CONVOLUTIONAL NEURAL NETS FOR SPEECH

YU / HUGHES

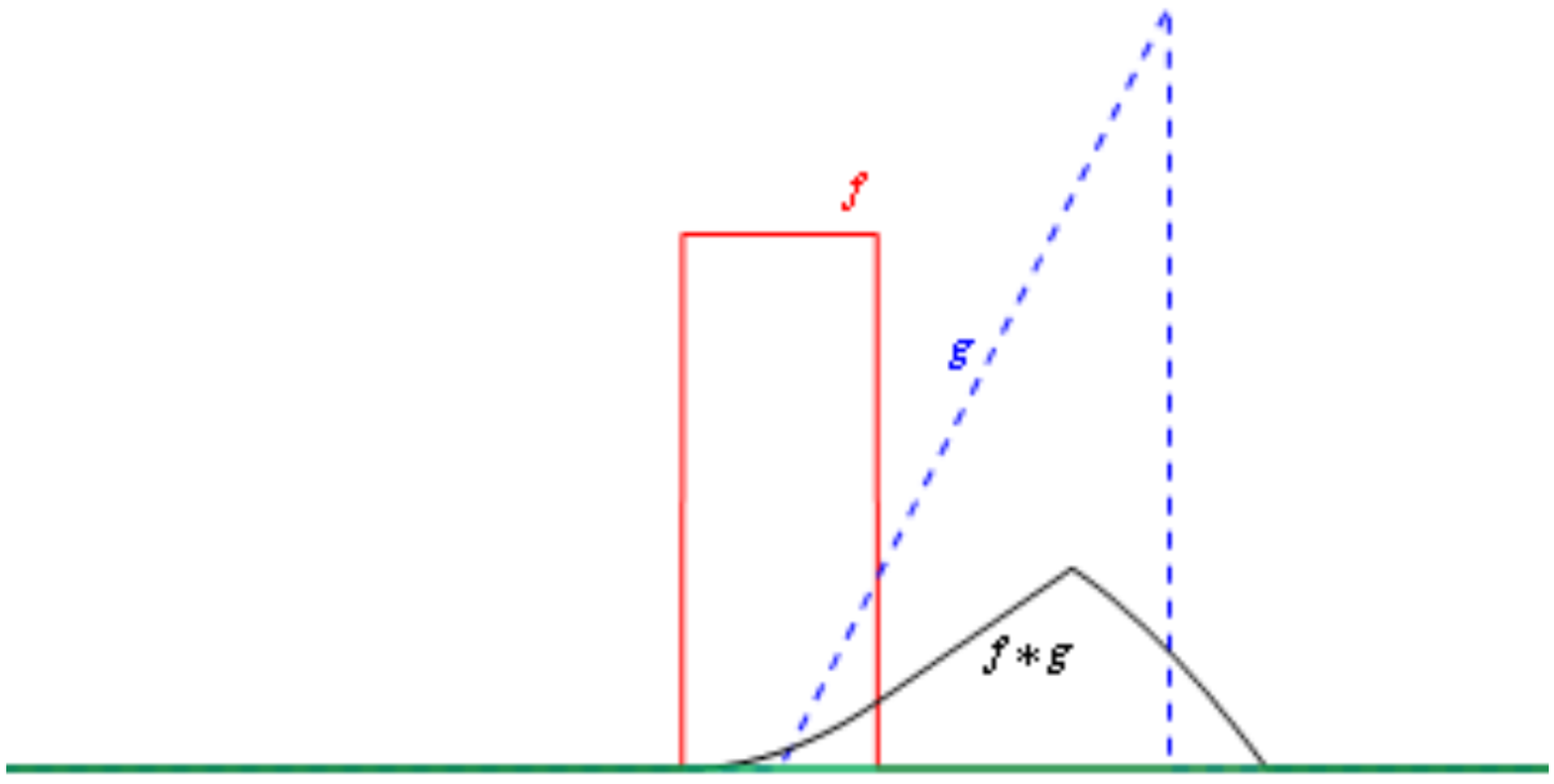
NOVEMBER 9, 2021

UMass Amherst

CONVOLUTION REVIEW

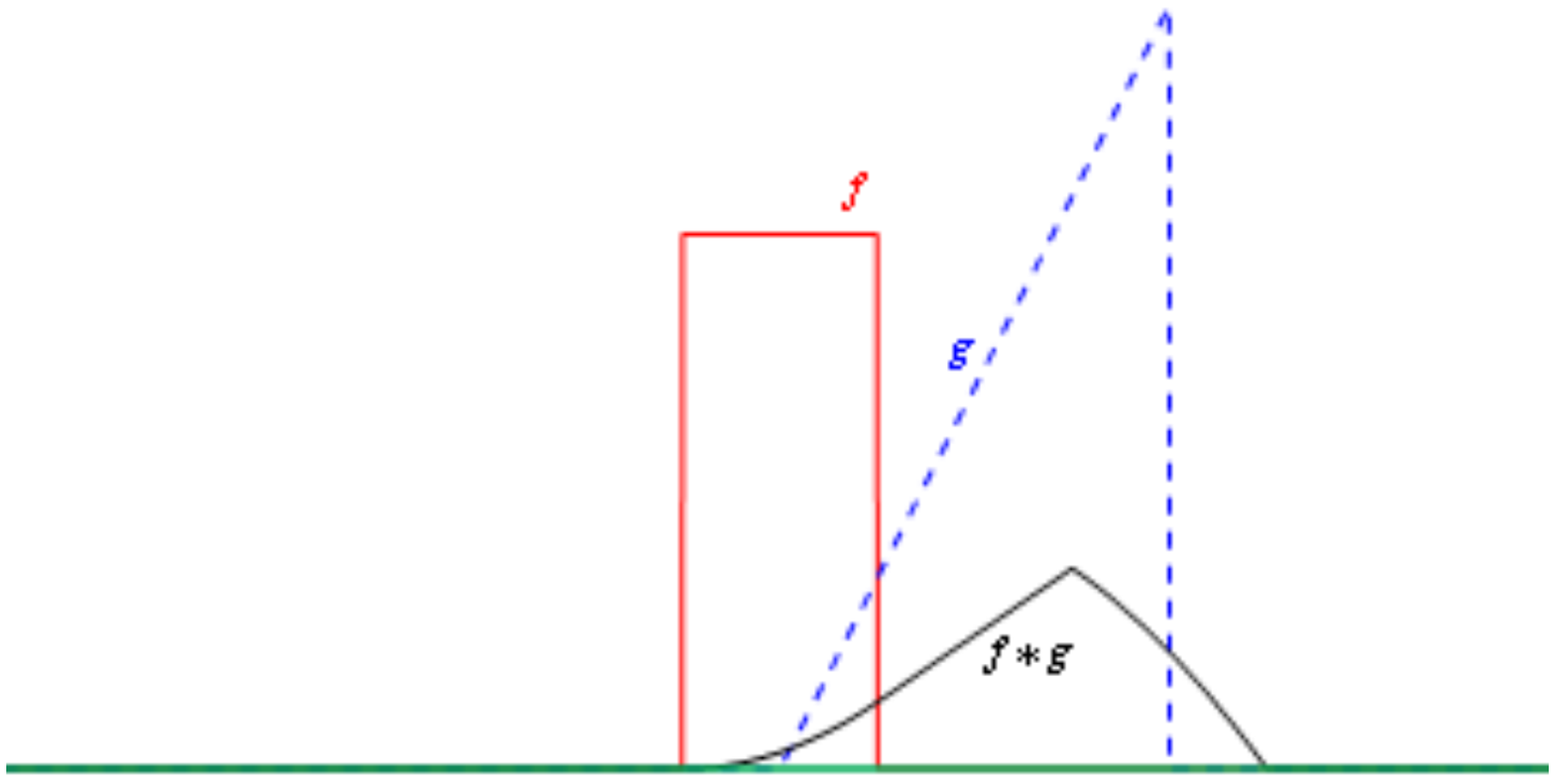
FLIP AND SHIFT! RECT AND RAMP

https://upload.wikimedia.org/wikipedia/commons/f/f8/Convolution_Animation_%28Boxcar_and_Ramp%29.gif



FLIP AND SHIFT! RECT AND RAMP

https://upload.wikimedia.org/wikipedia/commons/f/f8/Convolution_Animation_%28Boxcar_and_Ramp%29.gif

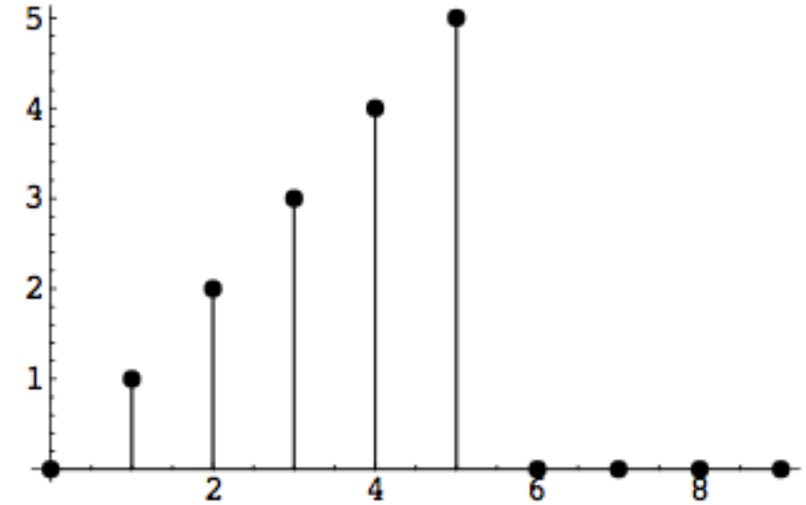


DIGITAL CONVOLUTION

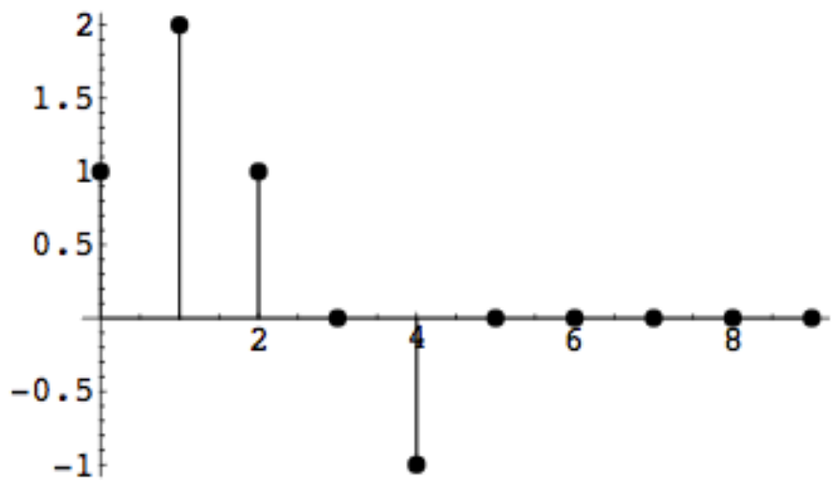
Shown at the right are two functions that we would like to process by convolution.

$f(n)$ has length $L = 6$ and $h(n)$ has length $K = 5$.

The functions have been extended by zero-padding to length $P = L + K - 1 = 10$.



Function $f(n)$.



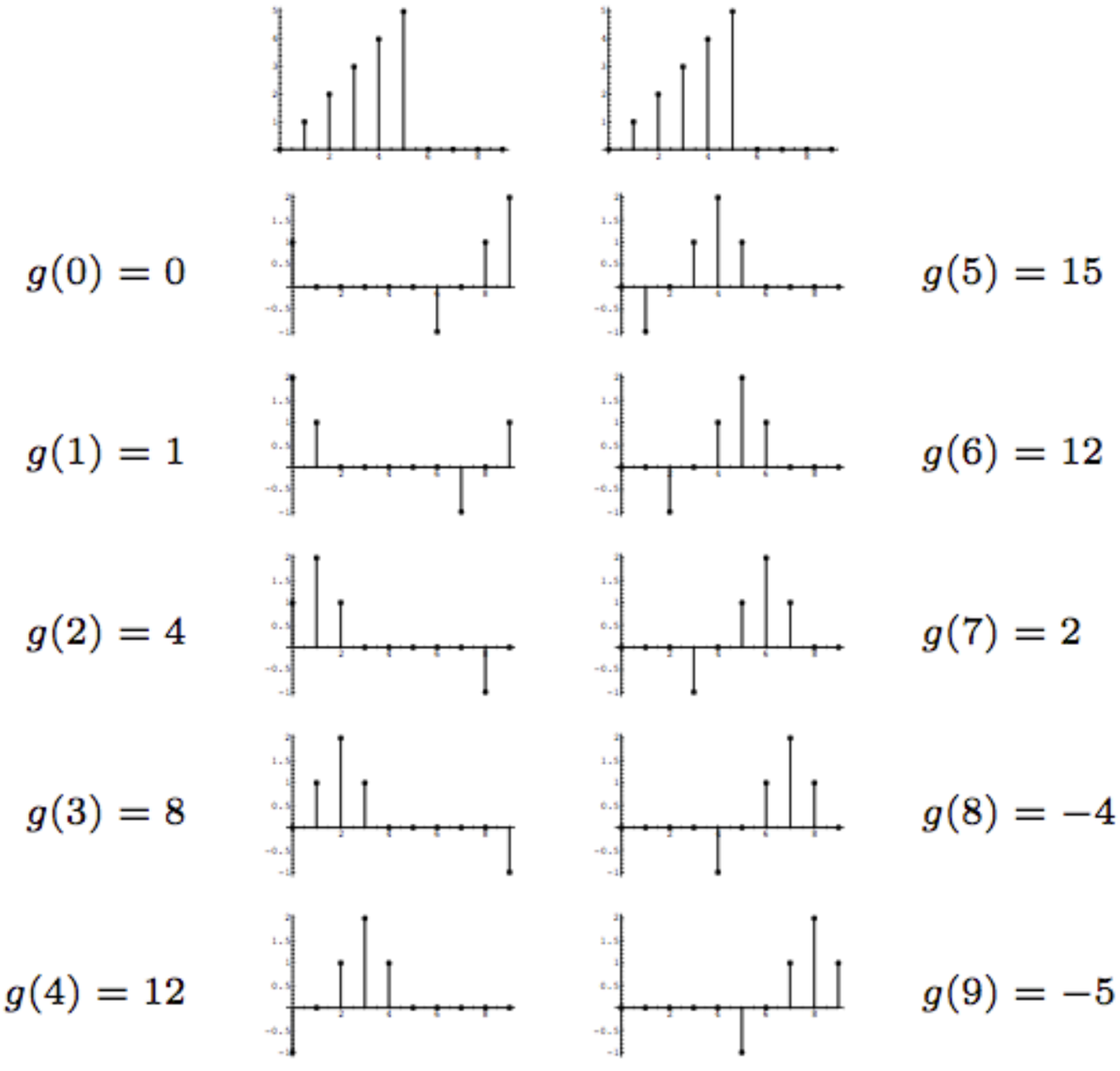
Function $h(n)$.

http://www.cis.rit.edu/class/simg782/lectures/lecture_14/lec782_05_14.pdf

DIGITAL CONVOLUTION

The convolution of $f_P(n)$ with $h_P(n)$ is shown at the right. One period of $f_P(n)$ is shown at the top of each column. The cyclic shifts $h_P(n-m)$ of the filter response are shown in rows 2-6.

