Module outline

Application:

- Text classification with ATIS data

Model:

- Recurrence
- Long-short term memory cell
- Different recurrent networks

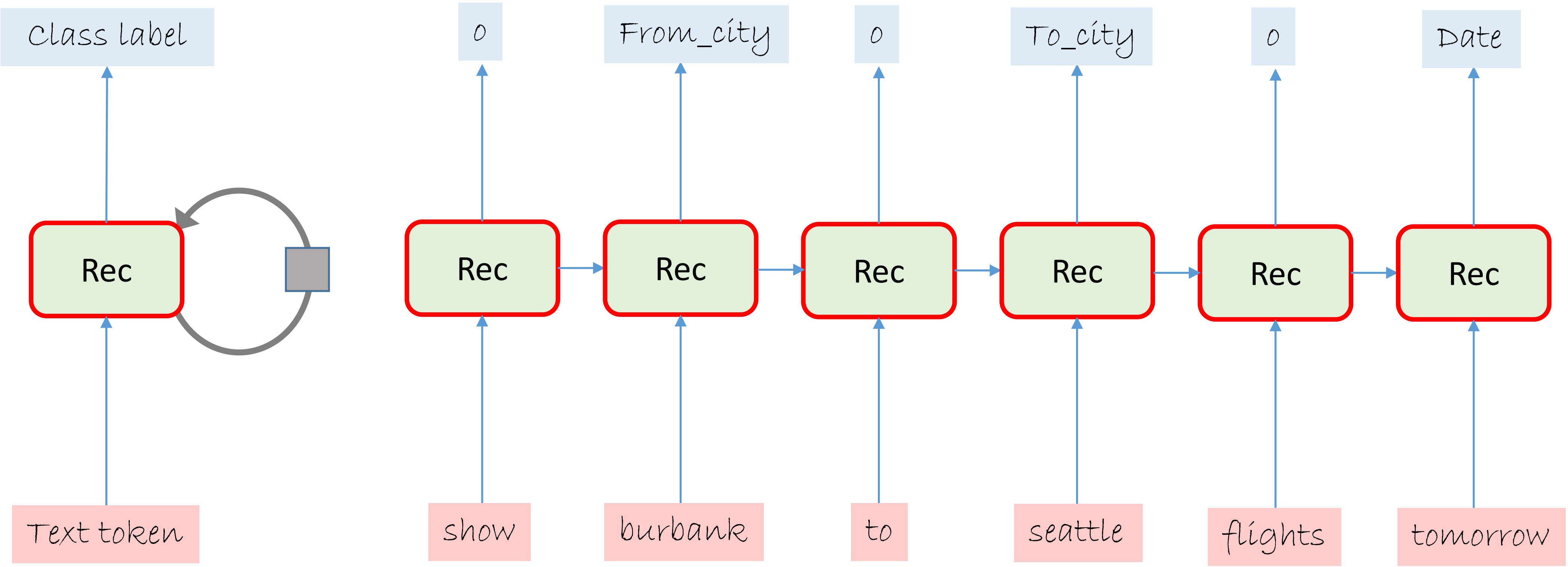
Concept:

- Embedding

Train-Test-Predict Workflow

Sequences (many to many)

Problem: Tagging entities in Air Traffic Controller (ATIS) data



ATIS data

Domain:

✓ ATIS contains human-computer queries from the domain of Air Travel Information Services.

Data summary:

✓ 943 unique words a.k.a. : Vocabulary

✓ 129 unique tags a.k.a.: Labels

✓ 26 intent tags: not used in this tutorial

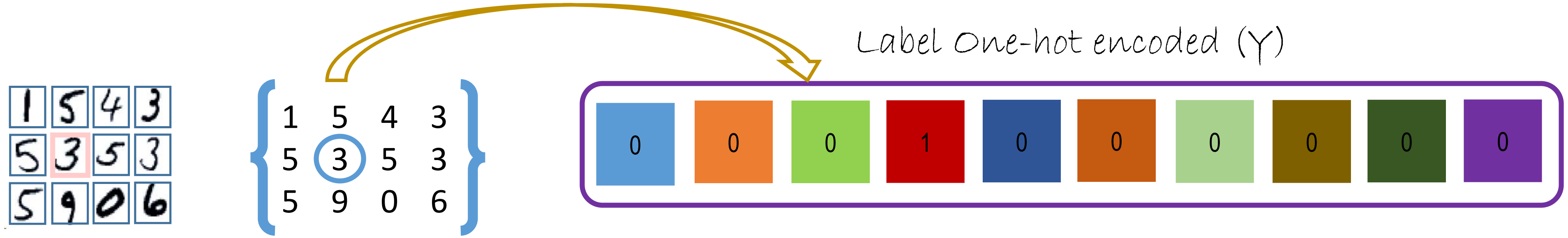
Sequence Id	Input Word (sample)	Word Index (in vocabulary) S0	Word Label	Label Index (S2)
19	# BOS	178:1	# O	128:1
19	# please	688:1	# O	128:1
19	# give	449:1	# O	128:1
19	# me	581:1	# O	128:1
19	# the	827:1	# O	128:1
19	# flights	429:1	# O	128:1
19	# from	444:1	# O	128:1
19	# boston	266:1	# B-fromloc.city_name	48:1
19	# to	851:1	# O	128:1
19	# pittsburgh	682:1	# B-toloc.city_name	78:1
19	# on	654:1	# O	128:1
19	# thursday	845:1	# B-depart_date.day_name	26:1
19	# of	646:1	# O	128:1
19	# next	621:1	# B-depart_date.date_relative	25:1
19	# week	910:1	# O	128:1
19	# EOS	179:1	# O	128:1

Sequence Id:	19 indicates – this sentence is the 19th sentence in the data set
Word Index:	###:1 indicates the position of the corresponding word in the vocabulary (total 943 words)
Label Index:	###:1 indicates the position of the corresponding tag in tag index (total 129 tags)

Sequence Tagging (Input / Label Pre-processing)

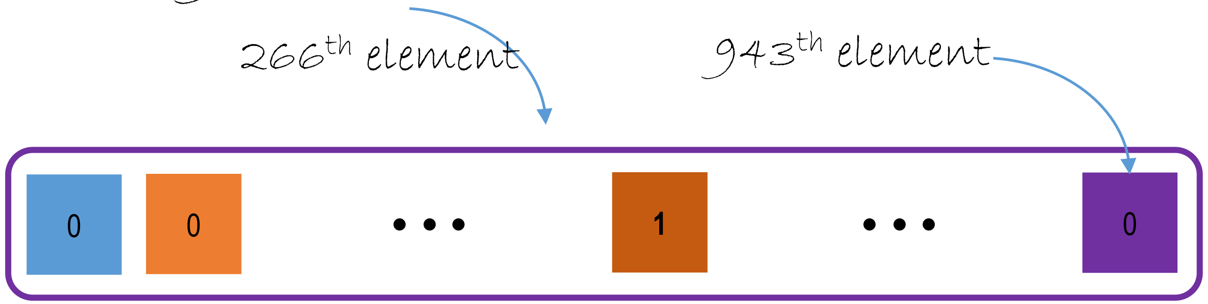
Create a numerical representation of the input words

For MNIST data:



For each word - One-hot representation is a vector with 943 elements

# boston	266:1



For each label – one-hot representation is a vector with 129 elements

Embedding

One-hot Encoding

Numerical representation of text

Word Embedding

Technique to map words or phrases to vector of real numbers.

