DNN-HMM

洪青阳

纲要

7.1 深度学习

7.2 深度神经网络（DNN）

7.3 DNN与HMM的结合

7.4 不同DNN结构

7.5 本章小结

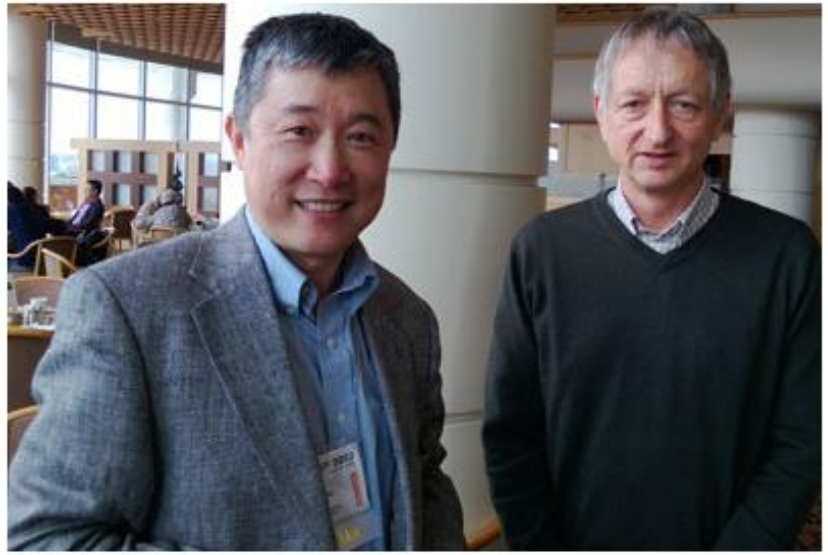
7.1 深度学习

- 2006年，Hinton提出使用受限波尔兹曼机（restricted Boltzmann machine，RBM）对神经网络的节点做初始化，即深度置信网络（deep belief network，DBN）。

- DBN预训练方法缓解了神经网络局部最优解问题，将隐含层推动到了7层，神经网络真正意义上有了“深度”，由此揭开了深度学习的热潮。

深度学习与语音识别

- 2012年，微软研究院的俞栋和邓力等人提出了上下文相关的深度神经网络与隐马尔可夫模型融合的声学模型（CD-DNN-HMM），在大词汇量的连续语音识别（LVCSR）任务上取得了显著的进步，相比于传统的GMM-HMM系统获得超过20%的相对性能提升，这项工作是深度学习在语音识别上具有重大意义的成果。

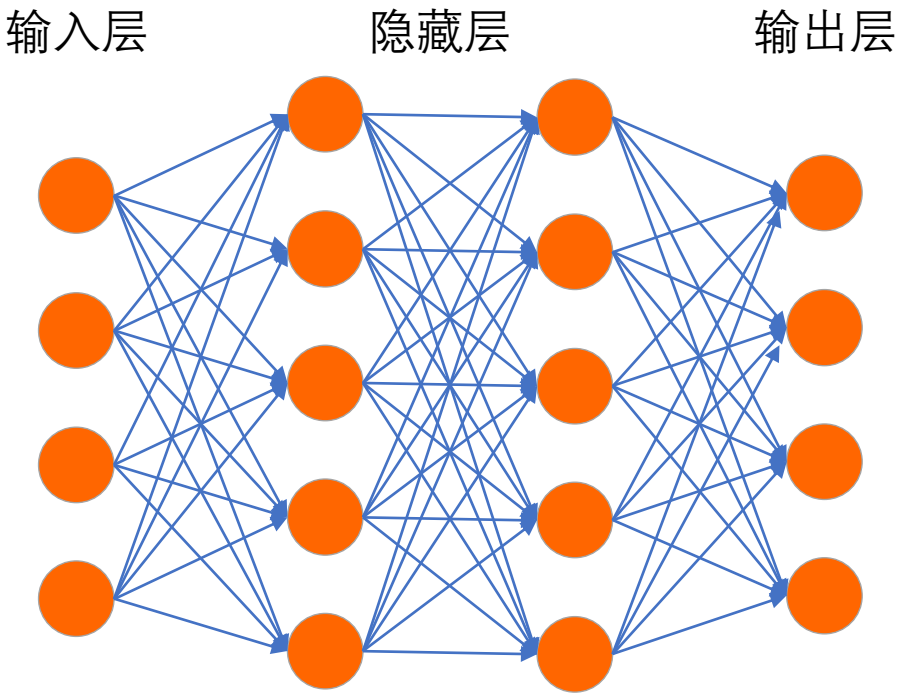


Li Deng (*left*) and Geoff Hinton.

A core development that enables Skype translation came from Redmond researcher Li Deng. He invited Geoff Hinton, a professor at the University of Toronto, to visit Redmond in 2009 to work on new neural-network learning methods, based on a couple of seminal papers from Hinton and his collaborators in 2006 that had brought new

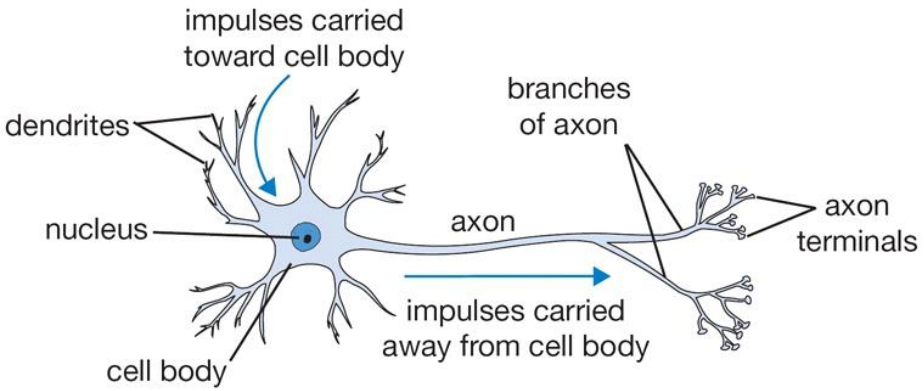
俞栋

7.2 深度神经网络(DNN)



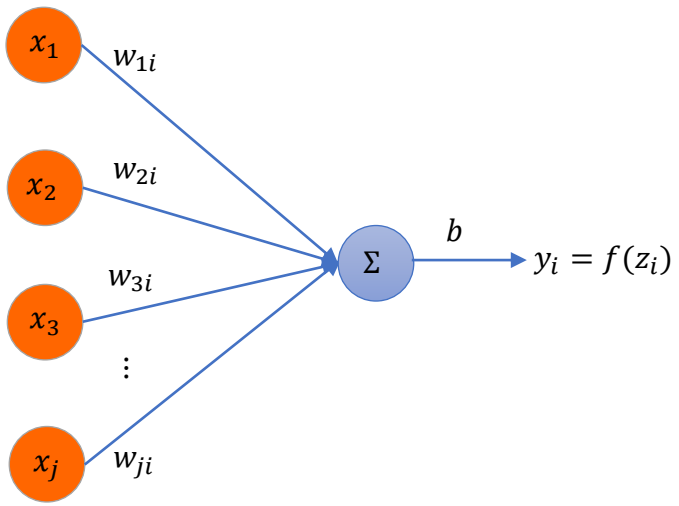
神经元

- 神经元是大脑的主要计算单元。人类大脑平均有860亿个神经元。神经元相互连接，通过树突接受其他神经元的信号，对这些信号进行计算之后，通过轴突将信号传递给下一个神经元。据估计，人类大脑平均有1014-1015个突触。
- 突触的一个关键特性是它可以缩放通过它的信号大小，这个比例因子可以被称为权重(weight)。普遍认为，大脑学习的方式是通过改变突触的权重实现的。因此，不同的权重导致对输入产生不同的响应。



From: https://cs231n.github.io/neural-networks-1

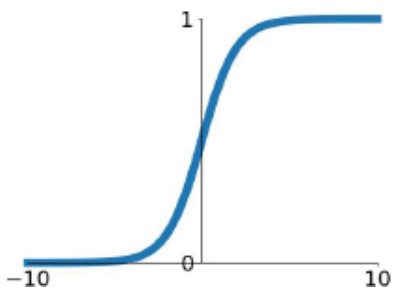
激活函数



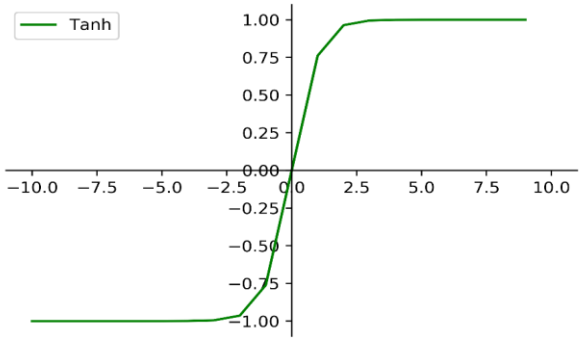
神经元的输出$z_i = \sum_j w_{ji} x_j + b$

激活函数

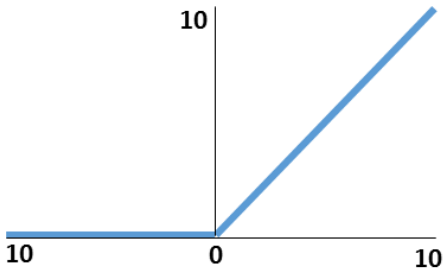
- Sigmoid: $f(z) = 1/(1 + \exp(-z))$



- $\mathrm{Tanh(x)} = \frac{\sinh(z)}{\cosh(z)} = \frac{e^z - e^{-z}}{e^z + e^{-z}}$



- ReLU: $f(z) = \max(0, z)$



激活函数

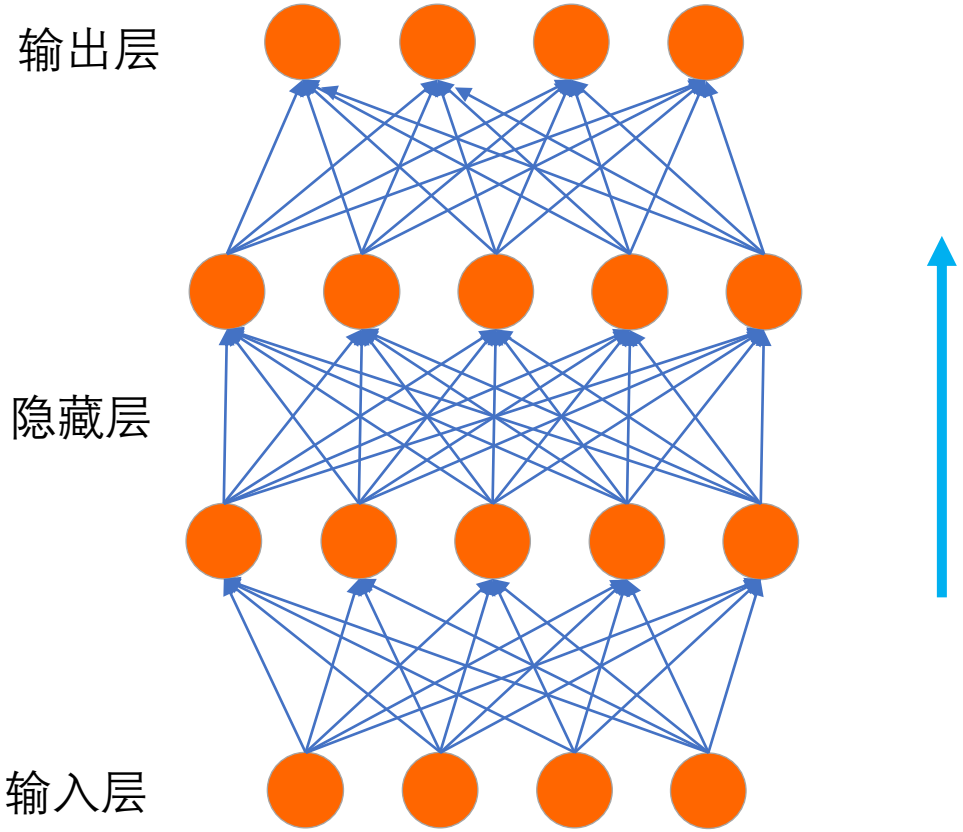
（1）对于深度神经网络，中间的隐层的输出必须有一个激活函数。否则多个隐层的作用和没有隐层相同。

