Hands on Keras

Cecilia Jarne

cecilia.jarne@unq.edu.ar

Twitter: @ceciliajarne

CONICET

Universidad Nacional de Quilmes

WTPC workshop en técnicas de programación científica

Necessary Libraries

Available in the following sites:

`https://www.anaconda.com/distribution/`
`https://www.scipy.org/install.html`
`https://www.tensorflow.org/install/`
`https://keras.io/`

Please create a github account:
`https://github.com/`

WTPC workshop en técnicas de programación científica

Hands-On Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow CNTK, or Theano. It was developed with a focus on enabling fast experimentation.
Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two,)and others via complementary projects).
- Runs seamlessly on CPU and GPU.

WTPC workshop en técnicas de programación científica

Core Models: Sequential

The Sequential model is a linear stack of layers. 3 main Steps:

1. Specify the input shape
2. Compile()
3. Fit()

Hands-On Keras

Model compilation It need 3 elements:

- An optimizer. This could be the string identifier of an existing optimizer (such as rmsprop or adagrad), or an instance of the Optimizer class.
- A loss function. This is the objective that the model will try to minimize. It can be the string identifier of an existing loss function (such as categorical_crossentropy or mse), or it can be an objective function.
- A list of metrics.

```
1
2 model.compile(loss='mean_squared_error', optimizer='sgd', metrics=[metrics.mae, metrics.
      categorical_accuracy])
```

WTPC workshop en técnicas de programación científica

Training

Keras models are trained on Numpy arrays of input data and labels. For training a model, you will typically use the fit function.

Hands-On Keras:Leyers

Layer Types: Convolutional layers

- Conv1D
- Conv2D
- Conv3D
  Specially fast Convolutional layers:
  - CuDNNGRU
  - CuDNNLSTM

Hands-On Keras

Layer Types: Recurrent layers

- RNN
- SimpleRNN
- GRU
- LSTM
- ConvLSTM2D
- StackedRNNCells

Losses:

- mean_squared_error
- mean_squared_error(y_true, y_pred)
- mean_absolute_error(y_true, y_pred)
- mean_absolute_percentage_error(y_true, y_pred)
- mean_squared_logarithmic_error(y_true, y_pred)
- categorical_crossentropy(y_true, y_pred)

Metrics

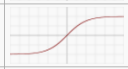
- accuracy
- binary_accuracy
- binary_accuracy(y_true, y_pred)
- categorical_accuracy(y_true, y_pred)
- top_k_categorical_accuracy(y_true, y_pred, k=5)

Optimizers

```
keras.optimizers.SGD(lr=0.01, momentum=0.0, decay=0.0, nesterov=False)
keras.optimizers.RMSprop(lr=0.001, rho=0.9, epsilon=1e-08, decay=0.0)
keras.optimizers.Adagrad(lr=0.01, epsilon=1e-08, decay=0.0)
keras.optimizers.Adadelta(lr=1.0, rho=0.95, epsilon=1e-08, decay=0.0)
keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
```

Hands-On Keras

Activations:

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Example:

```
1 from keras.layers import Activation, Dense
2 model.add(Dense(64))
3 model.add(Activation('tanh'))
```

WTPC Workshop en técnicas de programación científica