أكاديمية طويق
Tuwaiq Academy

Week #6

Day #30

Date: 16 – 10 – 2025

Team Members:

Wajd Alrabiah

Lama Almoutiry

Fatima Alsubaie

Supervisor: Hani Alshafi

Adult Census Income Dataset

About Dataset

" This data was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)). *The prediction task is to determine whether a person makes over $50K a year*. "

Source: Kaggle available on: Link

Task Type: Binary classification – predict whether an individual's annual income is >50K or <=50K.

Number of Records: 32,561 records.

Number of Features: 14 demographic, social, and economic attributes

Target Variable: income (binary: <=50K or >50K).

Feature	Type
age	Numeric
workclass	Categorical
fnlwgt	Numeric
education	Categorical
education.num	Numeric
marital.status	Categorical
occupation	Categorical
relationship	Categorical
race	Categorical
sex	Categorical
capital.gain	Numeric
capital.loss	Numeric
hours.per.week	Numeric
native.country	Categorical

Data Cleaning and Preprocessing

- The dataset originally contained **missing values represented as (?)** in several columns.
- We **dropped all rows containing these missing values** to ensure clean data for training our neural network models.
- Target variable (income) encoded as 0/1.
- Dataset split: **70% train, 15% validation, 15% test**.
- Numerical features scaled using StandardScaler.

Neural Network Architectures

Model	Layers & Activation	Regularization / Dropout
Model 1	Input → Dense(64, ReLU) → Dense(32, ReLU) → Dense(1, Sigmoid)	None
Model 2	Input → Dense(128, ReLU) → Dense(64, ReLU) → Dense(32, ReLU) → Dense(1, Sigmoid)	None
Model 3	Input → Dense(256, ReLU, L2) → BatchNorm → Dropout(0.3) → Dense(128, ReLU, L2) → BatchNorm → Dropout(0.3) → Dense(64, ReLU) → Dropout(0.2) → Dense(1, Sigmoid)	L2 + Dropout + BatchNorm

Callbacks Used

- **EarlyStopping:** Patience = 10, restored best weights to prevent overfitting.
- **ModelCheckpoint:** Saved best weights only (model{num}_best_model.weights.h5).

Training & Optimization

- Optimizer: **Adam**, learning rate = 0.001(not changed across models)
- Loss: **Binary Crossentropy** (not changed across models)
- Metrics: **Accuracy**
- Batch size: 32
- Epochs: 50–100 depending on the model, with EarlyStopping applied.

Evaluation

Model	Test Accuracy	Loss
Model 1	**0.8451**	**0.3437**
Model 2	**0.8334**	**0.3501**
Model 3	**0.8343**	**0.3798**

Insights

- The dataset is **class-imbalanced** → high accuracy alone may not reflect real performance.
- Cleaning missing values (?) and one-hot encoding categorical features improved model training.
- Scaling numerical features ensured faster convergence.
- Simpler models (Model 1) are surprisingly strong baselines.
- Deeper models need **regularization** to avoid overfitting.
- **TensorBoard** and callbacks (EarlyStopping, ModelCheckpoint) are essential for monitoring training and selecting the best model.
- Proper preprocessing (cleaning, encoding, scaling) is critical for neural network performance on structured tabular data.

