Multilabel Classification and Deep Learning

Zachary Chase Lipton

Critical Review of RNNs:
http://arxiv.org/abs/1506.00019

Learning to Diagnose:
http://arxiv.org/abs/1511.03677

Conditional Generative RNNS:
http://arxiv.org/abs/1511.03683

Outline

- **Introduction to Multilabel Learning**
- Evaluation
- Efficient Learning & Sparse Models
- Deep Learning for Multilabel Classification
- Classifying Multilabel Time Series with RNNs

Supervised Learning

- General problem, desire a labeling function

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

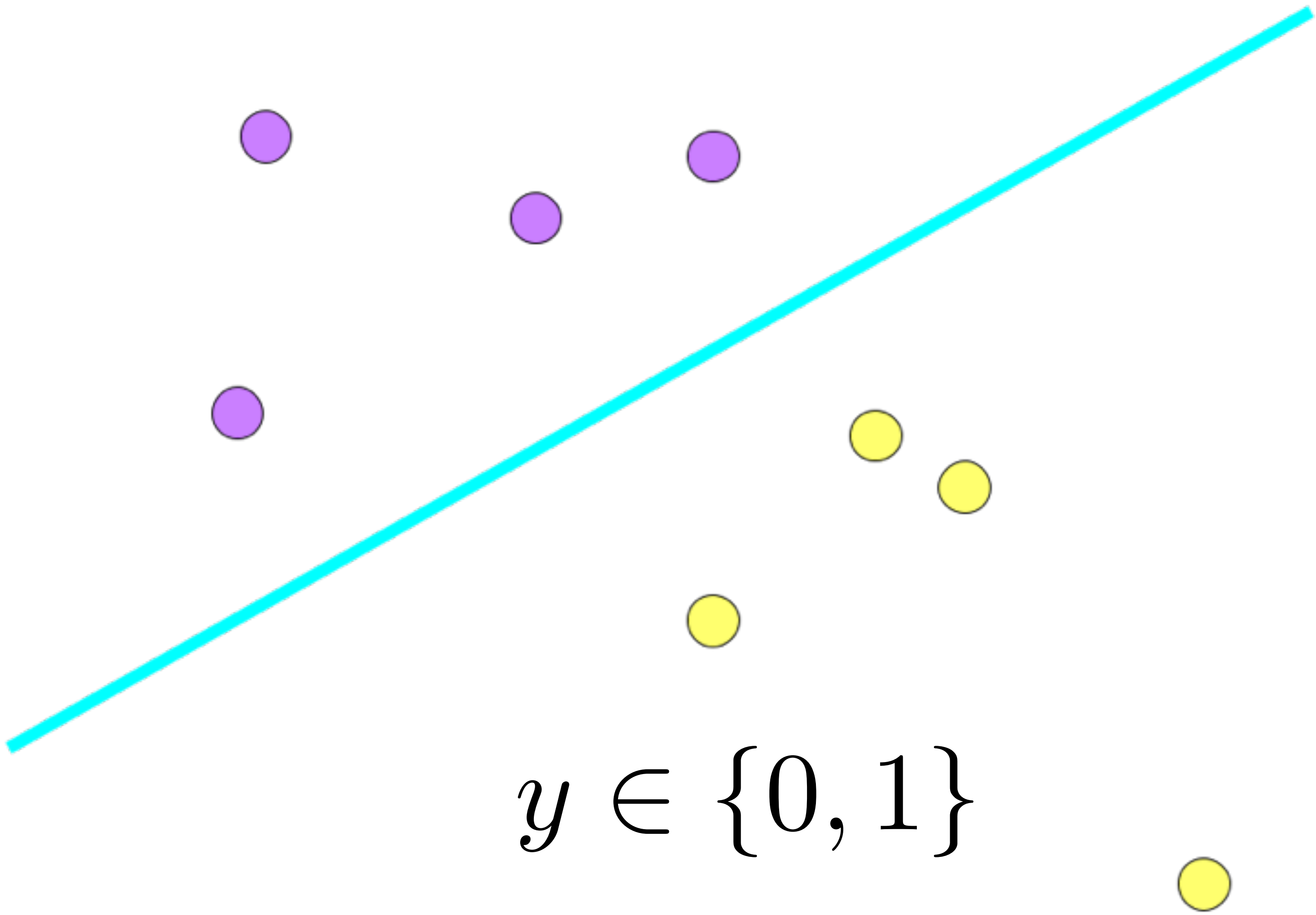
- ERM principle - choose the model $\hat{f}$ in hypothesis class $\mathcal{H}$ that minimizes loss on the training sample $S \in \{\mathcal{X} \times \mathcal{Y}\}^n$

- Most research assumes simplest case

$$\mathcal{X} = \mathcal{R}^d, \mathcal{Y} = \{0, 1\}$$

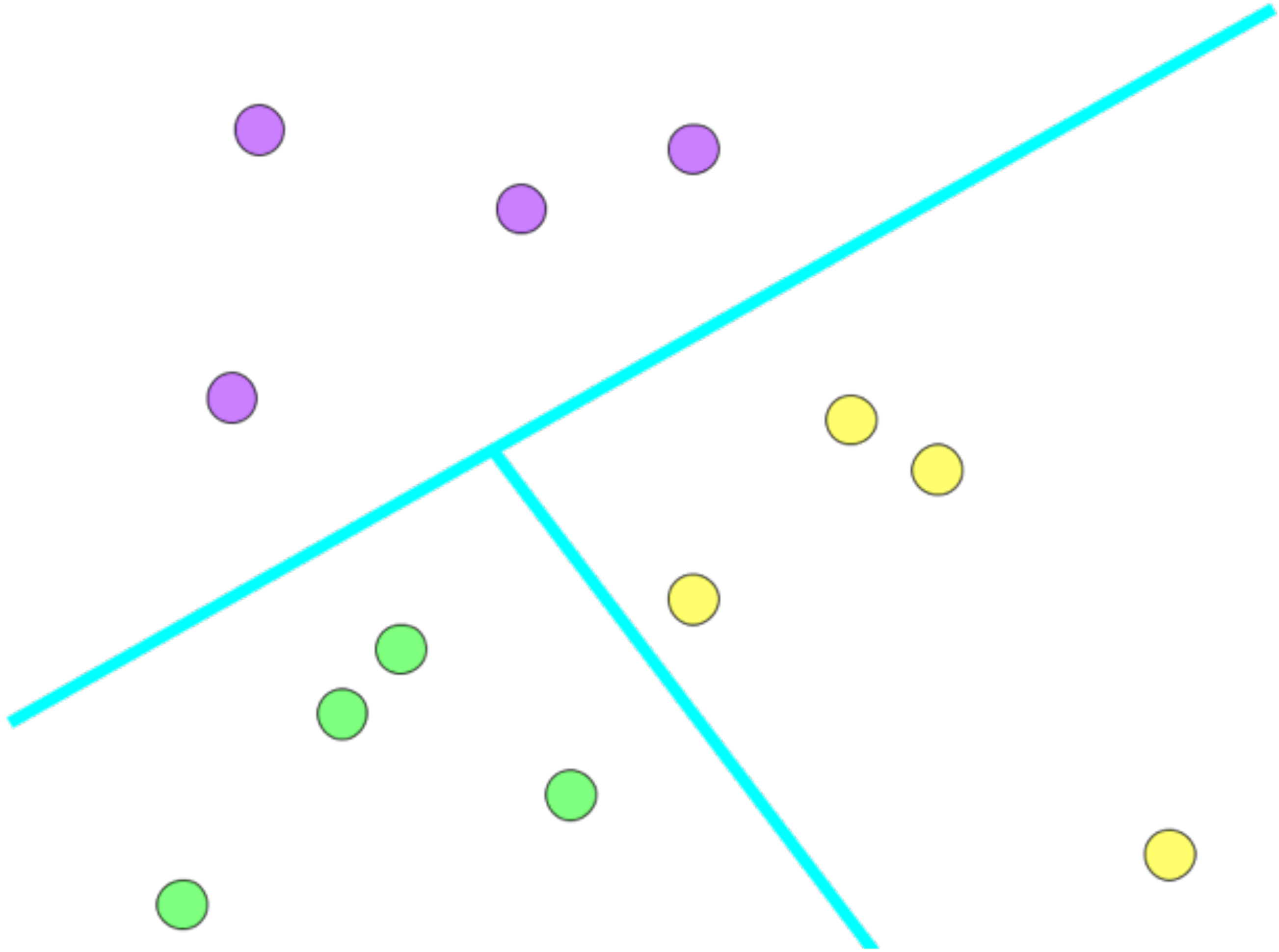
- Real world much messier

Binary Classification



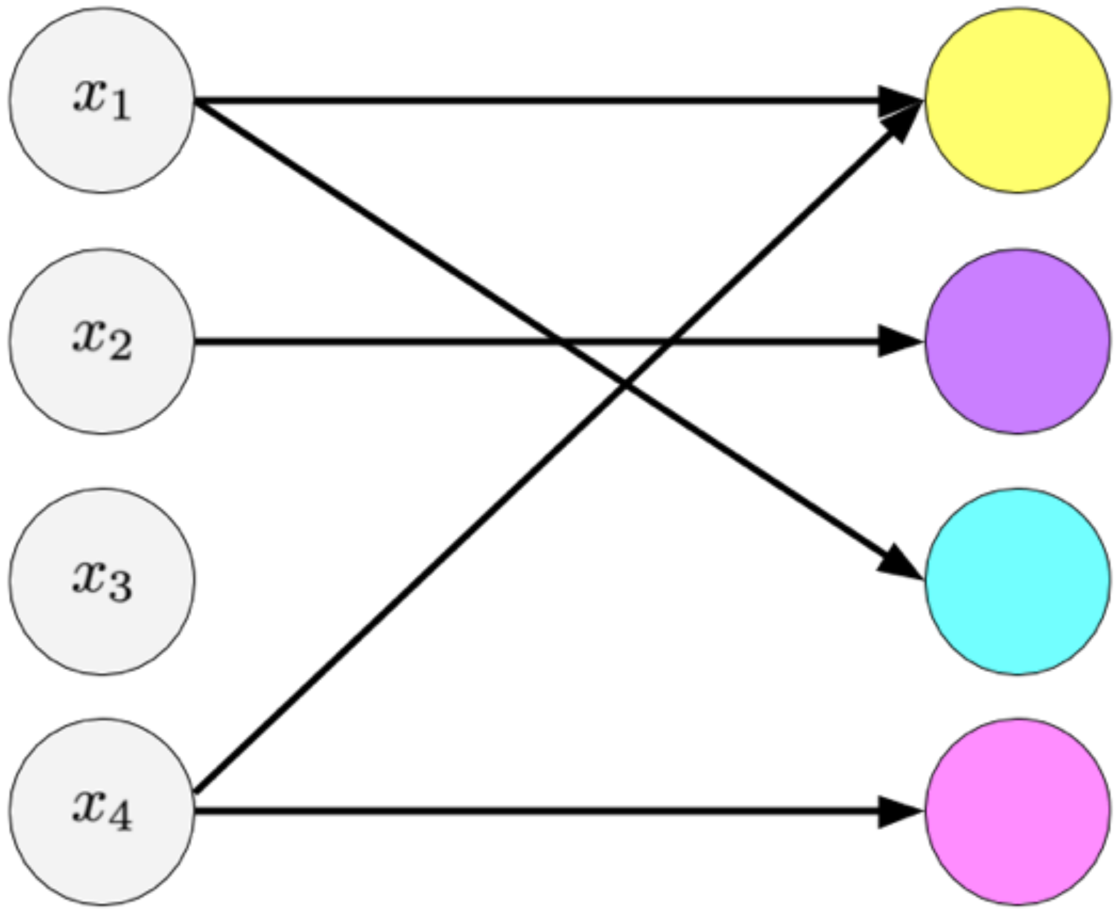
$$y \in \{0, 1\}$$

Multiclass Classification



$$y \in \{c_1, c_2, ..., c_L\}$$

Multilabel Classification



$$y \subseteq \{c_1, c_2, ..., c_L\}$$

Why Multilabel?