The value of $g_P(m)$ is found by multiplying $h_P(m-n)$ with $f_P(n)$ and summing. The result for each shift is listed beside the corresponding filter shift.



http://www.cis.rit.edu/class/simg782/lectures/lecture_14/lec782_05_14.pdf

DIGITAL CONVOLUTION

One Way to Think of Convolution

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

$$x[j] * h[j] = \sum_{k} x[k] \cdot h[j-k]$$

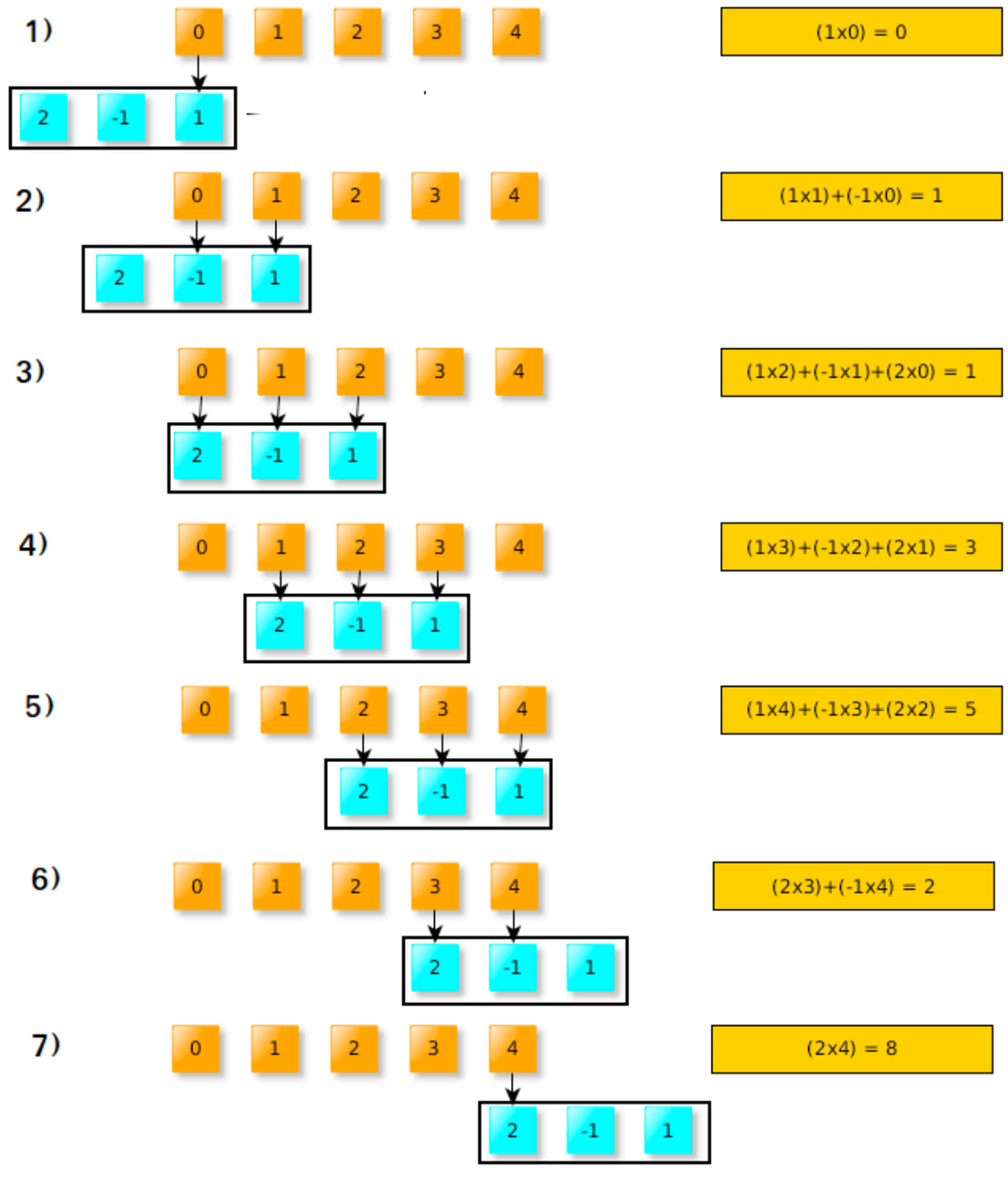
Think of it this way:

- Shift a copy of h to each position t (or discrete position k)
- Multiply by the value at that position $x(t)$ (or discrete sample $x[k]$)
- Add shifted, multiplied copies for all t (or discrete k)

http://situs.biomachina.org/hn06/talks/Baldwin/convolution_filters_new.pdf

FOR A WORKED OUT NUMERICAL EXAMPLE…

https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution



FOR A WORKED OUT NUMERICAL EXAMPLE...

https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution

```
In [1]: import numpy as np

In [2]: x = np.array([0,1,2,3,4])

In [3]: w = np.array([1,-1,2])

In [4]: res = np.convolve(x,w)

In [5]: print res
[0 1 1 3 5 2 8]
```

REVIEW QUESTION: DIGITAL CONVOLUTION

https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution

Work through the convolution example at https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution#doing-by-hand

NEURAL NETS IN SPEECH RECOGNITION ARCHITECTURE

FUNDAMENTAL EQUATION OF SPEECH RECOGNITION

Most likely word sequence

Word sequence

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

Observations (Acoustic feature vectors)

FUNDAMENTAL EQUATION OF SPEECH RECOG

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

$$P(\mathbf{W} \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})}{P(\mathbf{X})}$$

$$\propto P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})$$

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W}) \;\; P(\mathbf{W})$$

FUNDAMENTAL EQUATION OF SPEECH RECOG

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

$$P(\mathbf{W} \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})}{P(\mathbf{X})}$$

$$\propto P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})$$

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W})\, P(\mathbf{W})$$

Acoustic Model

FUNDAMENTAL EQUATION OF SPEECH RECOG

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X})$$

$$P(\mathbf{W} \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})}{P(\mathbf{X})}$$

$$\propto P(\mathbf{X} \mid \mathbf{W})P(\mathbf{W})$$

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W}) \; P(\mathbf{W})$$

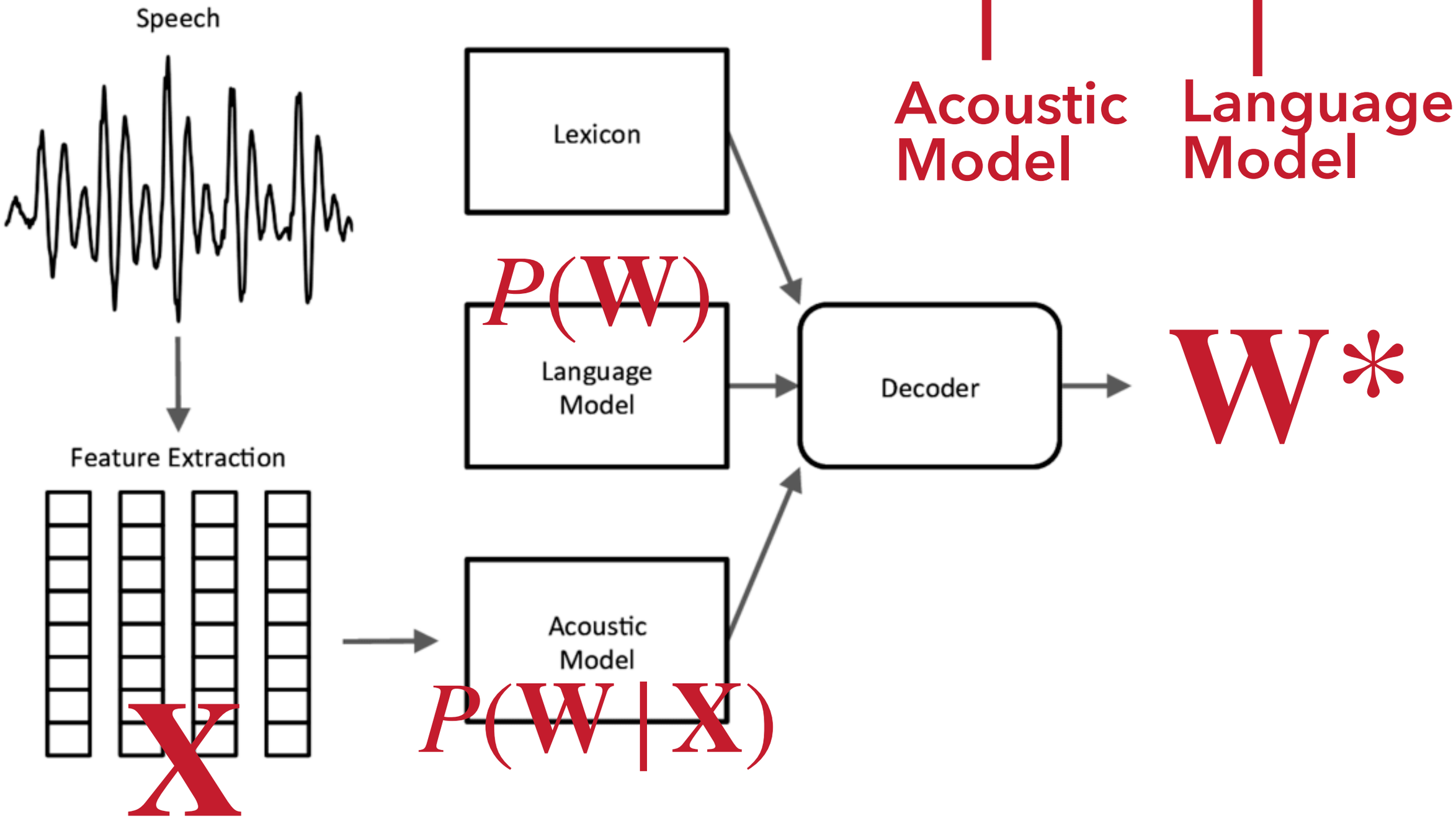
Acoustic Model

Language Model

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} P(\mathbf{X} \mid \mathbf{W}) \, P(\mathbf{W})$$

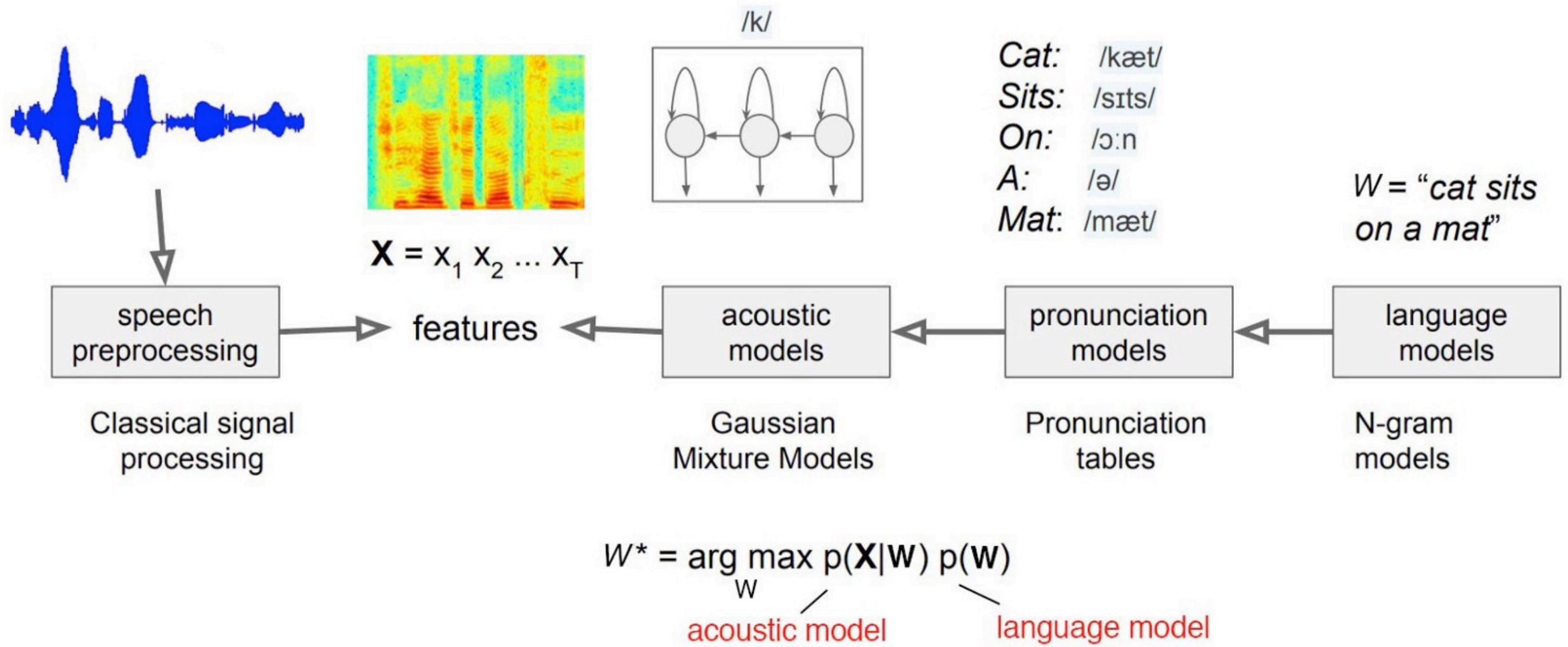
Acoustic Model

Language Model



https://www.researchgate.net/profile/Tom-Nuorivaara/publication/307560344/figure/fig1/AS:401591556427776@1472758296782/The-structure-of-a-typical-ASR-system.png

CLASSICAL SPEECH RECOGNITION



Pronunciation model = Lexicon

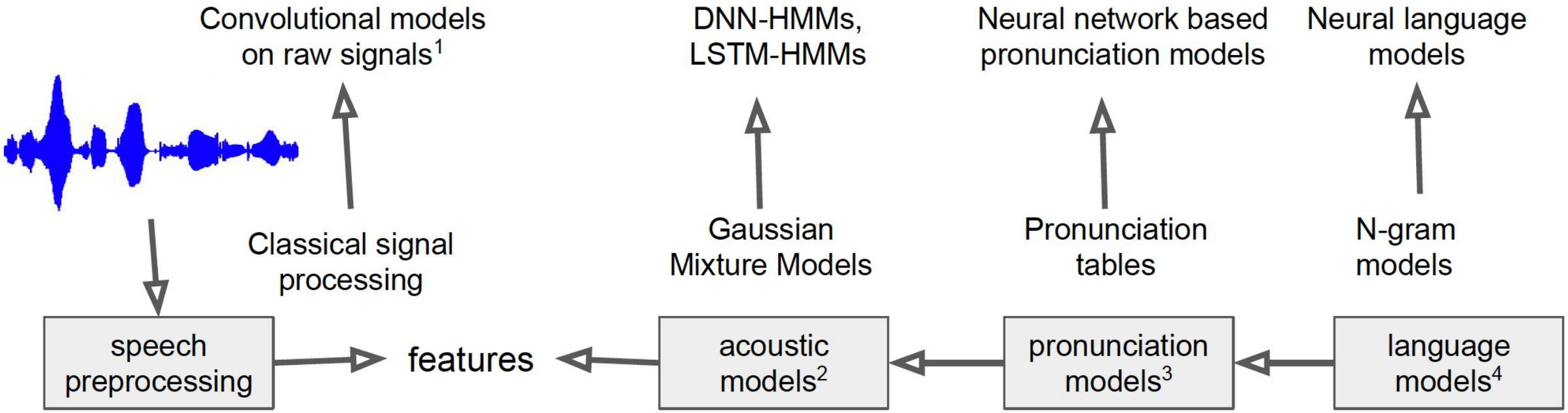
https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf

NEURAL NETWORK "INVASION"

Speech Recognition -- the neural network invasion

- Each of the components seems to be better off with a neural network

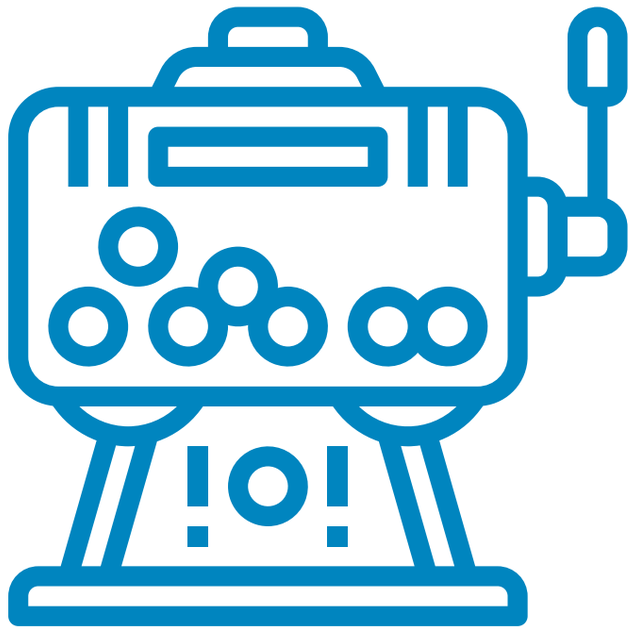


1. Jaitly, Navdeep, and Geoffrey Hinton. "Learning a better representation of speech soundwaves using restricted boltzmann machines." *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011.
2. Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal Processing Magazine* 29.6 (2012): 82-97.
3. Rao, Kanishka, et al. "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks." *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.
4. Mikolov, Tomas, et al. "Recurrent neural network based language model." *Interspeech*. Vol. 2. 2010.