Maps **one-hot encoded** vector to a lower dimensional space

Linear Embedding

Multiply a matrix with one-hot encoded vector ($\mathbf{W_e}\,\vec{X}^T$)

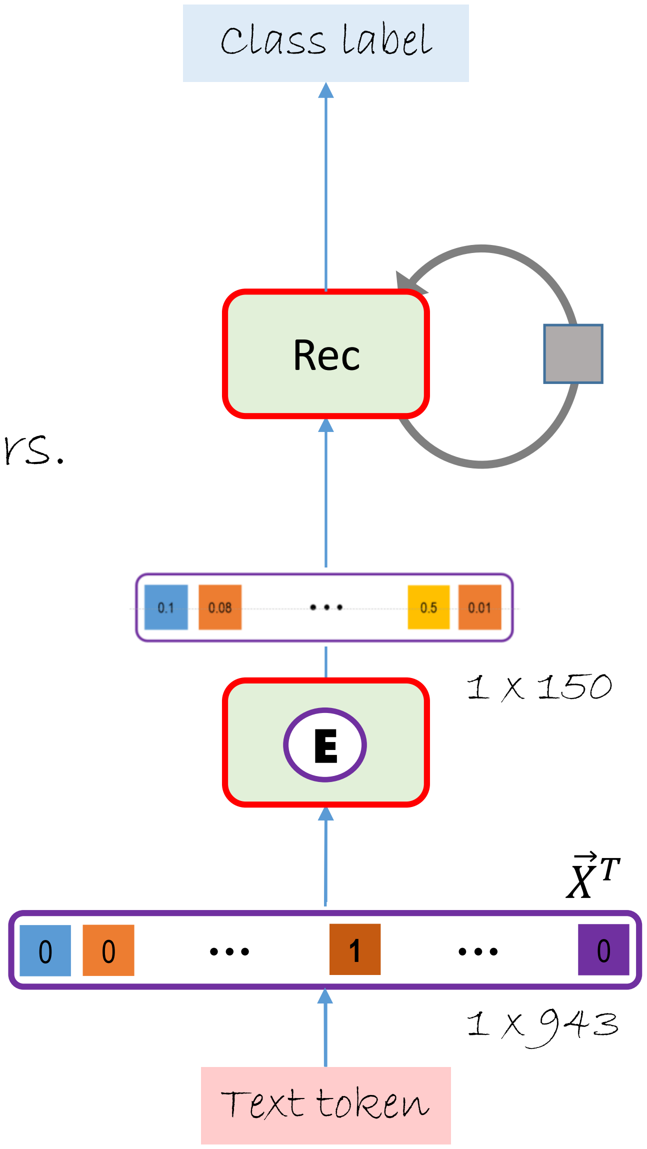
$\vec{X}^T$: vector of size 1 x 943

$\mathbf{W_e}$: matrix of size 150 x 943

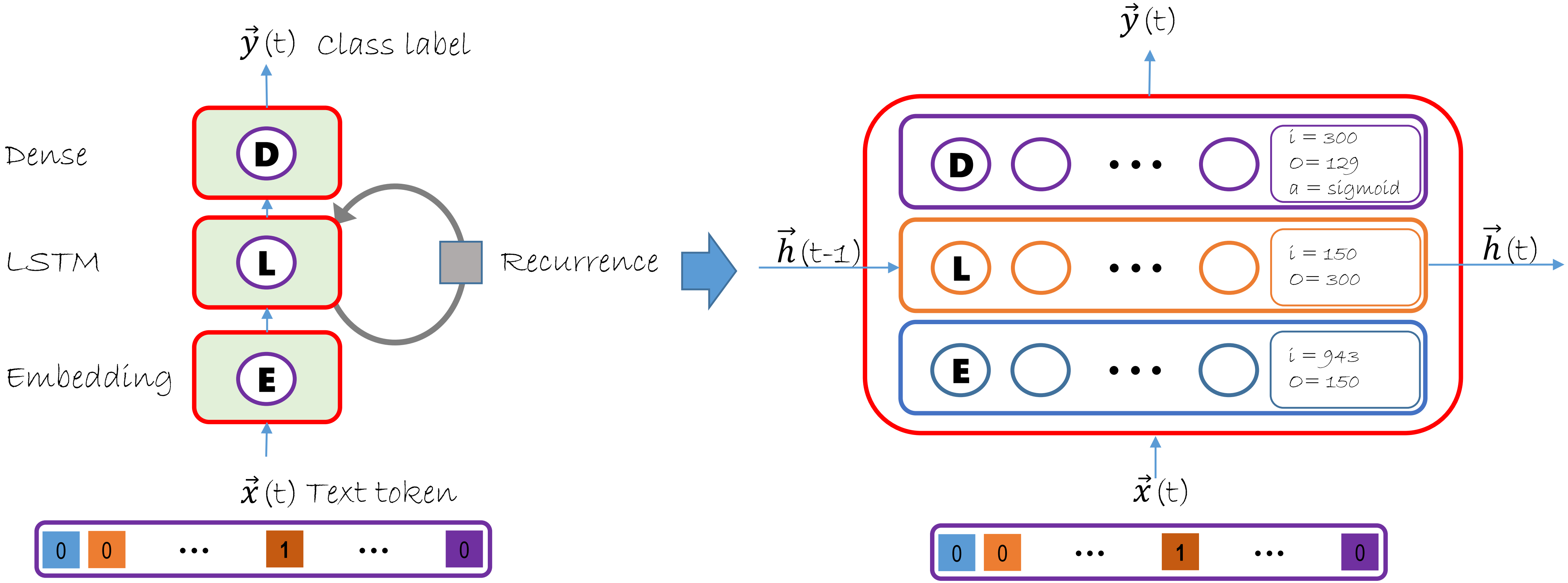
Popular Embedding

Glove (https://en.wikipedia.org/wiki/Glove_(machine_learning))

Word2vec (https://en.wikipedia.org/wiki/Word2vec)



Model



$\vec{y}$(t) Class label

Dense

LSTM

Embedding

D

L

E

Recurrence

$\vec{x}$(t) Text token

0 0 … 1 … 0

$\vec{y}$(t)

$\vec{h}$(t-1)

$\vec{h}$(t)