- **Superset of both BC and MC:**
  BC when $|L|$ = 1, MC when $y \in L$

- **Natural for many real problems:**
  Clinical diagnosis
  Predicting purchases
  Auto-tagging news articles
  Activity recognition
  Object detection

- **Easy to formulate:**
  Take L tasks and slap them together

Naive Baseline

- **Binary relevance:**
  Separately train $|L|$ classifiers $f_l : \mathcal{X} \to \{0, 1\}$
- **Pros:**
  Simple to execute, easy to understand
  strong baseline
- **Cons:**
  Computational cost: $|L|\times$
  Leaves some information on the table (correlation betw. labels)

Challenges

- **Efficiency**
  Develop classifiers that do not scale in time or space complexity with the number of labels

- **Performance**
  Make use of the extra labels to achieve better accuracy, generalization

- **Evaluation**
  How do we evaluate a multilabel classifier's performance across 10s, 100s, 1000s, or even 1M labels?

Outline

- Introduction to Multilabel Learning
- **Evaluation**
- Efficient Learning & Sparse Models
- Deep Learning for Multilabel Classification
- Classifying Multilabel Time Series with RNNs

Why not accuracy?

- **Often extreme class imbalance**
  When blind classifier gets 99.99%,
  can be optimal to be uninformative

- **Varying base rates across labels**
  E.g.: MeSH dataset: Human applies to 71% of articles, platypus in <.0001%

F1 Score

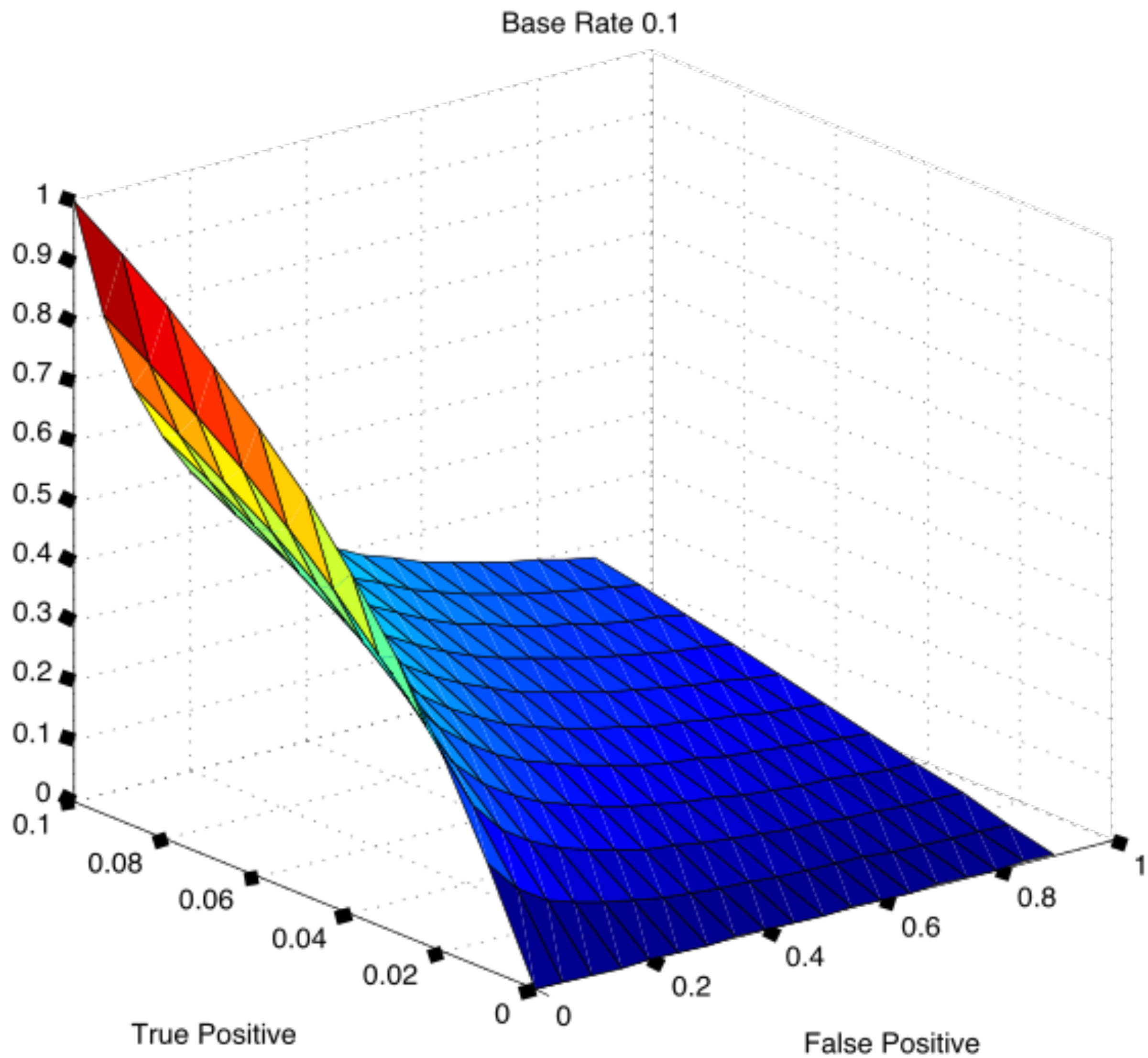
- Easy to calculate from confusion matrix

	Actual +	Actual -
Predicted +	tp	fp
Predicted -	fn	tn

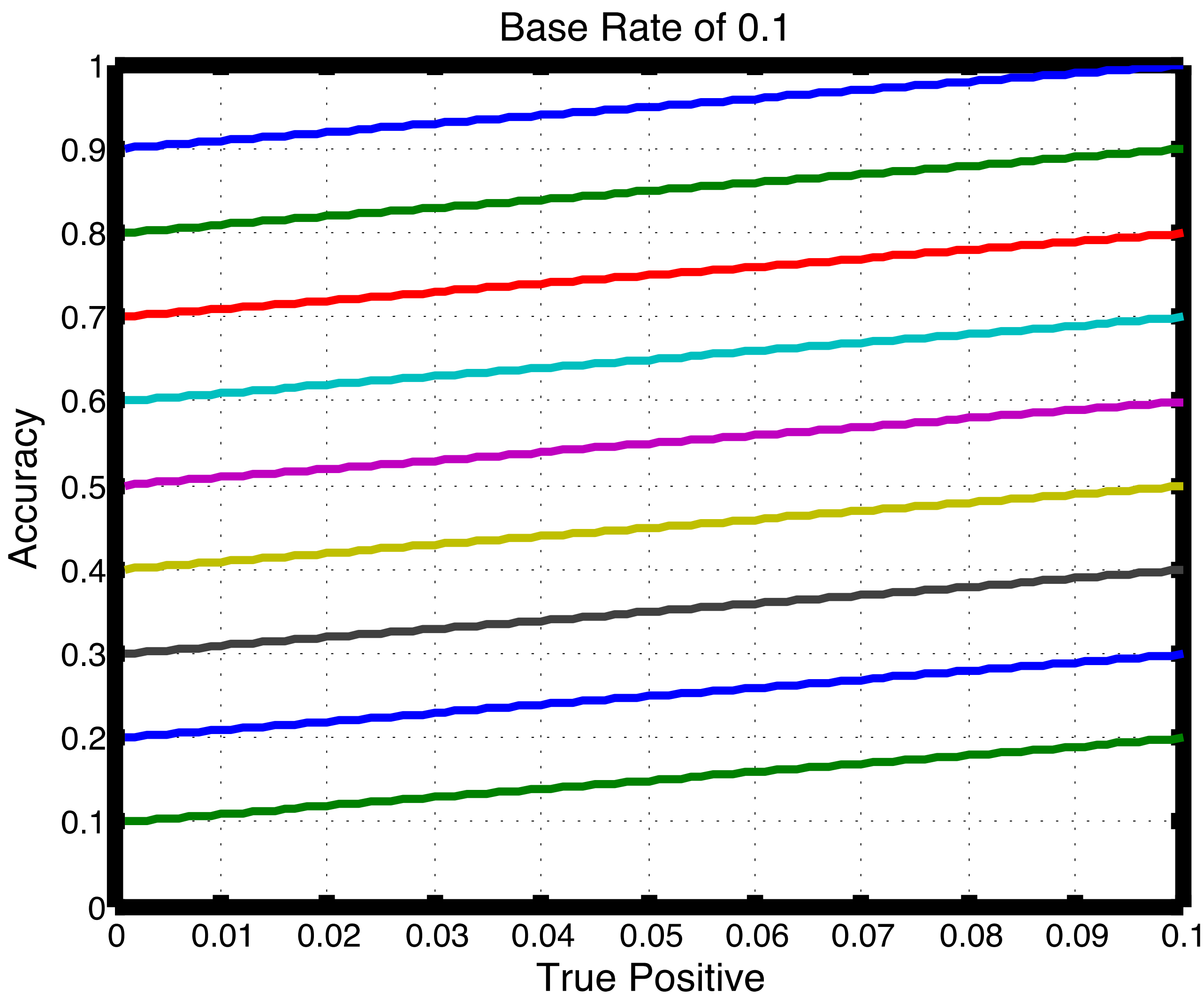
- Harmonic mean of precision $\frac{tp}{tp+fp}$ and recall $\frac{tp}{tp+fn}$

