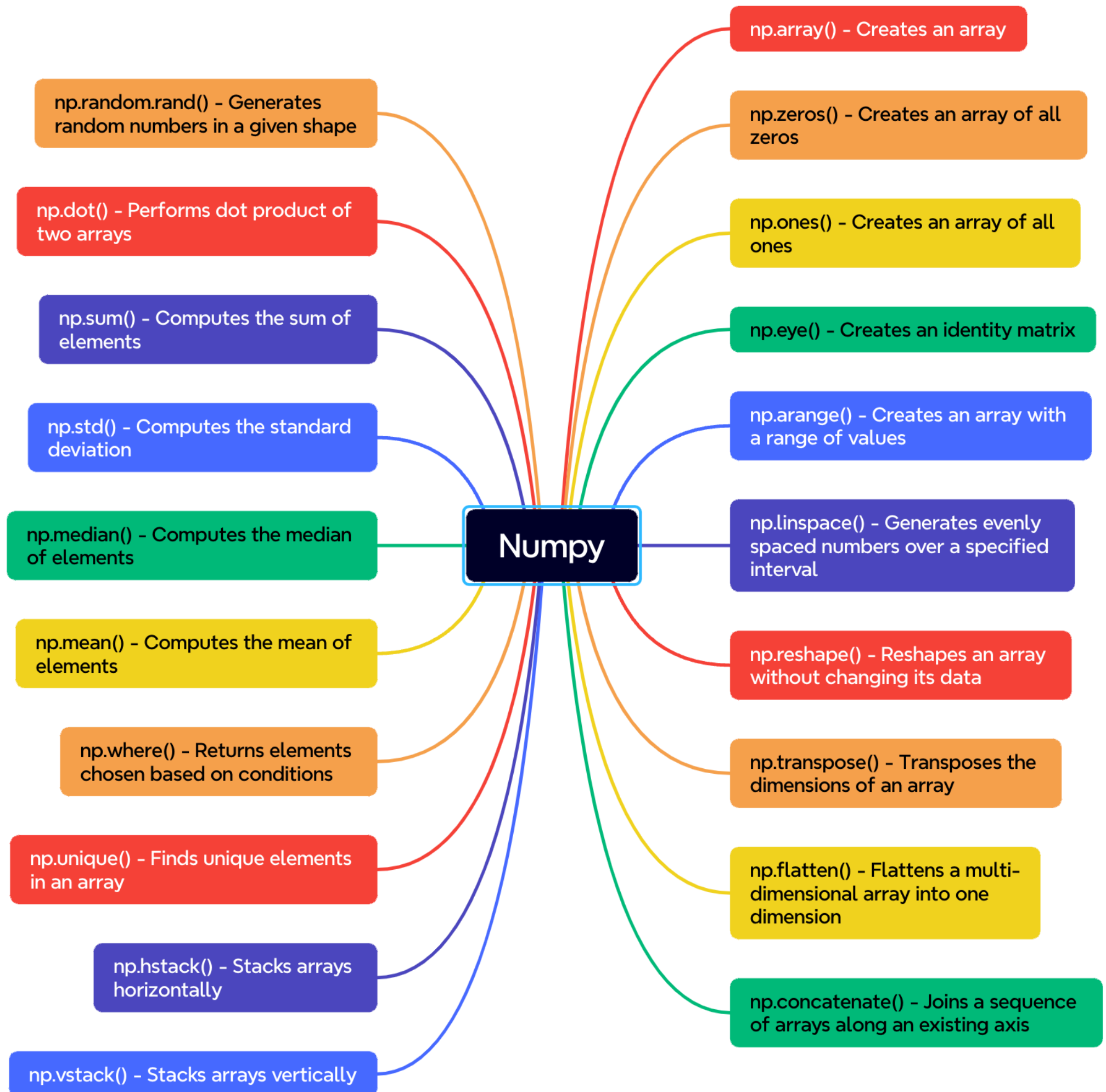


# Top 100 Commands Every DataScientist Should Know

