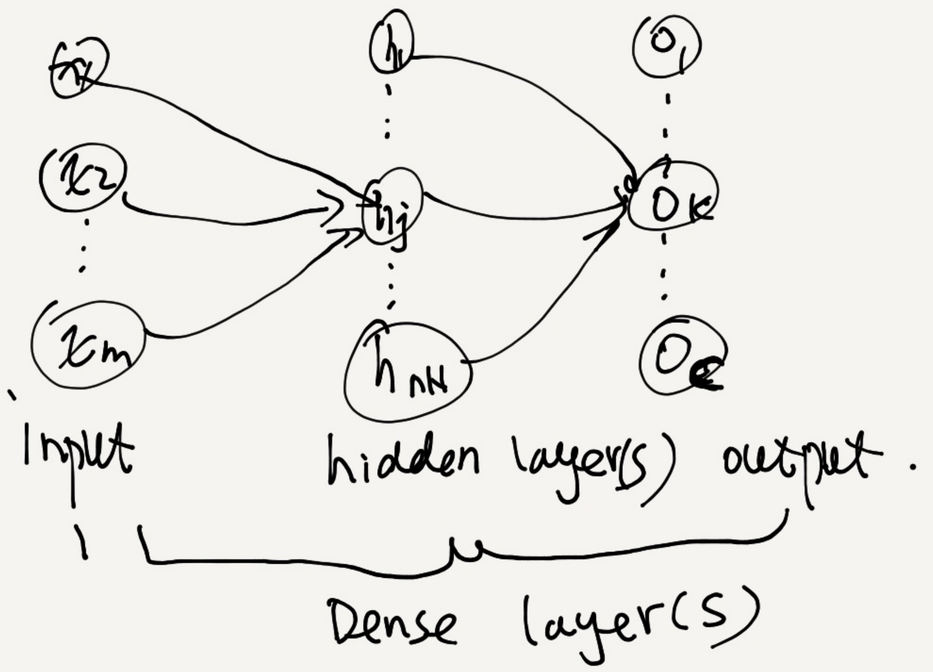
Convolutional Neural Networks

1. Fully connected NNs (FCNNs)



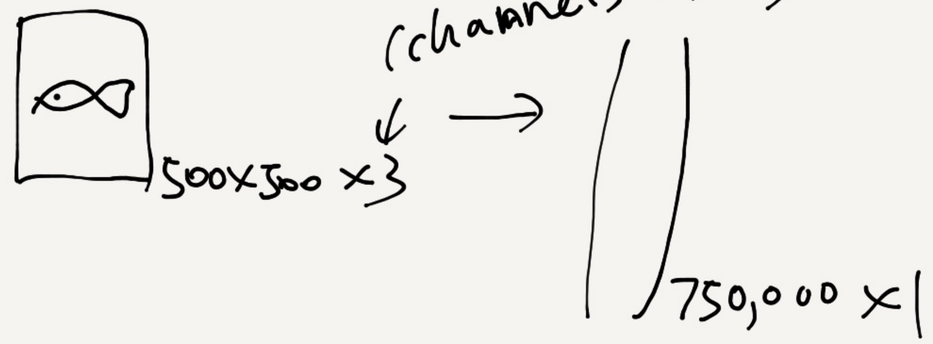
weights for this 3-layer NN:

$$m \times n_H + n_H \times C$$

$$\simeq n_H \times (m + c) \quad \text{(w/o bias)}$$

Drawbacks of FCNNs:

1) if m (features) is large, the # weights is large too,



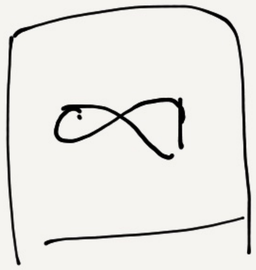
2) In DL, we may have many hidden layers, which leads to large n_H.