$$\mathrm{F1} = \frac{2 \cdot tp}{2 \cdot tp + fp + fn}$$

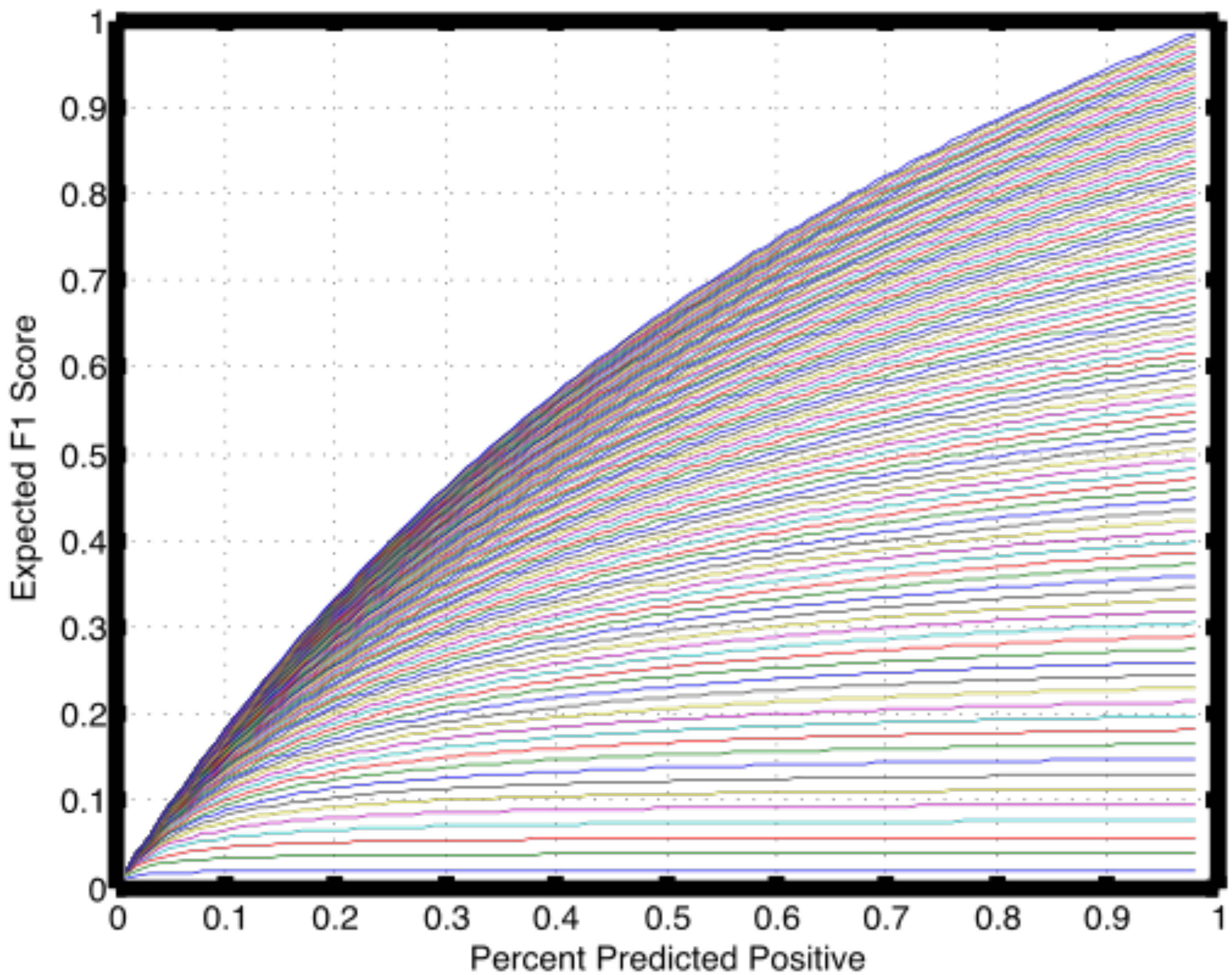
F1 given fixed base rate



Compared to Accuracy



Expected F1 for Uninformative Classifier



Multilabel Variations

Micro F1 calculated over all entries

Example 1	TP	FP	FN	TN
Example 2	FP	FP	FN	TP
Example 3	FN	TP	FN	FP
…	TN	TP	TP	TN

Macro F1

- Macro: F1 calculated separately for each label and averaged

	Label 1	Label 2	Label 3	Label 4
Example 1	TP	FP	FN	TN
Example 2	FP	FP	FN	TP
Example 3	FN	TP	FN	FP
…	TN	TP	TP	TN

Characterizing the Optimal Threshold

- Threshold can be expressed in terms of the conditional probabilities of scores given labels

$$\frac{b \cdot p(s|t=1)}{(1-b) \cdot p(s|t=0)} \geq J$$

- When scores are calibrated probabilities, optimal threshold is precisely half the F1 it achieves.

$$s \geq \frac{tp}{2tp+fn+fp} = \frac{F}{2}$$

Problems with F1

- Sensitive to thresholding strategy
- Hard to tell who has the best algorithms and who is smart about thresholding
- Micro-F1 biased towards common labels
- Macro-F1 biased against them

Some alternatives

- Any threshold indicates a cost sensitivity:
When you know the cost, specify it and use weighted accuracy

- AUC exhibits same dynamic range for every label (blind classifier gets 0, perfect is 1)

- Macro-averaged AUC scores may give a better sense of performance across all labels

****high AUC for rare labels can be misleading.**
can achieve AUC of .99 produce useless results for IR

Outline

- Introduction to Multilabel Learning
- Evaluation
- **Efficient Learning & Sparse Models**
- Deep Learning for Multilabel Classification
- Classifying Multilabel Time Series with RNNs

The problem

- With many labels, binary relevance models can be huge and slow
- 10k labels + 1M features = 80GB of parameters
- We want compact models
  Fast to train and evaluate, cheap to store

Linear Regression

- The bulk of computation is label agnostic (compute inverse $(X^TX)^{-1}$

$$\theta = (X^TX)^{-1}X^Tb$$
$$\theta = (X^TX)^{-1}X^TB$$

- Can do this especially fast when we reduce dimensionality of X via SVD.

- Problem: Unsupervised dim reduction -> lose signal of rare features -> mess up rare labels

Sparsity

- For auto-tagging tasks, features are often high-dimensional sparse bag-of-words or n-grams

- Datasets for web-scale information retrieval tasks are large in the number of examples, thus SGD is the default optimization procedure
- Absent regularization, the gradient is sparse and training is fast
- Regularization destroys the sparsity of the gradient
- Number of features and labels are large, dense stochastic updates are computationally infeasible

Regularization

- Goals: achieve model sparsity, prevent overfitting
- ℓ_1 regularization is induces sparse models
- ℓ_2^2 regularization is thought to achieve more accurate models in practice
- Elastic net, balances the two

$$F(\boldsymbol{w}) = L(\boldsymbol{w}) + \lambda_1 \cdot |\boldsymbol{w}|_1 + \frac{1}{2}\lambda_2 \cdot |\boldsymbol{w}|_2^2$$

Balancing Regularization with Efficiency

- To regularize while maintaining efficiency, can use a lazy updating scheme, first described by Carpenter (2008)
- For each feature, remember the last time it was nonzero
- When a feature is nonzero at some step t+k, perform a closed form update
- We derive lazy updates for elastic net regularization on both standard SGD and FoBoS (Duchi & Singer)

Lazy Updates for Elastic Net

Theorem 1 *To bring the weight w_j current from time ψ_j to time k using SGD, the constant time update is*

$$
\begin{aligned}
w_j^{(k)} = \operatorname{sgn}(w_j^{(\psi_j)}) \Bigg[|w_j^{(\psi_j)}| \frac{P(k-1)}{P(\psi_j - 1)} \\
- P(k-1) \cdot (B(k-1) - B(\psi_j - 1)) \Bigg]_+
\end{aligned}
\tag{1}
$$

where $P(t) = (1 - \eta^{(t)}\lambda_2) \cdot P(t-1)$ with base case $P(-1) = 1$ and $B(t) = \sum_{\tau=0}^{t} \eta^{(\tau)}/P(\tau - 1)$ with base case $B(-1) = 0$.

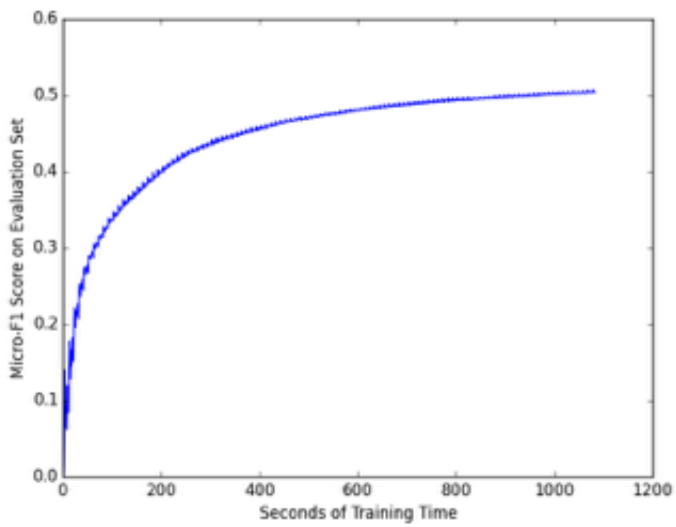
Theorem 2 *A constant-time lazy update for FoBoS with elastic net regularization and decreasing learning rate to bring a weight current at time k from time ψ_j is*

$$
\begin{aligned}
w_j^{(k)} = \operatorname{sgn}(w_j^{(\psi_j)}) \Bigg[|w_j^{(\psi_j)}| \frac{\Phi(k-1)}{\Phi(\psi_j - 1)} - \\
\Phi(k-1) \cdot \lambda_1 \left(\beta(k-1) - \beta(\psi_j - 1) \right) \Bigg]_+
\end{aligned}
\tag{2}
$$

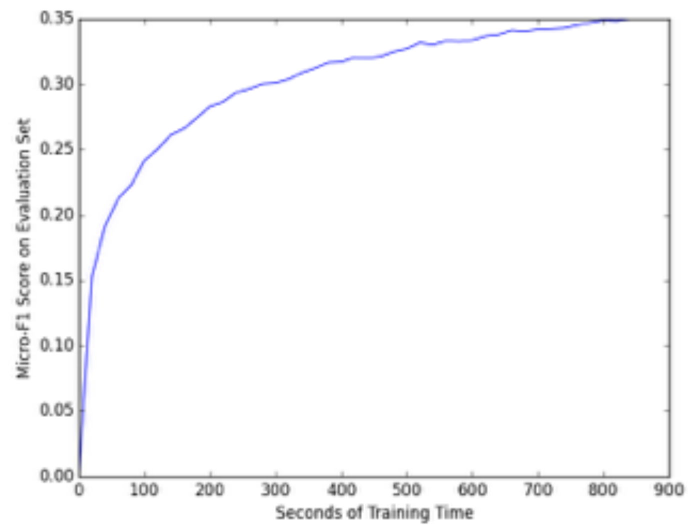
where $\Phi(t) = \Phi(t-1) \cdot \frac{1}{1+\eta^t \lambda_2}$ with base case $\Phi(-1) = 1$ and $\beta(t) = \beta(t-1) + \frac{\eta^{(t)}}{\Phi(t-1)}$ with base case $\beta(-1) = 0$.

Empirical Validation

- On two largest datasets in Mulan repository of multilabel datasets, we can train to convergence on a laptop in just minutes
- *rcv1*: 490x speedup, *bookmarks*: 20x speedup



