Kissipo Learning for Deep Learning

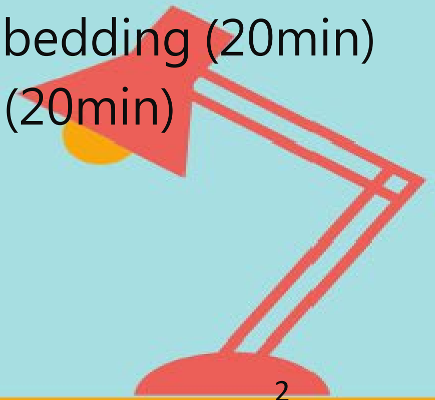
Topic 16: RNN with PyTorch (20min)

Hsueh-Ting Chu

KLDL-W6-T16

Topics

- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: Kissipo Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with PyTorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)



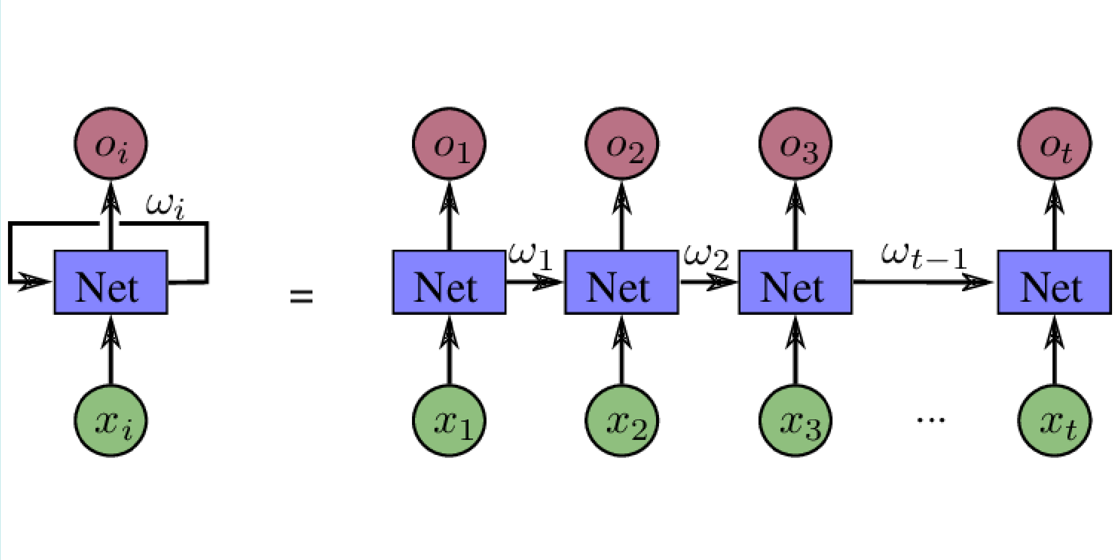
Week 6 Topics

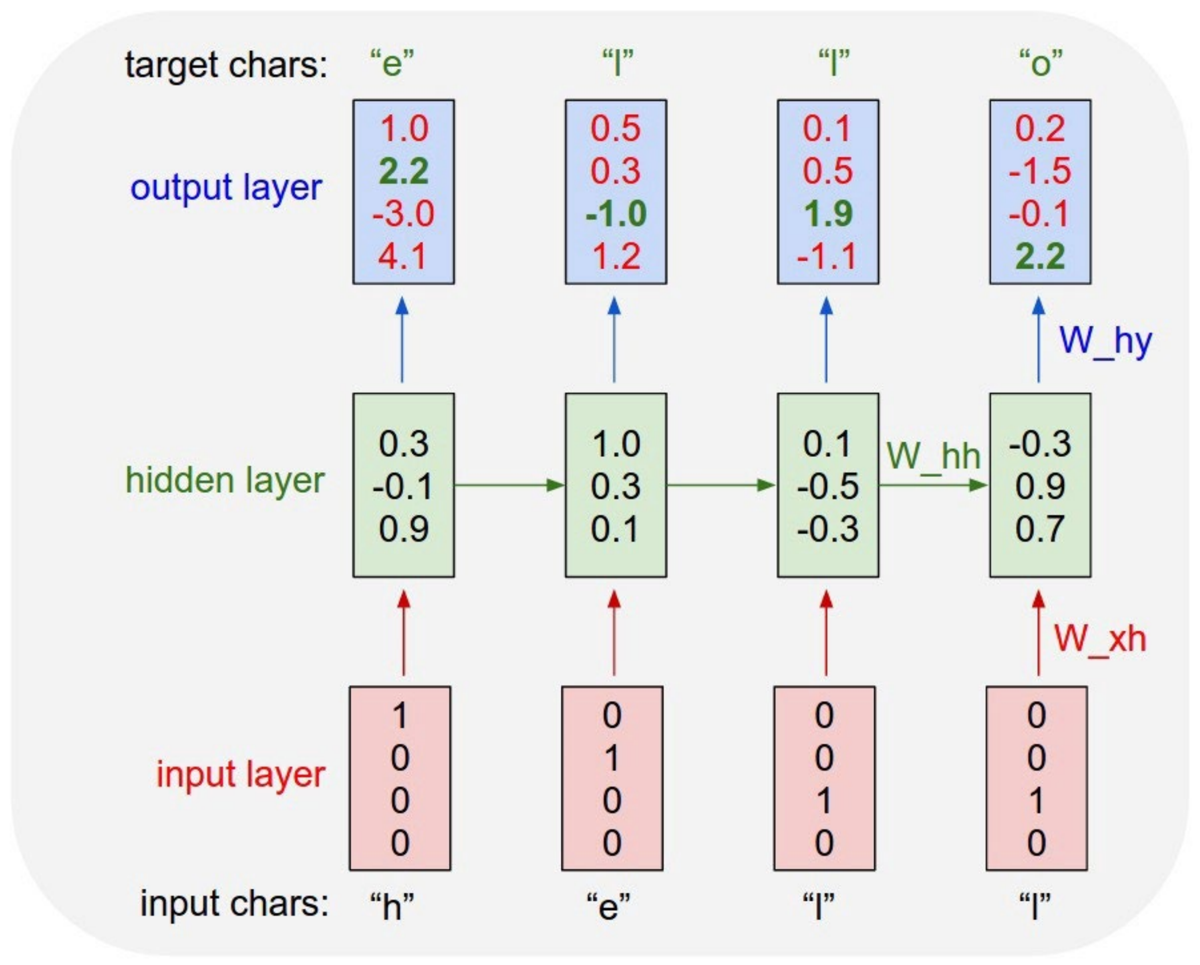
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)

Topic 16 Content

- Topic 16: PyTorch RNN (20min)
  - Recurrent Neural Network using PyTorch
  - Define a PyTorch RNN models
  - Train a PyTorch RNN models

