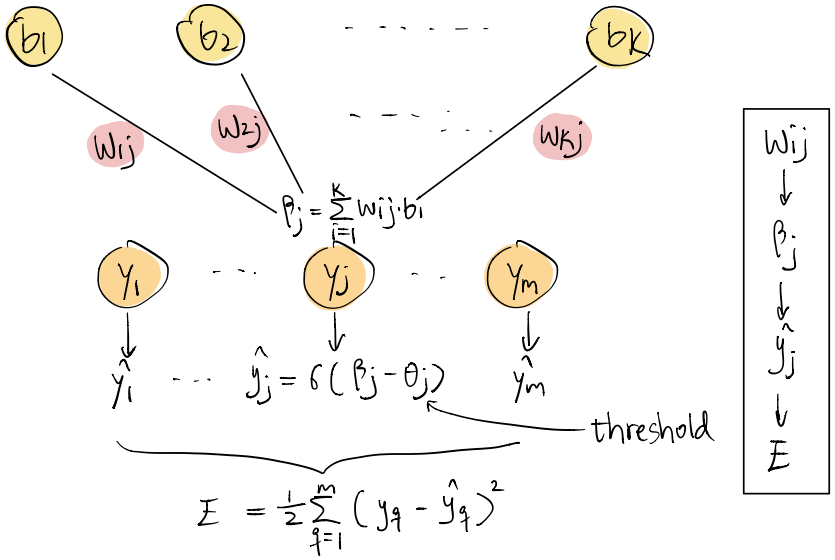
Neural Networks

Index

- BP
- Perceptron
- Feed Forward
- RNN , LSTM
- CNN
- GAN

BackPropagation

On an input data point :



gradient descent :

update rule : $w \leftarrow w + \Delta w$ for any parameter w

$\Delta w = -\eta \frac{\partial E}{\partial w}$, where η is the learning rate

calculate gradient using chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{ij}}$$

$$= -(y_j - \hat{y}_j) \cdot \sigma'(\beta_j - \theta_j) \cdot b_i$$

when the activation is sigmoid, its derivative is:

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

$$= -(y_j - \hat{y}_j)\hat{y}_j(1-\hat{y}_j) \cdot b_i$$

Objective of BP:

minimize cumulative error on training set:

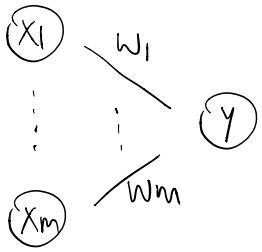
$$E = \frac{1}{m}\sum_{k=1}^{m} E_k$$

Methods to avoid overfitting:

1. Early Stopping: if training accuracy ↑ but validation accuracy ↓, then stop early.
2. Regularization: 类似 SRM, 在 objective 中增加描述网络复杂度的项.

Perceptron

Simplest NN, no hidden layer

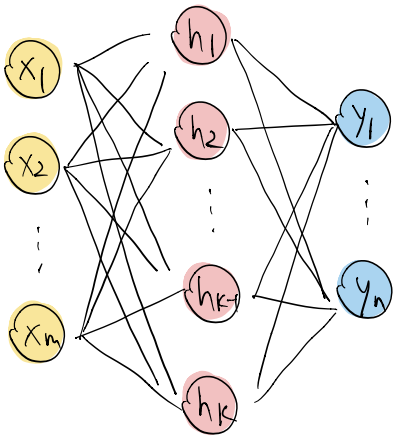


$$y = f(w^T x - \theta)$$

Cons: only converge for linear seperable problems

Feed Forward NN / Multi-layer FF NN

Fully connected neural network, with one or more hidden layers.

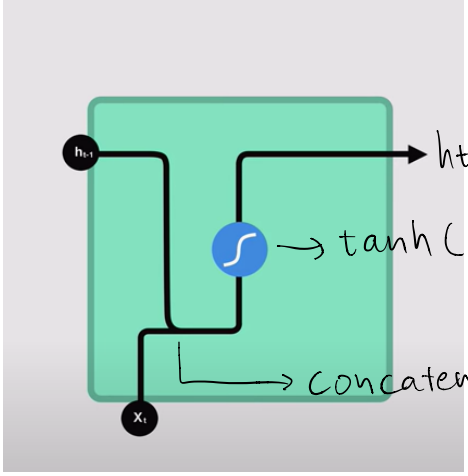


Cons:

