ML Model for Belt Type Prediction: Approach and Challenges

Problem type:

Multi Label Image classification

Introduction: This report outlines the approach taken to develop a multi-label classification model for predicting belt types in a manufacturing setting. The objective was to accurately assign multiple labels to each belt based on its characteristics. This information is critical for e-commerce setups

Dataset Description:

A .csv file with following columns:

Image_urls: Containing URLs of images on which the model will be trained.

Labels_crops: Gives dimensions of specific object to be classified within the images.

Labels: Comma separated values of multiple labels an image belongs to.

Approach:

1.Data Collection and Preprocessing:

a. **Downloading images**: URLs can be directly used each time the images are fed into a CNN model after converting them to numpy arrays using

```
response = urllib.request.urlopen(image_url)
```

But it uses extensive computational resources each time the numpy array list of images is created. So for my usecase, I preferred to download all the images in a directory.

b. **.csv file processing including One-hot encoding**: The csv file is transformed such that there's a separate column for each label which is marked either 1 or 0 based on the labels assigned to each image.

c. Extra columns like 'labels', 'image_url' were deleted and an 'index' column was added.

d. Images were cropped according to the given label crops.

e. Images were resized and normalized according to the CNN model and available computational resources.

f. Input and output features are converted into suitable type and shaped to feed to the CNN model.

2. Model Selection: Explored various multi-label classification algorithms like Convolutional Neural Networks (CNNs), YOLO due to their proficiency in image-related tasks.

3. Model Architecture: Chose a CNN-based architecture with multiple output nodes, each corresponding to a specific belt type label with 'softmax' as the activation function.

4. Loss Function and Evaluation Metrics: Utilized binary cross-entropy loss for each output label, considering the multi-label nature of the problem. Evaluation was done using accuracy and F1-score, which balances precision and recall for each label.

5. Training and Validation: Split the dataset into training and validation sets, keeping the bias-variance tradeoff in mind to prevent overfitting.

6. Hyperparameter Tuning: Iteratively tuned hyperparameters like learning rate, batch size, and dropout rate using techniques such as random search and grid search to achieve optimal model performance.

Challenges Faced:

1. Imbalanced Data: The dataset exhibited class imbalance, with some belt types having significantly fewer examples. This led to biased predictions for dominant classes and poor performance for minority classes.

2. Label Dependencies: Certain belt types shared common features, resulting in label dependencies. For example, 'Braided' belts often had similarities with 'Punk' belts. This made it challenging for the model to make distinct predictions.

3. Limited computational resources: Building a CNN model to a specific problem required extensive computational resources. Even incorporating small changes lead to re-computation at many levels which was a complex task. With enough computational resources, the performance of this model can be increased by a significantly large percentage.

Possible future works:

1. **Data Augmentation Strategies**: advanced data augmentation techniques like random erasing and mixup can be used to tackle data imbalance, artificially generating diverse samples for underrepresented classes. Funtions like ImageDataFenerator can be used for it.

2. **Label Correlation Modeling**: A label correlation matrix can be introduced to account for dependencies between belt types. This matrix can be incorporated as a regularization term in the loss function, encouraging the model to respect label relationships.

3. **Pre-trained models**: The abilitites of pre-trained models like ResNet can be utilized for increased performance.

Conclusion:

Developing a multi-label classification model for belt type prediction required a holistic approach that encompassed data preprocessing, model selection, architecture design, and innovative solutions to overcome challenges. The implemented strategies enhanced the model's performance, particularly in dealing with imbalanced data and label dependencies, leading to more accurate and robust predictions across various belt types. Further iterations could explore ensemble techniques and attention mechanisms to achieve even higher accuracy and robustness.