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lectures/cs224n-2017-lecture12.pdf

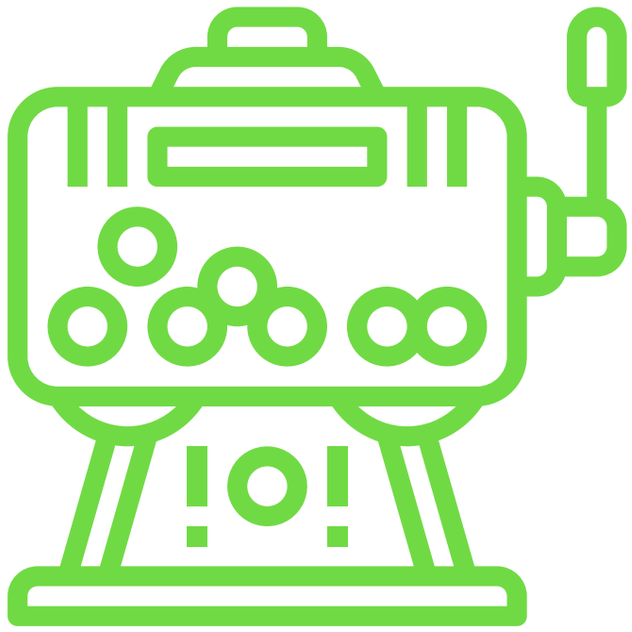
OBSERVATION PROBABILITIES

Machine m_1



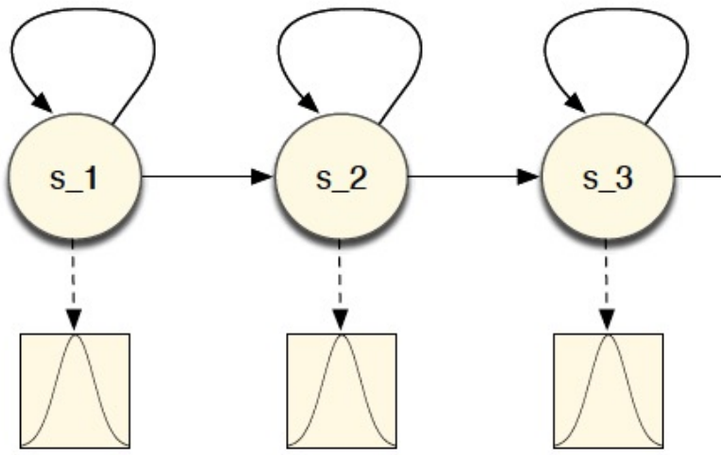
60% grapefruit
40% apple

Machine m_2

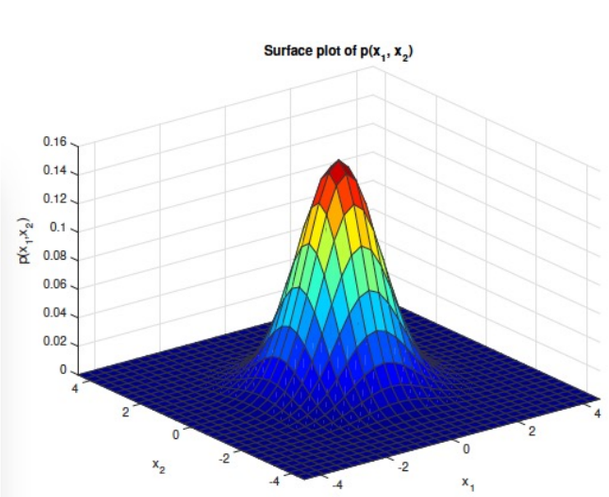


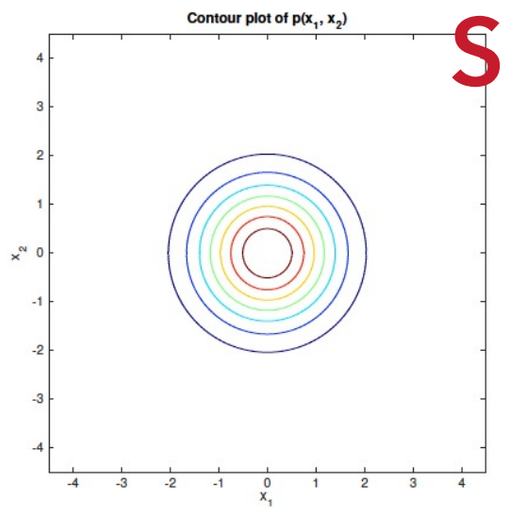
10% grapefruit
90% apple

GAUSSIAN MODEL OF OBSERVATION PROBS



/s/

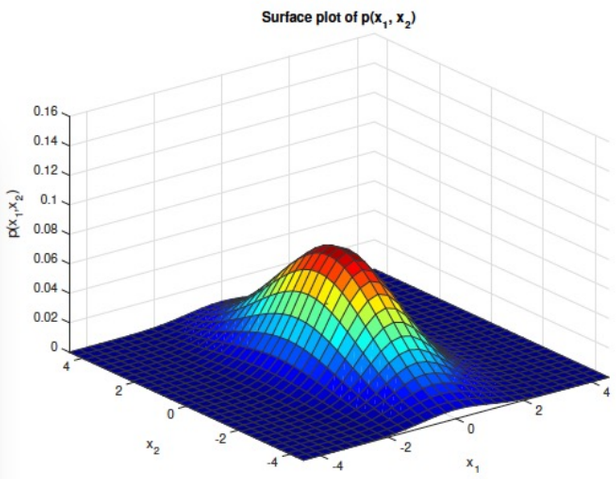


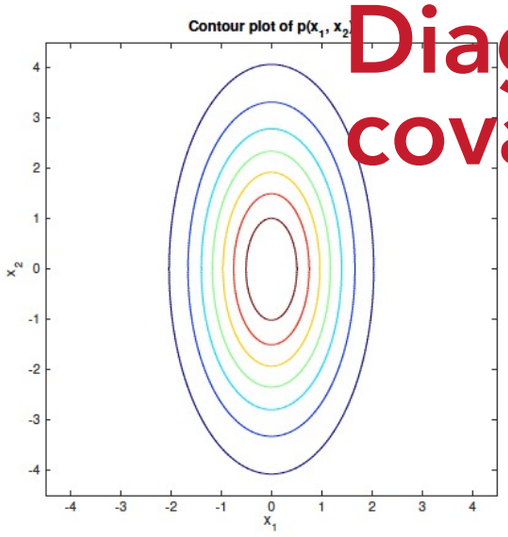


Spherical

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \rho_{12} = 0$$

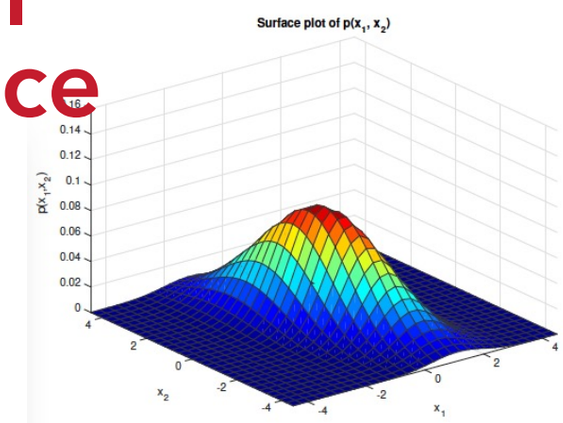
Diagonal covariance

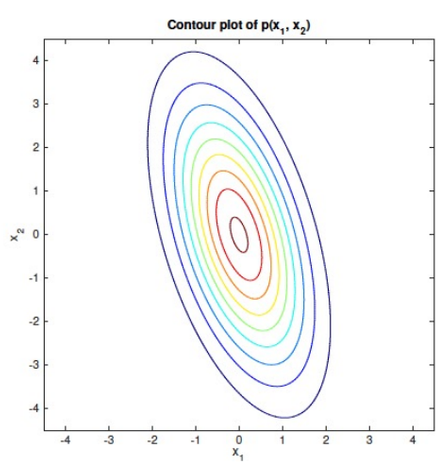




$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \qquad \rho_{12} = 0$$

Full covariance

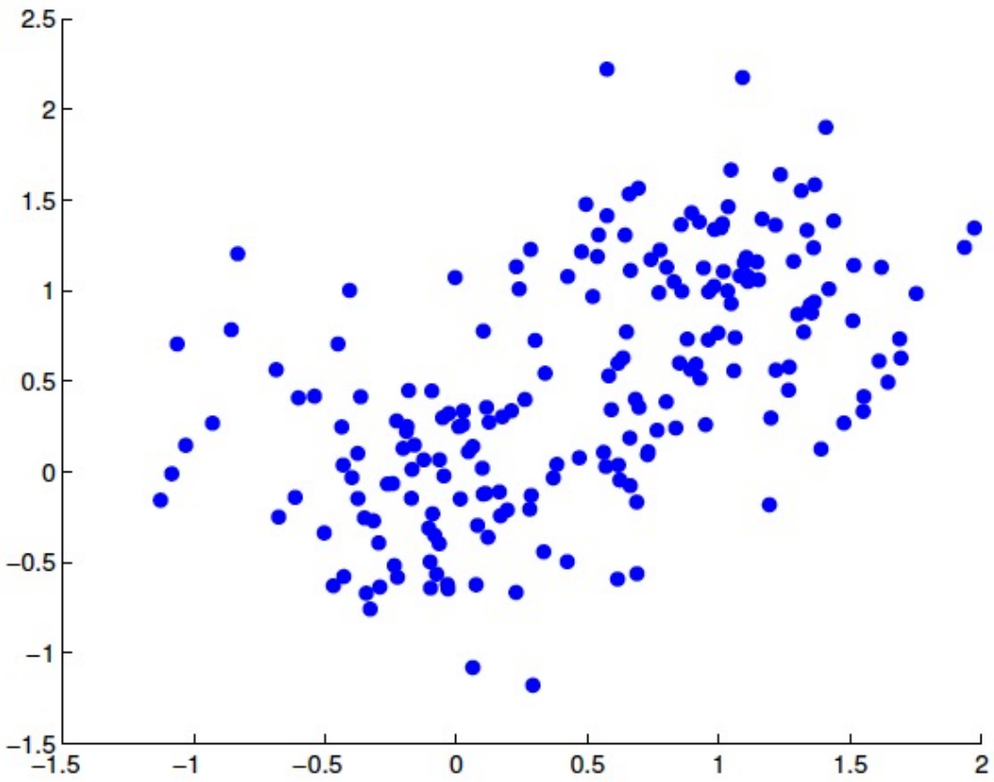




$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1 & -1 \\ -1 & 4 \end{pmatrix} \qquad \rho_{12} = -0.5$$

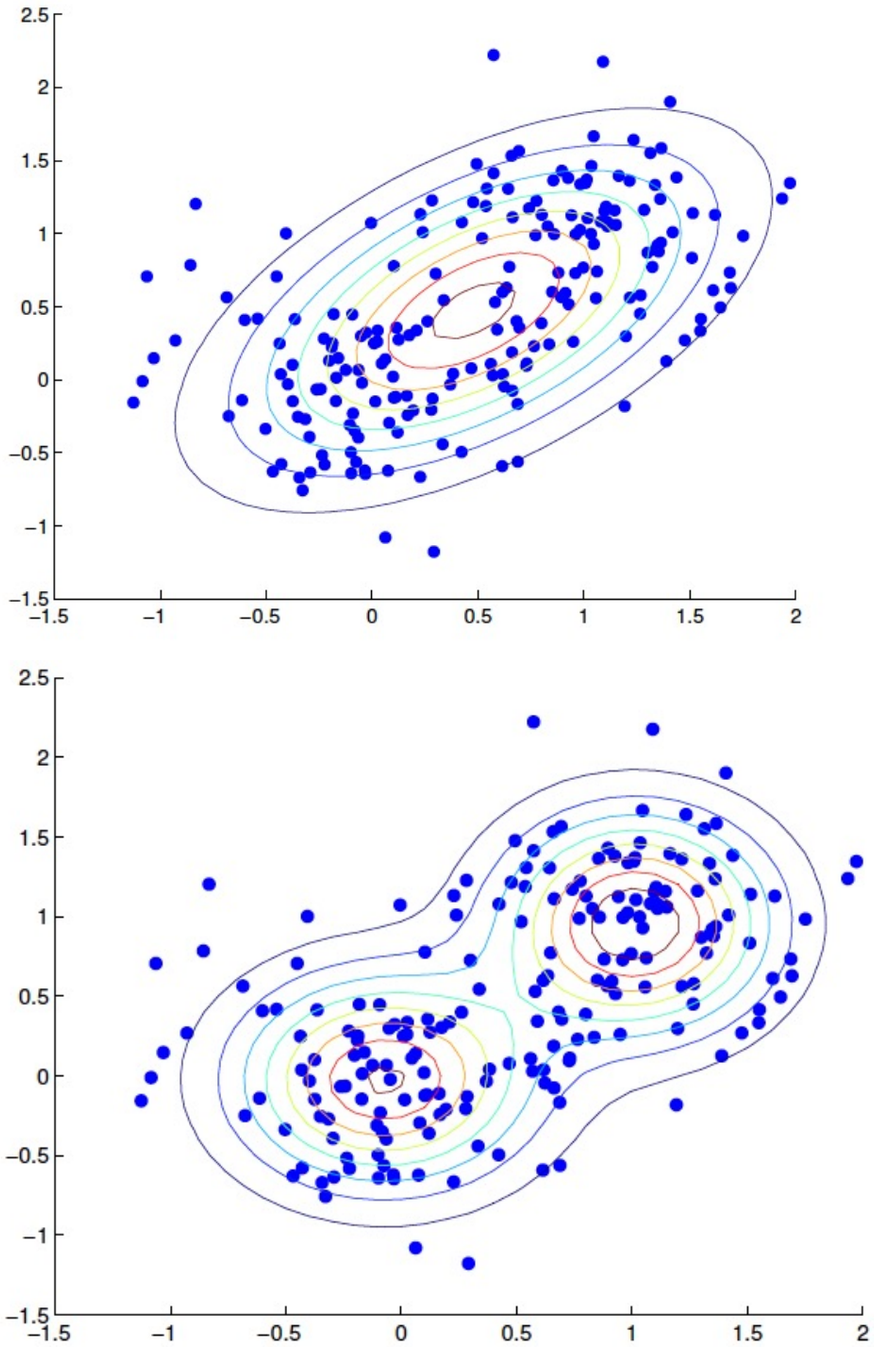
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr06-gmms.pdf