D … i = 300 O= 129 a = sigmoid

L … i = 150 O= 300

E … i = 943 O= 150

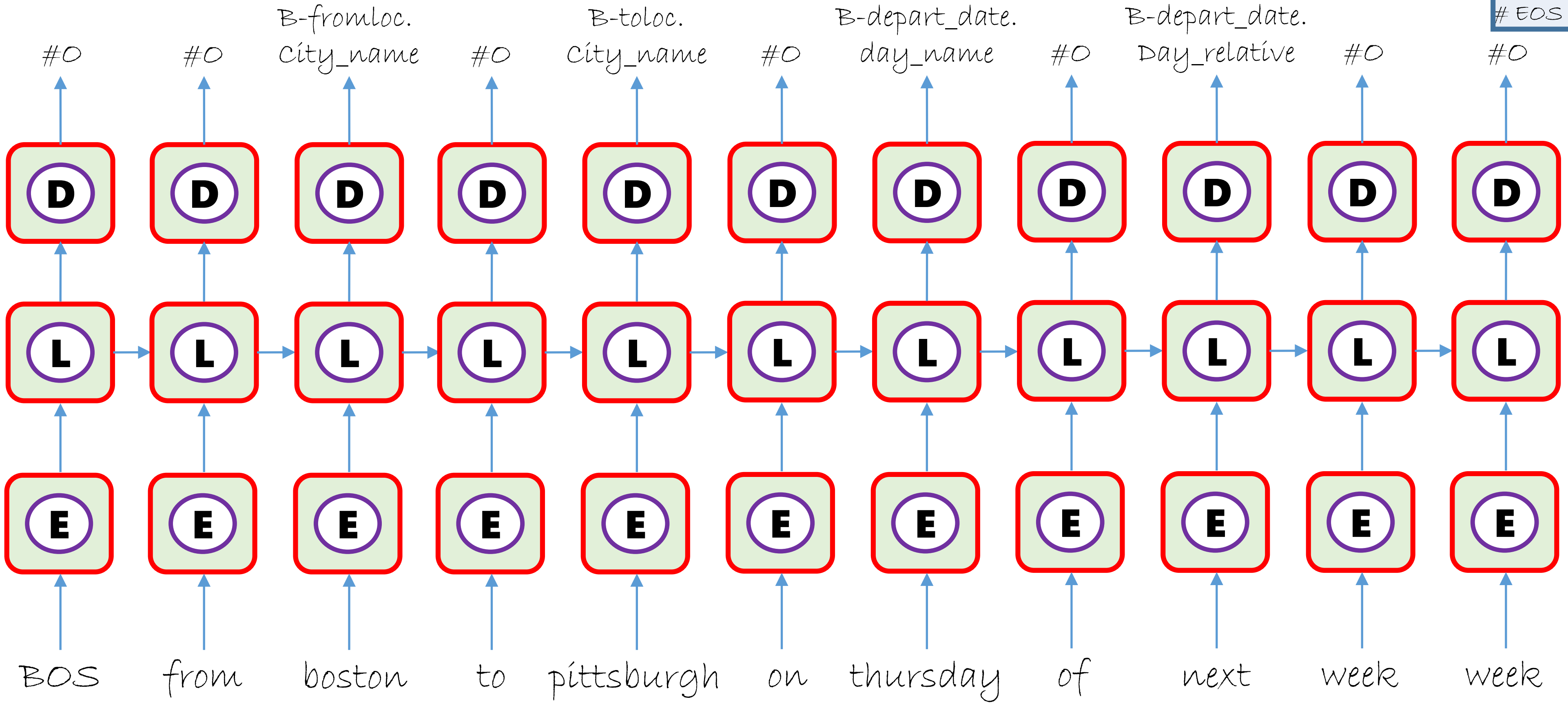
$\vec{x}$(t)

0 0 … 1 … 0

Text classification

Problem: Tagging entities in Air Traffic Controller (ATIS) data

# BOS	# O
# from	# O
# boston	# B-fromloc.city_name
# to	# O
# pittsburgh	# B-toloc.city_name
# on	# O
# thursday	# B-depart_date.day_name
# of	# O
# next	# B-depart_date.date_relative
# week	# O
# EOS	# O



Text classification

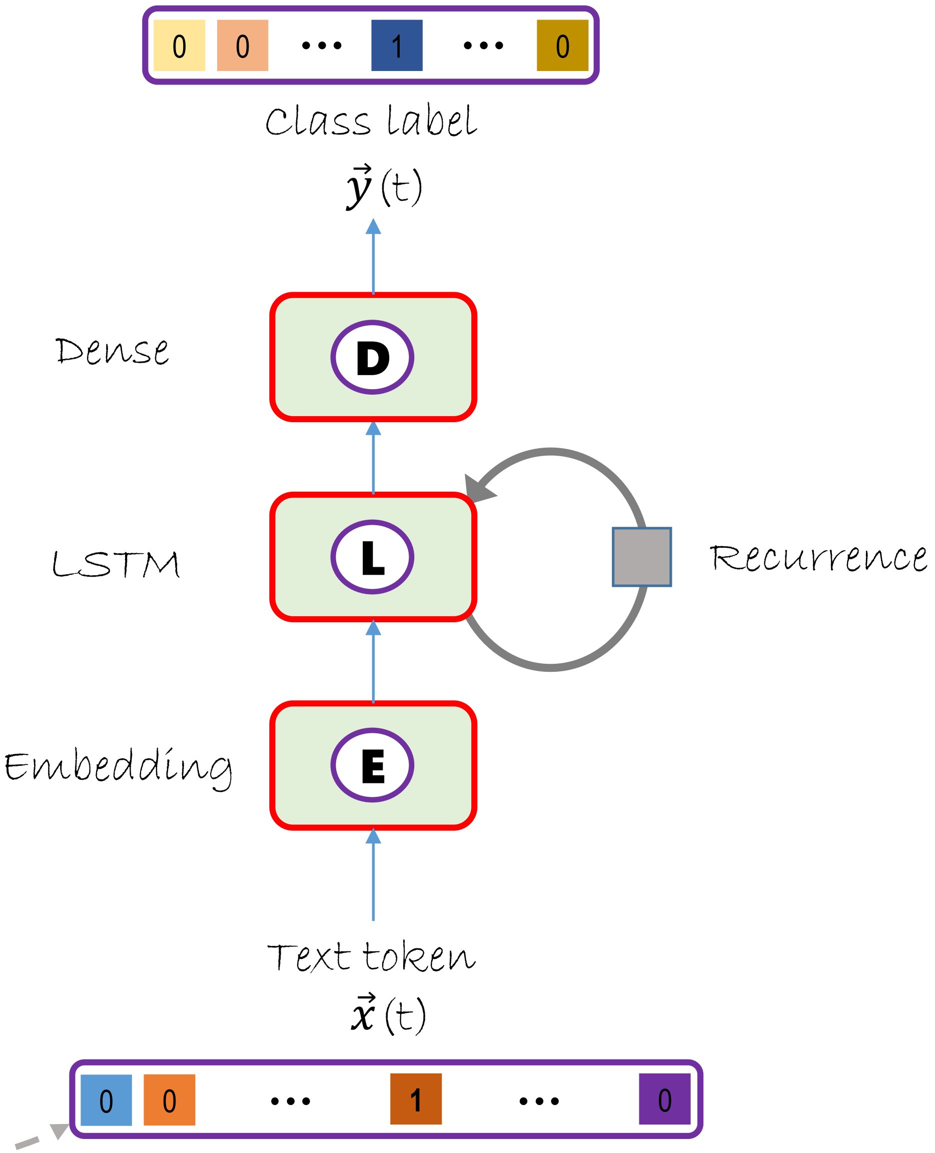
Problem: Tagging entities in Air Traffic Controller (ATIS) data

