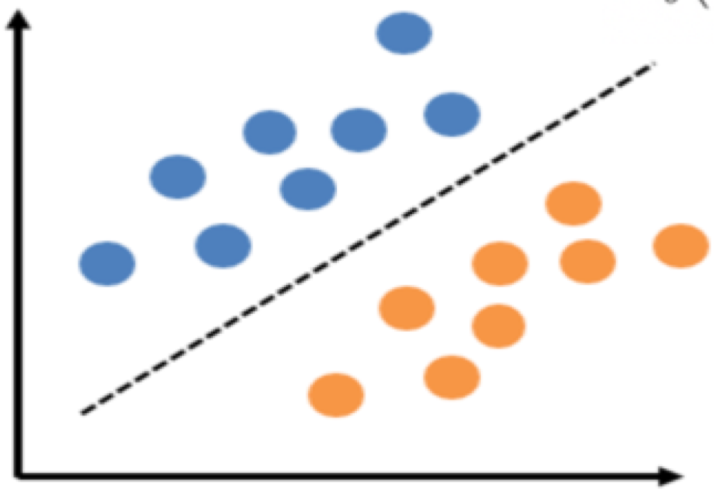
Separability: Linear and Non-linear

Linear

$$h_\theta(x) = \theta_0 + \theta_1 x$$

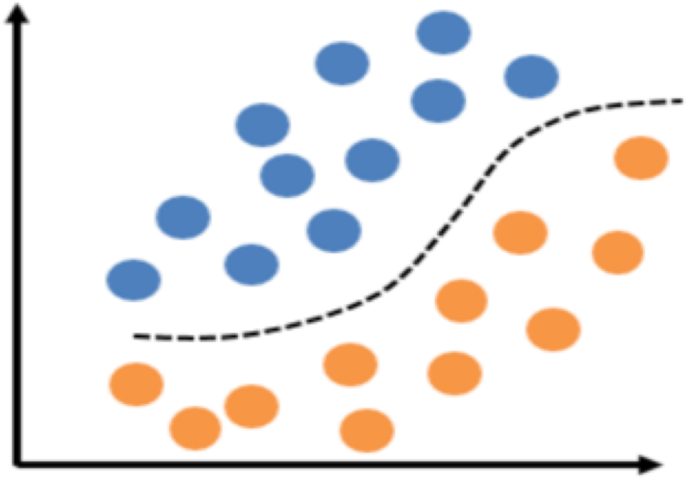


One variable for linearly separable

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Nonlinear



Multiple variables for separability

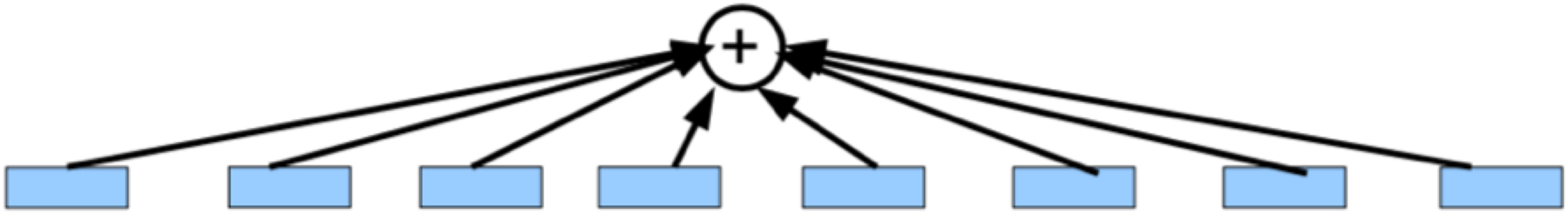
Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

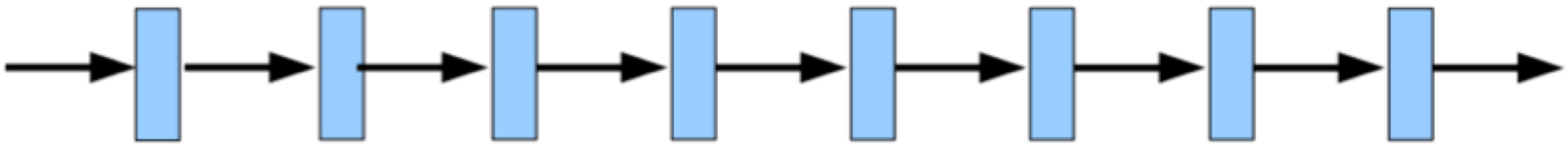
Source: https://jtsulliv.github.io/perceptron/; http://cs229.stanford.edu/syllabus.html

