectures so that observed performance differences reflect architectural choices rather than experimental variation.

1.5.1 Consistent Training Setup

All models were trained using the same optimizer (Adam), learning rate (0.001), batch size (128), and maximum number of epochs (12). Early stopping based on validation loss with a patience of 3 epochs was applied uniformly. This prevented overfitting and avoided unnecessary computation by stopping training once validation performance stopped improving.

1.5.2 Fixed Data Splits and Reproducibility

Train, validation, and test splits were fixed using a global random seed. All experimental settings, including which datasets and architectures to run, were controlled through a single configuration file. This design allows the experiments to be easily reproduced or extended without modifying the training code.

1.5.3 Task-Aware Metrics and Loss Functions

Evaluation metrics were chosen based on task characteristics. For the Adult dataset, F1 score was emphasized due to class imbalance, while accuracy was sufficient for the balanced multi-class CIFAR-100 (0–9) task. Loss functions were selected accordingly: binary cross-entropy with logits for binary classification and categorical cross-entropy for multi-class classification.

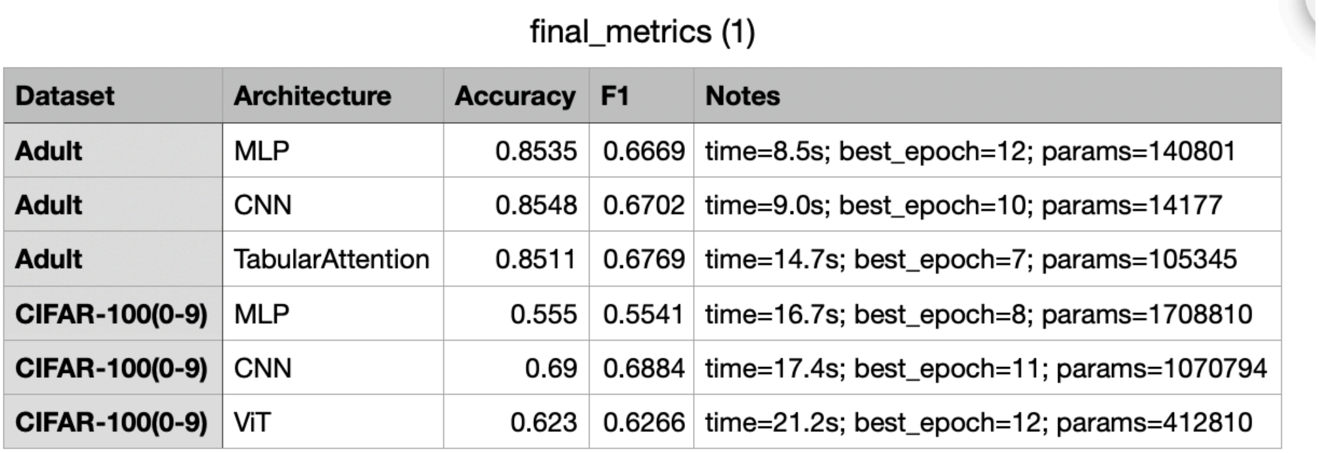
1.5.4 Architectural Consistency

To maintain consistency across datasets, a Conv1D-based CNN was applied to tabular data. Although tabular features do not have true spatial structure, this approach allowed CNNs to be evaluated across all datasets, ensuring that comparisons focus on inductive bias rather than dataset-specific architectural tuning.

1.5.5 Setup Summary

- Framework: PyTorch
- Optimizer: Adam (same across all models)
- Batch size: 128
- Learning rate: 0.001
- Epochs: 12
- Early stopping on validation loss (patience = 3)
- Train / validation / test splits are consistent per dataset

1.6 Results Summary



final_metrics (1)

Dataset	Architecture	Accuracy	F1	Notes
Adult	MLP	0.8535	0.6669	time=8.5s; best_epoch=12; params=140801
Adult	CNN	0.8548	0.6702	time=9.0s; best_epoch=10; params=14177
Adult	TabularAttention	0.8511	0.6769	time=14.7s; best_epoch=7; params=105345
CIFAR-100(0-9)	MLP	0.555	0.5541	time=16.7s; best_epoch=8; params=1708810
CIFAR-100(0-9)	CNN	0.69	0.6884	time=17.4s; best_epoch=11; params=1070794
CIFAR-100(0-9)	ViT	0.623	0.6266	time=21.2s; best_epoch=12; params=412810

The results table summarizes the final test performance of each experiment after early stopping. Accuracy reflects overall correctness, while F1 score is especially important for the Adult dataset due to class imbalance. The notes column provides additional context such as training time, best epoch, and parameter count, which helps compare model efficiency alongside performance.