- can't process sequential data
- no memory $\Rightarrow$ RNN

RNN

At time t, $b_i(t) = f(b_i(t-1), w^T x - \theta)$



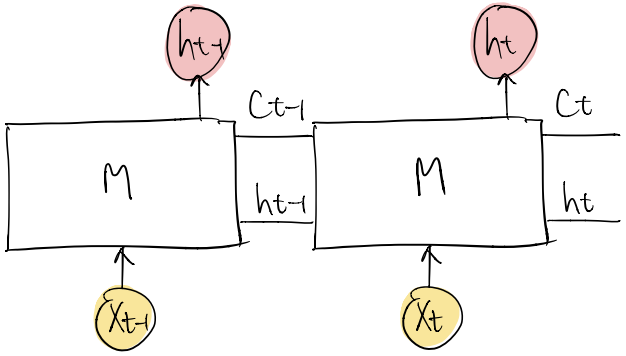
Cons:

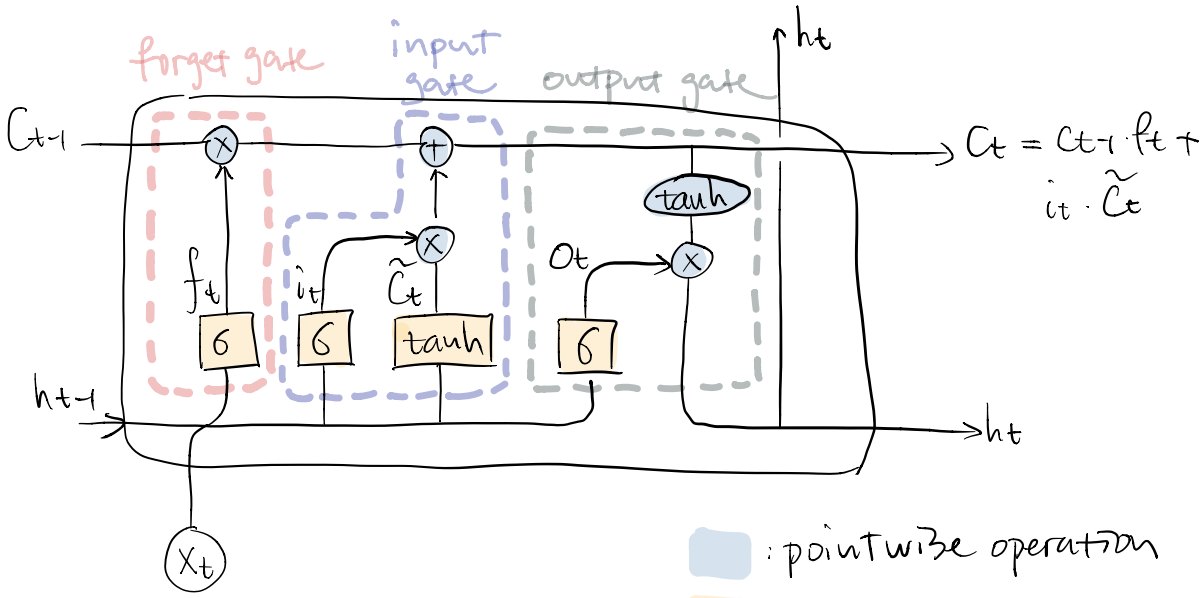
- vanishing gradient & exploding gradient

  ⇒ LSTM

LSTM

A type of RNN that solves the gradient problem by selecting which to remember from the past