rcv1



bookmarks

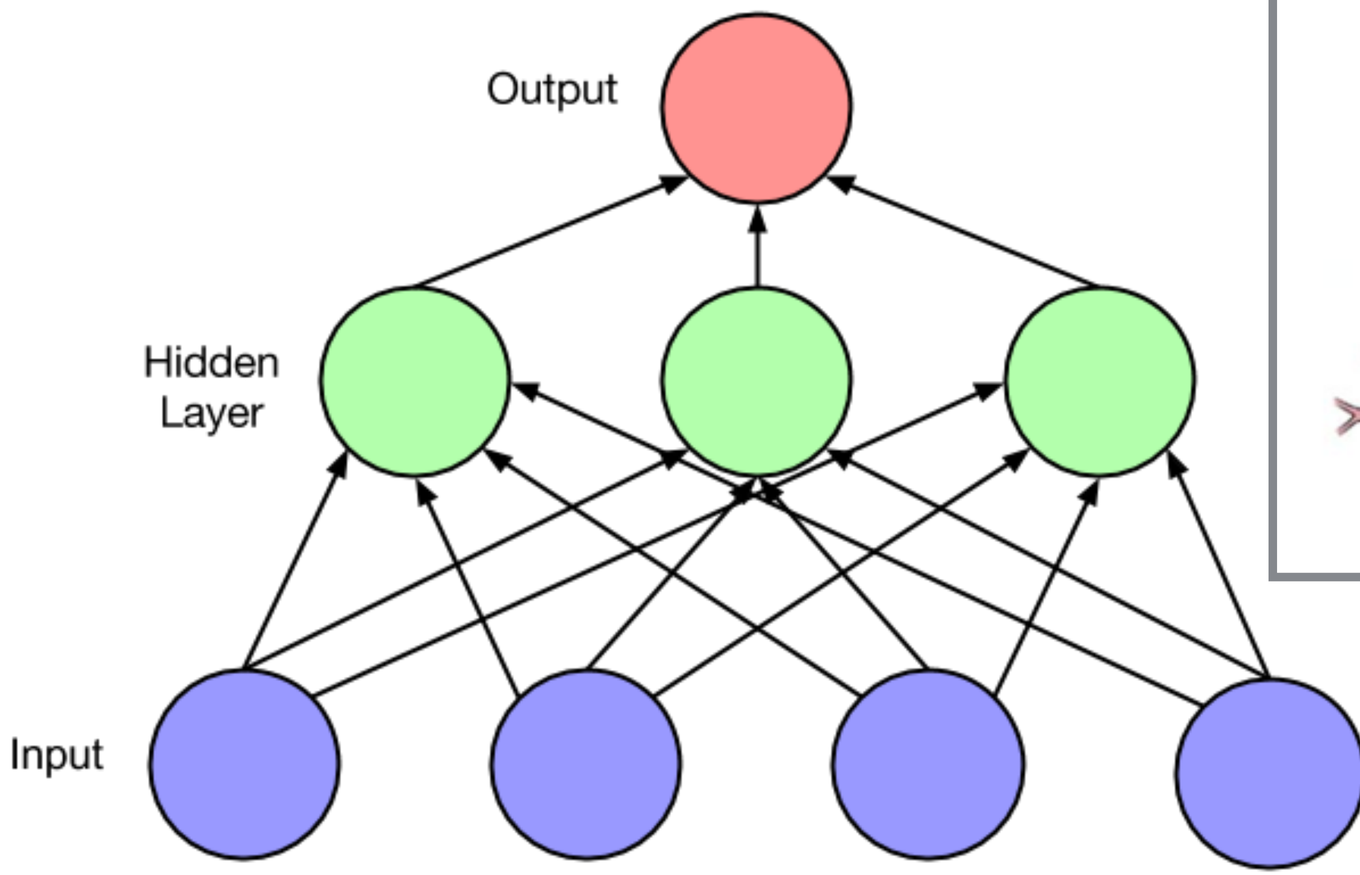
Outline

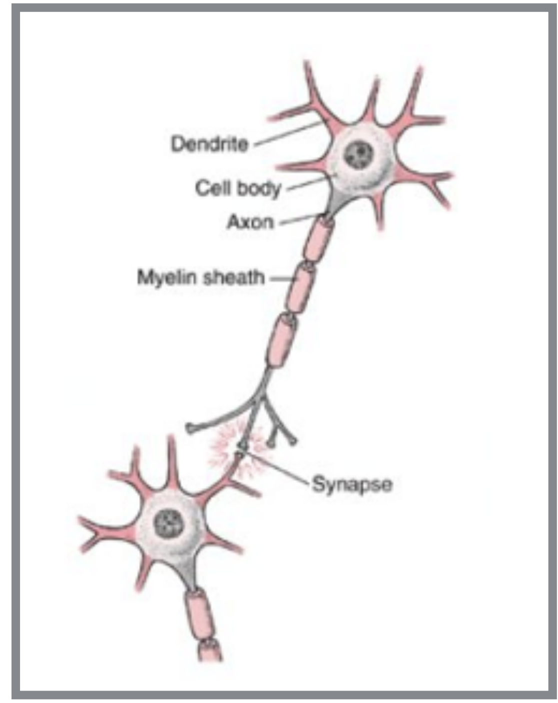
- Introduction to Multilabel Learning
- Evaluation
- Efficient Learning & Sparse Models
- **Deep Learning for Multilabel Classification**
- Classifying Multilabel Time Series with RNNs

Performance

- Efficiency is nice, but we'd also like performance
- Neural networks can learn *shared representations* across labels.
- Both regularizes each label's model and exploits correlations between labels
- In extreme multilabel, may use significantly less parameters than logistic regression

Neural Network



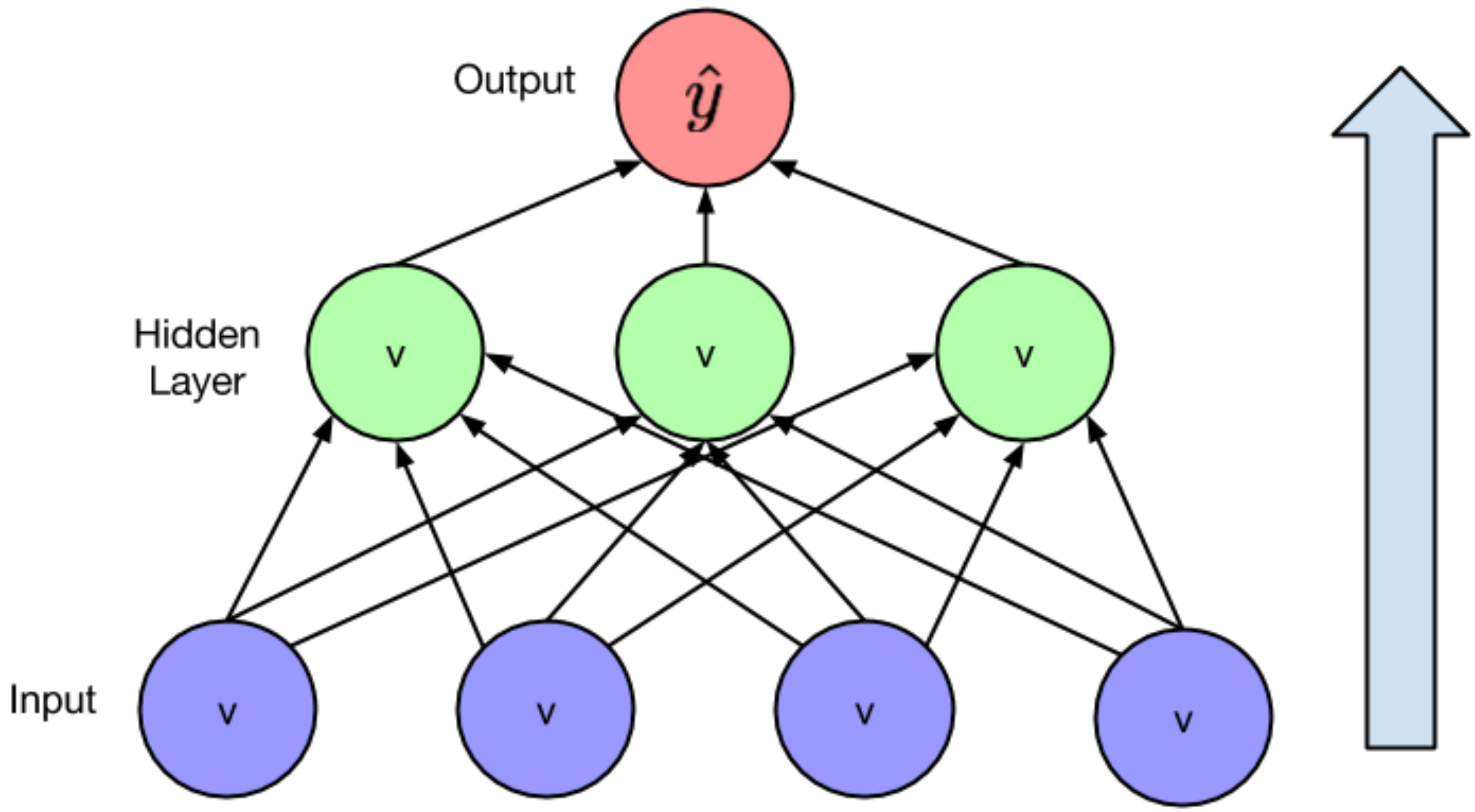


Training w Backpropagation

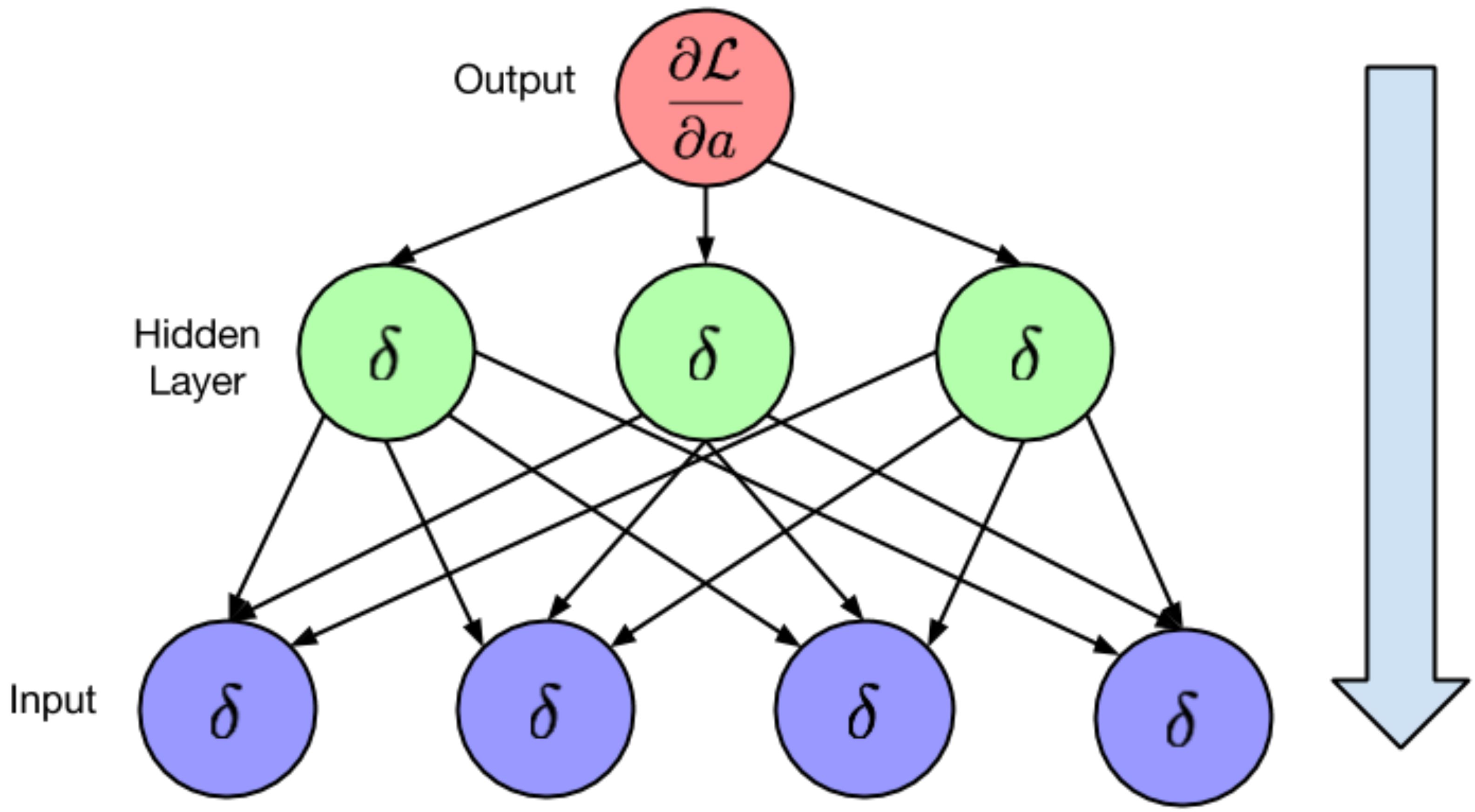
- Goal: calculate the derivative of loss function with respect to each parameter (weight) in the model
- Update the weights by gradient following:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} \mathcal{L}_i$$