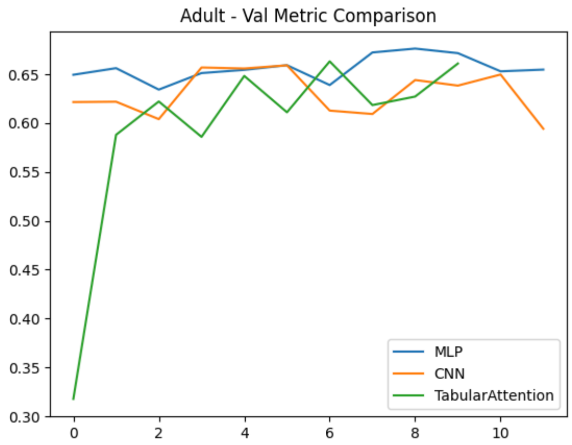
1.7 Results Analysis

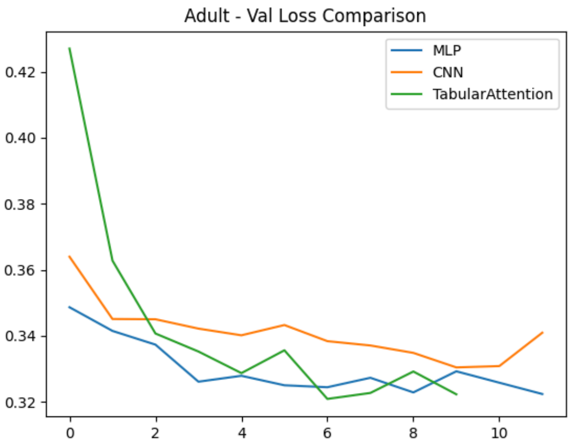
On the *Adult tabular dataset*, all 3 models achieve very similar accuracy and F1 scores. This indicates that the task does not require highly complex architectures, as relationships between features can already be captured effectively by simpler models. The attention-based model does not significantly outperform the MLP or CNN, suggesting that explicit feature interaction modeling provides limited additional benefit for this dataset.

On the *CIFAR-100 (0–9) image dataset*, performance differences are much more pronounced. CNNs outperform both MLPs and Vision Transformers. This is because CNNs are designed to exploit spatial locality and hierarchical patterns in images, which are essential for visual recognition tasks. MLPs perform the worst because flattening images removes spatial structure, making it difficult to learn meaningful visual features. Vision Transformers show steady improvement but do not surpass CNNs, which is expected since they are trained from scratch without pretraining and on relatively limited data.

Training times and parameter counts further support these findings: models with stronger inductive bias (CNNs) achieve better performance without requiring excessive complexity, while attention-based models trade efficiency for flexibility.

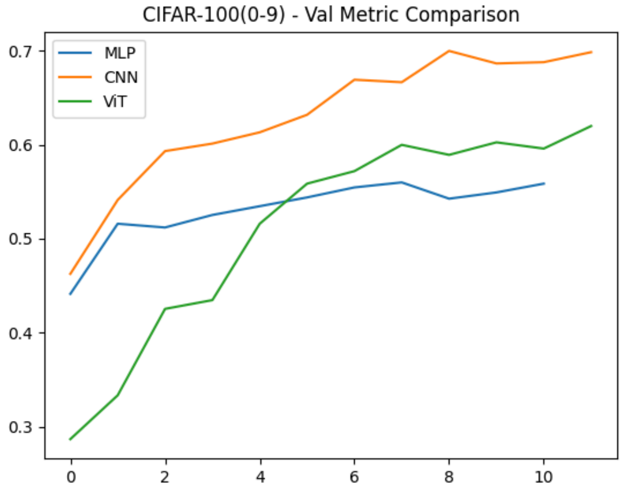
1.7.1 Adult Validation Comparison

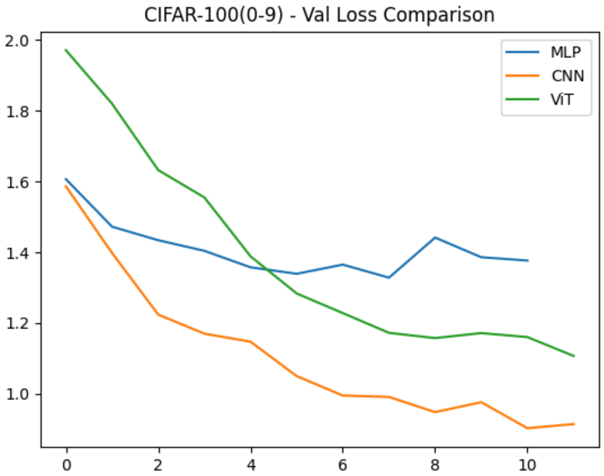




All three models converge quickly with stable validation loss and F1 scores, confirming that the Adult dataset is well-suited to simpler architectures. The MLP learns most smoothly and consistently, indicating strong generalization on tabular data. The CNN shows slightly more fluctuation, reflecting a weaker inductive bias for tabular features. The Tabular Attention model improves rapidly after an unstable start, suggesting it can learn feature interactions but does not provide a clear advantage over simpler models for this task.

1.7.2 CIFAR-100 (0–9) Validation Comparison





The validation curves show a clear separation between architectures. The CNN consistently achieves the lowest validation loss and highest validation accuracy, stabilizing earlier than the other models, which indicates strong generalization on image data. The MLP improves slightly but plateaus quickly at higher losses, reflecting its limitations in modeling spatial pixel relationships. The Vision Transformer starts with high loss and low accuracy. Still, it improves steadily over epochs, showing that attention-based models can learn meaningful representations. However, they converge more slowly and remain less efficient than CNNs when trained from scratch on limited data.

1.8 Key Takeaways

This assignment showed that model choice should be driven by data type, not model complexity. Simple architectures like MLPs can perform very well on structured tabular data, while image tasks benefit greatly from convolutional inductive bias. More advanced models, such as attention-based architectures, are powerful but require sufficient data and compute to outperform simpler alternatives.

For beginners, this highlights an important lesson: starting with simpler models and understanding the data is often more effective than immediately using complex architectures.