（2）对于二分类问题，输出层是Sigmoid函数，其函数值恰好可以解释为属于正类的概率（概率的取值范围是0~1）。

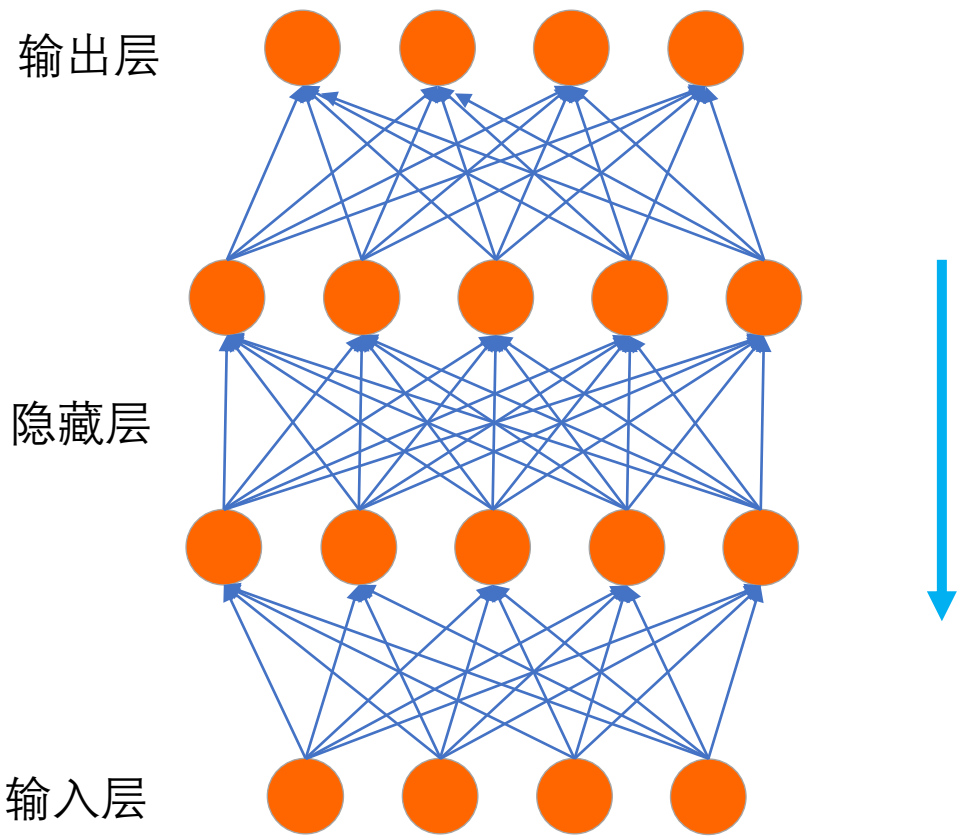
（3）对于多分类问题，输出层就要采用Softmax函数，其输出值代表输入值属于各个类的概率。Softmax函数首先对所有输出值求和，然后对每个输出值进行规整。为了避免指数运算溢出，一般每个输入值要减去输入最大值 z_{max}，如下所示。

$$f(z_i) = e^{(z_i - z_{max})} / \sum_{k=1}^{K} e^{(z_k - z_{max})}$$

前向传播(FeedForward)



反向传播(BackForward)



训练准则

- 交叉熵(Cross Entropy, CE)损失函数

$$C = -\sum_{i} \hat{y}_i ln y_i$$

$$y_i = f(z_i) = e^{z_i} / \sum_{k=1}^{K} e^{z_k}$$

其中K是DNN输出层的节点数，i是输出层节点索引，$\hat{y}_i$是真实的标签，z_i是DNN的输出层第i个节点原始输出，y_i是其激活输出。

- 针对语音识别

$$C = -\sum_{t}^{T} \sum_{i} \hat{y}_i(t) ln y_i(t)$$

其中T是特征序列的帧数。

训练准则

- 训练模型的目标是使得训练数据能够在模型上获得尽可能一致的输入和输出，换句话说就是找到一个合适的模型系数$\{W, b\}$使得期望损失C最小化。

$$\underset{W,b}{argmin}\, C$$

- 随机梯度下降(SGD)算法

$$W' \leftarrow W - \varepsilon \Delta \boldsymbol{W}$$
$$b' \leftarrow b - \varepsilon \Delta \boldsymbol{b}$$

其中ε是学习率。

推导过程

$$\frac{\partial C}{\partial z_i} = \sum_j \frac{\partial C_j}{\partial y_j}\frac{\partial y_j}{\partial z_i}$$

$$\frac{\partial C_j}{\partial y_j} = \frac{\partial(-\hat{y}_j ln y_j)}{\partial y_j} = -\hat{y}_j \frac{1}{y_j}$$

如果$i = j$,

$$\frac{\partial y_i}{\partial z_i} = \frac{\partial \frac{e^{z_i}}{\sum_k e^{z_k}}}{\partial z_i} = \frac{\sum_k e^{z_k} e^{z_i} - (e^{z_i})^2}{\left(\sum_k e^{z_k}\right)^2} = \left(\frac{e^{z_i}}{\sum_k e^{z_k}}\right)\left(1 - \frac{e^{z_i}}{\sum_k e^{z_k}}\right) = y_i(1 - y_i)$$

如果$i \neq j$,

$$\frac{\partial y_j}{\partial z_i} = \frac{\partial \frac{e^{z_j}}{\sum_k e^{z_k}}}{\partial z_i} = -e^{z_j}\left(\frac{1}{\sum_k e^{z_k}}\right)^2 e^{z_i} = -y_i\, y_j$$

$$\frac{\partial C}{\partial z_i} = \sum_{j\neq i} \frac{\partial C_j}{\partial y_j}\frac{\partial y_j}{\partial z_i} + \sum_{j=i} \frac{\partial C_j}{\partial y_j}\frac{\partial y_j}{\partial z_i} = \sum_{j\neq i} -\hat{y}_j \frac{1}{y_j}(-y_i\, y_j) + -\hat{y}_i \frac{1}{y_i}(y_i(1 - y_i)) = y_i \sum_j \hat{y}_j - \hat{y}_i$$

针对分类问题，假定标签$\hat{y}_j$只有一个类别是1，其它为0，则

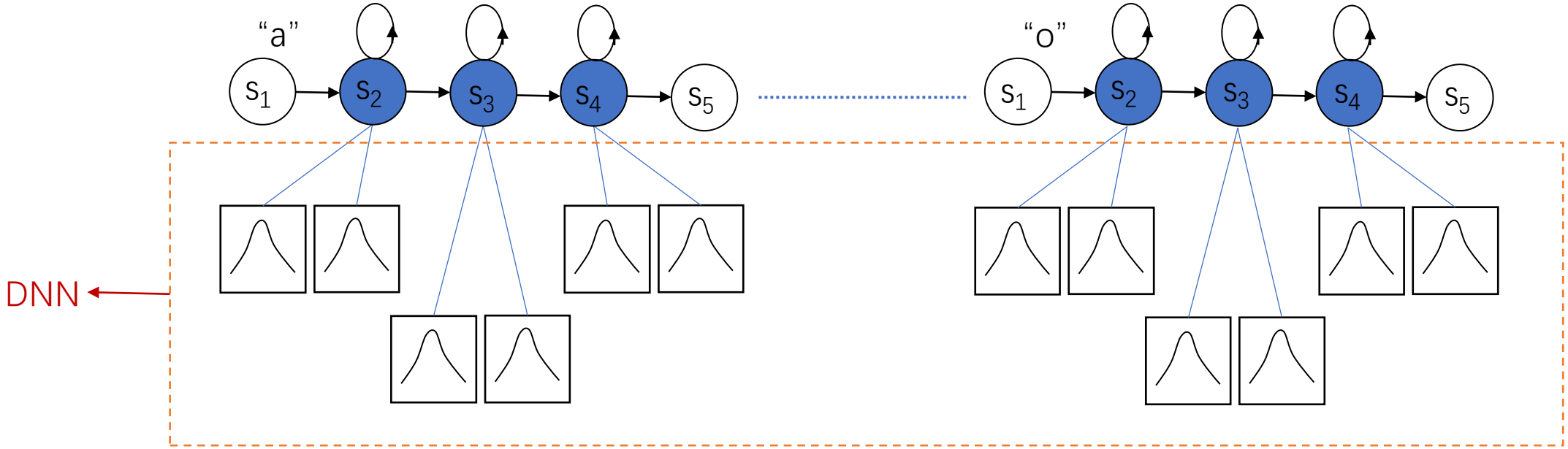
$$\frac{\partial C}{\partial z_i} = y_i - \hat{y}_i$$

$$\Delta \boldsymbol{W} = \nabla_W C = \frac{\partial C}{\partial W} = \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial w_i} = (y_i - \hat{y}_i)x^T$$

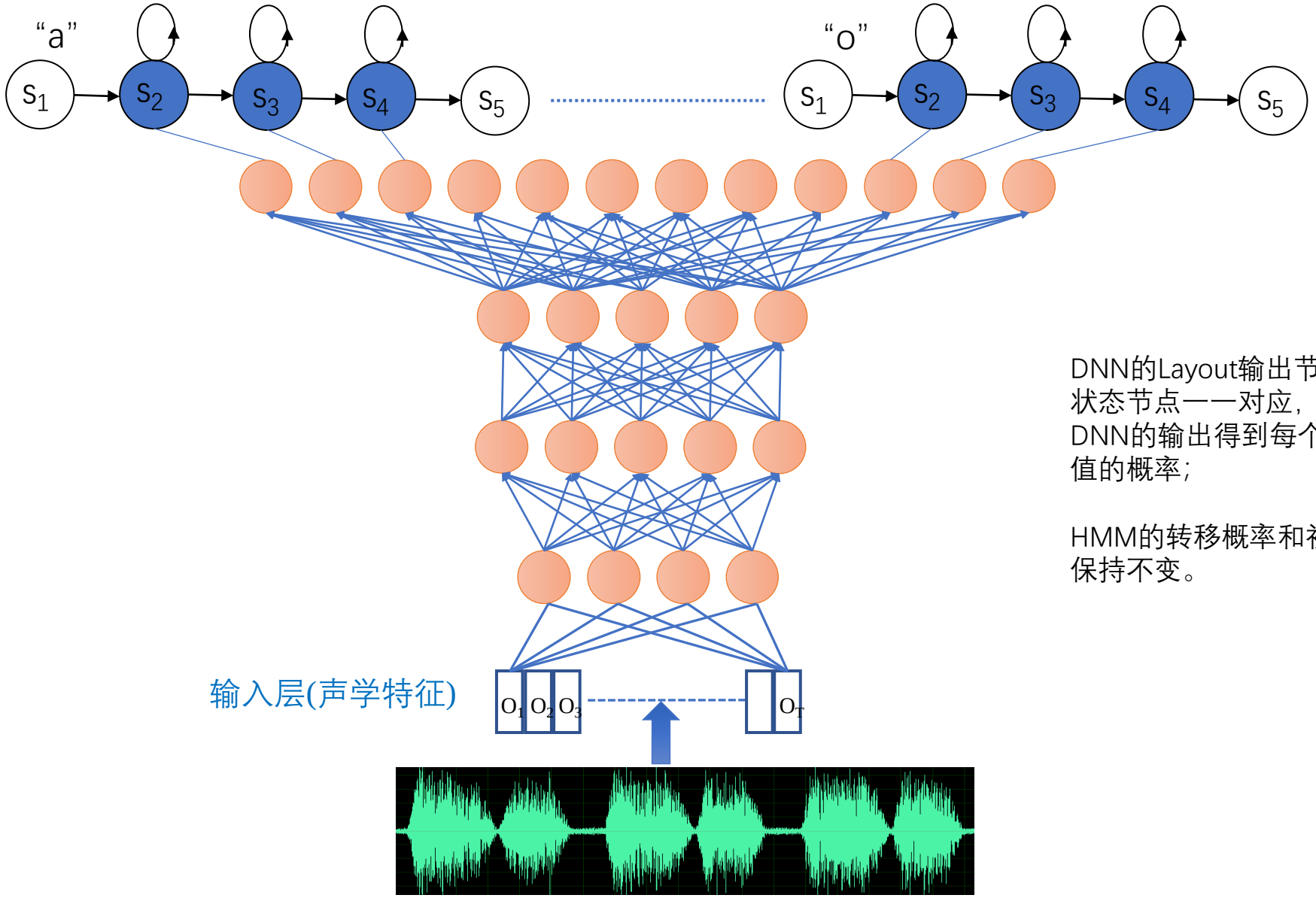
$$\Delta \boldsymbol{b} = \nabla_b C = \frac{\partial C}{\partial b} = \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial b} = (y_i - \hat{y}_i)$$

