Wildcard
Pharmaceutical Consulting

RDKit UGM 2017

SMILES enumeration as Data Augmentation for Neural Network Modeling of Molecules

Outline

- Short intro to presenter
- Neural Nets 0 to 120 mph
- LSTM Recurrent Neural Networks
- Data Augmentation
- SMILES enumeration
- DHFR example
- Tox21 shootout
- Conclusion

Esben Jannik Bjerrum

- Ph.D (Computational Chemistry), University of Copenhagen (DK)
- Industry experience:
  - Drug Discovery IT Support and Databases, LEO Pharma A/S (DK)
- Postdoc #1
  - Protein NMR, Department of Biology, Copenhagen University (DK)
- Postdoc #2
  - Chemometrics and automatic PLS model tuning, Department of Food, Copenhagen University (DK)
- Wildcard Pharmaceutical Consulting

Wildcard
Pharmaceutical Consulting

Molecular data science and machine learning

Computational Chemistry

Molecular Machine Learning

Research and Laboratory IT services

Are Neural Networks the future?

- My Previous experience:
  - Slow to train
  - Prone to over fitting
  - Complex (many hyper parameters)
  - Not as good and robust predictions as simpler models
- But recently they gained momentum (again-again)
  - GPU computing
    - Theano and Tensorflow
  - Better weight initialization
  - Activation functions (Relu)
  - Regularization (Dropout, noise, weight regularization)
  - Advances in network architectures: CNN's, RNN's (LSTM, GRU)
  - Larger amounts of data

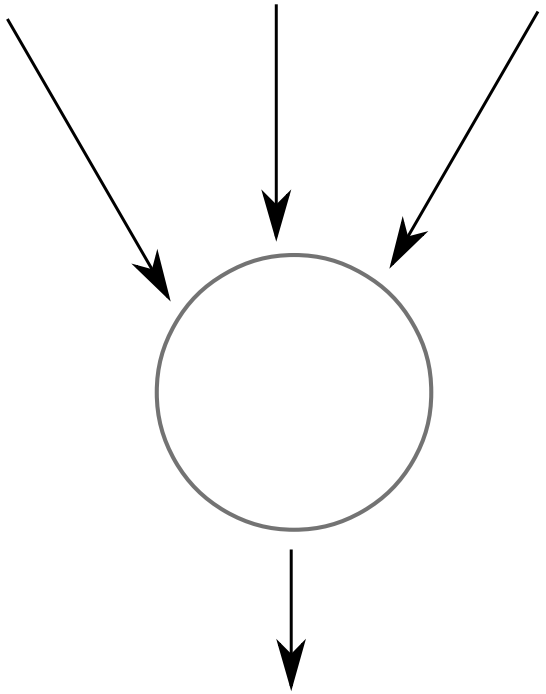
A spectrum of Machine Learning Tasks

- Statistical End
  - Low number of samples
  - "Simple" model can capture variation
  - Main problem is to separate noise from signal
- MLR, PLS, SVM

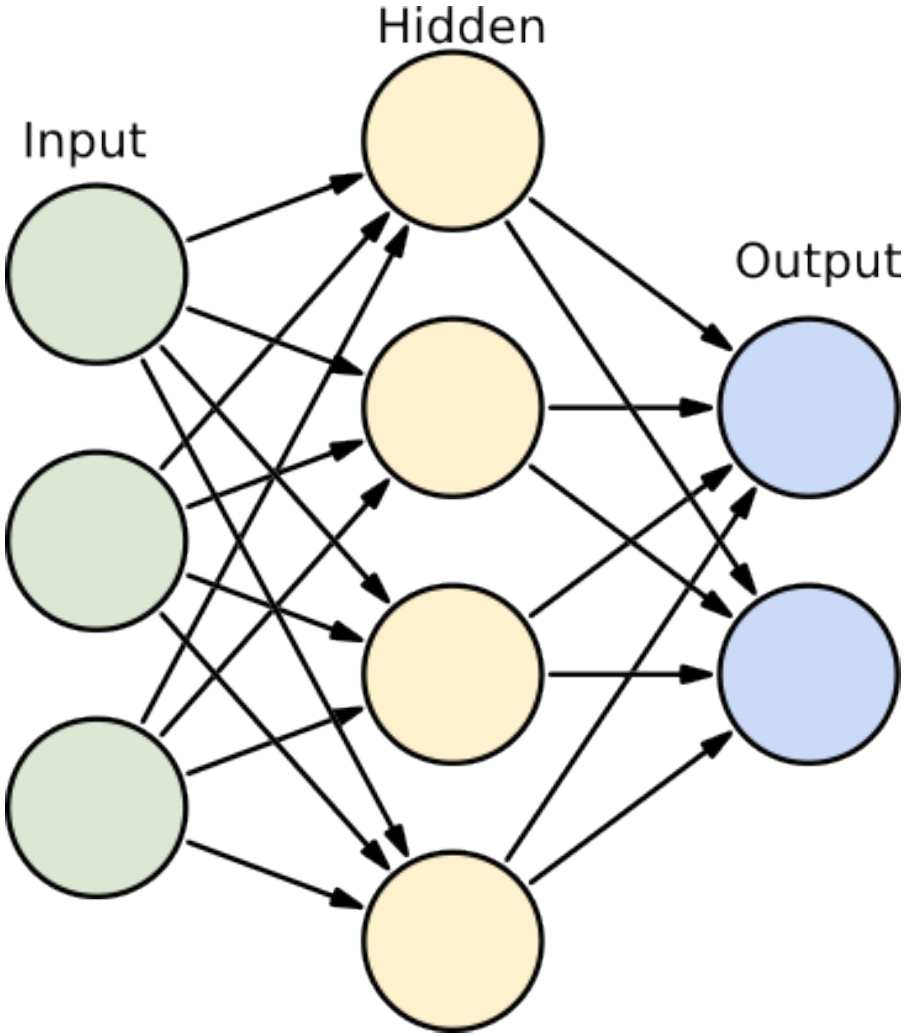
- AI End
  - High dimensional data
  - Complicated model: Example Link picture to label
  - Main problem: How to represent and model data
- Use Neural nets and let back propagation figure it out.

Artificial Neurons

Input	0.1	1	-0.9
weights	20	-1	-2
Weighted sum	20*0.1 + -1*1 + -2*-0.9 = 2.8		
Activation Function	tanh(6) = 0.992		
Output	0.992		

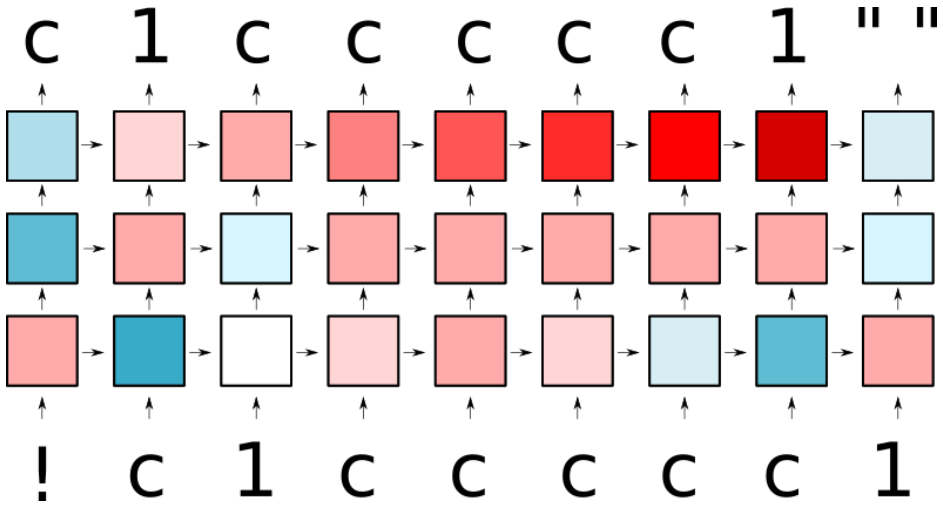


Feed Forward Neural Networks



- Fixed size input
- One or more hidden layers

Recurrent Neural Networks



- Sequences of features as inputs
- The same task for every element of a sequence, with the output being affected by the previous computations
- Modeling of sequences such as text, tweets, time series etc.