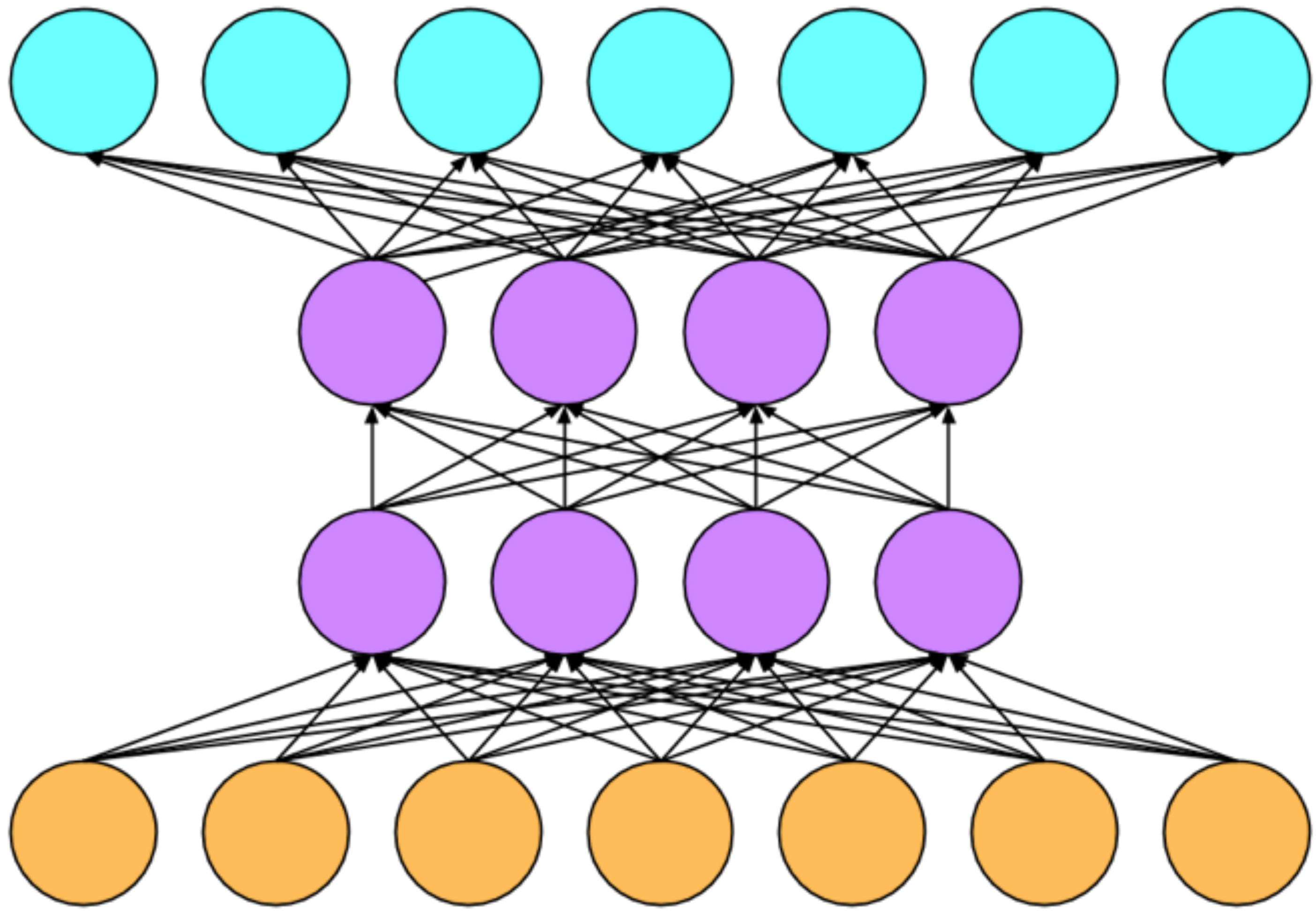
Forward Pass



Backward Pass



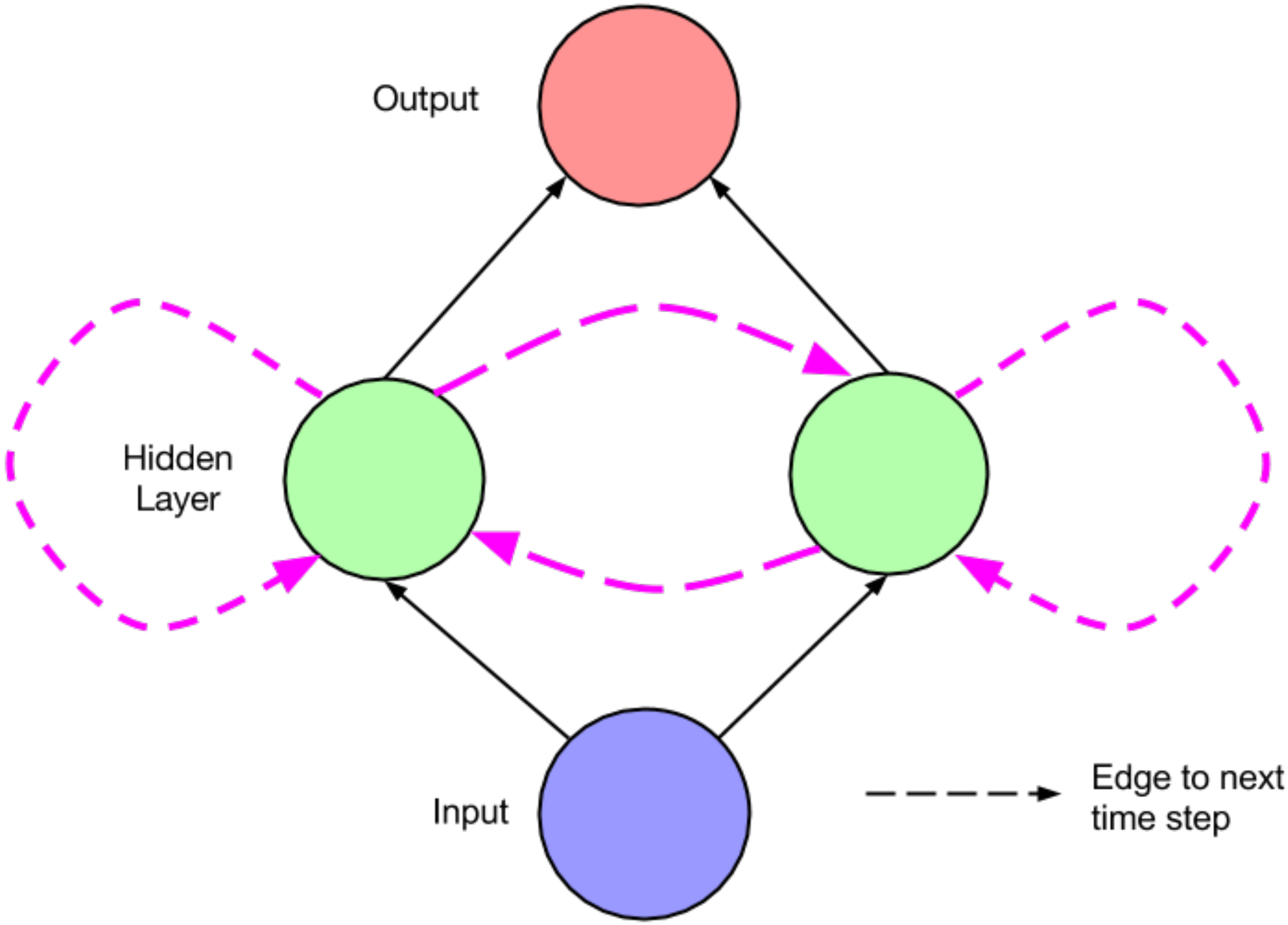
Multilabel MLP



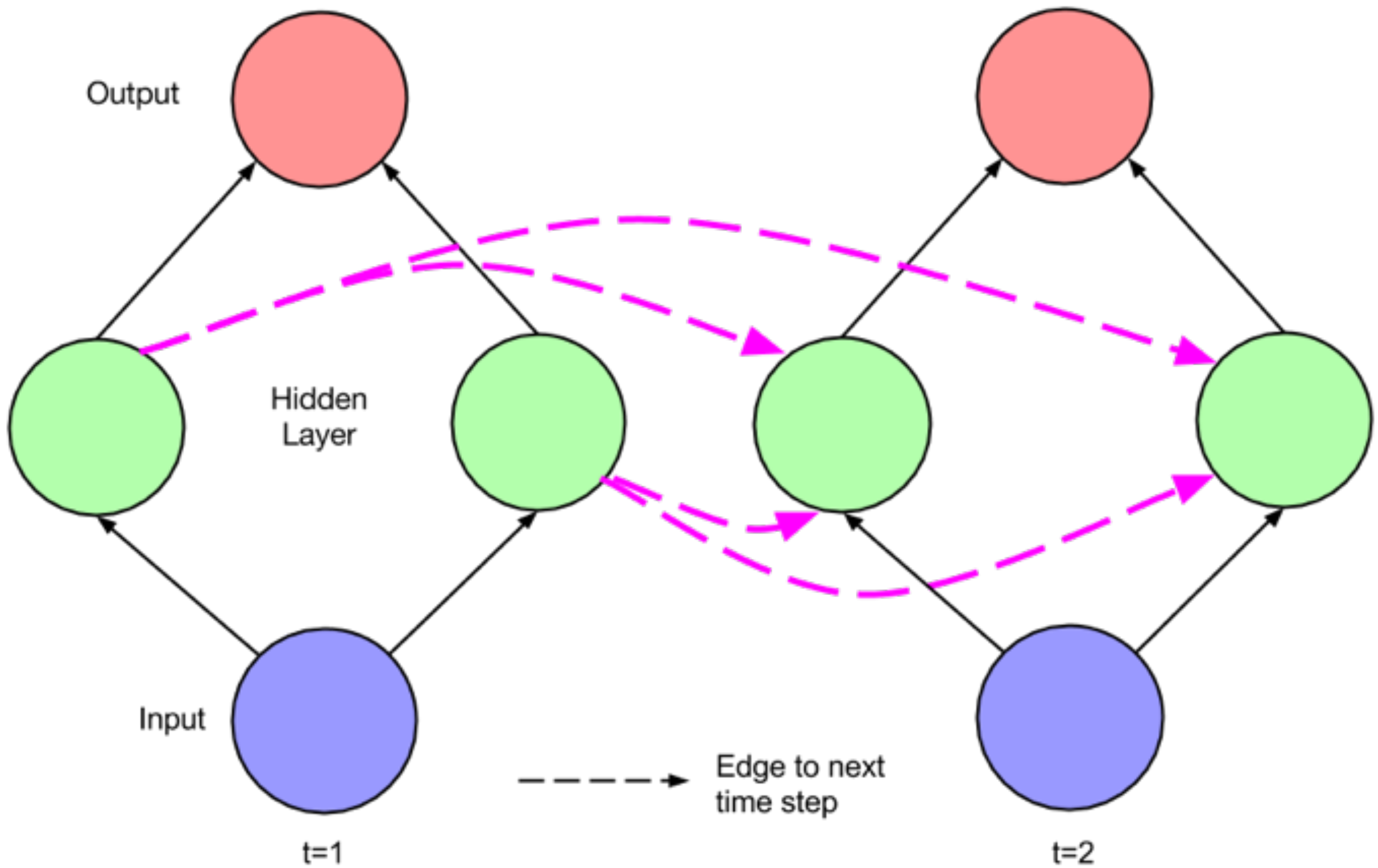
Outline

- Introduction to Multilabel Learning
- Evaluation
- Efficient Learning & Sparse Models
- Deep Learning for Multilabel Classification
- **Classifying Multilabel Time Series with RNNs**

To Model Sequential Data: Recurrent Neural Networks



Recurrent Net (Unfolded)