7.3 DNN与HMM的结合

- GMM 本质上仍属于浅层结构，表征能力不够强，且要求特征元素之间相互独立。而DNN拥有更强的表征能力，能够对复杂的语音变化情况进行建模。
- GMM被DNN替代;



DNN-HMM（所有音素）



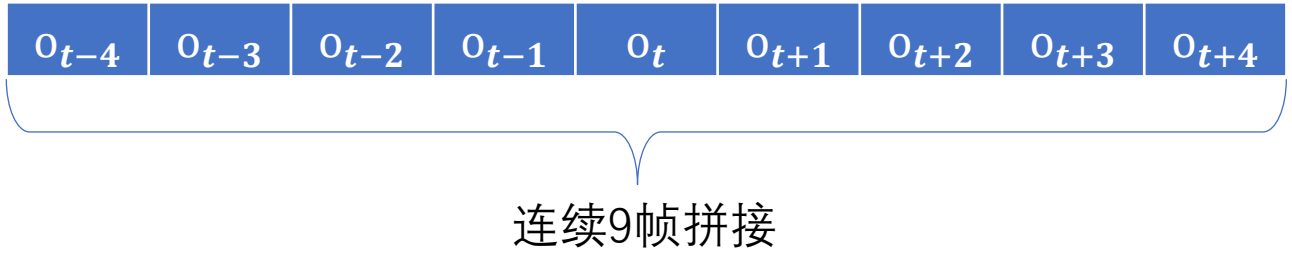
DNN的Layout输出节点与HMM的状态节点一一对应，因此可通过DNN的输出得到每个状态针对观察值的概率；

HMM的转移概率和初始状态概率保持不变。

DNN-HMM优势

- 相对于 GMM，DNN 对 HMM 中后验概率的估计不需要很苛刻的数据分布假设，特别是同一帧特征元素之间不需要相互独立，因此声学特征不一定要采用MFCC或PLP，而可以采用更原始的特征如FBank。

- DNN能够对包含多帧数据的特征进行处理，表征更丰富的语音上下文变化特征，因此输入层往往采用多帧拼接再送入网络，如果每帧有39维，前后9帧则有$39 \times 9 = 351$维拼接后的特征。



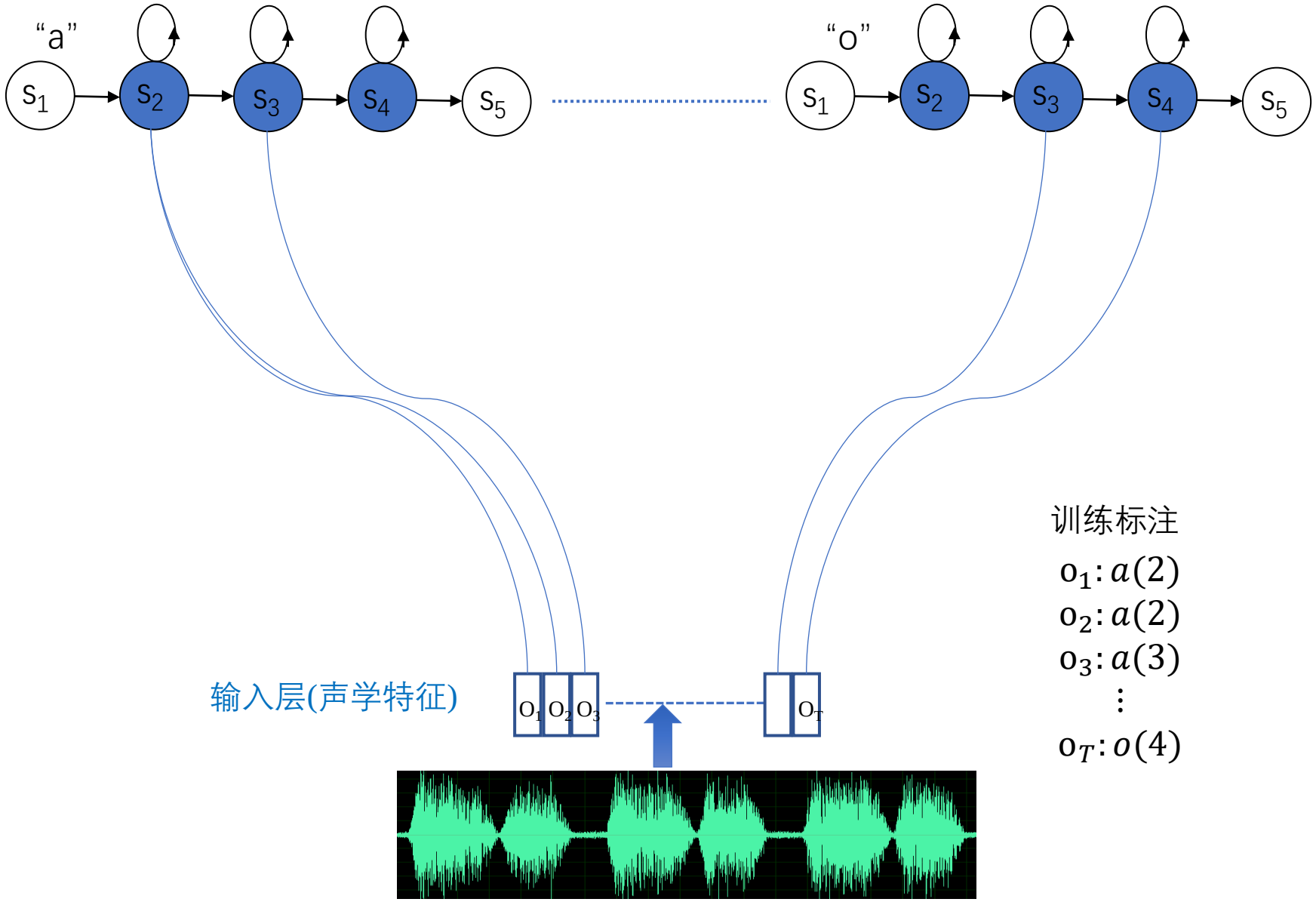
DNN-HMM的训练

Step1：利用GMM-HMM系统，生成训练数据帧与状态的对齐；

Step2：构建DNN-HMM，将GMM-HMM的HMM提取出来作为DNN-HMM中的HMM，GMM则用整个DNN替换，音素的后验概率则可以通过DNN的输出向量得到；

