Image Captioning SOC

Name - Gohil Megh Hiteshkumar
Roll Number - 210050055

9th July 2023

Module 1 - Getting on with ML

A systematic framework is essential for a design of a machine learning model system. The framework helps in development, training and evaluation of the ML model. This framework typically comprises of following components

- **Data Preprocessing** : This step involves preparing the data for model. For this we do normalization/scaling, handling missing values, and addressing outliers. It ensures that the data is in a suitable format for the subsequent steps.
- **Feature Engineering** : Feature extraction or engineering involves transforming the raw input data into a numerical representation. Techniques such as one-hot encoding for categorical variables, dimensionality reduction, or creating new features based on domain knowledge are employed in this step.
- **Model Selection** : Based on the nature of the data and the specific task at hand, an appropriate machine learning algorithm is chosen. The selection involves classification algorithms (e.g., logistic regression, support vector machines, decision trees), regression algorithms (e.g., linear regression, random forests), clustering algorithms, or neural network architectures. The goal is to select a model that is best suited for the given problem.
- **Model Training** : In this step, the selected model is trained using labeled data. During training, suitable loss and accuracy functions are determined based on the task. The model's performance is assessed by monitoring the loss and accuracy metrics. This helps in identifying overfitting or underfitting. Hyperparameter tuning is applied to optimize the model's performance.
- **Model Evaluation** : Once the model is trained, its performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's predictive capabilities and its ability to generalize unseen data. Regularization techniques are commonly used to enhance the model's performance and ensure its applicability to the target task.

1 Mdule 2 - CNN : Convolutional Neural Network

Convolutional neural networks (CNNs) are a powerful type of artificial neural network commonly used for tasks such as image recognition and classification. CNNs incorporate convolutional layers, which enable them to capture spatial relationships and patterns within images. An image is an input here, loss is calculated using appropriate loss function. To minimize this loss, techniques like backpropagation and gradient descent are used.

1.1 More about convolution layer :

Convolution layer consist of following hyperparameters:

- K : number of filters.
- F : spatial extent of the filter.
- S : known as stride. It is value by which the filter will slide.
- P : zero padding to maintain size spatially.

Assuming the dimenison of image as $W \times H \times D$. Each filter with size $F \times F \times D$(same as depth of image) is convolved with the image. Here convolution is done by sliding the filter over image and taking dot product of the filter with image. Now we get an "Activation map" i.e. $W_1 \times H_1 \times 1$, there are K such maps. We stack them together to get $W_1 \times H_1 \times K$ image. So convolutional layer converts input $W \times H \times D$ to $W_1 \times H_1 \times K$.
Here, the size of filters are smaller than the size of image and hence can result in shriking of volume spatially. TO avoid this and to preseve the spatial size for representation, we use zero-padding. We add padding to borders by introducing zeroes. Commonly, convolution layer has $S = 1$ and $P = \frac{F-1}{2}$. Thus, $W_1 = \frac{W-F+2*P}{S} + 1$ $H_1 = \frac{H-F+2*P}{S} + 1$
A convolutional neural network (CNN) is composed of neurons with local connectivity, where each neuron takes input x_i and w_i values from neighboring axons. These inputs undergo convolution with a function f, and the resulting output $f(\sum_i wixi + b)$ is transmitted to the output axon.
In a CNN, each neuron produces an activation map, which is a $W_1 \times H_1$ sheet of neuron outputs. These neurons are connected to small regions in the input. Importantly, all neurons in a particular activation map share the same set of parameters.
In this context, a "$F \times F$ filter" corresponds to a receptive field of size $F \times F$ for each neuron. For instance, if we have 5 filters of size 5×5×3 applied to an input volume with dimensions 32×32×3, the convolutional layer consists of neurons arranged in a 3D grid of size 28×28×5. Consequently, there will be 5 different neurons, each examining the same region in the input volume.
This arrangement allows CNNs to effectively capture local patterns and spatial relationships within images or other input data. By utilizing shared parameters, CNNs can efficiently extract features and learn hierarchical representations, enabling them to excel in tasks such as image recognition, object detection, and more.

1.2 Pooling layer :

Apart from convolutional layer, there are pooling layers. The pooling layer in a convolutional neural network (CNN) reduces the size of the representations, making them more manageable.

It operates independently on each activation map. The pooling layer has two hyperparameters: the spatial extent of filters (F) and the stride (S). It produces a smaller volume of size $W_2 \times H_2 \times D_2$, where $W_2 = \frac{W_1 - F}{S}$, $H_2 = \frac{H_1 - F}{S}$, and $D_2 = D_1$. The pooling layer introduces no additional parameters and performs a fixed function on the input. Common settings include $F = 2, S = 2$, or $F = 3, S = 2$. Max pooling, which takes the maximum value within each filter, and average pooling are commonly used types of pooling.

1.3 Activation Layer :

The activation layer in a neural network adds non-linearity and complexity to the model. It applies an activation function to each element of a matrix. Common activation functions include ReLU, sigmoid, and tanh.

1.4 Fully Connected Layer :

The fully connected layer is present at the end of a neural network and contains neurons that connect to the entire input volume, similar to an ordinary neural network. It computes scores and is fully connected. In convolutional neural networks (CNNs), the fully connected layer is used for tasks such as image classification and recognition in computer vision, leveraging their ability to detect patterns and features.

2 Module 3 - Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are designed for processing sequential data, where the output at each step depends on the previous data. However, RNNs often encounter the vanishing and exploding gradient problem during backpropagation which can affect training effectiveness. To address the exploding gradient problem, gradient clipping is commonly used. It limits the magnitude of the gradients during training, preventing them from becoming too large and destabilizing the learning process.

On the other hand, the vanishing gradient problem in RNNs can be mitigated by using Long Short-Term Memory (LSTM) networks. LSTMs are a special type of RNN that are specifically designed to handle long-term dependencies and avoid the vanishing gradient issue.

Similar to traditional RNNs, LSTMs have a hidden state (H_{t-1} and H_t) that represents short-term memory. Additionally, LSTMs introduce a cell state (C_{t-1} and C_t) that represents long-term memory. The architecture of an LSTM includes several gates:

Forget Gate: The forget gate determines how much of the previous long-term memory to retain, based on the previous hidden state and current input. It outputs a binary value that is multiplied with the previous long-term memory to determine what information should be forgotten or remembered.

Input Gate: Similar to the forget gate, the input gate operates on the previous hidden state and current input. Its purpose is to decide which new information should be incorporated into the LSTM's memory state.

Output Gate: The output gate takes into account both the current long-term memory and current short-term memory. It determines the information to be output from the LSTM.

By utilizing these gates, LSTMs are capable of capturing and preserving important information over long sequences, enabling them to effectively model and remember dependencies in sequential data.

3 NLP

Natural Language Processing (NLP) is an exciting field that explores how computers and human language can interact seamlessly. Its primary goal is to develop algorithms and models that can comprehend, interpret, and generate human language effectively. A vital aspect of NLP is preprocessing, where techniques like tokenization, removing unnecessary words, refining word forms, and handling special characters are employed. By employing NLP, we can enhance various areas, including sentiment analysis, machine translation, text summarization, and chatbots. Ultimately, NLP strives to facilitate seamless communication and deeper understanding between humans and computers.