FUNCTIONAL API REVIEW REPORT
INFO 7374 - Team 11

Dhanashri Palodkar Parag Bhingarkar Rajeshree Kale

Introduction

“**More data beats clever algorithms, but better data beats more data.**”

Creating better data depends on the model. A model is a sequence of data with fully configurable modules which can be plugged in easily with any framework to get trained and performed effortlessly. Keras being user-friendly, modular, extensible with python base makes creating models straightforward. The core data structure of Keras is organizing layers and deals with three different methodologies of layering which are Sequential model- a linear stack of layers and Keras functional API.

Sequential API is linear stacking of layer where a deep network is created by creating an instance of layers and passing through constructor. it is a restricted model since it needs to know the shape of layers heretofore. We can use this API to create the model for RNN, CNN for machine translations and sentiment analysis. But what to do when we have multiple inputs and multiple output model or a shared layer framework? In such complex architecture, Keras gives an option of Functional API which allows building graphs and layers. Functional API is used for problems Image classification based on color. In this article, we are focusing on Functional API

Densely Connected Networks:

A dense network is a network in which the number of links of each node is close to the maximal number of nodes. Each node is linked to almost all other nodes. The total connected case in which exactly each node is linked to each other node is called a completely connected network. Examples of higher link density are epidemic spreading, the neural network of the brain and telecommunication networks.

Example of Neural Networks:

CIFAR 10 IMAGE CLASSIFICATION: The model receives black and white 64×64 images as input, then has a sequence of two convolutional and pooling layers as feature extractors, followed by a fully connected layer to interpret the features and an output layer with a sigmoid activation for two-class predictions.

- A layer instance is callable (on a tensor), and it returns a tensor
- Input tensor(s) and output tensor(s) can then be used to define a Model
- Such a model can be trained just like Keras Sequential models.

Advantages:

All models are callable, just like layers-

With the functional API, it is easy to reuse trained models: you can treat any model as if it were a layer, by calling it on a tensor. Note that by calling a model you aren't just reusing the architecture of the model, you are also reusing its weights.

Multiple Input and Multiple Output

Functional API is an alternative to sequential models. Functional APIs allows us to build a model that can accept multiple inputs and produce multiple outputs. Let's consider the example of Twitter. We seek to predict how many retweets and likes a news headline will receive on

Twitter. The primary input to this model would be the headline of the tweet and secondary input could be the date or time of the tweet. Adding the second input would increase the prediction accuracy.

Shared Layers Model

Functional API allows flexibility to create networks irrespective of the size of the network. Shared Layers model can also be rephrased as Symmetric Model since it involves multiple symmetric layers spread with a single input. In Shared layer, we introduce multiple convoluted layers with different shapes or kernel. Functional API helps to build it architecture. In this model whenever we introduce a layer at input, we create a new tensor which adds a node,and when we call the same input multiple times it introduces multiple nodes indexed as 0,1,2,3 which is also called as **Layered node**.

Let's consider the data set of criminal activities for criminal detection, where we have to create a model that can tell if all the crimes are from the same person or not which will allow us to compare criminal by the similarity of their crime information and its instance. To achieve this result, we need to design a model that encodes the criminal information into vectors and then concatenates the vectors and then adds a logistic regression; this output is a probability that if the two crimes share the same criminal.

Inception Module

The concept of Inception module focuses on the kernel size. Basically, the proper kernel size depends on the image data sets. When we have a dataset of images with two types of information distribution i.e. global and local, A larger kernel is preferred for information that is distributed more globally, and a smaller kernel is preferred for information that is distributed more locally. So, in that case, having a proper size of the kernel is very important to get most data out of the image/video. Inception module is the solution to this problem, having filters with multiple sizes would help to scan and retrieve maximum data.

Residual connection on a convolution layer

Cornell University presents the aforementioned framework as a solution to the deeper neural network where it reformulates the layer learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. These residual networks are easier to optimize and can gain accuracy.

Shared vision Model

It uses the image-processing module on two inputs, to classify whether two MNIST digits are the same digit or different digits.

Visual & Video Question answering Model

Recent developments in Deep Learning has paved the way to accomplish tasks involving multimodal learning. Visual Question Answering (VQA) is one such challenge which requires high-level scene interpretation from images combined with language modeling of relevant Q&A. Given an image and a natural language question about the image, the task is to provide an accurate natural language answer. A Keras implementation of one such end-to-end system can be used to accomplish the task.

FUNCTIONAL API REVIEW REPORT
INFO 7374 - Team 11

Dhanashri Palodkar Parag Bhingarkar Rajeshree Kale

1) Selects the correct one-word answer when asked a natural-language question about a picture.
2) Works by encoding the question into a vector, encoding the image into a vector, concatenating the two
3) Trains on top of logistic regression over some vocabulary of potential answers.

A video QA model is required to locate and explore a sequence of frames to firstly recognize a set of specific objects and activities and secondly identify the relationship between objects and the relationship between objects and actions.

Example of Visual Question Answering Model:

Free-form, open-ended questions collected for images via Amazon Mechanical Turk. Note that commonsense knowledge is needed along with a visual understanding of the scene to answer many questions.

Applications:

VQA has the distinctive advantage of pushing the frontiers on “AI-complete” problems while being amenable to automatic evaluation. Given the recent progress in the community, we believe the time is ripe to take on such an endeavor.