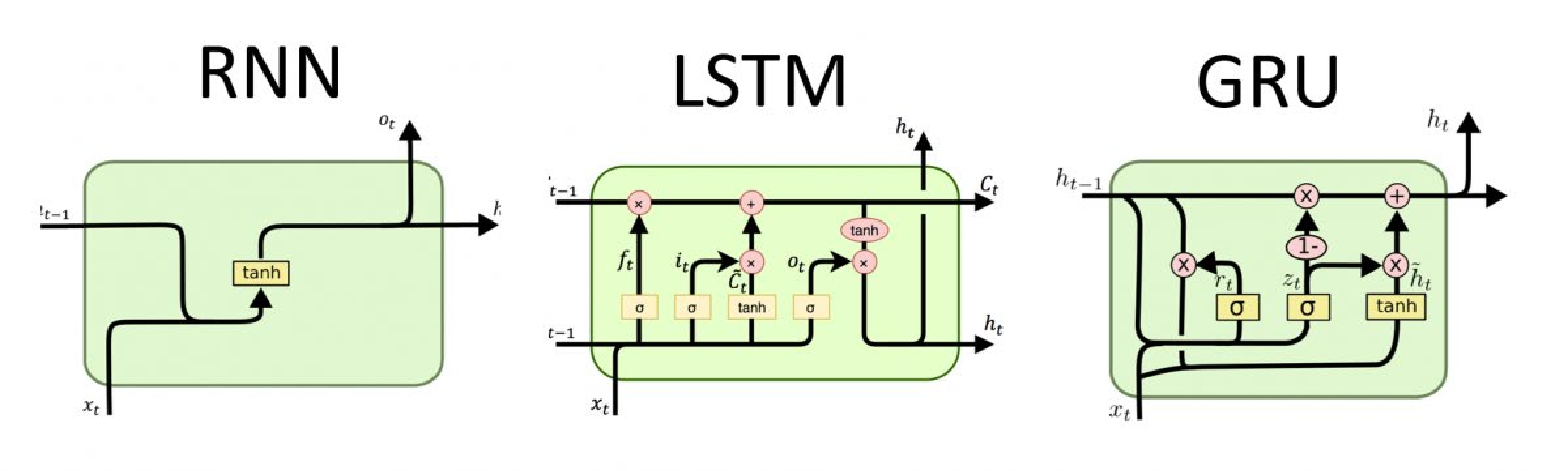
Recurrent Neural Network (RNN)





RNN (Recurrent Neural Network) is a type of neural network characterized by a recurrent structure that allows information to be retained for use throughout the sequence when processing sequence data. This makes RNNs suitable for processing speech recognition, text generation, machine translation and other tasks.

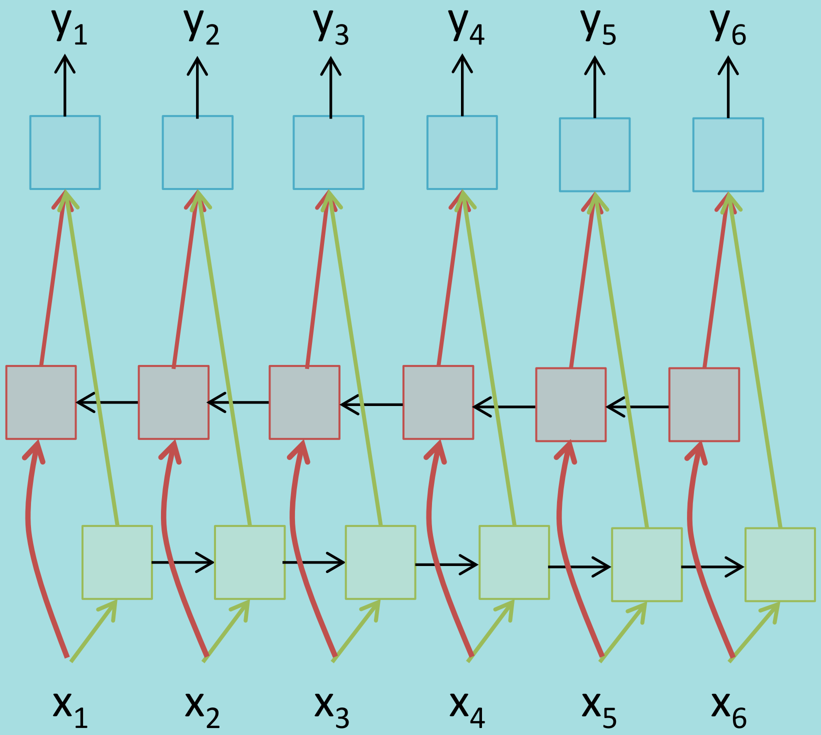
GRU RNNs



RNN
o_t
h_{t-1}
tanh
h_t
x_t
LSTM
h_t
C_{t-1}
C_t
×
+
tanh
f_t
i_t
$\tilde{C}_t$
o_t
σ
σ
tanh
σ
h_{t-1}
h_t
x_t
GRU
h_t
h_{t-1}
X
+
1-
X
X
r_t
z_t
$\tilde{h}_t$
σ
σ
tanh
x_t