# BOS	# O
# from	# O
# boston	# B-fromloc.city_name
# to	# O
# pittsburgh	# B-toloc.city_name
# on	# O
# thursday	# B-depart_date.day_name
# of	# O
# next	# B-depart_date.date_relative
# week	# O
# EOS	# O

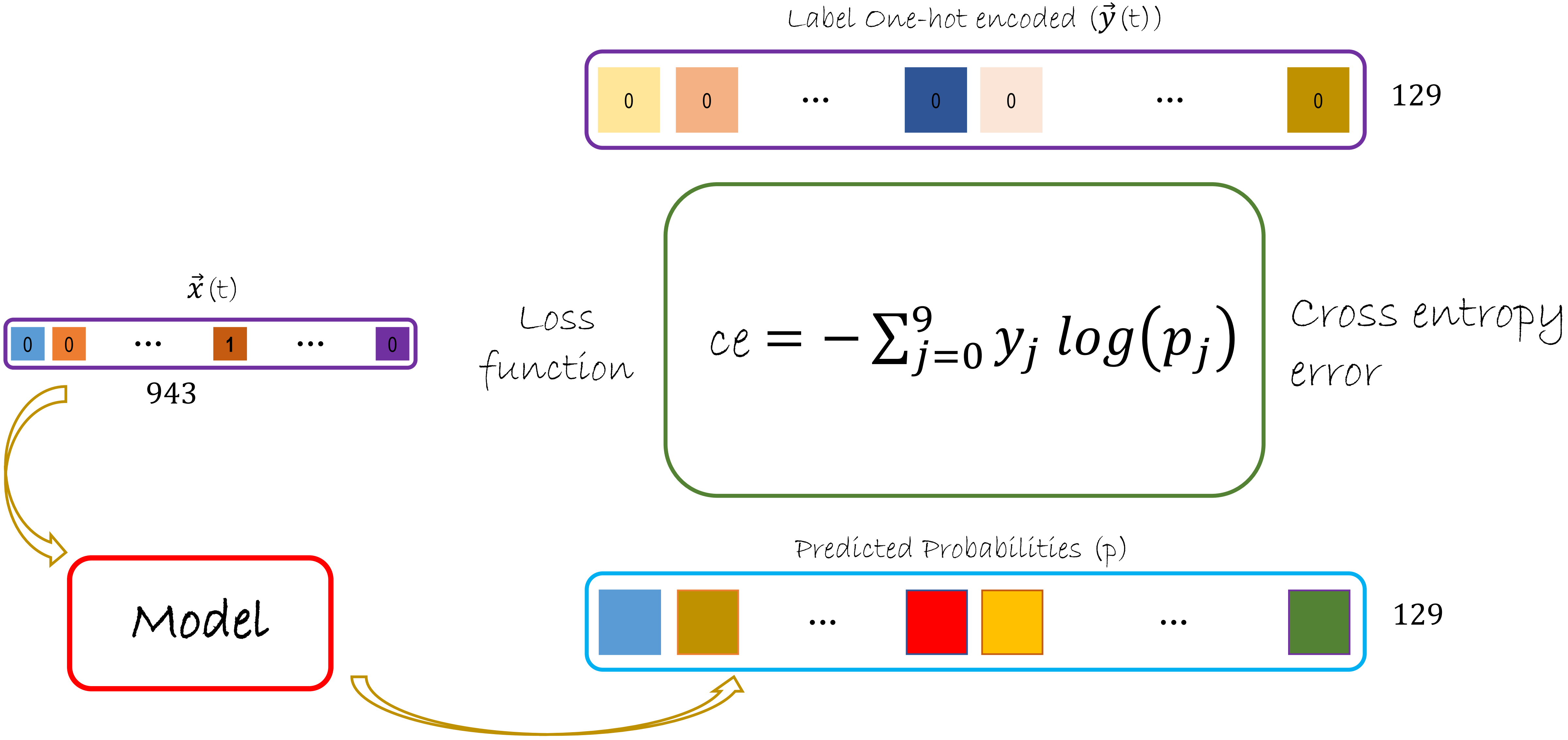
```
'BOS from boston to Pittsburgh on Thursday of next week EOS'
```

Input feature (1 x 11 x (1x943))

#1



Error or Loss Function



Label One-hot encoded ($\vec{y}$(t))

0 0 … 0 0 … 0

129

$\vec{x}$(t)

