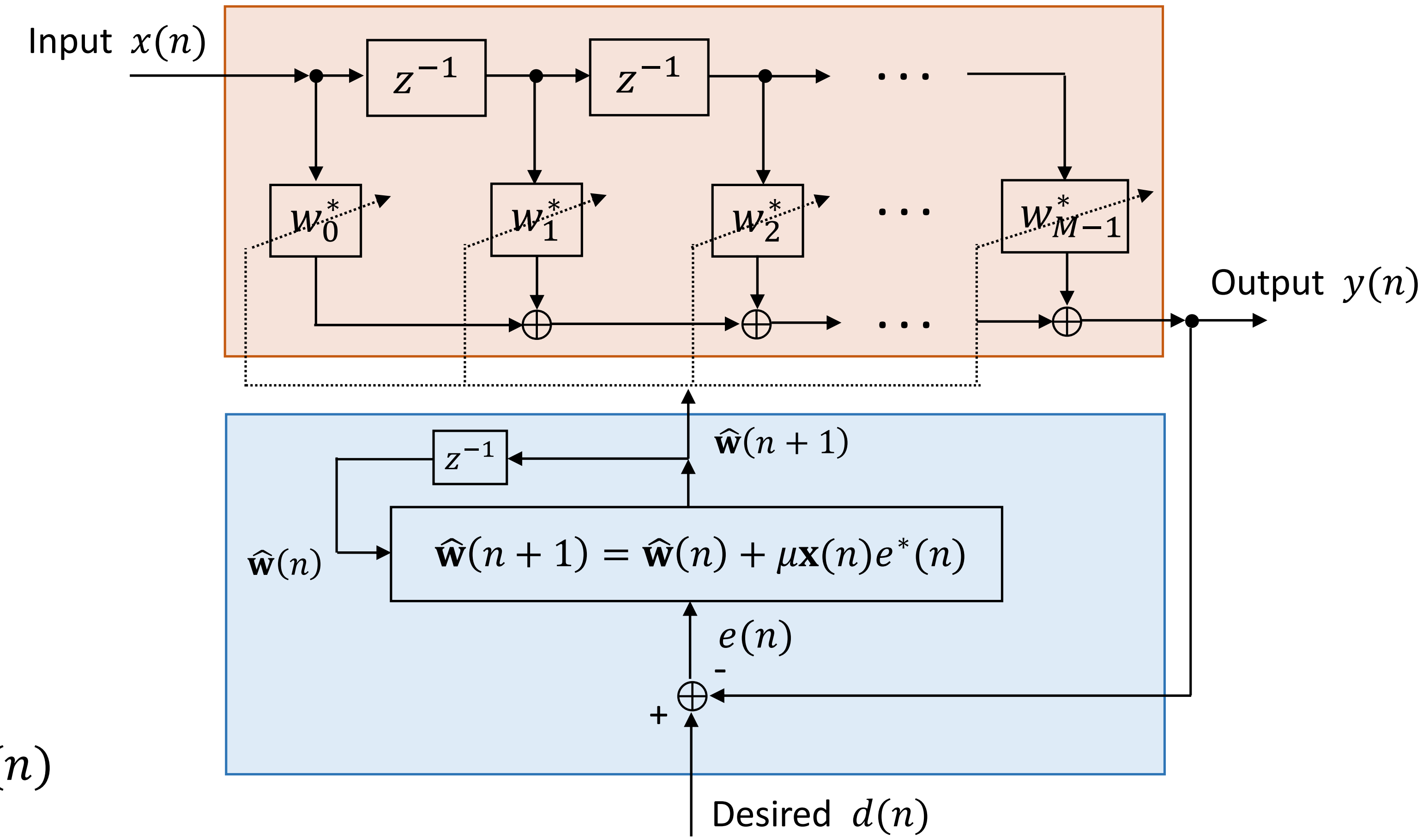
LMS filter

- Architecture diagram



- Update weights for every input $x(n)$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{x}(n)e^*(n)$$

RLS filter (1/2)

- $\boldsymbol{\Phi}(n)\mathbf{w}(n) = \mathbf{z}(n) \;\rightarrow\; \mathbf{w}(n) = \boldsymbol{\Phi}^{-1}(n)\mathbf{z}(n)$

  where $\boldsymbol{\Phi}(n) = \sum_{l=1}^{n} \lambda^{n-l}\,\mathbf{x}(l)\mathbf{x}^H(l) + \delta\cdot\lambda^n\mathbf{I}$, $\mathbf{z}(n) = \sum_{l=1}^{n} \lambda^{n-l}\,\mathbf{x}(l)d^*(l)$

- Recursive Least Squares

$$\boldsymbol{\Phi}(n) = \lambda\boldsymbol{\Phi}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n)$$

$$\mathbf{z}(n) = \lambda\mathbf{z}(n-1) + \mathbf{x}(n)d^*(n)$$

We can apply the recursive relations to avoid matrix inversion.