GAUSSIANS TO GAUSSIAN MIXTURE MODELS



$$\mu_1 = (0,0)^T \qquad \mu_2 = (1,1)^T$$

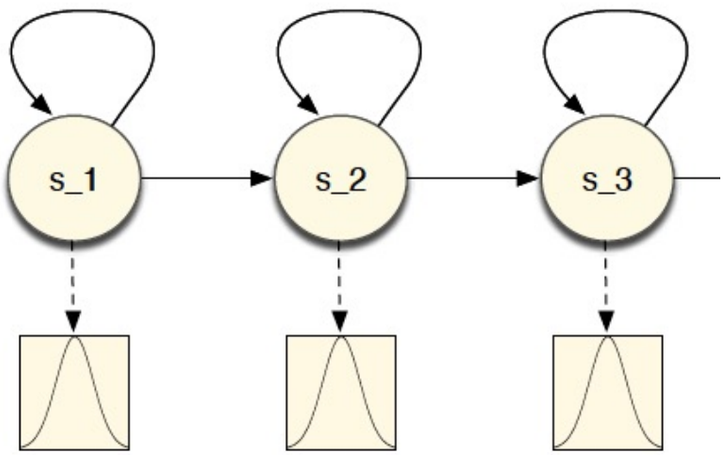
$$\Sigma_1 = \Sigma_2 = 0.2\,\mathrm{I}$$



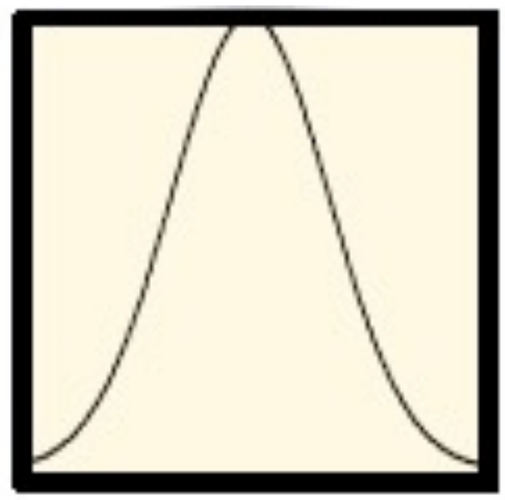
Fitted with a two component GMM using EM

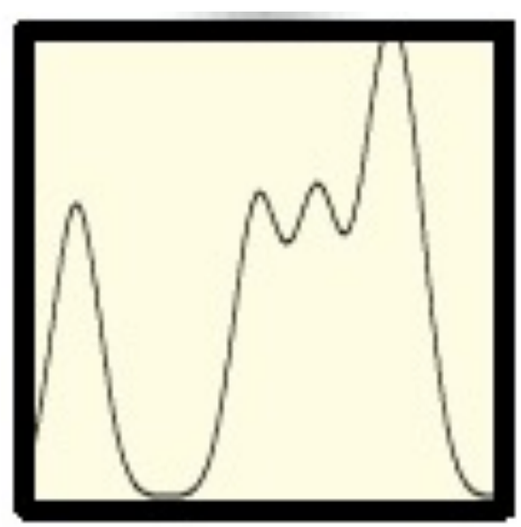
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr06-gmms.pdf

GAUSSIANS TO GAUSSIAN MIXTURE MODELS



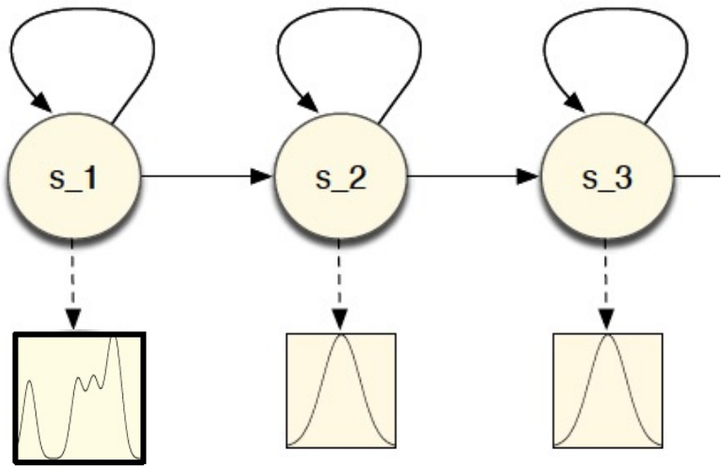
/s/



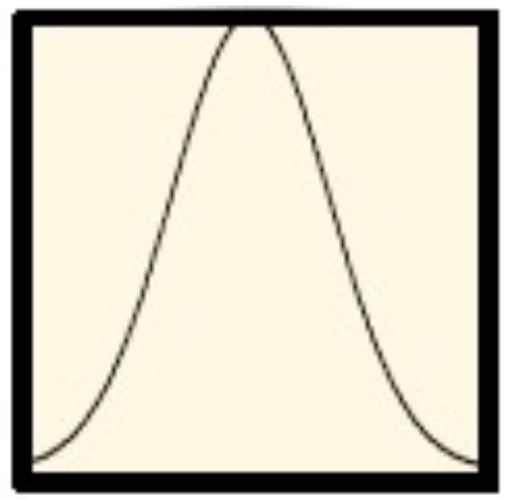


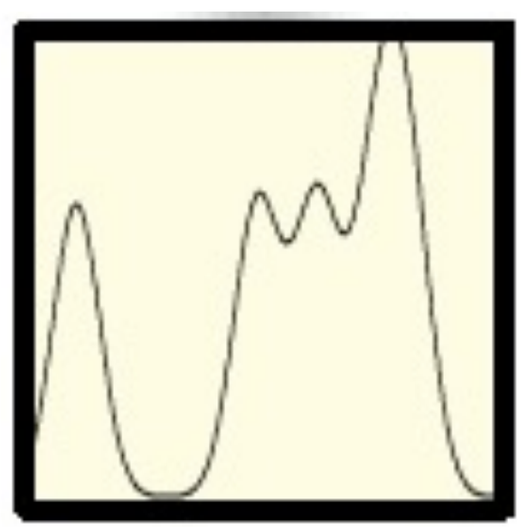
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

GAUSSIANS TO GAUSSIAN MIXTURE MODELS



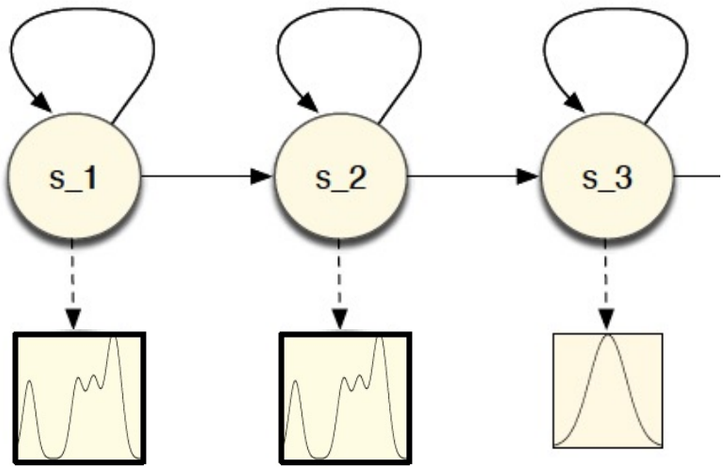
/s/



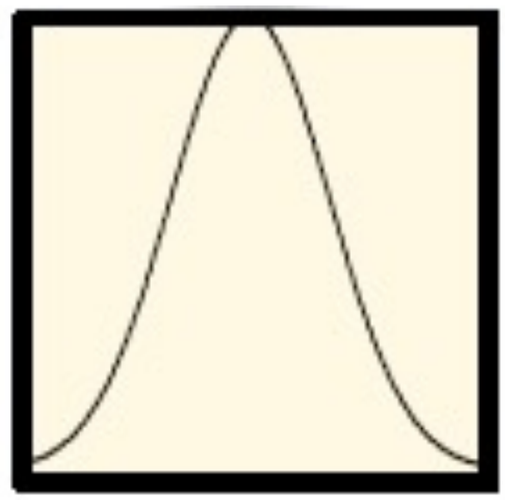


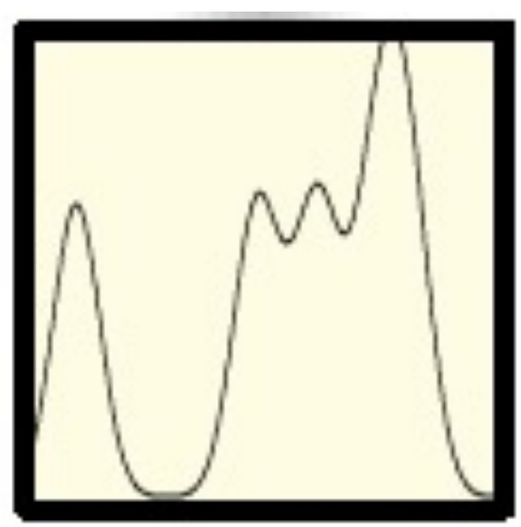
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

GAUSSIANS TO GAUSSIAN MIXTURE MODELS



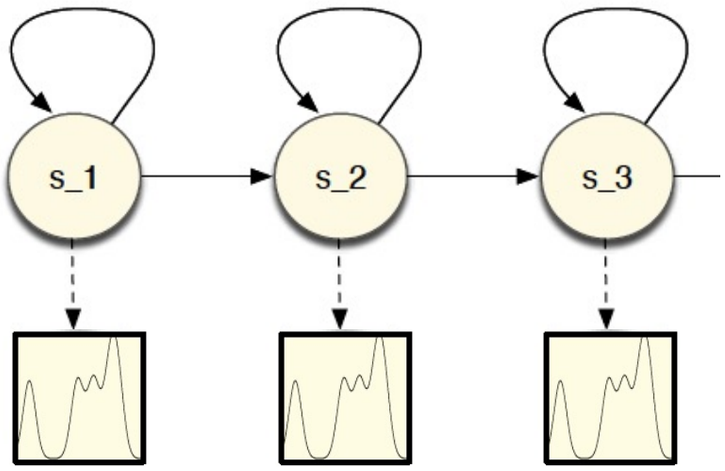
/s/



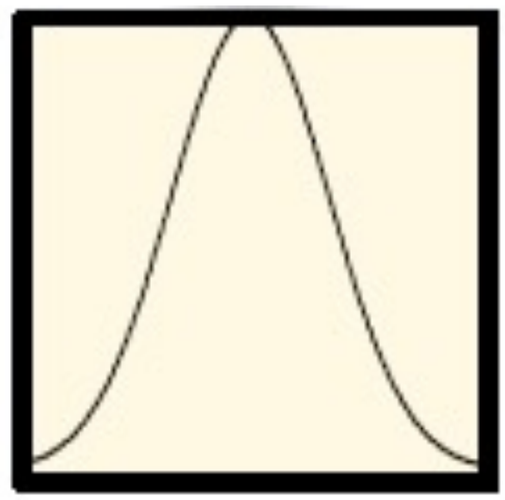


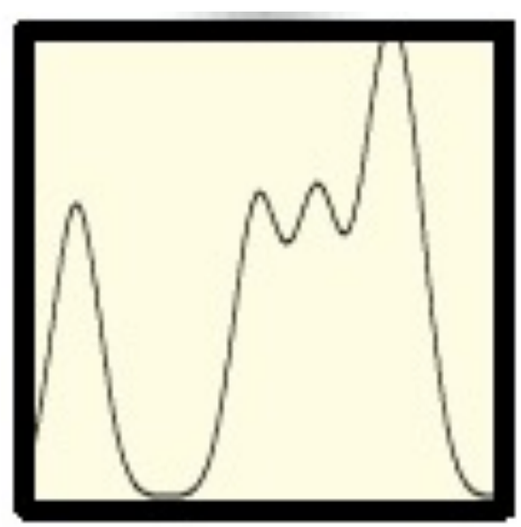
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

