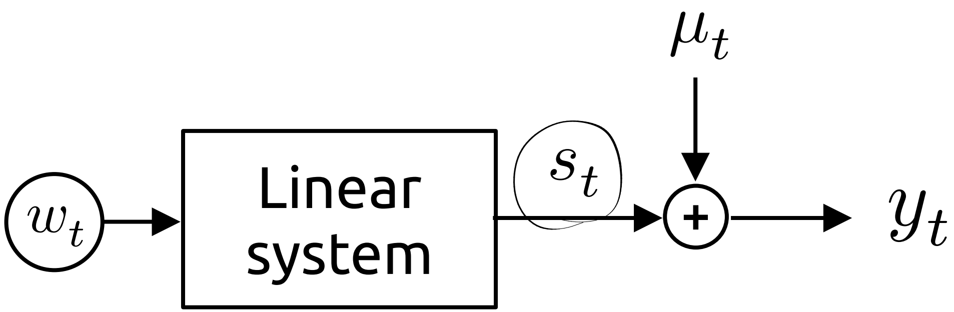
Statistics and Data Science for Engineers
E178 / ME276DS

Time series analysis
Part 2

Recall



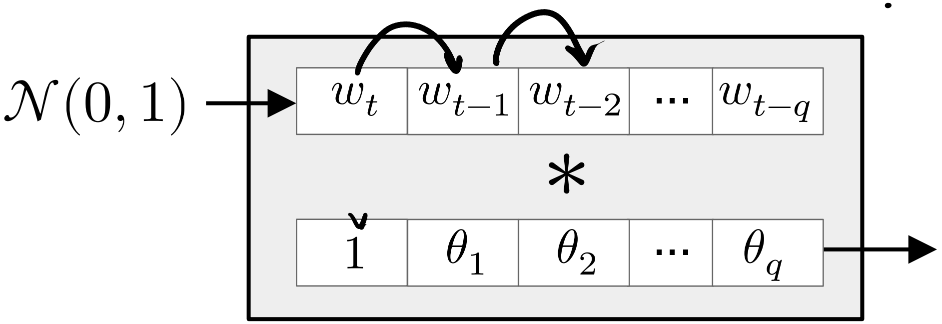
- μ_t ... deterministic signal
- s_t ... zero-mean stochastic signal

Stationary linear processes

Moving average

q memory slots.

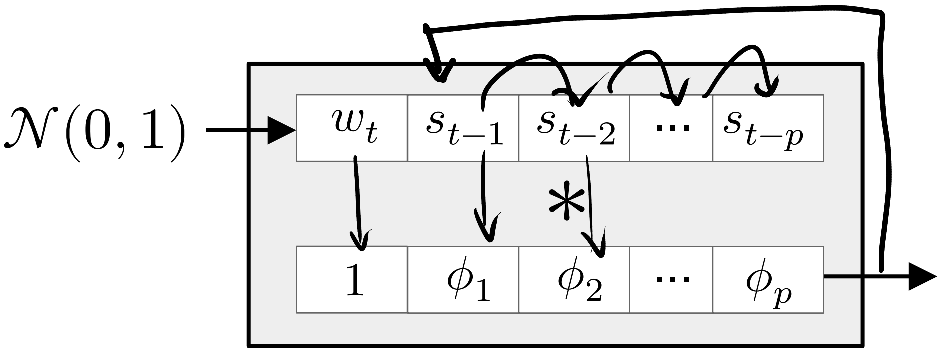
MA(q)



$$s_t = w_t + \sum_{j=1}^{q} \theta_j w_{t-j}$$

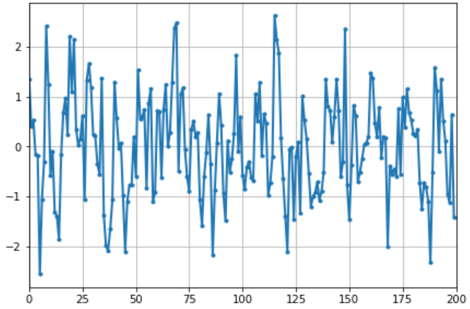
auto regressive

AR(p)