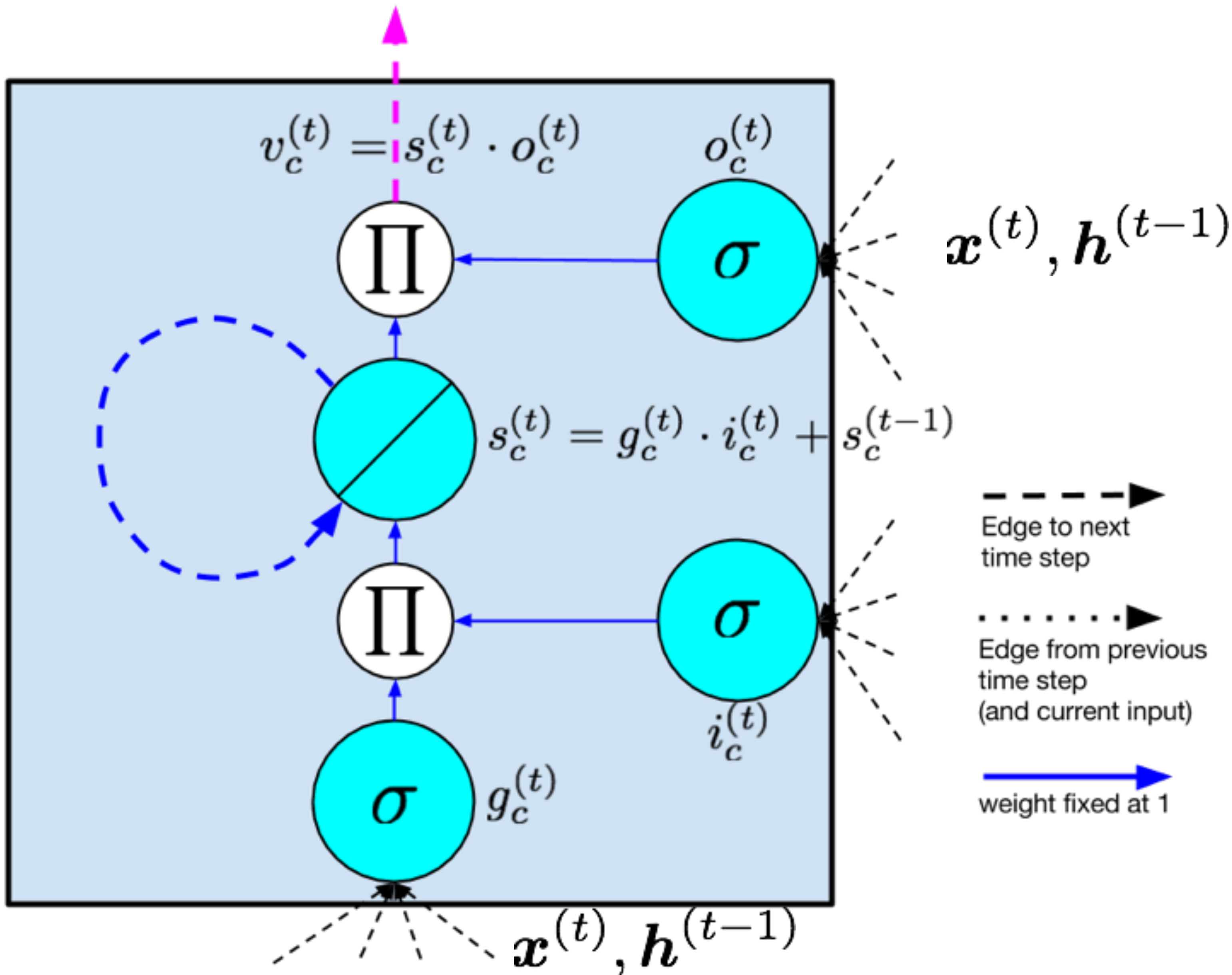


$$h^{(t)} = \sigma(W_{hx}\boldsymbol{x}^{(t)} + W_{hh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h)$$

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}(W_{yh}\boldsymbol{h}^{(t)} + \boldsymbol{b}_y)$$

LSTM Memory Cell

(Hochreiter & Schmidhuber, 1997)



LSTM Forward Pass

$$\boldsymbol{g}^{(t)} = \phi(W_{gx}\boldsymbol{x}^{(t)} + W_{ih}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_g)$$

$$\boldsymbol{i}^{(t)} = \sigma(W_{ix}\boldsymbol{x}^{(t)} + W_{ih}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i)$$

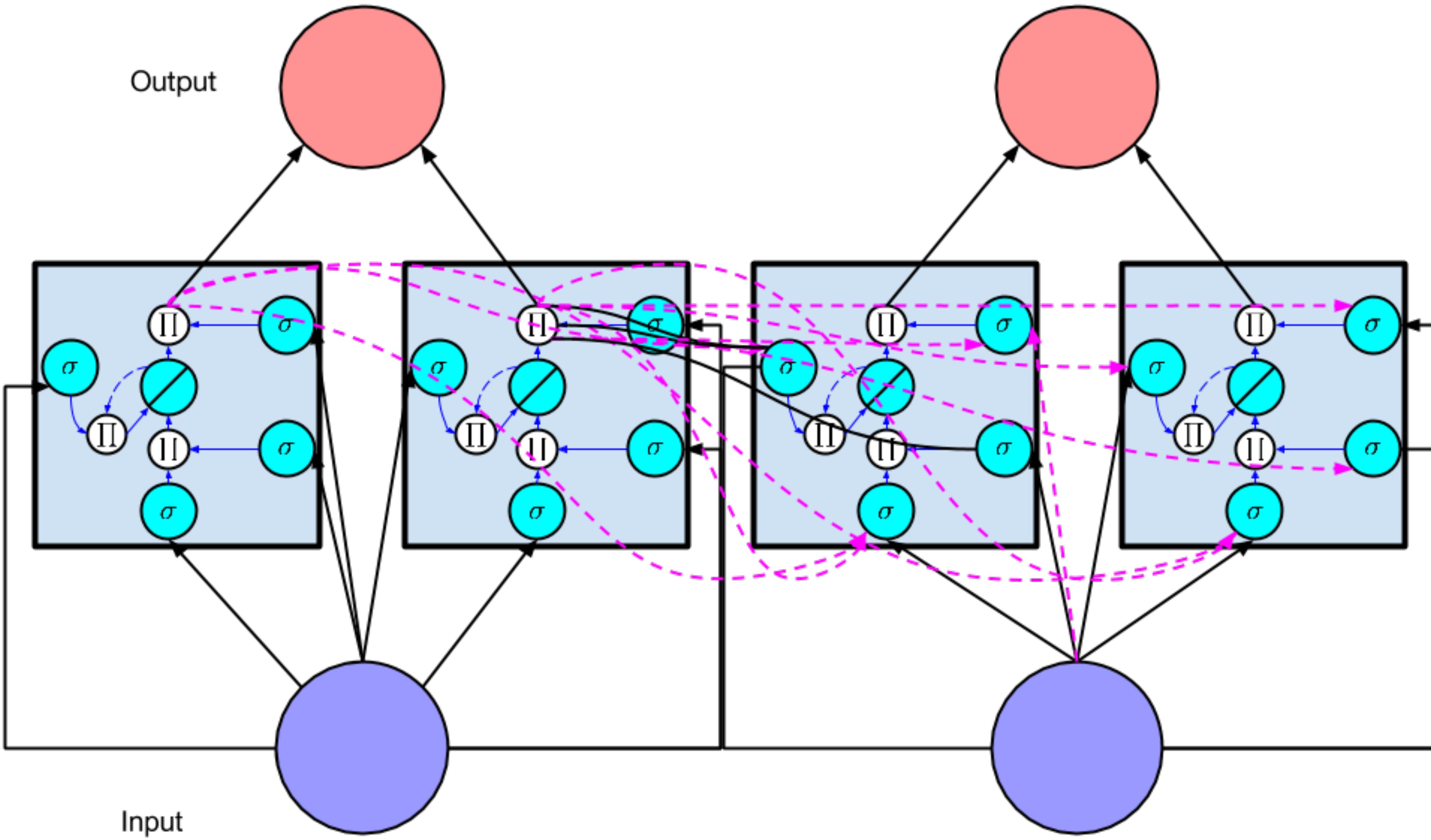
$$\boldsymbol{f}^{(t)} = \sigma(W_{fx}\boldsymbol{x}^{(t)} + W_{fh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f)$$

$$\boldsymbol{o}^{(t)} = \sigma(W_{ox}\boldsymbol{x}^{(t)} + W_{oh}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o)$$

$$\boldsymbol{s}^{(t)} = \boldsymbol{g}^{(t)} \odot \boldsymbol{i}^{(i)} + \boldsymbol{s}^{(t-1)} \odot \boldsymbol{f}^{(t)}$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{s}^{(t)} \odot \boldsymbol{o}^{(t)}$$

LSTM (full network)



Unstructured Input

$$\boldsymbol{x}_i =$$

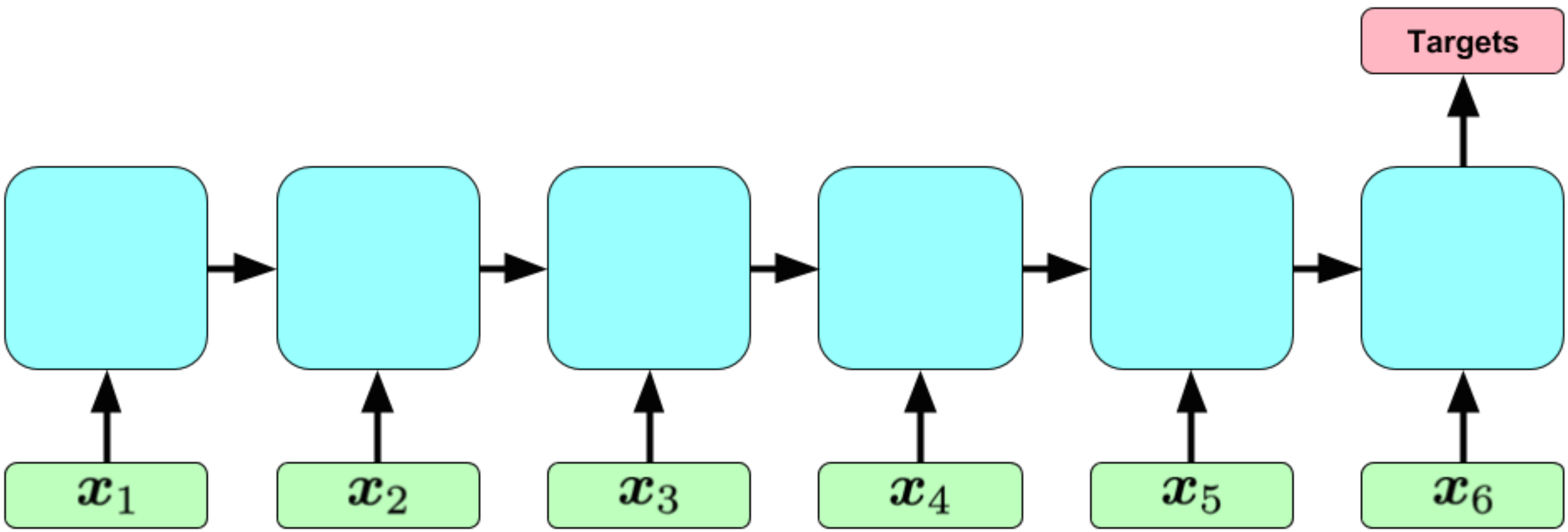
Modeling Problems

- **Examples:** 10,401 episodes
- **Features:** 13 time series (sensor data, lab tests)
- **Complications:** Irregular sampling, missing values, varying-length sequences

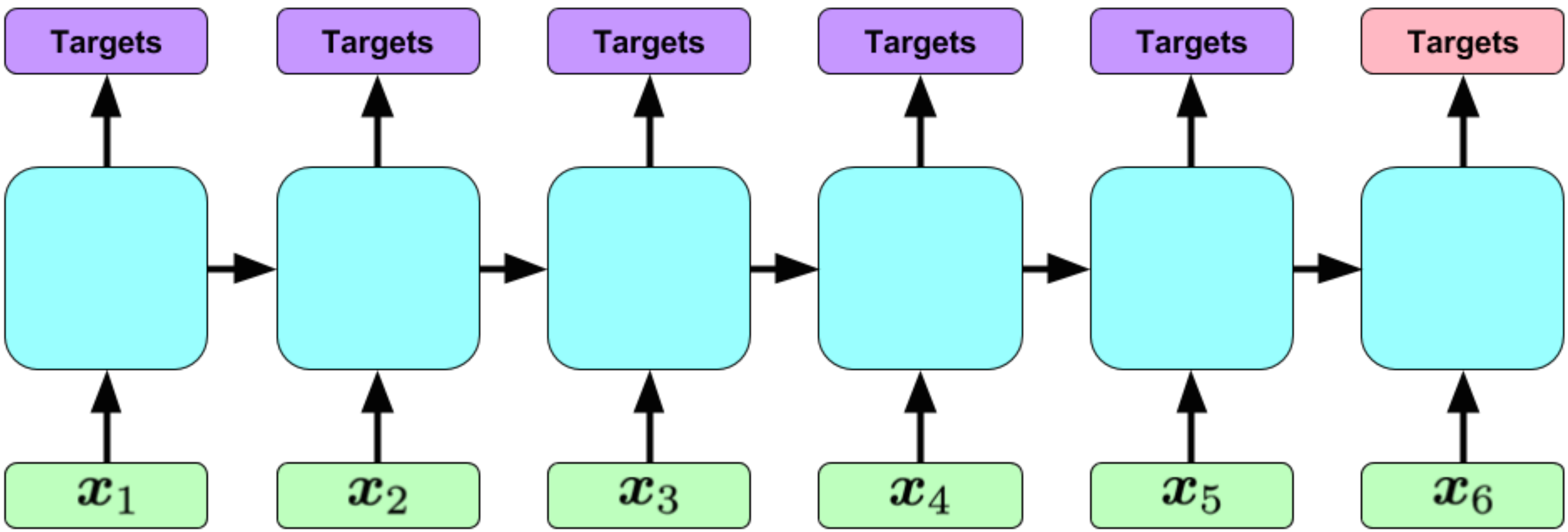
How to models sequences?