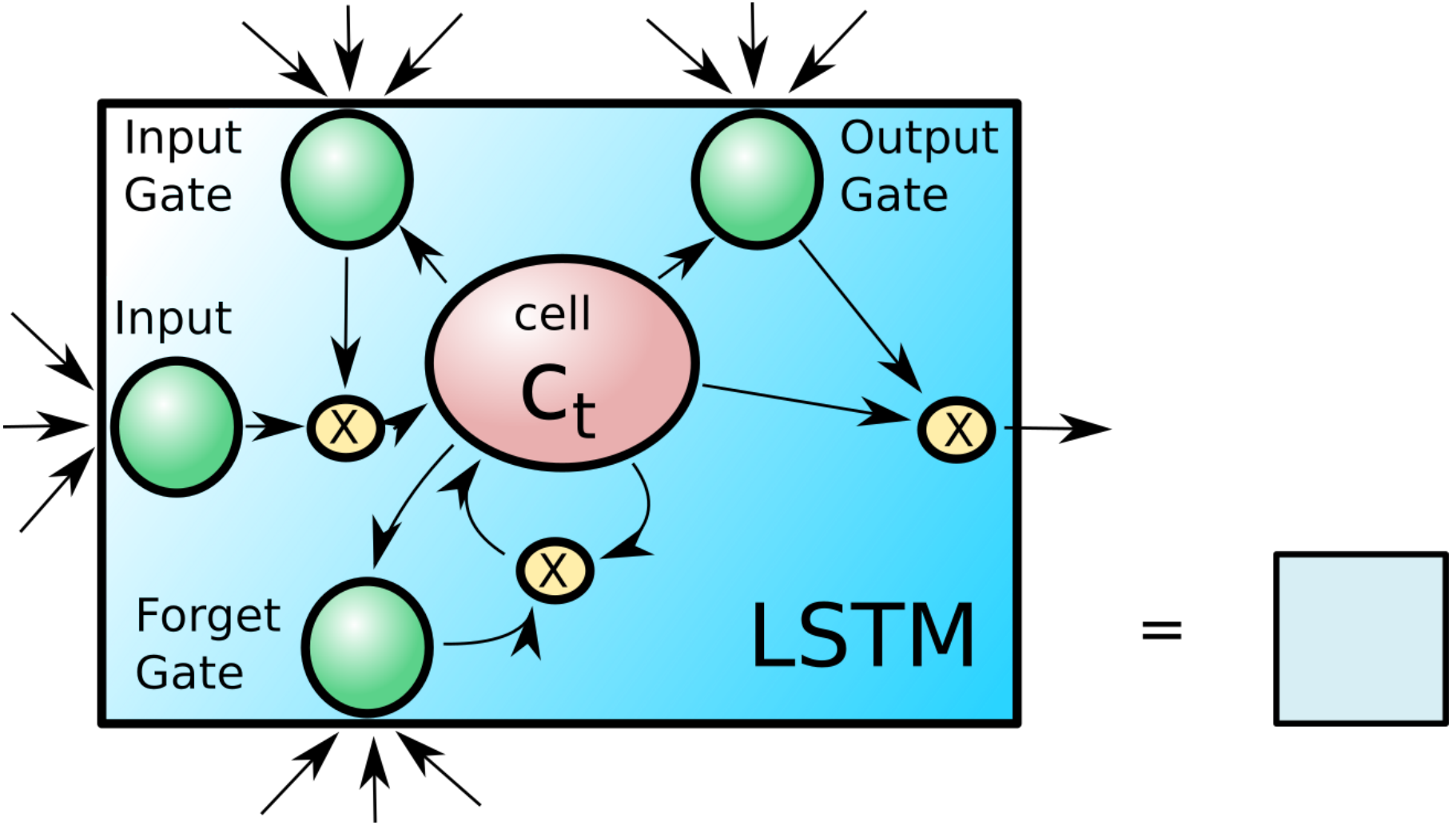
SMILES Notation

- **S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification
- Linear molecular notation format
- Atoms as Letters (lowercase aromatics)
- Special bonds = # : -
- Branching ( )
- Rings via numbering: c1ccccc1
- [Hydrogenation and charge]
- cis-trans isomery / \
- R/S Stereochemistry @ @@

Idea

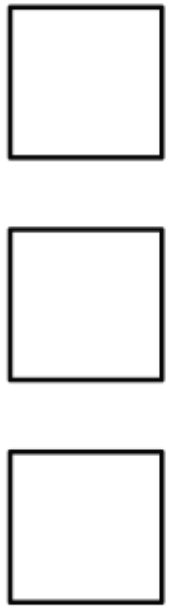
- Recurrent neural networks are good at modeling sequences such as text
- SMILES is a linear text molecular representation
- ….
- ….
- Must … try … to … combine!

LSTM-architecture



LSTM RNN in Action

LSTM RNN in Action

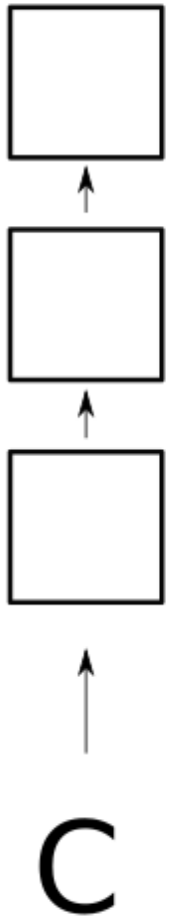


[1,0,0,0]

