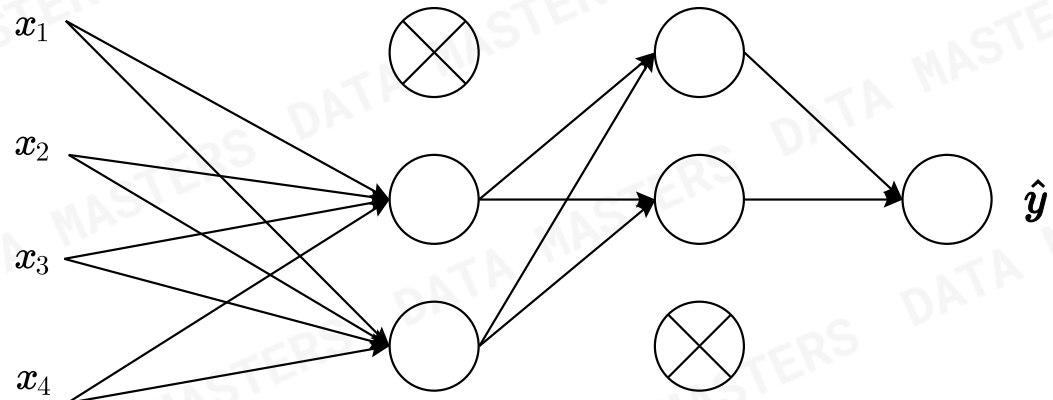
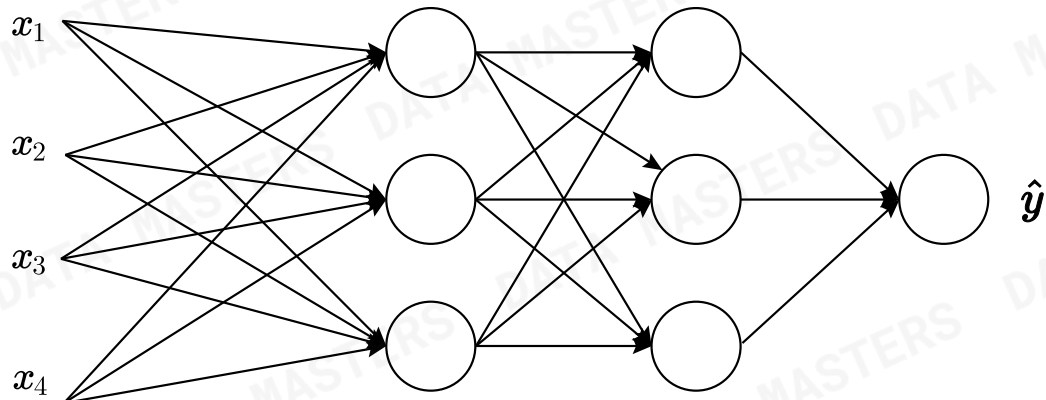


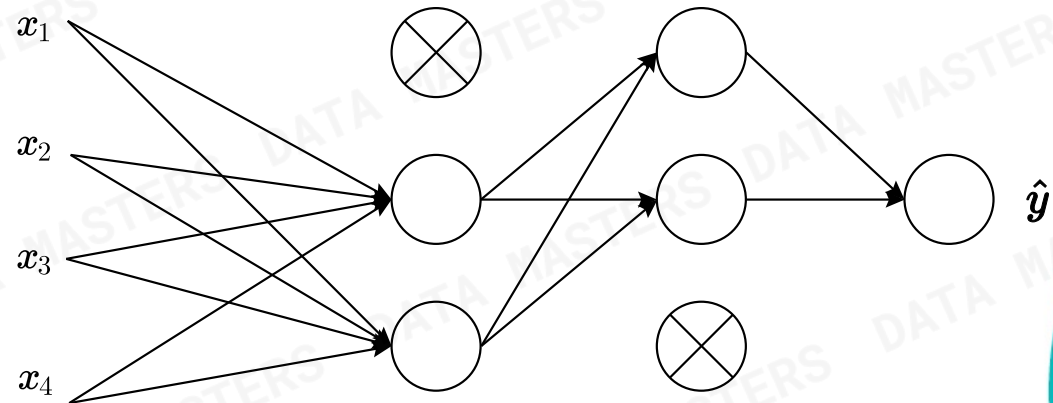
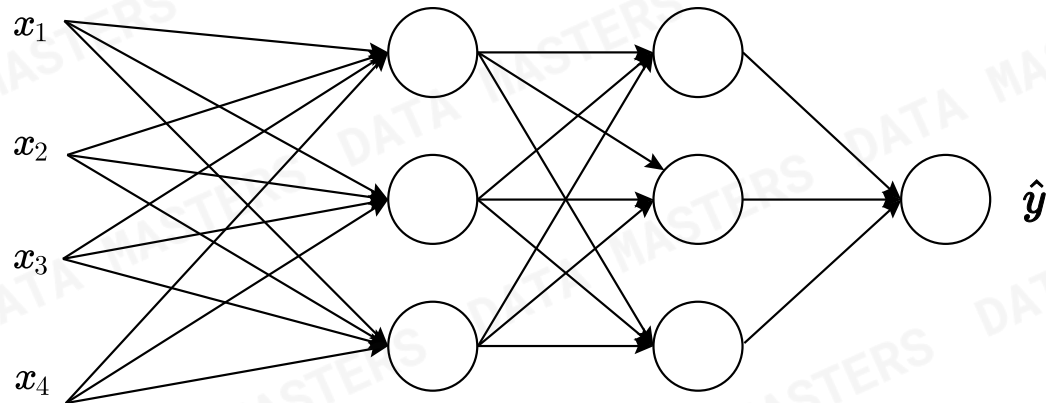
# Dropout

- Consiste in una eliminazione randomica di neuroni da una rete neurale
- È una forma di regolarizzazione -> Previene overfitting
- Riduce l'interdipendenza nell'addestramento dei neuroni



# Dropout

- Può essere applicato a tutti layer nascosti e al layer di input (*naturalmente non sul layer di output*)
- Viene fatta backpropagation sulla rete *diminished*



# Dropout

- Se consideriamo un singolo neurone, un certo numero di unità di input devono generare un input con un valore sensato
- Se eliminiamo uno degli input, il neurone di output non può dipendere troppo da quell'input:
  - I valori dei pesi verranno quindi distribuiti in maniera più equa negli altri input (simile alla regolarizzazione L2)
  - Per questo il dropout è una forma di regolarizzazione

# Dropout



Fase di  
**Training**

Per ogni livello nascosto, per ogni sample, ad ogni iterazione, si ignora il nodo con una probabilità pari a  $p$

- Usa tutte le funzioni di attivazione, riducendole di un fattore  $p$
- Non usiamo il dropout qui perché durante le predizioni in fase di test risultati randomici **non sono desiderabili**
- Il dropout a test-time causerebbe delle predizioni con un rumore più accentuato



Fase di  
**Testing**

# Dropout

- Rende la rete capace di addestrare feature meno interdipendenti dalle altre
- Utilizza dei subset sempre diversi di livelli per l'addestramento (*simula modelli di rete diversi fra loro*)
- Se  $H$  è il numero di unità nascoste, abbiamo  $2^H$  modelli possibili di rete neurale in fase di train. In fase di test sono considerate tutte le unità ma ridotte di un fattore  $p$




**Vantaggi**



# Dropout

**Raddoppia** (circa) il numero di epoche necessarie,  
pur diminuendo la durata dell'addestramento



**Svantaggi**