Lesson 10: Discovering Hidden Patterns

10.1. Introduction

Machine learning is a field of study and is concerned with algorithms that learn from examples.

Classification is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain. An easy to understand example is classifying emails as "spam" or "not spam."

10.2. Classification

There are many different types of classification tasks that you may encounter in machine learning and specialized approaches to modeling that may be used for each.

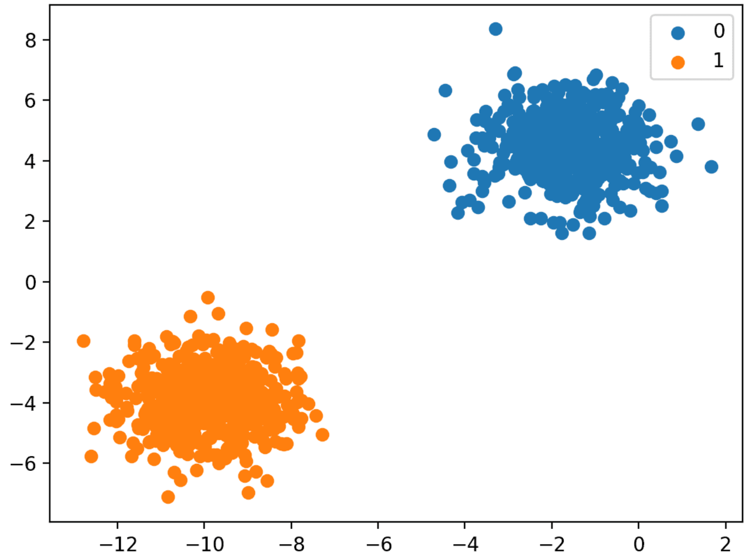
From a modeling perspective, classification requires a training dataset with many examples of inputs and outputs from which to learn.
A model will use the training dataset and will calculate how to best map examples of input data to specific class labels. As such, the training dataset must be sufficiently representative of the problem and have many examples of each class label.

10.2.1. Binary Classification

Binary classification refers to those classification tasks that have two class labels.
Examples include:

Email spam detection (spam or not).
Churn prediction (churn or not).
Conversion prediction (buy or not).

Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state.
For example "*not spam*" is the normal state and "*spam*" is the abnormal state. Another example is "*cancer not detected*" is the normal state of a task that involves a medical test and "*cancer detected*" is the abnormal state.
The class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1.
It is common to model a binary classification task with a model that predicts a Bernoulli probability distribution for each example.
The Bernoulli distribution is a discrete probability distribution that covers a case where an event will have a binary outcome as either a 0 or 1. For classification, this means that the model predicts a probability of an example belonging to class 1, or the abnormal state.
Popular algorithms that can be used for binary classification include:
Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machine, and Naive Bayes.



Scatter Plot of Binary Classification Dataset

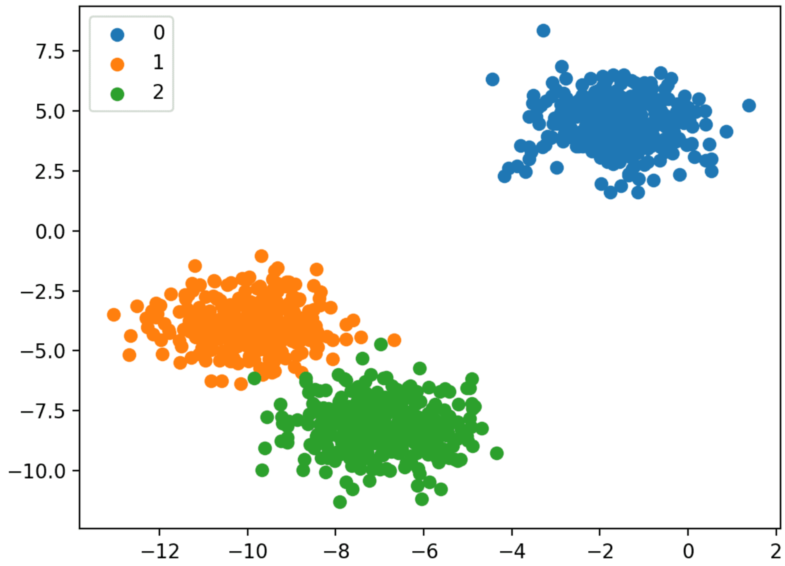
10.2.2. Multi-Class Classification

Multi-class classification refers to those classification tasks that have more than two class labels.
Examples include:

Face classification.
Plant species classification.
Optical character recognition.