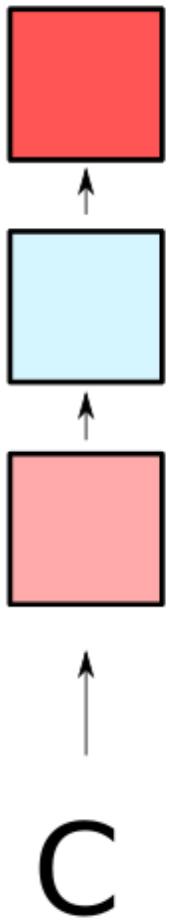
C

LSTM RNN in Action

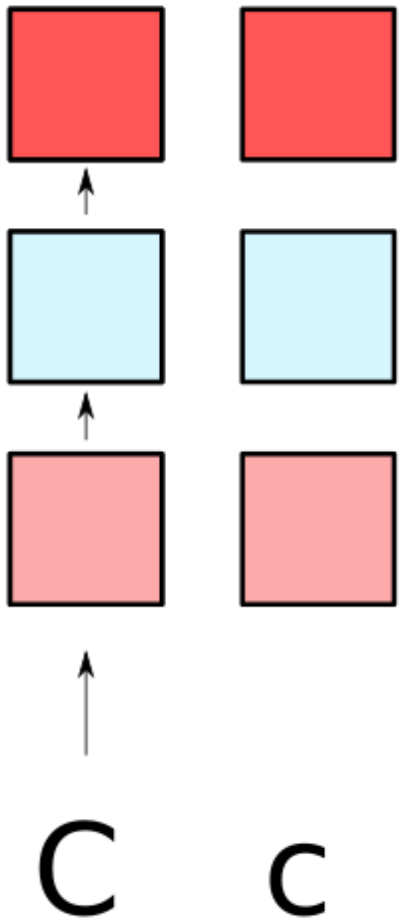
LSTM RNN in Action



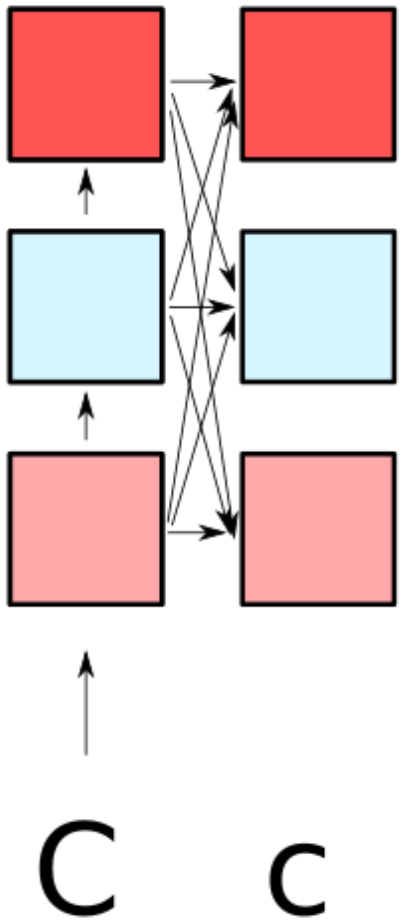
LSTM RNN in Action



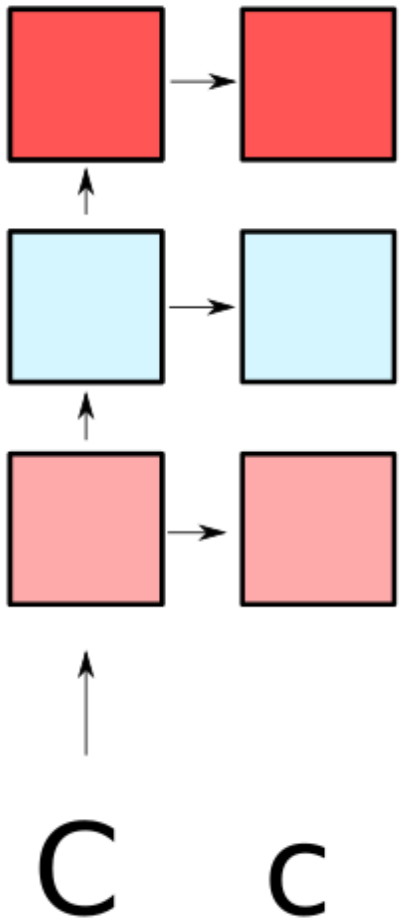
LSTM RNN in Action



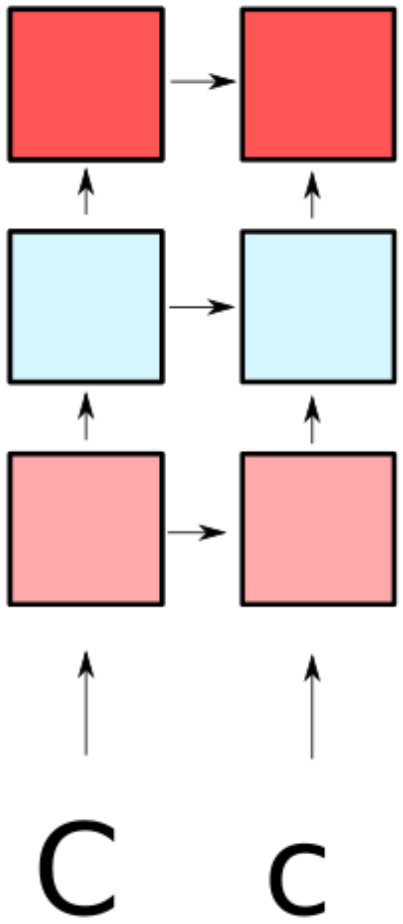