0 0 … 1 … 0

943

Loss function

$$ce = -\sum_{j=0}^{9} y_j \, log(p_j)$$

Cross entropy error

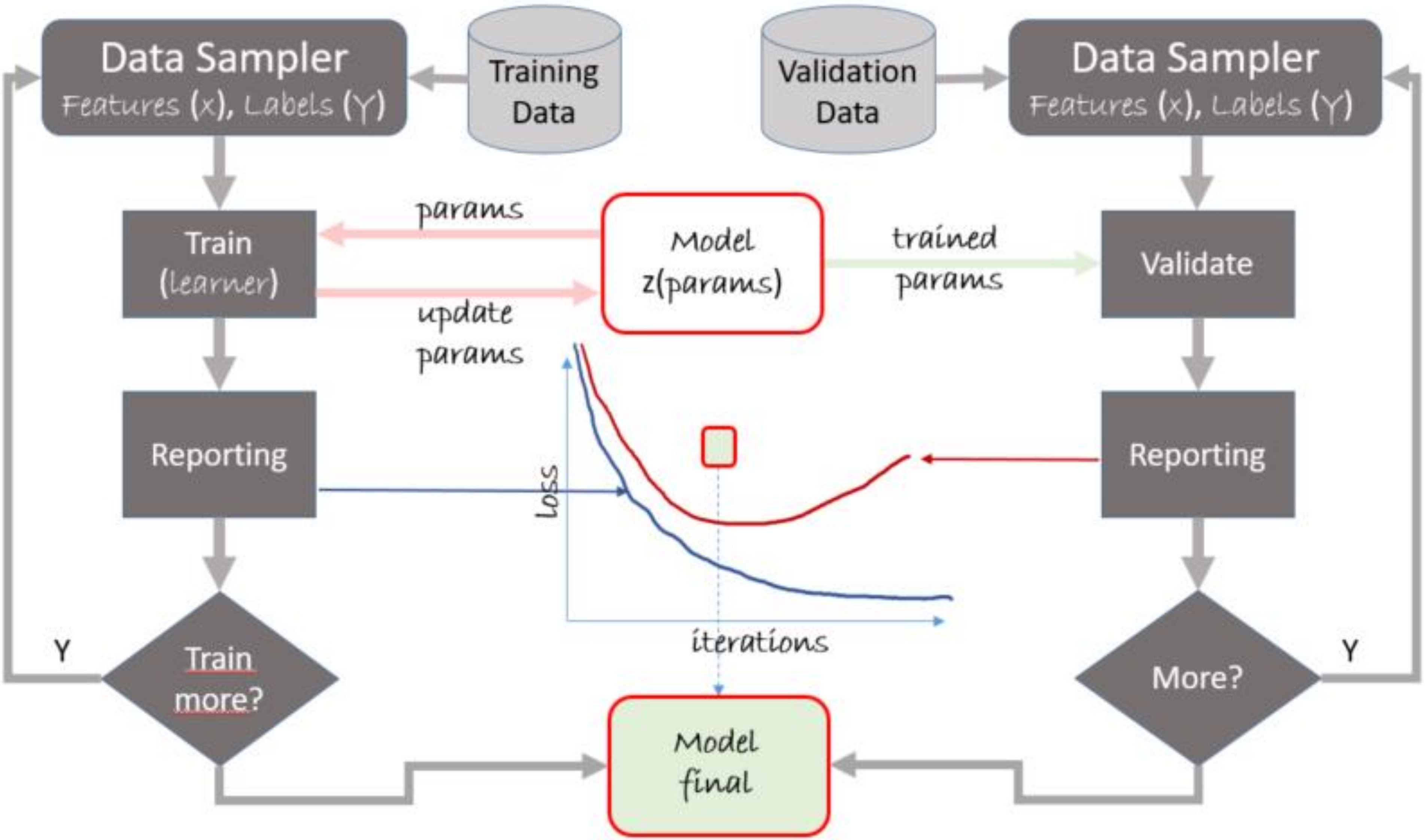
Model

Predicted Probabilities (p)

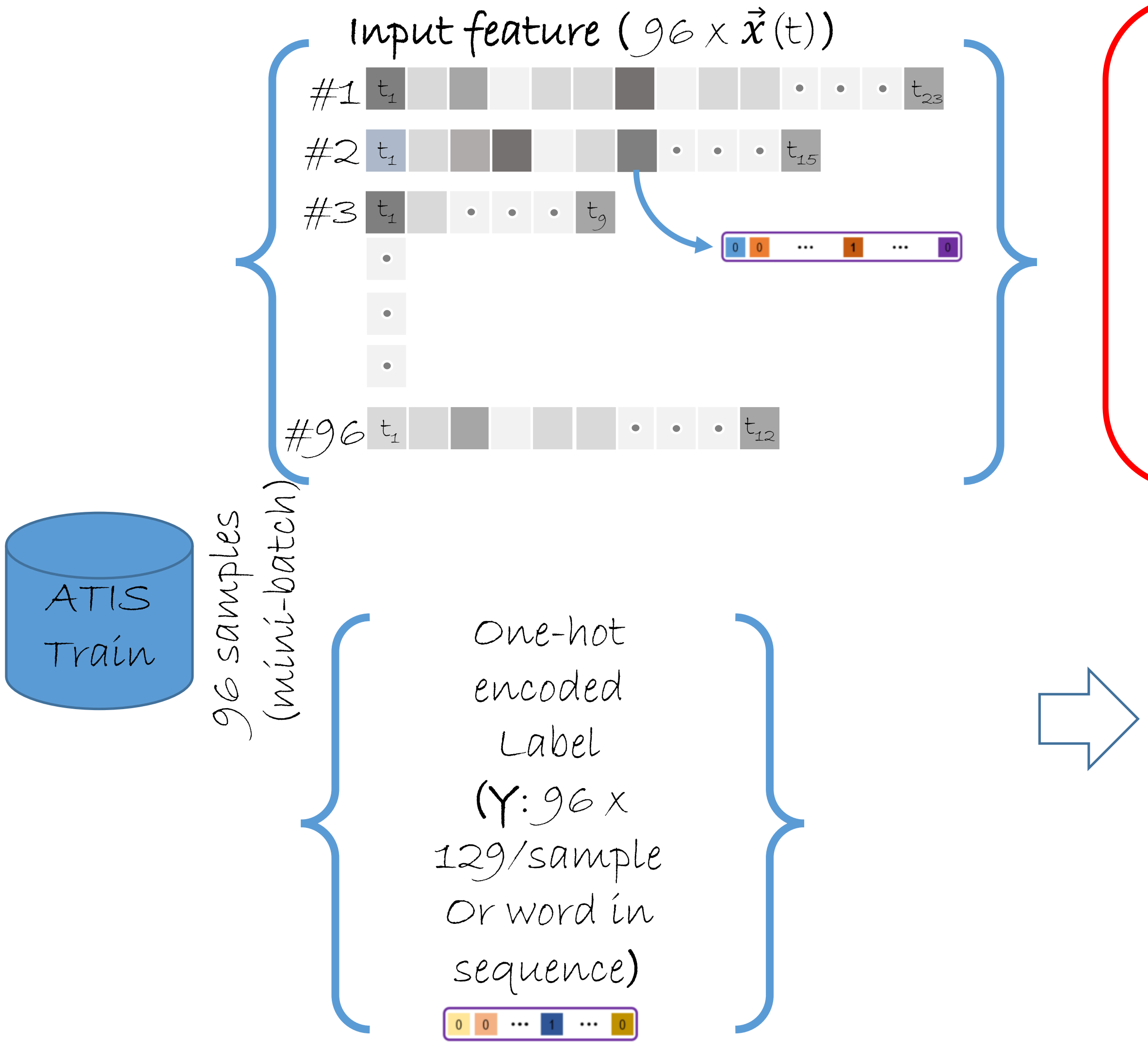
… …

129

Train / Validation Workflow



Train Workflow



```
z = model():
        return
          Sequential([
              Embedding(emb_dim=150),
              Recurrence(LSTM(hidden_dim=300),
                         go_backwards=False),
              Dense(num_labels = 129)
          ])
```