- Markov models
- Conditional Random Fields
- **Problem: Cannot model long range dependencies**

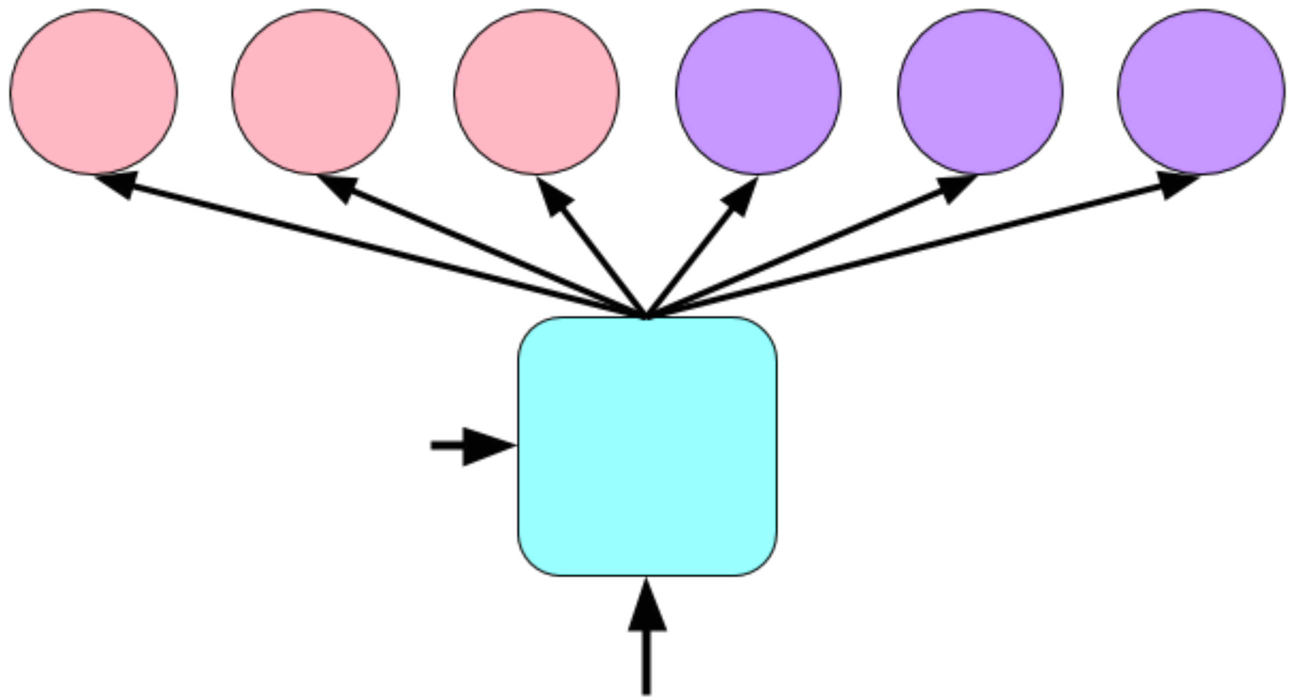
Simple Formulation



Target Replication



Auxiliary Targets



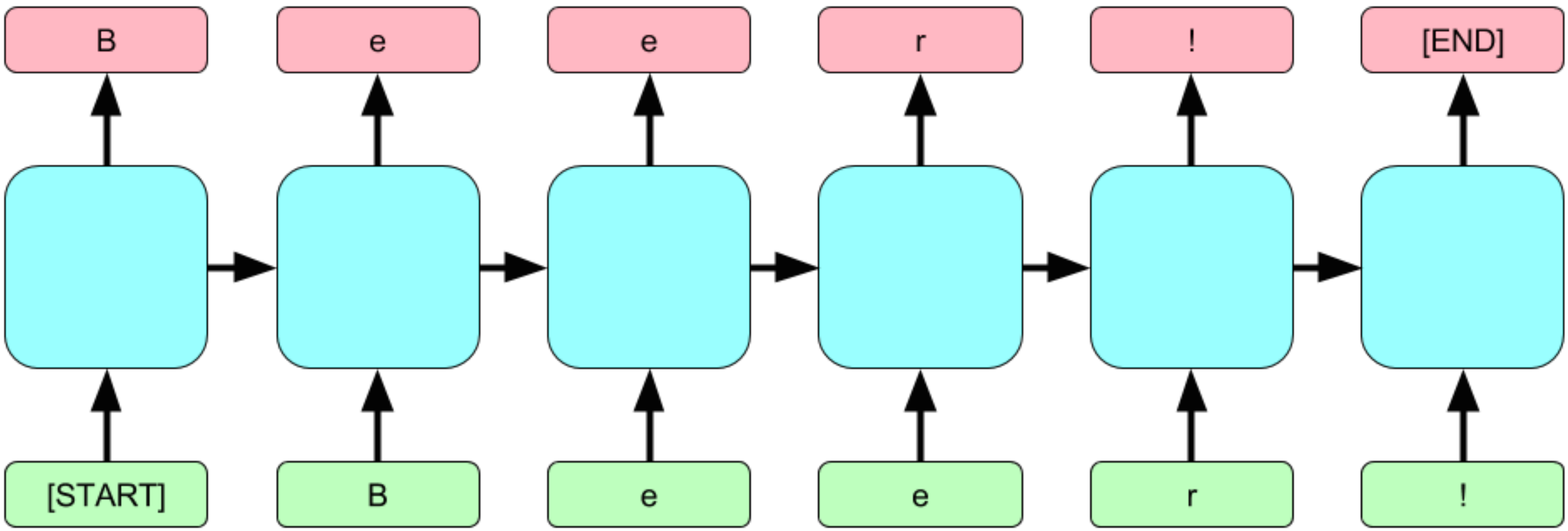
Results

Classification performance for 128 ICU phenotypes					
Model	Micro AUC	Macro AUC	Micro F1	Macro F1	Prec. at 10
Base Rate	0.7128	0.5	0.1346	0.0343	0.0788
Logistic Regression, First 6 + Last 6	0.8122	0.7404	0.2324	0.1081	0.1016
Logistic Regression, Expert features	0.8285	0.7644	0.2502	0.1373	0.1087
MLP, First 6 + Last 6	0.8375	0.7770	0.2698	0.1286	0.1096
MLP, Expert features	**0.8551**	**0.8030**	**0.2930**	**0.1475**	**0.1170**
LSTM Models with two 64-cell hidden layers					
LSTM	0.8241	0.7573	0.2450	0.1170	0.1047
LSTM, AuxOut (Diagnoses)	0.8351	0.7746	0.2627	0.1309	0.1110
LSTM-AO (Categories)	0.8382	0.7748	0.2651	0.1351	0.1099
LSTM-TR	0.8429	0.7870	0.2702	0.1348	0.1115
LSTM-TR-AO (Diagnoses)	0.8391	0.7866	0.2599	0.1317	0.1085
LSTM-TR-AO (Categories)	0.8439	0.7860	0.2774	0.1330	0.1138
LSTM Models with Dropout (probability 0.5) and two 128-cell hidden layers					
LSTM-DO	0.8377	0.7741	0.2748	0.1371	0.1110
LSTM-DO-AO (Diagnoses)	0.8365	0.7785	0.2581	0.1366	0.1104
LSTM-DO-AO (Categories)	0.8399	0.7783	0.2804	0.1361	0.1123
LSTM-DO-TR	**0.8560**	**0.8075**	**0.2938**	0.1485	0.**1172**
LSTM-DO-TR-AO (Diagnoses)	0.8470	0.7929	0.2735	0.1488	0.1149
LSTM-DO-TR-AO (Categories)	0.8543	0.8015	0.2887	0.1446	0.1161
LSTM-DO-TR (Linear Gain)	0.8480	0.7986	0.2896	**0.1530**	0.1160
Ensembles of Best MLP and Best LSTM					
Mean of LSTM-DO-TR & MLP	0.8611	0.8143	0.2981	0.1553	0.1201
Max of LSTM-DO-TR & MLP	**0.8643**	**0.8194**	**0.3035**	**0.1571**	**0.1218**

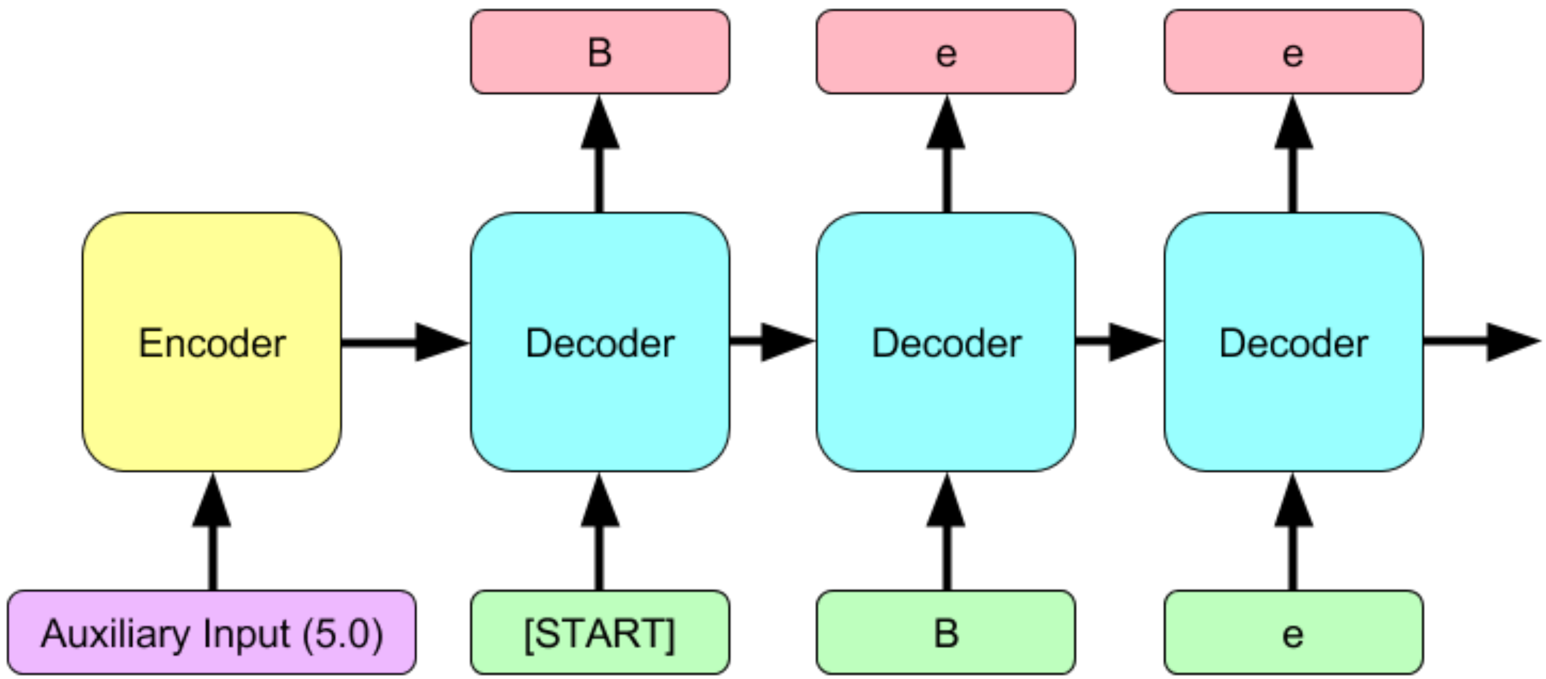
Outline

- Introduction to Multilabel Learning
- Evaluation
- Efficient Learning & Sparse Models
- Deep Learning for Multilabel Classification
- Jointly Learning to Generate and Classify Beer Reviews