Step3：优化DNN-HMM，通过反向传播算法从音素级别的帧对齐序列中得到DNN的参数。

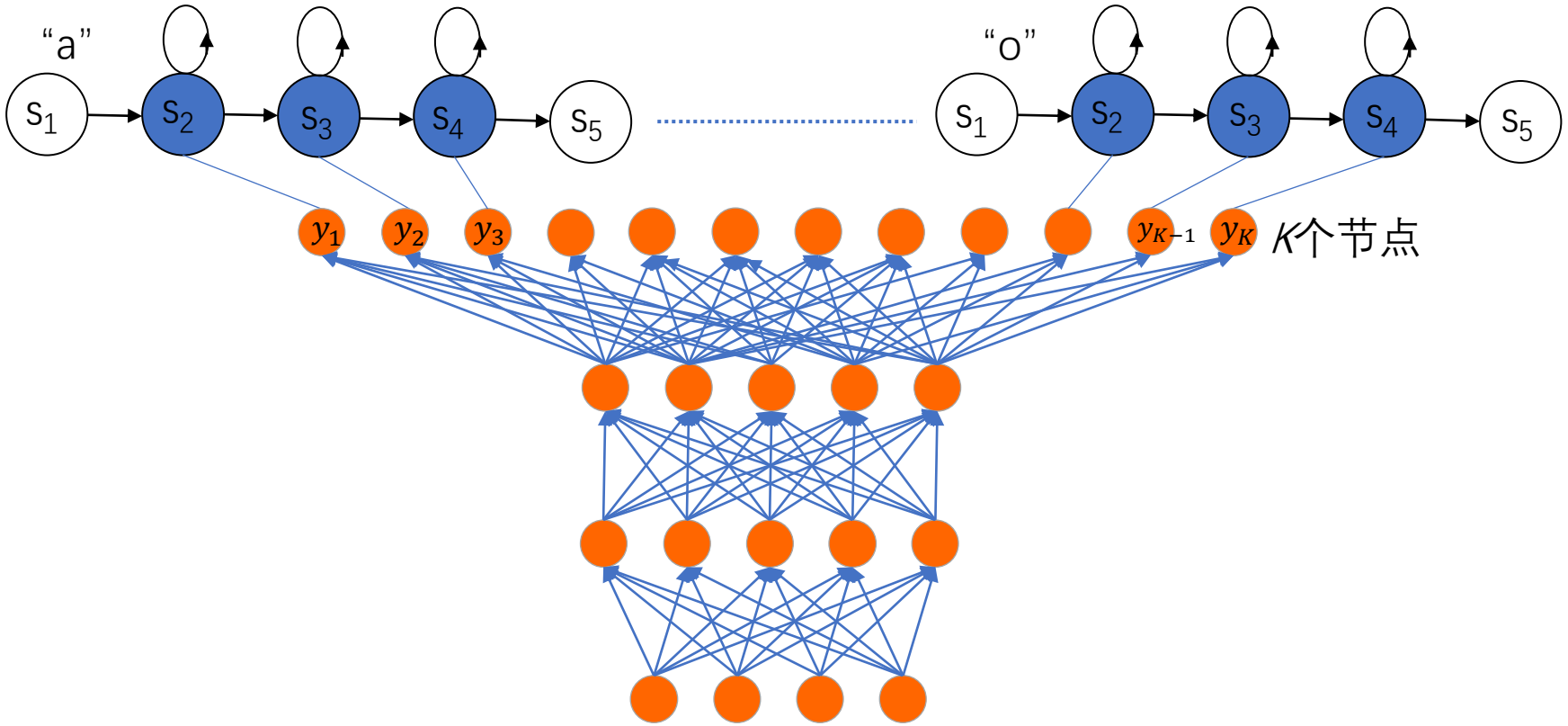
Step1: 帧与状态的对齐



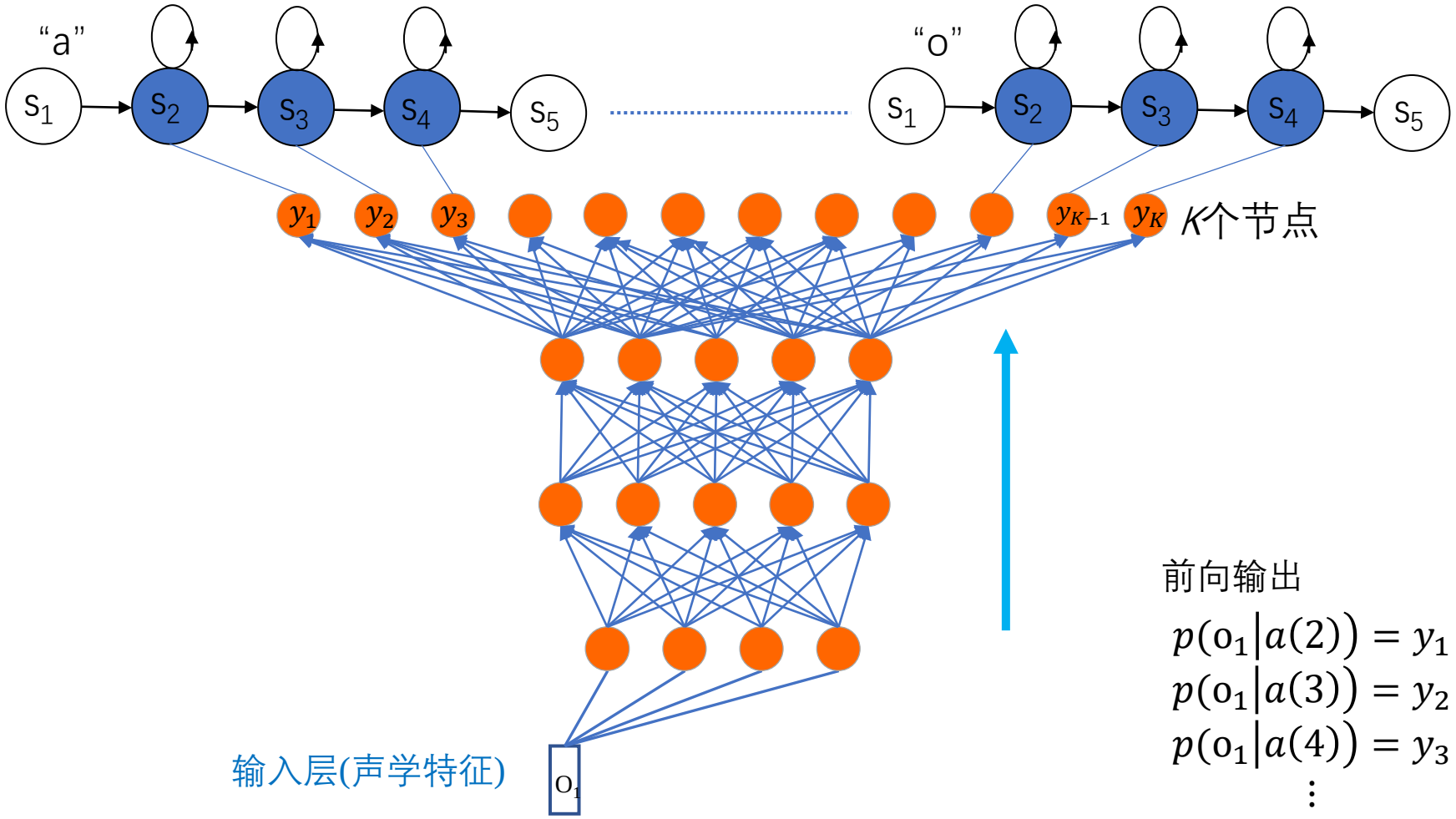
训练标注

$o_1: a(2)$
$o_2: a(2)$
$o_3: a(3)$
$\vdots$
$o_T: o(4)$

Step2: 构建DNN-HMM



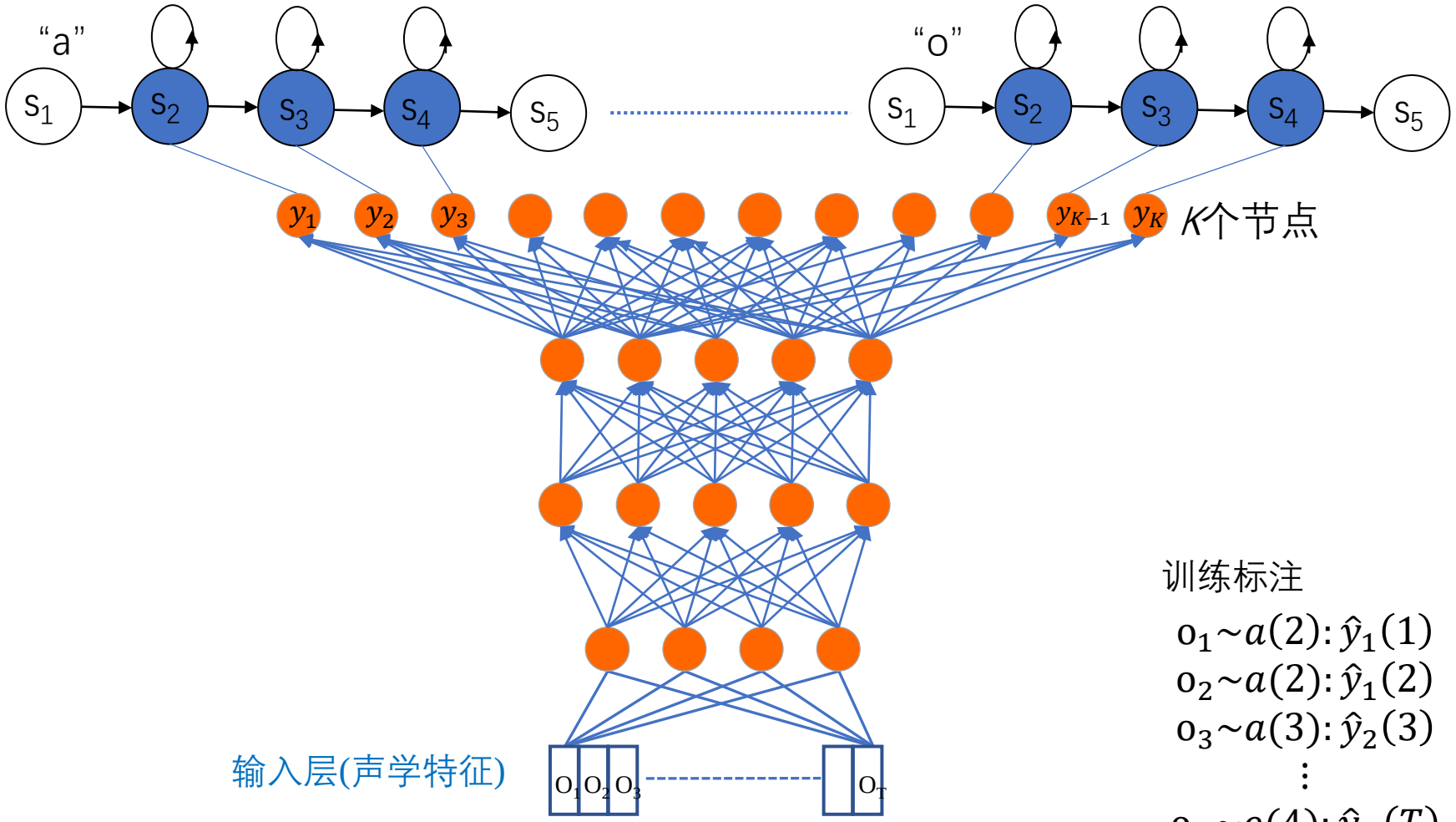
Step2: 构建DNN-HMM



前向输出

$$p(o_1|a(2)) = y_1$$
$$p(o_1|a(3)) = y_2$$
$$p(o_1|a(4)) = y_3$$
$$\vdots$$
$$p(o_1|o(4)) = y_K$$

