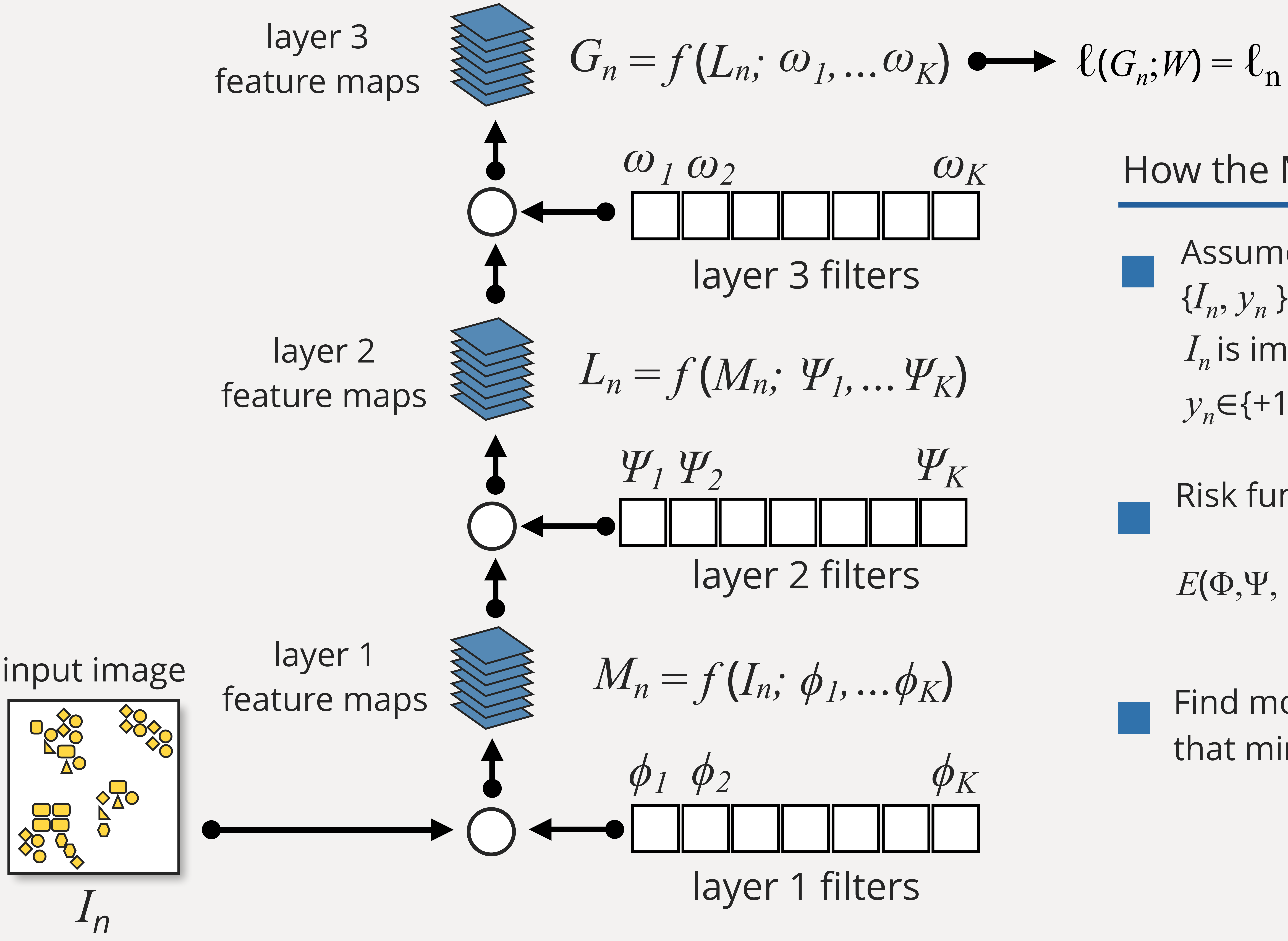
Training the Network



How the Model Learns

- Assume we have labeled images
  $\{I_n, y_n\}\ n = 1, N$
  I_n is image n
  $y_n \in \{+1, -1\}$ is associated label