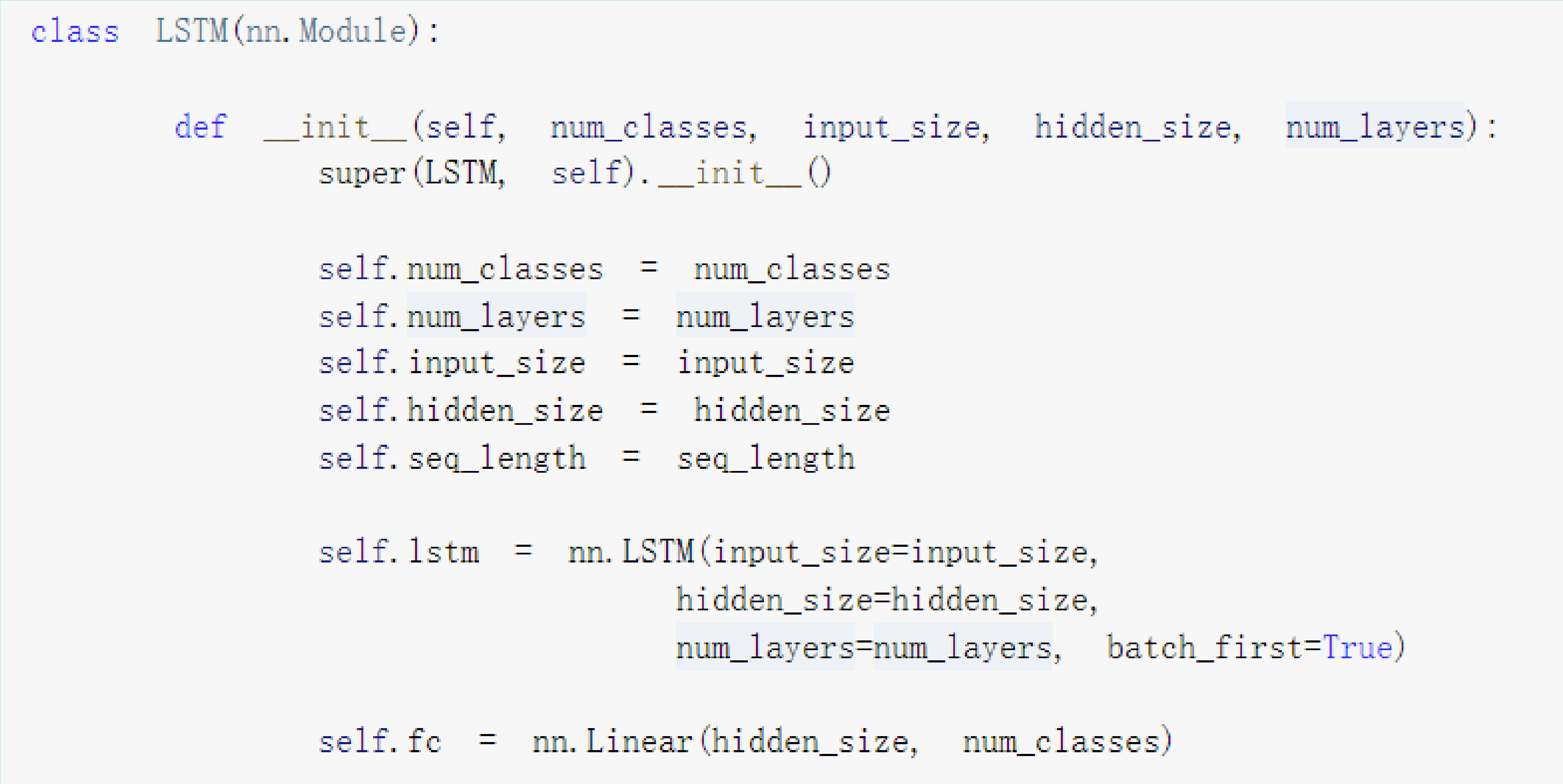
Bi-directional RNNs

- RNNs can process the input sequence in forward and in the reverse direction



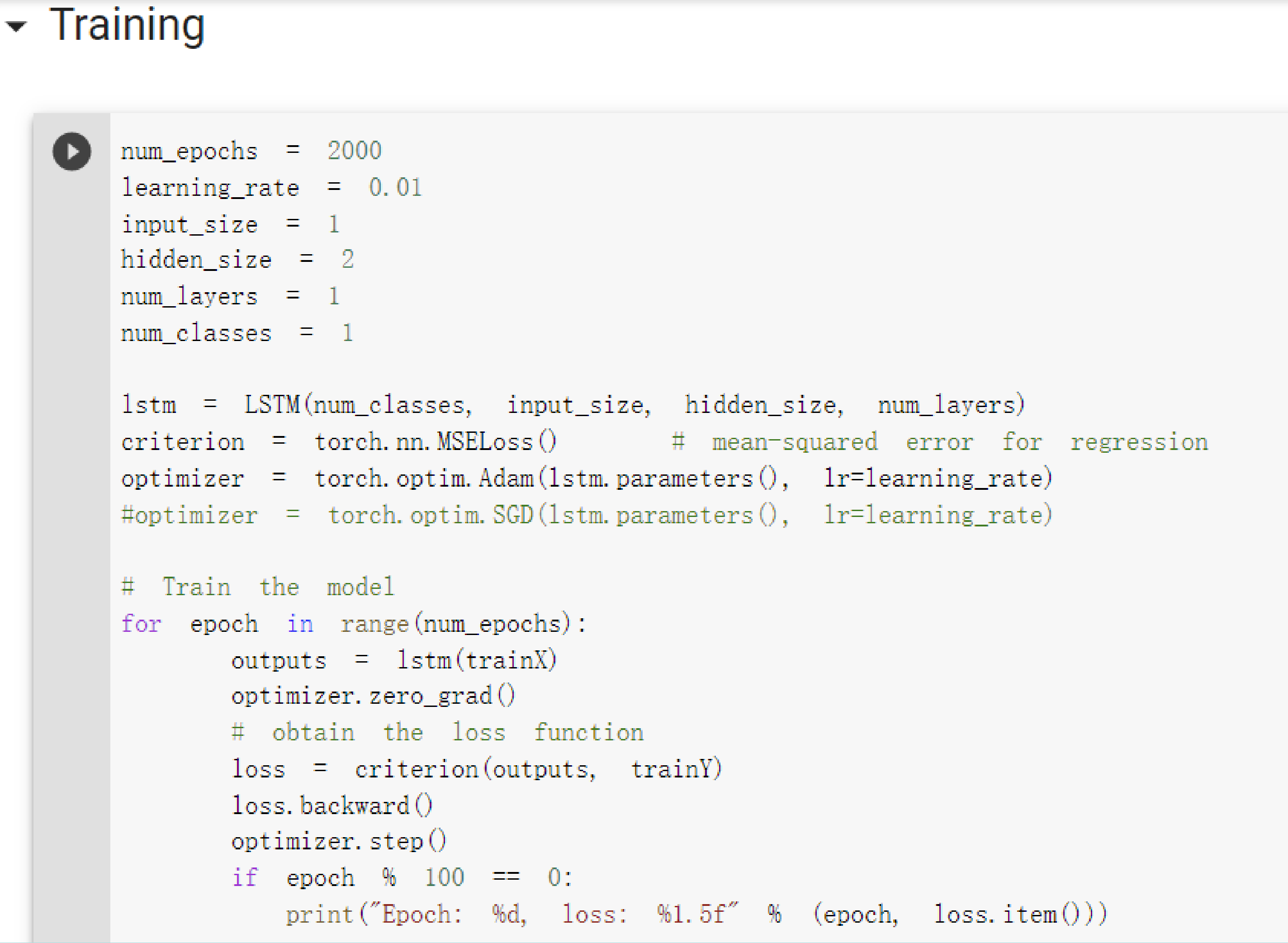
- Popular in speech recognition

Define a LSTM model



```
class LSTM(nn.Module):

    def __init__(self, num_classes, input_size, hidden_size, num_layers):
        super(LSTM, self).__init__()

        self.num_classes = num_classes
        self.num_layers = num_layers
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.seq_length = seq_length

        self.lstm = nn.LSTM(input_size=input_size,
                            hidden_size=hidden_size,
                            num_layers=num_layers, batch_first=True)

        self.fc = nn.Linear(hidden_size, num_classes)
```