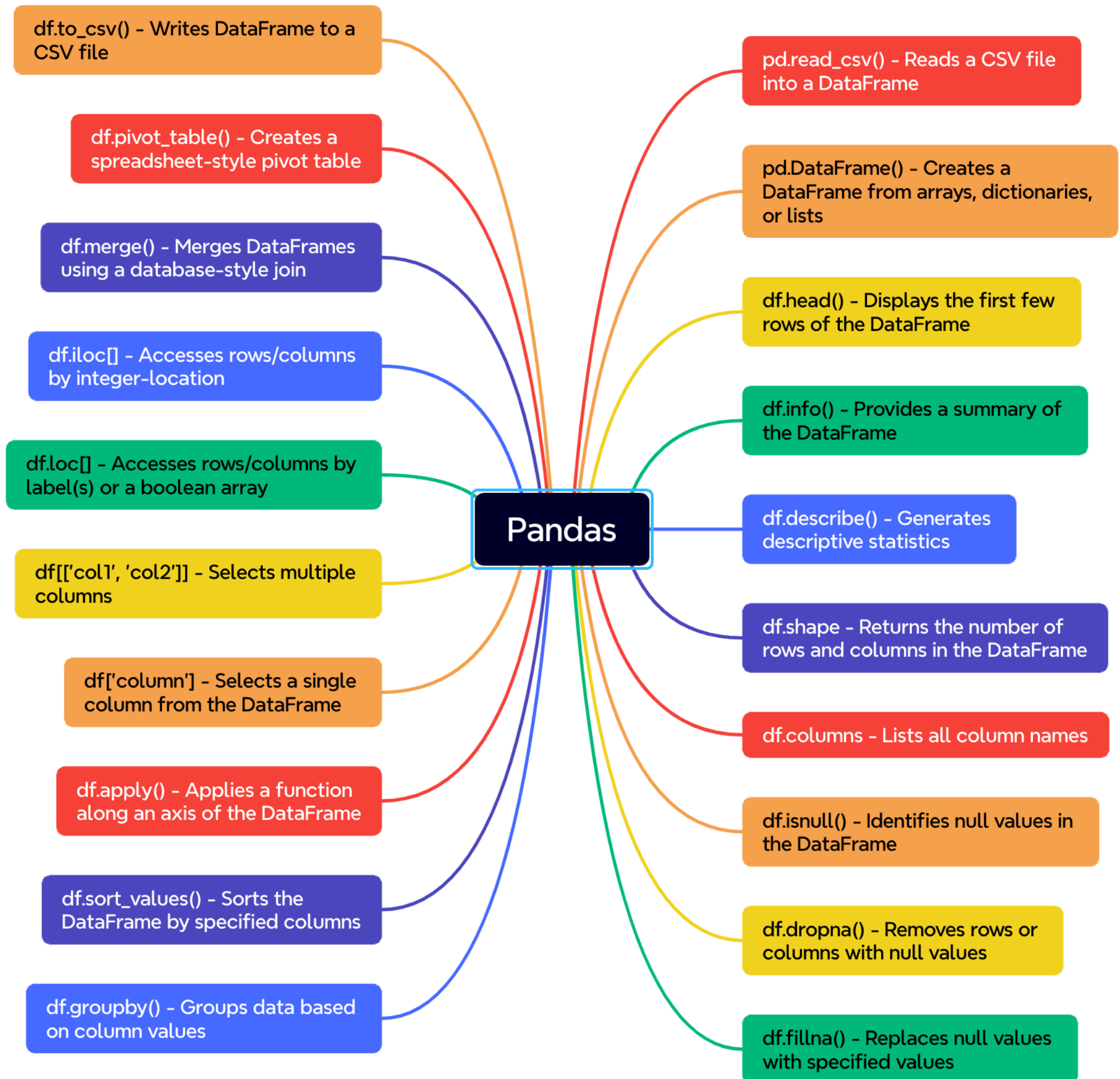
**Top 20 from Numpy, Pandas, Scikit-learn,  
Tensorflow, Pytorch**