Loss `cross_entropy_with_softmax(Z,Y)`

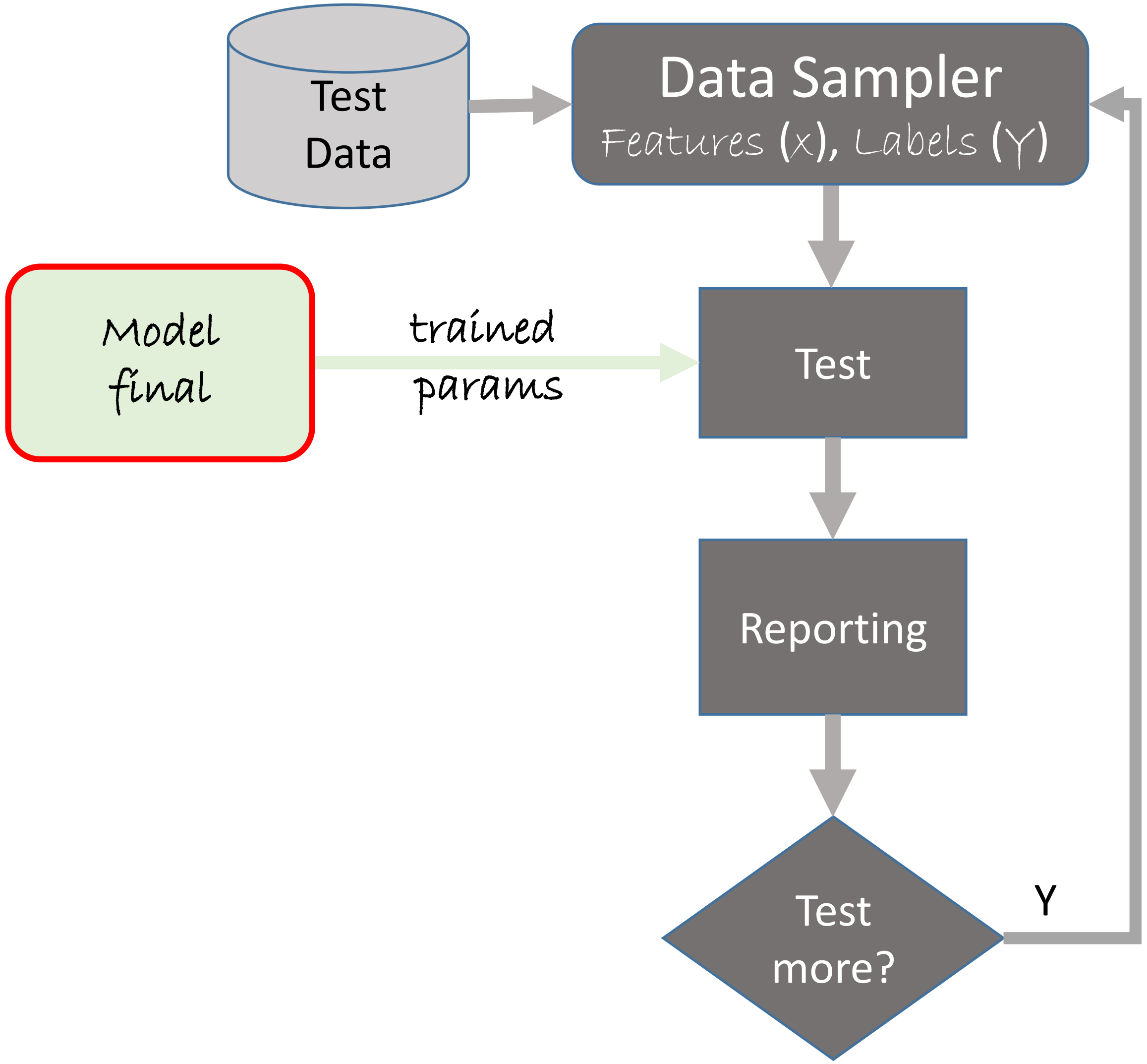
Error `classification_error(Z,Y)`

Trainer(model, (loss, error), learner)

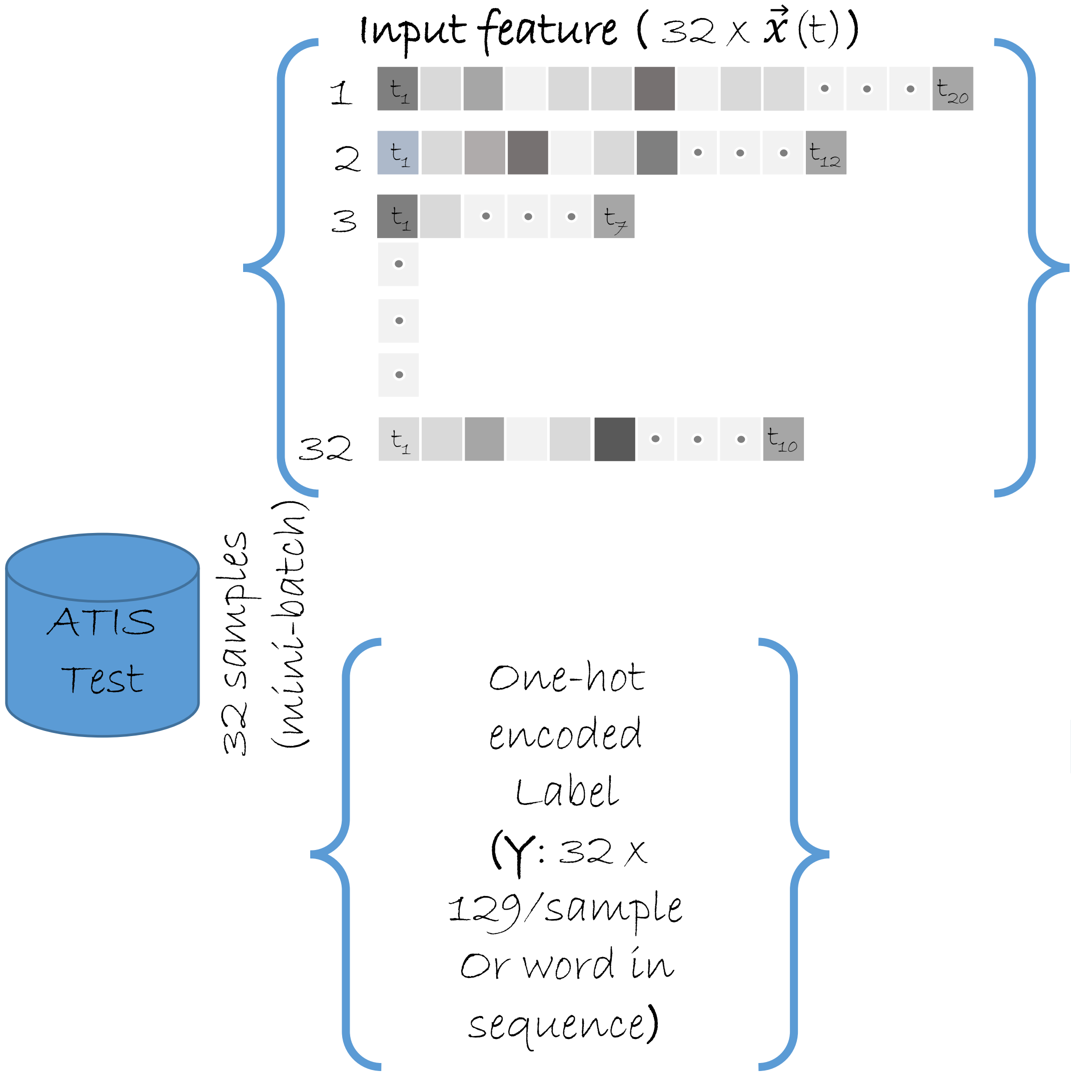
Trainer.**train**_minibatch({X, Y})