LSTM RNN in Action



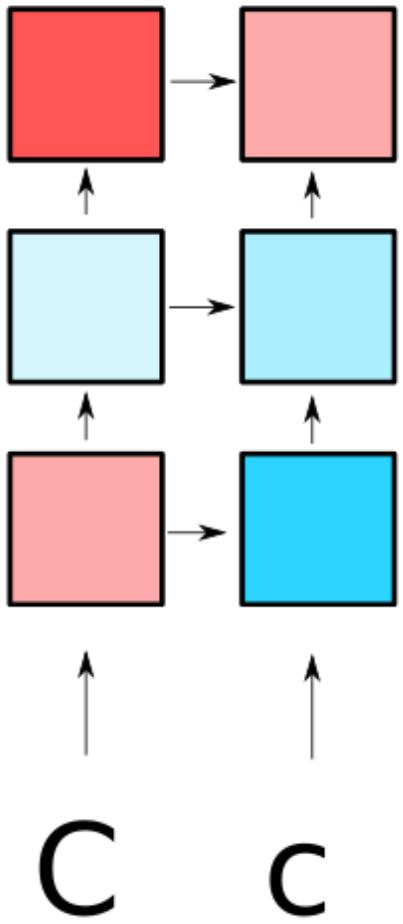
LSTM RNN in Action



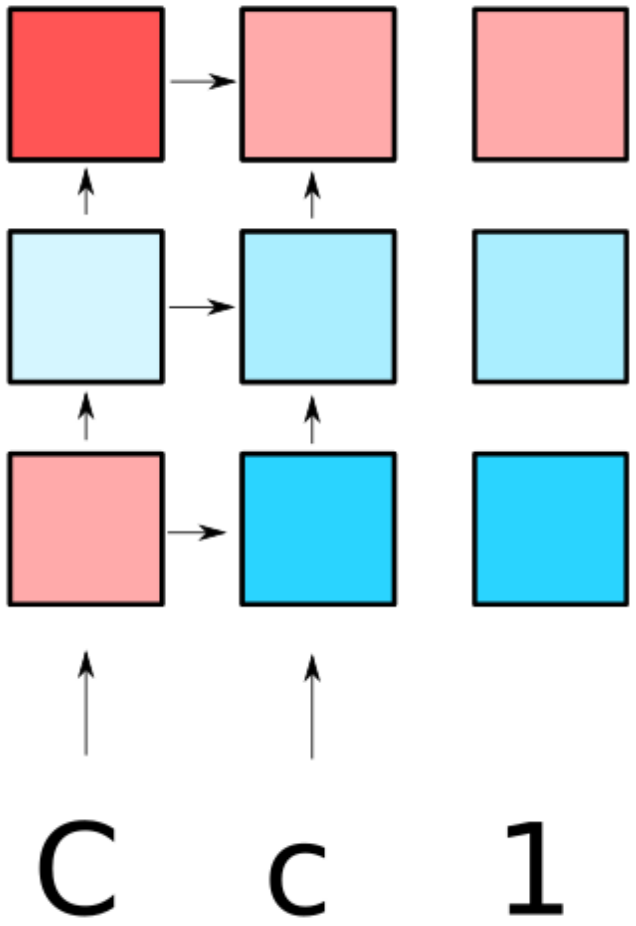
LSTM RNN in Action



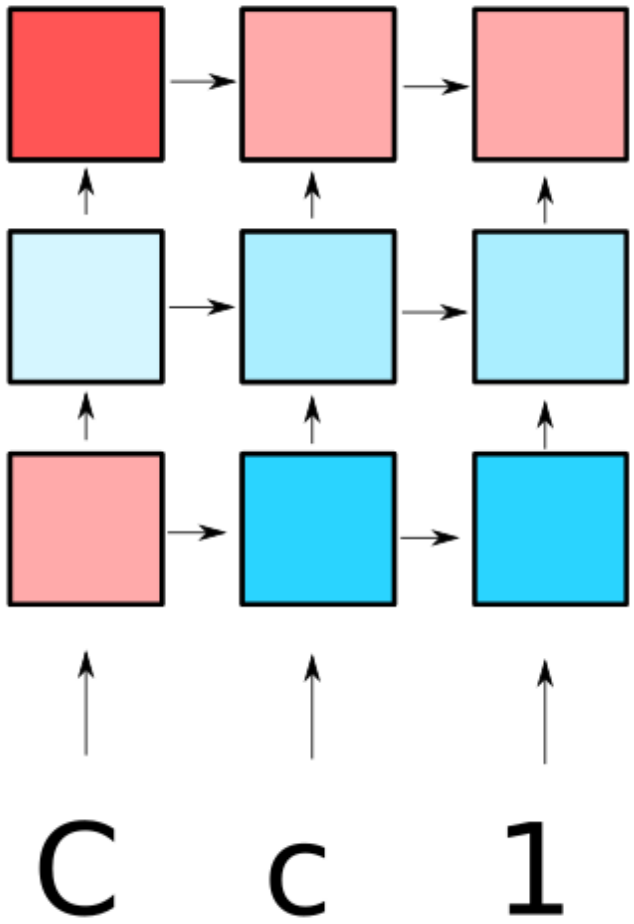
LSTM RNN in Action



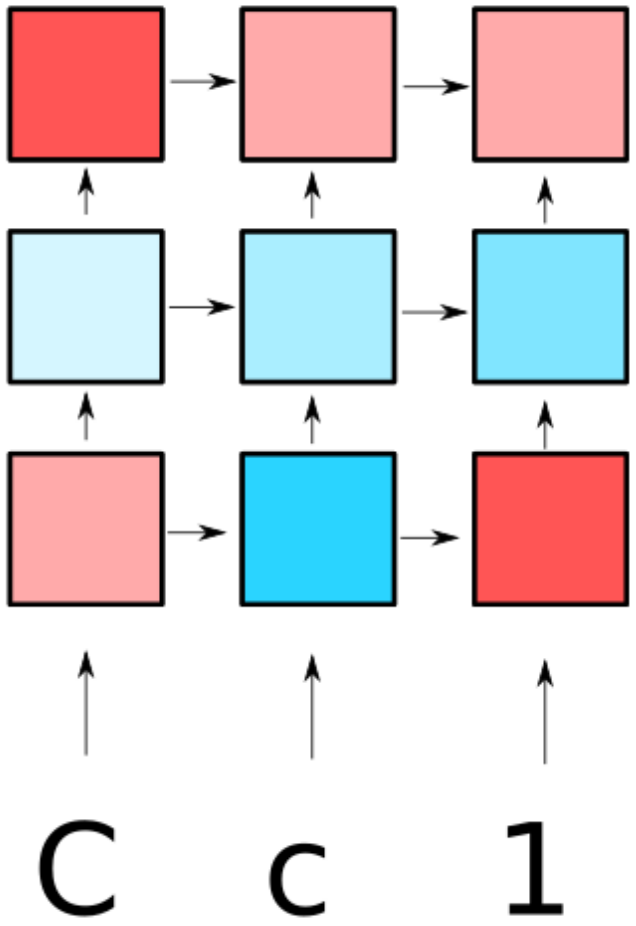
LSTM RNN in Action



