

# Keras Interface for TMVA

Stefan Wunsch

September 29, 2016

# What is Keras?

- ▶ **High level API** for Theano and Tensorflow
- ▶ Seamless switching between the backends
- ▶ **Extends** backend API with predefined layers, initialization techniques and training callbacks

## **Their statement:**

*Being able to go from idea to result with the least possible delay is key to doing good research.*

theano



# Keras Interface for TMVA

## Why using third-party packages in TMVA?

- ▶ **Huge community** which results in **stable software**
- ▶ Field of machine learning **changes rapidly**; hard to keep pace, e.g., with Google Deepmind
- ▶ **Staying up-to-date** without implementing by yourself
- ▶ **High benefit with low maintenance**

 tensorflow / tensorflow	 Watch ▾	3,030	 Unstar	32,260	 Fork	13,915
 fchollet / keras	 Watch ▾	650	 Unstar	8,255	 Fork	2,555
 Theano / Theano	 Watch ▾	435	 Star	4,568	 Fork	1,642

## Task Assignment

- ▶ **TMVA** handles preprocessing, dataloading, validation, testing and comparison to other MVA methods
- ▶ **Keras** provides model definition, training and prediction

## In Detail: Model Definition

- ▶ **Configuration strings** are **not suitable** for complex models
- ▶ Model is defined in **Python** and stored to file (Keras feature)

**Example** Python code (3-layer feed-forward network):

```
model = Sequential()
model.add(Dense(num_hidden_nodes, init='normal',
                activation='relu', W_regularizer=l2,
                input_dim=num_input_nodes))
model.add(Dense(num_output_nodes, init='normal',
                activation='softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer=SGD(lr=0.01), metrics=['accuracy'])

model.save('model.h5')
```

**That's all!**

## In Detail: Import Model in TMVA

- ▶ Book method as usual with `model.h5` as parameter

```
factory->BookMethod(dataloader,  
    TMVA::Types::kPyKeras, "PyKeras",  
    "H:!V:VarTransform=D,N:FilenameModel=model.h5:\  
    NumEpochs=50:BatchSize=128:SaveBestOnly=true:\  
    LearningRateSchedule=30,0.005;40,0.001");
```

**Acceleration**, e.g., for Theano:

- ▶ **GPU**: Just run `export THEANO_FLAGS='device=gpu'`, works out of the box if **CUDA** is installed
- ▶ **CPU**: Use `export THEANO_FLAGS='openmp=True'` to use **OpenMP** (done by default if more than one core is detected)

# Features

- ▶ **Analysis types:** Binary classification, regression and multi-class classification
- ▶ **Training callbacks:** Schedule learning rate, save best model only, continue training
- ▶ **Models:** Everything you can build with Keras (see backup)
- ▶ **Training with sample weights** supported
  
- ▶ **Suggestions** for other features?

Backup

# Keras Features

## Layers

- ▶ Fully connected (Dense)
- ▶ Flatten, reshape, permute, repeat, merge, masking, ...
- ▶ Batch normalization
- ▶ Convolution (1D, 2D, 3D) and zero-padding
- ▶ Pooling (1D, 2D, 3D)
- ▶ Recurrent (LSTM, GRU, RNN)
- ▶ Lambdas (define your own high-level layers)
- ▶ ...

## Regularizations

- ▶ Dropout
- ▶ L1 and L2 for weights, bias and activations
- ▶ Noise



# Keras Features (2)

## Optimizers

- ▶ SGD
- ▶ RMSprop
- ▶ Adagrad
- ▶ Adadelta
- ▶ Adam
- ▶ Adamax
- ▶ Nadam

# Keras Features (3)

## Activations

- ▶ softmax
- ▶ softplus
- ▶ softsign
- ▶ relu
- ▶ tanh
- ▶ sigmoid
- ▶ hard sigmoid
- ▶ linear
- ▶ Advanced activations (LeakyReLU, PReLU, ELU, ParametricSoftplus, SReLU, ThresholdedReLU)