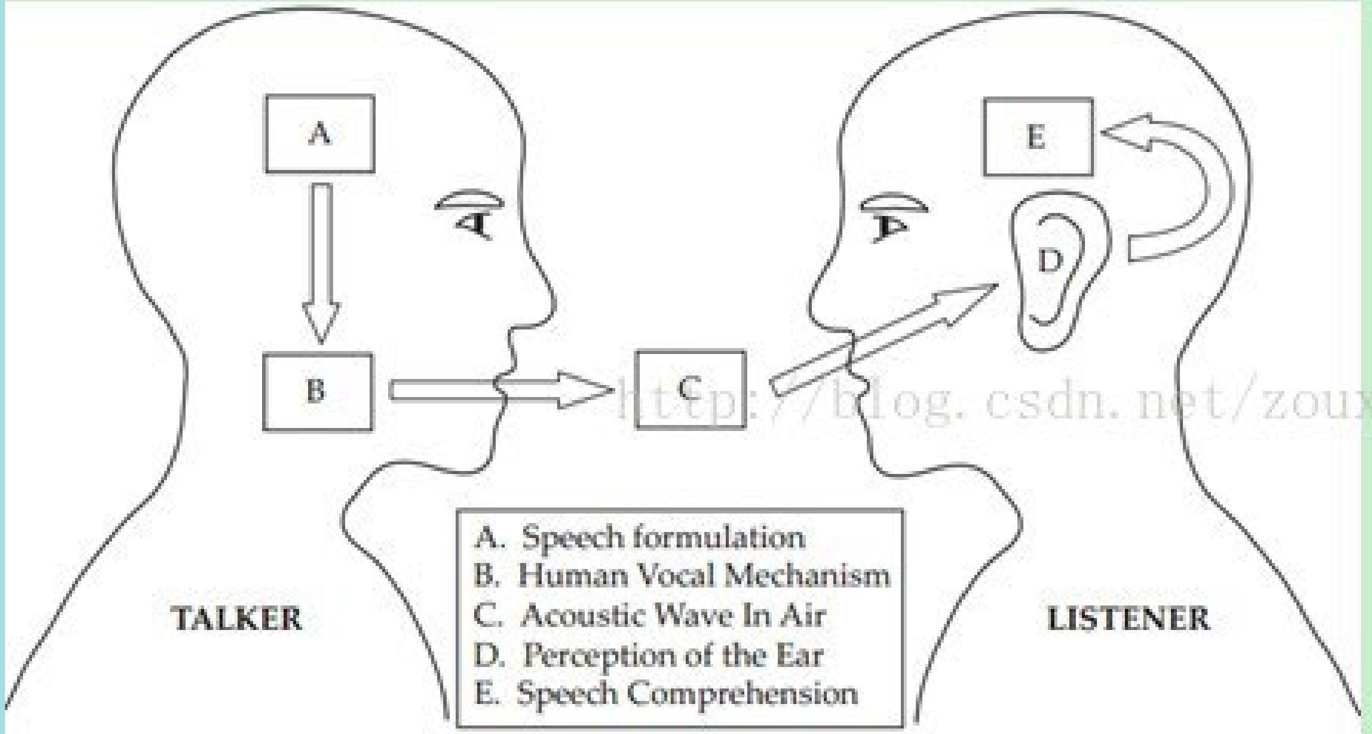
Train the LSTM model

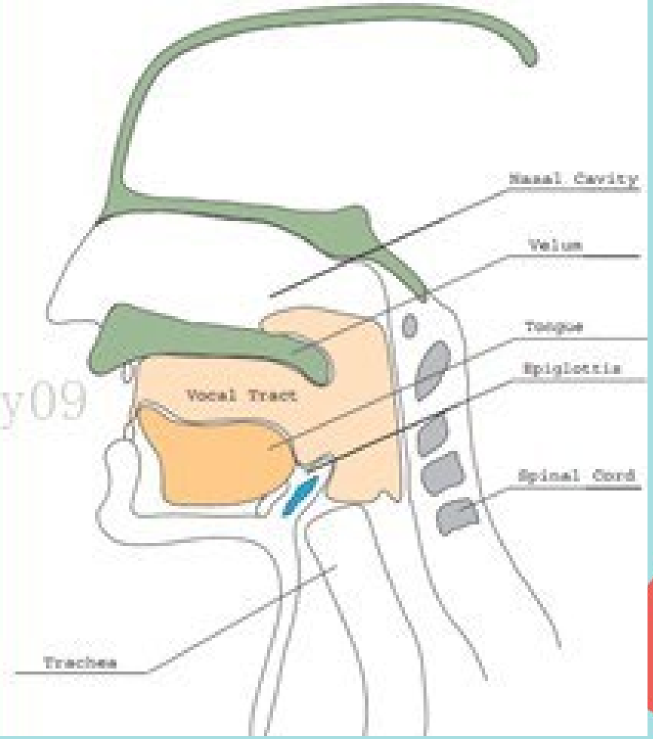


Training

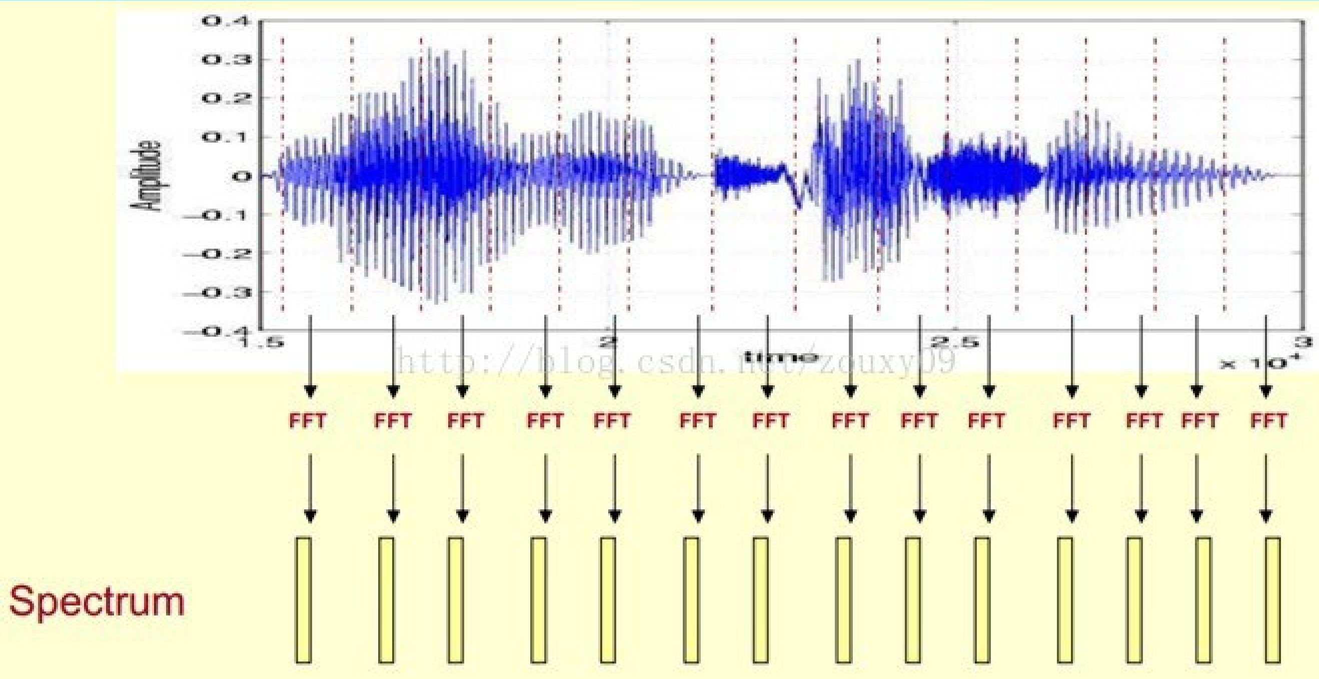
```
num_epochs = 2000
learning_rate = 0.01
input_size = 1
hidden_size = 2
num_layers = 1
num_classes = 1

lstm = LSTM(num_classes, input_size, hidden_size, num_layers)
criterion = torch.nn.MSELoss()    # mean-squared error for regression
optimizer = torch.optim.Adam(lstm.parameters(), lr=learning_rate)
#optimizer = torch.optim.SGD(lstm.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    outputs = lstm(trainX)
    optimizer.zero_grad()
    # obtain the loss function
    loss = criterion(outputs, trainY)
    loss.backward()
    optimizer.step()
    if epoch % 100 == 0:
      print("Epoch: %d, loss: %1.5f" % (epoch, loss.item()))
```