ResNet (2015, 150 layers)

2. How can we reduce the # of model parameters.?

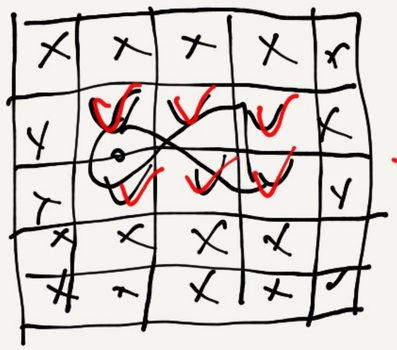
① reduce m.

feature engineering to produce high-level features.



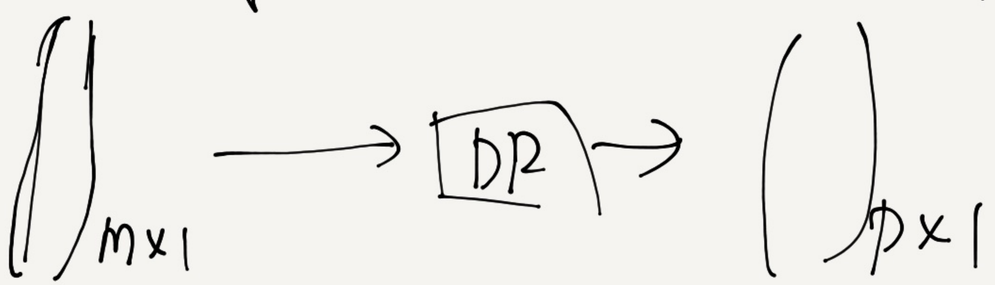
→ extract features of objects in the images and create a new feature vector

$$f = \begin{pmatrix} F_1 \\ f_2 \\ \vdots \\ F_p \end{pmatrix} \quad p \ll m.$$

② 

→ locate objects and only use object-related grids

③ Dimensionality - reduction (DR) methods, → (automatic process)

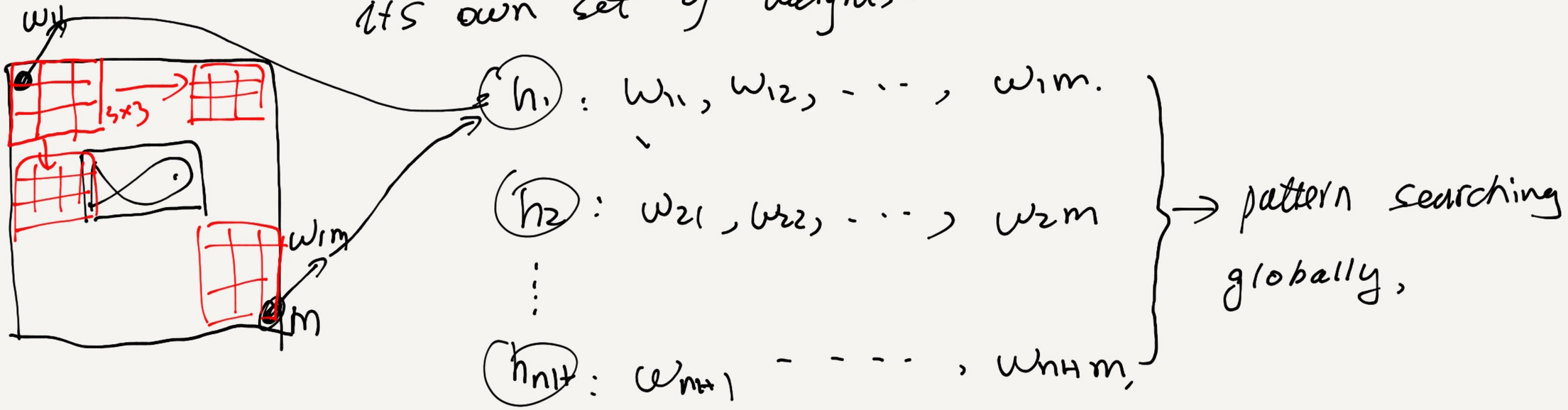


$p \ll m$

PCA, manifold learning (non-linear DR approaches)

④ down sampling: we may do not need to have high resolution images to recognize objects in images.