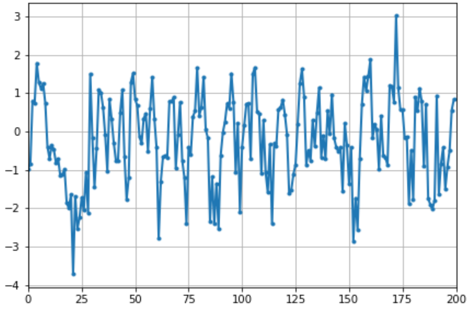


$$s_t = w_t + \sum_{j=1}^{p} \phi_j s_{t-j}$$

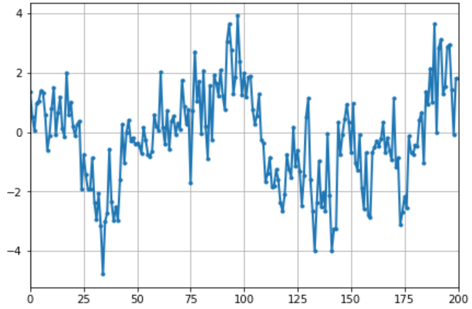
MA(2)



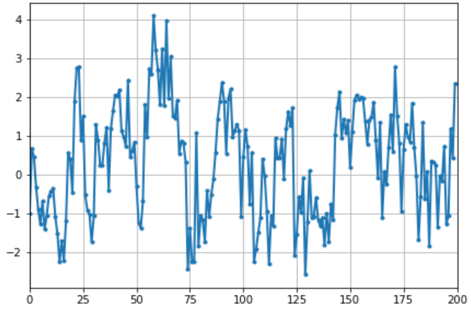
MA(5)



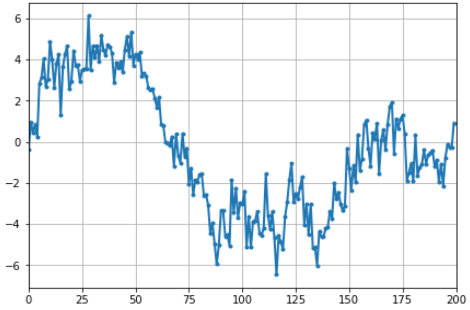
MA(10)



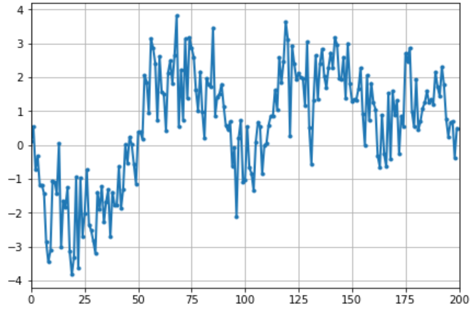
AR(2)



AR(5)



AR(10)

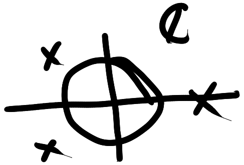


Polynomial representation

Backward shift operator B: $Bs_t = s_{t-1}$

MA(q)	AR(p)
$s_t = w_t + \sum_{j=1}^{q} \theta_j w_{t-j}$	$s_t = w_t + \sum_{j=1}^{p} \phi_j s_{t-j}$
$= w_t + \sum_{j=1}^{q} \theta_j B^j w_t$	$s_t = w_t + \sum_{j=1}^{p} \phi_j B^j s_t$
$= \left(1 + \sum_{j=1}^{q} \theta_j B^j\right) w_t$	$\left(1 - \sum_{j=1}^{p} \phi_j B^j\right) s_t = w_t$
$= \theta(B)\, w_t$	$\phi(B)\, s_t = w_t$

$$\cancel{s_t} = \theta(B)\cdot\phi(B)\cdot\cancel{s_t}$$



	MA(q)	AR(p)
Polynomial	$s_t = \underbrace{\theta(B)} \, w_t$	$\underbrace{\phi(B)} \, s_t = w_t$
Relationship	$\boxed{\theta(B)\phi(B) = 1}$	
Stationary	Always	roots$(\phi(B))$ have magnitude > 1
Causal	roots$(\theta(B))$ have magnitude > 1	Always

ARMA(p, q)

$$s_t = w_t + \underbrace{\sum_{j=1}^{q} \theta_j w_{t-j}}_{\text{MA}} + \underbrace{\sum_{j=1}^{p} \phi_j s_{t-j}}_{\text{AR.}}$$

$$\boxed{s_t = \frac{\theta(B)}{\phi(B)} w_t}$$

- Hyper-parameters: q and p. ... orders of $\theta(B)$, $\phi(B)$.
- Tunable parameters: $\underbrace{\theta_j\text{'s and } \phi_j\text{'s}}$.

... complicated.

