



Foto: Thomas Josek

Deep Learning for Language Analysis

Deep Learning Introduction – Technical Introduction

How do I get started with DL?

Caffe



DL4J
Deeplearning4j



MatConvNet

MINERVA

mxnet



theano



Tools and Framework





Tools and Frameworks

Documentation and Code

- > Download: <https://github.com/DH-Cologne/SummerSchool2019>
- > Or Git clone if you have Git installed



Tools and Frameworks

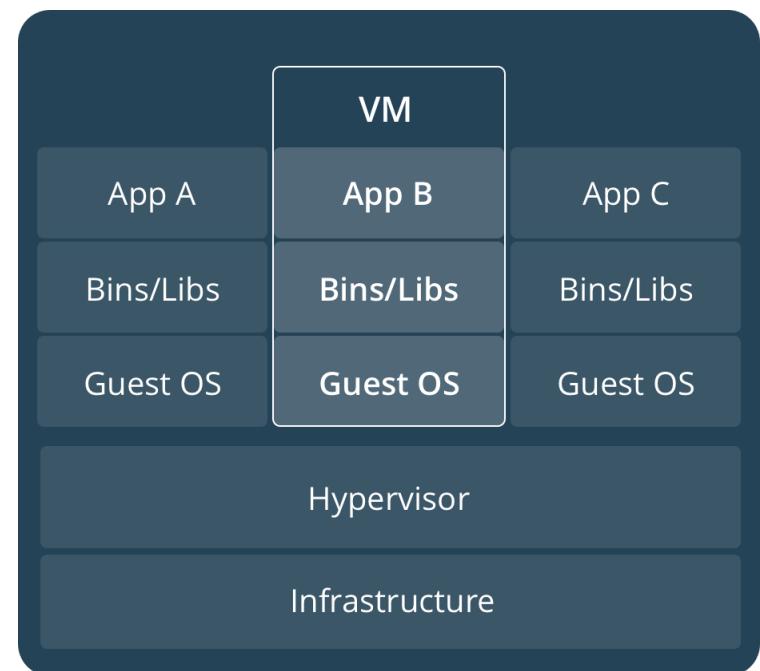
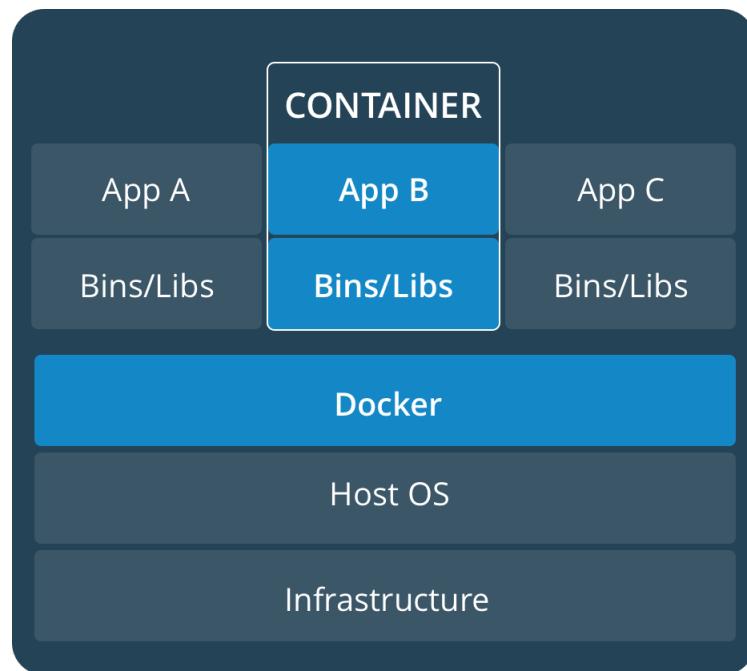
Why Docker? From the Homepage....

- **Flexible:** Even the most complex applications can be containerized
- **Lightweight:** Containers leverage and share host kernel
- **Interchangeable:** Deploy updates and upgrades on-the-fly
- **Portable:** Build locally, deploy to the cloud, and run anywhere
- **Scalable:** Increase and automatically distribute container replicas
- **Stackable:** Stack services vertically on-the-fly

Tools and Frameworks



Container vs. Virtual Machine



Source: <https://docs.docker.com/get-started/>



Tools and Frameworks

Why Jupyter Notebook?

- > Browser based environment for coding
- > Ad hoc coding
- > Learn and try out Python
- > Contains: Live Code, Visualizations, explanatory text

Tools and Frameworks

Why Keras?

- > Quickly train and test model
- > Nice and easy to use API
- > Wraps multiple frameworks
- > (Simplified) interface to Theano, CNTK or TensorFlow
- > TensorFlow is default API

Pipeline

1. Define Network
2. Compile it
3. Fit it
4. Evaluate it
5. Make Predictions

How do I build a Neural Network in Keras?



Define Network

- > Create an instance of the sequential class
- > Define sequence of layers
- > Add new lines (each line is a new layer)
- > First Layer: number of inputs (can differ depending to the network type)

```
model = Sequential()  
  
model.add(Dense(32, activation='relu', input_shape=(784,)))  
model.add(Dense(10, activation='softmax'))  
  
model.summary()
```

How do I build a Neural Network in Keras?



Compile it

- > Transforms a simple sequence of layers into a highly efficient series of matrix transforms
- > Intended to be executed on the GPU (depending on the configuration set)
- > Optimization: Train the network
- > Loss Function: Evaluate the network

```
# For a multi-class classification problem
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# For a binary classification problem
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# For a mean squared error regression problem
model.compile(optimizer='rmsprop',
              loss='mse')
```

How do I build a Neural Network in Keras?



Fit Network

- > Adapting the weights of the training data set
- > Input X => Matrix of input patterns
- > Output Y => Array of matching output patterns

```
1 X = count_vectorizer.fit_transform(traindf['text'])
```

```
1 label_map = {'HillaryClinton':1, 'realDonaldTrump':-1}
2 Y = list(traindf['handle'].apply(lambda x: label_map[x]))
```

```
1 print(X.shape)
2 print(len(Y))
```

(5600, 49)
5600

```
1 model.fit(X, Y, batch_size=32, epochs=10)
```

How do I build a Neural Network in Keras?



Evaluate Network and use it to make predictions

- > How good does model work? Evaluate it!
 - > Input X => Matrix of input patterns (test data)
 - > Output Y => Array of matching output patterns (test data)

```
: model.evaluate(  
    data,  
    labels,  
    batch_size=32,  
    verbose=1,  
    sample_weight=None)  
  
32/1000 [........................] - ETA: 0s  
: [0.69383435416221617, 0.5390000000000003]
```

```
: model.predict(  
    data,  
    batch_size=32,  
    verbose=1)  
  
32/1000 [........................] - ETA: 0s
```