Step3: 反向传播优化DNN-HMM



训练标注

$o_1 \sim a(2): \hat{y}_1(1)$

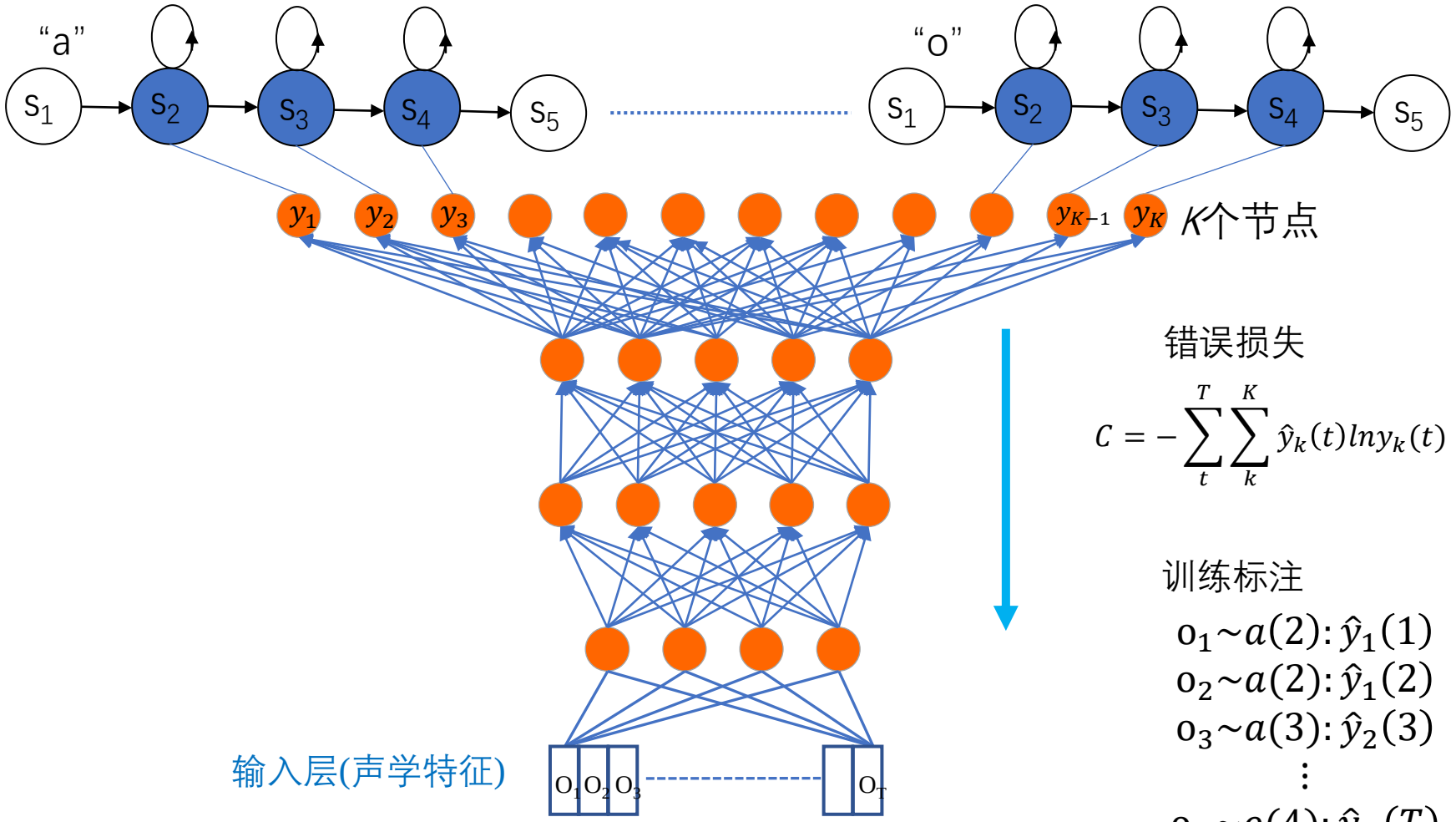
$o_2 \sim a(2): \hat{y}_1(2)$

$o_3 \sim a(3): \hat{y}_2(3)$

$\vdots$

$o_T \sim o(4): \hat{y}_K(T)$

Step3: 反向传播优化DNN-HMM



错误损失

$$C = -\sum_{t}^{T}\sum_{k}^{K} \hat{y}_k(t) ln y_k(t)$$

训练标注

$o_1 \sim a(2): \hat{y}_1(1)$

$o_2 \sim a(2): \hat{y}_1(2)$

$o_3 \sim a(3): \hat{y}_2(3)$

$\vdots$

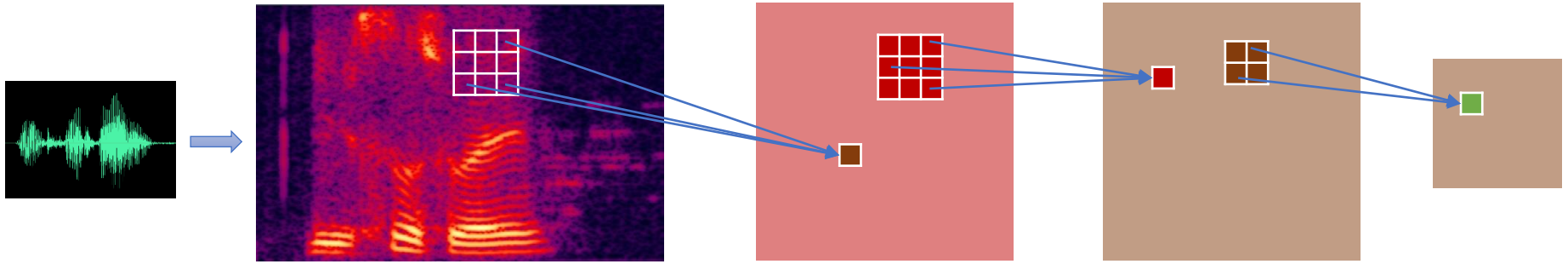
$o_T \sim o(4): \hat{y}_K(T)$

7.4 不同DNN结构

- CNN
- LSTM
- GRU
- FSMN
- TDNN
- TDNN-F

CNN

- 语音信号可以用语谱图来表示，因此可借鉴图像处理的方式，采用卷积神经网络，在时间轴和频率轴上提取局部特征，如下图所示。

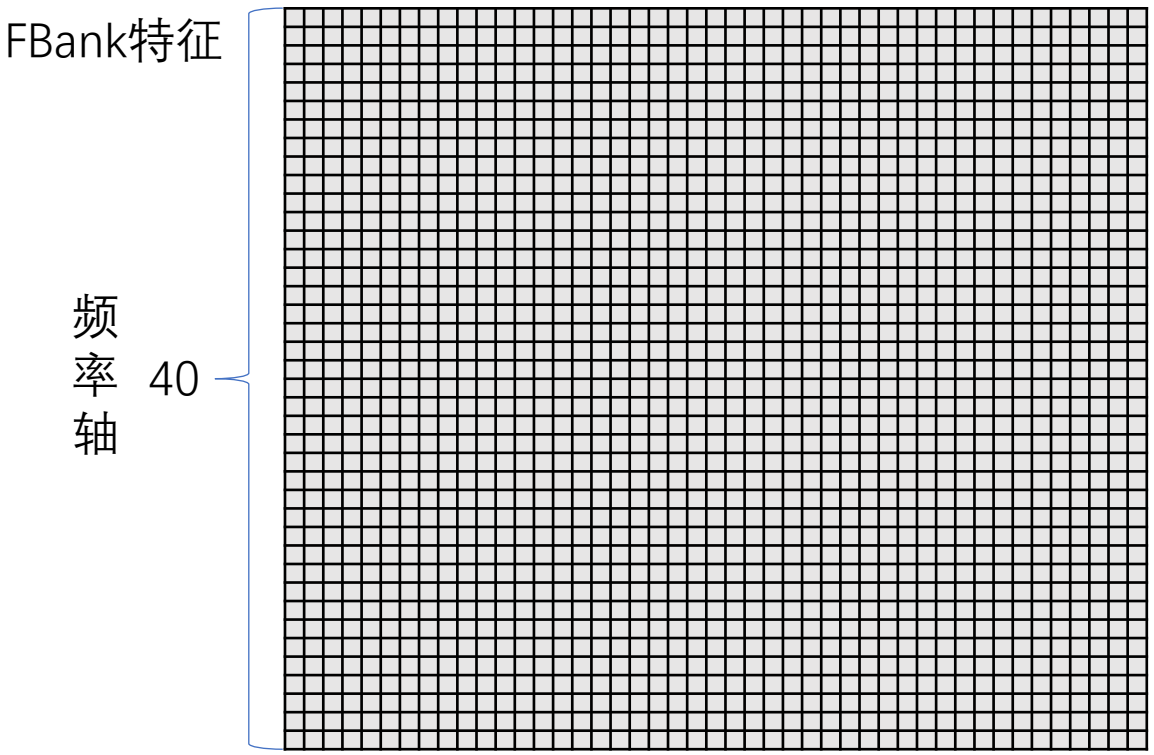