Your cheat sheet before  
every Interview

# Numpy



# Pandas





# Scikit-learn

## Scikit-Learn

`roc_auc_score()` - Calculates the Area Under the Receiver Operating Characteristic Curve

`accuracy_score()` - Calculates the accuracy of a classification model

`mean_squared_error()` - Computes Mean Squared Error for regression evaluation

`classification_report()` - Generates a detailed classification performance report

`confusion_matrix()` - Computes the confusion matrix to evaluate classification

`Pipeline()` - Creates a pipeline to streamline pre-processing and modeling

`GridSearchCV()` - Performs exhaustive search for optimal hyperparameters

`cross_val_score()` - Evaluates a model with cross-validation

`KNeighborsClassifier()` - Fits a k-nearest neighbors model for classification

`DecisionTreeClassifier()` - Fits a decision tree model for classification

`train_test_split` - Splits data into training and test sets

`StandardScaler()` - Scales features to zero mean and unit variance

`MinMaxScaler()` - Scales features to a given range, typically [0, 1]

`SimpleImputer()` - Handles missing values by filling them with a specified strategy

`PCA()` - Principal Component Analysis for dimensionality reduction

`KMeans()` - Clustering algorithm to classify data into k clusters

`LogisticRegression()` - Applies logistic regression for binary classification

`LinearRegression()` - Fits a linear regression model

`RandomForestClassifier()` - Fits a Random Forest model for classification

`SVC()` - Support Vector Classifier for classification tasks

# Tensorflow

## TensorFlow

`tf.GradientTape()` - Records gradients for automatic differentiation

`tf.optimizers.Adam()` - Initializes the Adam optimizer for training

`tf.losses.SparseCategoricalCrossentropy()` - Loss function for classification tasks

`tf.data.Dataset.shuffle()` - Shuffles the elements of a dataset

`tf.data.Dataset.batch()` - Combines elements of a dataset into batches

`tf.data.Dataset.from_tensor_slices()` - Creates a dataset from tensor slices

`tf.keras.Model.predict()` - Generates output predictions for input samples

`tf.keras.Model.evaluate()` - Evaluates the model on a test dataset

`tf.keras.Model.fit()` - Trains the model with data

`tf.keras.Model.compile()` - Configures a model for training

`tf.constant()` - Creates a constant tensor

`tf.Variable()` - Creates a variable tensor

`tf.placeholder()` - Placeholder for feeding data (used in TensorFlow 1.x)

`tf.convert_to_tensor()` - Converts data to a tensor

`tf.Session()` - Executes operations in a session (used in TensorFlow 1.x)