Unlike binary classification, multi-class classification does not have the notion of normal and abnormal outcomes. Instead, examples are classified as belonging to one among a range of known classes.
The number of class labels may be very large on some problems. For example, a model may predict a photo as belonging to one among thousands or tens of thousands of faces in a face recognition system.
Problems that involve predicting a sequence of words, such as text translation models, may also be considered a special type of multi-class classification. Each word in the sequence of words to be predicted involves a multi-class classification where the size of the vocabulary defines the number of possible classes that may be predicted and could be tens or hundreds of thousands of words in size.
It is common to model a multi-class classification task with a model that predicts a Multinoulli probability distribution for each example.
The Multinoulli distribution is a discrete probability distribution that covers a case where an event will have a categorical outcome, e.g. K in {1, 2, 3, …, K}. For classification, this means that the model predicts the probability of an example belonging to each class label.
Many algorithms used for binary classification can be used for multi-class classification.
Popular algorithms that can be used for multi-class classification include:
k-Nearest Neighbors, Decision Trees, Naive Bayes, Random Forest, Gradient Boosting.
Algorithms that are designed for binary classification can be adapted for use for multi-class problems.
This involves using a strategy of fitting multiple binary classification models for each class vs. all other classes (called one-vs-rest) or one model for each pair of classes (called one-vs-one).

One-vs-Rest: Fit one binary classification model for each class vs. all other classes.
One-vs-One: Fit one binary classification model for each pair of classes.
Binary classification algorithms that can use these strategies for multi-class classification include: Logistic Regression, Support Vector Machine.



Scatter Plot of Multi-Class Classification Dataset

10.2.3. Multi-Label Classification

Multi-label classification refers to those classification tasks that have two or more class labels, where one or more class labels may be predicted for each example.
Consider the example of photo classification, where a given photo may have multiple objects in the scene and a model may predict the presence of multiple known objects in the photo, such as "*bicycle*," "*apple*," "*person*," etc.
This is unlike binary classification and multi-class classification, where a single class label is predicted for each example.
It is common to model multi-label classification tasks with a model that predicts multiple outputs, with each output taking predicted as a Bernoulli probability distribution. This is essentially a model that makes multiple binary classification predictions for each example.
Classification algorithms used for binary or multi-class classification cannot be used directly for multi-label classification. Specialized versions of standard classification algorithms can be used, so-called multi-label versions of the algorithms, including:
Multi-label Decision Trees, Multi-label Random Forests, Multi-label Gradient Boosting
Another approach is to use a separate classification algorithm to predict the labels for each class.

10.2.4. Imbalanced Classification

Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed. Typically, imbalanced classification tasks are binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

Examples include: Fraud detection, Outlier detection, Medical diagnostic tests.

These problems are modeled as binary classification tasks, although may require specialized techniques.

Specialized techniques may be used to change the composition of samples in the training dataset by undersampling the majority class or oversampling the minority class.

Examples include:

Random Undersampling, SMOTE Oversampling.

Specialized modeling algorithms may be used that pay more attention to the minority class when fitting the model on the training dataset, such as cost-sensitive machine learning algorithms.

Examples include:

Cost-sensitive Logistic Regression, Cost-sensitive Decision Trees, Cost-sensitive Support Vector Machines.

Finally, alternative performance metrics may be required as reporting the classification accuracy may be misleading.

Examples include:

Precision, Recall, F-Measure.

10.3. Deep Learning

Deep structured learning or hierarchical learning or deep learning in short is part of the family of machine learning methods which are themselves a subset of the broader field of Artificial Intelligence.

Deep learning is a class of machine learning algorithms that use several layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.

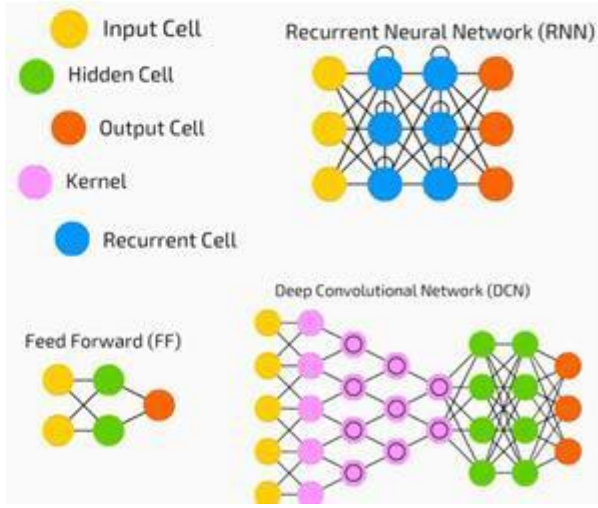
Deep neural networks, deep belief networks and recurrent neural networks have been applied to fields such as computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, and bioinformatics where they produced results comparable to and in some cases better than human experts have.