Learner
Adam, adagrad etc, are solvers to estimate

Test workflow



Test workflow



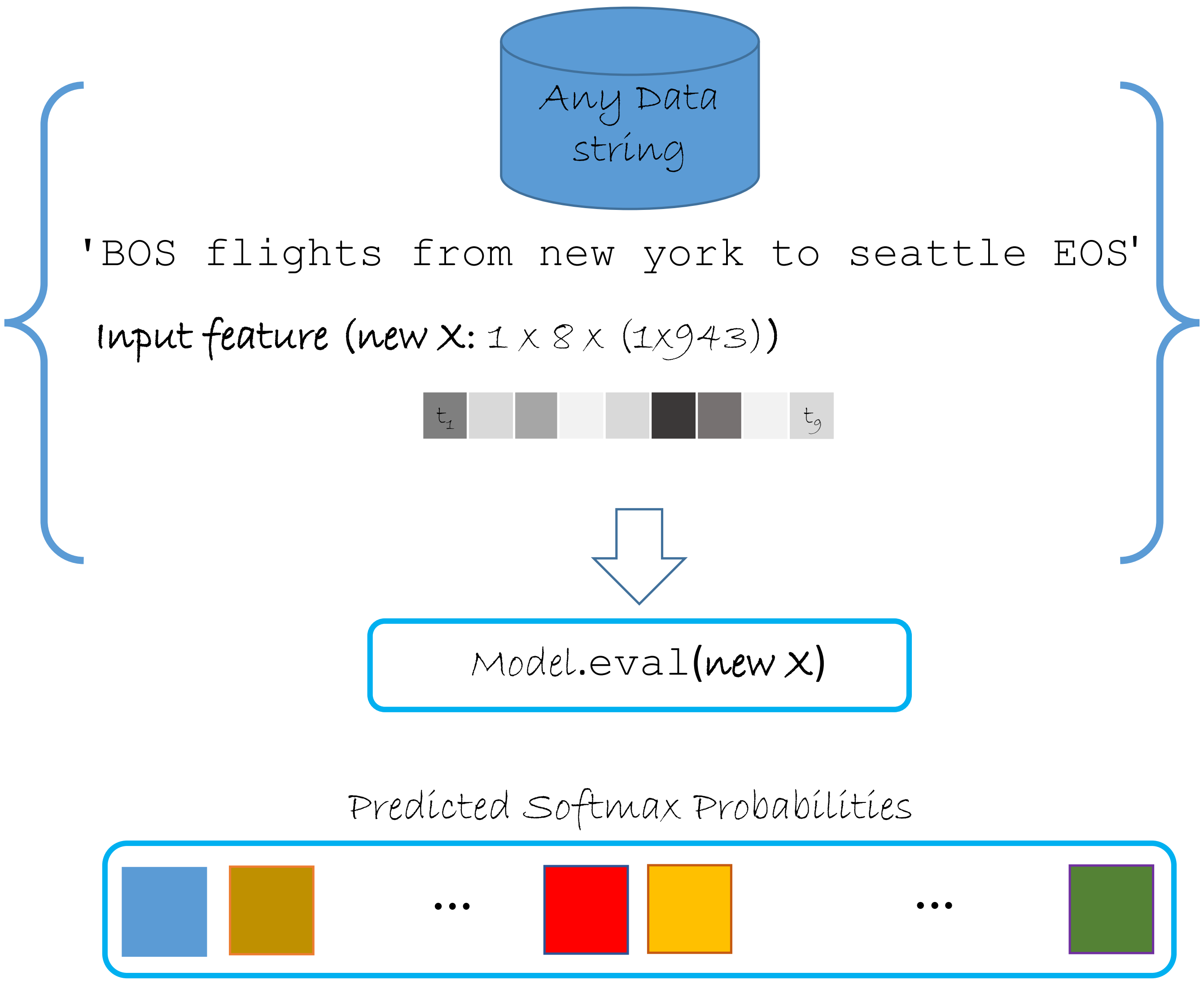
One-hot encoded Label (Y: 32 x 129/sample Or word in sequence)

```
z = model():
        return
           Sequential([
                 Embedding(emb_dim=150),
                 Recurrence(LSTM(hidden_dim=300),
                                  go_backwards=False),
                 Dense(num_labels = 129)
           ])
```

```
Trainer.test_minibatch({X, Y})
```

Returns the classification error as % incorrectly labeled tokens.

Prediction workflow



Any Data string

'BOS flights from new york to seattle EOS'

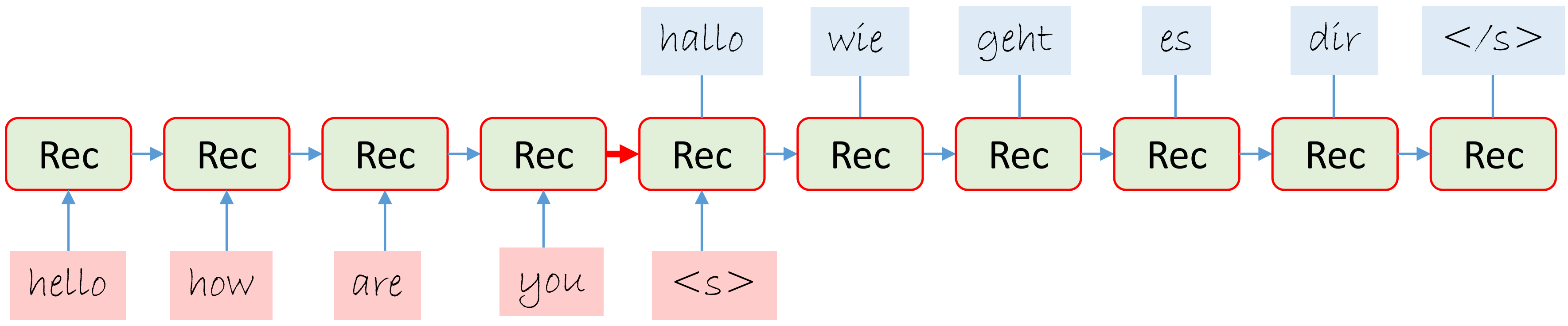
Input feature (new X: 1 x 8 x (1x943))