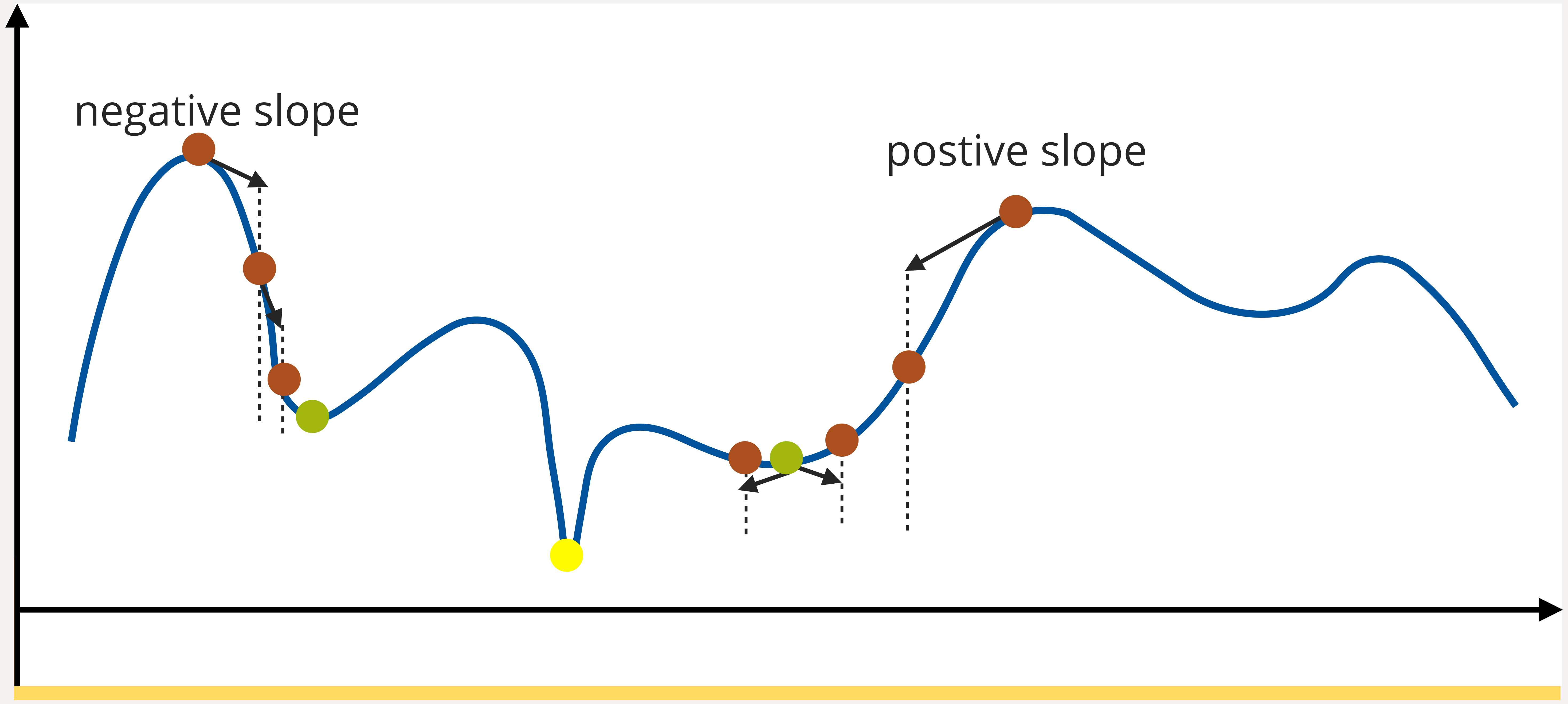
- Risk function of model parameters

$$E(\Phi, \Psi, \Omega, W) = 1/N \sum_{n=1}^{N} loss(y_n, \ell_n)$$

- Find model parameters $\hat{\Phi}, \hat{\Psi}, \hat{\Omega}, \hat{W}$ that minimize $E(\Phi, \Psi, \Omega, W)$

Gradient Descent

$$\Theta = \{\ \Phi, \Psi,\ \Omega, W\}$$



$$\Theta_{t+1} = \Theta_t - \alpha \underbrace{\nabla_{\Theta} E(\Theta_t)}_{\text{multi-dimensional "slope"}}$$

Gradient Descent → Stochastic Gradient Descent

$$\Theta_{t+1} = \Theta_t - \alpha \nabla_\Theta E(\Theta_t)$$

$$\Theta_{t+1} = \Theta_t - \alpha \nabla_\Theta \hat{E}(\Theta_t)$$

$$\hat{E}_t(\Phi, \Psi, \Omega, W) = 1/|S_t| \sum_{n \in S_t} loss(y_n, \ell_n)$$

random subset of data

Massive N

- Choose a **random** data subset
- Estimate gradient by data point
- Update parameters using gradient from random subset
- Leads to similar solutions at faster rate