ICTSS00120
Artificial Intelligence Skill Set

Week 8: Hyperparameter Tuning

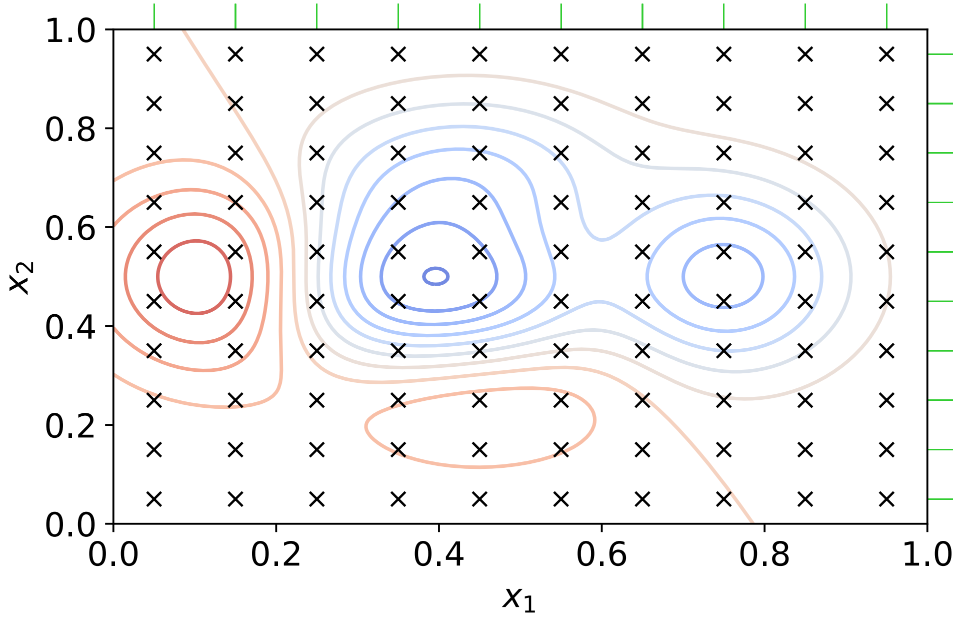
Lecturer: Jordan Hill

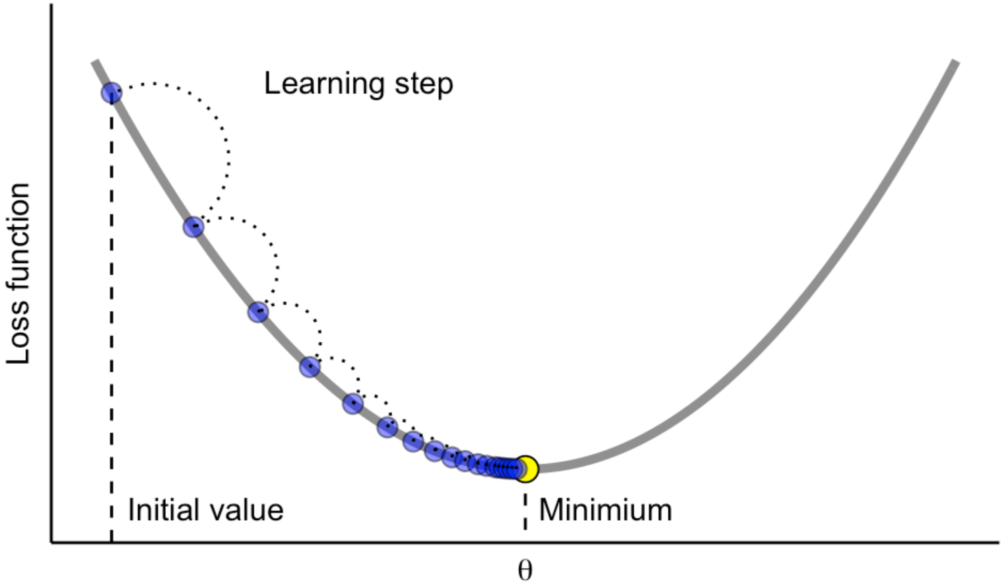
Learning Objectives

- Understand what hyperparameters are and their role in machine learning models.
- Learn about different methods for hyperparameter tuning.
- Implement hyperparameter tuning using scikit-learn.
- Evaluate the performance of machine learning models using various metrics (e.g., f-score, accuracy, precision/recall, loss metrics, confusion matrix).

What are Hyperparameters?

- **Definition**: Hyperparameters are external configurations to the model that must be set before the learning process begins.
- **Examples**:
  - Learning rate in gradient descent
  - Number of layers and nodes in a neural network
  - Regularization parameters (e.g., L1, L2)
  - Max depth of a decision tree





Why Hyperparameter Tuning?

- **Optimize Model Performance**:
  - Carefully chosen hyperparameters can significantly improve model accuracy and efficiency.
- **Avoid Overfitting/Underfitting**:
  - Regularization parameters can help to generalize the model better.
- **Balancing Performance Metrics**:
  - Tuning parameters like the class weight can balance precision and recall.

Common Methods for Hyperparameter Tuning

Grid Search

- **Definition**: Exhaustive search over specified parameter values for an estimator.
- **GridSearchCV** in scikit-learn performs hyperparameter tuning using cross-validation.

Random Search

- **Definition**: Randomly sampling parameter values from a specified distribution.
- Often quicker than grid search for large hyperparameter spaces.

Evaluating Model Performance

Common Metrics

- **Accuracy**: Proportion of true results among the total number of cases.
- **Precision**: Proportion of true positive results in terms of positive results returned by the classifier.
- **Recall**: Proportion of true positive results in terms of all samples that should have been identified as positive.
- **F1 Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: A table used to describe the performance of a classification model.

Calculating Evaluation Metrics with Scikit-Learn

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

# Predictions
y_pred = grid_search.predict(X_test)

# Calculating metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

# Printing results
print ( f"Accuracy: {accuracy}" )
print ( f"Precision: {precision}" )
print ( f"Recall: {recall}" )
print ( f"F1 Score: {f1}" )
print ( f"Confusion Matrix:\n{conf_matrix}" )
```

Visualizing the Confusion Matrix

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting the confusion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

Practical Exercise: Grid Search for Spam Detection

Steps:

1. **Data Preprocessing**:
    - Load and preprocess the spam dataset.
2. **Model Selection**:
    - Choose a classification model (e.g., SVM, RandomForest).
3. **Hyperparameter Tuning**:
    - Use GridSearchCV for hyperparameter tuning.
4. **Model Evaluation**:
    - Evaluate the performance using the discussed metrics.

Refer to the lab sheet for detailed steps and code snippets.

Summary and Next Steps

Key Points

- Definition and importance of hyperparameters.
- Common methods for hyperparameter tuning: Grid Search, Random Search.
- Implementing hyperparameter tuning using scikit-learn.
- Evaluating model performance using various metrics.

Homework

1. Complete the practical exercise on hyperparameter tuning.
2. Review the paper "Study on the effect of preprocessing methods for spam email detection" by Ruskanda, F.Z., 2019.

Next Week: Deep Learning Foundations

Questions & Answers

Q&A:

- Any questions from today's session??

Contact: jordan.hill@nmtafe.wa.edu.au