Deep CNN提取局部特征过程

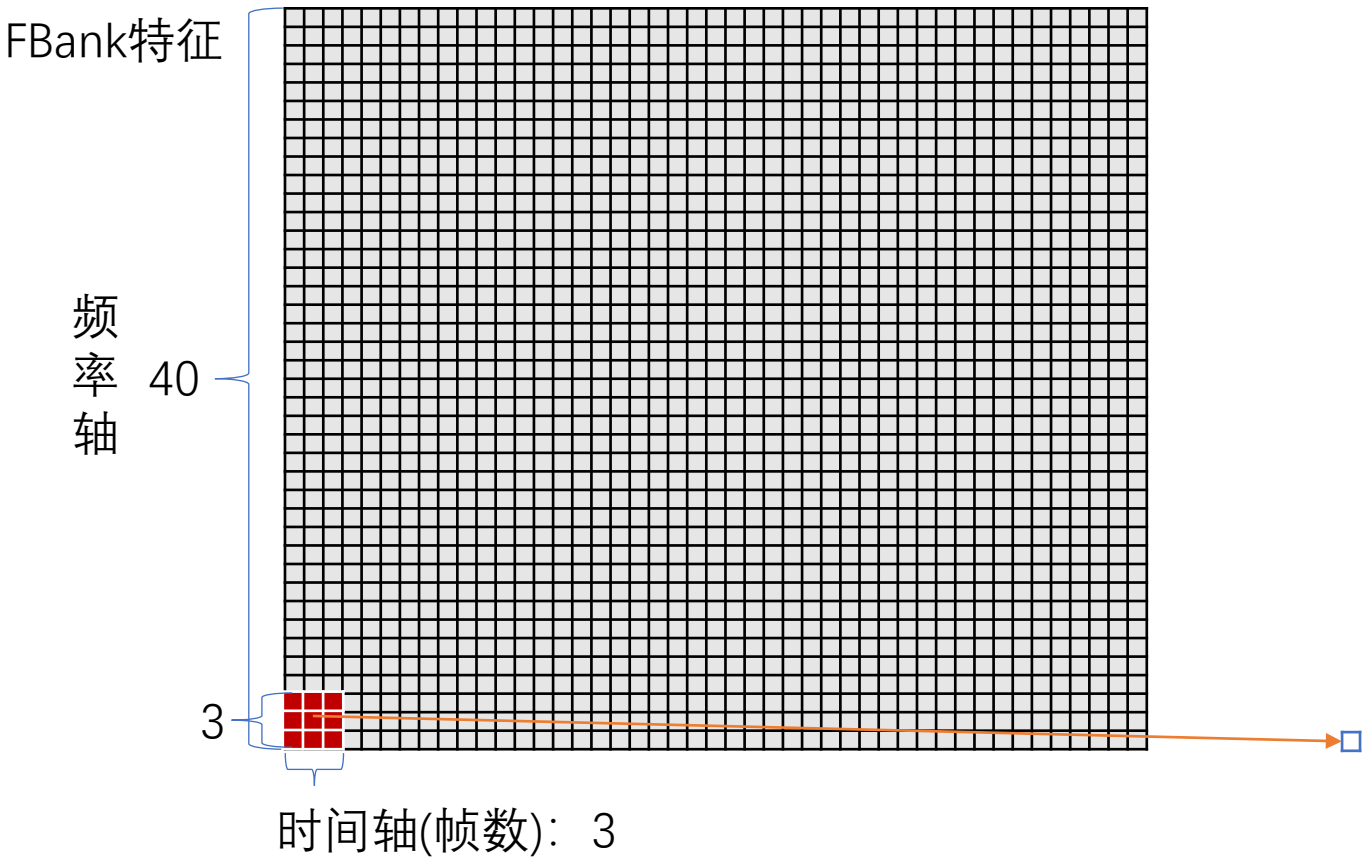
CNN的输入

- 特征维度：40 (FBank)
- 连续3帧
- 则CNN的输入维度(input_dim)为：40*3

输入特征

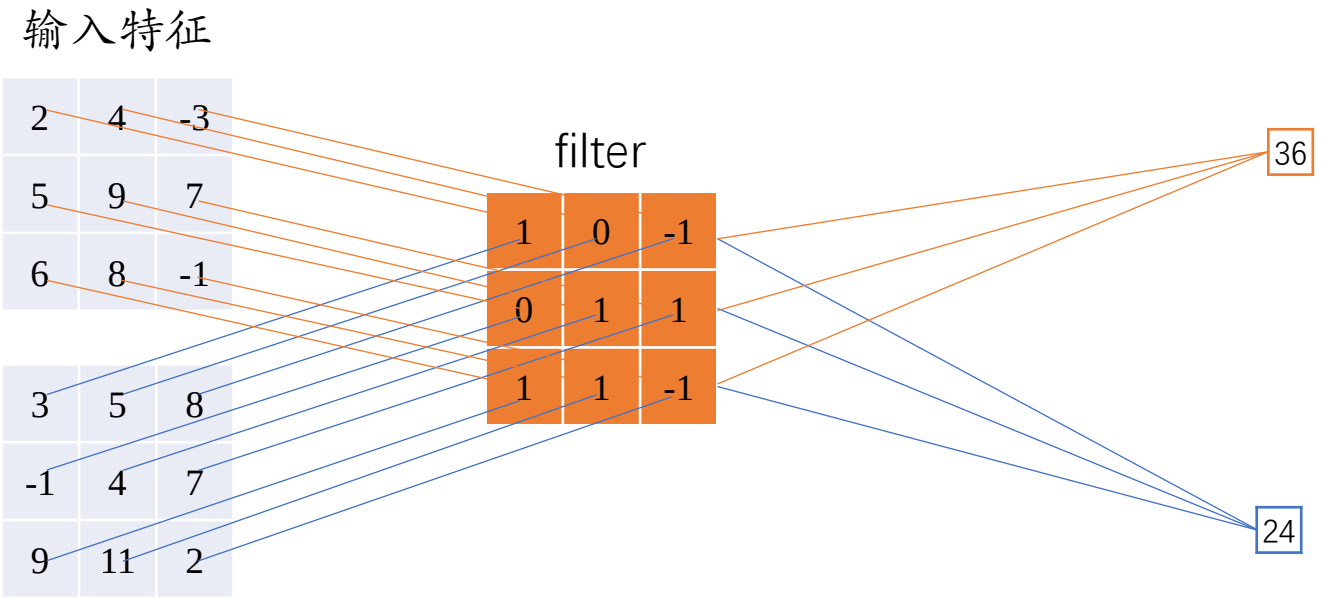


1个filter输出



- 单个filter: 3*3

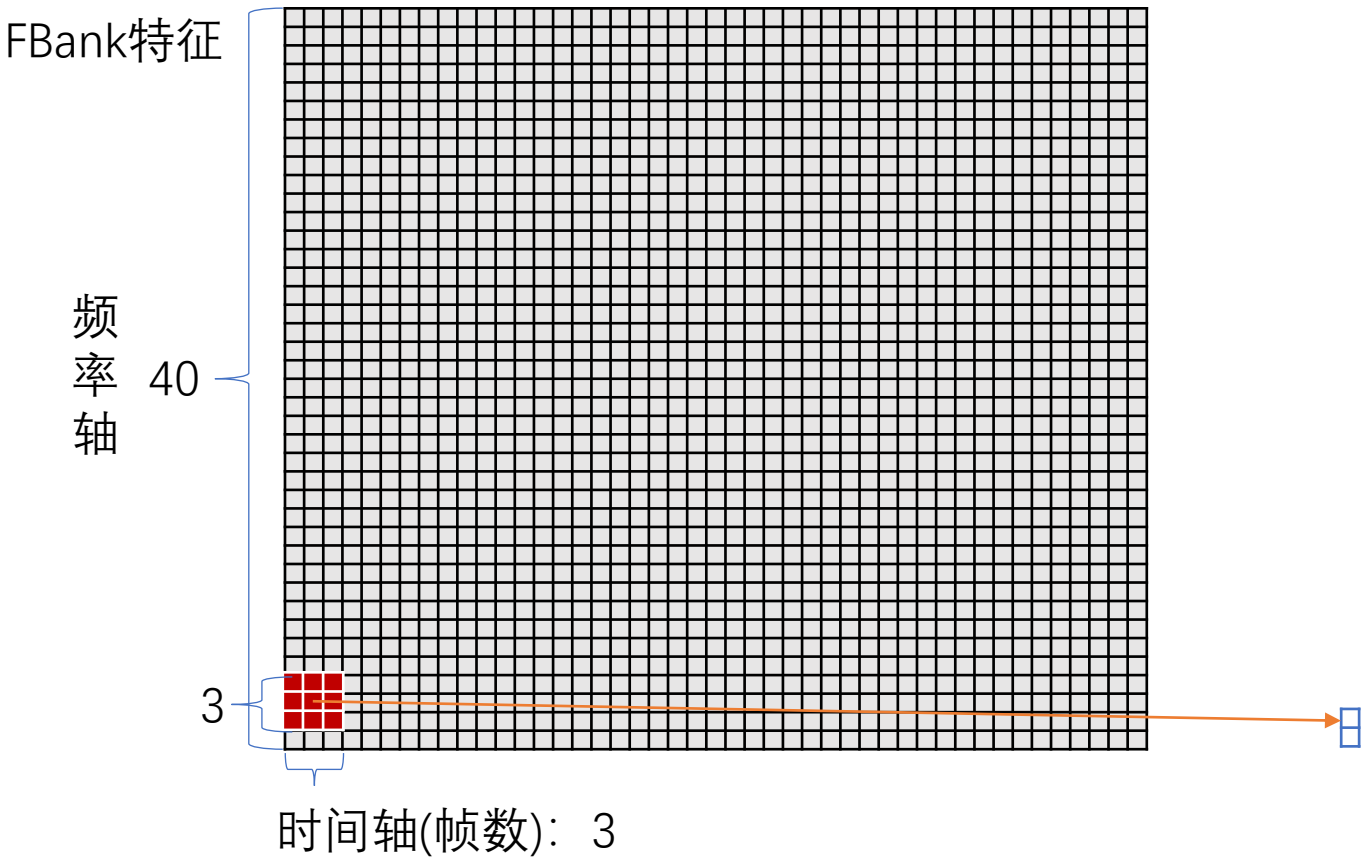
1个filter输出



$$2\times1+4\times0+(-3)\times(-1)+5\times0+9\times1+7\times1+6\times1+8\times1+(-1)\times(-1)=36$$

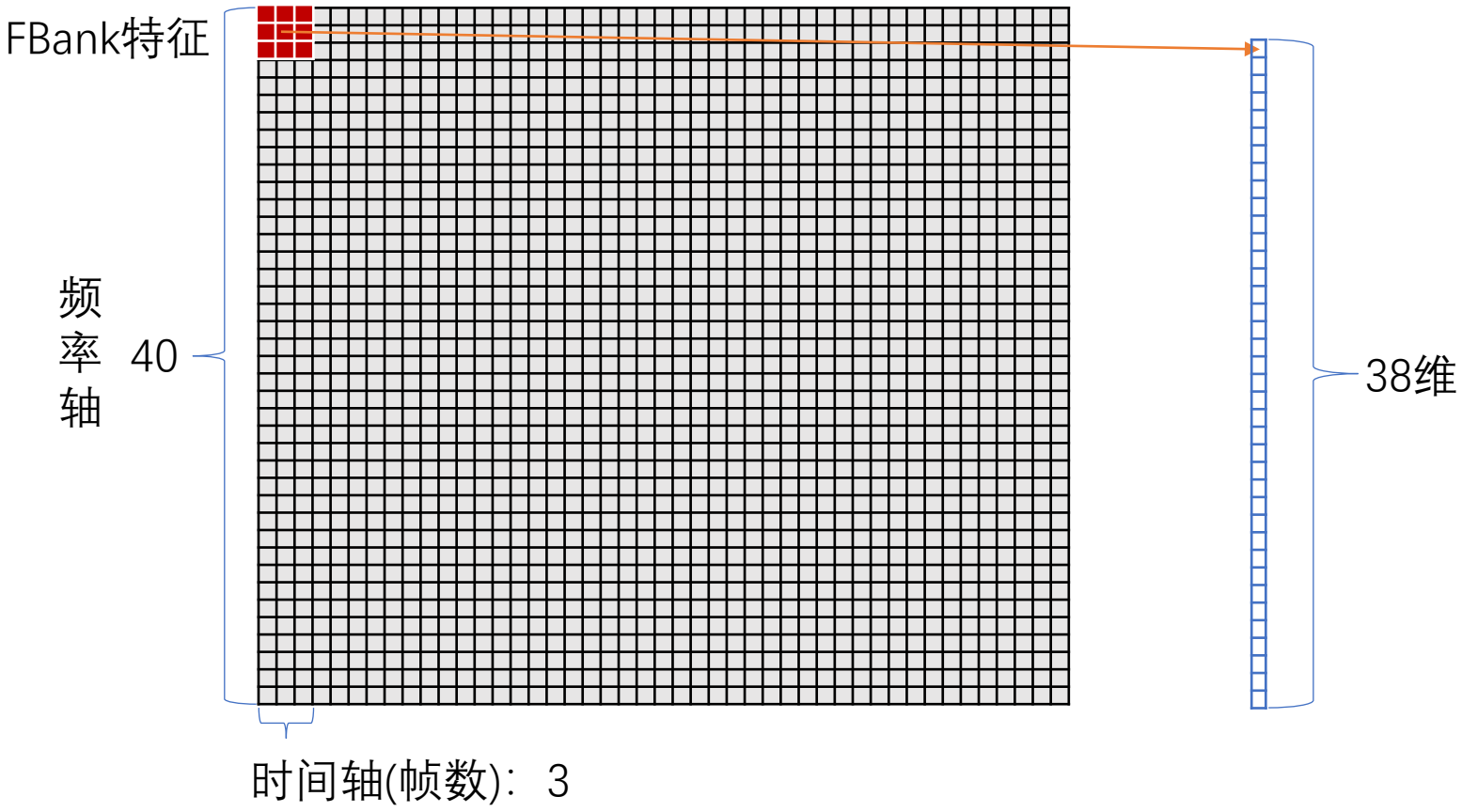
$$3\times1+5\times0+8\times(-1)+(-1)\times0+4\times1+7\times1+9\times1+11\times1+2\times(-1)=24$$

1个filter输出



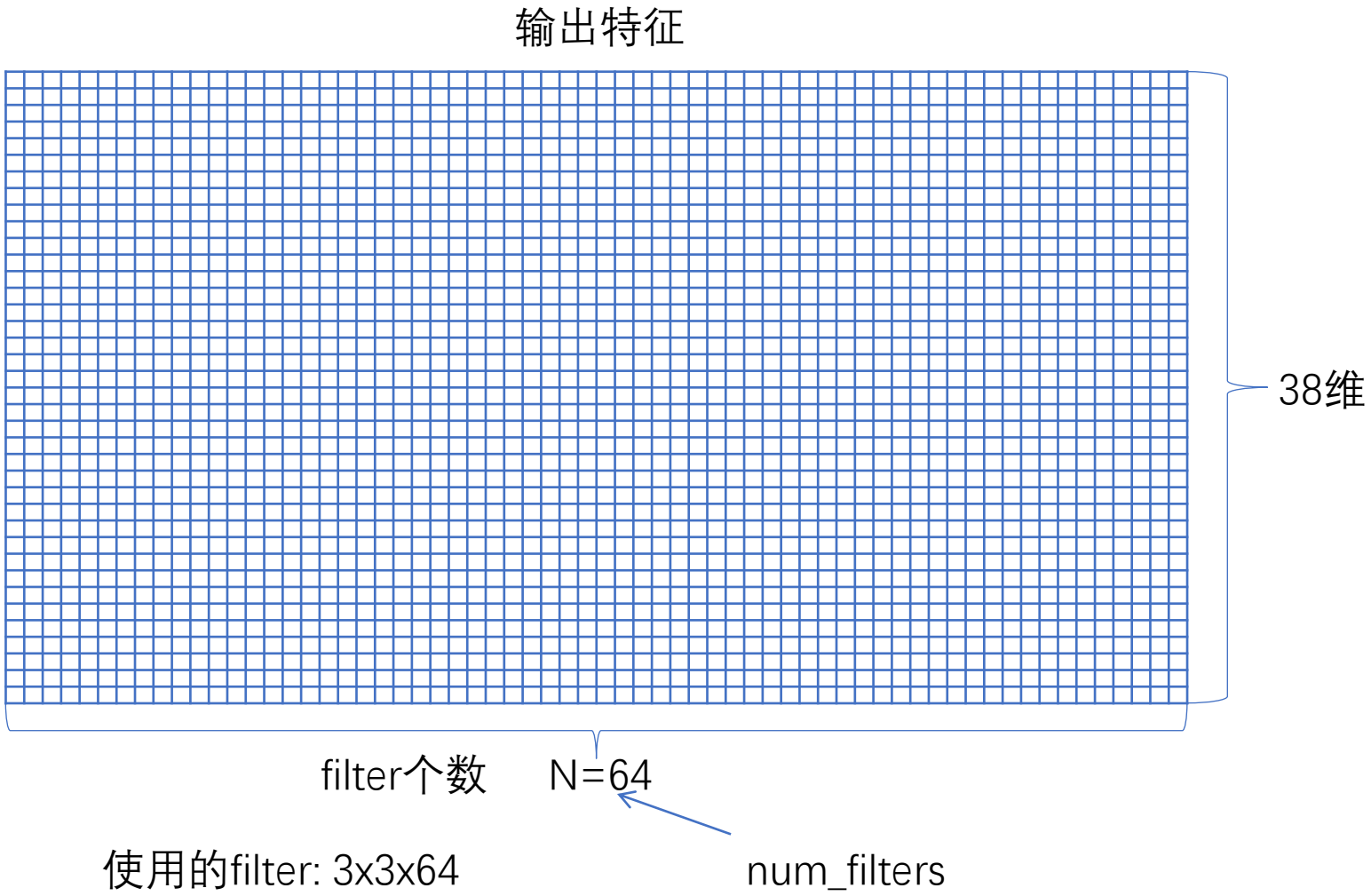
- 单个filter: 3*3
- 频率轴步进为1，特征维度40，则filter输出节点：(40-3)/1+1=38

