CLASSIFICATION ALGORITHM APPLICATION FOR PREDICTION OF DISEASE EVOLUTION

Teresa Vega Martínez and Alberto Montero Solera

teresavegamar@uma.es 13albertomontero@uma.es
Ingeniería de la Salud.
Universidad de Málaga.

This report systematically addresses key questions in machine learning and deep learning, covering foundational concepts, practical applications, and challenges in model development. We examine the importance of dataset splitting for unbiased model evaluation, explain cross-validation to improve model reliability, and describe artificial neural networks and their role in complex pattern recognition tasks. Additionally, the report highlights advanced deep learning methods, including recent applications like Google Translate and AlphaGo, which have demonstrated the transformative potential of these models.
The report also discusses common issues like overfitting and data scarcity, particularly in specialized fields like biomedical research. Techniques such as regularization, data augmentation, and transfer learning are explored as solutions to improve model performance and generalization. This structured approach provides a clear, concise overview of essential machine learning concepts, making it a useful resource for students and professionals in engineering and data science.

1 Questions

1.1 When developing a supervised method, why do we need to split the dataset into training and test sets?

In supervised learning, we aim to create a model that can generalize well to new, unseen data. To evaluate the model's performance objectively, we split the dataset into a **training set** and a **test set**:

- **Training Set:** This portion of the data is used to train the model. The model learns patterns, relationships, and associations from the input data and adjusts its parameters to minimize error.
- **Test Set:** This is a separate portion of the data that the model has not seen during training. After training, the test set is used to evaluate the model's performance on unseen data. This provides an estimate of how well the model is likely to perform in the real world, as it simulates the model's behavior on new data.

By using separate sets for training and testing, we prevent the model from simply memorizing the training data (overfitting) and ensure that it can generalize well. Without this split, we might mistakenly believe that the model performs well because it fits the training data closely, but in reality, it may perform poorly on new data.

1.2 What is cross-validation?

Cross-validation is a technique used to improve the robustness and reliability of model evaluation, particularly when the dataset is limited. It involves splitting the data into multiple subsets (or "folds") and performing multiple rounds of training and testing. The most common form is k-fold cross-validation, where the dataset is divided into k folds:

1. The model is trained on $k1$ folds and tested on the remaining fold.

2. This process is repeated k times, each time with a different fold used as the test set and the remaining *k1* folds as the training set.
3. The model's performance is then averaged over all k iterations to obtain a more stable and unbiased estimate of its generalization performance.

Cross-validation helps reduce the risk of overfitting and provides a more accurate assessment of how the model will perform on unseen data. It is especially useful when we have a small dataset, as it maximizes the use of all available data for training and testing.

1.3 What is artificial neural networks?

Artificial Neural Networks (ANNs) are computational models inspired by the human brain, designed to recognize patterns and relationships within data. They consist of interconnected nodes, or "neurons," arranged in layers:

1. Input Layer: Receives the raw data input.
2. Hidden Layers: Intermediate layers that process and transform the data by learning features or patterns. Complex ANNs may have many hidden layers, leading to a deep neural network.
3. Output Layer: Produces the final output, such as a class label in classification tasks or a predicted value in regression tasks.

Each connection between neurons has a weight, which determines the strength of influence one neuron has on another. Neurons also have an activation function that decides whether the neuron should be "activated" or "fired" based on the input it receives.

ANNs learn through a process called backpropagation, where the model adjusts weights based on the error between predicted and actual outputs. The objective is to minimize this error, effectively improving the model's accuracy.

ANNs are widely used for complex tasks such as image recognition, natural language processing, and more, due to their ability to model nonlinear relationships and learn from large amounts of data.

1.4 What is Deep Learning (DL)?

Deep Learning (DL) is a branch of machine learning that involves using artificial neural networks with many layers to automatically learn and extract complex patterns from large amounts of data. Unlike traditional machine learning, which often requires manually selected features, deep learning models can identify relevant features from raw data on their own by adjusting millions of internal parameters (weights). This capability has made DL very powerful for tasks like image recognition, natural language processing, and speech analysis, where relationships in the data are complex and require sophisticated representation.