Visualization

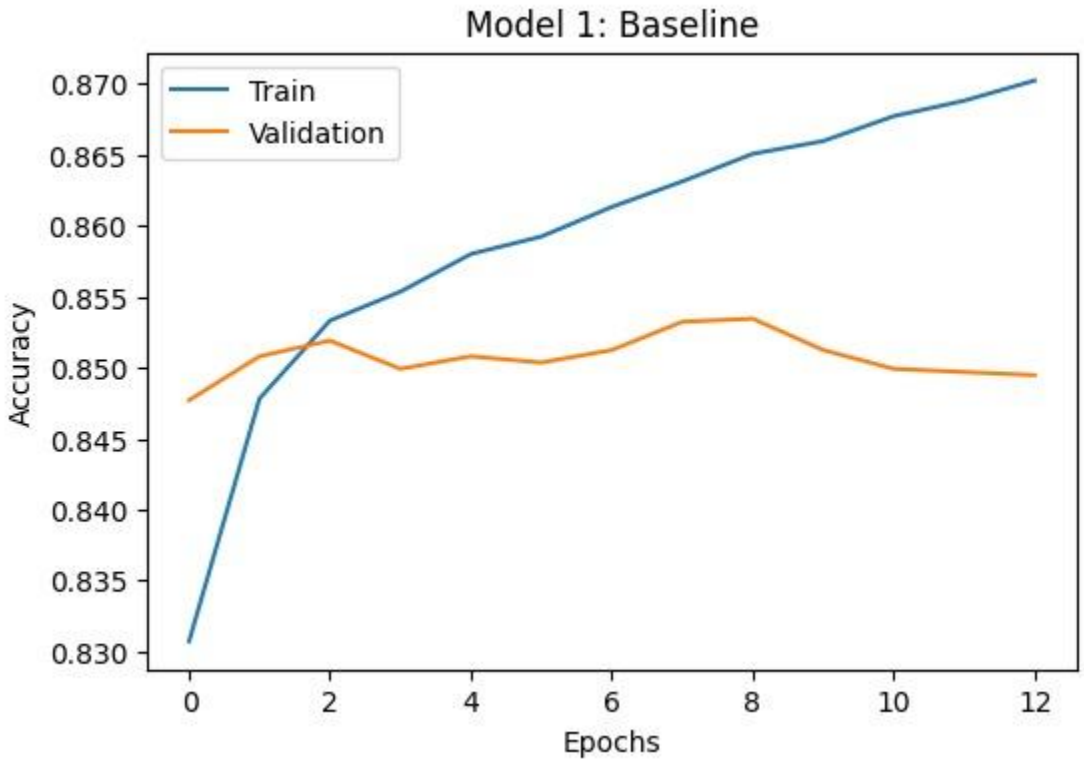

Figure 1 First Model Eval

Overfitting is evident after 2–3 epochs; the model learns details on the train but does not improve accuracy.

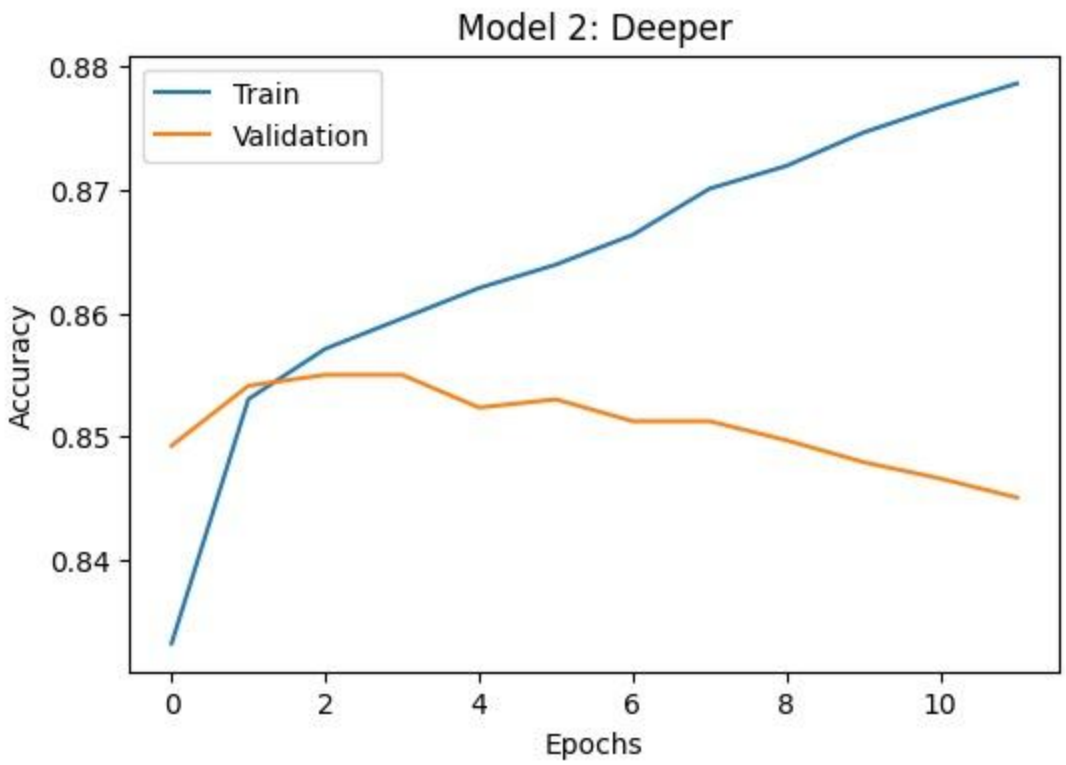

Figure 2 Second Model Eval

The deep model is better able to learn from the train data, but is more susceptible to overfitting without regularization.