$\boxed{\text{statsmodels t.s.}}$

Big picture

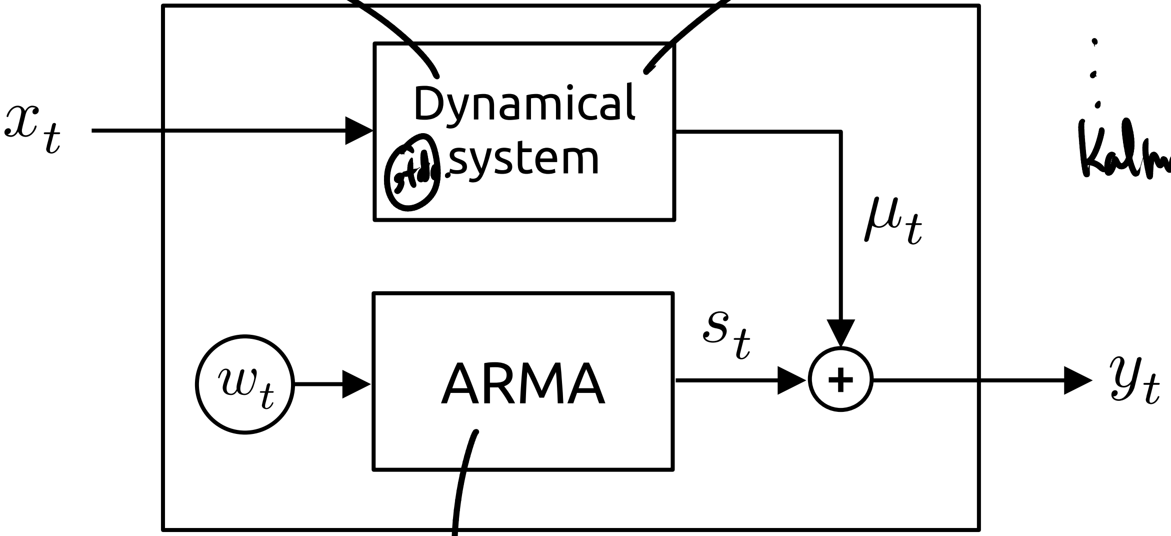
memory.

$\dot{x} = Ax + Bu$

$\mu_t = Cx$

$\vdots$

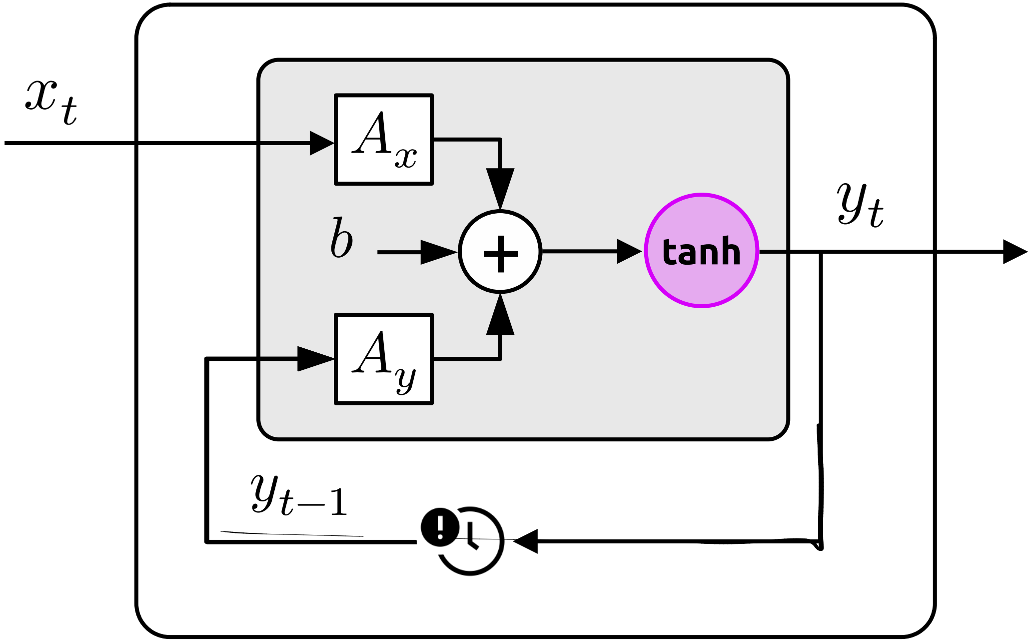
Kalman filters

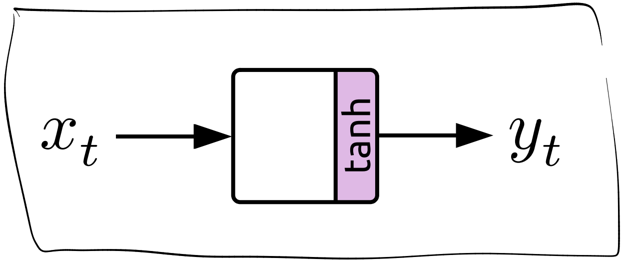


memory. (feedback).