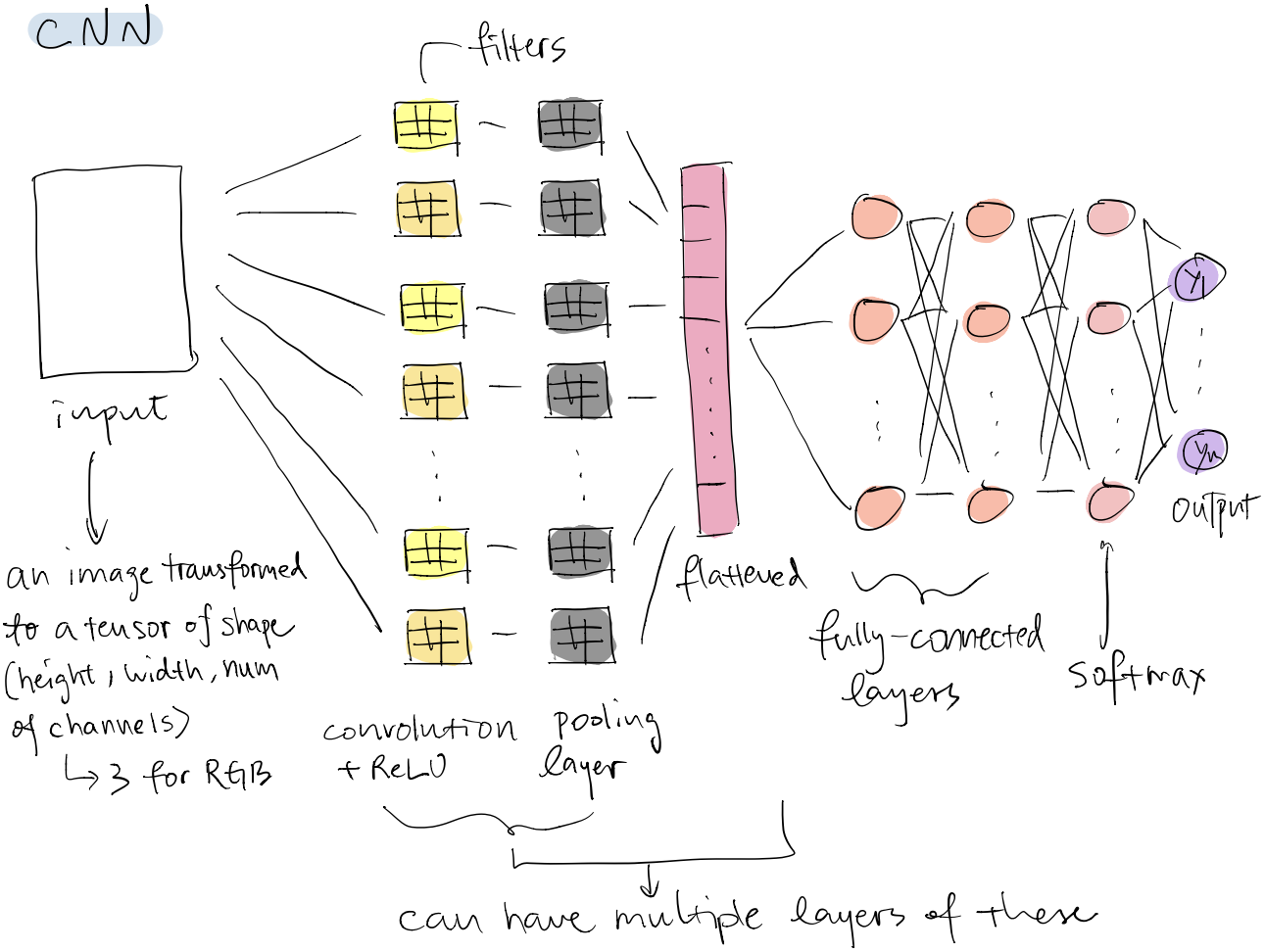




Module components:

1. Forget gate : decide which info to forget from the previous cell state
2. Input gate : decide which new info to store into the current cell state
3. Output gate : decide what to output

CNN



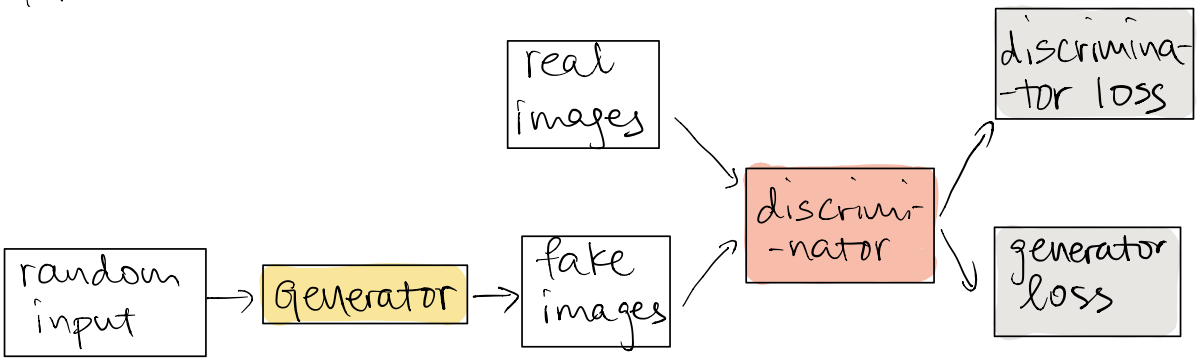
Pros: - sparse connection
- shared neuron

Padding

Stride: shift the filter window by how many units a time

Dropout: randomly remove a proportion of units from the network

GAN



for 1, 2, ···, T:

① discriminator trains for 1~m epochs while generator weights remain constant.

② generator trains for 1~n epochs while discriminator weights remain constant:

1. sample random noise
2. produce fake images
3. let discriminator classify
4. calculate generator loss
5. BP