Automatic speech

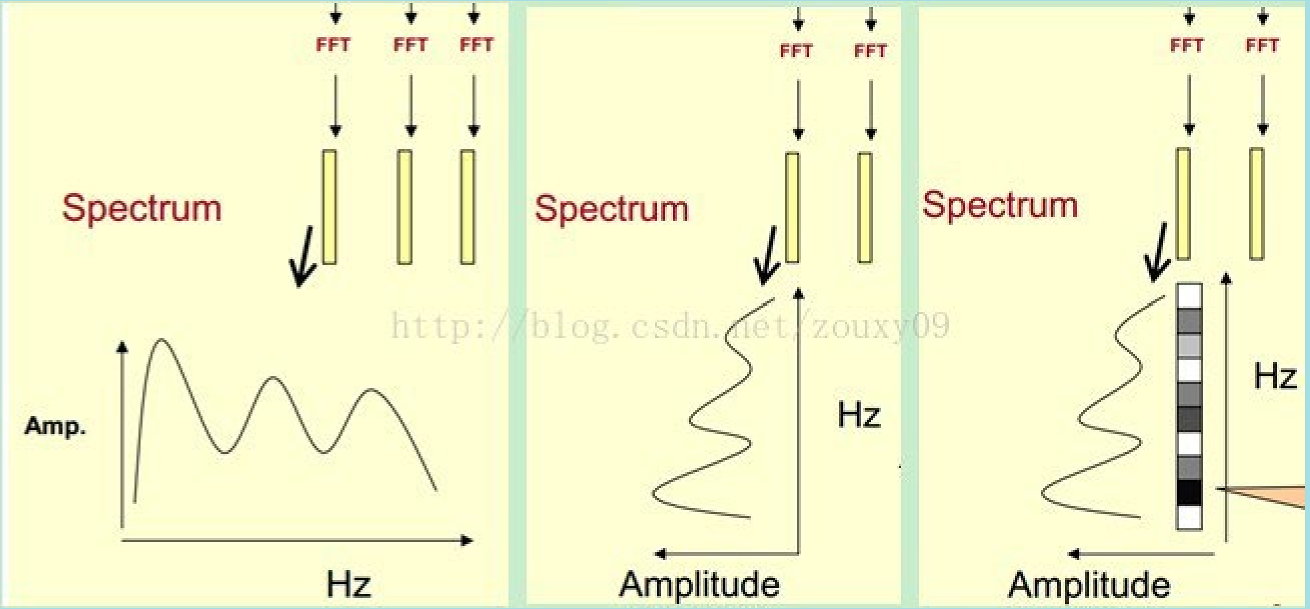




聲譜圖（Spectrogram）

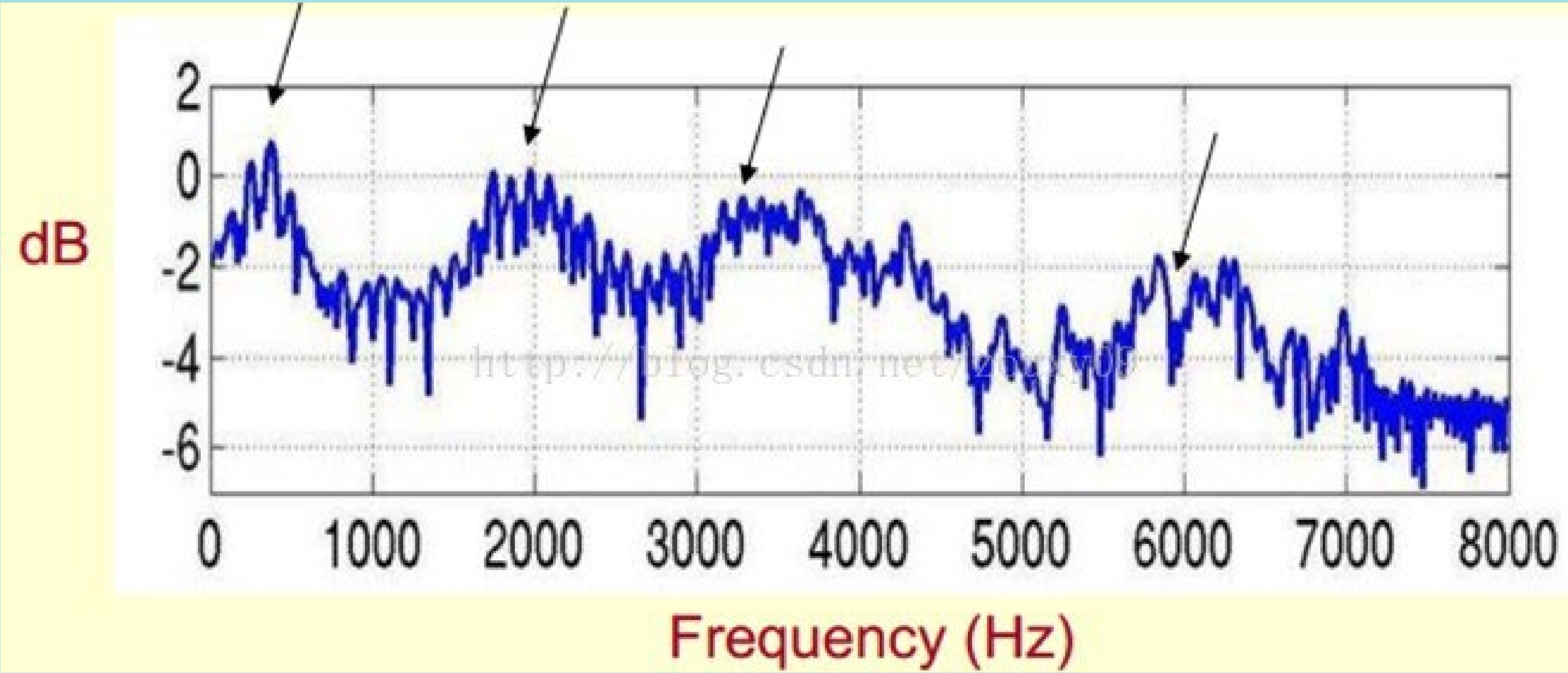


頻譜種類

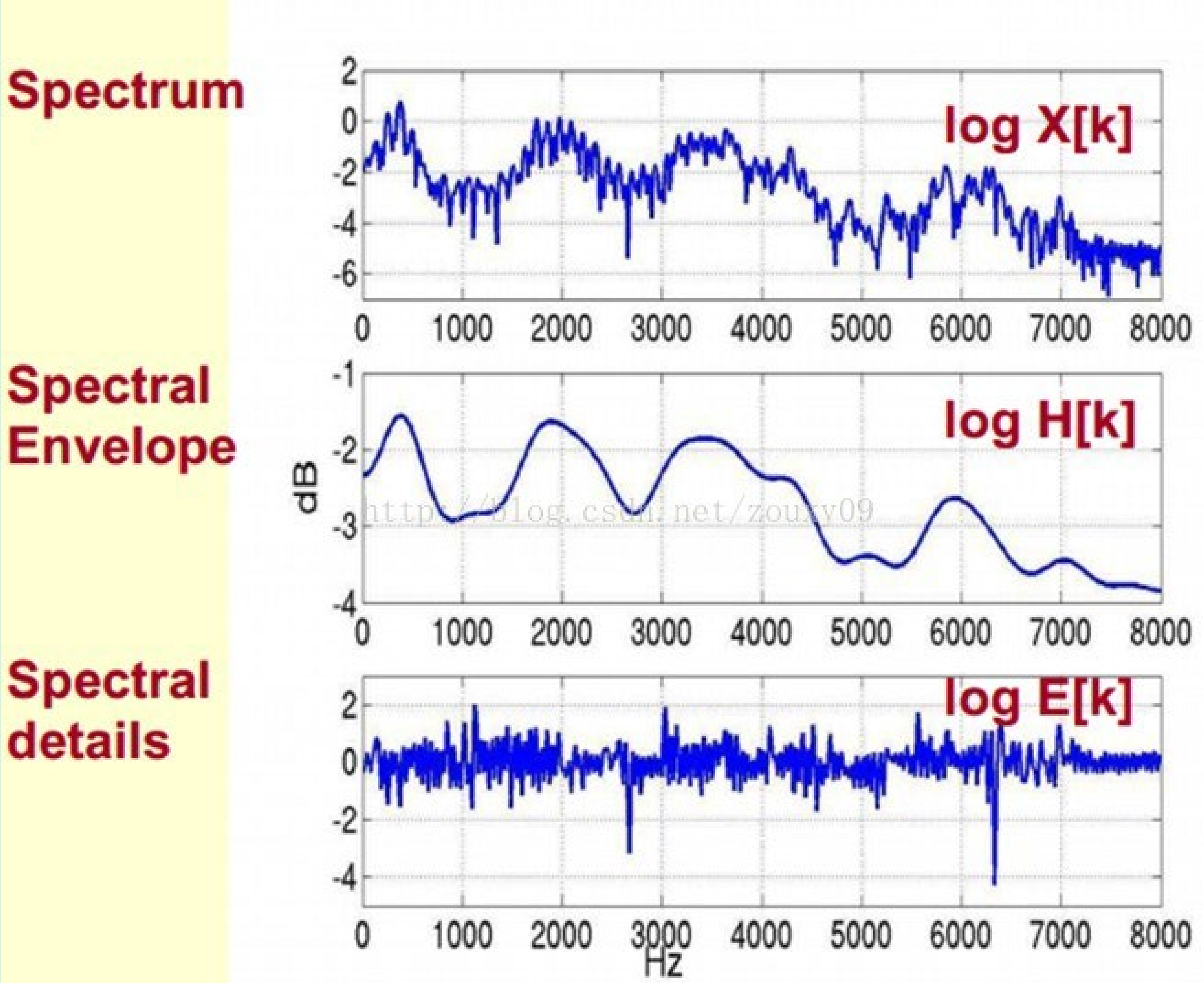


頻譜圖有三種，即線性振幅譜、對數振幅譜、自功率譜

倒譜分析（Cepstrum Analysis）

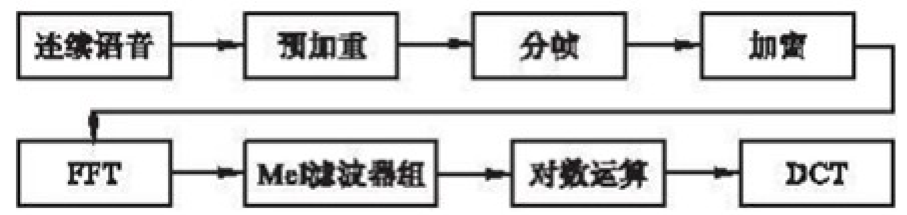


Mel-frequency cepstral coefficients(MFCC)/ 梅爾倒頻譜



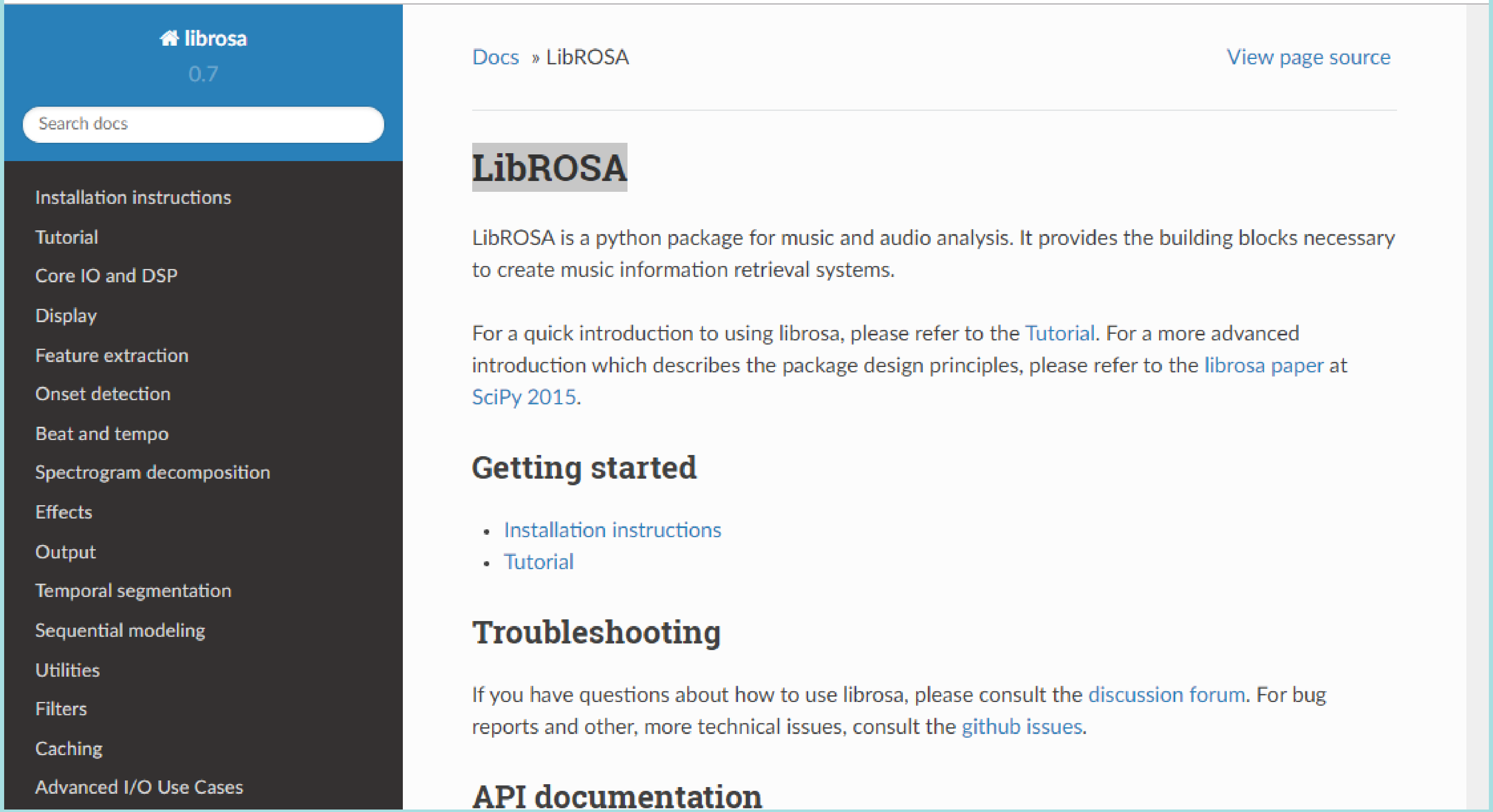
包絡和頻譜的細節

$$mel(f) = 2595 * \log_{10}(1 + f / 700)$$



连续语音 → 预加重 → 分帧 → 加窗 → FFT → Mel滤波器组 → 对数运算 → DCT

LibROSA



LibROSA

LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

For a quick introduction to using librosa, please refer to the Tutorial. For a more advanced introduction which describes the package design principles, please refer to the librosa paper at SciPy 2015.

Getting started

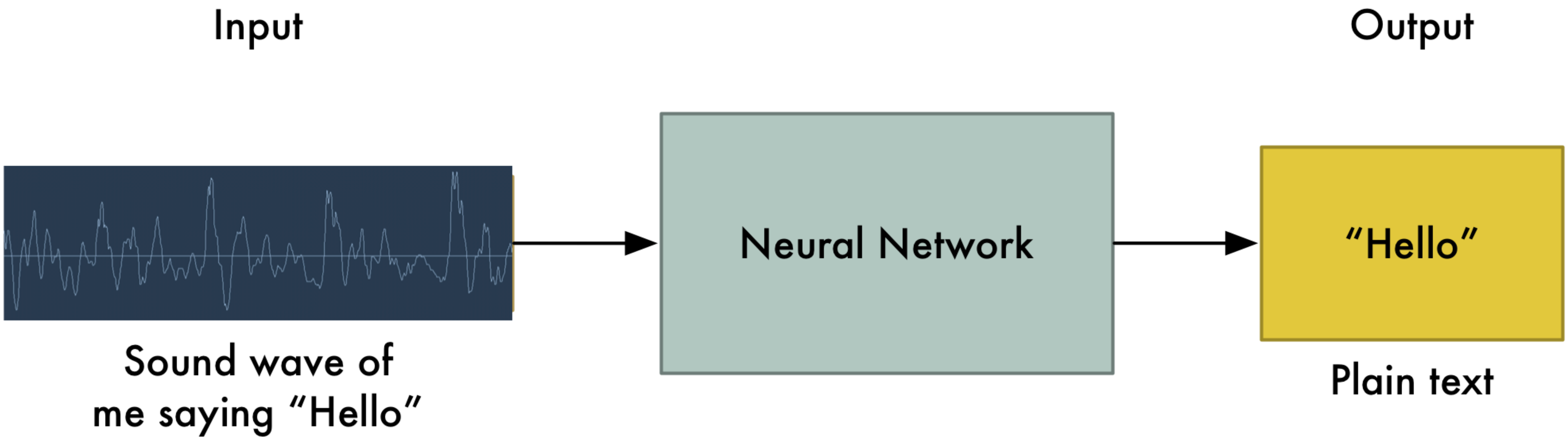
- Installation instructions
- Tutorial

Troubleshooting

If you have questions about how to use librosa, please consult the discussion forum. For bug reports and other, more technical issues, consult the github issues.