Recurrent neural networks (RNNs)

- simple RNN
- LSTM.





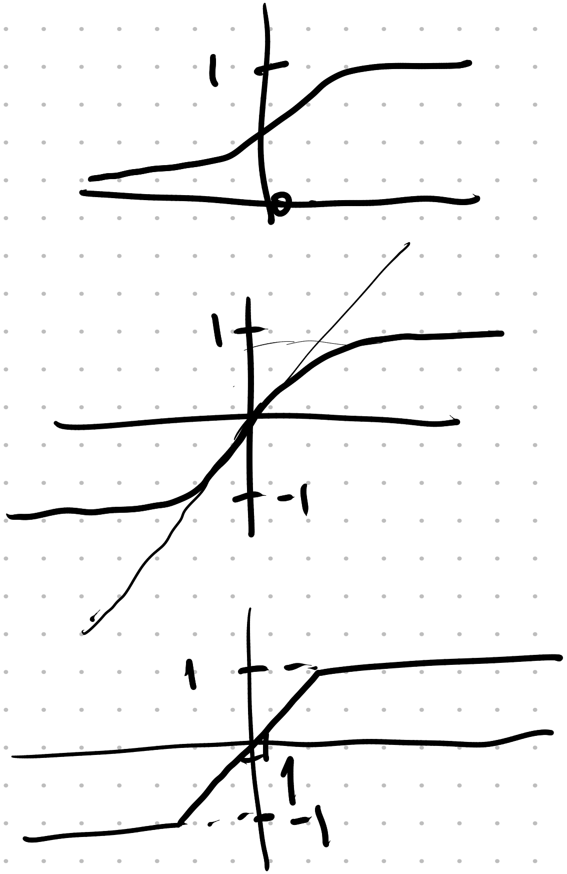
Recurrent = feedback.

$$y_t = \tanh\left(A_y\, y_{t-1} + A_x\, x_t + b\right).$$

Activation functions:

