# LSTM TOOLKITS

## … or how to get started

**Thomas Breuel**
**University of Kaiserslautern**

# how can you apply LSTM networks?

- **RNNLIB**
- **rnnscript**
- **OCRopus**
- **PyBrain**
- **JANN**
- **lstmlib**
- **lstm-g**
- **(possibly others)**

# things to ask

- **programming language**
- **extensibility**
- **license**
- **matrix-based / connection-based**

# RNNLIB

- C++, command line
- datasets in NetCDF format
- reference implementation
- full MDRNN
- training is fast but memory intensive
- not designed for use as library
- compiler-related problems in some versions
- open source

# rnnscript

- **RNNLIB bound to Python**
- **same numerical code as RNNLIB**
- **can be trained incrementally (huge datasets)**
- **otherwise, similar to RNNLIB**
- **new network components cannot be implemented in Python**
- **some features not bound (yet)**
- **used for animations, testing, evaluations**

# OCRopus LSTM implementation

- pure Python implementation of LSTM
- (some native code being added for speedup)
- LSTM, BDLSTM, CTC
- no: MDLSTM, sequence classification
- extensible in Python
- small, compact
- being broken out into a separate library
- matrix-based abstractions, close to math
- open source, Apache license

# PyBrain

- general purpose machine learning library with focus on neural networks
- implements LSTM networks, reinforcement learning, SRNN, many other features
- optional native code speedup (no experience)
- fairly slow in our benchmarks, no CTC
- BSD license

# JANNlab

- Java-based artificial neural network library
- implements LSTM, among other layer types
- connection/unit-based
- 1D LSTM, BLSTM
- no: MDRNN, CTC
- extensible in Java
- GPL3 license

# observations

- **we need a clean, efficient C++ implementation of LSTM and supporting classes**
- **matrix-based in order to take advantage of linear algebra tools and libraries**

- **optional Python bindings**

# lstmlib (in the works)

- **C++ based, using Armadillo linear algebra**
- **closely follows the OCRopus Python implementation**
- **full featured: BLSTM, MDLSTM, CTC, ...**
- **similar Python bindings**
- **will be open source**
- **intended to be much easier to embed and extend than RNNLIB**