API documentation

Speech Recognition



Input

Output

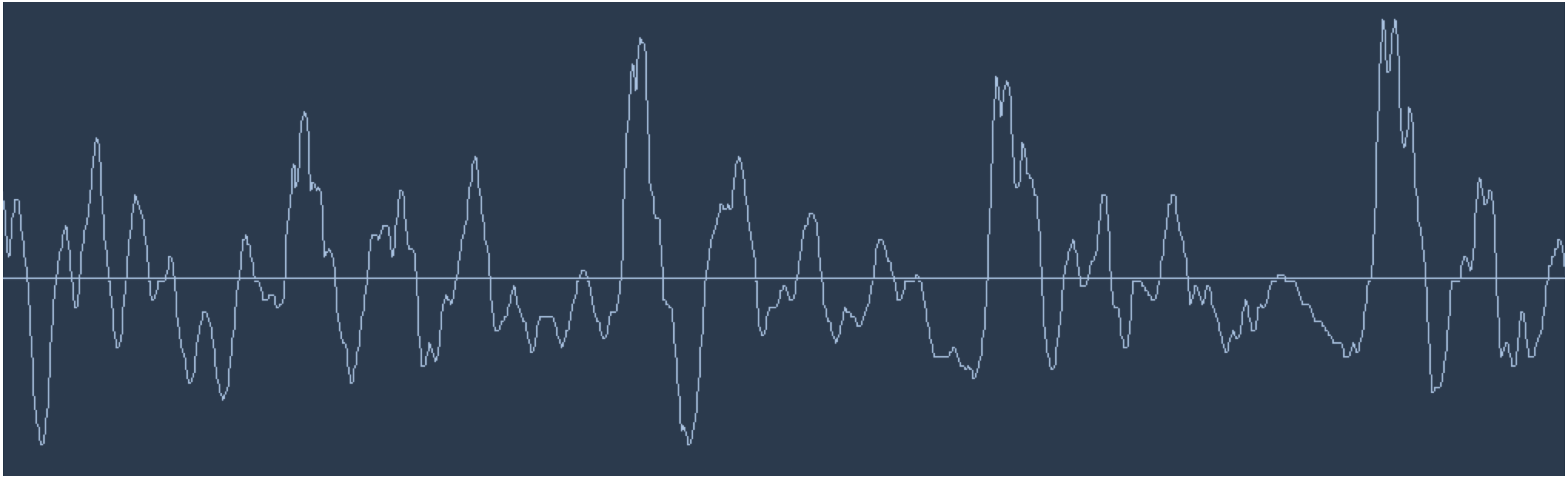
Neural Network

“Hello”

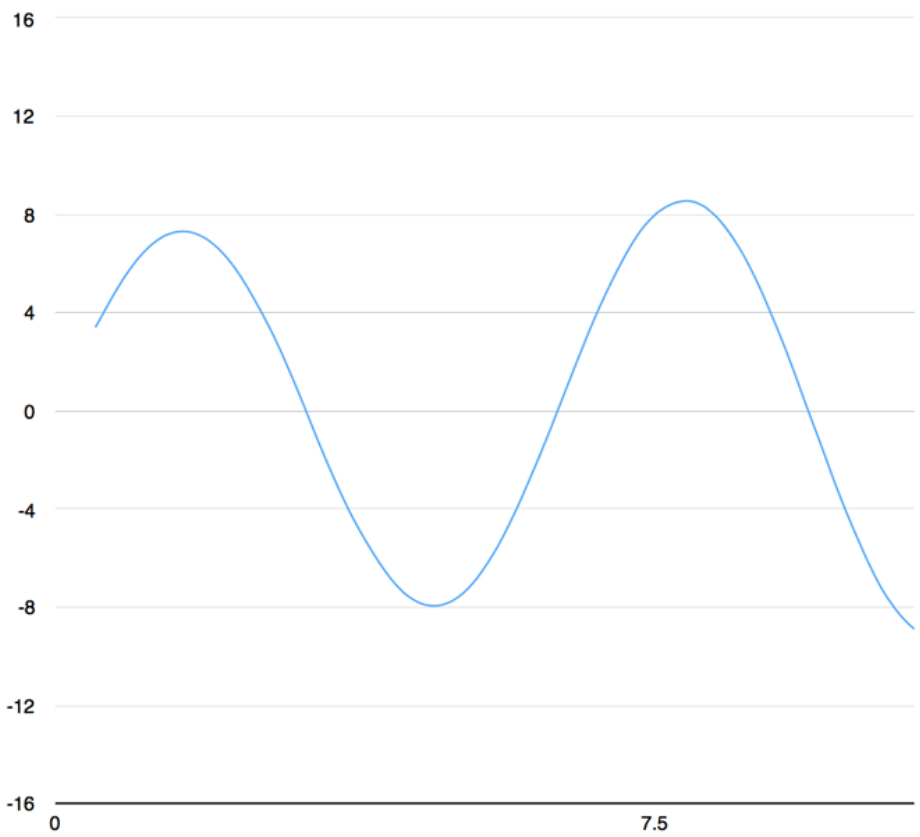
Sound wave of
me saying “Hello”