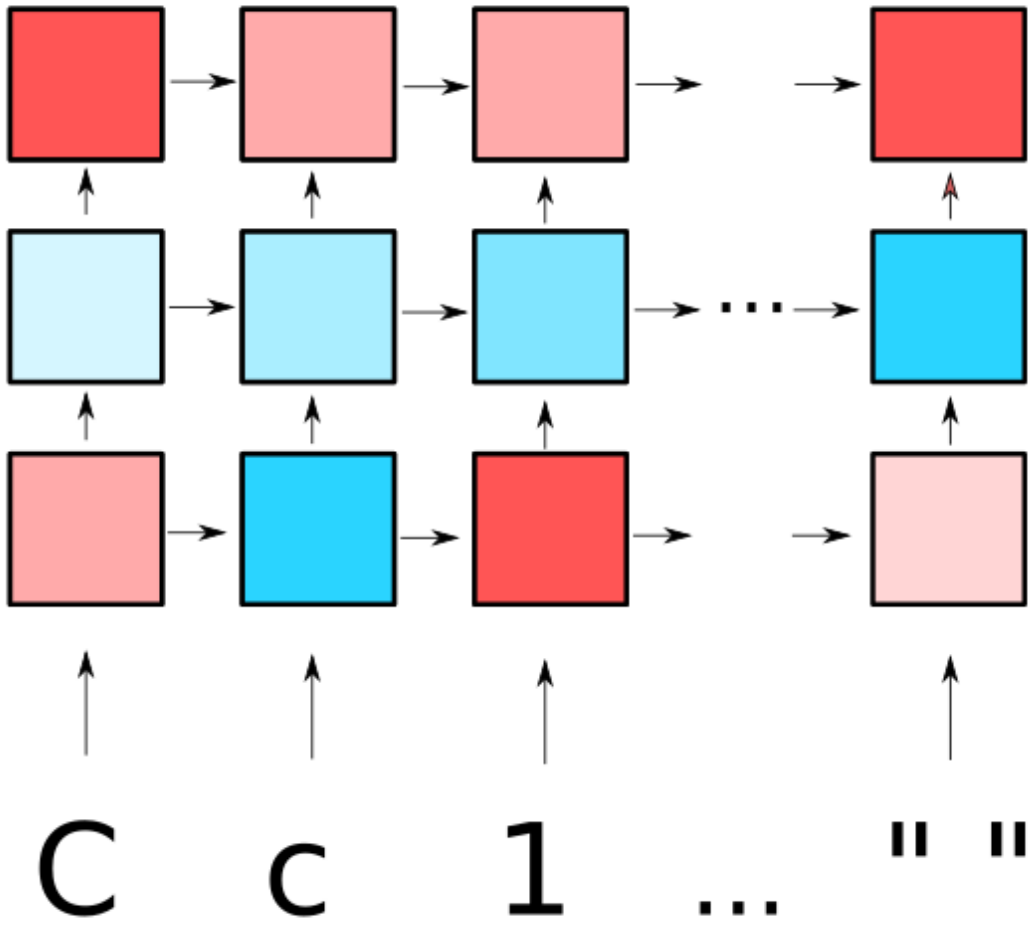
LSTM RNN in Action



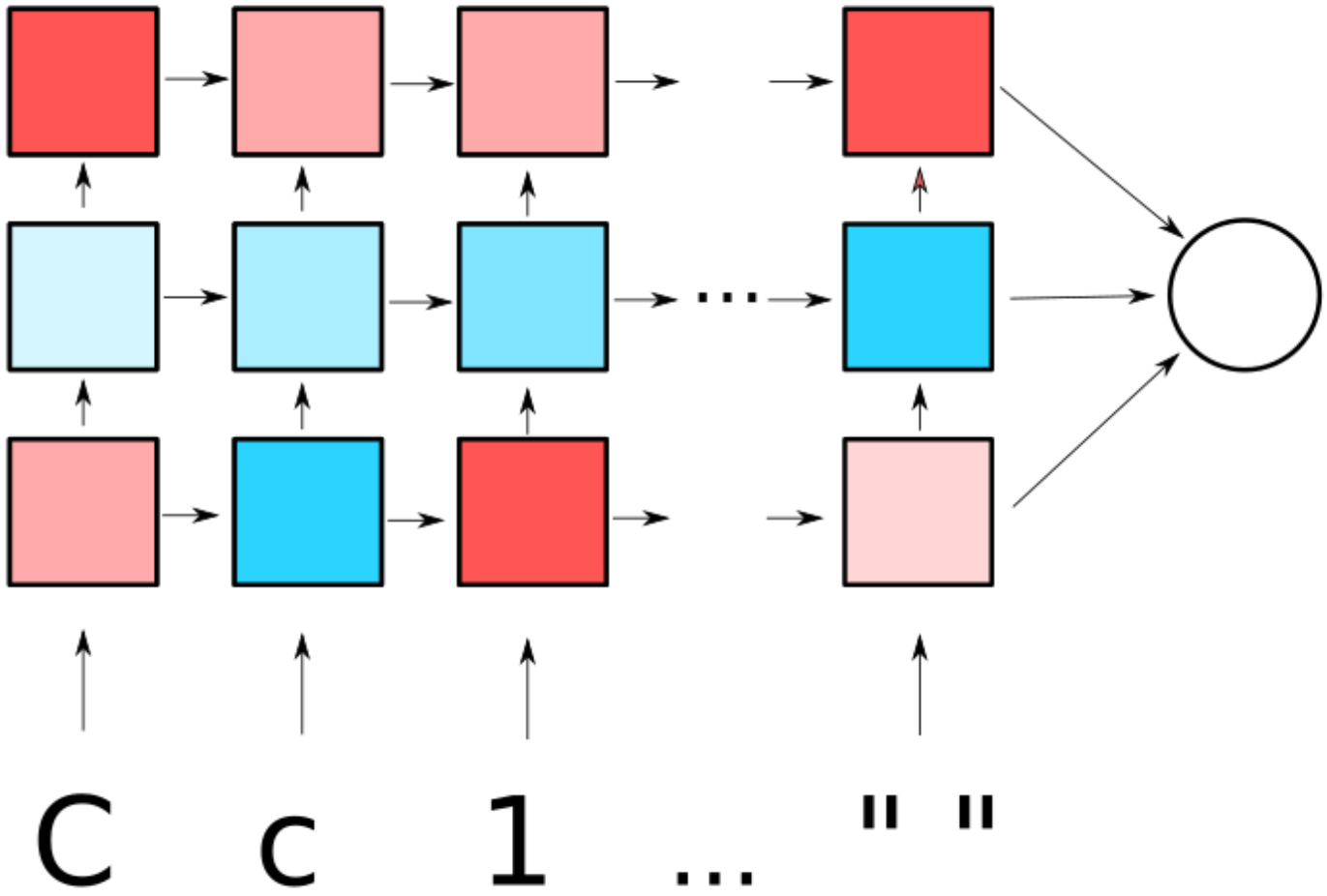
LSTM RNN in Action



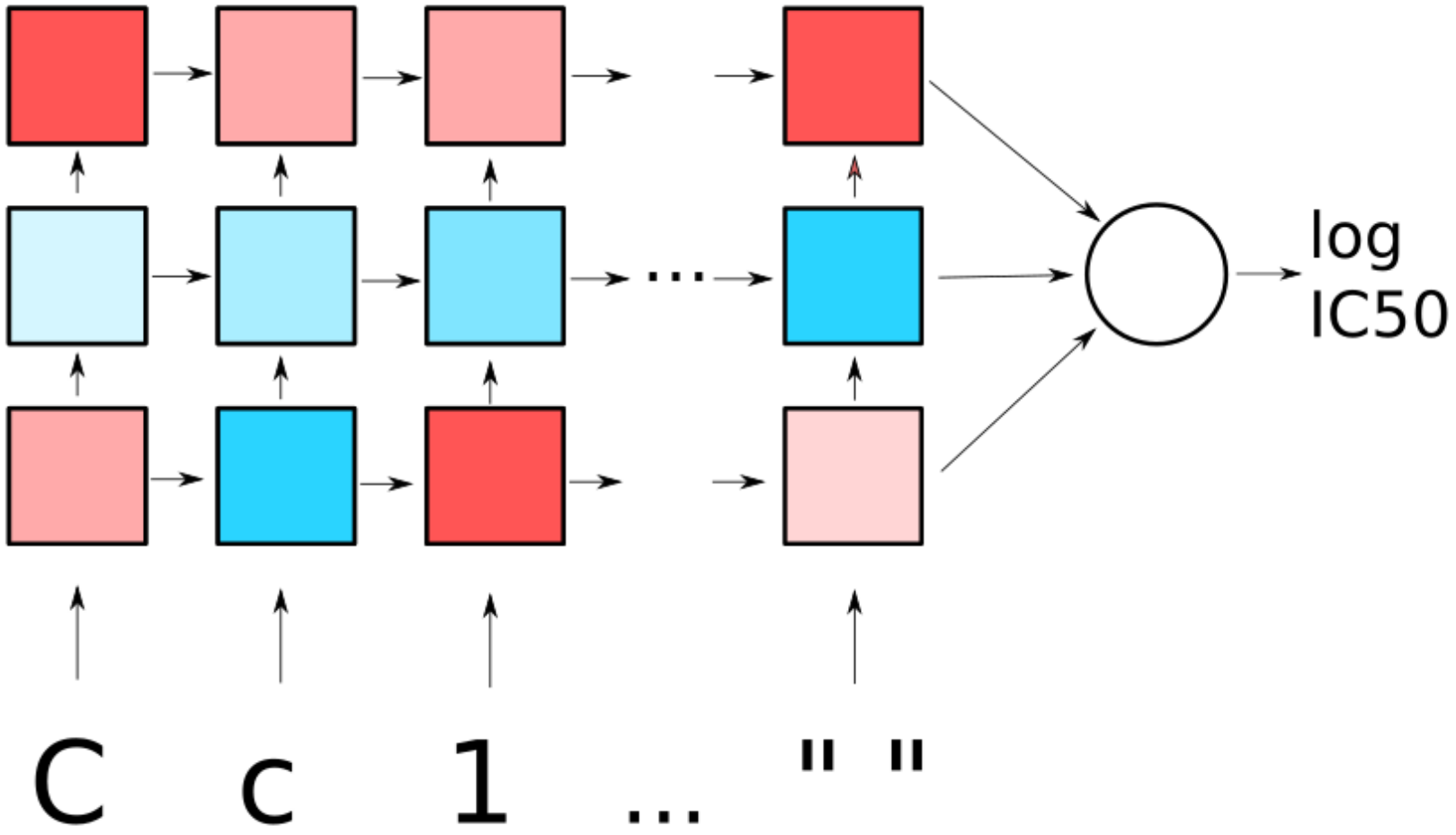
LSTM RNN in Action



LSTM RNN in Action



LSTM RNN in Action



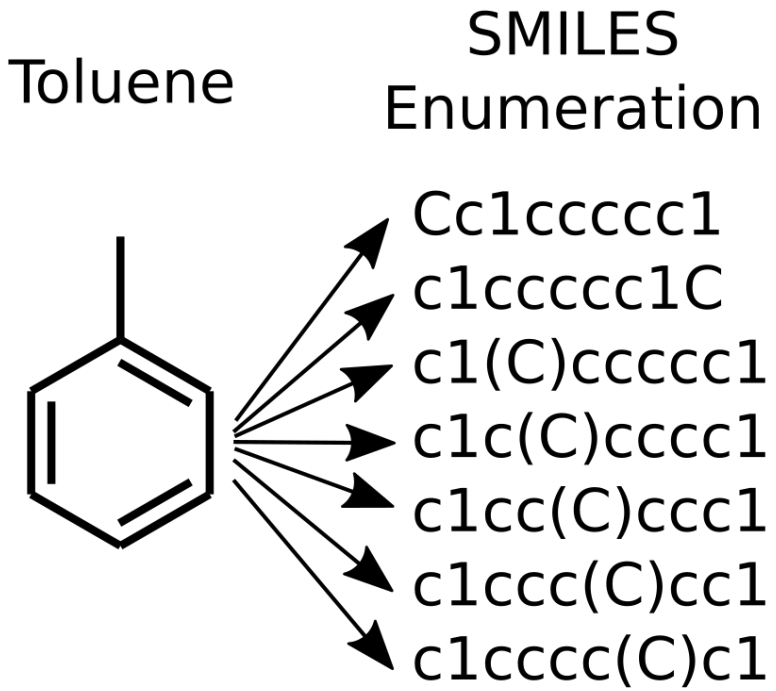
But we need a lot of DATA!

