Applied Machine Learning Classification with Tensorflow

Ngan Le

thile@uark.edu

Recap: Classification with sklearn

- X_train, X_validate, y_train, y_validate = train_test_split(….)
- model = LogisticRegression(….)
- search = GridSearchC
- search.fit(X_train, y_
- print(search.best_est
- model = search.best_estimator_
- model.fit(X_train, y_train)
- predictions = model.predict(X_validate)

linear_model.LogisticRegression([penalty, …])	Logistic Regression (aka logit, MaxEnt) classifier.
linear_model.LogisticRegressionCV(*[, Cs, …])	Logistic Regression CV (aka logit, MaxEnt) classifier.
linear_model.PassiveAggressiveClassifier(*)	Passive Aggressive Classifier
linear_model.Perceptron(*[, penalty, alpha, …])	Read more in the User Guide.
linear_model.RidgeClassifier([alpha, …])	Classifier using Ridge regression.
linear_model.RidgeClassifierCV([alphas, …])	Ridge classifier with built-in cross-validation.
linear_model.SGDClassifier([loss, penalty, …])	Linear classifiers (SVM, logistic regression, etc.) with SGD training.

Use the **TensorFlow** toolkit to create a deep neural network that can perform **classification**

Dataset: UCI Heart Disease

Predicting the presence of heart disease

Dataset: Features

Feature	Description
age	age in years
sex	sex 0 = female 1 = male
cp	chest pain type 1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 = asymptomatic

Dataset: Features (continued)

Feature	Description
trestbps	resting blood pressure in Hg
chol	serum cholesterol in mg/dl
fbs	is fasting blood sugar > 120 mg/dl 0 = false 1 = true
restecg	results of a resting electrocardiograph 0 = normal 1 = ST-T wave abnormality 2 = left ventricular hypertrophy

Dataset: Features (continued)

Feature	Description
thalach	max heart rate
exang	exercise induced angina 0 = no 1 = yes
oldpeak	measurement of an abnormal ST depression
slope	slope of peak of exercise ST segment 1 = upslope 2 = flat 3 = downslope

Dataset: Features (continued)

Feature	Description
ca	count of major blood vessels colored by fluoroscopy 0, 1, 2, 3, or 4
thal	presence heart condition 0 = unknown 1 = normal 2 = fixed defect 3 = reversible defect

The Model: Output Layer

```
    tf.keras.layers.Dense(1, activation=tf.nn.sigmoid)
```

The Model: Loss

```
    model.compile(
        loss='binary_crossentropy',
        optimizer='Adam',
        metrics=['accuracy']
    )
```

The Model: Early Stopping

```
    tf.keras.callbacks.EarlyStopping(
        monitor='loss',
        min_delta=1e-3,
        patience=5,
    )
```

Your Turn