Plain text

Turning Sounds into Bits



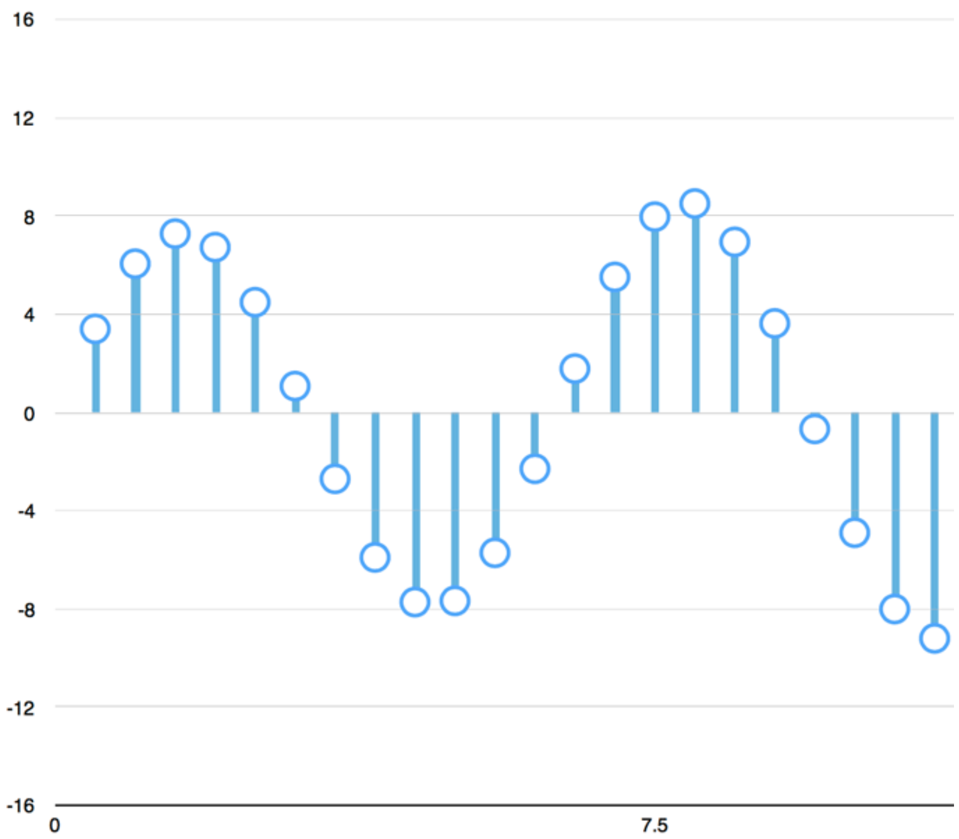
Sampling



Original Analog Signal

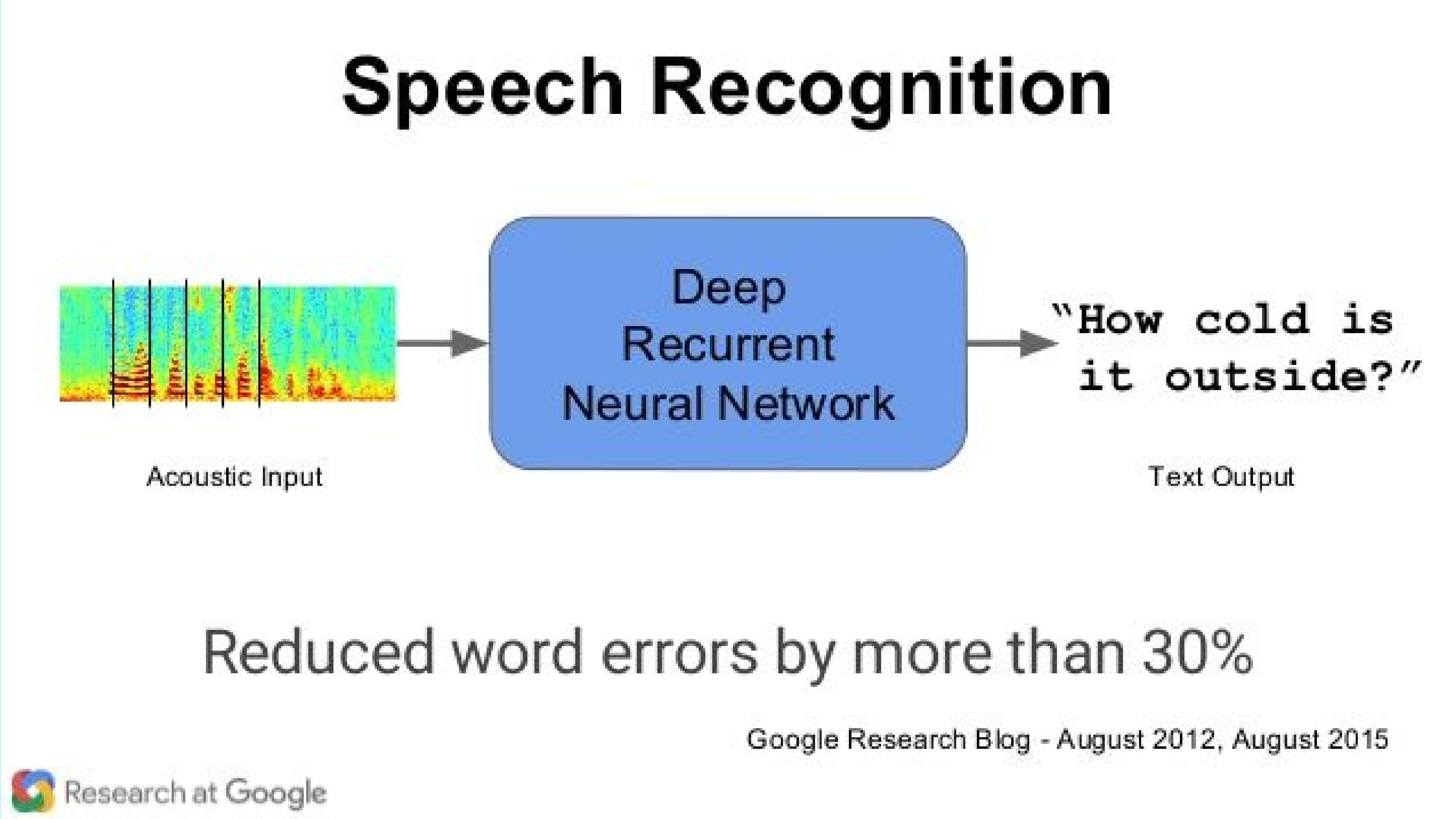
?

==



Sampled Digital Signal

RNN for Acoustic input



Speech Recognition

Acoustic Input

Deep Recurrent Neural Network

"How cold is it outside?"

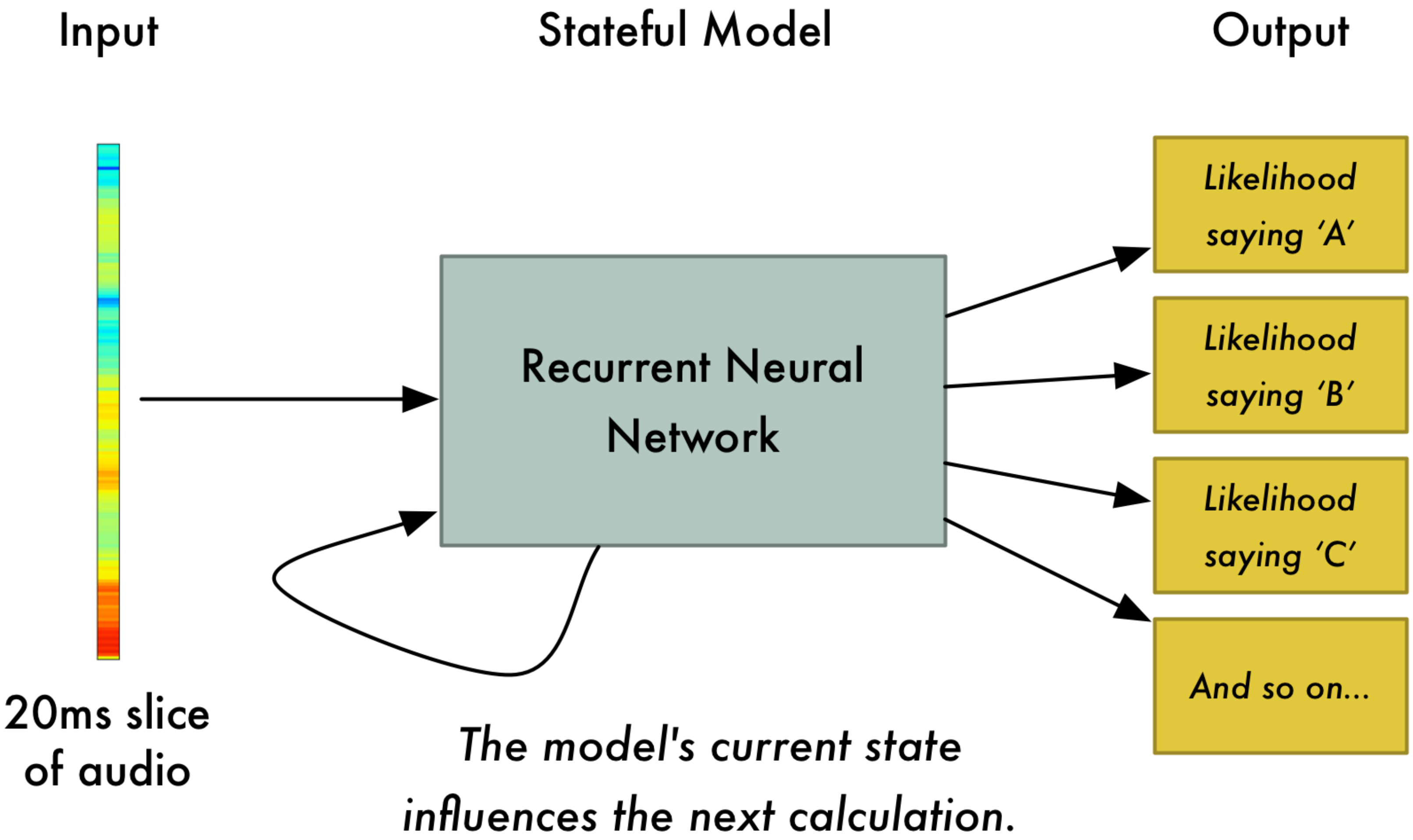
Text Output

Reduced word errors by more than 30%

Google Research Blog - August 2012, August 2015

Research at Google

Recognizing Characters from Short Sounds



Thanks!

Q&A