Data Augmentation

- A cat is a cat is a cat...
- Modified pictures with a recognizable cat
- Zooming, cropping, mirroring, flipping, rotation, hue, color, contrast, etc. + combinations
- => Neural network gets more training examples from same number of labeled pictures

SMILES Enumeration

- The serialization of molecules is ambiguous
- Canonical SMILES ensures a 1:1 relationship between molecule and SMILES
- I go the other way and generate multiple SMILES for the same molecule
- Works as data augmentation?



RDKit Code 1

```
from rdkit import Chem
from randomizemolblock import RandomizeMolBlock

def RandomizeMol(mol):
    """Function that randomizes an RDKit mol by round tripping a molblock"""
    mb = Chem.MolToMolBlock(mol)
    mb = RandomizeMolBlock(mb)
    return Chem.MolFromMolBlock(mb)

def randomize_smile(smile):
    """randomize a SMILES"""
    mol = RandomizeMol(Chem.MolFromSmiles(smile))
    return Chem.MolToSmiles(mol, canonical=False)
```

RDKit Code 2

```
def get_mol_set(smile, tries=10000, split=True):
    """Make a set of unique randomized SMILES"""
    s = set()
    canonical = Chem.MolToSmiles(Chem.MolFromSmiles(smile))
    s.add(canonical)
    for i in range(tries):
        s.add(randomize_smile(smile))
    print "total %s found"%(len(s))
    if split:
        s.remove(canonical)
        return canonical, s
    else:
        return s
```

Free NN Tools and High level API's

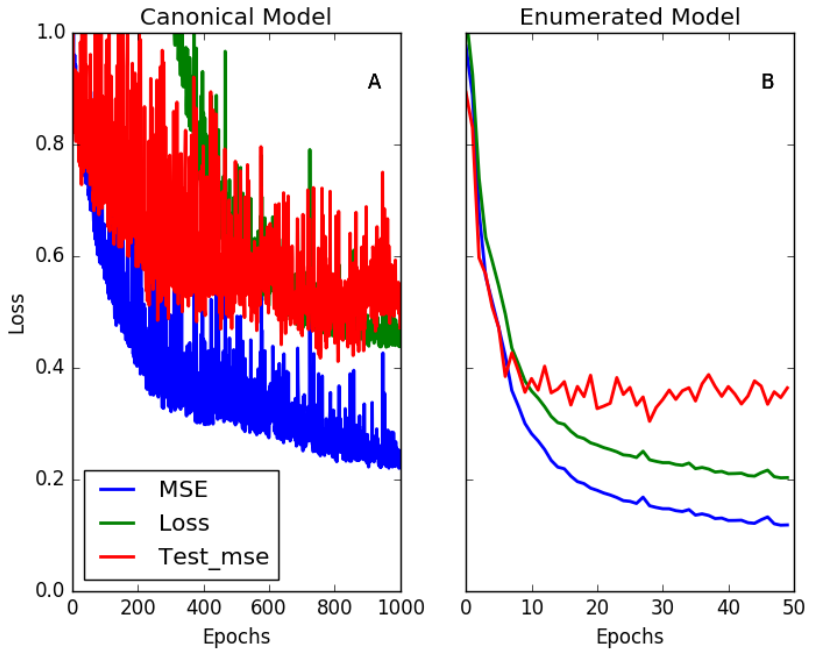
- Python Tensor libraries for GPU computing
  - Theano
  - TensorFlow (Google)
- High Level Python Api's
  - Keras (Now part of TensorFlow)

Example Keras Code

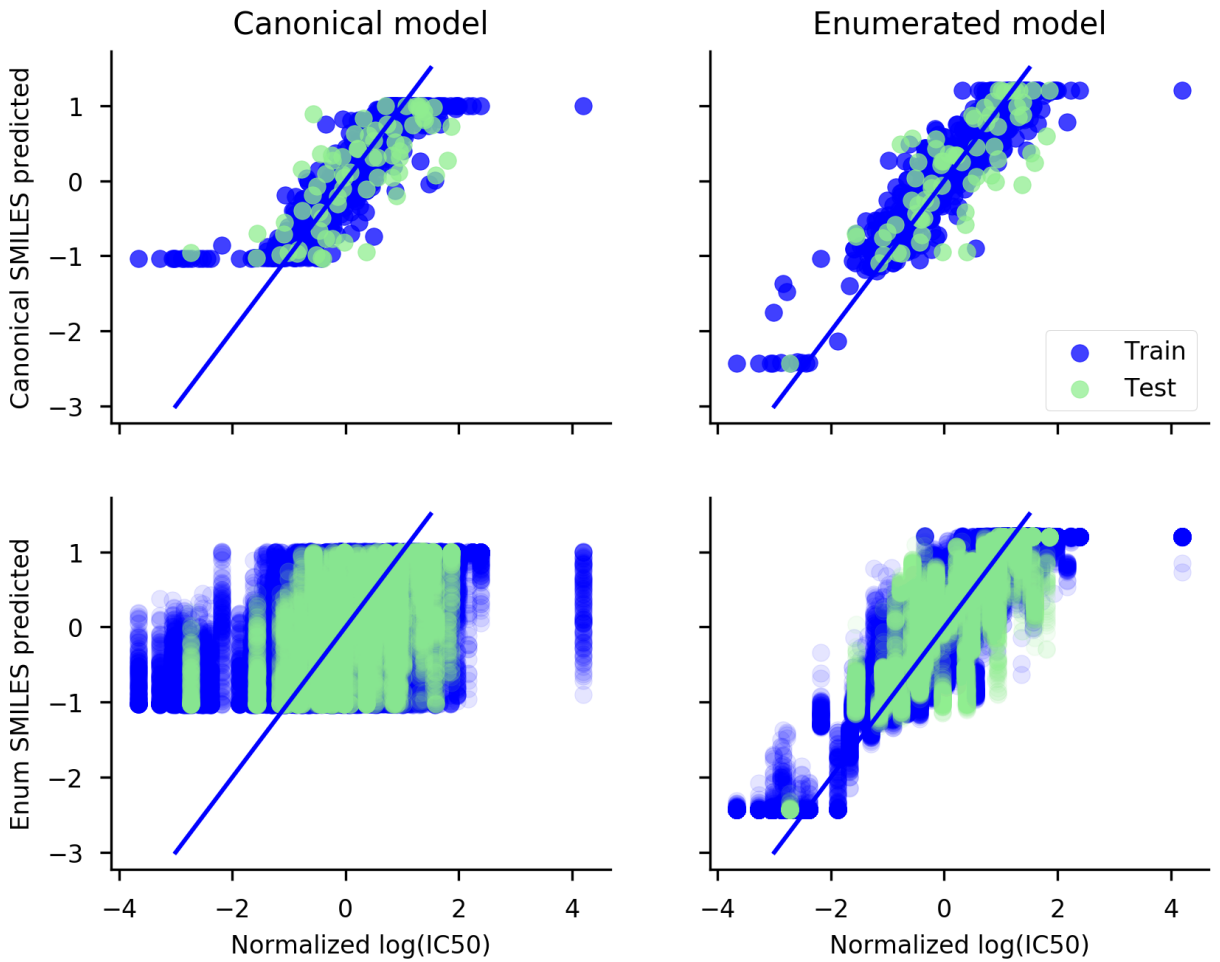
```
#Define model

model = Sequential()

model.add(LSTM(64, input_shape=(X.shape[1], X.shape[2]), dropout_W =
0.19))

wr2 = WeightRegularizer(l2 = 0.01, l1 = 0.005)

model.add(Dense(y.shape[0], activation = "linear",W_regularizer=wr2))


#Compile model to GPU/CPU code

model.compile(loss='mse', optimizer='RMSprop')


#Train

train_history= model.fit(X_train,y_train, nb_epoch=50, batch_size=800,
validation_data=(X_test,y_test))
```