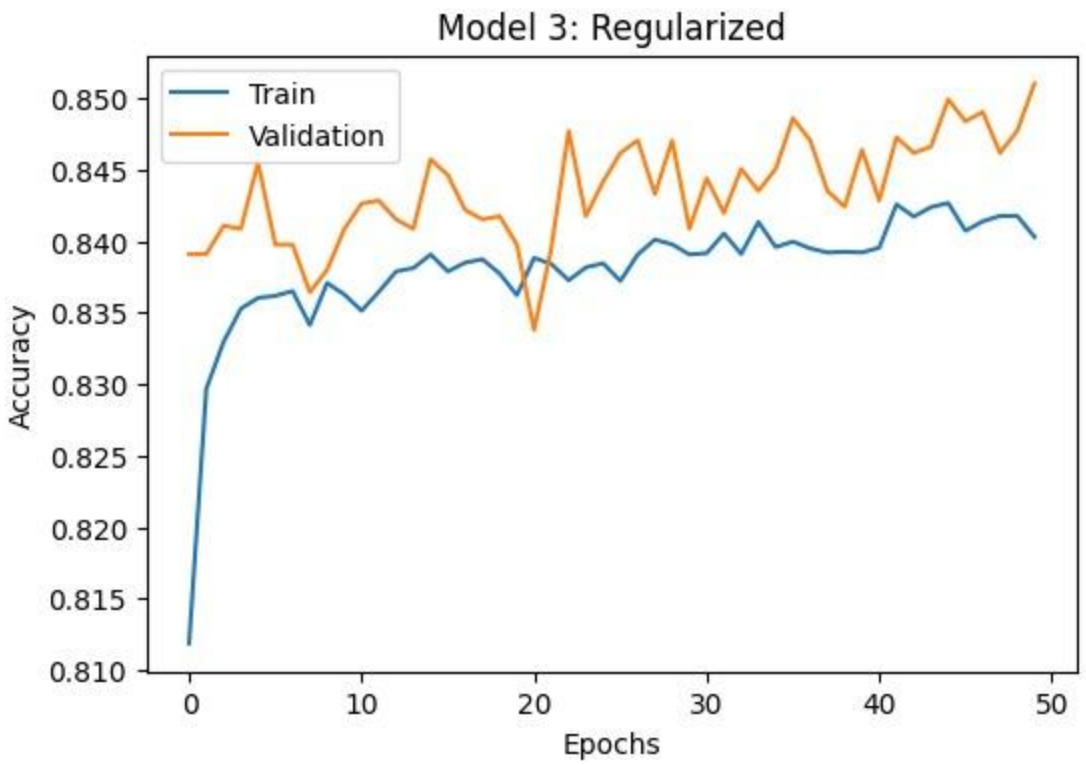

Figure 3 Third Model Eval

I think that regularization improved the overfit problem in previous models, even though the model did not use early stops, but we want to improve the model to break the oscillation in the future.