RLS filter (2/2)

- Let $\mathbf{P}(n) = \mathbf{\Phi}^{-1}(n)$
- $\delta > 0$ and $0 < \lambda \leq 1$
- Algorithm

1: $\hat{\mathbf{w}}(0) = \mathbf{0}$

2: $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$

3: for $n = 1,2,3, \ldots$ do

4: $\quad \mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^H\mathbf{P}(n-1)\mathbf{x}(n)}$

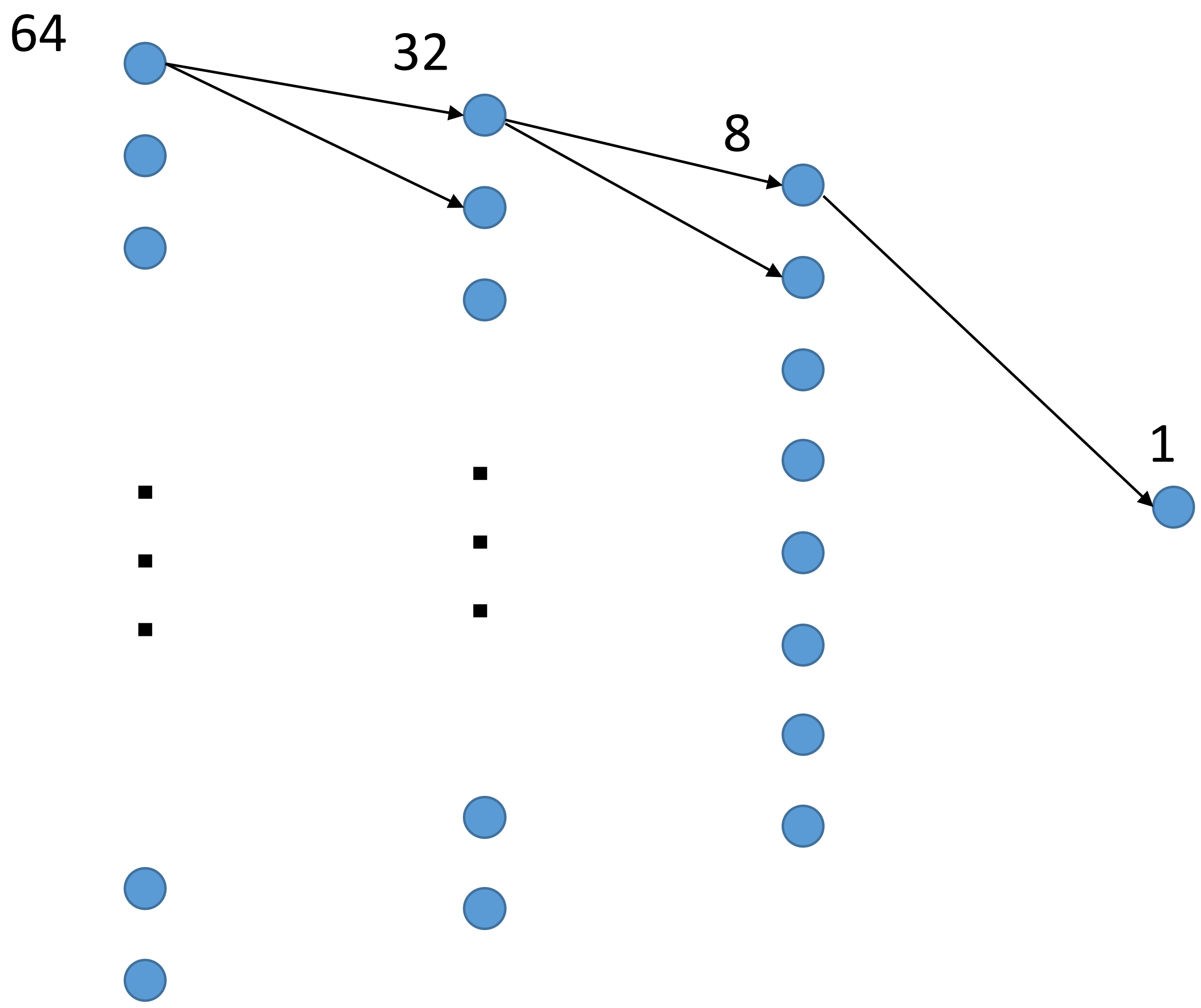
5: $\quad \xi(n) = d(n) - \hat{\mathbf{w}}^H(n-1)x(n)$

6: $\quad \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n)$

7: $\quad \mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1)$

8: end for

DNN



Optimizer:

Adam, learning rate=0.001

Loss function:

mse

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_{true}^{(i)} - y_{pred}^{(i)}\right)^2$$

Metrics:

mae

$$\frac{1}{n}\sum_{i=1}^{n}\left|y_{true}^{(i)} - y_{pred}^{(i)}\right|$$