DHFR Test of SMILES Enumeration

- 756 dihydrofolate inhibitors with *P. carinii* DHFR inhibition data.
- Canonical dataset: **602** train and **71** test molecules
- Enumerated dataset: **79143** and **9412** SMILES
- Enumeration made tuning and training easier
- Enumeration gave lower loss and R^2 on test set

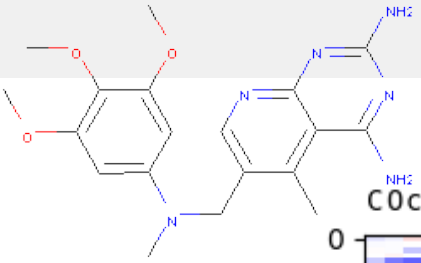


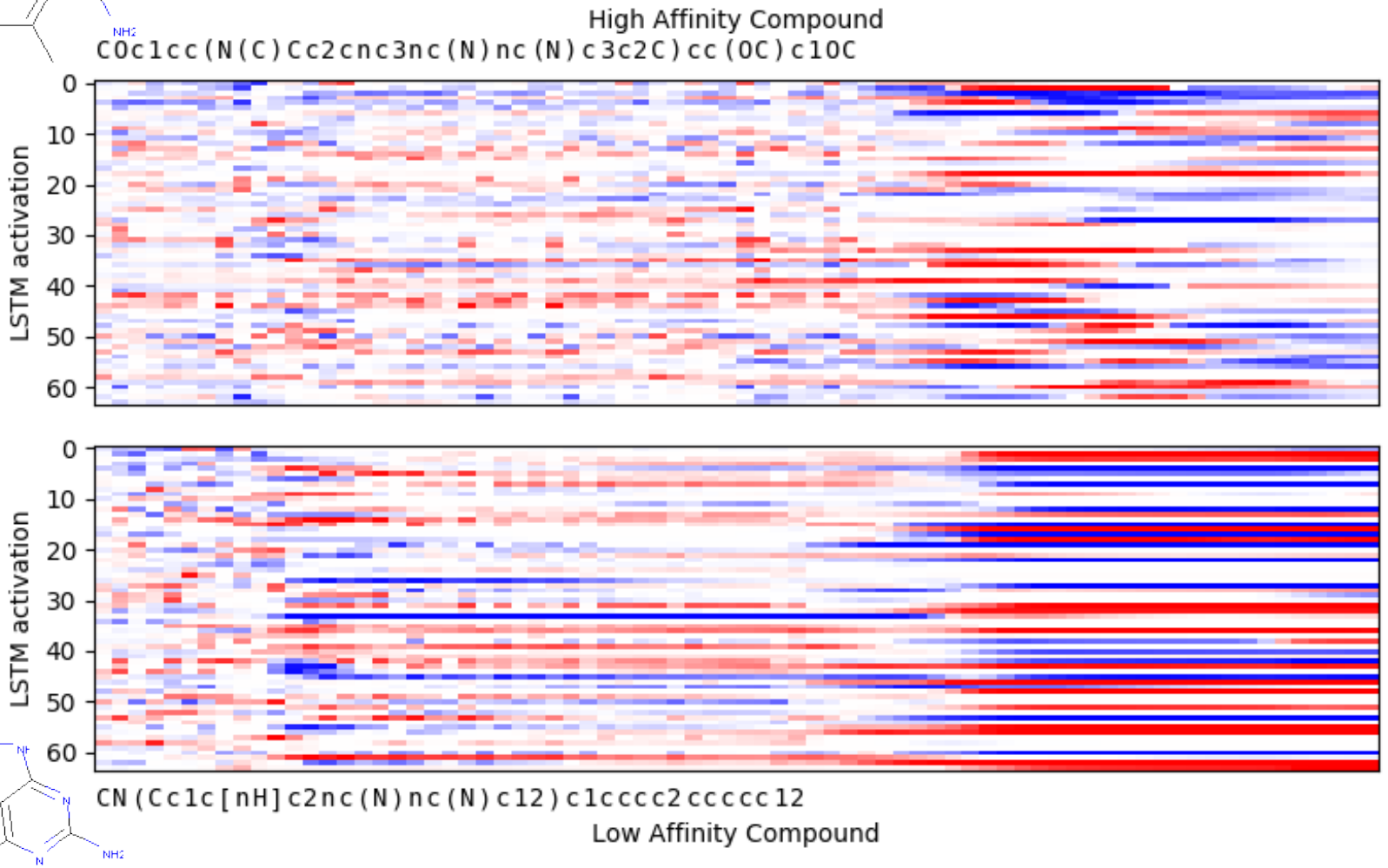
Comparison of Models

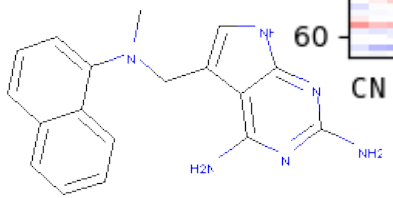


- E. J. Bjerrum. “SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules” https://arxiv.org/abs/1703.07076