Epoch Accuracy

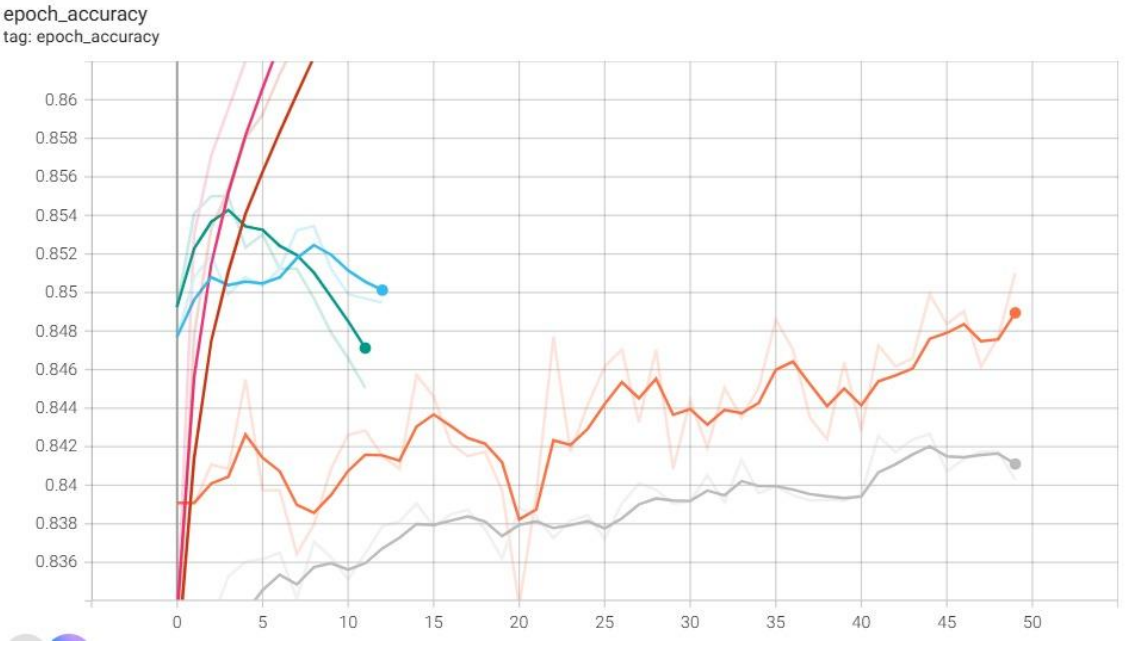

Figure 4: epoch accuracy.

Epoch Loss

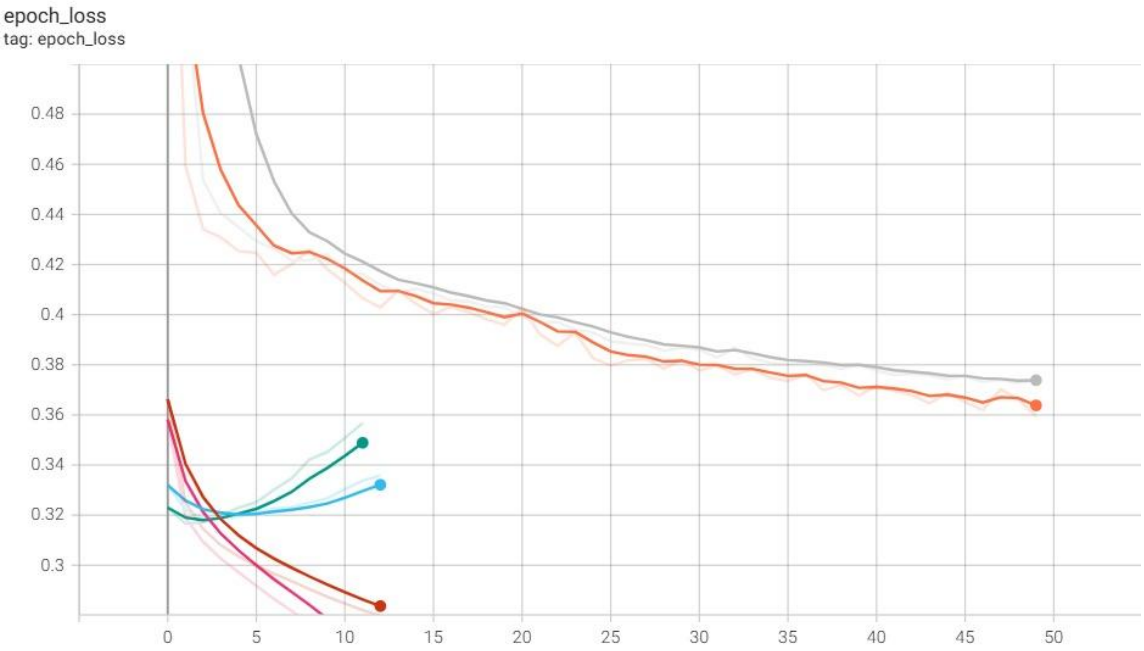

Figure 5: epoch loss.