GAUSSIANS TO GAUSSIAN MIXTURE MODELS



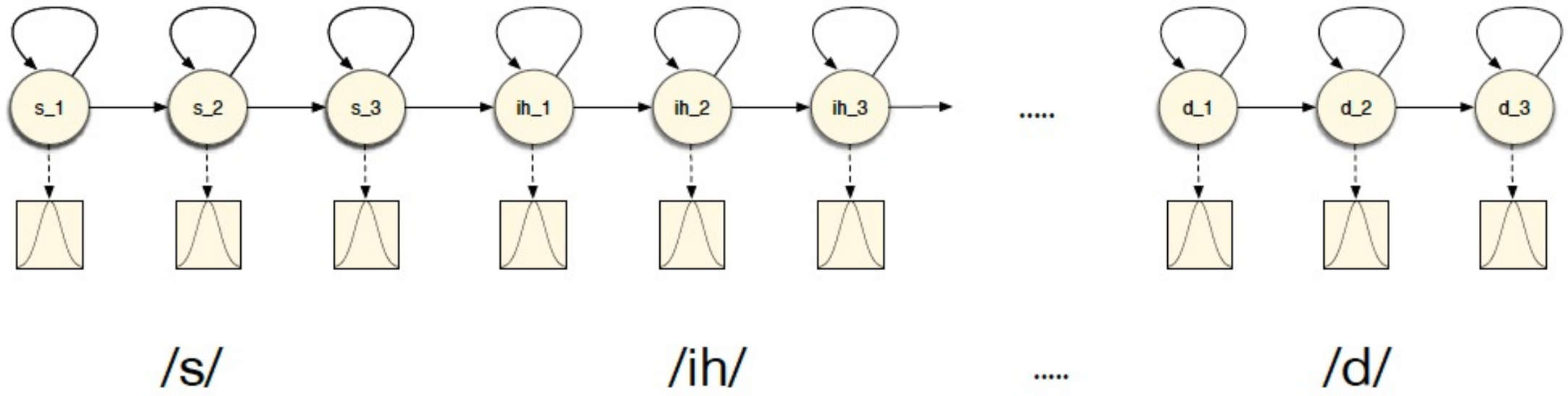
/s/

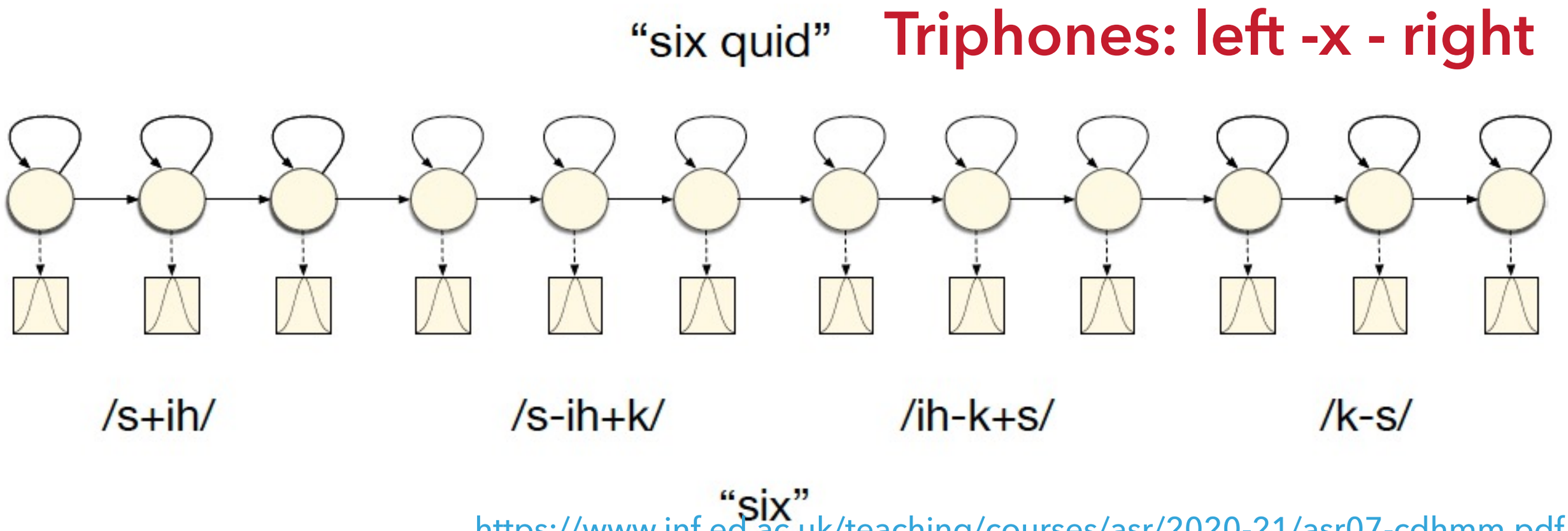




https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

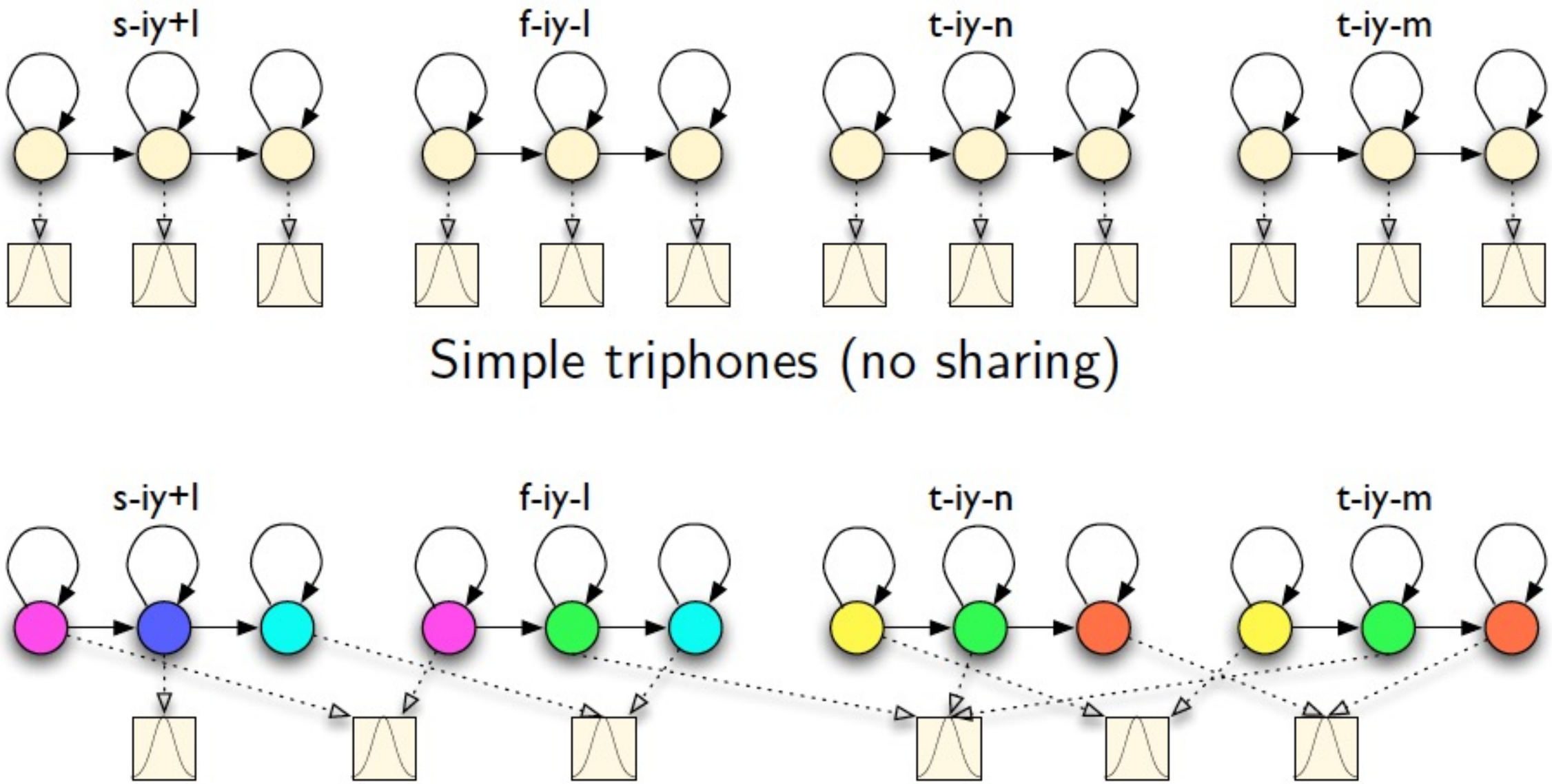
CONTEXT DEPENDENT PHONE MODELS





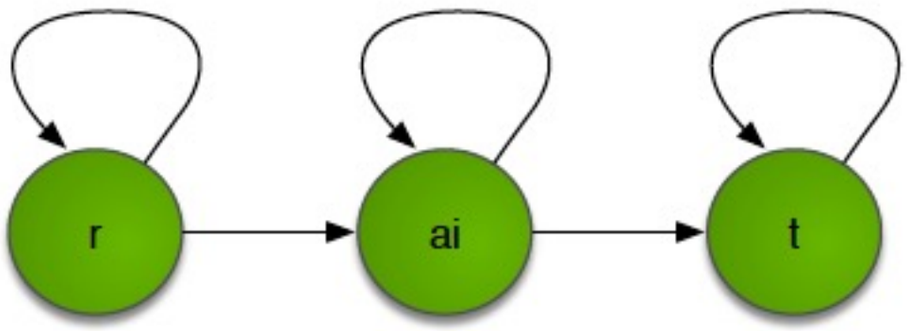
https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

TOO MANY PARAMETERS: SHARE STATES

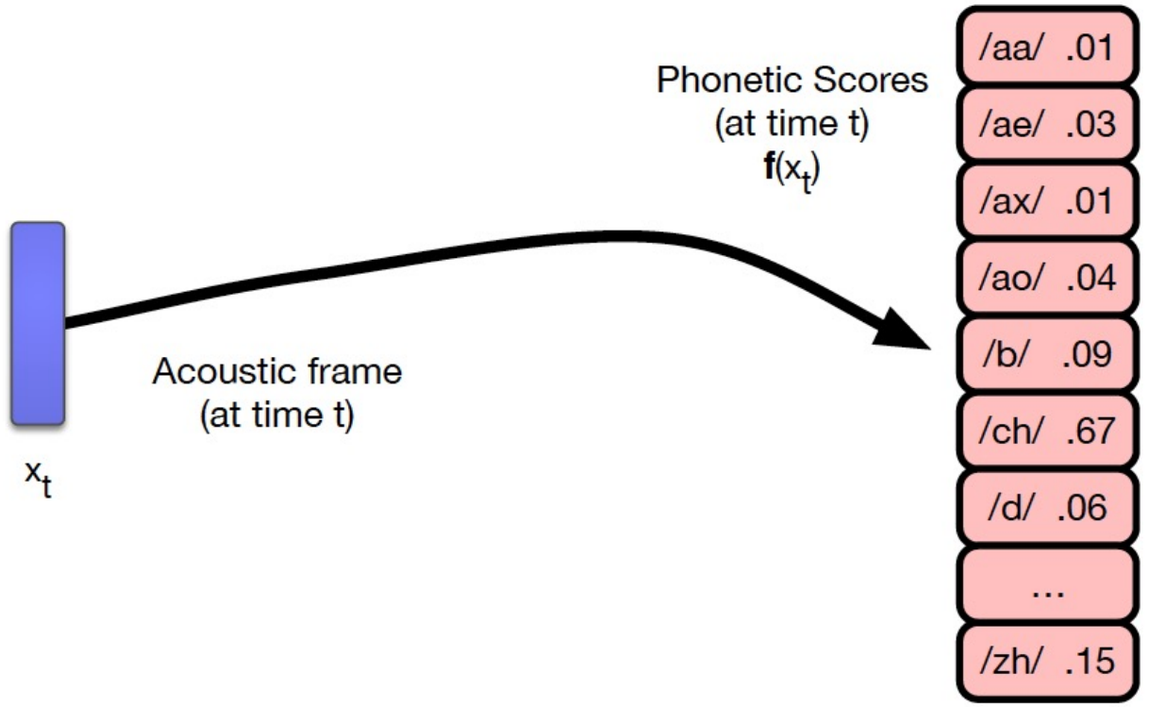


https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr07-cdhmm.pdf

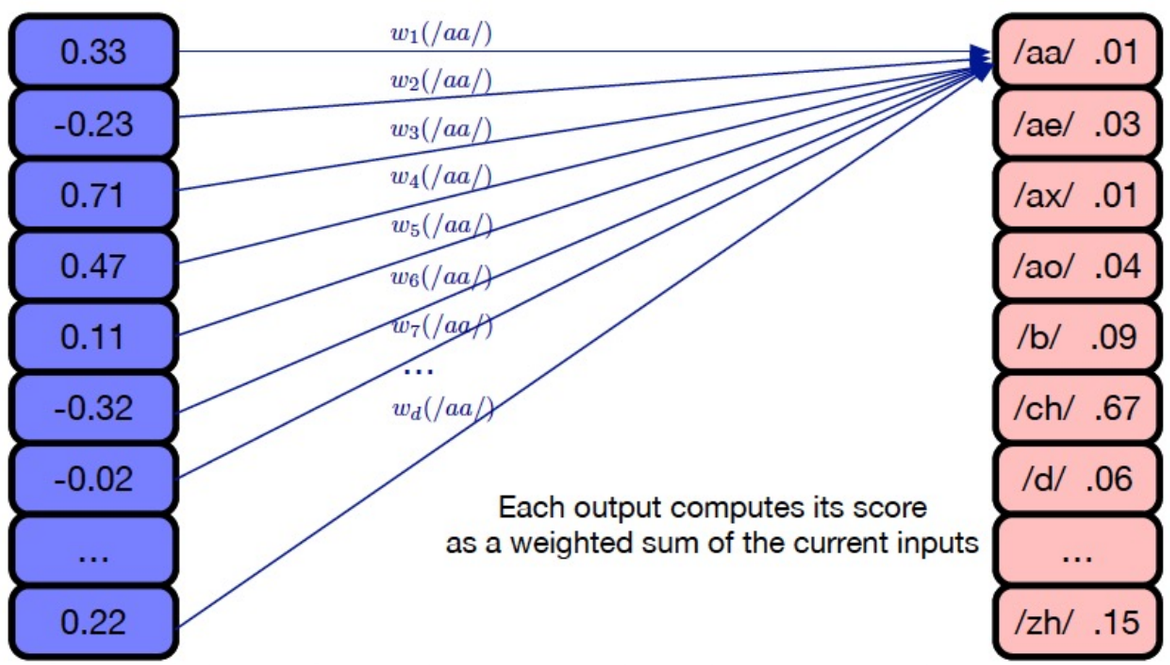
FROM GMMS TO NEURAL NETS



One-state-per phone

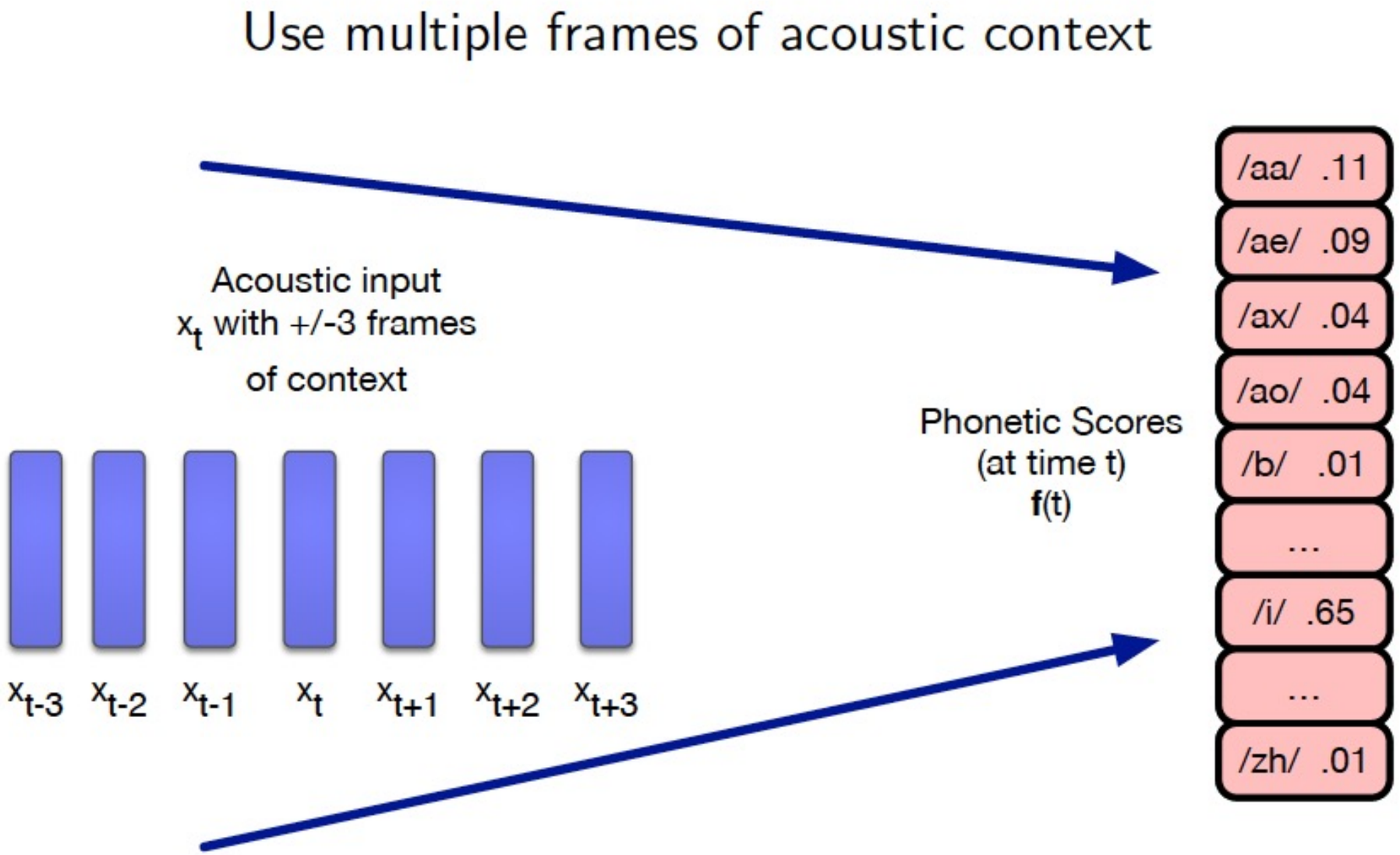


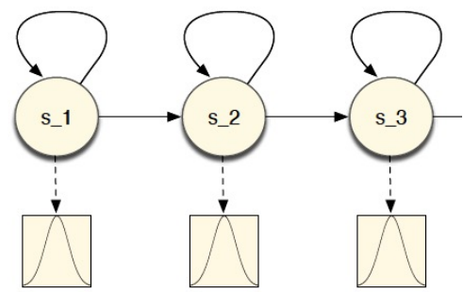
Compute the phonetic scores using a single layer neural network (linear regression!)



https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr10-nnintro.pdf

MODELING ACOUSTIC CONTEXT WITH NNS





/s/

Correlated features! No more conditional independence, cf.