1个filter输出



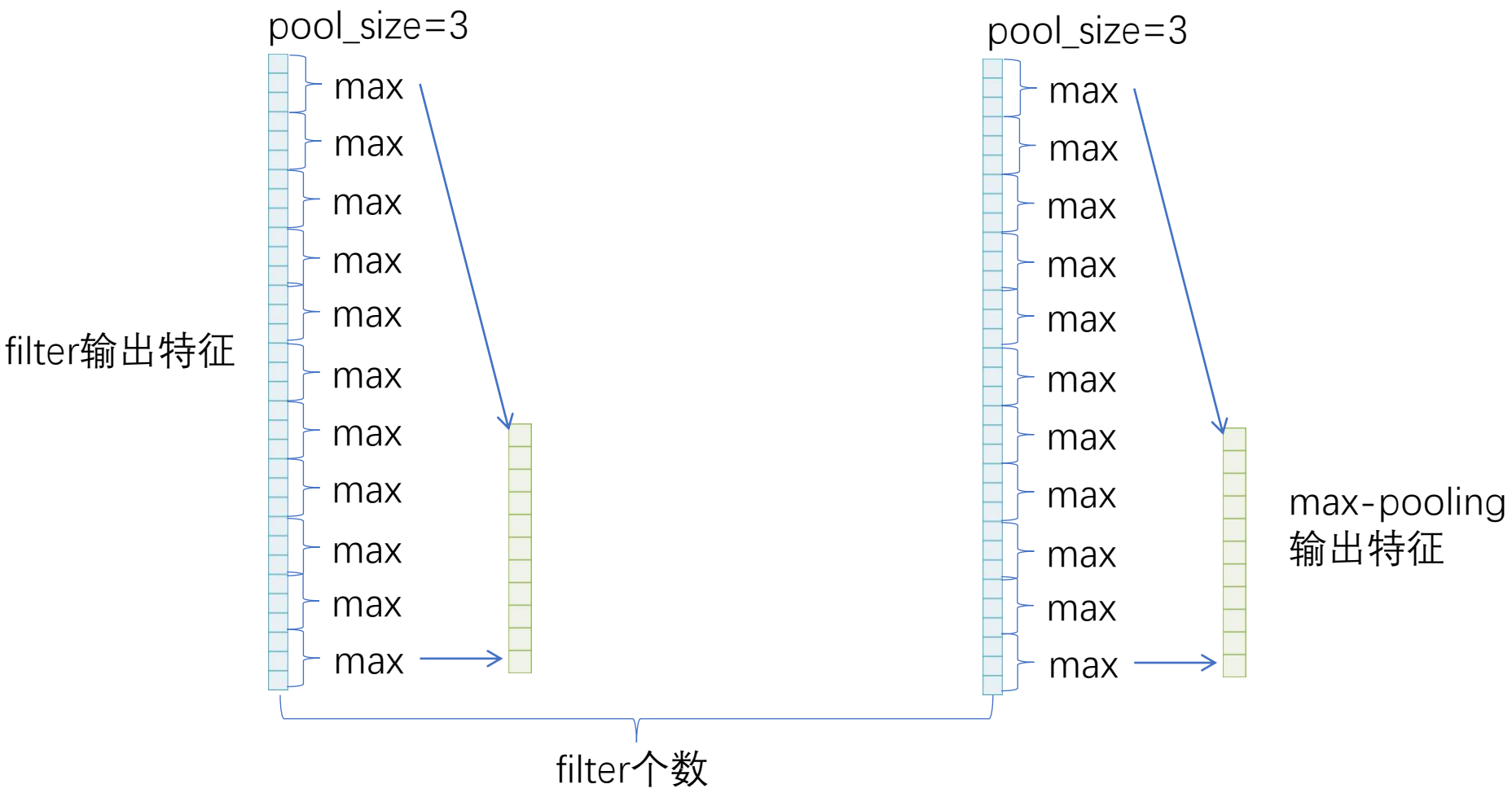
- 单个filter: 3*3
- 频率轴步进为1，特征维度40，则filter输出节点：(40-3)/1+1=38

N个filter输出



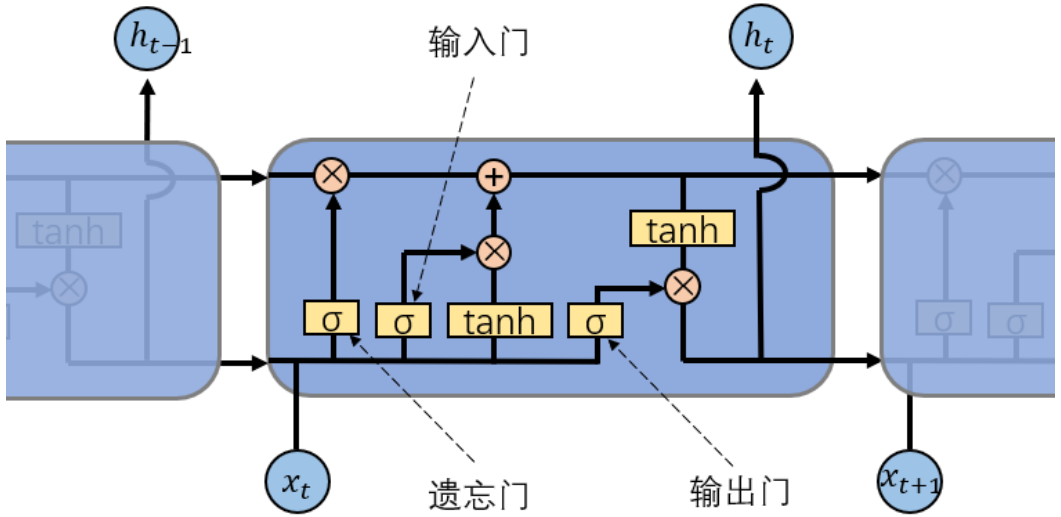
使用的filter: 3x3x64

max-pooling输出



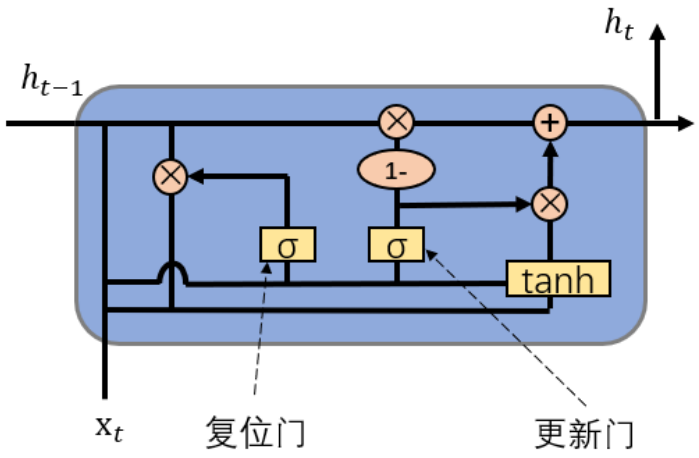
LSTM

- 1997年，一种被称为长短时记忆单元（LSTM）的结构被引入到RNN中。LSTM成功地克服了传统RNN所存在的梯度消失问题，并且能够很好地完成诸如嘈杂输入序列的模式识别等RNN无法应对的任务。



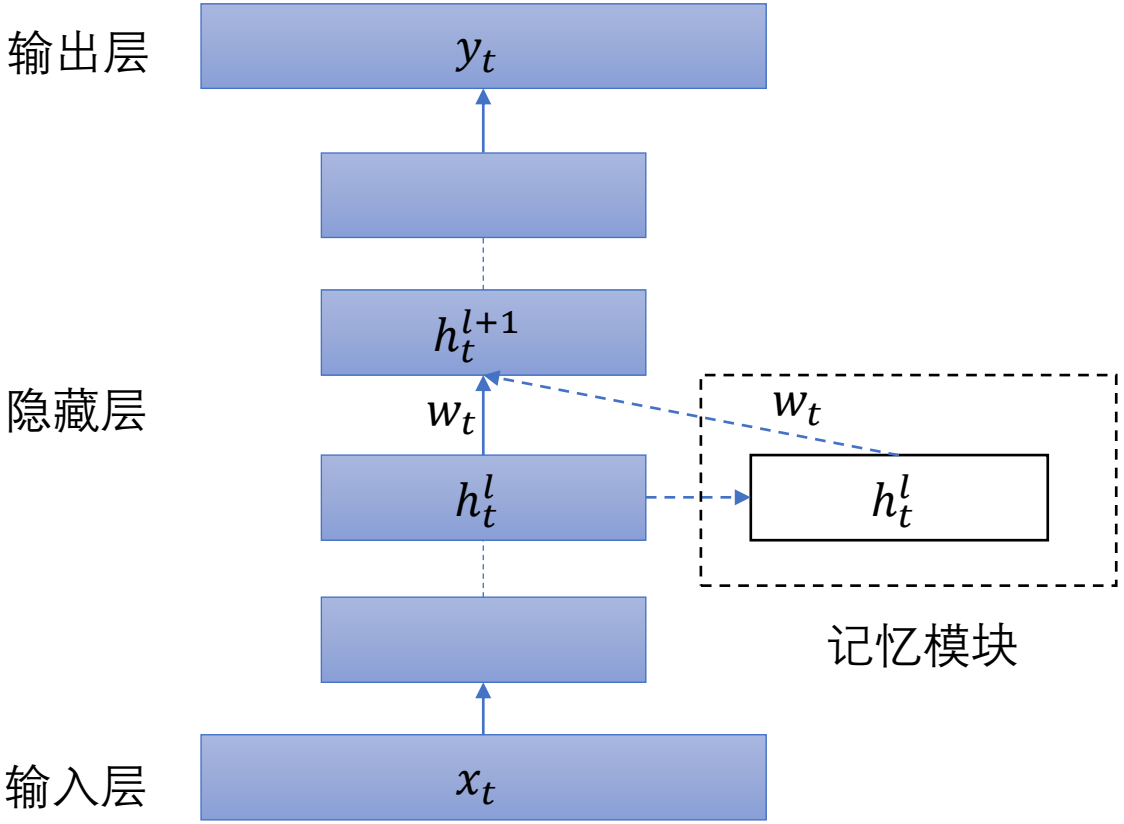
GRU

- GRU的循环单元内只包含2个门控结构，即复位门和更新门。
- 作为LSTM的改进版，GRU的更新门是由LSTM的遗忘门和输入门合并而成的，一个更新门就实现了LSTM两个门的功能，同时还合并了细胞状态和隐藏状态，再加上其他的一些改进设计，这些都使得GRU的模型结构比LSTM更为简单轻便。