Deep Learning Algorithms and Networks −

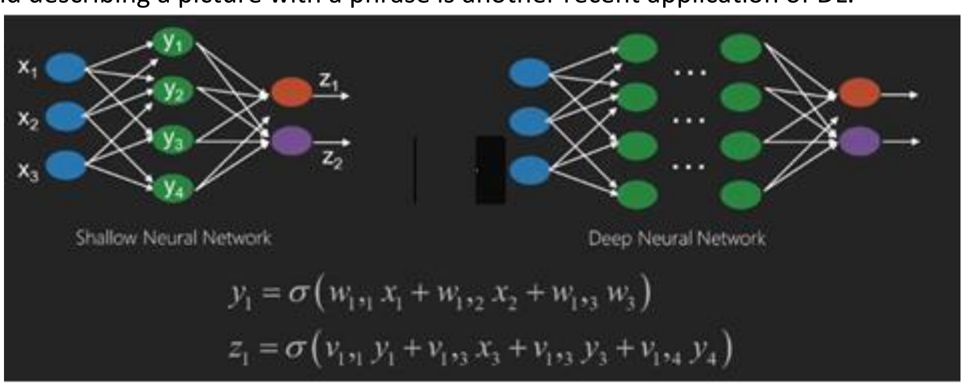
- are based on the unsupervised learning of multiple levels of features or representations of the data. Higher-level features are derived from lower level features to form a hierarchical representation.
- use some form of gradient descent for training.

Deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

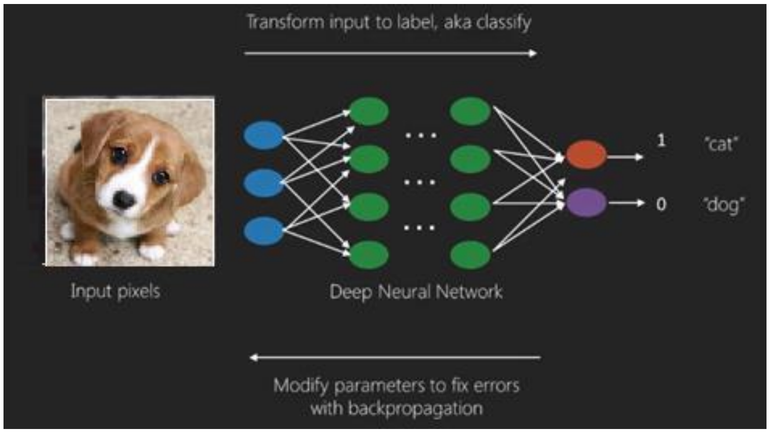
The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks. We have an input, an output, and a flow of sequential data in a deep network.



Neural networks are widely used in supervised learning and reinforcement learning problems. These networks are based on a set of layers connected to each other.
In deep learning, the number of hidden layers, mostly non-linear, can be large; say about 1000 layers.
DL models produce much better results than normal ML networks.
We mostly use the gradient descent method for optimizing the network and minimizing the loss function.
We can use the **Imagenet**, a repository of millions of digital images to classify a dataset into categories like cats and dogs. DL nets are increasingly used for dynamic images apart from static ones and for time series and text analysis.
Training the data sets forms an important part of Deep Learning models. In addition, Backpropagation is the main algorithm in training DL models.
DL deals with training large neural networks with complex input output transformations.
One example of DL is the mapping of a photo to the name of the person(s) in photo as they do on social networks and describing a picture with a phrase is another recent application of DL.



Neural networks are functions that have inputs like x1,x2,x3…that are transformed to outputs like z1,z2,z3 and so on in two (shallow networks) or several intermediate operations also called layers (deep networks).
The weights and biases change from layer to layer. ‘w’ and ‘v’ are the weights or synapses of layers of the neural networks.
The best use case of deep learning is the supervised learning problem.Here,we have large set of data inputs with a desired set of outputs.



Here we apply back propagation algorithm to get correct output prediction.
The most basic data set of deep learning is the MNIST, a dataset of handwritten digits.
We can train deep a Convolutional Neural Network with Keras to classify images of handwritten digits from this dataset.
The firing or activation of a neural net classifier produces a score. For example, to classify patients as sick and healthy, we consider parameters such as height, weight and body temperature, blood pressure etc.
A high score means patient is sick and a low score means he is healthy.
Each node in output and hidden layers has its own classifiers. The input layer takes inputs and passes on its scores to the next hidden layer for further activation and this goes on till the output is reached.
This progress from input to output from left to right in the forward direction is called **forward propagation.**

Lesson 10: Review Questions

1. Differentiate between classification and regression.
2. Discuss different types of classification and write a python program to demonstrate each,
3. Explain three metrics used to measure performance of a machine learning model.
4. Differentiate forward propagation and backward propagation as used in deep learning.
5. What is a recurrent neural network?