https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr10-nnintro.pdf

NNS ALLOW UNCORRELATED FEATURES

- Can't use filter bank features because strongly correlated with one another, requiring full covariance and diagonal covariance Gaussians for GMMs
- NNs don't require uncorrelated feature vectors
  - Can use filter bank vectors
  - Can use multiple frames to model context-dependency

https://www.inf.ed.ac.uk/teaching/courses/asr/2020-21/asr10-nnintro.pdf

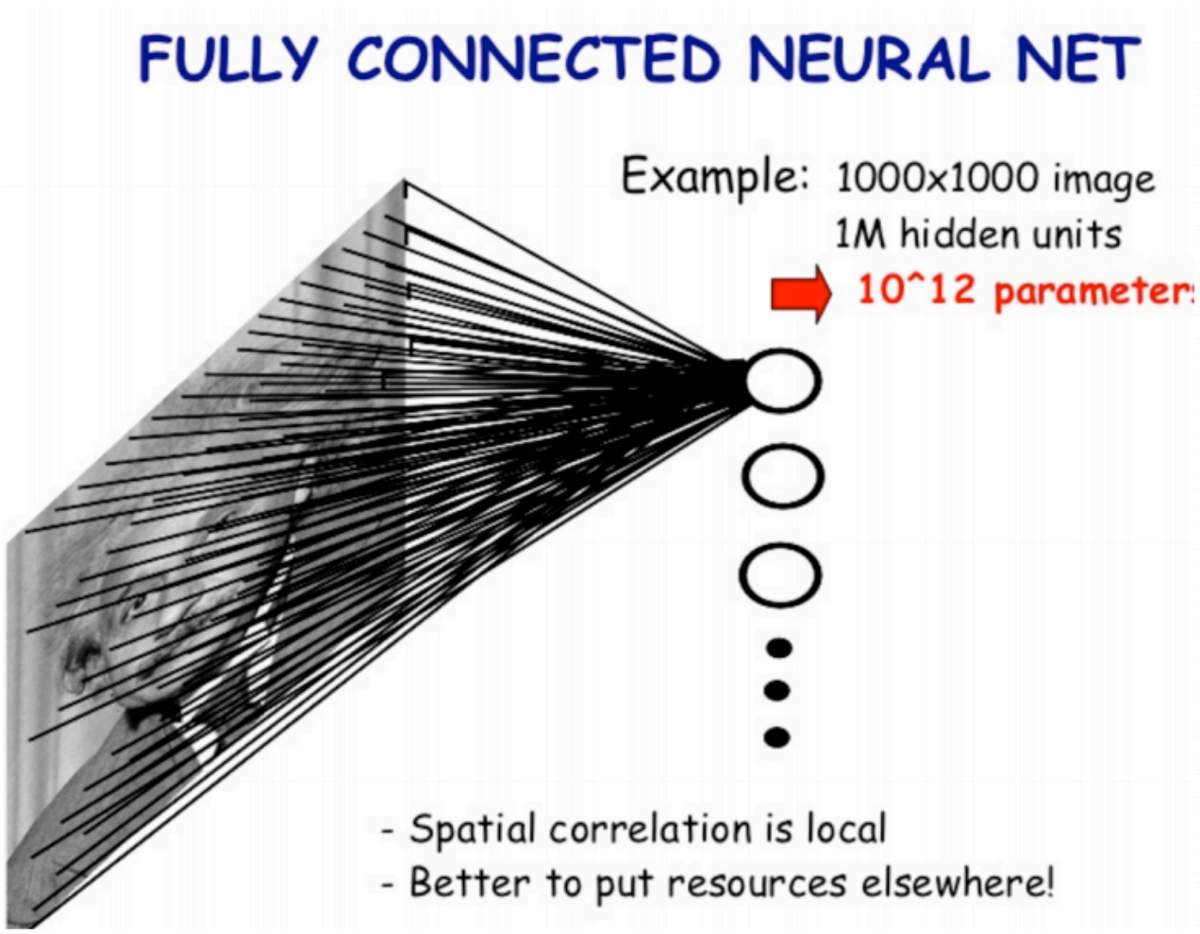
CNNS FOR SPEECH

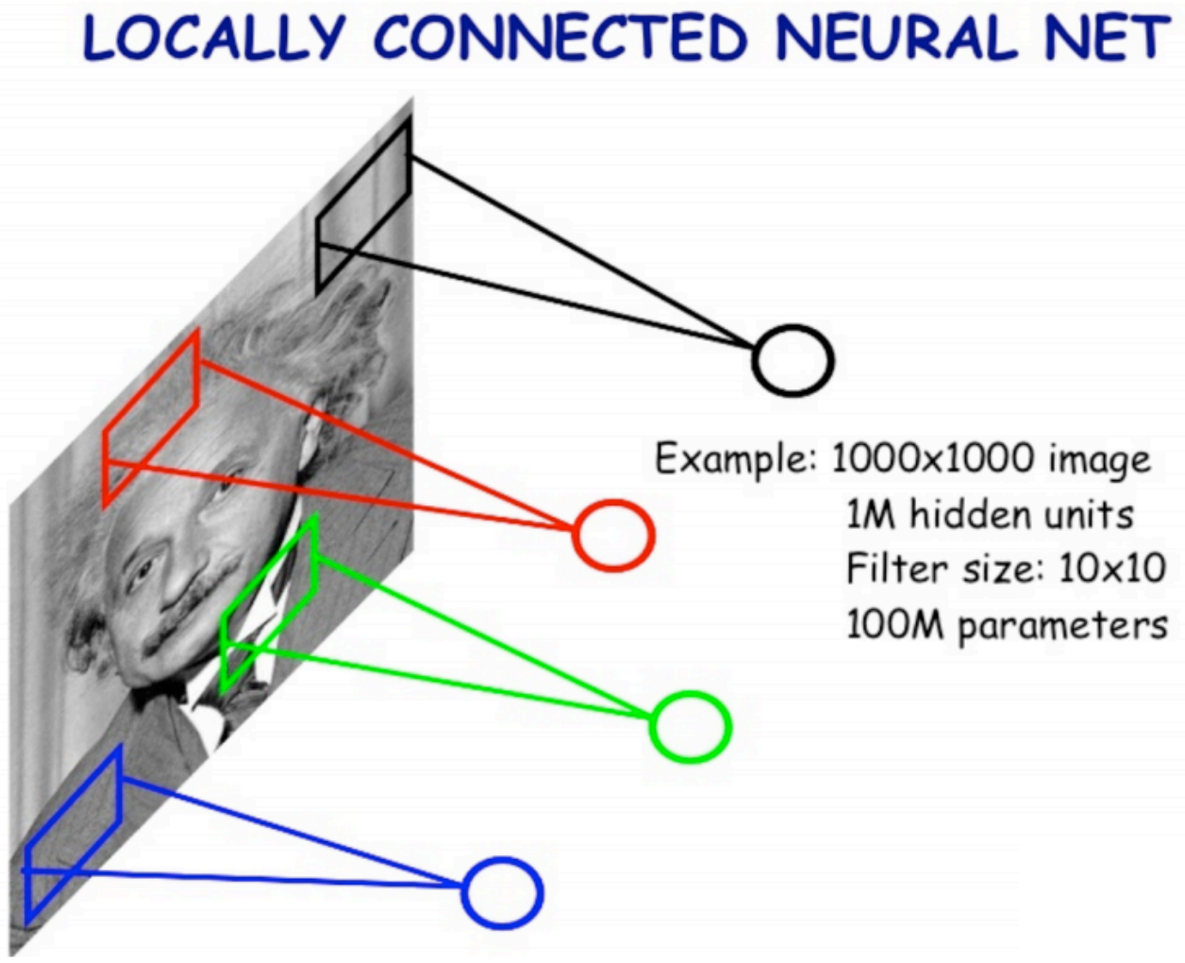
TIME-DELAY NEURAL NETS

- In deep neural nets, we model acoustic context by including neighboring frames in input layer
- In TDNNs (time-delay neural nets), each layer processes a window of context from previous layer
- Each higher layer has a wider "receptive" field, wider acoustic context
- "TDNNs are 1D convolutional neural nets (without pooling and with dilations)"
  - (What part of them is 1D?)

https://kaleidoescape.github.io/tdnn/

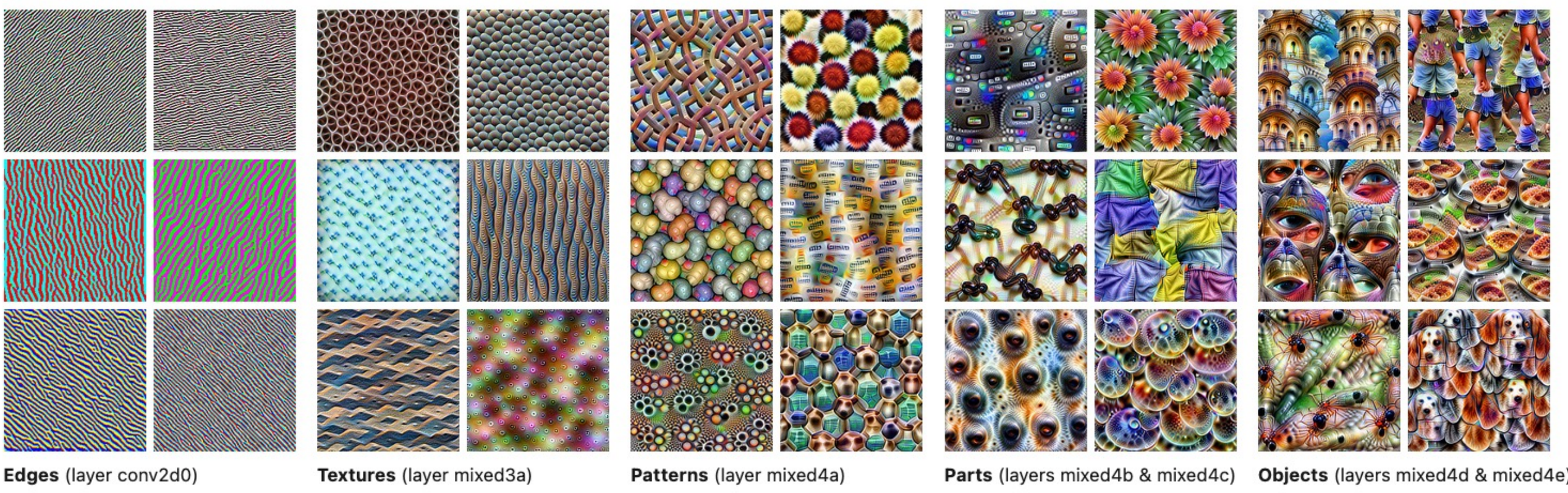
CNNS AND LOCAL FEATURES





https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolutional_neural_networks