⑤. Hidden nodes relate to the patterns of images, each hidden node has its own set of weights.

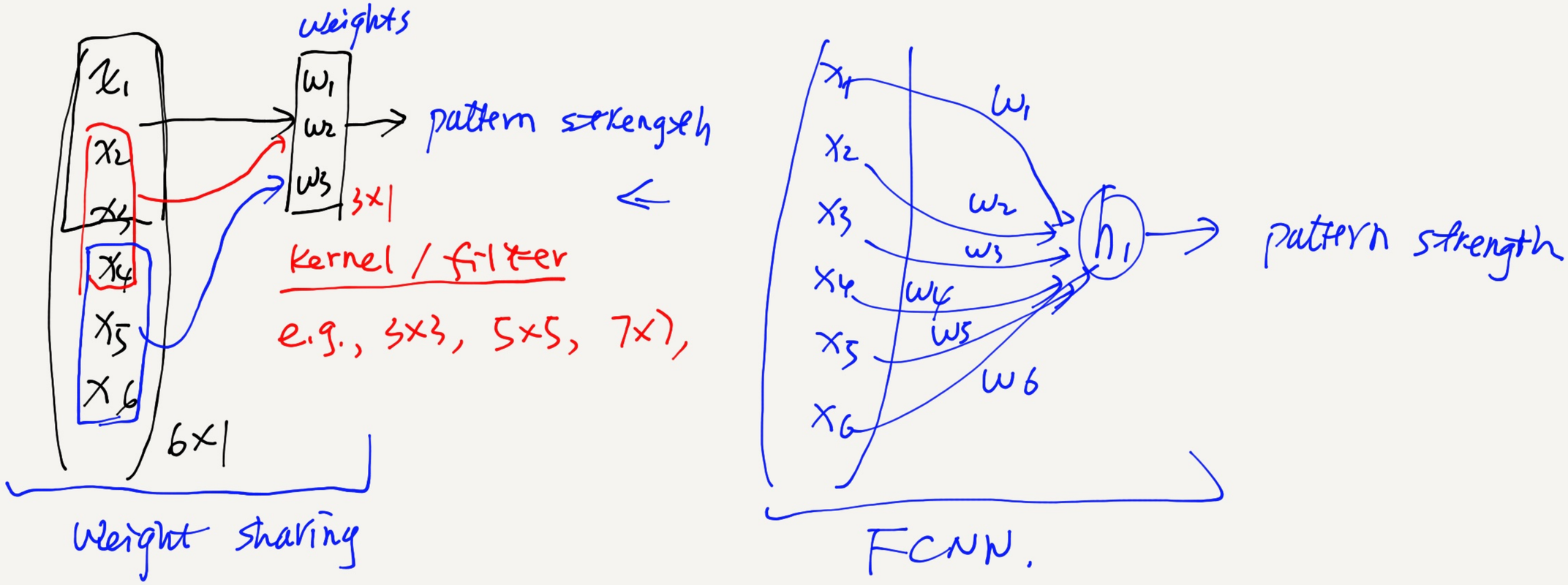


patterns are usually defined in a small image region.

Image regions share weights.

⇓ Can we search locally using a small set of weights. → sliding window. → # of weights = window size.

# of connections / parameters to. 9. ← 3×3


weights
x_1
x_2
x_4
x_4
x_5
x_6
w_1
w_2
w_3
3×1
pattern strength
kernel / filter
e.g., 3×3, 5×5, 7×7,
6×1
Weight sharing
x_1
x_2
x_3
x_4
x_5
x_6
w_1
w_2
w_3
w_4
w_5
w_6
h_1
pattern strength
FCNN.