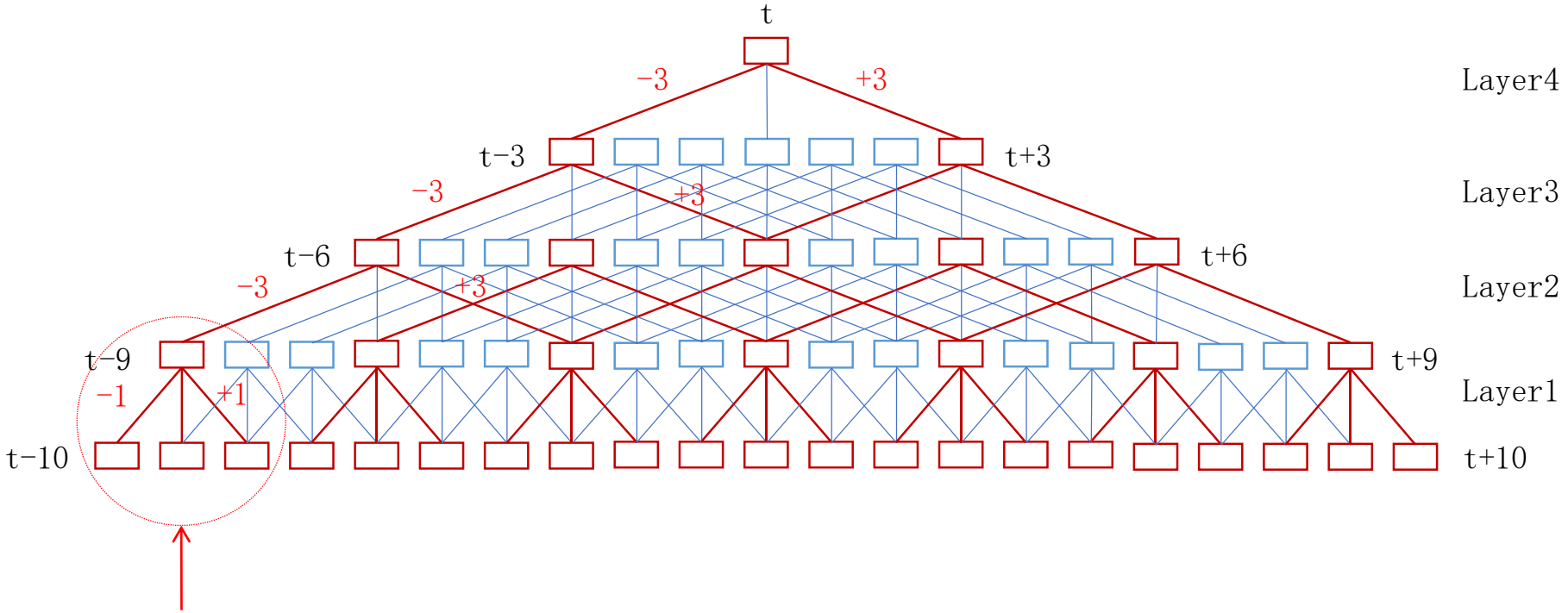
FSMN

- 前馈型序列记忆网络（feed-forward sequential memory network，FSMN）主要是基于传统DNN 结构的改进，在DNN 的隐层旁增加了一个“记忆模块”，这个记忆模块用来存储对判断当前语音帧有用的语音信号的历史信息和未来信息。不同于LSTM/GRU，FSMN只需要等待有限长度的未来语音帧即可。



时延神经网络(TDNN)

- TDNN最早用于音素识别，是CNN的先驱。
- TDNN从底层到顶层，每层的输入都通过下层的上下文窗口获得。
- 能够描述上下层结点之间的时间关系。



输入层特征连续三帧

相当于1维的CNN（核为3×1）

TDNN配置（Kaldi）

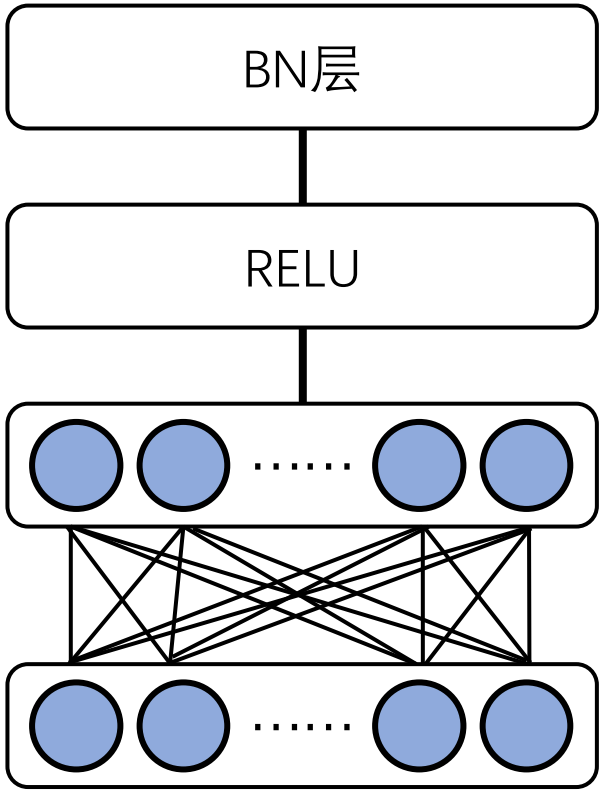
```
input dim=40 name=input
relu-batchnorm-layer name=tdnn1 dim=625
relu-batchnorm-layer name=tdnn2 input=Append(-1,0,1) dim=625
relu-batchnorm-layer name=tdnn3 input=Append(-1,0,1) dim=625
relu-batchnorm-layer name=tdnn4 input=Append(-3,0,3) dim=625
relu-batchnorm-layer name=tdnn5 input=Append(-3,0,3) dim=625
relu-batchnorm-layer name=tdnn6 input=Append(-3,0,3) dim=625
relu-batchnorm-layer name=prefinal-chain input=tdnn6 dim=625 target-rms=0.5
output-layer name=output include-log-softmax=false dim=3759 max-change=1.5
output-layer name=output-xent include-log-softmax=true dim=3759 max-change=1.5
```

TDNN（语音识别）对应的网络结构信息

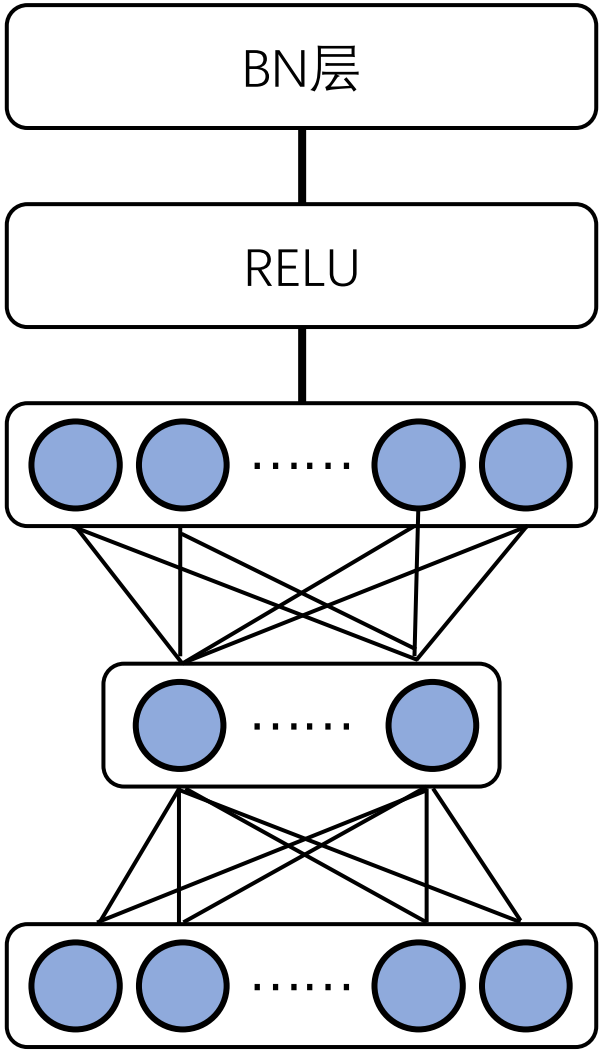
TDNN-F

- 通过奇异值分解(SVD)，把每个权重矩阵因子分解为两个更小的矩阵，减少神经网络参数。
- TDNN-F限定其中一个矩阵为半正交的(semi-orthogonal)。
- 类似残差网络，TDNN-F采用跳层连接，减少梯度的消失。

TDNN-F

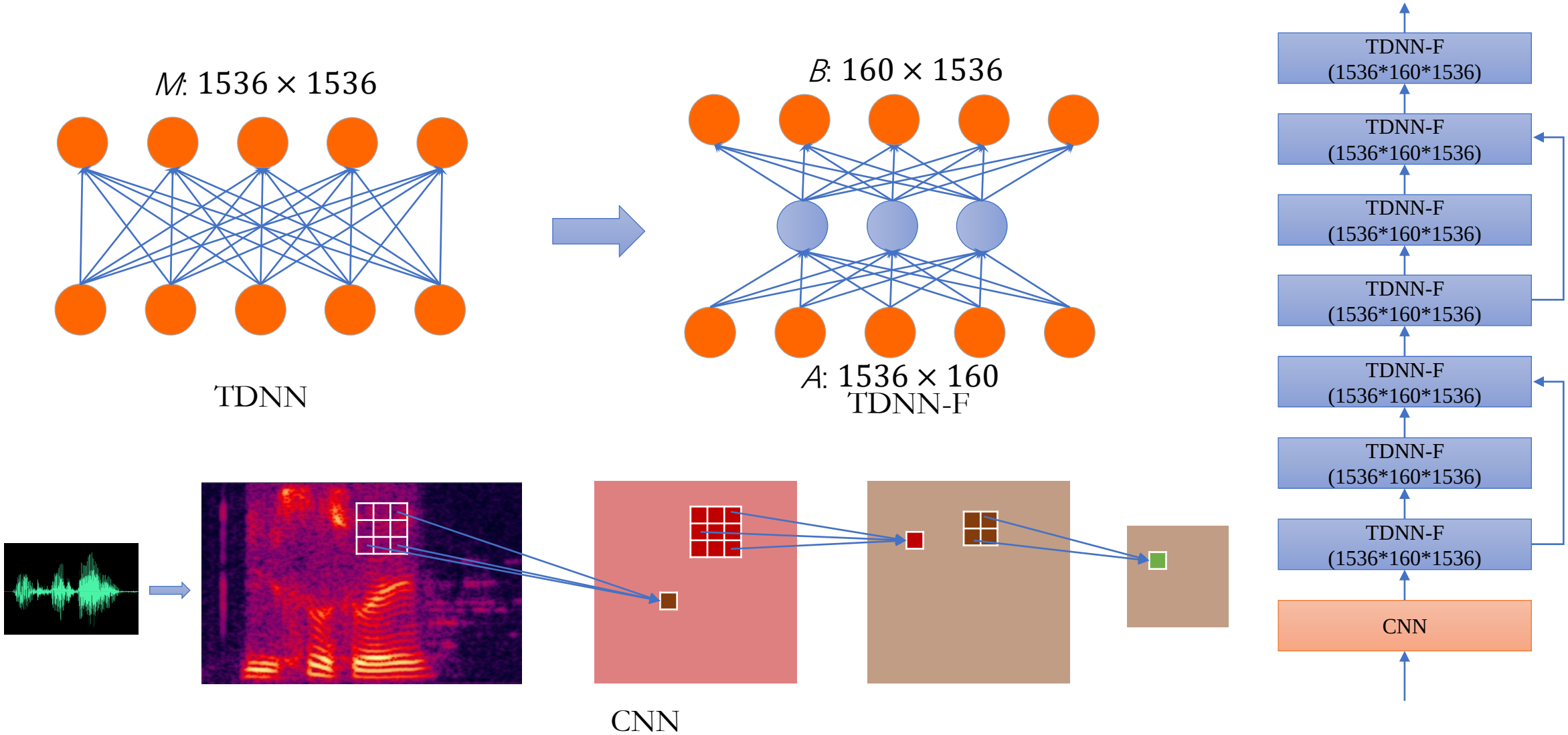


（a）标准前馈神经网络



（b）带有半正交矩阵的网络

CNN+TDNN-F


M: 1536 × 1536
TDNN
B: 160 × 1536
A: 1536 × 160
TDNN-F
CNN
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
TDNN-F (1536*160*1536)
CNN

7.5 本章小结

- 本章详细介绍了语音识别DNN-HMM架构，其中所有的音素HMM共用一个DNN，DNN的输出层节点对应HMM的每个发射状态。DNN相对统计模型GMM，具有更强的表征能力，能够对复杂的语音变化情况进行建模。

- 本章还具体描述了DNN-HMM的训练步骤，包括帧与状态的对齐过程、DNN的后向传播过程等。

- DNN可以有不同的网络结构，本章重点介绍了语音识别常用的CNN、LSTM/GRU、TDNN和TDNN-F模型。