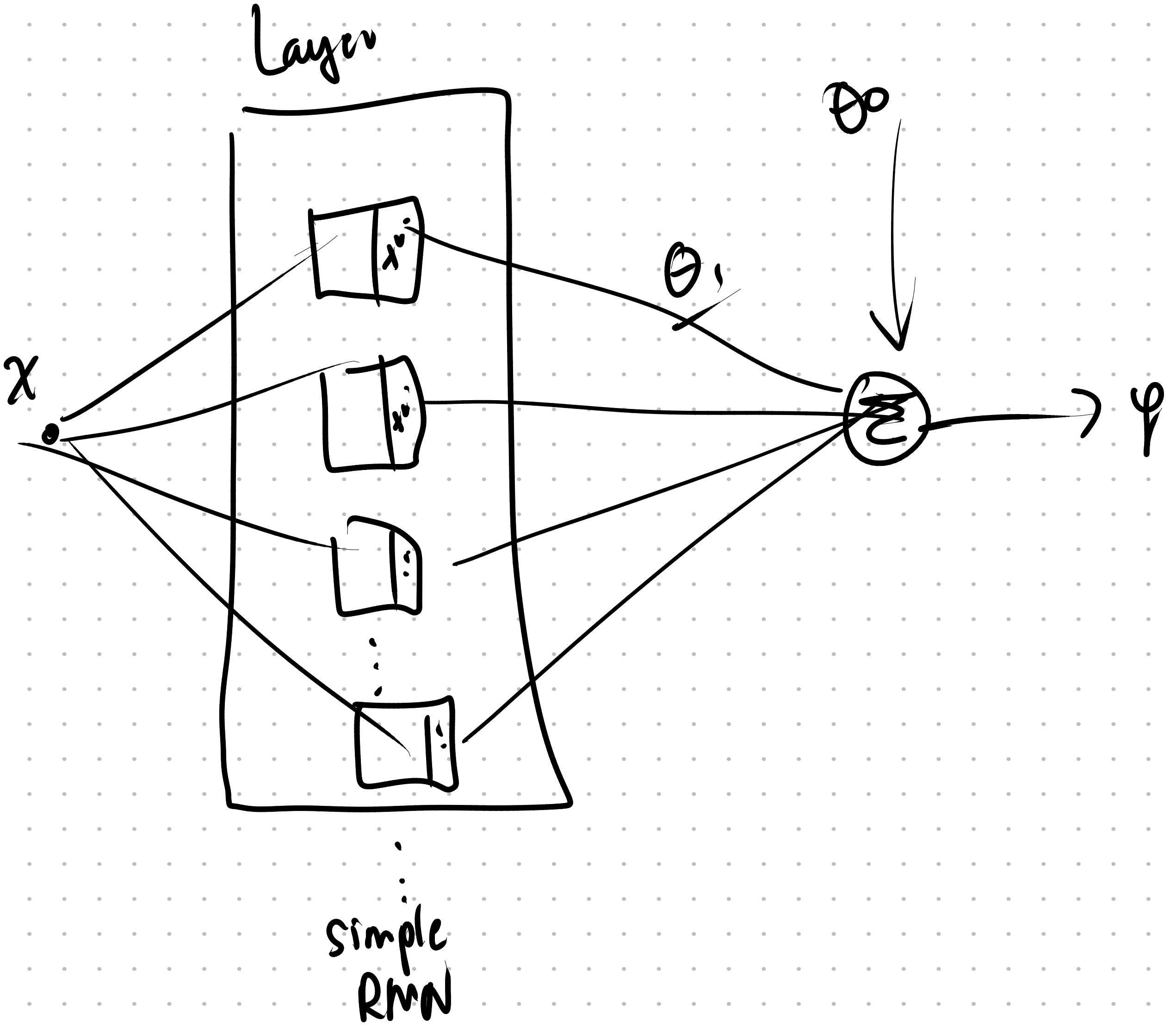
Sigmoid

Tanh



... smooth thresholdin. (saturation)

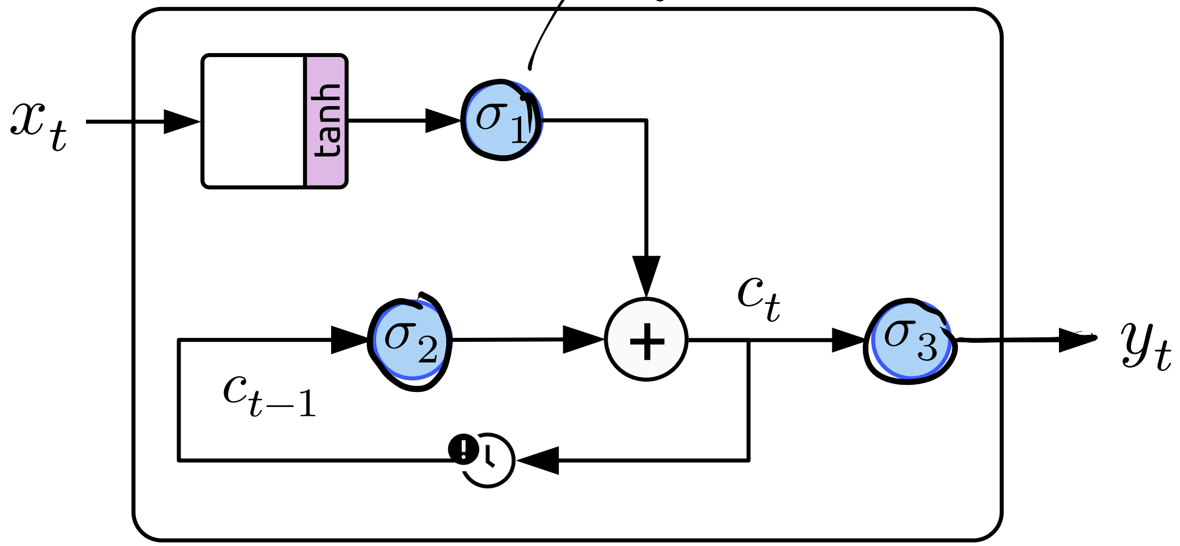
hard saturation.



Layer
θ_0
θ_1
x
Σ
φ
simple RNN

Long-short term memory cell (LSTM)

