

Machine Learning

Neural Network Implementation with Keras

Curtis Larsen

Utah Tech University—Computing

Spring 2025

Objectives

Objectives:

- ▶ Prepare data
- ▶ Train model
- ▶ Visualize learning curve
- ▶ Predict with model
- ▶ Tune model

Prepare Data

Issues

Scikit-Learn has better preprocessing capabilities than Keras or TensorFlow

- ▶ Numerical scaling can be done in either.
- ▶ Polynomial features is easier in Scikit-Learn.
- ▶ Imputing is easier in Scikit-Learn.
- ▶ Preprocessing can be completed one-time and cached.

Strategy

- ▶ Build a Scikit-Learn pipeline.
- ▶ Do all preprocessing in Scikit-Learn.
- ▶ Do *not* add a model at the end.
- ▶ Fit the pipeline to the training data.
- ▶ Transform the training data and save it.
- ▶ Transform the testing data and save it.

Sample Code

Sample Code

Train Model

Model Architecture

Structure considerations:

- ▶ Sequential model stacks layers in a linear fashion.
- ▶ Input layer size is determined by the data shape.
- ▶ Output layer is determined by the number of classes.
- ▶ Output layer activation is determined by type of prediction.
- ▶ Hidden layer size is a hyperparameter.
- ▶ Hidden layer count is a hyperparameter.
- ▶ Hidden layer activation function is a hyperparameter.

Optimizer

- ▶ Controls model updates.
- ▶ Stochastic gradient descent (SGD) is common.
- ▶ Newer optimizers can work faster.
- ▶ Learning rate is an important hyperparameter.
- ▶ When to stop.

Other

- ▶ Validation metrics
- ▶ Callbacks

Sample Code

Sample Code

Learning Curve

Plot

- ▶ Training metrics
- ▶ Validation metrics
- ▶ Signs of under/over fitting

Predict

Sample Code

Sample Code

Tune

- ▶ Add layers to get more complex fit.
- ▶ Add units to layers to get more complex fit.
- ▶ “Stretch Pants” models.
- ▶ Optimizer
- ▶ Metrics
- ▶ Learning rate
- ▶ Early stopping

Summary

- ▶ Prepare data with `sklearn` pipeline
- ▶ Build and train model with `Keras`
- ▶ Tune model, informed by learning curve