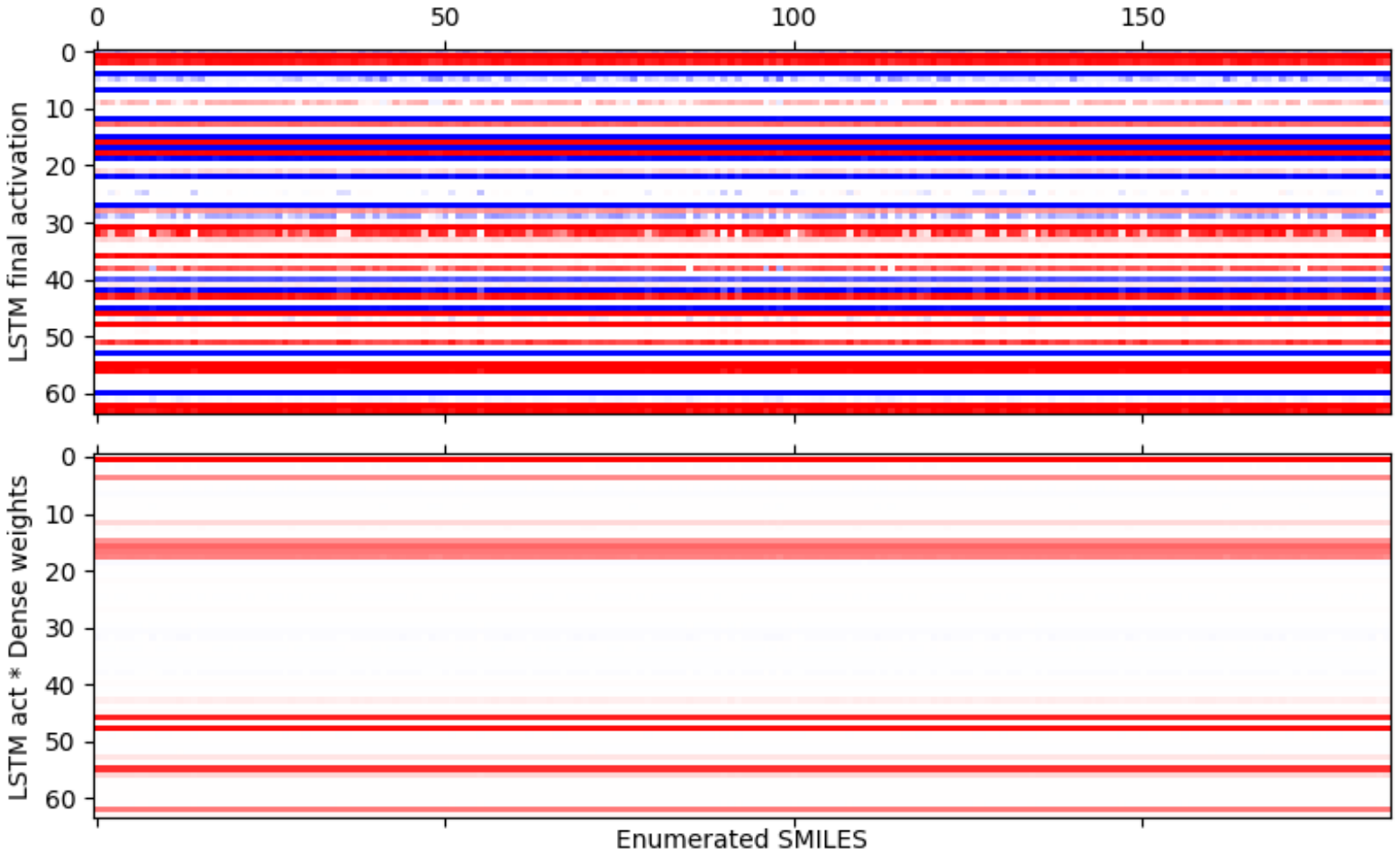
LSTM Activation







Final Vector Variability



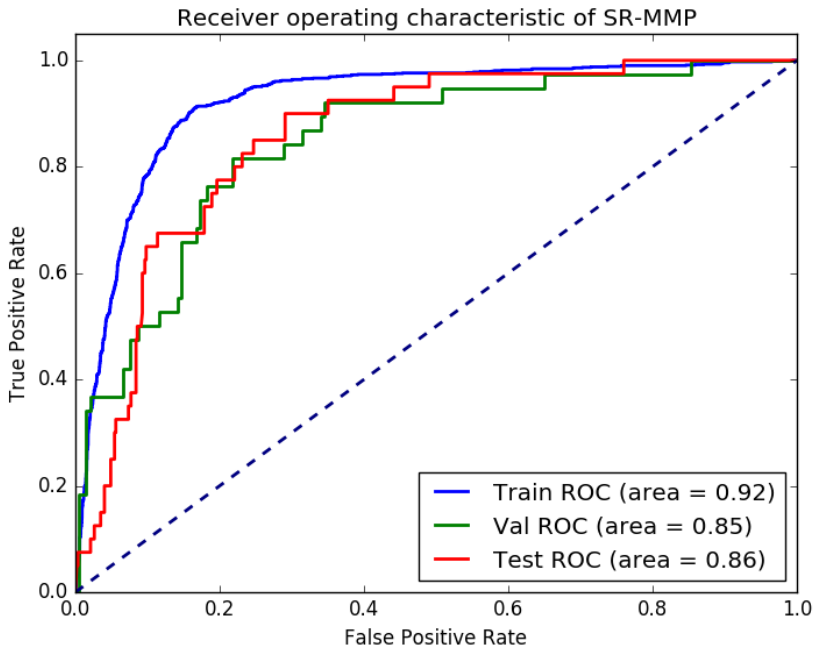
TOX21 Dataset Shootout

- LSTM-NN trained on Tox21 dataset
- 11 thousand compounds, 12 assays
- Co-modeling of assay results: 98295 data points
- Over 700.000 augmented samples

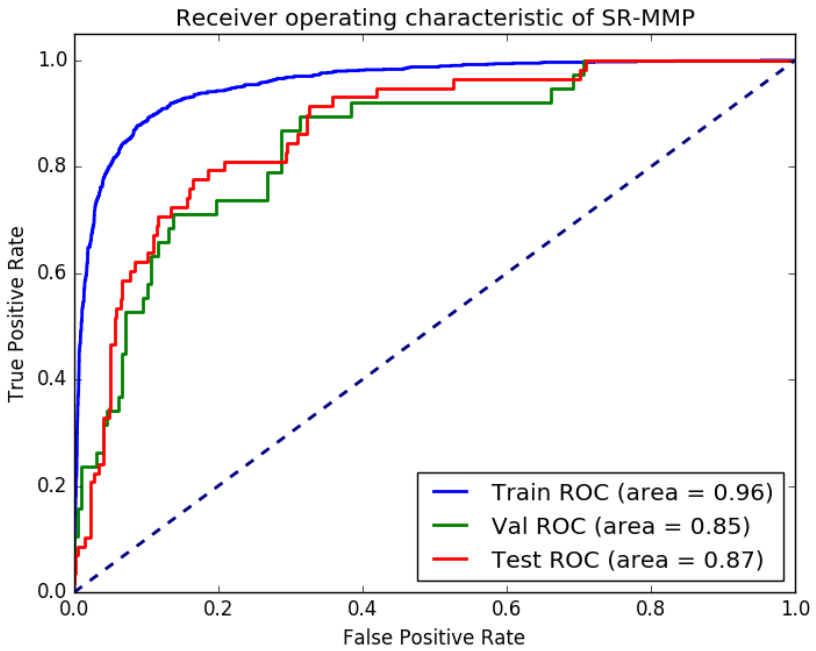
- Feed forward neural network
- Morgan Fingerprints as molecular representation
- Both models hyper parameters were optimized with Gaussian processes

Performance is on par

Smiles LSTM-DNN



FP DNN



Auc Val 0.85, single DNN with MT learning
DeepTox: Toxicity Prediction using
Deep Learning: doi: 10.3389/fenvs.2015.00080

LSTM in QSAR

- Enables input of smiles strings directly to QSAR model
- Slow to train
- Needs “large” datasets
- Data augmentation helps
- On par with optimized FFNN models with FP/descriptors as input

- Future investigation
  - Test more datasets
  - How to best co-model data types
  - Could be merged with other types of input (e.g. FP, descriptors, Assay results)
  - Can final vector be used for other purposes?

Conclusions

- Neural Networks can be used to handle unusual tasks
- Spill over effects from research in other areas
- Good, free, high level API's are available
- LSTM/GRU suitable for SMILES input and output
- Flexible Integration
  - Co-modeling of outputs
  - Merging of inputs
  - Feedback to/from work flows
- Consider the nature of the problem (dataset size, complexity, wanted outcome), NN's may not be the optimal choice
- Many hyper parameters to tune as well as architecture choice
  - Gaussian Process Optimization
- I'm quite enthusiastic :-)

Thank you

- Looking for research collaborations
- Questions/Contact: esben@wildcardconsulting.dk
- linkedin.com/in/esbenbjerrum
- Science blog and more:

www.wildcardconsulting.dk/ useful-information