第2章 データの入出力/型/構造から実行メカニズムまで

レッスン2…TensorFlowプログラミングの基礎知識

渡邉 輝

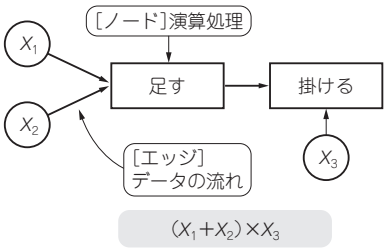

図1 TensorFlowの基本概念「グラフ」のイメージ
データがエッジ(矢印線)によって流れノードでデータに対しての演算処理が実行される．エッジは必ず一方通行で両方向になることはない．その性質から有向非巡回グラフとも呼ばれる

ここでは前章で構築した開発環境を利用して，コーディングをしながらTensorFlowの基本への理解を深めていきます．

● 準備…開発環境の立ち上げ

Dockerで開発環境を作成した方は，Docker Quickstart Terminalを立ち上げ，TensorFlow開発用コンテナを起動し(docker startコマンド)，コンテナOSにアクセスしたらJupyter Notebookを起動してください(jupyter notebookコマンド)．

マシンに直接インストールした方は，ターミナルまたはコマンド・プロンプトからJupyter Notebookだけを起動します．ブラウザからJupyter Notebookへアクセスし，サンプル・コーディング用の新規ファイルを作成したら準備完了です．

● 各ノードをエッジでつないだものがグラフ

TensorFlowは「グラフ」という概念に基づいて算術演算を実装するよう設計されています．従ってTensorFlowを利用する場合には，「グラフ」とは何か，それをどのように実装，実行するのかを理解しなければなりません．

グラフを構成する要素はノードとエッジです(**図1**)．ノードは丸や四角で表現され，特定の演算処理を実装します．エッジはノードをつなぐ矢印線でデータの流れを表現します．1つのノードの観点から見ると，自分自身に向けられたエッジは入力を意味し，外向きのエッジは演算後の出力を意味します．つまり，グラフとは一連の演算処理です．データがエッジによって流れを制御され，ノードに実装された演算処理によって次々と姿を変えていくことになります．TensorFlowはグラフを構築するための部品を提供します．

表1 TensorFlowのデータ型一覧

データ型	Pythonの型	説　明
DT_FLOAT	tf.float32	32ビット浮動小数点
DT_DOUBLE	tf.float64	64ビット浮動小数点
DT_INT8	tf.int8	8ビット符号あり整数
DT_INT16	tf.int16	16ビット符号あり整数
DT_INT32	tf.int32	32ビット符号あり整数
DT_INT64	tf.int64	64ビット符号あり整数
DT_UINT8	tf.uint8	8ビット符号なし整数
DT_UINT16	tf.uint16	16ビット符号なし整数
DT_STRING	tf.string	変数長のバイト配列(テンソルの各要素がバイト配列)
DT_BOOL	tf.bool	真偽値
DT_COMPLEX64	tf.complex64	32ビット浮動小数点を2つ使う複素数(それぞれ実数部分と虚数部分を表現する)
DT_COMPLEX128	tf.complex128	64ビット浮動小数点を2つ使う複素数(それぞれ実数部分と虚数部分を表現する)
DT_QINT8	tf.qint8	量子化された処理に使われる8ビット符号あり整数
DT_QINT32	tf.qint32	量子化された処理に使われる32ビット符号あり整数
DT_QUINT8	tf.quint8	量子化された処理に使われる8ビット符号なし整数

● グラフ内に流せるデータはテンソルという構造をとる

グラフ内に流すことのできるデータはテンソルとい