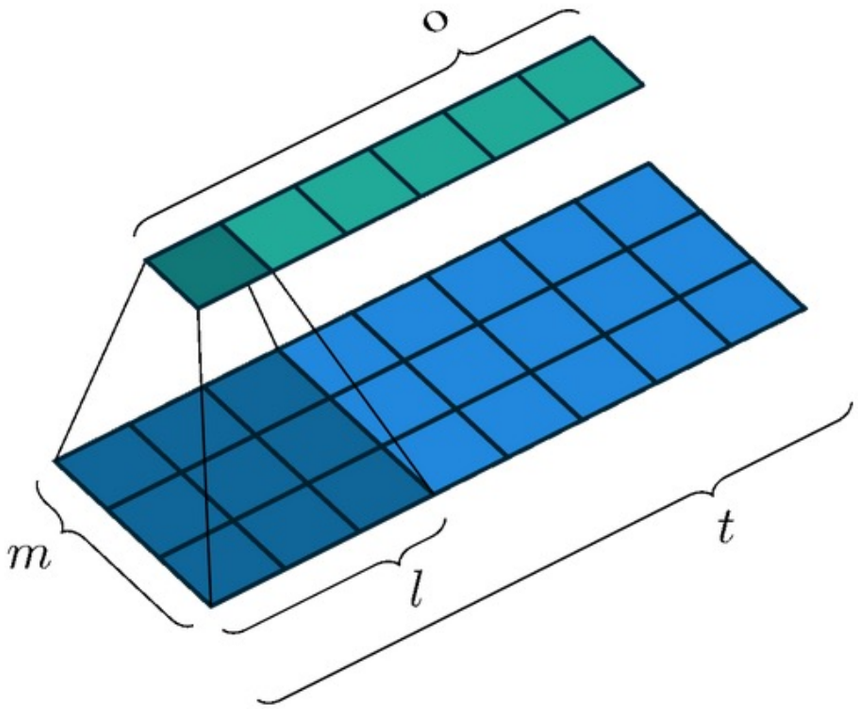
CNNS AND LOCAL FEATURES

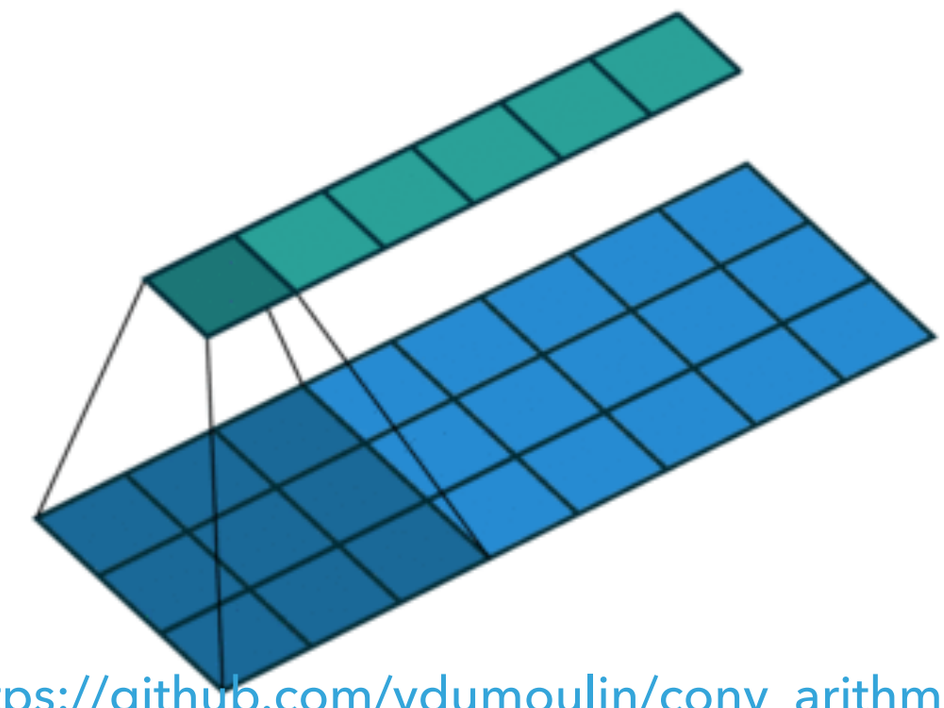


Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

https://distill.pub/2017/feature-visualization/

TIME-DELAY NEURAL NETS



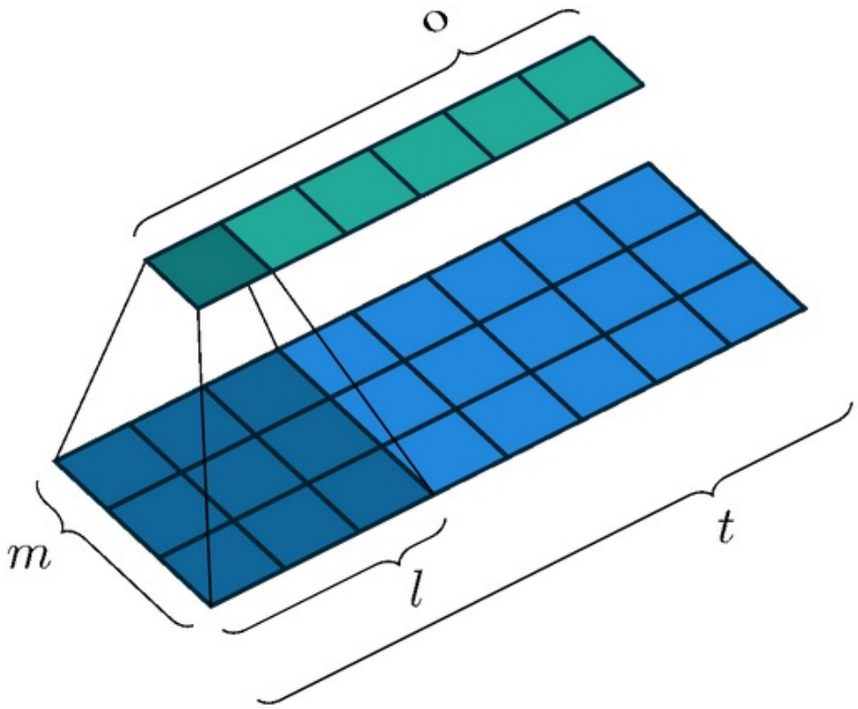


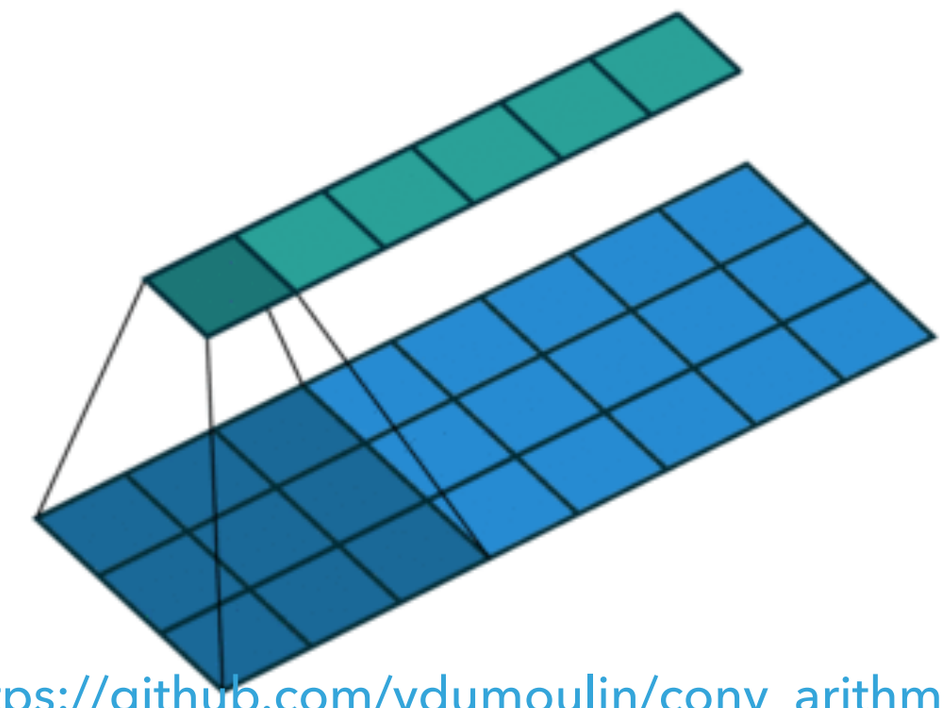
https://github.com/vdumoulin/conv_arithmetic

- Input m-*dimensional* filter bank vector for each frame of speech signal, with *t* frames in total (light blue)
- Kernel/filter weight matrix **W** with height of *m* and width of *l* (dark blue)
- Output of length *o*, determined by number of times kernel can fit across the frames of input: (green)

https://kaleidoescape.github.io/tdnn/

TIME-DELAY NEURAL NETS





https://github.com/vdumoulin/conv_arithmetic

- Input m-*dimensional* filter bank vector for each frame of speech signal, with *t* frames in total (light blue)
- Kernel/filter weight matrix **W** with height of *m* and width of *l* (dark blue)
- Output of length *o*, determined by number of times kernel can fit across the frames of input: (green)

https://kaleidoescape.github.io/tdnn/

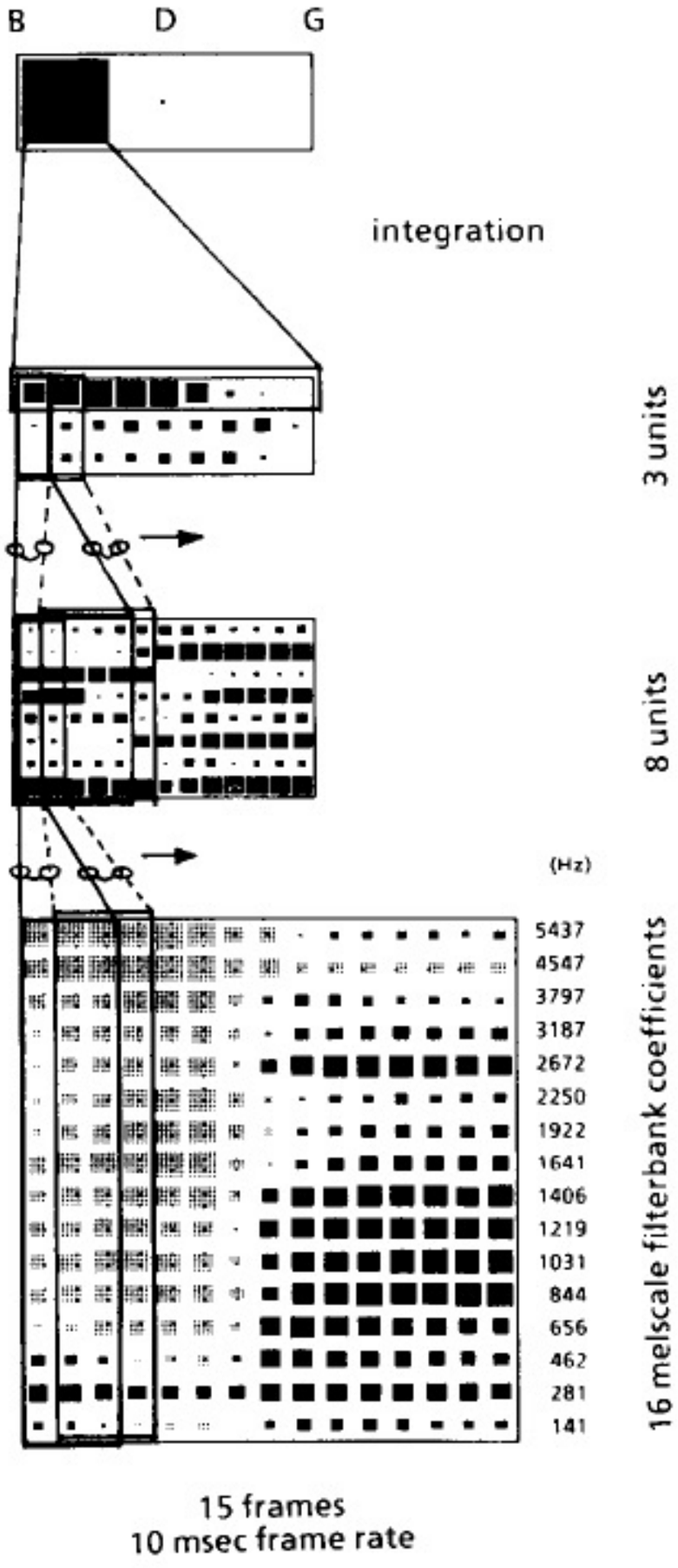


Fig. 2. The architecture of the TDNN.

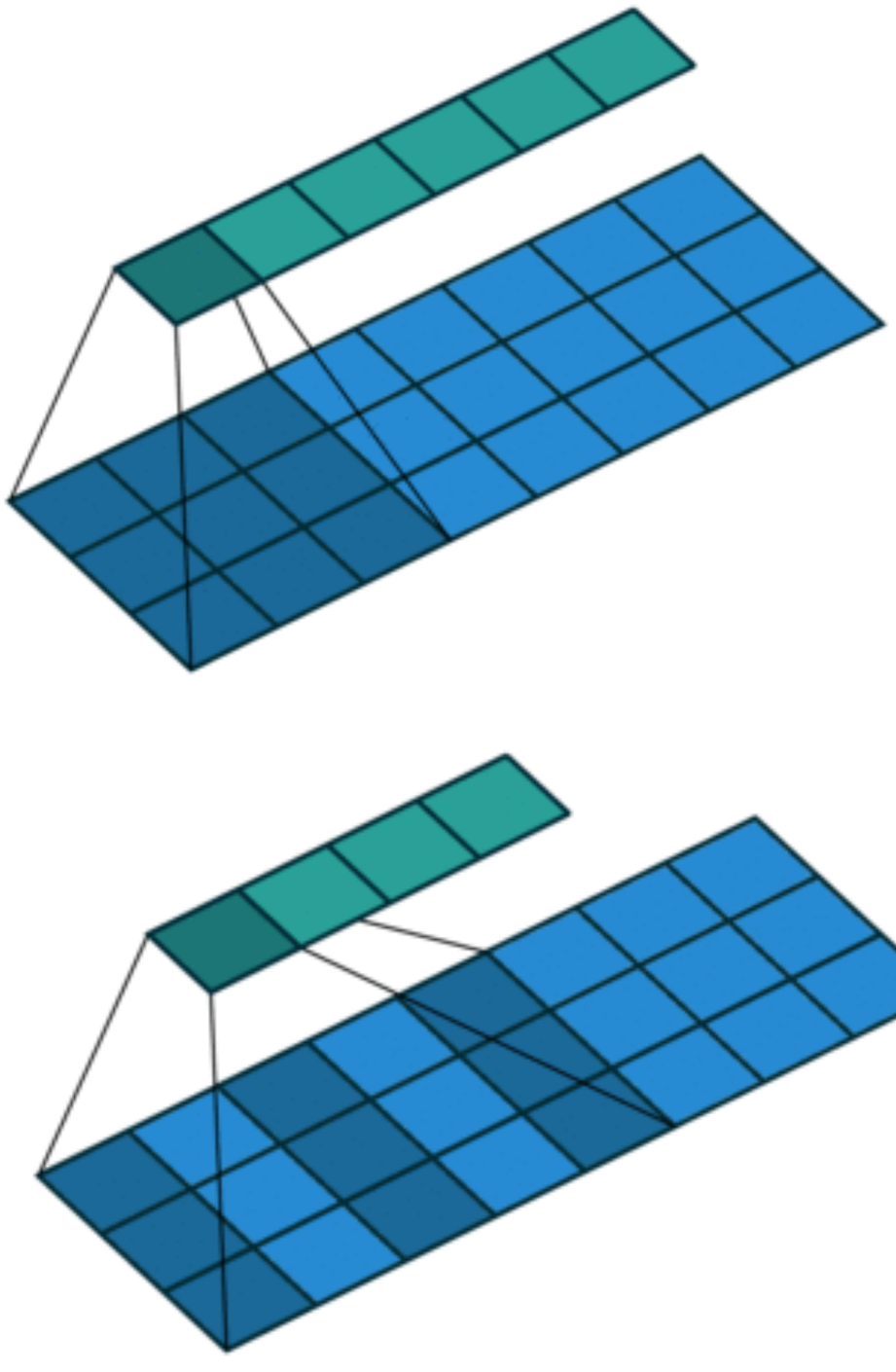
[-2,2]: 5 frames (*t-2*, … *t*, …, *t+2*)

[-1,1]: 3 frames (*t-1*, *t*, *t+1*)

Stride = 1, no dilation (contiguous sampling)

Waibel et al. 1989

SUB-SAMPLED TIME-DELAY NEURAL NETS



https://kaleidoescape.github.io/tdnn/

Peddinti et al. (2015)

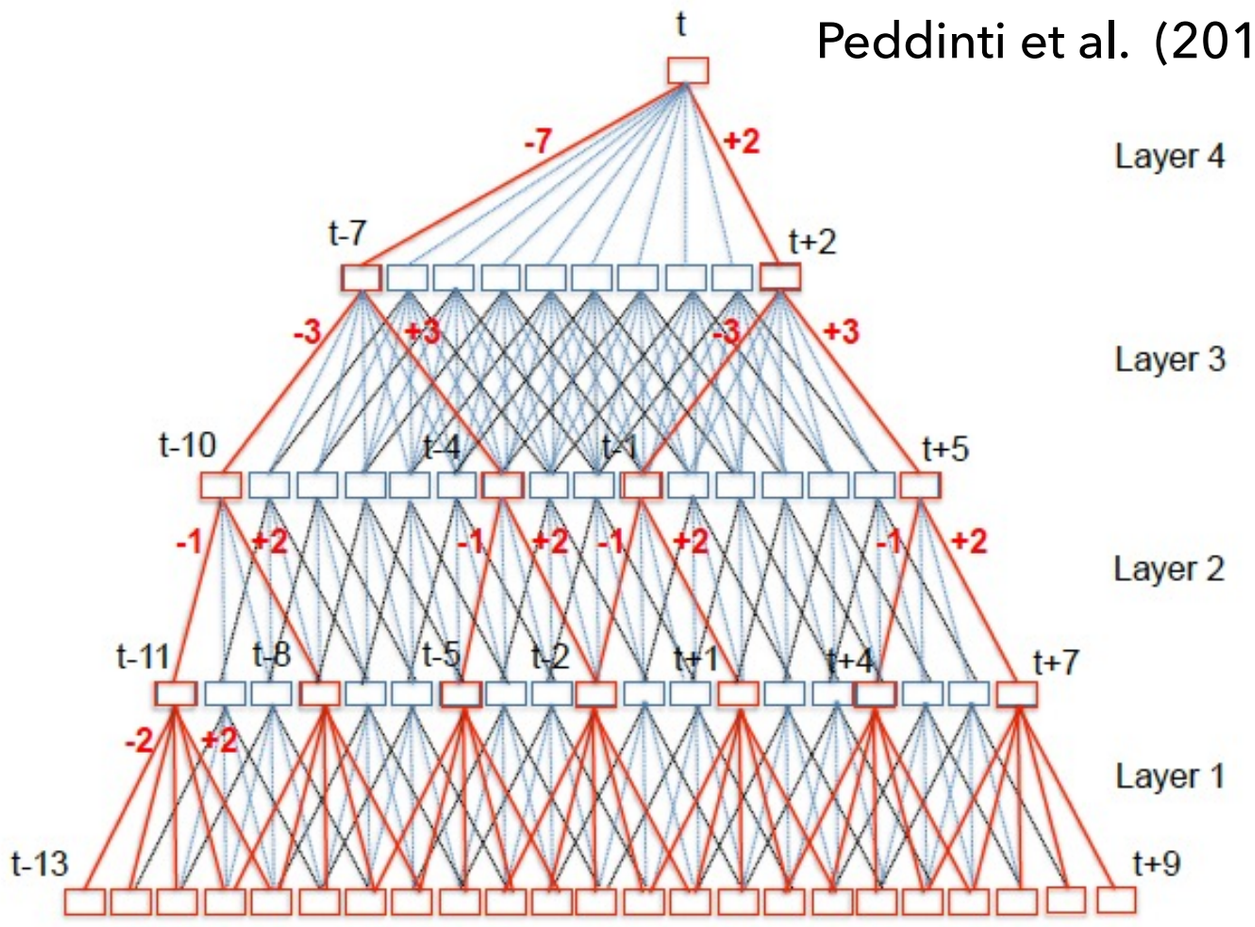
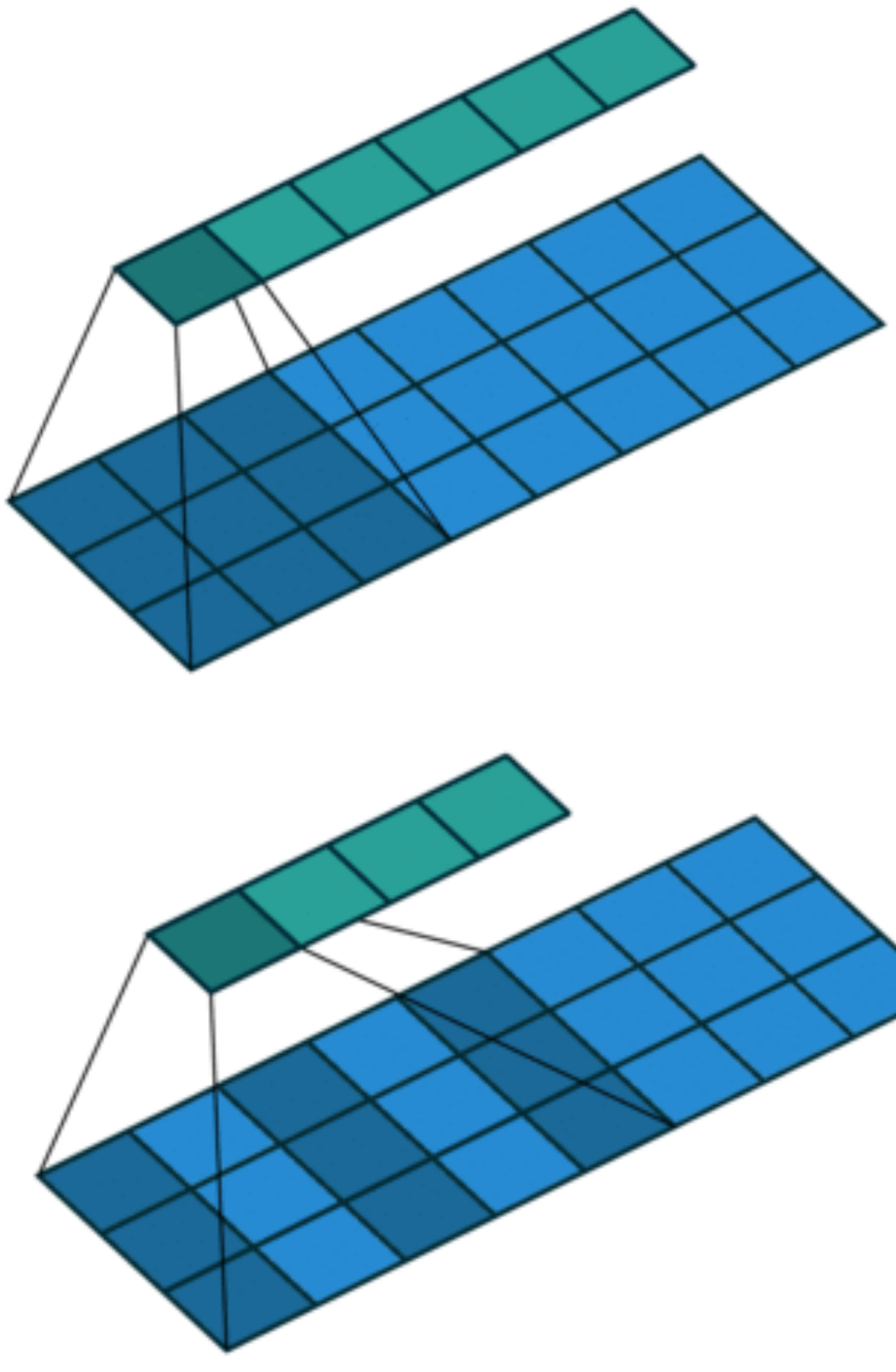


