

Democratizing Deep Learning with the ktrain Library

The **ktrain** library is a free, open-source Python package developed by IDA researcher Arun Maiya to help build, tune, train, and deploy neural networks. The **ktrain** software package acts as a lightweight wrapper around the deep learning library TensorFlow Keras (in addition to other libraries). It is designed to make sophisticated, state-of-the-art machine learning models easier to apply by domain experts who may have little experience in data science and artificial intelligence (AI). It is also well-suited for rapid-prototyping by experienced practitioners.

In as little as three or four lines of code, **ktrain** allows you to easily and quickly employ fast and easy-to-use pre-canned models for textual data (e.g., DTIC technical reports and non-English news articles), visual data (e.g., satellite imagery and photos), and graph data (e.g., social networks). This out-of-the-box support includes:

- **Text:**

- *Text Classification*: auto-categorize documents by topic or sentiment
- *Text Regression*: predicting numbers (e.g., prices) purely from text descriptions
- *Sequence Labeling*: extract sequences of words that represent some concept of interest
- *Topic Discovery*: reveal latent subjects and themes buried in a large document collection

(continued)

- *Document Similarity*: find and score new documents that are semantically similar to documents of interest
- *Document Recommendation Engine*: recommend documents that are thematically related to given text
- *Question Answering*: submit questions to a large text corpus and receive exact answers
- *Text Summarization*: generate short summaries of long documents
- **Vision**:
 - *Image Classification*: autocategorize images across various dimensions (e.g., automatically recognize what is depicted in aerial imagery)
- **Graphs**:
 - *Graph Node Classification*: autocategorize nodes in a given graph (e.g., identify troublesome social media accounts)
 - *Graph Link Prediction*: predict missing links in social networks

Other features include tools to help automatically tune hyperparameters like the learning rate, explainable AI to help make sense of predictions, and support for non-English text analyses (e.g., Chinese and Arabic). Through automation or semiautomation, **ktrain** facilitates the full machine-learning workflow from curating and preprocessing inputs (i.e., ground-truth-labeled training data) to training, tuning, troubleshooting, and deploying models. Where possible, **ktrain** automates (either algorithmically or through setting reasonable defaults), but also allows users to make choices that best fit their unique application requirements. In this way, **ktrain** differs from typical automatic machine learning (AutoML) solutions in that it uses automation to *augment* and *complement* human engineers rather than attempting to replace them. In doing so, the strengths of both are better exploited. This concept has been recently branded as Augmented Machine Learning or AugML.

All deep learning tasks in **ktrain** can be accomplished using a similar set of four easy steps, as shown in the example at right.

Example: Building a Chinese-Language Sentiment Analyzer Using BERT (a deep learning model by Google)

```
import ktrain
from ktrain import text as txt

# STEP 1: load and preprocess data
trn, val, preproc = txt.texts_from_folder('data/ChnSentiCorp',
                                         maxlen=75,
                                         preprocess_mode='bert')

# STEP 2: load model
model = txt.text_classifier('bert', trn, preproc=preproc)
learner = ktrain.get_learner(model, train_data=trn, val_data=val, batch_size=32)

# STEP 3: estimate learning rate (a neural network hyperparameter)
learner.lr_find(show_plot=True)

# STEP 4: train model using 1cycle learning rate schedule for 4 epochs
learner.fit_onecycle(2e-5, 4)
```



Arun S. Maiya (amaiya@ida.org) is a member of the research staff in the Information Technology and Systems Division of IDA's Systems and Analyses Center. Arun holds a PhD in computer science from the University of Illinois at Chicago.

Research was supported by IDA's Independent Research Program.