1.5 What is new in DL models with respect to traditional feedforward neural networks?

Deep Learning models extend traditional feedforward neural networks in several important ways:

- While traditional feedforward neural networks have only a few layers, DL models use **deep architectures with many layers**. This increased depth allows them to learn hierarchical representations and capture very complex relationships within the data.
- DL models often include **specialized layers** tailored for different types of data. Convolutional Neural Networks (CNNs), for example, use convolutional layers to process spatial data, like images, by capturing spatial hierarchies (e.g., edges, shapes). Recurrent Neural Networks (RNNs) and Transformers are used for sequence data, such as text or time-series data, by modeling dependencies over time.

- Unlike traditional models, which rely on predefined features, DL models can **automatically extract high-level features**, reducing the need for manual feature engineering. For instance, in image recognition, they can progressively learn features like edges, shapes, and eventually entire objects without explicit programming.
- DL models are designed to handle **large datasets and complex calculations**. They benefit from GPU (Graphics Processing Unit) acceleration, which makes it feasible to train deep models on massive datasets, something traditional neural networks would struggle with.

1.6 Google (among others) has produced astonishing results in the application of DL models in different domains. Mention two of these cases describing shortly the problem solved.

- Google implemented neural machine translation (NMT), a deep learning-based approach, to improve its translation service. Traditional translation systems relied on fixed rules or statistical phrase-based translations, which often resulted in unnatural translations. Using DL, **Google Translate** learns from millions of text pairs across languages and can understand context better, producing fluent and accurate translations. The DL model can generalize across languages and capture subtle linguistic nuances that were previously challenging.
- **Google DeepMind's AlphaGo** demonstrated the power of DL in solving complex strategic problems, specifically the ancient board game Go, which has an enormous number of possible moves. Traditional algorithms could not handle this complexity effectively. AlphaGo used a combination of deep reinforcement learning and neural networks to evaluate the board positions and make decisions. By learning from human games and playing against itself, it reached a superhuman level, eventually defeating the world champion in Go—a task previously thought to be decades away from feasibility.

1.7 What is overfitting and how DL models avoid it?

Overfitting occurs when a model learns not only the main patterns in the training data but also the noise or irrelevant details. This results in a model that performs well on training data but poorly on new, unseen data. In other words, an overfitted model becomes too complex for the data it was trained on, capturing specificities instead of general trends.

DL models use several techniques to mitigate overfitting:

- **Regularization:** Methods like **L2 regularization** penalize large weights, which discourages the model from becoming overly complex. Dropout is another regularization technique that randomly deactivates neurons during training, forcing the model to generalize better by relying on different subsets of neurons each time.
- **Data Augmentation:** Data augmentation artificially increases the training dataset by creating modified versions of the existing samples, such as flipping, rotating, or adding noise to images. This prevents the model from memorizing the exact training samples and makes it more robust to variations.
- **Early Stopping:** By monitoring the model's performance on a separate validation set during training, we can stop training once the model's accuracy on this set stops improving, thus preventing it from overfitting the training data.
- **Batch Normalization:** This technique normalizes the inputs of each layer, stabilizing and speeding up the training process. It also acts as a form of regularization, helping the model generalize better.

1.8 Lack of data is a big limitation regarding the application of DL models to biomedical problems. What techniques can be applied to alleviate this problem.

In biomedical applications, data scarcity is a common issue because collecting labeled data, such as medical images with annotations, is often expensive, time-consuming, or constrained by privacy regulations. To tackle this limitation, we can apply several techniques:

1. In fields like medical imaging, where each image is valuable, **data augmentation** is commonly used to increase the effective dataset size. For example, applying rotations, scaling, or flipping to images helps the model become more invariant to transformations, effectively learning from a larger sample space.
2. **Transfer learning** allows us to take a model pre-trained on a large, general-purpose dataset (like ImageNet for images) and fine-tune it on a smaller biomedical dataset. By using the knowledge gained from the large dataset, the model can adapt and learn specific features in the biomedical domain more effectively.
3. Deep learning models like GANs (Generative Adversarial Networks) or VAEs (Variational Autoencoders) can be used to **generate synthetic data** that mimics real biomedical data. These models create new data samples that look similar to the real data, expanding the dataset without needing real samples.
4. In **semi-supervised learning**, a model learns from a small labeled dataset combined with a larger unlabeled dataset. This is useful in situations where labeling data is expensive, as the model can leverage both types of data to improve its performance.
5. **Sharing data** across research institutions (with appropriate privacy safeguards) can allow models to be trained on larger and more diverse datasets, improving generalization and performance.