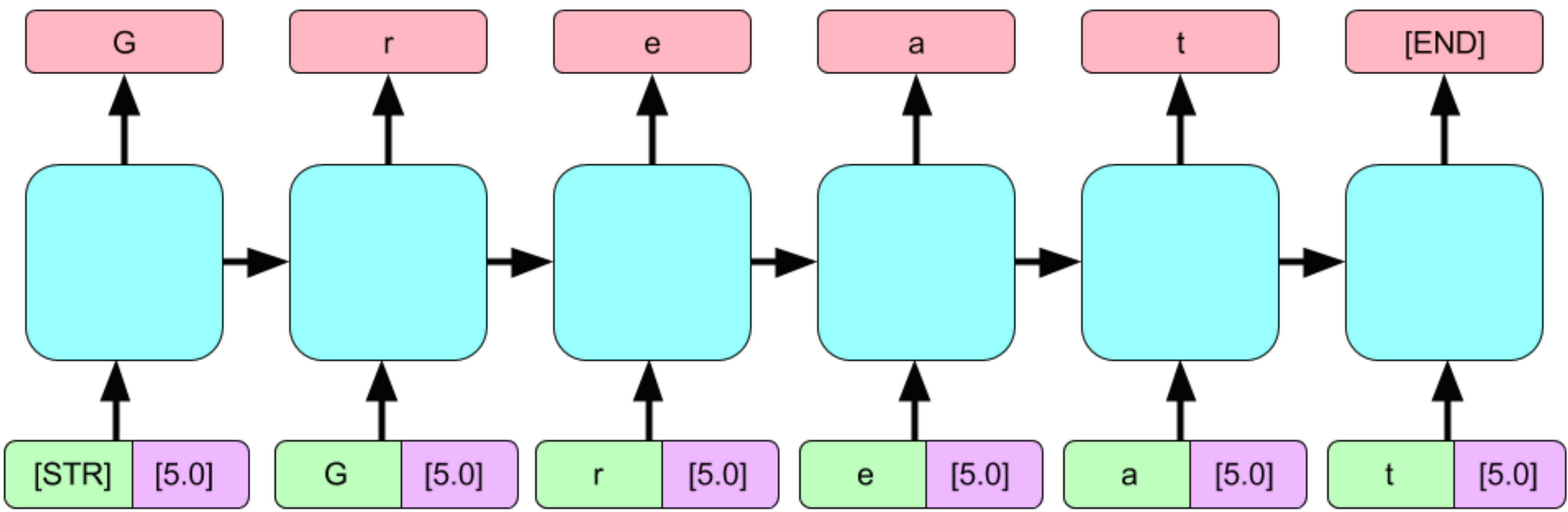
RNN Language Model



Past Supervised Approaches relied upon Encoder-Decoder Model



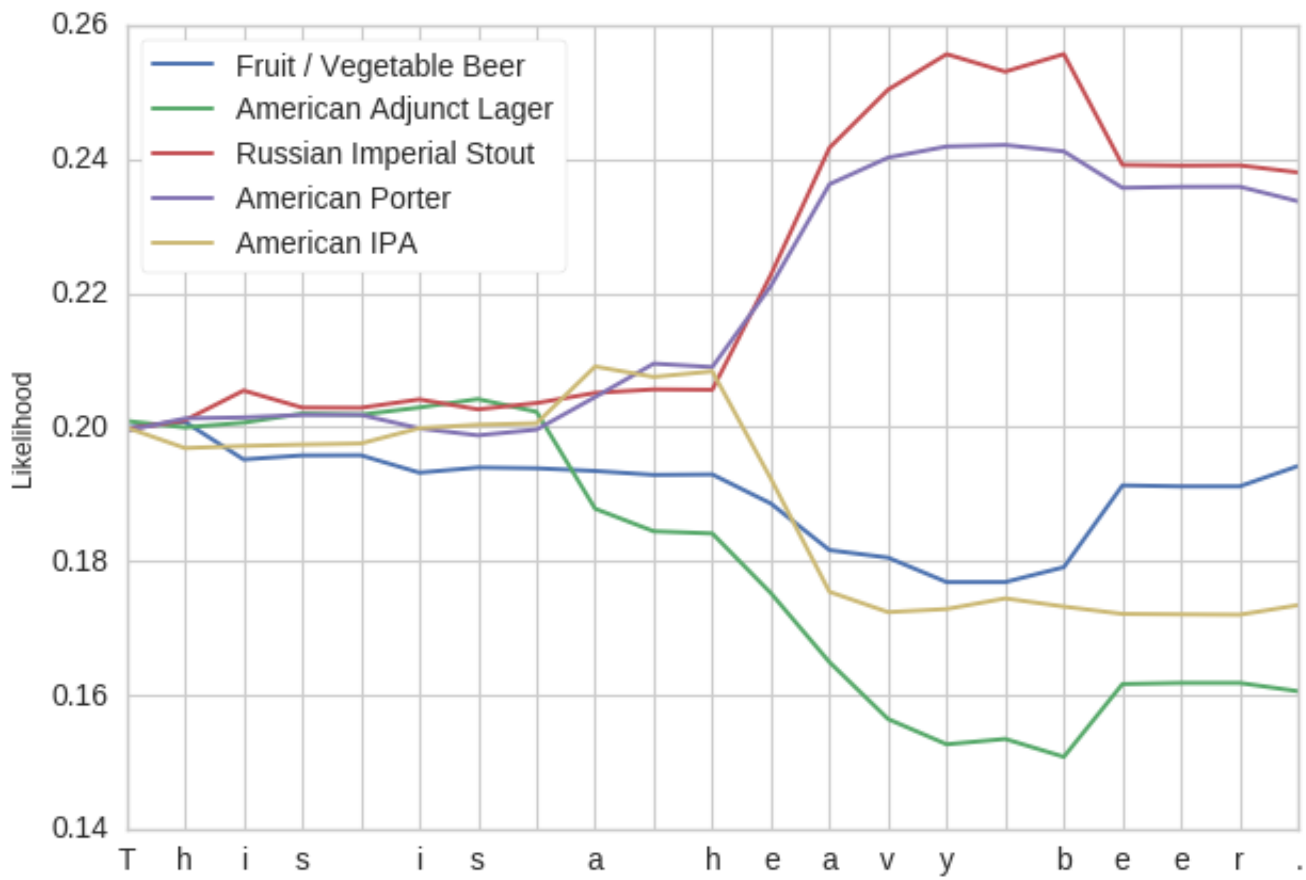
Bridging Long Time Intervals with Concatenated Inputs



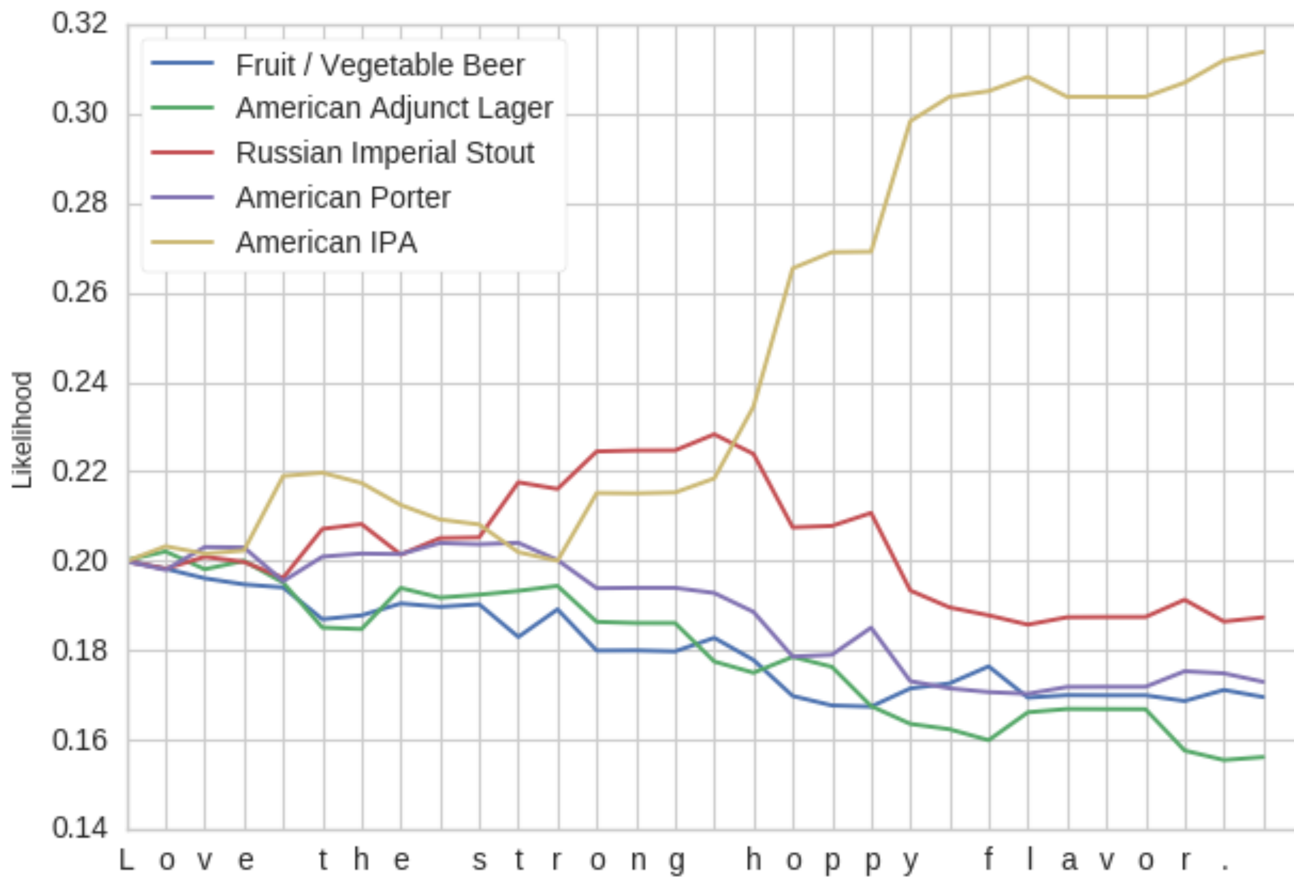
Example

A.5 FRUIT/VEGETABLE BEER
<STR>On tap at the brewpub. A nice dark red color with a nice head that left a lot of lace on the glass. Aroma is of raspberries and chocolate. Not much depth to speak of despite consisting of raspberries. The bourbon is pretty subtle as well. I really don’t know that I find a flavor this beer tastes like. I would prefer a little more carbonization to come through. It’s pretty drinkable, but I wouldn’t mind if this beer was available. <EOS>

Character-based Classification



“Love the Strong Hoppy Flavor”



Thanks!

Contact:
zlipton@cs.ucsd.edu
zacklipton.com

Critical Review of RNNs:
http://arxiv.org/abs/1506.00019

Learning to Diagnose:
http://arxiv.org/abs/1511.03677

Conditional Generative RNNS:
http://arxiv.org/abs/1511.03683