Learning Features

Linear combination : $f(x) \approx \sum_j g_j$

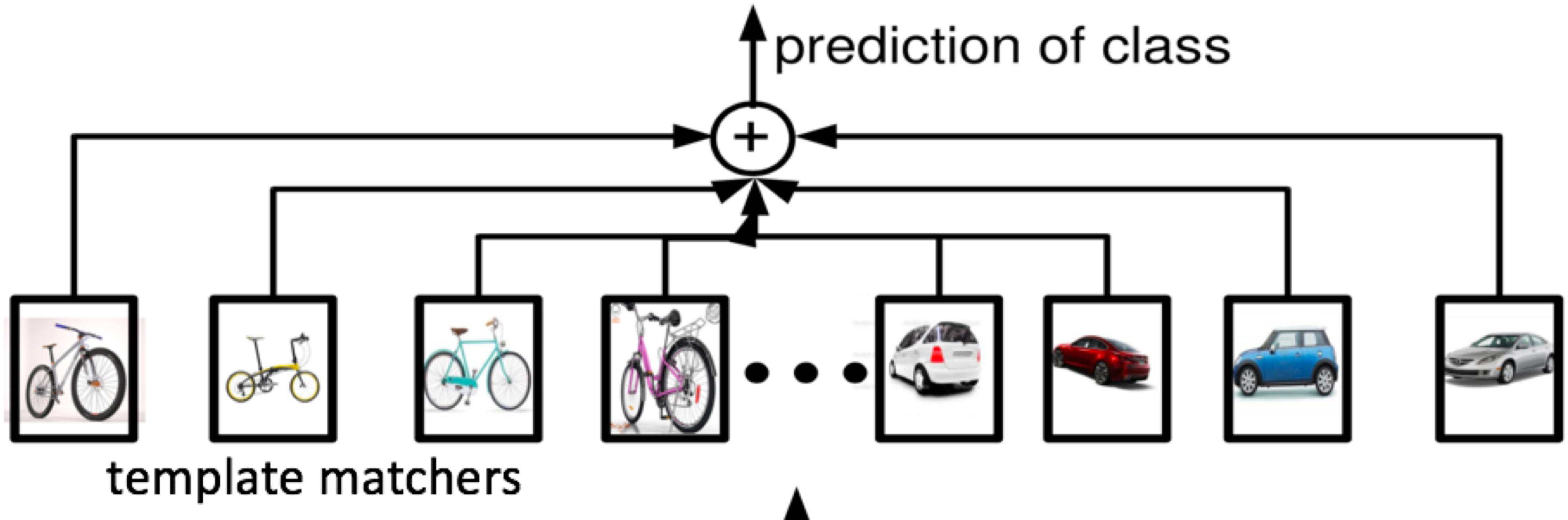


Composition : $f(x) \approx g_1(g_2(\dots g_n(x)\dots))$



Source: Fergus, Ranzato

Linear Combination

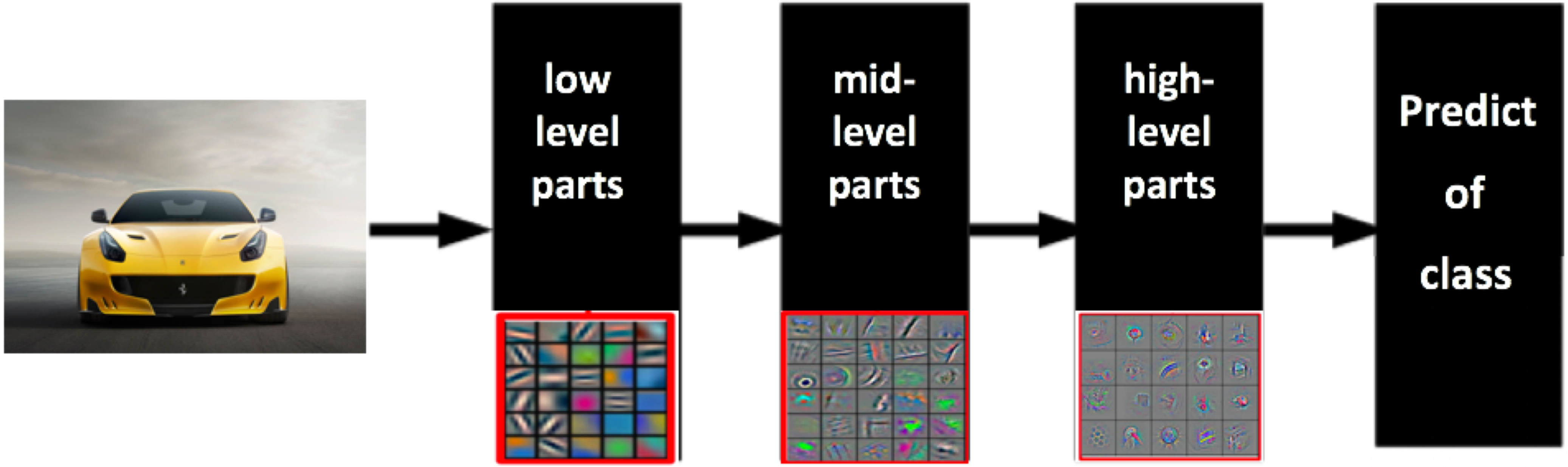


Shallow !

$$f(x) \approx \sum_j g_j$$

Source: Fergus, Ranzato

Composition



- **Advantage: intermediate concepts can be re-used**
- Given lots of data => **engineer less** and learn more!