`tf.keras.Sequential()` - Initializes a sequential model

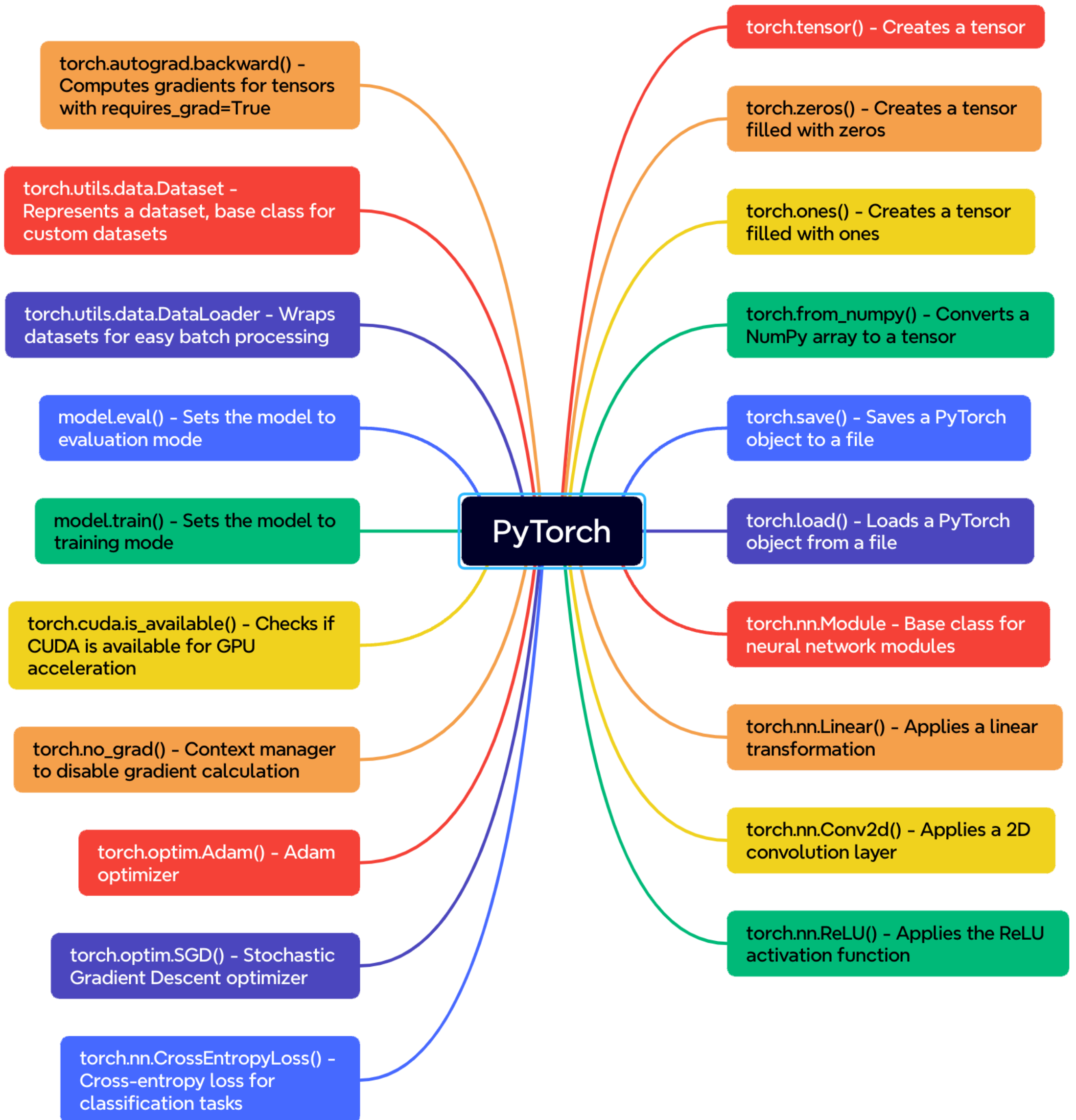
`tf.keras.layers.Dense()` - Adds a fully connected layer to a model

`tf.keras.layers.Conv2D()` - Adds a 2D convolution layer

`tf.keras.layers.MaxPooling2D()` - Adds a max pooling layer for down-sampling

`tf.keras.layers.Dropout()` - Adds dropout regularization to reduce overfitting

# PyTorch



# Contact for Online and Offline DataScience Course:

WhatsApp : +91-9008477433

[hello@neuronnetwork.in](mailto:hello@neuronnetwork.in)

[anasp.machinelearning@gmail.com](mailto:anasp.machinelearning@gmail.com)

Website : [www.neuronnetwork.in](http://www.neuronnetwork.in)

LinkedIn: <https://www.linkedin.com/in/anas-p-296b38258/>

YouTube: <https://www.youtube.com/@MLWithAP>



**NEURON**  
NETWORK