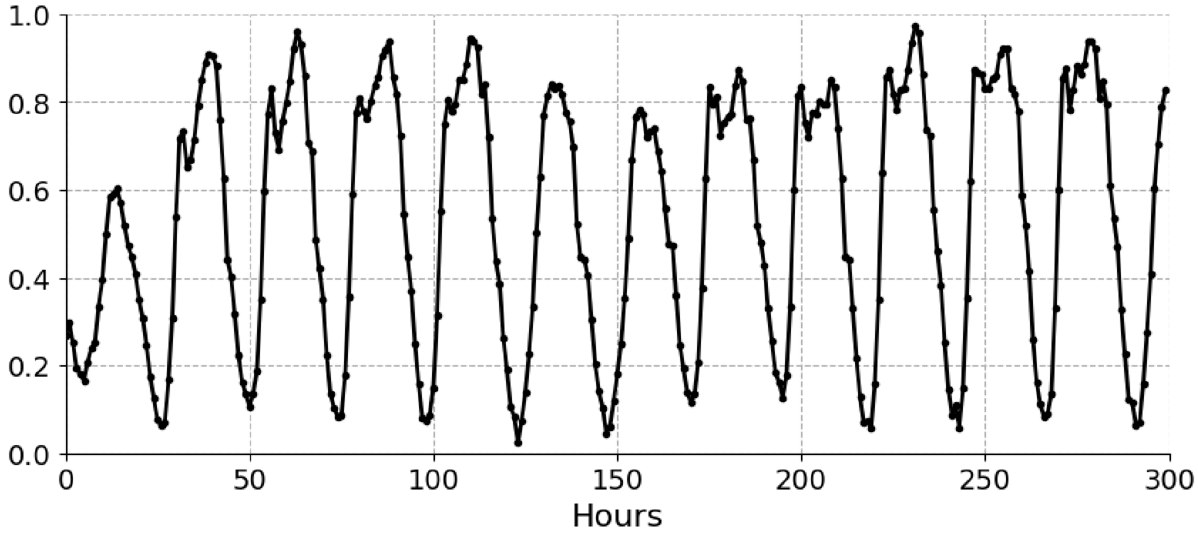
regulates the inputs.



- Each σ_i is a *sigmoid gate*: $\sigma_i = \sigma(A_{x,i} x_t + A_{y,i}\, y_{t-1} + b_i)$
- c_t is a memory vector.

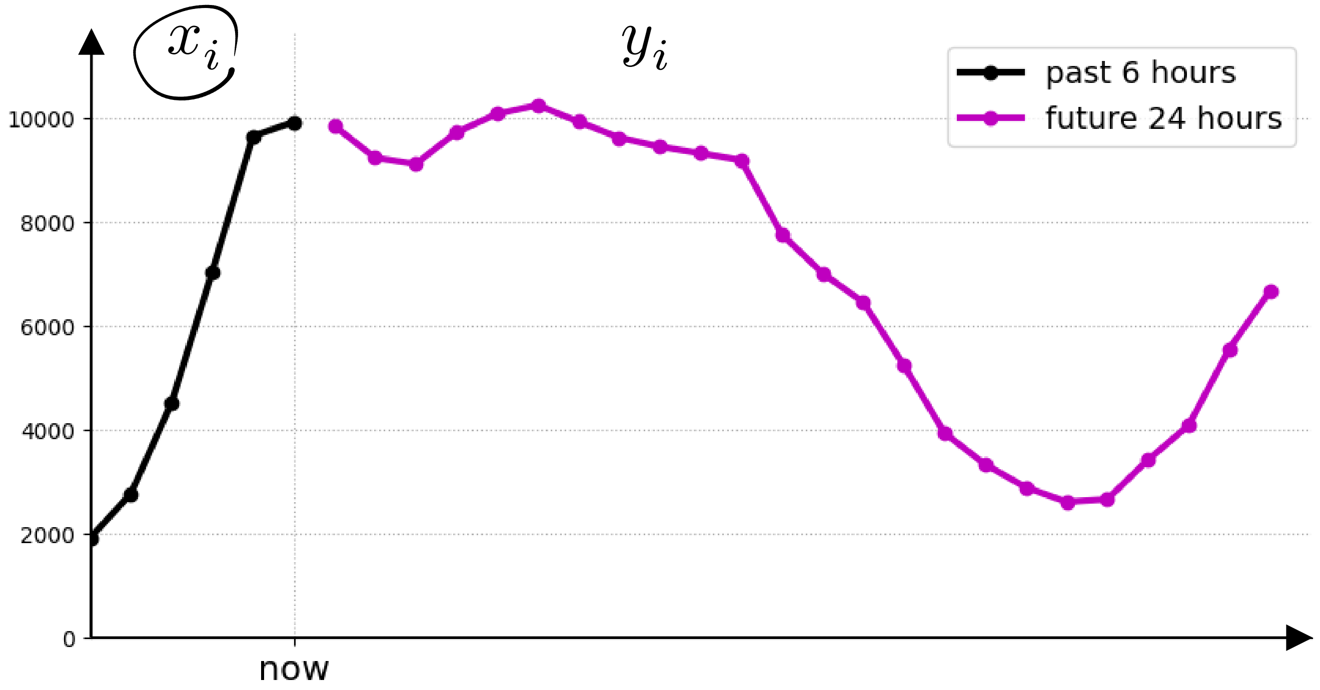
sigmoid.