Figure 1: Computation in TDNN with sub-sampling (red) and without sub-sampling (blue+red)

Layer	Input context	Input context with sub-sampling
1	$[-2, +2]$	$[-2, 2]$
2	$[-1, 2]$	$\{-1, 2\}$
3	$[-3, 3]$	$\{-3, 3\}$
4	$[-7, 2]$	$\{-7, 2\}$
5	$\{0\}$	$\{0\}$

SUB-SAMPLED TIME-DELAY NEURAL NETS



https://kaleidoescape.github.io/tdnn/

Peddinti et al. (2015)

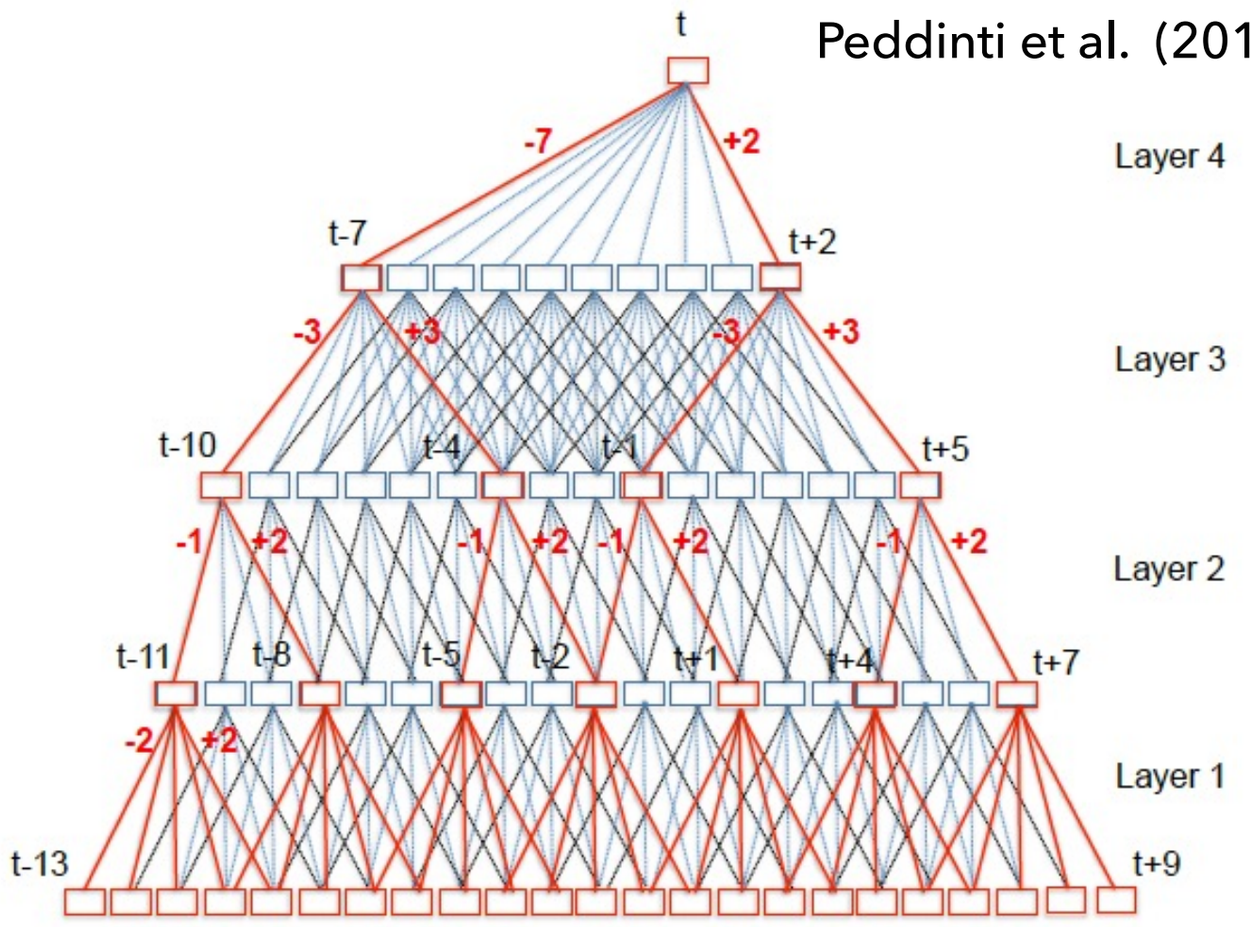
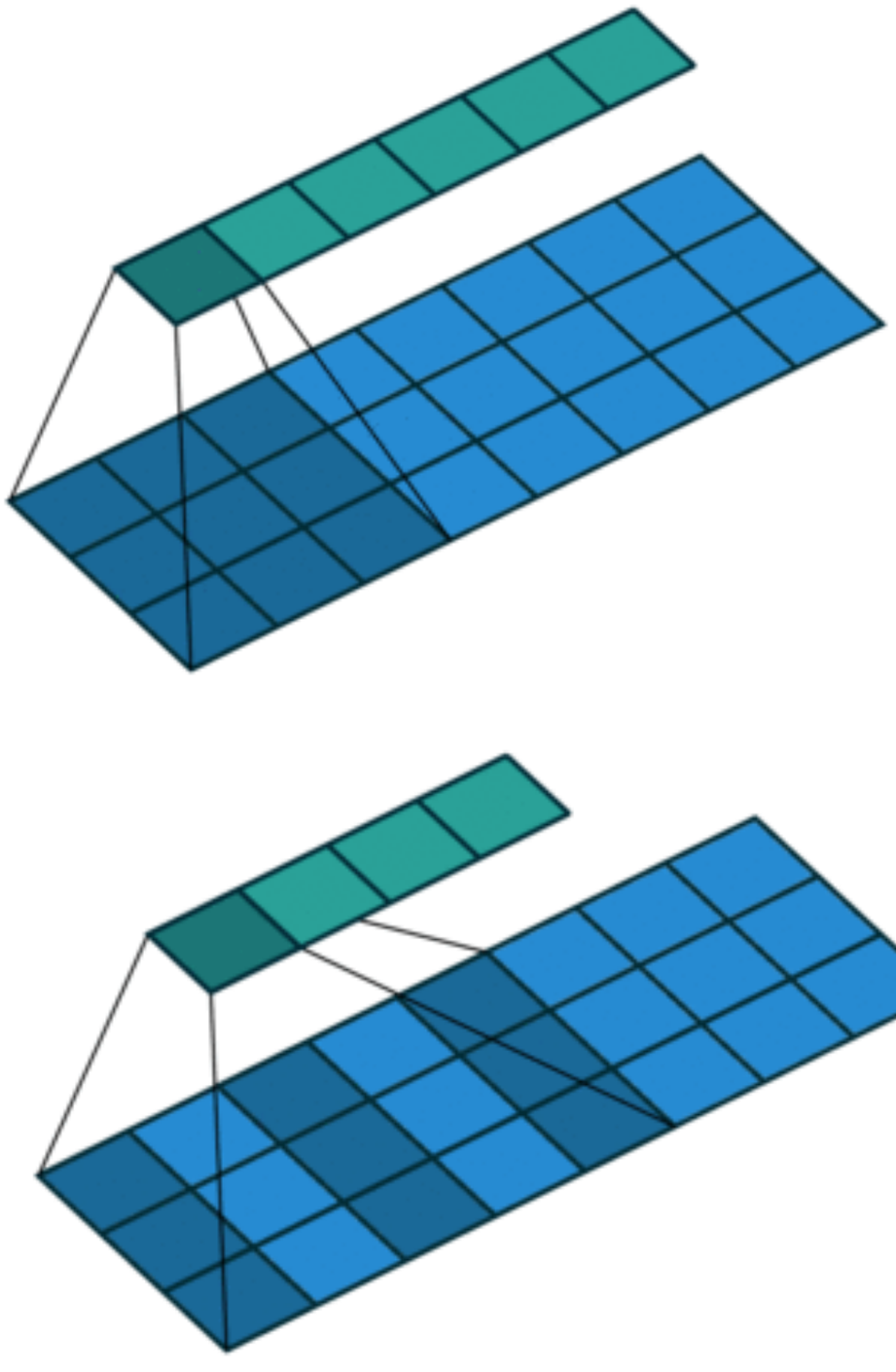


Figure 1: Computation in TDNN with sub-sampling (red) and without sub-sampling (blue+red)

Layer	Input context	Input context with sub-sampling
1	$[-2, +2]$	$[-2, 2]$
2	$[-1, 2]$	$\{-1, 2\}$
3	$[-3, 3]$	$\{-3, 3\}$
4	$[-7, 2]$	$\{-7, 2\}$
5	$\{0\}$	$\{0\}$

SUB-SAMPLED TIME-DELAY NEURAL NETS



https://kaleidoescape.github.io/tdnn/

Peddinti et al. (2015)

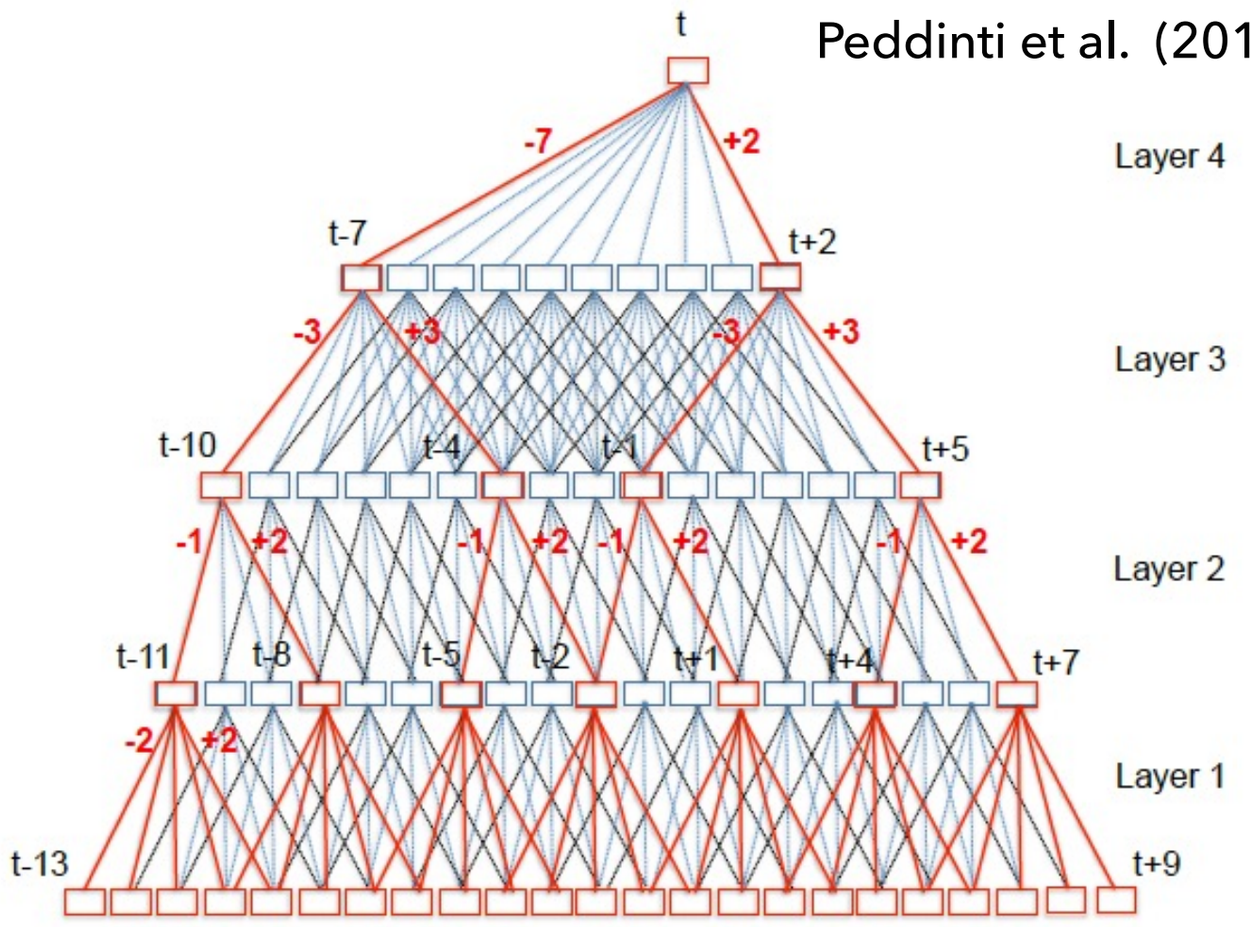


Figure 1: Computation in TDNN with sub-sampling (red) and without sub-sampling (blue+red)

Layer	Input context	Input context with sub-sampling
1	$[-2, +2]$	$[-2, 2]$
2	$[-1, 2]$	$\{-1, 2\}$
3	$[-3, 3]$	$\{-3, 3\}$
4	$[-7, 2]$	$\{-7, 2\}$
5	$\{0\}$	$\{0\}$

"2D" CONVOLUTION BY HAND

https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/convolution#doing-by-hand-1