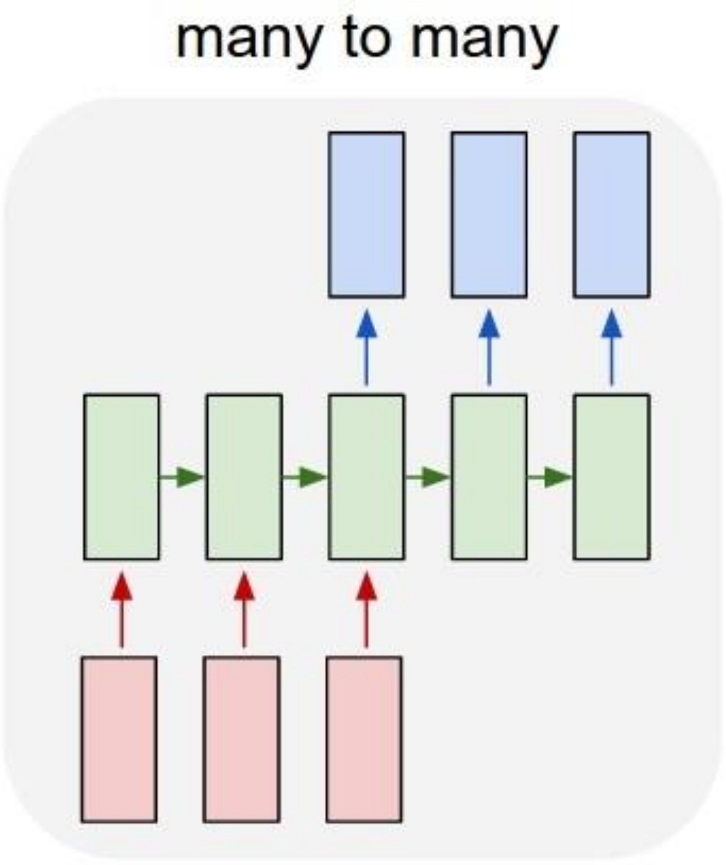
Model.eval(new X)

Predicted Softmax Probabilities

Output prediction (1 x 8 x (1x129))

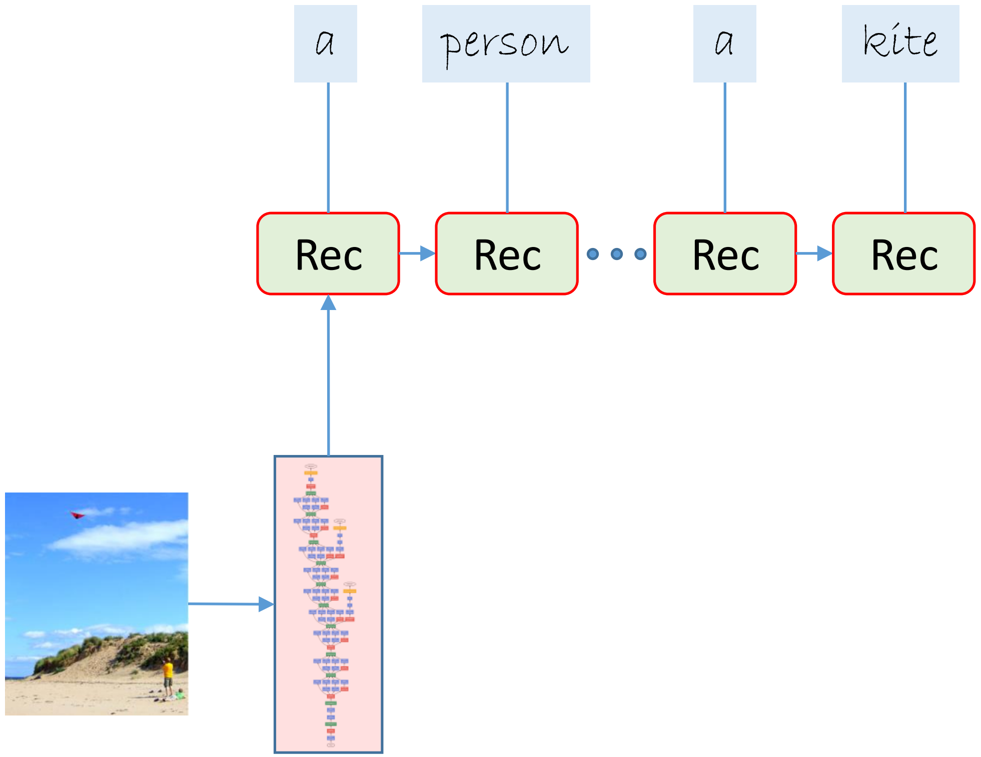
Sequences (many to many)





http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Sequences (one to many)

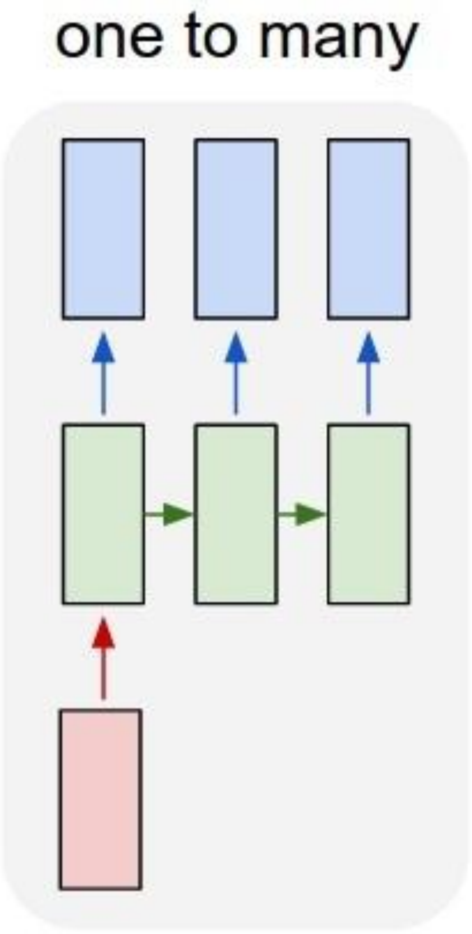


A person on a beach flying a kite.

A black and white photo of a train on a train track.

A person skiing down a snow covered slope.

A group of giraffe standing next to each other.



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Vinyals et al (https://arxiv.org/abs/1411.4555)

Conclusion

Deep learning concepts

- Loss functions, Mini-batch
- Activation functions
- Convolution, Pooling
- Recurrence, LSTM, Dropout, Embeddings

Deep neural networks models

- Multi-class logistic regression
- Multi-layered perceptron
- Convolutional neural networks
- Recurrent networks with LSTM
- Recurrent networks with LSTM and word embeddings

Train-Test-Predict using DNN models