Example: Freeway traffic flow prediction



- y_t is hourly flows on a freeway.
- Problem: Predict the next 24 hours from the previous 6 hours.

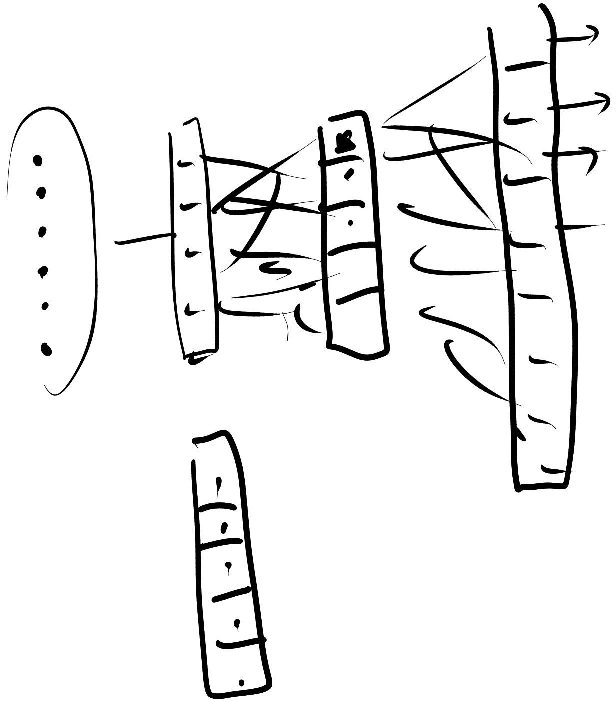
Training data $\mathcal{D}_{\text{train}} = \underline{\{(x_i, y_i)\}_N}$



Perceptron

```
model_dense = Sequential([
    Flatten(),
    Dense(5, activation="relu"),
    Dense(24)
])
```

forecast 24 h..



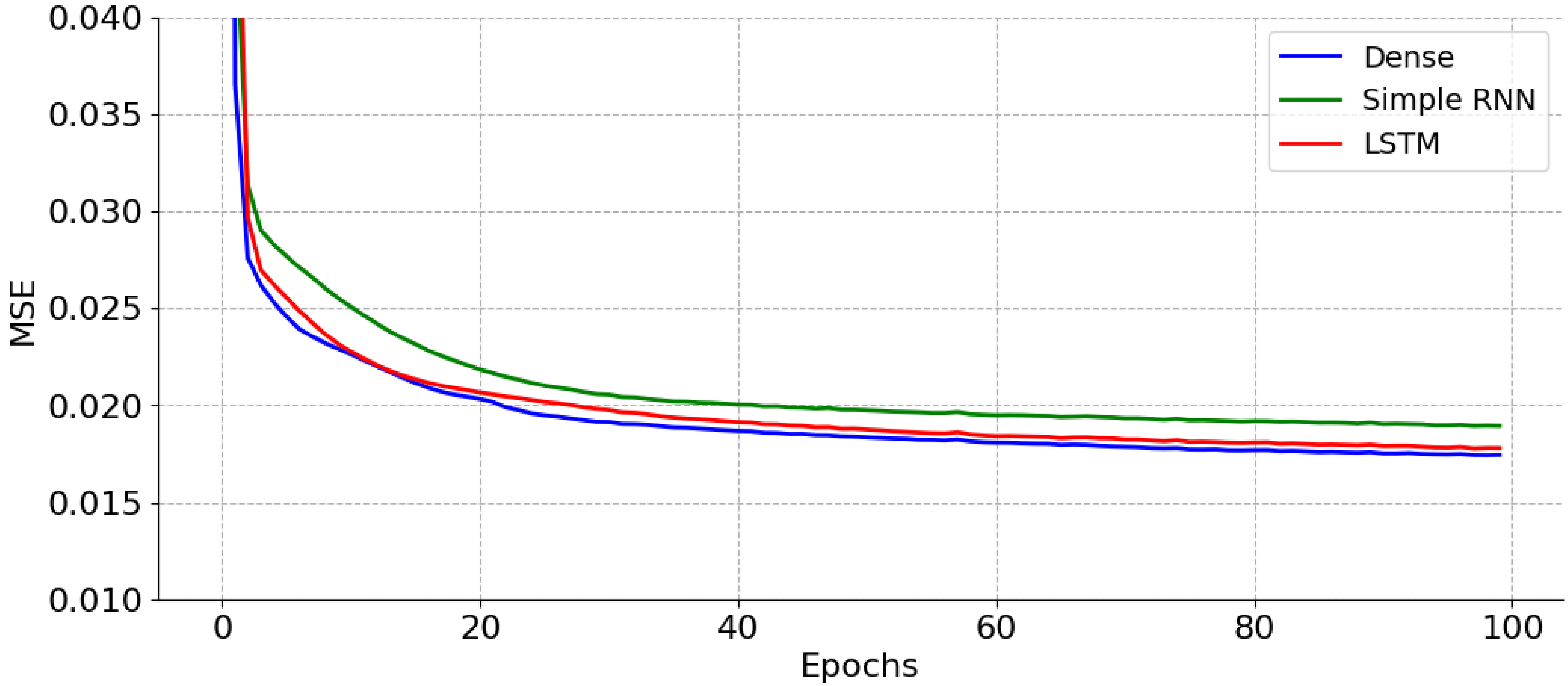
Simple RNN

```
model_simprnn = Sequential([
    SimpleRNN(5,input_shape=(1,6)),
    Dense(24)
])
```