Deep !

$$f(x) \approx g_1(g_2(\ldots g_n(x)\ldots))$$

Source: Fergus, Ranzato

Linear Regression

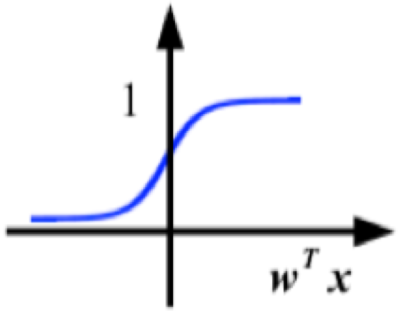
Hypothesis => Logistic Regression: **Linear Classifier**

$\boxed{\text{Input: } \boldsymbol{x} \in R^D}$

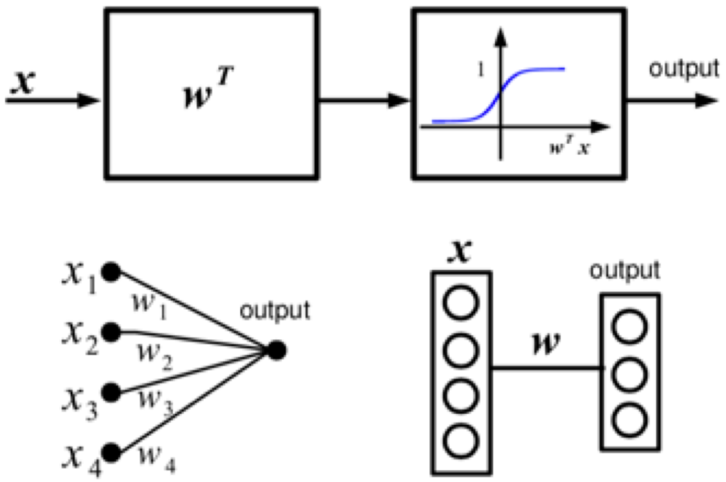
$\boxed{\text{Binary label: } y \in \{-1,+1\}}$

Parameters: $\boldsymbol{w} \in R^D$

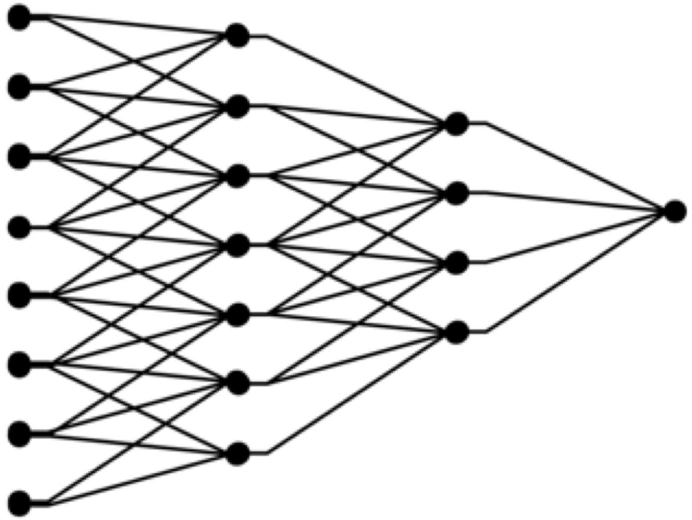
Output prediction: $p(y=1|\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{w}^T \boldsymbol{x}}}$



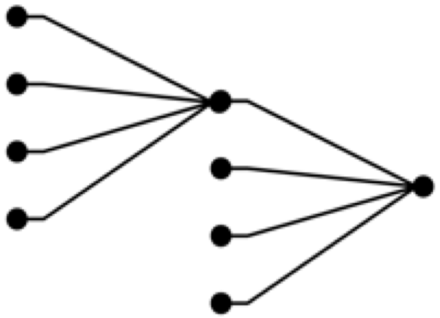
1 Graphical Representation



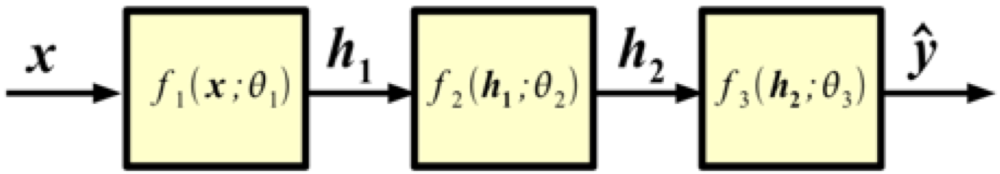
3 Logistic Regression → Neural Nets



2 Logistic Regression → Neural Nets



4 Neural Nets



NOTE: In practice, each module does **NOT** need to be a logistic regression classifier

Figure credit: Fergus, Ranzato