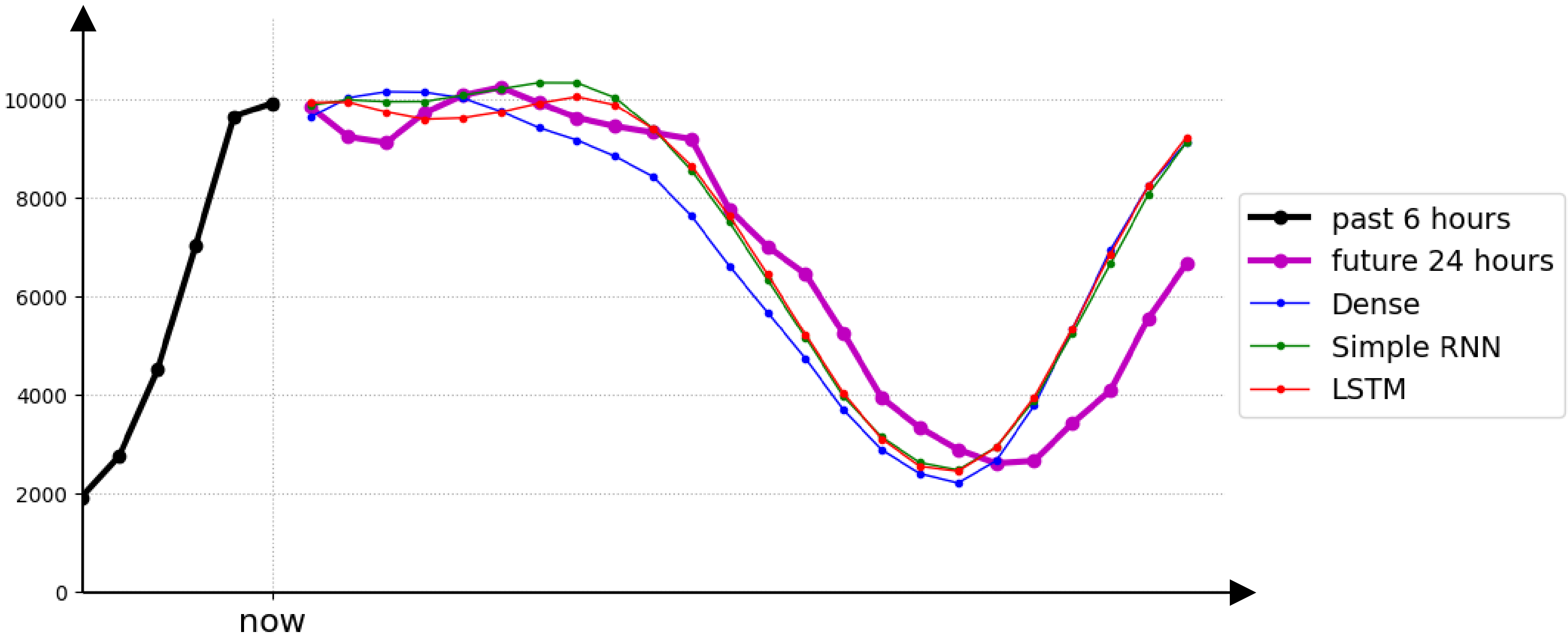
LSTM

```
model_lstm = Sequential([
    LSTM(5, input_shape=(1, 6)),
    Dense(24)
])
```

Training MSE:



Prediction:



MAE on test data :

Dense: 21.7%

Simple RNN: 19.0%

LSTM: 17.7%